



CACS-205

Web Technology

(BCA, TU)

Ganesh Khatri
kh6ganesh@gmail.com

Architectural Issues of Web Layer

- The web layer is also referred to as the UI layer
- web layer is primarily concerned with presenting the user interface and the behavior of the application (handling user interactions/events)
- the web layer can also contain logic, core application logic is usually located in the services layer
- three Layers within the Web Layer are
 - HTML
 - CSS
 - JavaScript

Architectural Issues of Web Layer

- HTML(Content Layer) :
 - stores all the content that your customers want to read or look at.
 - includes text and images as well as multimedia
 - It's also important to make sure that every aspect of your site is represented in the content layer
- CSS(Style Layer) :
 - Stores all your styles for your web site to decorate your web page
 - defines the way the pages should look, and you can have separate style sheets for various media types
 - Store your CSS in an external style sheet so that you can get the benefits of the style layer across the site
 - Types – inline, internal and external

Architectural Issues of Web Layer

- JavaScript(Behavior Layer) :
 - JavaScript is the most commonly used language for writing the behavior layer
 - Used to make any web page more interactive with the users
 - You use this layer to interact directly with the DOM or Document Object Model

Benefits of Separating the layers

- When you're creating a Web page, it is important to keep the layers separate
- Using external style sheets is the best way to separate your content from your design
- And the same is true for using external JavaScript files.
- Some of the benefits of separating the layers are
 - Shared Resources
 - Faster downloads
 - Multi-person teams
 - Accessibility
 - Backwards compatibility

Benefits of Separating the layers

- Shared Resources
 - When you write an external CSS file or JavaScript file, you can use that file by any page on your Web site
 - There is no duplication of effort, and whenever the file changes, it changes for every page that uses it without you making more than one change.
- Faster Downloads
 - Once the script or stylesheet has been downloaded by your customer the first time, it is cached
 - Then every other page that is downloaded loads more quickly in the browser window

Benefits of Separating the layers

- Multi-Person Teams

- If you have more than one person working on a Web site at once, you can divide up the workload without worrying about permissions or content management
- You can also hire people who are style/design experts to work on the CSS while your scripters work on the JavaScript, and your writers work in the content files.

- Accessibility

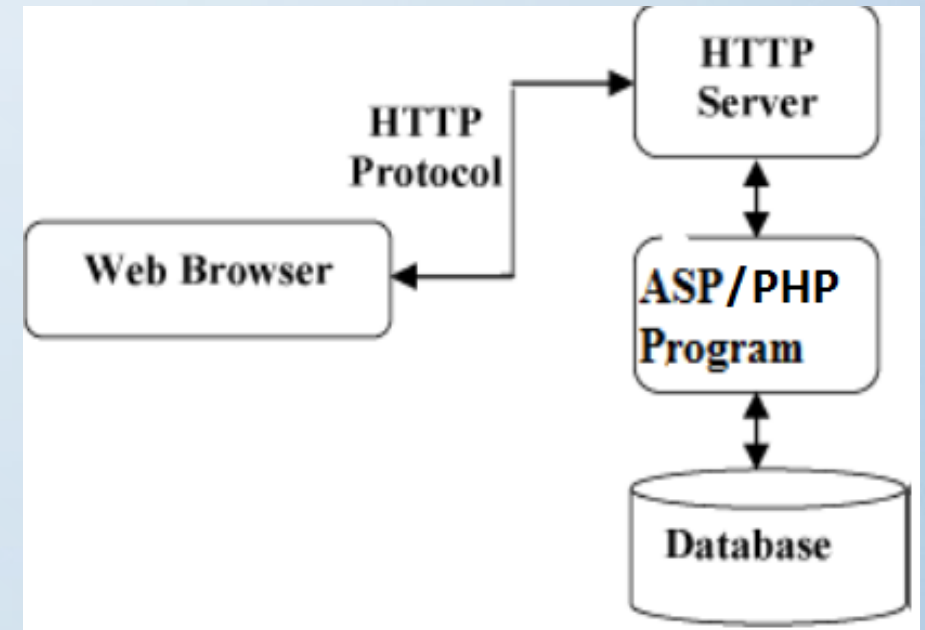
- External style sheets and script files are more accessible to more browsers, because they can be ignored more easily, and because they provide more options
- For example, you can set up a style sheet that is displayed only for screen readers or a script library that's only used by people on cell phones

Benefits of Separating the layers

- Backward Compatibility
 - When you have a site that is designed with the development layers,
 - it will be more backwards compatible because browsers that can't use technology like CSS and JavaScript can still view the HTML

HTTP(Hypertext Transfer Protocol)

- TCP/IP based communication protocol
- used to deliver different resources on the WWW.
- resources could be HTML files, image files, query results, or anything else
- browser works as an HTTP client because it sends requests to an HTTP server which is called Web server
- default port for HTTP servers to listen on is 80
- Features include –
 - Connectionless
 - Media independent
 - Stateless – result of being connectionless



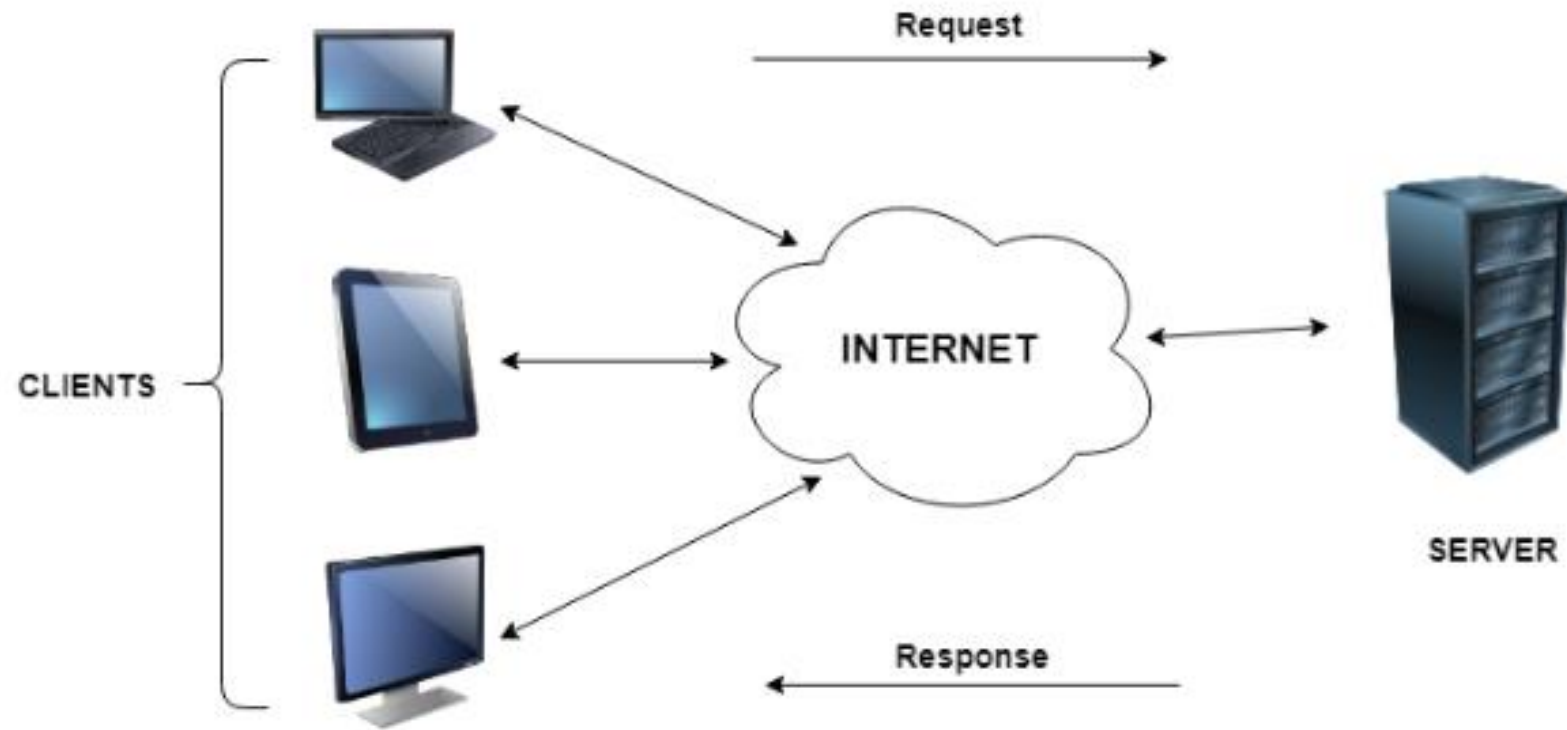
Features of HTTP

- HTTP is Connectionless
 - After a request is made, the client disconnects from the server and waits for a response.
 - The server must re-establish the connection after it processes the request.
- HTTP is media independent:
 - Any type of data can be sent by HTTP as long as both the client and server know how to handle the data content.
 - How content is handled is determined by the MIME specification
- HTTP is stateless:
 - This is a direct result of HTTP's being connectionless.
 - The server and client are aware of each other only during a request.
 - Afterwards, each forgets the other.
 - For this reason neither the client nor the browser can retain information between different requests across the web pages

HTTP

- HTTP is based on the request-response communication model
 - Client sends a request
 - Server sends a response
- Normally implemented over a TCP connection (80 is standard port number for HTTP)
- The following is the typical browser-server interaction using HTTP
 - User enters Web address in browser
 - Browser uses DNS to locate IP address
 - Browser opens TCP connection to server
 - Browser sends HTTP request over connection
 - Server sends HTTP response to browser over connection
 - Browser displays body of response in the client area of the browser window

Client – Server Model(Architecture)



Client - Server Architecture

- Client/server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client
- it has one or more clients connected to a central server over a network or internet connection
- is also known as a networking computing model or client/server network because all the requests and services are delivered over a network
- WWW is based on this architecture
- Client/server architecture is a producer/consumer computing architecture where the server acts as the producer and the client as a consumer

Client - Server Architecture : Advantages

- All the required data is concentrated in a single place i.e. the server.
- So it is easy to protect the data and provide authorization and authentication
- The server need not be located physically close to the clients. Yet the data can be accessed efficiently
- It is easy to replace, upgrade or relocate the nodes in the client server model because all the nodes are independent and request data only from the server
- All the nodes i.e clients and server may not be build on similar platforms yet they can easily facilitate the transfer of data
- Easy to implement security policies, since the data are stored in central location
- Simplified network administration

Client - Server Architecture : Disadvantages

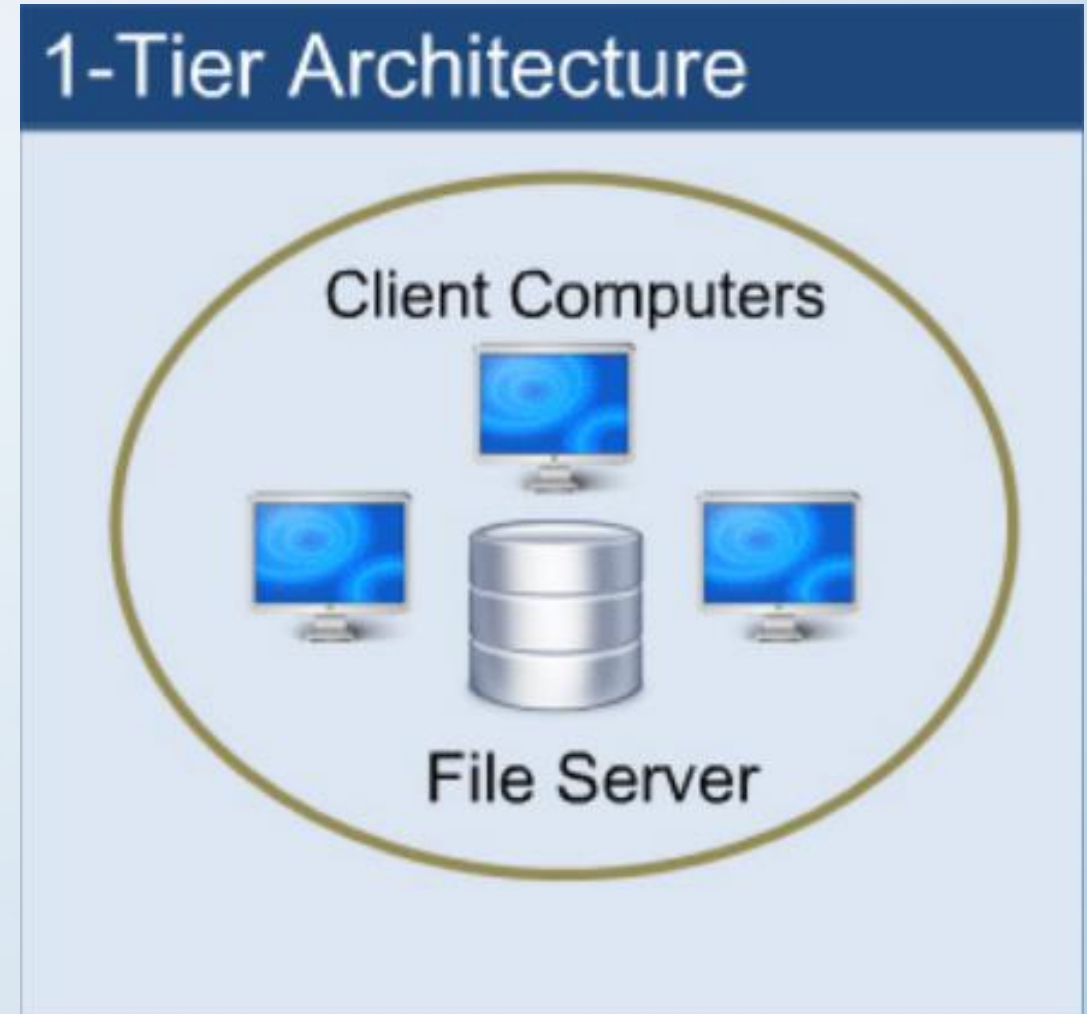
- Failure of the server causes whole network to be collapsed
- Expensive than Peer to Peer, dedicated powerful servers are needed
- Extra effort are needed for administering and managing the server
- If all the clients simultaneously request data from the server, it may get overloaded.

Client - Server Architecture : Types

- Three types :
 - One Tier
 - Two Tier
 - Three and Multi Tier

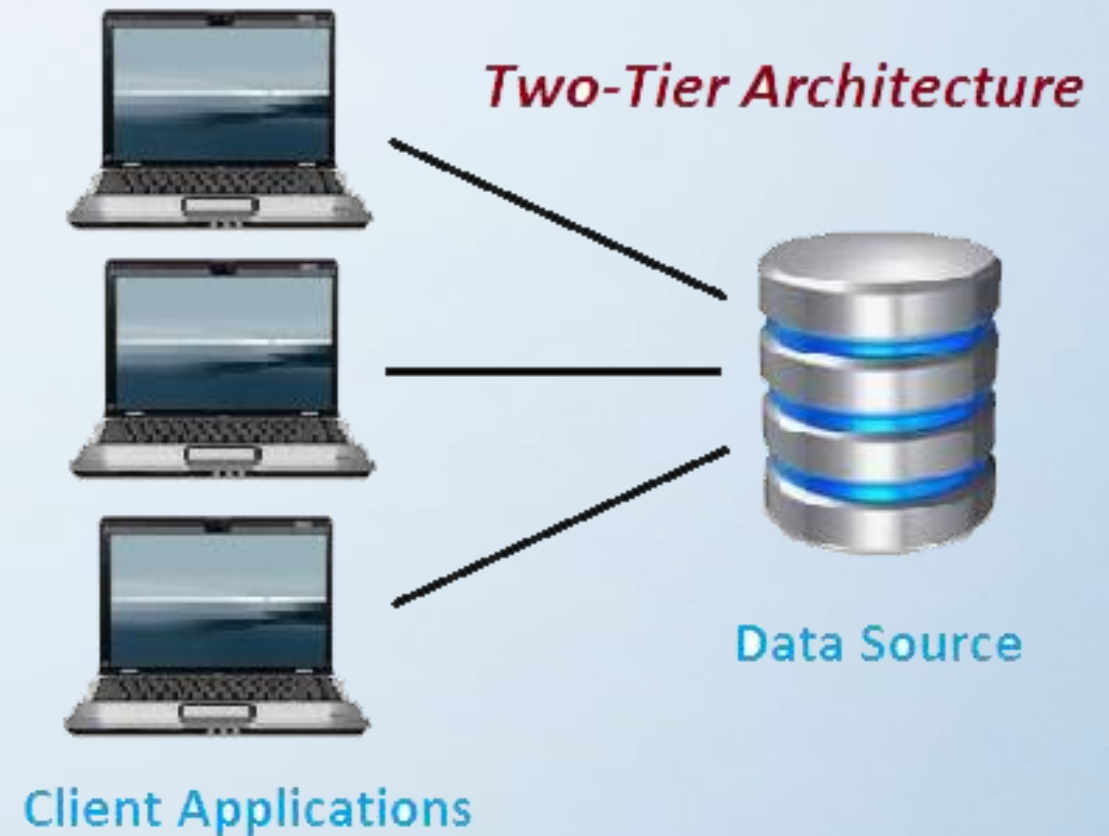
One Tier Architecture

- the user interface, marketing logic and data logic are present in the same system
- This kind of service is reasonable
- Presentation, Business, Data Access layers within a single software package
- The data is usually stored in the local system or a shared drive
- Completely unscalable. Only one user can access the system at a given time via the local client



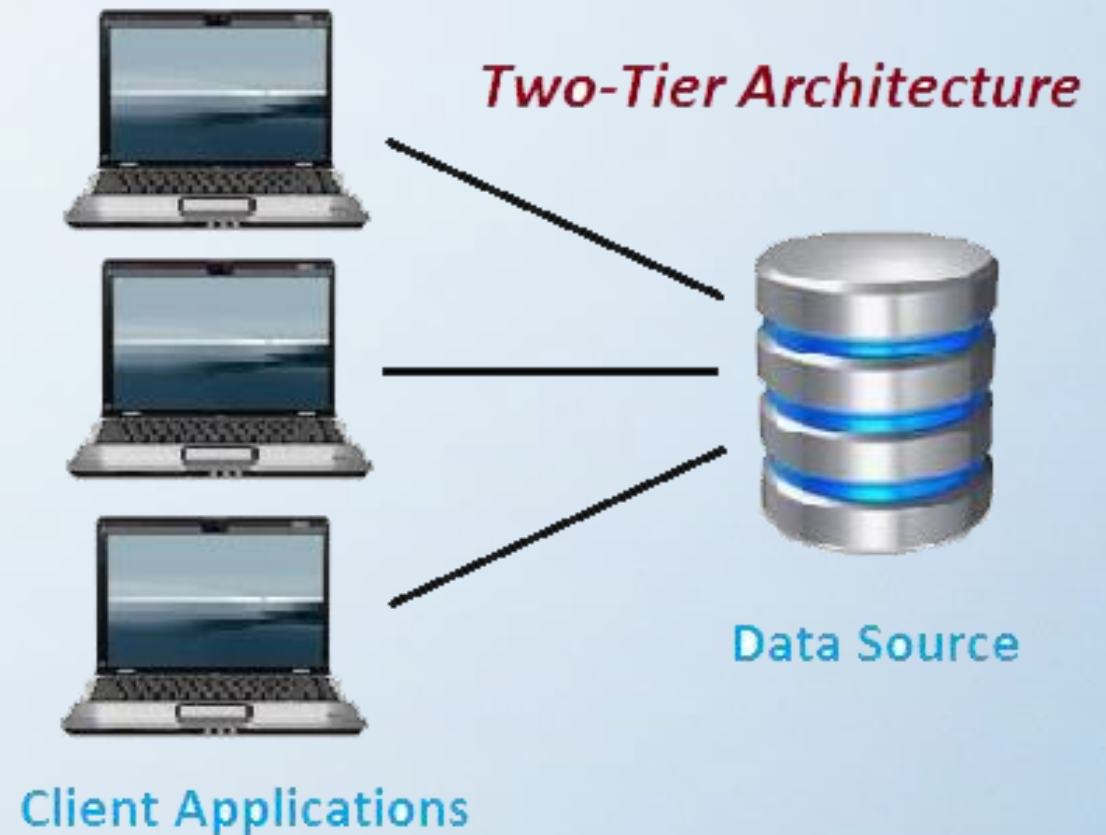
Two Tier Architecture

- the user interface is stored at client machine and the database is stored on the server
- Database logic and business logic are filed at either client or server but it needs to be maintained
- If Business Logic and Data Logic are collected at a client side, it is named as fat client thin server architecture
- If Business Logic and Data Logic are collected at a client side, it is named as fat client thin server architecture



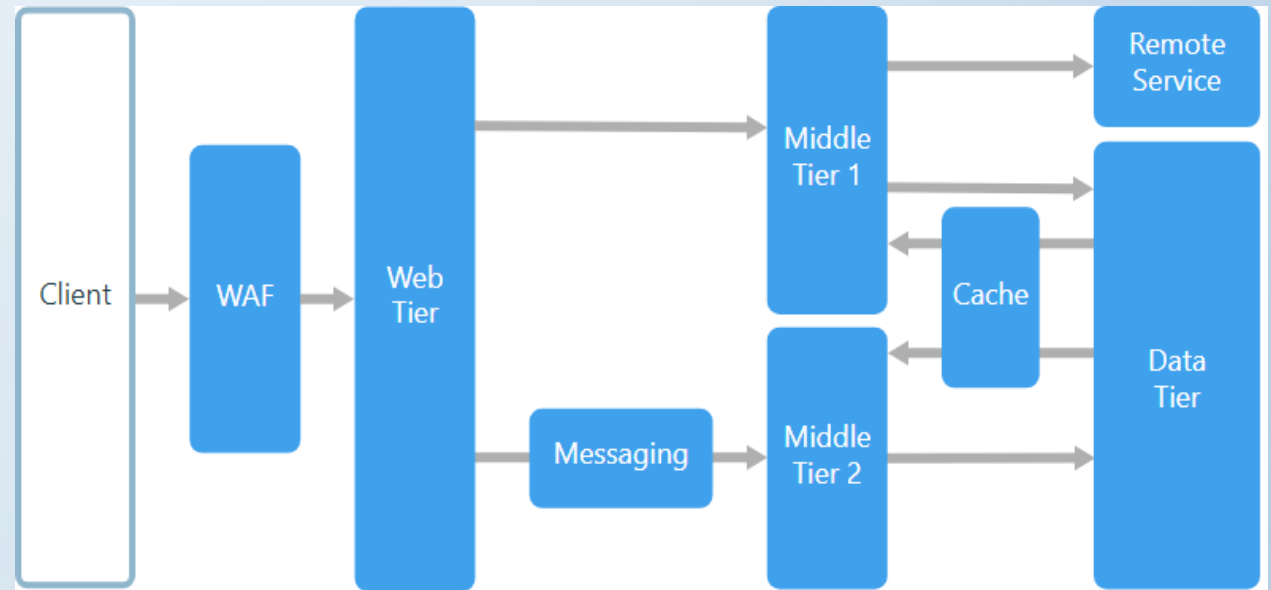
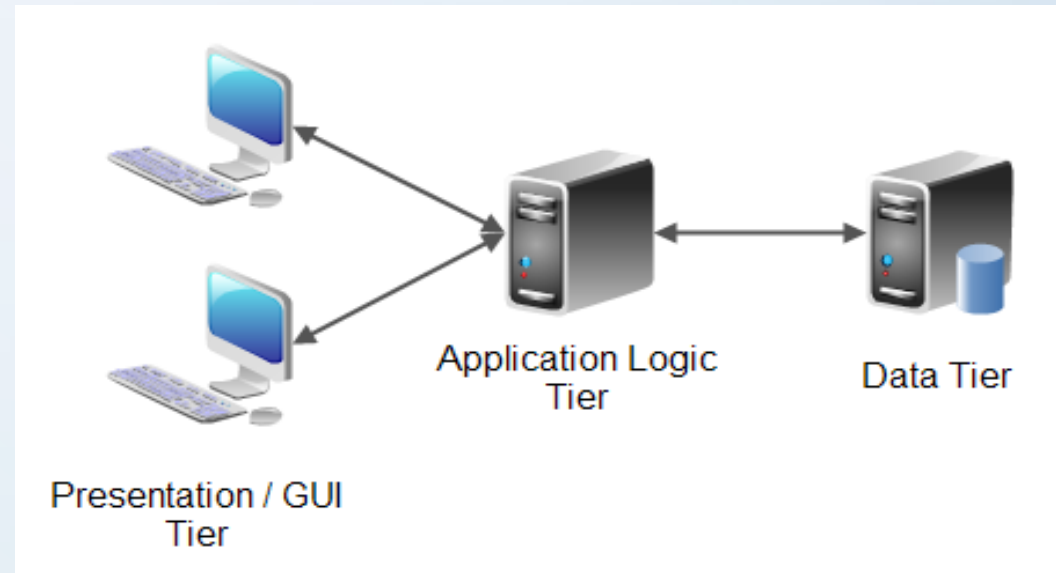
Two Tier Architecture

- in this architecture, client and server have to come in direct incorporation
- If a client is giving an input to the server, there shouldn't be any intermediate
- This is done for rapid results and to avoid confusion between different clients
- It is considered as affordable architecture
- Multiple users can connect to the server at once
- Is not suitable for security reasons



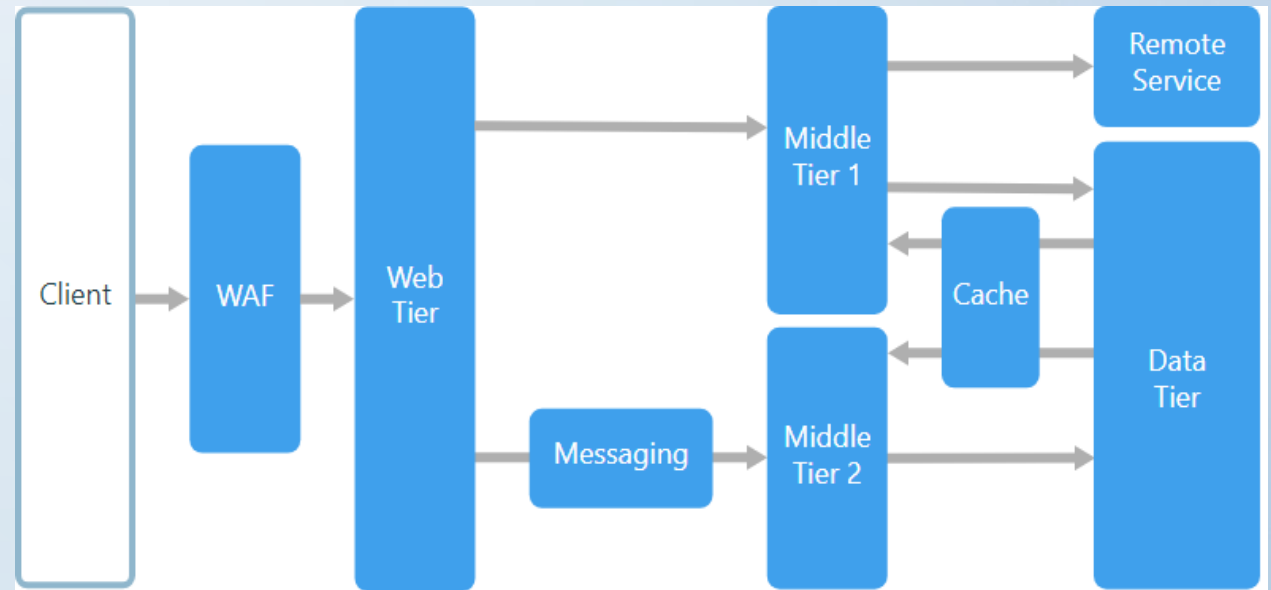
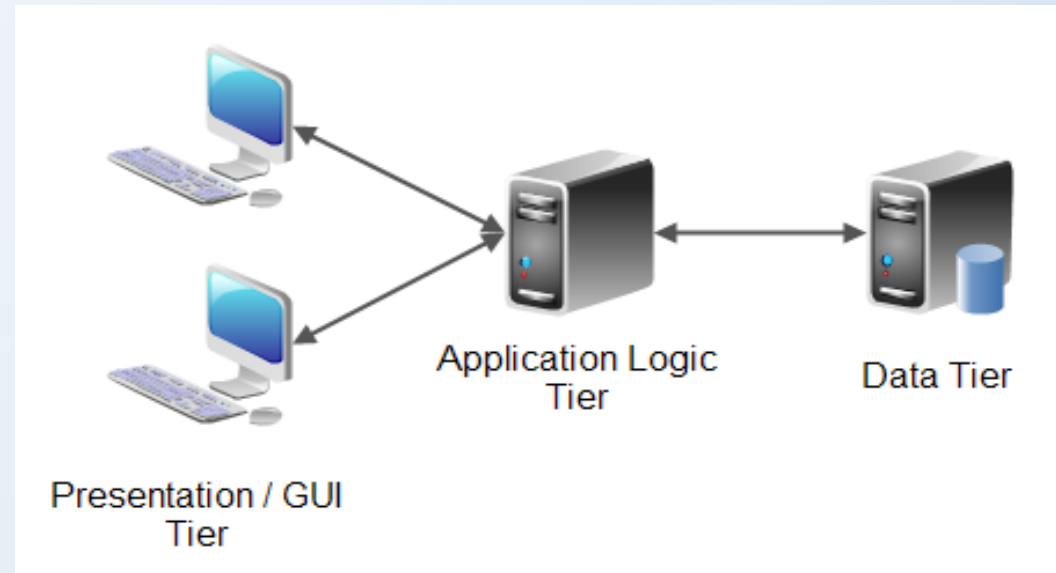
Multi Tier Architecture

- N-tier architecture (with N more than 2) is really 3 tier architecture in which the middle tier may or may not be split up into new tiers
- The application tier is broken down into separate parts
- 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



Multi Tier Architecture

- 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



What kind of system can benefit?

- Answer: N tier because,
 - application logic is partitioned among various servers
 - application partitioning creates an integrated information infrastructure which enables consistent, secure, and global access to critical data
 - significant reduction in network traffic, which leads to faster network communications, greater reliability
 - greater overall performance is also made possible in an N-Tier Client/Server architecture

Client Side Scripting

- refers to writing the class of computer programs (scripts) on the web that are executed at client-side, by the user's web browser, instead of server-side (on the web server)
- Usually scripts are embedded in the HTML page itself
- JavaScript , VBScript, Jscript, Java Applets etc. are the examples of client side scripting technologies
- JavaScript is probably the most widely used client-side scripting language
- Client-side scripts have greater access to the information and functions available on the user's browser, whereas server-side scripts have greater access to the information and functions available on the server
- Client-side scripts may also contain instructions for the browser to follow in response to certain user actions, (e.g., clicking a button)
- Often, these instructions can be followed without further communication with the server

Server Side Scripting

- Includes writing the applications executed by the server at run-time to process client input or generate document in response to client request
- So server side script consists the directives embedded in Web page for server to process before passing page to requestor
- usually used to provide interactive web sites that interface to databases or other data stores
- This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript
- primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores
- Examples : PHP, ASP, Perl, Java, Python etc.

Web 1.0 vs Web 2.0 vs Web 3.0

- Web 1.0
 - It is the “readable” phrase of the World Wide Web with flat data
 - there is only limited interaction between sites and web users
 - Web 1.0 is simply an information portal where users passively receive information without being given the opportunity to post reviews, comments, and feedback
- Web 2.0
 - It is the “writable” phrase of the World Wide Web with interactive data
 - Unlike Web 1.0, Web 2.0 facilitates interaction between web users and sites
 - so it allows users to interact more freely with each other
 - Web 2.0 encourages participation, collaboration, and information sharing
 - Eg. Youtube, Wiki, Flickr, Facebook, and so on

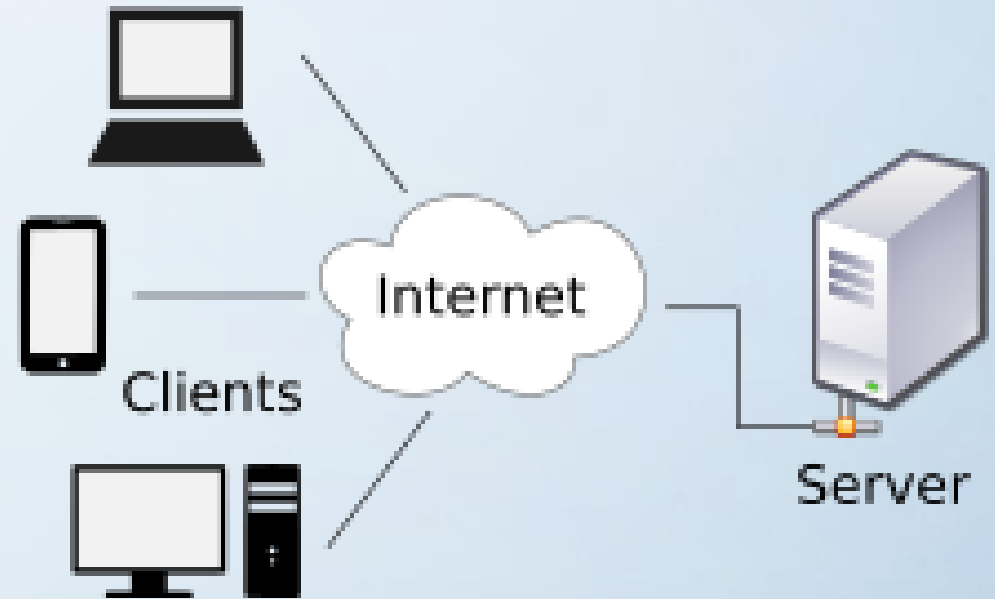
Web 3.0

- It is the “executable” phrase of Word Wide Web with dynamic applications, interactive services, and “machine-to-machine” interaction
- Web 3.0 is semantic web that refers to the future
- it is Web 2.0 with flavor of artificial intelligence
- computers can interpret information like humans and intelligently generate and distribute useful content tailored to the needs of users
- Example of Web 3.0 is Tivo, digital video recorder
 - It is recording program which can search the web and read what it finds to you based on your preferences
- Google voice search, Siri (Iphone) etc.



Web Clients

- is an electronic device or computer or computer program that requests a web server for web resources and server sends resource back to the client
- A web browser can be considered as a web client



Web Servers

- is a computer or computer software that listens and responds to a client computer's request made through a web browser
- in other words, a web server is server software, or hardware dedicated to running said software, that can satisfy WWW client requests
- machine that hosts web pages and other web documents
- provides web documents and other online services using HTTP
- web server can, in general, contain one or more websites



Web Server

Web Servers

- processes incoming network requests over HTTP and several other related protocols
- primary function of a web server is to store, process and deliver web pages to web clients
- communication between client and server takes place using the Hypertext Transfer Protocol (HTTP)
- pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to the text content



Web Server

Functions of a Web Server

- Stores and secures website data
 - store all website data and secures it from unauthorized users when it is properly configured
- Provides web database access
 - provide access to websites that are hosted
 - web hosting service providers own some web servers that are used in variable ways to provide different web hosting services, such as backend database servers
- Serve the end user requests
 - accept requests from different users connected over the internet and serve them accordingly
- Bandwidth controlling to regulate network traffic
 - it is a feature available in web server to minimize excess network traffic
 - web Hosts can set bandwidth values to regulate the rate of data transmission over the internet
 - this feature avoids the down time caused by high web traffic

Functions of a Web Server

- Virtual Hosting
 - is a type of web hosting service in which a web server hosts other software based virtual web-servers web sites, data, applications and other services
 - virtualized Web servers do possess this feature to provide virtual hosting
- Server Side Web Scripting
 - this feature of web server enables the user to create dynamic web pages
 - the popular server side scripting languages include PHP, Perl, Ruby, Python, Java, .NET, etc.

Load Limits of a Web Server

- Load limit of a web server is a maximum number of concurrent client connections per IP address that can be handled by it.
- it is usually between 2 and 80,000, by default between 500 and 1,000 per IP address (and TCP port)
- a server can serve only a certain maximum number of requests per second(RPS) depending on
 - its own settings
 - the HTTP request type
 - whether the content is static or dynamic
 - whether the content is cached
 - hardware and software limitations of the OS of the server computer
- When a web server is near to or over its limit, it becomes unresponsive

Causes of Server Overload

- Excess legitimate web traffic
 - thousands or even millions of clients connecting to the web site in a short interval
- Distributed Denial of Service attacks
 - A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer or network resource unavailable to its intended users
- XSS Worms
 - can cause high traffic because of millions of infected browsers or web servers



Causes of Server Overload

- Internal Bots
 - Traffic not filtered/limited on large web sites with very few resources (bandwidth, etc.)
- Web servers partial unavailability
 - This can happen because of required or urgent maintenance or upgrade, hardware or software failures, back-end (e.g., database) failures, etc.
 - in these cases, the remaining web servers get too much traffic and become overloaded



Server Anti - Overload Techniques

- Managing network traffic, by using
 - Firewalls to block unwanted traffic coming from bad IP sources or having bad patterns
 - HTTP traffic managers to drop, redirect or rewrite requests having bad HTTP patterns
 - Bandwidth management and traffic shaping, in order to smooth down peaks in network usage
- Deploying web cache techniques
- Using different domain names or IP addresses to serve different (static and dynamic) content by separate web servers, e.g.
 - <http://images.example.com>
 - <http://example.com>



Server Anti - Overload Techniques

- Using different domain names or computers to separate big files from small and medium-sized files; the idea is to be able to fully cache small and medium-sized files and to efficiently serve big or huge (over 10 – 1000 MB) files by using different settings
- Using many internet servers (programs) per computer, each one bound to its own network card and IP address
- Using many internet servers (computers) that are grouped together behind a load balancer so that they act or are seen as one big web server
- Adding more hardware resources (i.e. RAM, disks) to each computer
- Using more efficient computer programs for web servers, etc.

