

# Web Technology

## BIT 3rd Semester

---

Ashish Gautam  
Sr Software Engineer  
Hamro Patro

# Introduction to course

[https://docs.google.com/document/d/1z3xR3\\_t2FoAaoE4abLZiiAq7eMX6mQX-dl2ixdCdwu8/edit?usp=sharing](https://docs.google.com/document/d/1z3xR3_t2FoAaoE4abLZiiAq7eMX6mQX-dl2ixdCdwu8/edit?usp=sharing)

# Chapter I

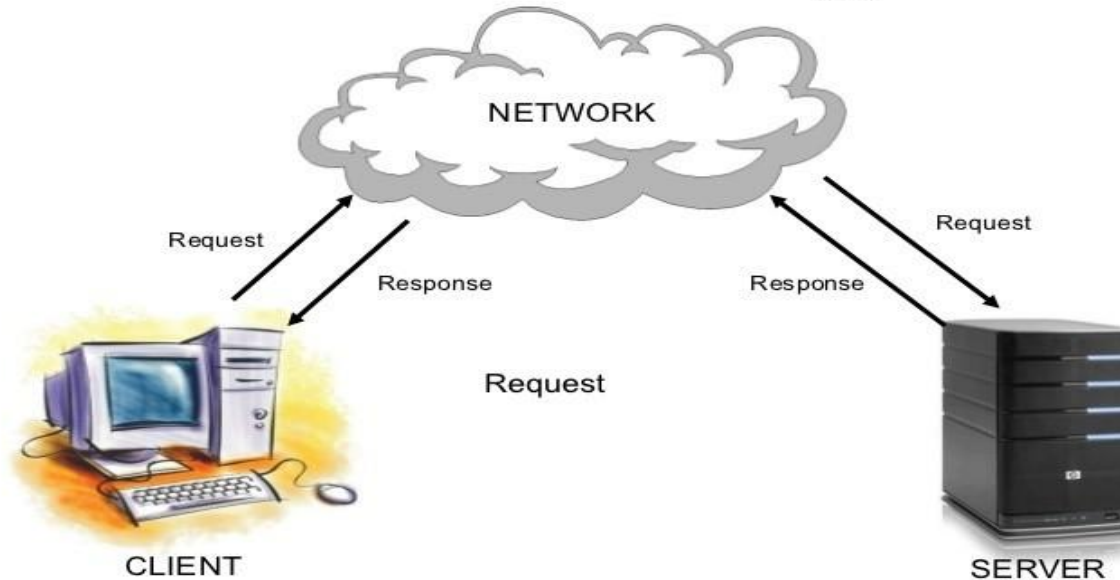
# What is Web Technology?

You probably know that computers don't communicate with each other the way that people do. Instead, computers require codes, or directions. These binary codes and commands allow computers to process needed information. Every second, billions upon billions of ones and zeros are processed in order to provide you with the information you need.

So what does that have to do with your ability to post your latest pictures online? Everything.

The methods by which computers communicate with each other through the use of markup languages and multimedia packages is known as **web technology**. In the past few decades, web technology has undergone a dramatic transition, from a few marked up web pages to the ability to do very specific work on a network without interruption. Let's look at some examples of web technology.

# Client-Server Technology



“Web technology” is the set of standards, protocols, and interfaces required to deliver information reliably and efficiently over the Internet to a user with a web browser interface.

# 1. Static web pages

Static Web pages are very simple. It is written in languages such as HTML, JavaScript, CSS, etc. For static web pages when a server receives a request for a web page, then the server sends the response to the client without doing any additional process. And these web pages are seen through a web browser. In static web pages, Pages will remain the same until someone changes it manually.



# What is Web Pages?

- Web pages are what make up the World Wide Web
- a document commonly written in HTML (Hypertext Markup Language) that is accessible through the Internet or other networks using an Internet browser
- is accessed by entering a URL address and may contain text, graphics, and hyperlinks to other web pages and files
- A Web page is a representation of a document that is actually located at a remote site. The information on a Web page is displayed online with the help of a Web browser such as Internet Explorer, Mozilla Firefox or Google Chrome. The Web browser is connected to the Web server, where the website's contents are hosted through HTTP. Every Web page corresponds to various types of information presented to the visitor in a visual and readable manner.

## Additional Note that:

Web page is not the same thing as a Web site. A Web site is a collection of pages. A Web page is an individual HTML document. This is a good distinction to know, as most techies have little tolerance for people who mix up the two terms.

The first web page went live on August 6, 1991. It was dedicated to information on the World Wide Web project and was made by Tim Berners-Lee. It ran on a NeXT computer at the European Organization for Nuclear Research.

The first web page address was <http://info.cern.ch/hypertext/WWW/TheProject.html>.



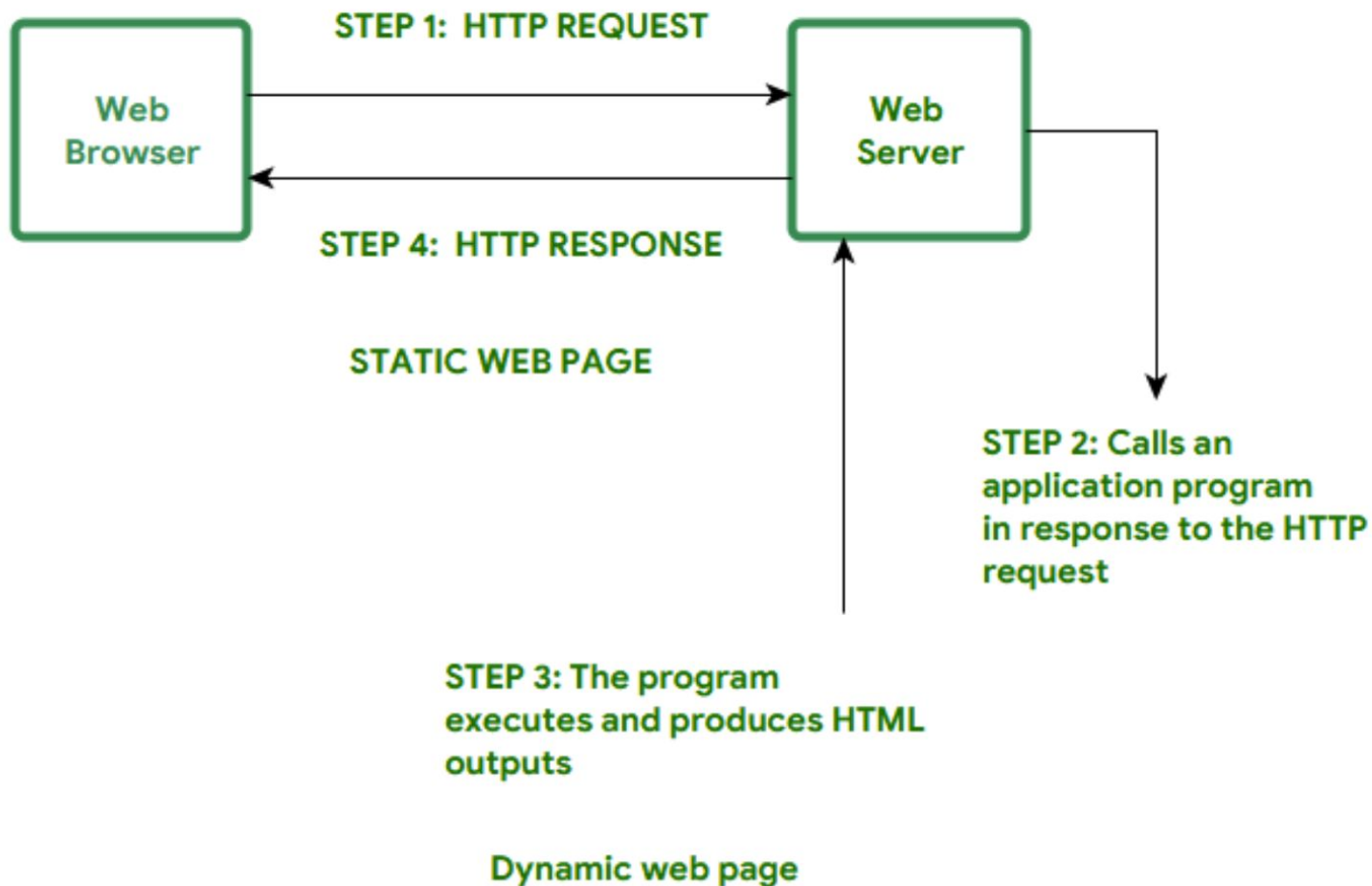
## 2. Dynamic Web Pages

A **dynamic web page** is a **web page** that displays different content each time it's viewed. For example, the **page** may change with the time of day, the user that accesses the webpage, or the type of user interaction.

A dynamic website can contain client-side scripting or server-side scripting to generate the changing content, or a combination of both scripting types. These sites also include HTML programming for the basic structure. The client-side or server-side scripting takes care of the guts of the site.

With client-side HTML scripting, the page can use JavaScript or another scripting language to change the data of page as it is dynamically built.

With server-side scripting, scripts are run on the server that hosts the page. The process for how the page is built is determined by parameters defined in the server-side scripting.



# Dynamic Vs Static Web Pages

SL.NO	STATIC WEB PAGE	DYNAMIC WEB PAGE
1.	In static web pages, Pages will remain same until someone changes it manually.	In dynamic web pages, Content of pages are different for different visitors.
2.	Static Web Pages are simple in terms of complexity.	Dynamic web pages are complicated.
3.	In static web pages, Information are change rarely.	In dynamic web page, Information are change frequently.
4.	Static Web Page takes less time for loading than dynamic web page.	Dynamic web page takes more time for loading.
5.	In Static Web Pages, database is not used.	In dynamic web pages, database is used.
6.	Static web pages are written in languages such as: HTML, JavaScript, CSS, etc.	Dynamic web pages are written in languages such as: CGI, AJAX, ASP, ASP.NET, etc.

# Issues In Web Pages

1. Poor or dated visual design
2. Slow upload speed
3. Lack of quality images
4. Clutter
5. Poor or confusing user journey
6. Low standard of written content
7. Poor standard of SEO
8. Hidden details or no obvious point of contact
9. **Browser issues**
10. **Technology**

# Web Tiers

A tier is like a layer in a cake. A cake can have multiple layers and a Web application can have multiple appears that perform different tasks.

Many Web surfers are familiar with the all-important Web tier, also known as the presentation tier. The presentation tier manages communication between a person's Web browser and a Web server. Your browser, for instance, may contain client-side JavaScript that runs locally in the browser. The Web server, on the other hand, receives requests from the browser and returns responses to it. A Web server can also contain one or more Web services that may communicate with browsers and desktop applications

### **Presentation tier**

This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing and shopping cart contents. It communicates with other tiers by which it puts out the results to the browser/client tier and all other tiers in the network. In simple terms, it is a layer which users can access directly (such as a web page, or an operating system's GUI).

### **Application tier (business logic, logic tier, or middle tier)**

The logical tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing.

### **Data tier**

The data tier includes the data persistence mechanisms (database servers, file shares, etc.) and the data access layer that encapsulates the persistence mechanisms and exposes the data. The data access layer should provide an [API](#) to the application tier that exposes methods of managing the stored data without exposing or creating dependencies on the data storage mechanisms. Avoiding dependencies on the storage mechanisms allows for updates or changes without the application tier clients being affected by or even aware of the change. As with the separation of any tier, there are costs for implementation and often costs to performance in exchange for improved scalability and maintainability.

## Presentation tier

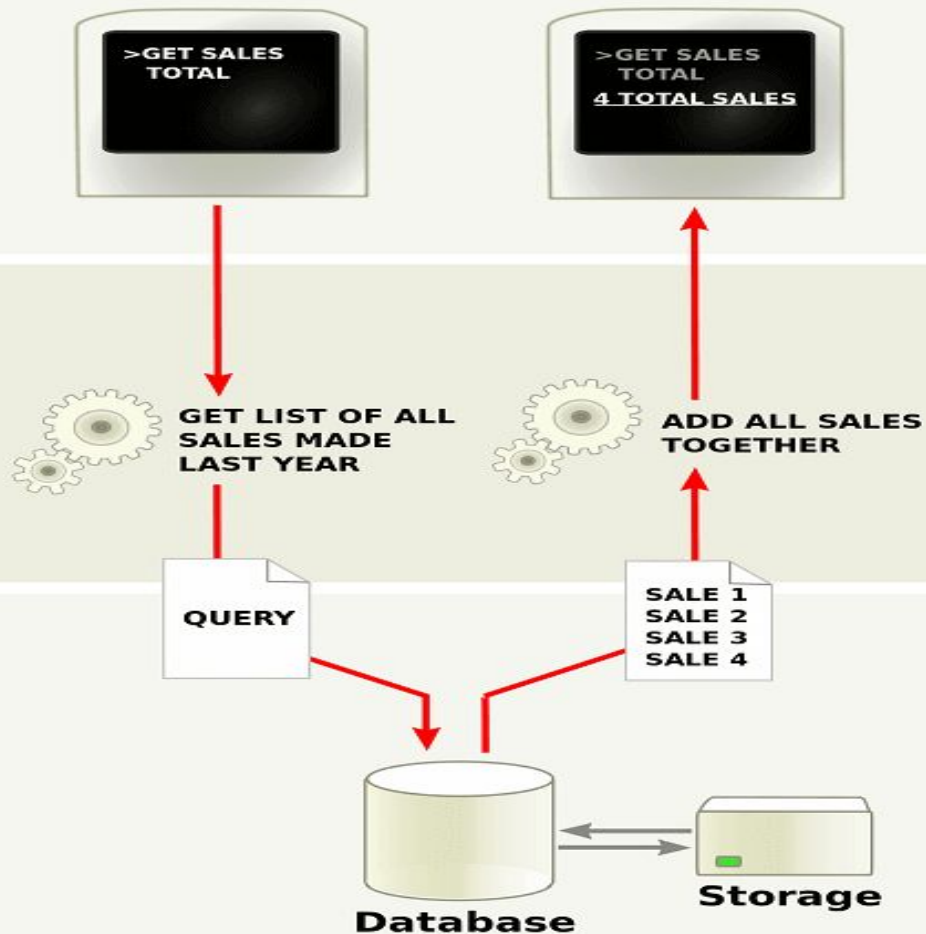
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

## Logic tier

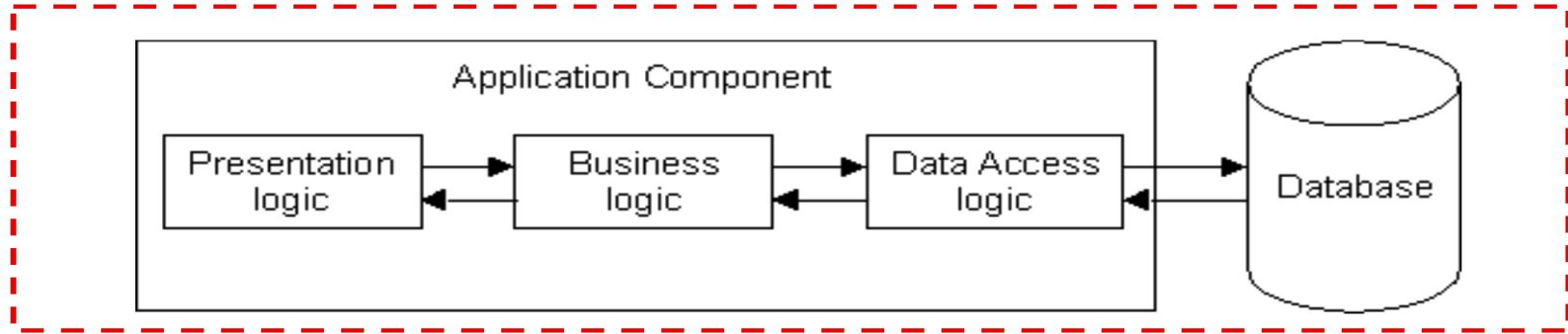
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

## Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



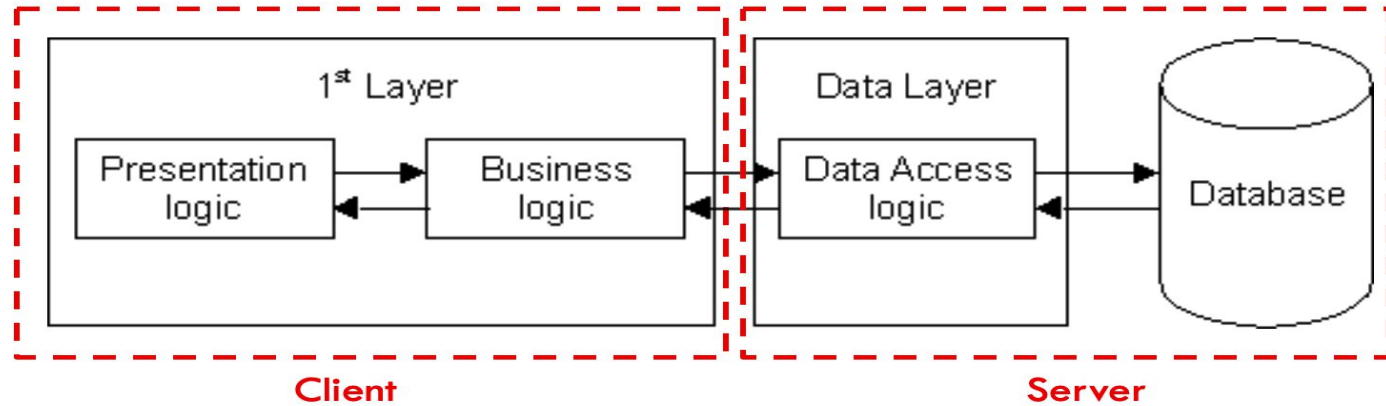
# 1-Tier Architecture



- All 3 layers are on the same machine
  - ▣ All code and processing kept on a single machine
- Presentation, Logic, Data layers are tightly connected
  - ▣ Scalability: Single processor means hard to increase volume of processing
  - ▣ Portability: Moving to a new machine may mean rewriting everything
  - ▣ Maintenance: Changing one layer requires changing other layers



# 2-Tier Architecture

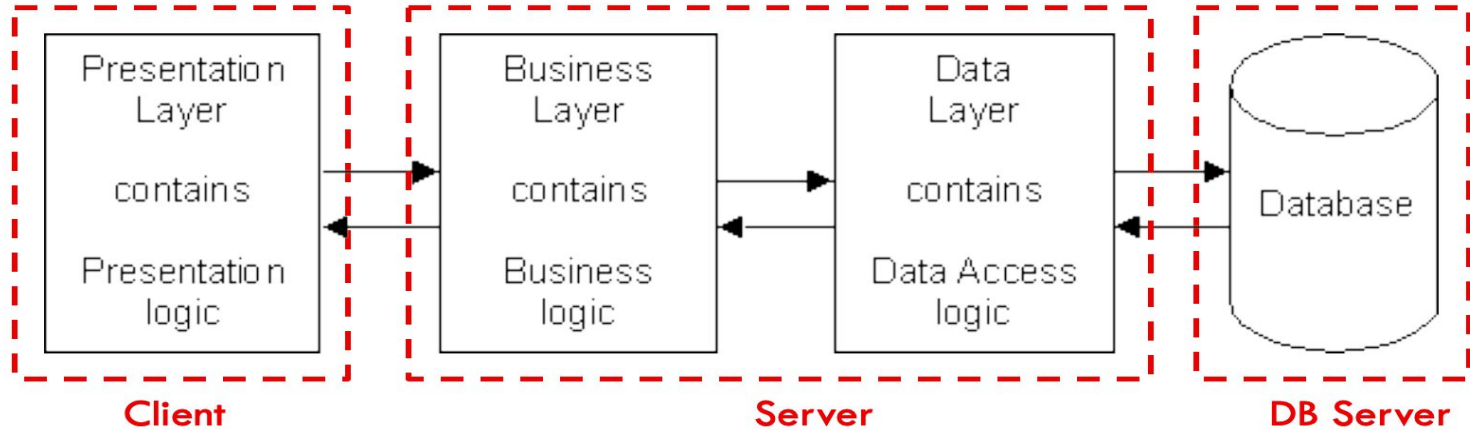


- ❑ Database runs on Server
  - ▣ Separated from client
  - ▣ Easy to switch to a different database
- ❑ Presentation and logic layers still tightly connected
  - ▣ Heavy load on server
  - ▣ Potential congestion on network
  - ▣ Presentation still tied to business logic

# Three-tier architecture

Three-tier architecture is a client-server [software architecture pattern](#) in which the [user interface](#) (presentation), [functional process logic](#) ("business rules"), [computer data storage](#) and [data access](#) are developed and maintained as independent [modules](#), most often on separate [platforms](#).<sup>[10]</sup> It was developed by [John J. Donovan](#) in Open Environment Corporation (OEC), a tools company he founded in [Cambridge, Massachusetts](#).

# 3-Tier Architecture



- Each layer can potentially run on a different machine
- Presentation, logic, data layers disconnected

# Significance of “Tiers”

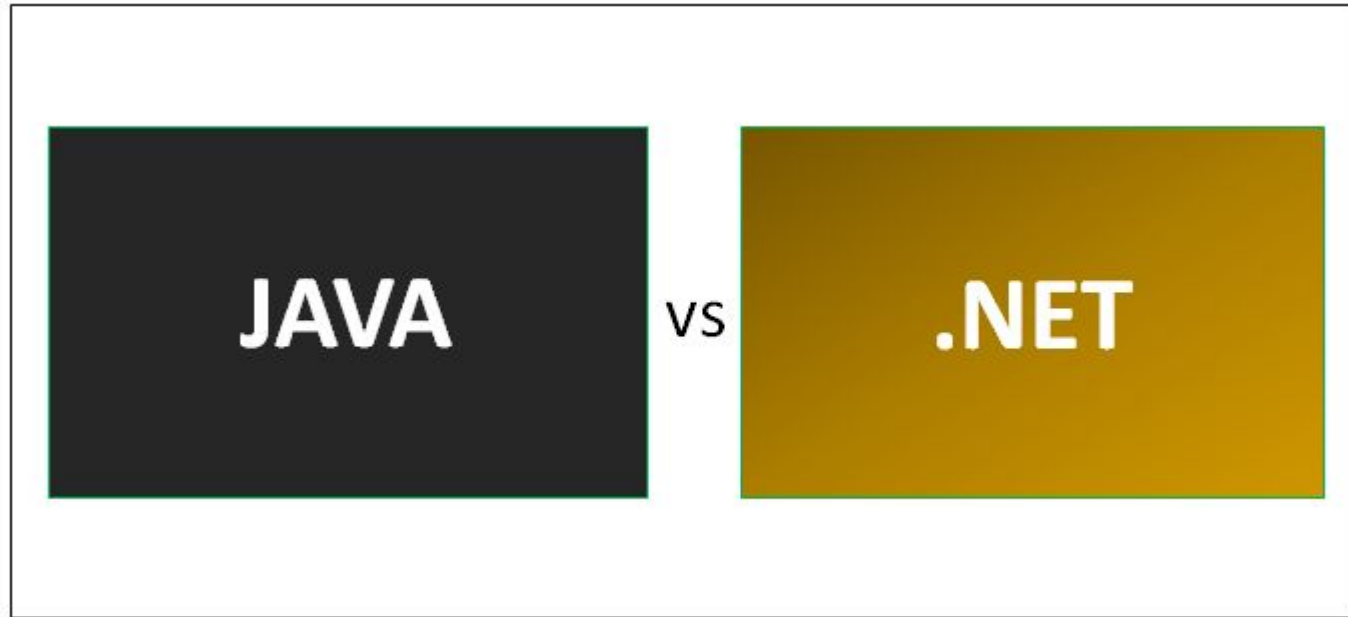
---

- N-tier architectures have the same components
  - ▣ Presentation
  - ▣ Business/Logic
  - ▣ Data
- N-tier architectures try to separate the components into different tiers/layers
  - ▣ Tier: physical separation
  - ▣ Layer: logical separation

# Advantages and Disadvantages of Multi-Tier Architectures

Advantages	Disadvantages
<ul style="list-style-type: none"><li>• Scalability</li></ul>	<ul style="list-style-type: none"><li>• Increase in Effort</li></ul>
<ul style="list-style-type: none"><li>• Data Integrity</li></ul>	<ul style="list-style-type: none"><li>• Increase in Complexity</li></ul>
<ul style="list-style-type: none"><li>• Reusability</li></ul>	
<ul style="list-style-type: none"><li>• Reduced Distribution</li></ul>	
<ul style="list-style-type: none"><li>• Improved Security</li></ul>	
<ul style="list-style-type: none"><li>• Improved Availability</li></ul>	

# Comparisons of Microsoft and java technologies



**The main difference** between Java and .NET is that Java, usually Java Enterprise Edition (JEE), can work on any operating system, while .NET works only on various versions of Windows. While a limited number of open-source implementations of .NET are available, this framework is still targeting Windows users. At the same time, Java takes advantage of the Java Virtual Machine (JVM), which follows the concept that code should run on any device and under any operating system, thus making itself a portable and platform-independent programming tool.

Both platforms provide **support for common programming languages** such as PHP, Ruby, and Python. Java developers can also use languages such as Java, JavaScript, Clojure, Groovy, and Scala while .NET developers can code in C#, F#, VB.NET, C++, and .NET. The two platforms also support other, less popular languages. However, the J2EE framework sets Java as a default language, while .NET is compatible with other languages such as C#, F#, and VB.net. This is a more flexible approach to development of business apps, although it requires more varied coding skills.

Below factors provide play a major role in a performance of Java:

- 1) Java is an interpreted language and hence the code is not converted to machine language until executed.
- 2) LINQ is not available with JAVA
- 3) Java 7 has the facility of using a switch statement on string variable but not the older frameworks

Below factors provide play a major role in a performance of .NET:

- 1).Net is compiled and run on the operating system where the code is deployed.
- 2) LINQ enables a user to write queries directly instead of using stored procedures
- 3) C# enables you to use a switch statement on a string variable.
- 4)C# supports native generic data support



## Advantages and Disadvantages

Java is a portable language and hence it can be run on any operating system. You can build your application or website easily on any platform. Java usually calls database less often as all its domain objects are already present. Many integrating connectors are available for Java and they are easily available over the internet. Also, code written in Java is backward compatible and can be easily migrated from one operating system to another.

One of the disadvantages of Java is security. Security is mostly compromised in platform agnostic solutions. Also, Scala is incompatible with [Groovy](#) and hence needs better development skills. Also though, multi-platform makes Java special, on the other hand, it tends to make Java slower.

The Microsoft provides a complete unified environment as .Net platform. It provides complete scalability which provides all tools and IDEs over the Microsoft network.

The disadvantage of .Net is that support is available only for Entity framework. It limits the object-relational support. The managed code runs slower than native code in this framework. The developers have to depend on Microsoft for all updates, features and its enhancements.

## Similarities

- 1) Java has a three-tiered architecture which enables developers to build tiered web applications.
- 2) It is one of the best languages for enterprise web development.

- 1) .Net also provides a multi-tier framework which provides facilities where business logic and data are captured.
- 2) It also provides the best platform as a framework and provides automatic garbage collection.

# Www-basic concepts

Stands for "World Wide Web."

It is important to know that this is not a synonym for the Internet. The World Wide Web, or just "the Web," as ordinary people call it, is a subset of the Internet

The Web consists of pages that can be accessed using a Web browser. The Internet is the actual network of networks where all the information resides. Things like Telnet, FTP, Internet gaming, Internet Relay Chat (IRC), and e-mail are all part of the Internet, but are not part of the World Wide Web

The Hyper-Text Transfer Protocol (HTTP) is the method used to transfer Web pages to your computer. With hypertext, a word or phrase can contain a link to another Web site. All Web pages are written in the hyper-text markup language (HTML), which works in conjunction with HTTP.

A broader definition of www provided by World Wide Web Consortium (W3C) is:

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

A broader definition of www provided by World Wide Web Consortium (W3C) is:

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

# Web Server

A Web [server](#) is a program that uses [HTTP](#) (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well.

While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting [web forms](#), including [uploading](#) of files.

Many generic web servers also support [server-side scripting](#) using [Active Server Pages](#) (ASP), [PHP](#) (Hypertext Preprocessor), or other [scripting languages](#). This means that the behaviour of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to generate HTML documents [dynamically](#) ("on-the-fly") as opposed to returning [static documents](#). The former is primarily used for retrieving or modifying information from [databases](#). The latter is typically much faster and more easily [cached](#) but cannot deliver [dynamic content](#).

Web servers can frequently be found [embedded](#) in devices such as [printers](#), [routers](#), [webcams](#) and serving only a [local network](#). The web server may then be used as a part of a system for monitoring or administering the device in question. This usually means that no additional software has to be installed on the client computer since only a web browser is required (which now is included with most [operating systems](#)).

# Client

Client is that which request for resource and data with server.

The client computer usually contains more end-user software than the server computer. A server usually contains more operating system components. Multiple users can log into a server at the same time. A client machine is simple and inexpensive whereas a server machine is more powerful and expensive.

# Example of client a

When a [bank](#) customer accesses [online banking](#) services with a [web browser](#) (the client), the client initiates a request to the bank's web server. The customer's [login](#) credentials may be stored in a [database](#), and the web server accesses the [database server](#) as a client. An [application server](#) interprets the returned data by applying the bank's [business logic](#), and provides the [output](#) to the web server. Finally, the web server returns the result to the client web browser for display.

In each step of this sequence of client–server message exchanges, a computer processes a request and returns data. This is the request-response messaging pattern. When all the requests are met, the sequence is complete and the web browser presents the data to the customer.

# Http protocol

HTTP (Hypertext Transfer Protocol) is perhaps the most popular application protocol used in the Internet (or The WEB).

HTTP is an *asymmetric request-response client-server* protocol as illustrated. An HTTP client sends a request message to an HTTP server. The server, in turn, returns a response message. In other words, HTTP is a *pull protocol*, the client *pulls* information from the server (instead of server *pushes* information down to the client).

HTTP is a stateless protocol. In other words, the current request does not know what has been done in the previous requests.

HTTP permits negotiating of data type and representation, so as to allow systems to be built independently of the data being transferred.

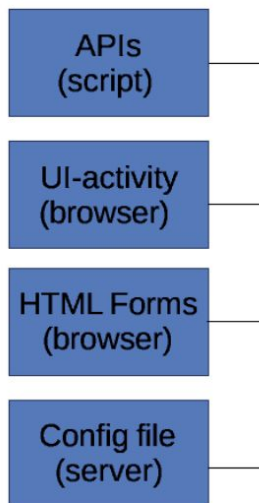
Quoting from the RFC2616: "The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers."



# HTTP Messages

HTTP messages are how data is exchanged between a server and a client. There are two types of messages: *requests* sent by the client to trigger an action on the server, and *responses*, the answer from the server.

Activity initiation



Translation  
into HTTP

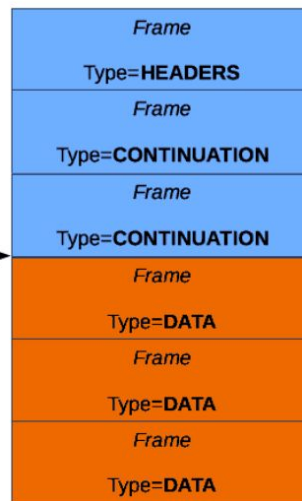
HTTP/1.x message

```
PUT /create_page HTTP/1.1
Host: localhost:8000
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: text/html
Content-Length: 345
```

```
Body line 1
Body line 2
...
```

Binary  
framing

HTTP/2 stream  
(composed of frames)

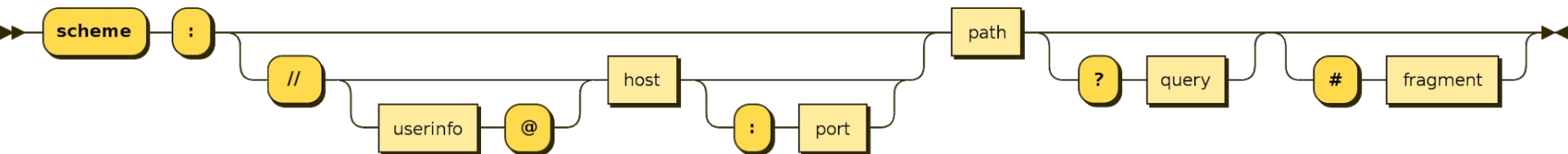


# URL

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.

A URL contains the following information:

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



## Parts of a URL

Using the URL **<https://whatis.techtarget.com/search/query?q=URL>** as an example, components of a URL can include:

- **The protocol or scheme.** Used to access a resource on the internet. Protocols include http, https, ftps, mailto and file. The resource is reached through the domain name system ([DNS](#)) name. In this example, the protocol is https.
- **Host name or domain name.** The unique reference that represents a webpage. For this example, [whatis.techtarget.com](https://whatis.techtarget.com).
- **Port name.** Usually not visible in URLs, but necessary. Always following a colon, port 80 is the default port for web servers, but there are other options. For example, :port80.
- **Path.** A path refers to a file or location on the web server. For this example, [search/query](#).
- **Query.** Found in the URL of [dynamic pages](#). The query consists of a question mark, followed by parameters. For this example, ?.
- **Parameters.** Pieces of information in a query string of a URL. Multiple parameters can be separated by ampersands (&). For this example, q=URL.
- **Fragment.** This is an internal page reference, which refers to a section within the webpage. It appears at the end of a URL and begins with a hashtag (#). Although not in the example above, an example could be #history in the URL <https://en.wikipedia.org/wiki/Internet#History>.