Web Essentials

UNIT-1 | 5 Hrs

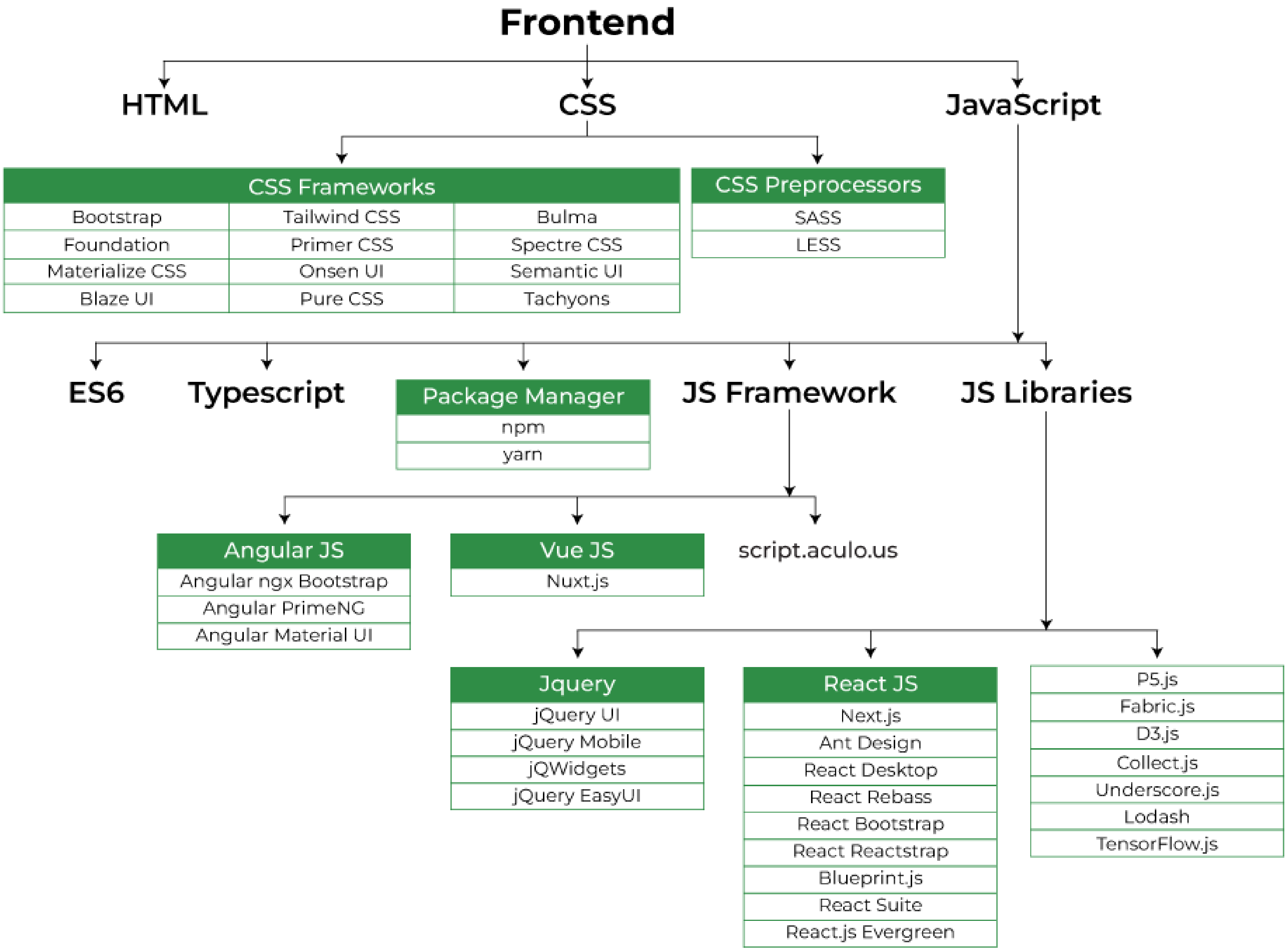
Yuba Raj Devkota | NCCS

Web Technology

Web Technology refers to the various tools and techniques that are utilized in the process of communication between different types of devices over the internet.

- **World Wide Web (WWW):** The World Wide Web is based on several different technologies : Web browsers, Hypertext Markup Language (HTML) and Hypertext Transfer Protocol (HTTP).
- **Web Browser:** The web browser is an application software to explore www (World Wide Web). It provides an interface between the server and the client and requests to the server for web documents and services.
- **Web Server:** Web server is a program which processes the network requests of the users and serves them with files that create web pages. This exchange takes place using Hypertext Transfer Protocol (HTTP).
- **Web Pages:** A webpage is a digital document that is linked to the World Wide Web and viewable by anyone connected to the internet has a web browser.
- **Web Development:** Web development refers to the building, creating, and maintaining of websites.

Classification of Web Development



Frontend

HTML

CSS

JavaScript

CSS Frameworks		
Bootstrap	Tailwind CSS	Bulma
Foundation	Primer CSS	Spectre CSS
Materialize CSS	Onsen UI	Semantic UI
Blaze UI	Pure CSS	Tachyons

CSS Preprocessors
SASS
LESS

ES6

Typescript

Package Manager
npm
yarn

JS Framework

JS Libraries

Angular JS
Angular ngx Bootstrap
Angular PrimeNG
Angular Material UI

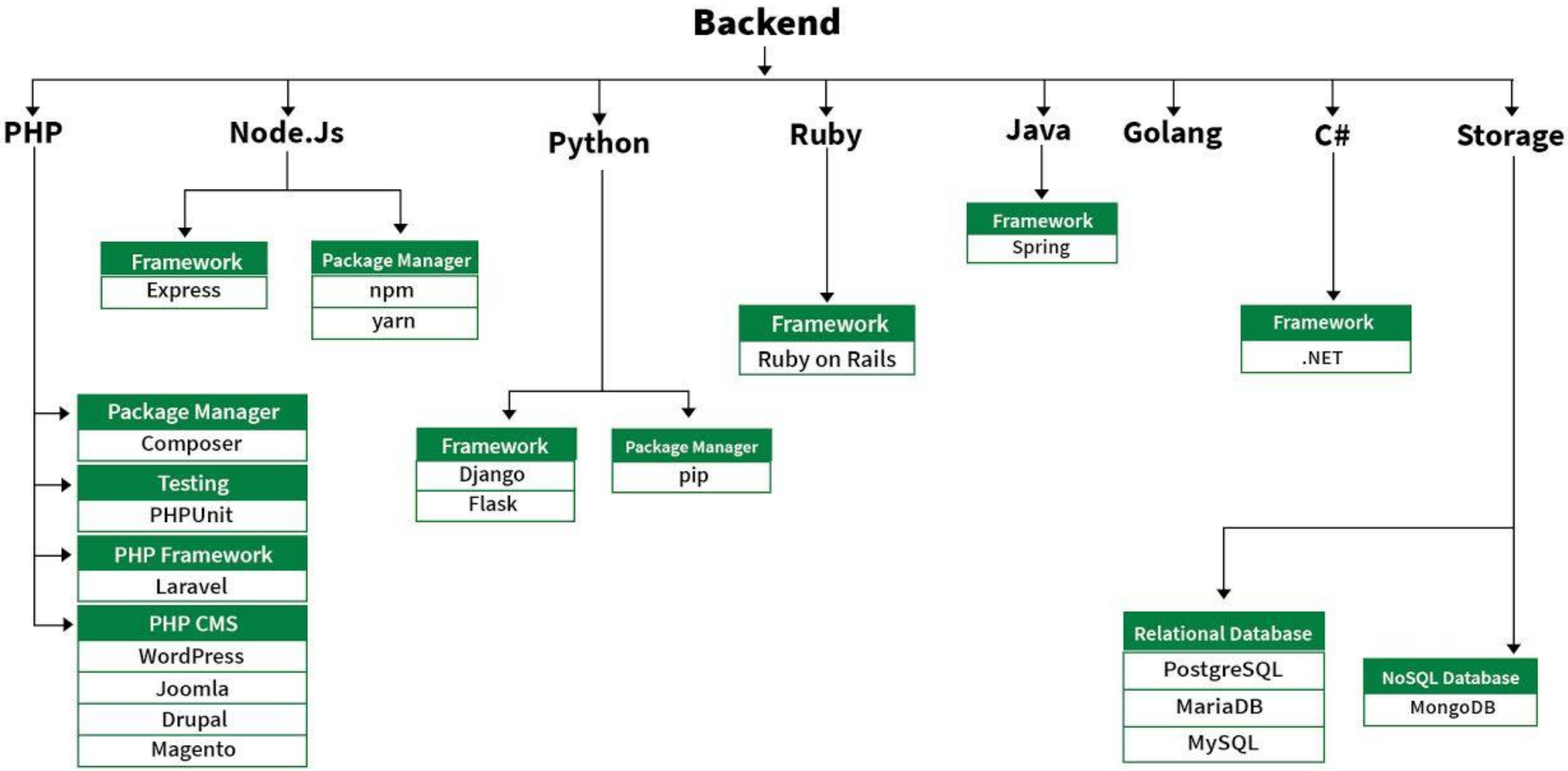
Vue JS
Nuxt.js

script.aculo.us

Jquery
jQuery UI
jQuery Mobile
jQWidgets
jQuery EasyUI

React JS
Next.js
Ant Design
React Desktop
React Rebass
React Bootstrap
React Reactstrap
Blueprint.js
React Suite
React.js Evergreen

P5.js
Fabric.js
D3.js
Collect.js
Underscore.js
Lodash
TensorFlow.js



Backend

PHP

Package Manager
Composer

Testing
PHPUnit

PHP Framework
Laravel

PHP CMS
WordPress
Joomla
Drupal
Magento

Node.Js

Framework
Express

Package Manager
npm
yarn

Python

Framework
Django
Flask

Package Manager
pip

Ruby

Framework
Ruby on Rails

Java

Framework
Spring

Golang

C#

Framework
.NET

Storage

Relational Database
PostgreSQL
MariaDB
MySQL

NoSQL Database
MongoDB

Data Format: Format of data is used by web applications to communicate with each other. It is light weight text based data interchange format which means, it is simpler to read and write.

Below are two common data formats used in web development.

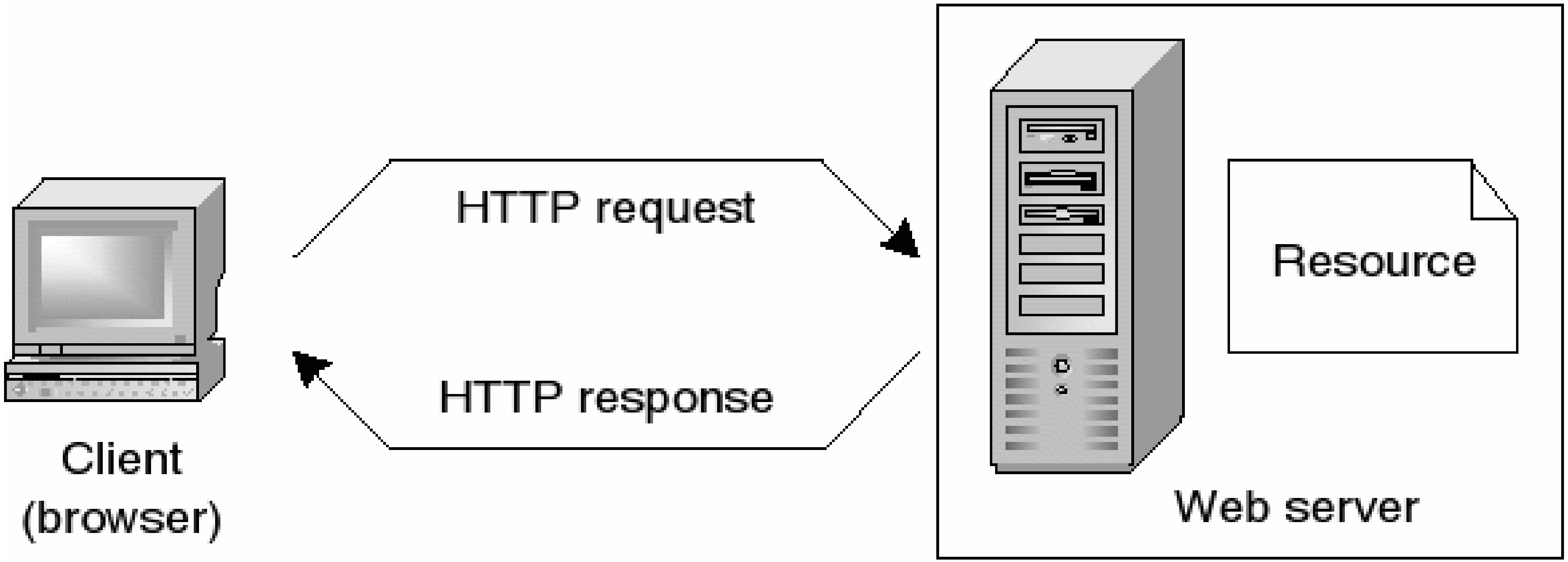
- **XML:** Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
- **JSON:** JSON or JavaScript Object Notation is a format for structuring data.

API: API is an abbreviation for Application Programming Interface which is a collection of communication protocols and subroutines used by various programs to communicate between them.

Web Protocols: Web protocols are set of rules followed by everyone communicating over the web.

- **HTTP:** The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.
  HTTP works as a request-response protocol between a client and server.
  A web browser may be the client, and an application on a computer that hosts a web site may be the server.

How Web Works



Static vs dynamic websites

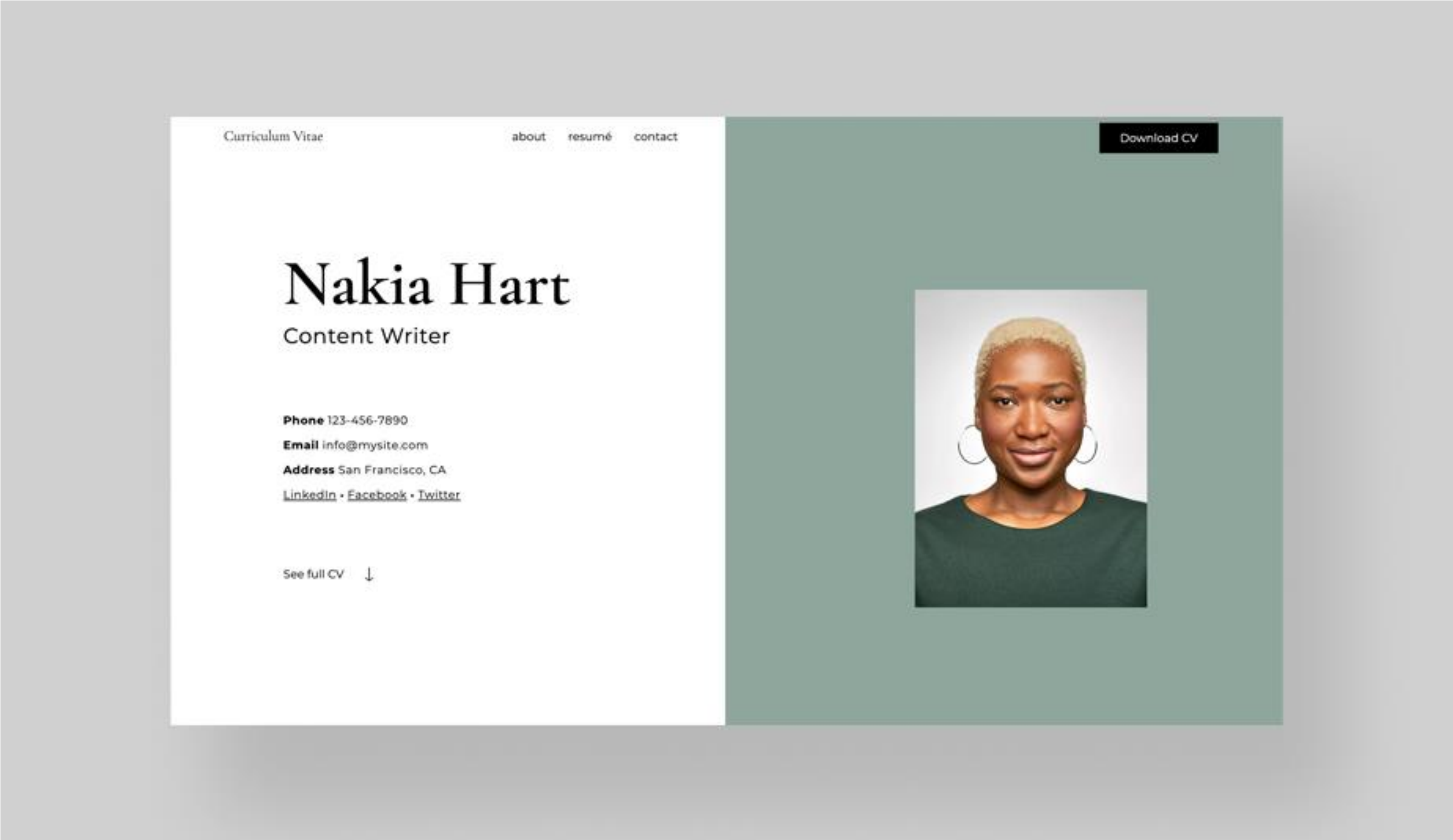
- What is a static website?
  - A static website is made up of web pages created using HTML, CSS and JavaScript (all examples of web development languages). Each page on a static website is stored as a single HTML file, which is delivered directly from the server to the web page exactly as is. This content essentially becomes a part of the design on your page, and won't change unless the original HTML file is edited at a code level.

Advantages:

- Faster page loading speed
- Quick creation
- Less risky

Disadvantages:

- Slow, if more pages
- Less efficient management



Nakia Hart

Content Writer

Phone 123-456-7890

Email info@mysite.com

Address San Francisco, CA

LinkedIn · Facebook · Twitter

What is a dynamic website?

- Built using server side language and technology, dynamic websites allow for the content of each page to be delivered and displayed dynamically, or on-the-fly, according to user behavior or from user-generated content.
- With a dynamic website all of your data and content are organized in a database or backend content management system (CMS), which connects to your website pages. The way this information is arranged and connected to your site's design controls how and when its content is revealed on a page.

Advantages:

- Easily Updated
- A better user experience
- Greater Functionality

Disadvantages:

- It requires more resource to create
- Performance Issue

Inspired by sand, wind and ocean - Earth Skincare offers an exquisite line of products. Earth products are being sold online only, so we needed to create a unique ordering and packaging experience to make customers feel like we're hosting them in our own store.

What Is Web 1.0?

People use the term “Web 1.0” to describe the earliest form of the Internet. Users saw the first example of a worldwide network that hinted at future digital communication and information-sharing potential. Here are a few characteristics found in Web 1.0:

- It’s made up of static pages connected to a system via hyperlinks
- It has HTML 3.2 elements like frames and tables
- HTML forms get sent through e-mail
- The content comes from the server's filesystem, not a relational database management system
- It features GIF buttons and graphics

Take a real-world dictionary, digitize everything in it, and make it accessible to people online to look at (but not be able to react to it). Boom. That’s Web 1.0.

What Is Web 2.0?

- If Web 1.0 was made up of a small number of people generating content for a larger audience, then Web 2.0 is many people creating even more content for a growing audience.
- Web 1.0 focused on reading; Web 2.0 focused on participating and contributing.
- It offers free information sorting, allowing users to retrieve and classify data collectively
- It contains dynamic content that responds to the user's input
- It employs Developed Application Programming Interfaces (API)
- It encourages self-usage and allows forms of interaction like:
  - Podcasting, Social media, Tagging, Blogging, Commenting
  - Curating with RSS, Social networking, Web content voting
- It's used by society at large and not limited to specific communities.

What Is Web 3.0?

- Web 1.0 is the "read-only Web," Web 2.0 is the "participative social Web," and Web 3.0 is the "read, write, execute Web.“
- It incorporates Artificial Intelligence and Machine Learning.
- It presents the connectivity of multiple devices and applications through the Internet of Things (IoT).
- It uses 3-D graphics. In fact, we already see this in computer games, virtual tours, and e-commerce.
- It can be used for Blockchain Games, Cryptocurrency etc.

Uses of Web 1.0, Web 2.0, Web 3.0

- Uses of Web 1.0: Web 1.0 functions as a CDN (content delivery network), allowing a chunk of the website to be displayed on the website. As a result, it can be used as a personal website. The users would be charged in terms of each page view. It is made up of directories that allow its users to get a certain collection of information.
- Uses of Web 2.0: The social web comprises numerous platforms and tools. People contribute their opinions, insights, experiences, and thoughts on these sites. Thus, Web 2.0 tends to interact substantially more with its end users. These end users are not only the users of the programs, but also the participants/viewers generated by podcasts, tagging, blogging, RSS curation, Web content voting, Social media, Social networking, Social bookmarking, and many more.
- Uses of Web 3.0: Web 3.0 are enhanced variations of the original Web 1.0 from the 1990s and early 2000s. Web 3.0 is the next generation of the current web that we are familiar with.

Architectural issues of web layer

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

- **HTML-The Content Layer:** The content layer is where you store all the content that your customers want to read or look at. This 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. That way, your customers who have JavaScript turned off or can't view CSS will still have access to the entire site, if not all the functionality.
- **CSS - the Styles Layer:** Store all your styles for your Web site in an external style sheet. This 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.
- **JavaScript - the Behavior Layer:** JavaScript is the most commonly used language for writing the behavior layer; ASP, CGI and PHP can also generate Web page behaviors. However, when most developers refer to the behavior layer, they mean that layer that is activated directly in the Web browser - so JavaScript is nearly always the language of choice. You use this layer to interact directly with the DOM or Document Object Model.

Benefits of separating the layers are:

- 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.
- 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.
- Backwards 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.

Software Architecture: One-Tier, Two-Tier, Three Tier, N Tier

Software Architecture: Software Architecture consists of One Tier, Two Tier, Three Tier and N-Tier architectures.

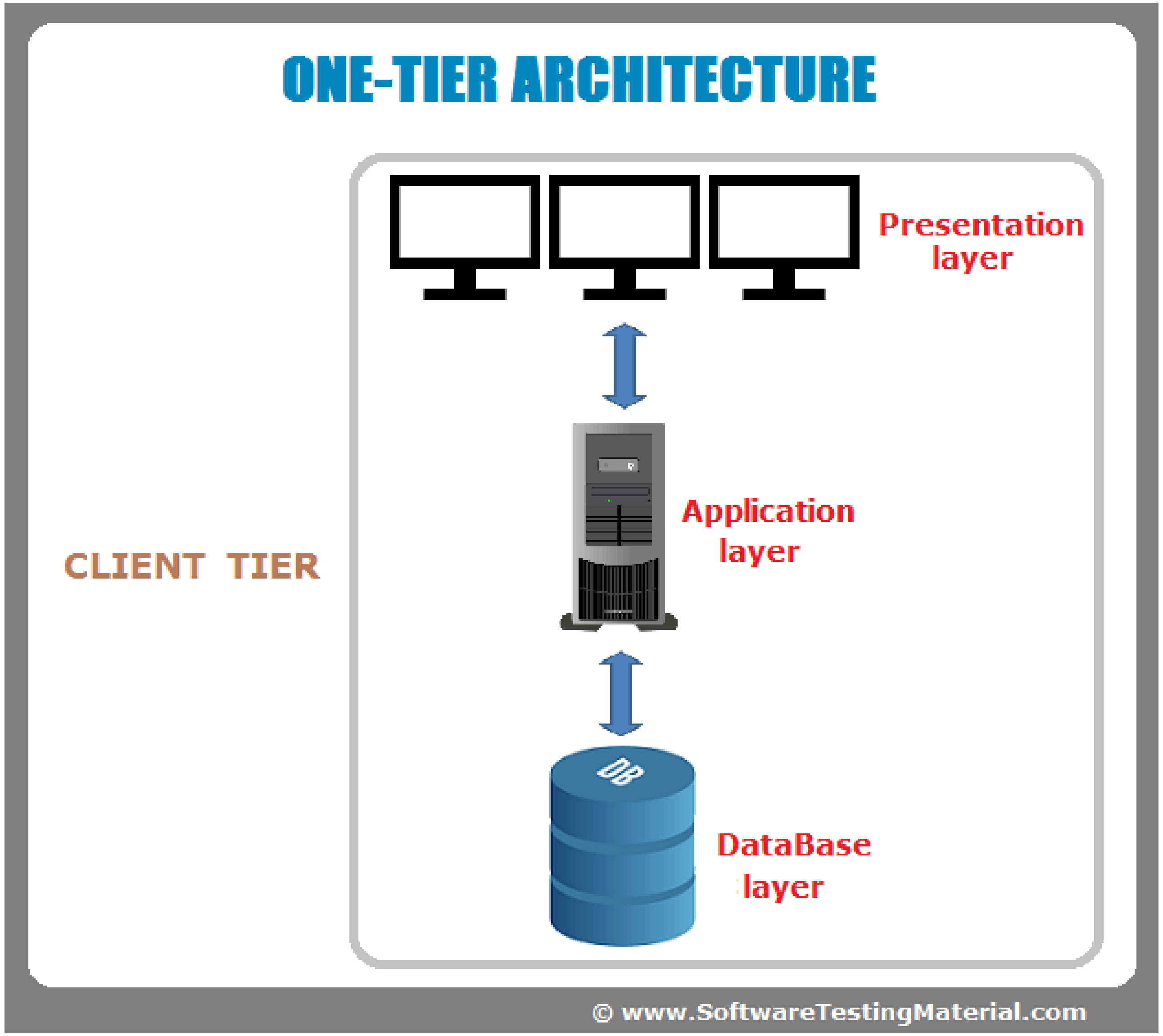
One Tier Architecture:

One Tier application AKA Standalone application

One tier architecture has all the layers such as Presentation, Business, Data Access layers in a single software package. Applications which handles all the three tiers such as MP3 player, MS Office are come under one tier application. The data is stored in the local system or a shared drive.

Advantage— Fast for a single user because communication with another system is not necessary.

Disadvantage— Completely unscalable. Only one user can access the system at a given time via the local client.



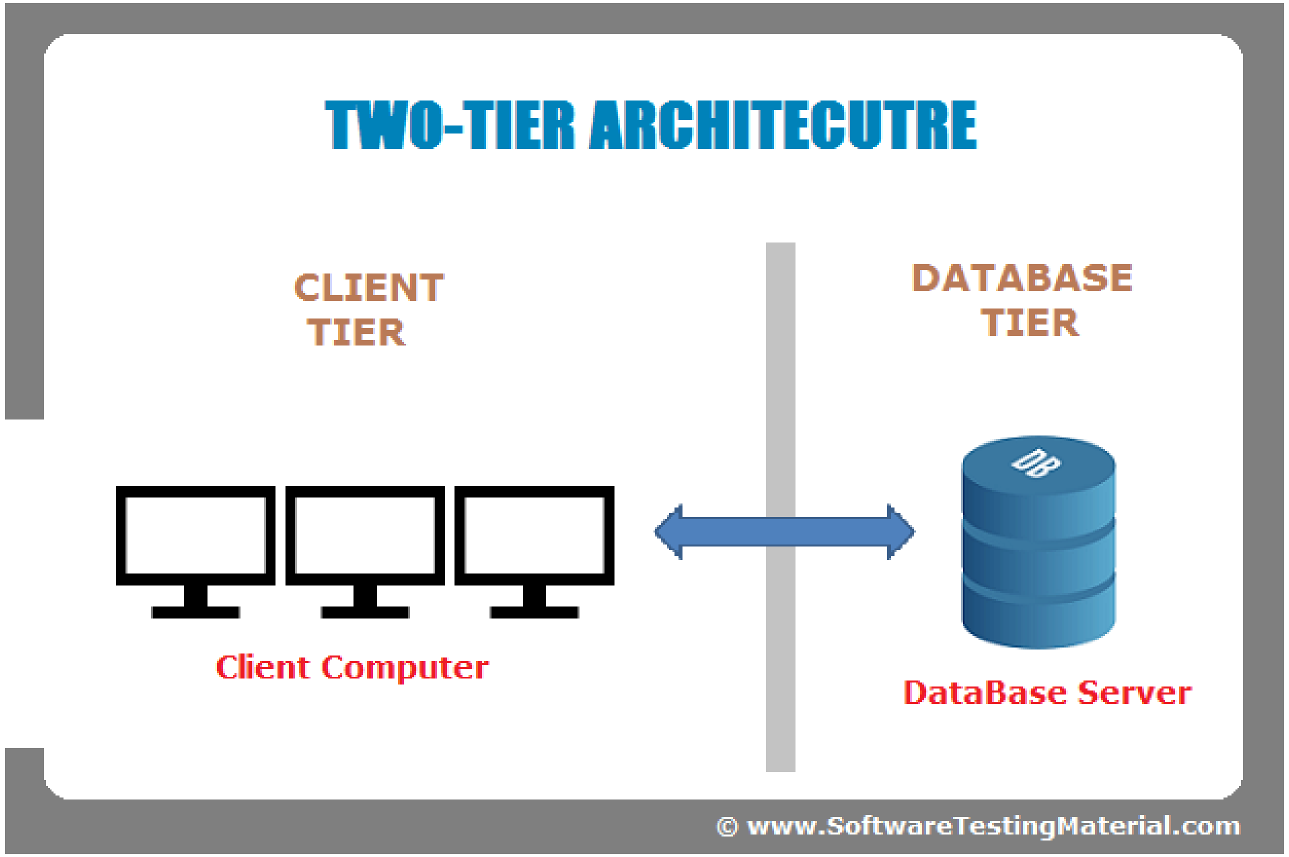
© www.SoftwareTestingMaterial.com

Two-Tier Architecture:

Two Tier application AKA Client-Server application

The Two-tier architecture is divided into two parts:

1. Client Application (Client Tier)
2. Database (Data Tier)



Client system handles both Presentation and Application layers and 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

Advantages

1. Applications can be easily developed due to simplicity.
2. Maximum user satisfaction is gained with accurate and fast prototyping of applications through robust tools.
3. Since this contains static business rules it’s more applicable for homogenous environments.
4. Database server and business logic is physically close, which offers higher performance.

Disadvantages

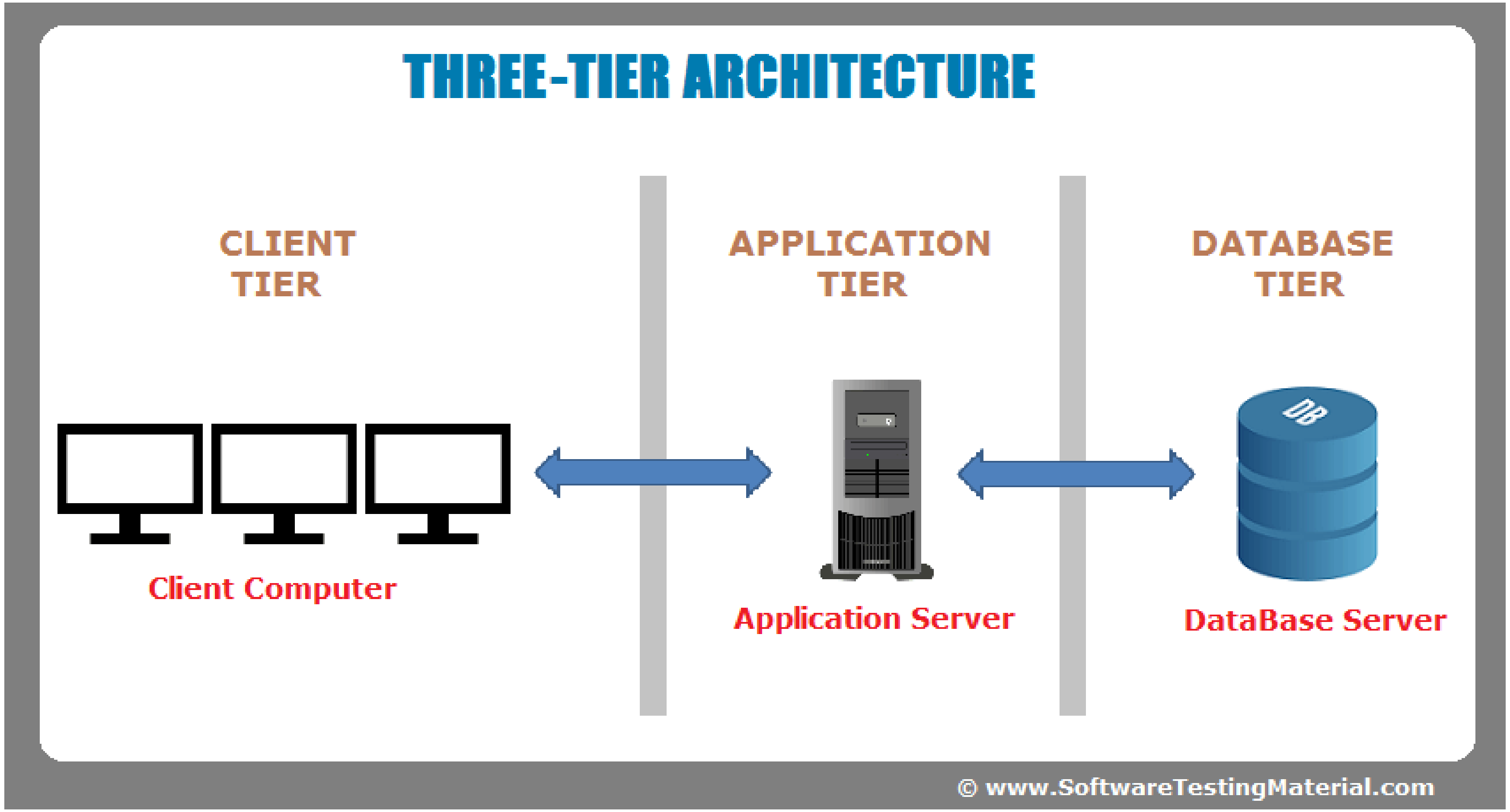
1. Heterogeneous environments/Business environments with rapidly changing rules and regulations are not suitable since the database server has to handle the business logic which slows down database performance.
2. Since client beholds most of the application logic, problems arise in controlling the software version and re-distributing of new versions.
3. Security wise this is complicated as users need to have separate login information for every SQL server.
4. Client tools and SQL middleware implemented in 2-tier environment is proprietary which remains cautious on long term feasibility.

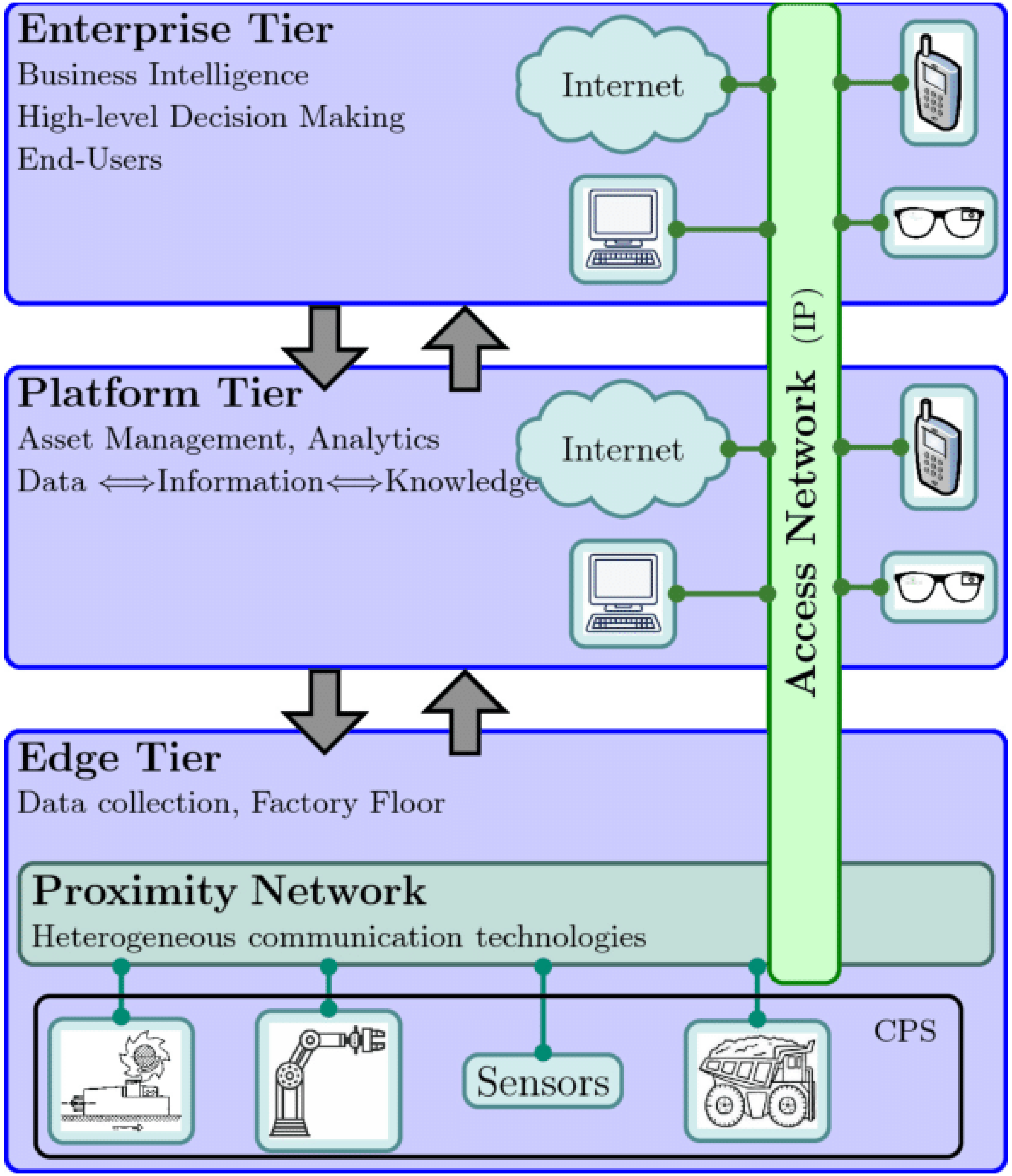
Three-Tier Architecture:

The Three-tier architecture is divided into three parts:

1. Presentation layer (Client Tier)
2. Application layer (Business Tier)
3. Database layer (Data Tier)

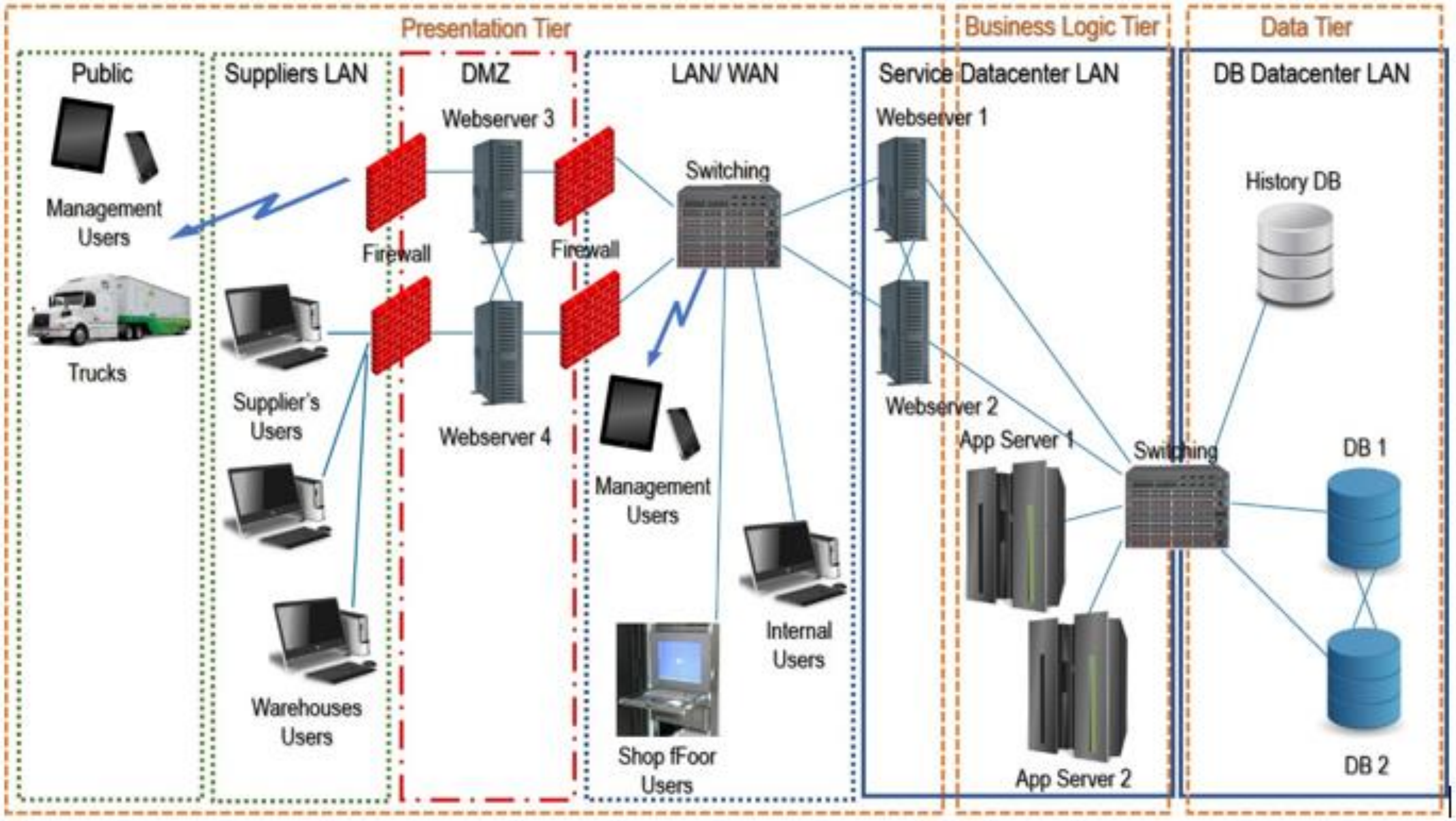
Three Tier application AKA Web Based application




Enterprise Tier
Business Intelligence
High-level Decision Making
End-Users
Internet
Platform Tier
Asset Management, Analytics
Data ⟺Information⟺Knowledge
Internet
Access Network (IP)
Edge Tier
Data collection, Factory Floor
Proximity Network
Heterogeneous communication technologies
CPS
Sensors

What is N-Tier Architecture?

N-tier (or multi-tier) architecture refers to software that has its several layers rendered by distinct IT environments (tiers) under a client-server logic. The user interface (Presentation Tier) runs in a separate environment than the "computation" (Business Logic Tier) which in turn also runs in a distinct environment from the database engine and instances (Data Tier).



What are the Benefits of N-Tier Architecture?

There are several benefits to using n-tier architecture for your software. These are scalability, ease of management, flexibility, and security.

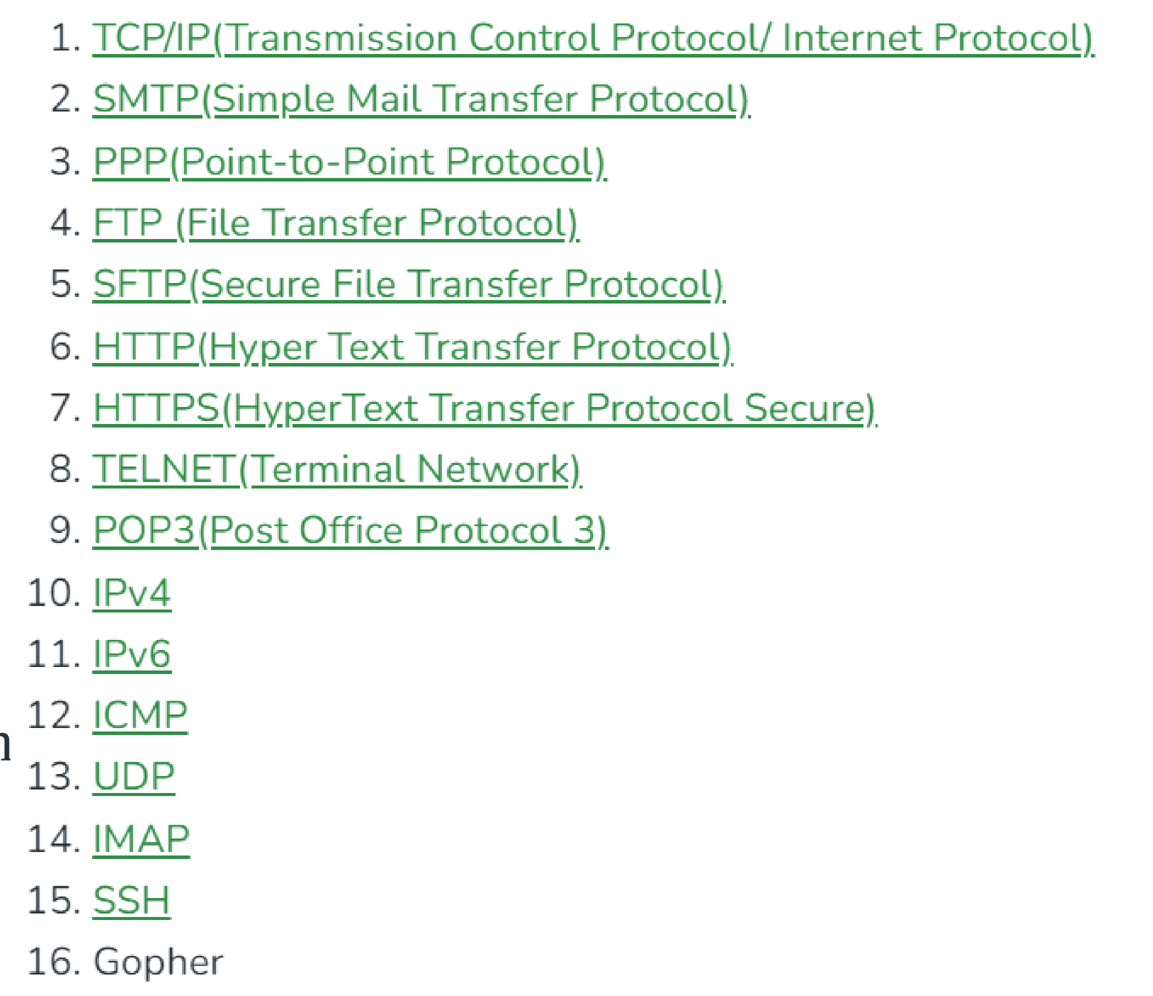
- **Secure:** You can secure each of the three tiers separately using different methods.
- **Easy to manage:** You can manage each tier separately, adding or modifying each tier without affecting the other tiers.
- **Scalable:** If you need to add more resources, you can do it per tier, without affecting the other tiers.
- **Flexible:** Apart from isolated scalability, you can also expand each tier in any manner that your requirements dictate.

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.

Types of Internet Protocols

- Internet Protocols are a set of rules that governs the communication and exchange of data over the internet. Both the sender and receiver should follow the same protocols in order to communicate the data. In order to understand it better, let’s take an example of a language. Any language has its own set of vocabulary and grammar which we need to know if we want to communicate in that language. Similarly, over the internet whenever we access a website or exchange some data with another device then these processes are governed by a set of rules called the internet protocols.
- The internet and many other data networks work by organizing data into small pieces called packets. Each large data sent between two network devices is divided into smaller packets by the underlying hardware and software. Each network protocol defines the rules for how its data packets must be organized in specific ways according to the protocols the network supports.

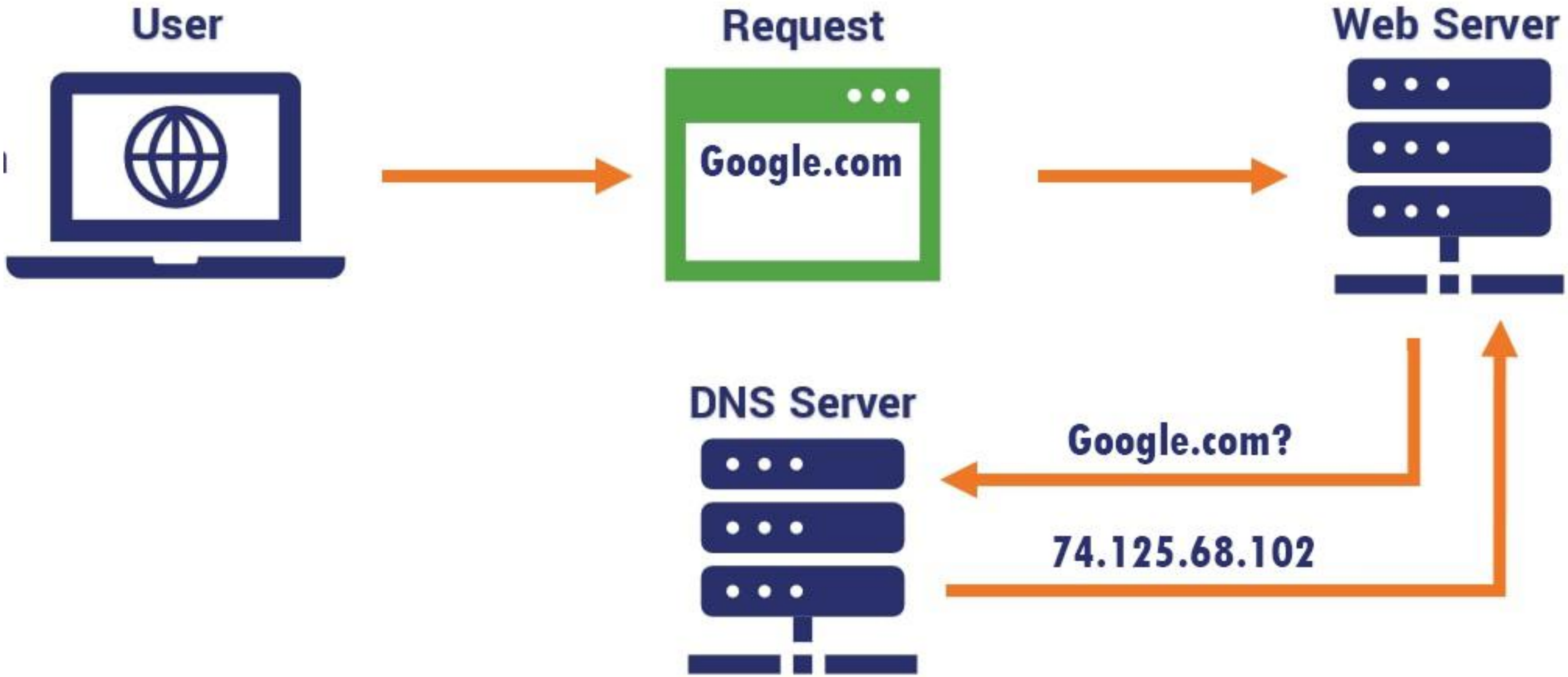


1. TCP/IP(Transmission Control Protocol/ Internet Protocol)
2. SMTP(Simple Mail Transfer Protocol)
3. PPP(Point-to-Point Protocol)
4. FTP (File Transfer Protocol)
5. SFTP(Secure File Transfer Protocol)
6. HTTP(Hyper Text Transfer Protocol)
7. HTTPS(HyperText Transfer Protocol Secure)
8. TELNET(Terminal Network)
9. POP3(Post Office Protocol 3)
10. IPv4
11. IPv6
12. ICMP
13. UDP
14. IMAP
15. SSH
16. Gopher

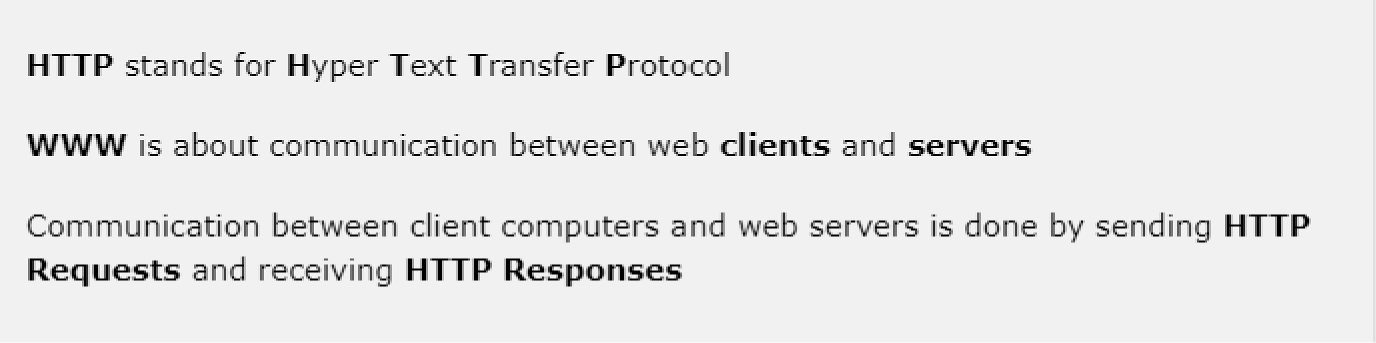
- **TCP/IP(Transmission Control Protocol/ Internet Protocol)**
  - These are a set of standard rules that allows different types of computers to communicate with each other. The IP protocol ensures that each computer that is connected to the Internet is having a specific serial number called the IP address.
  - TCP specifies how data is exchanged over the internet and how it should be broken into IP packets. It also makes sure that the packets have information about the source of the message data, the destination of the message data, the sequence in which the message data should be re-assembled, and checks if the message has been sent correctly to the specific destination.
  - The TCP is also known as a connection-oriented protocol.
- **UDP**
  - UDP (User Datagram Protocol) is a connectionless, unreliable transport layer protocol. Unlike TCP, it does not establish a reliable connection between devices before transmitting data, and it does not guarantee that data packets will be received in the order they were sent or that they will be received at all. Instead, UDP simply sends packets of data to a destination without any error checking or flow control.
  - UDP is typically used for real-time applications such as streaming video and audio, online gaming, and VoIP (Voice over Internet Protocol) where a small amount of lost data is acceptable and low latency is important. UDP is faster than TCP because it has less overhead. It doesn’t need to establish a connection, so it can send data packets immediately.
  - It also doesn’t need to wait for confirmation that the data was received before sending more, so it can transmit data at a higher rate.

Domain Name Server (DNS)

- DNS stands for domain name server or domain name system. It is a type of system in which a domain name is converted to an IP address. The example of a domain name is www.google.com and the example of IP address is 192.168.1.1.
- DNS is a type of phonebook in which all the IP addresses are stored with associated domain names.
- When a computer or mobile user types a domain name in the address bar of the browser then the browser sends a request to the DNS server. The DNS server finds the IP address and then connect it with the hosting and then the web page is displayed on the browser.
- There are some rules to define a domain name. For example, it is not allowed to use a hyphen (-) at the start and end of the domain name. Also, the domain name cannot be more than 253 characters.
- DNS acts as a communicator between humans and computers. The computer understands IP addresses while humans understand English words. So DNS convert English words (e.g. www.google.com) to IP address (e.g. 74.125.68.102)
- The IP address is understandable by the browser. The browser sends a request to the Google server and Google servers send the web page to the browser.


User
Request
Web Server
Google.com
DNS Server
Google.com?
74.125.68.102

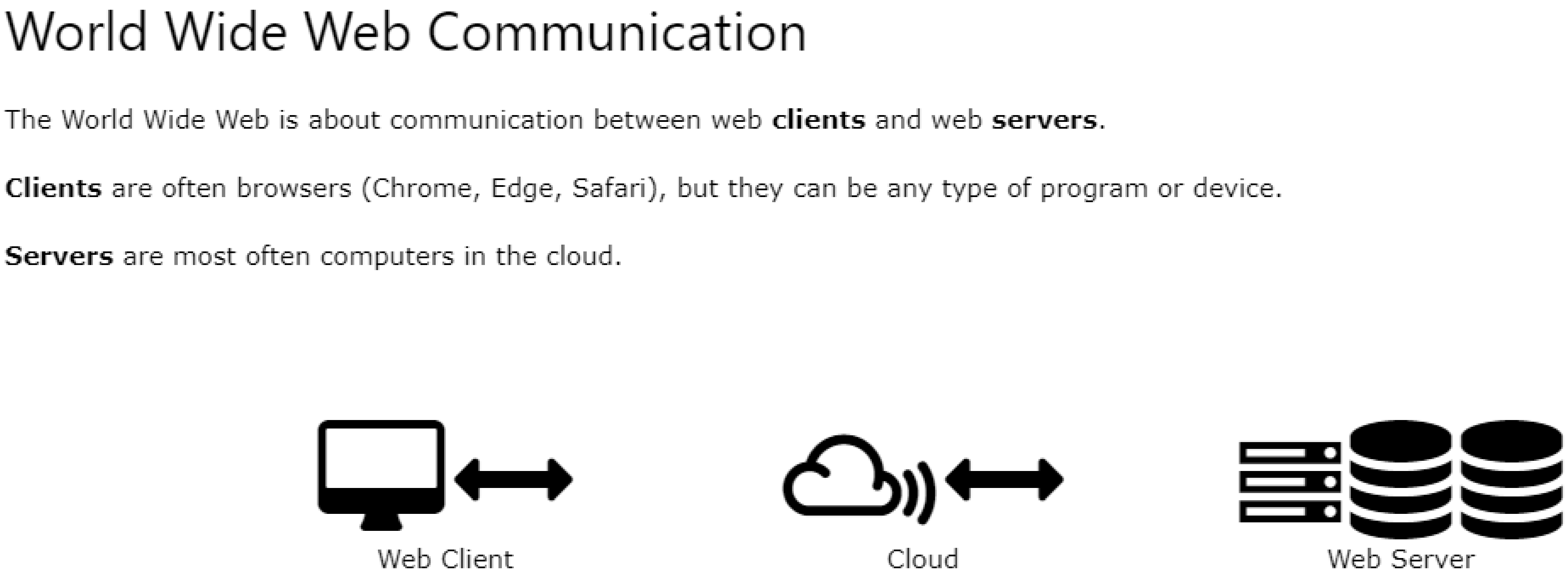
What is HTTP?



HTTP stands for **H**yper **T**ext **T**ransfer **P**rotocol

WWW is about communication between web **clients** and **servers**

Communication between client computers and web servers is done by sending **HTTP Requests** and receiving **HTTP Responses**



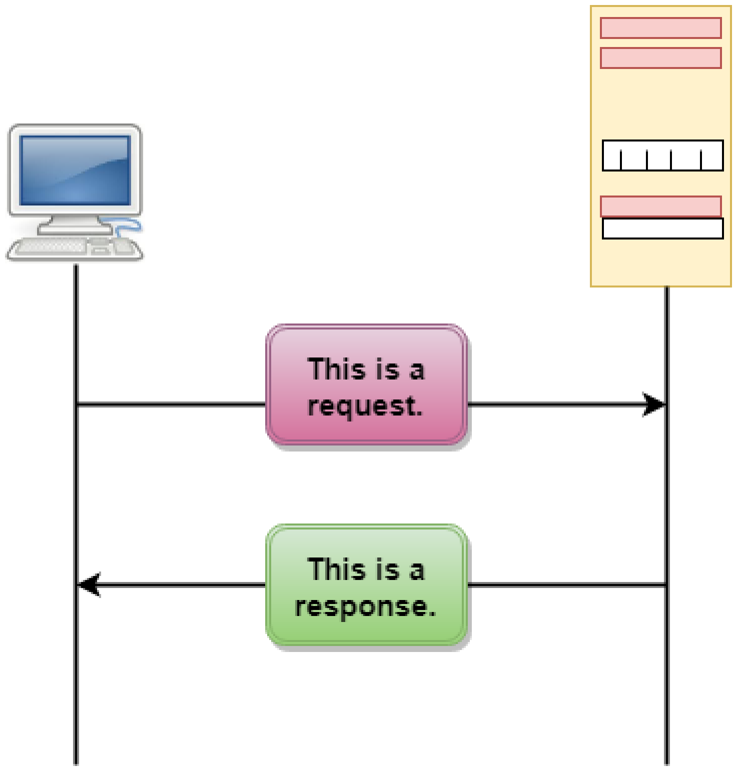
World Wide Web Communication

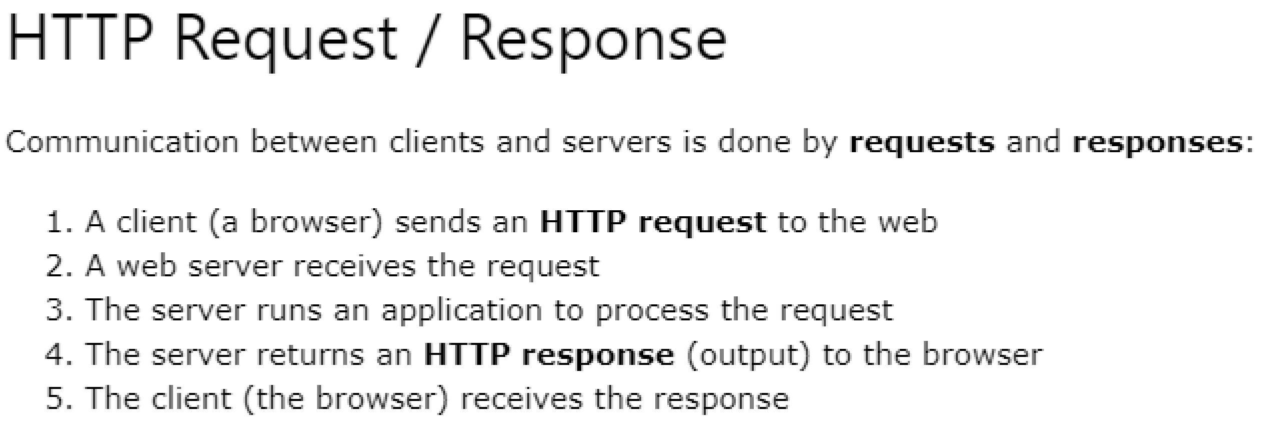
The World Wide Web is about communication between web **clients** and web **servers**.

Clients are often browsers (Chrome, Edge, Safari), but they can be any type of program or device.

Servers are most often computers in the cloud.

HTTP Transactions

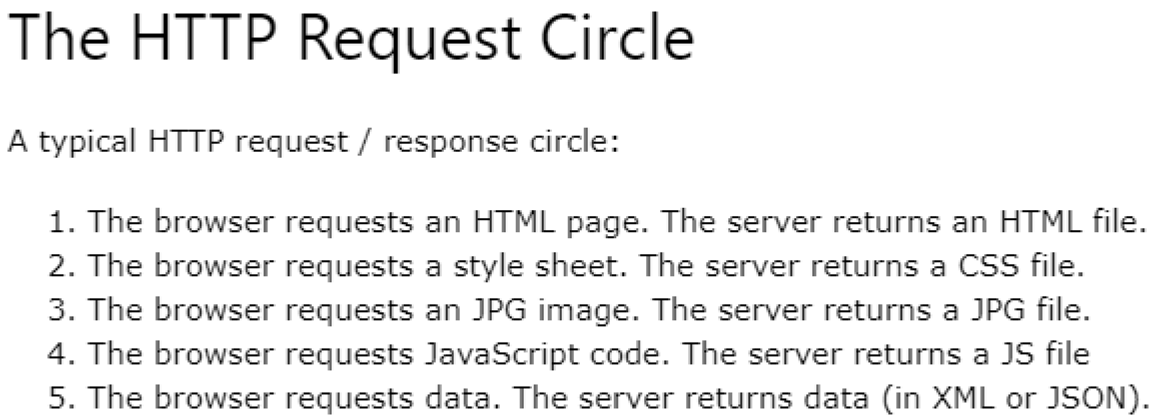




HTTP Request / Response

Communication between clients and servers is done by **requests** and **responses**:

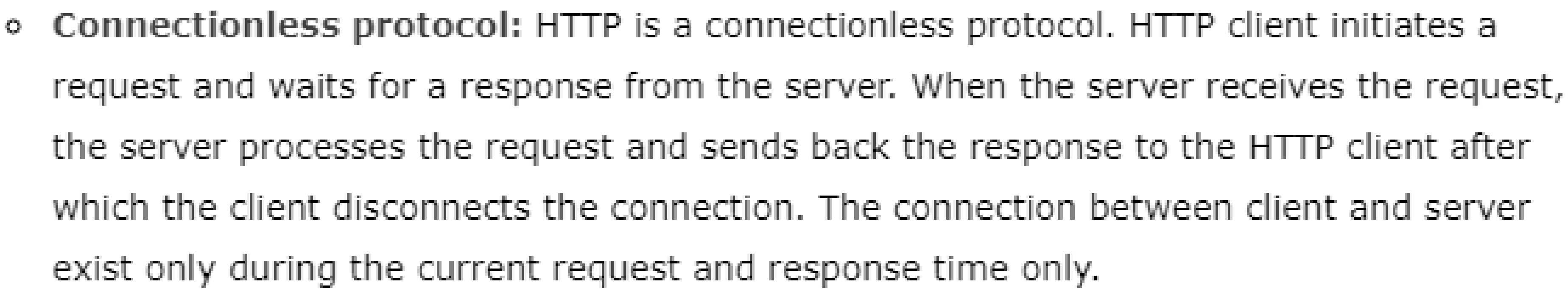
1. A client (a browser) sends an **HTTP request** to the web
2. A web server receives the request
3. The server runs an application to process the request
4. The server returns an **HTTP response** (output) to the browser
5. The client (the browser) receives the response



The HTTP Request Circle

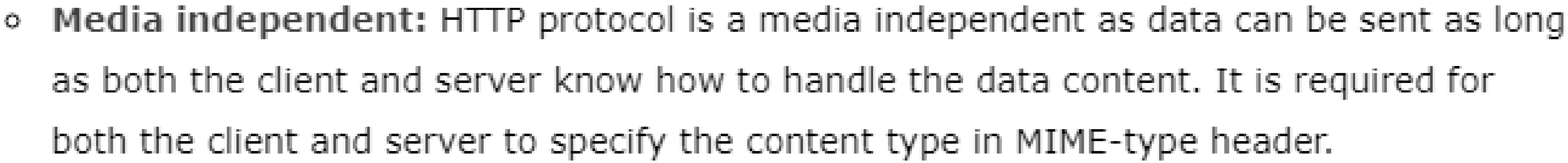
A typical HTTP request / response circle:

1. The browser requests an HTML page. The server returns an HTML file.
2. The browser requests a style sheet. The server returns a CSS file.
3. The browser requests an JPG image. The server returns a JPG file.
4. The browser requests JavaScript code. The server returns a JS file
5. The browser requests data. The server returns data (in XML or JSON).

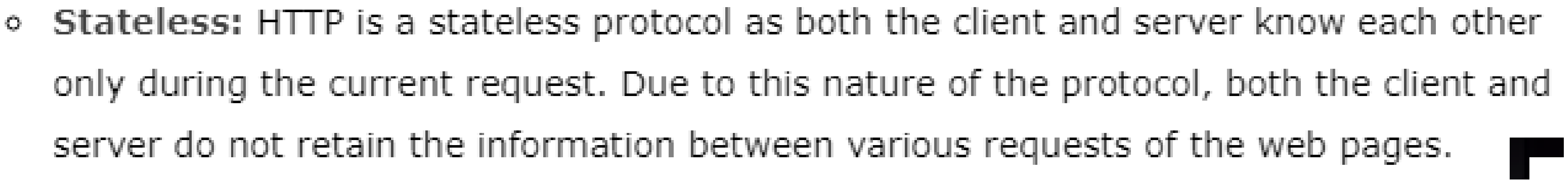


- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.

Features of HTTP:

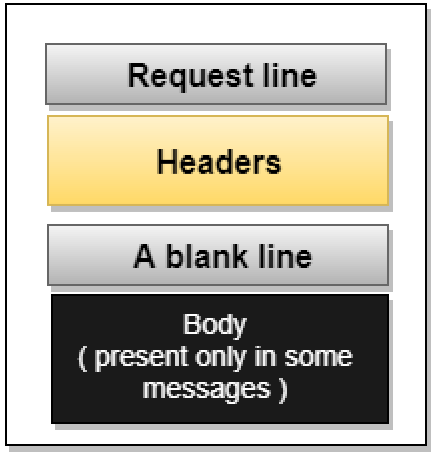


- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.

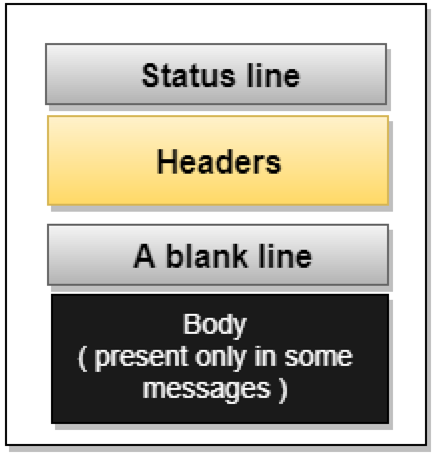


- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

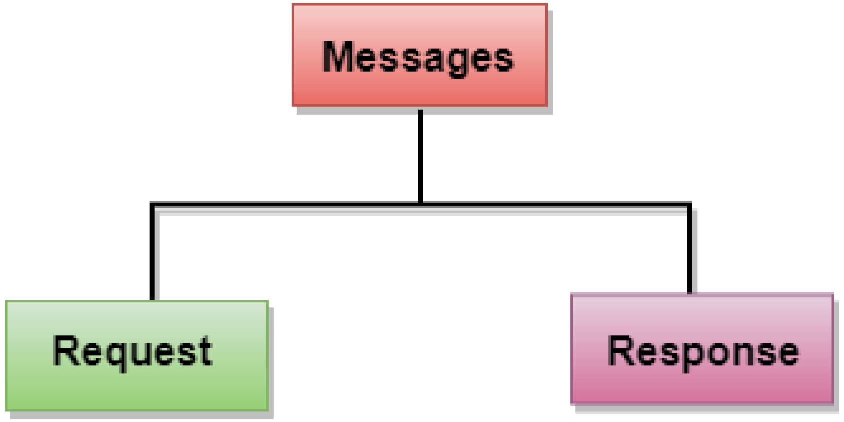
Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.



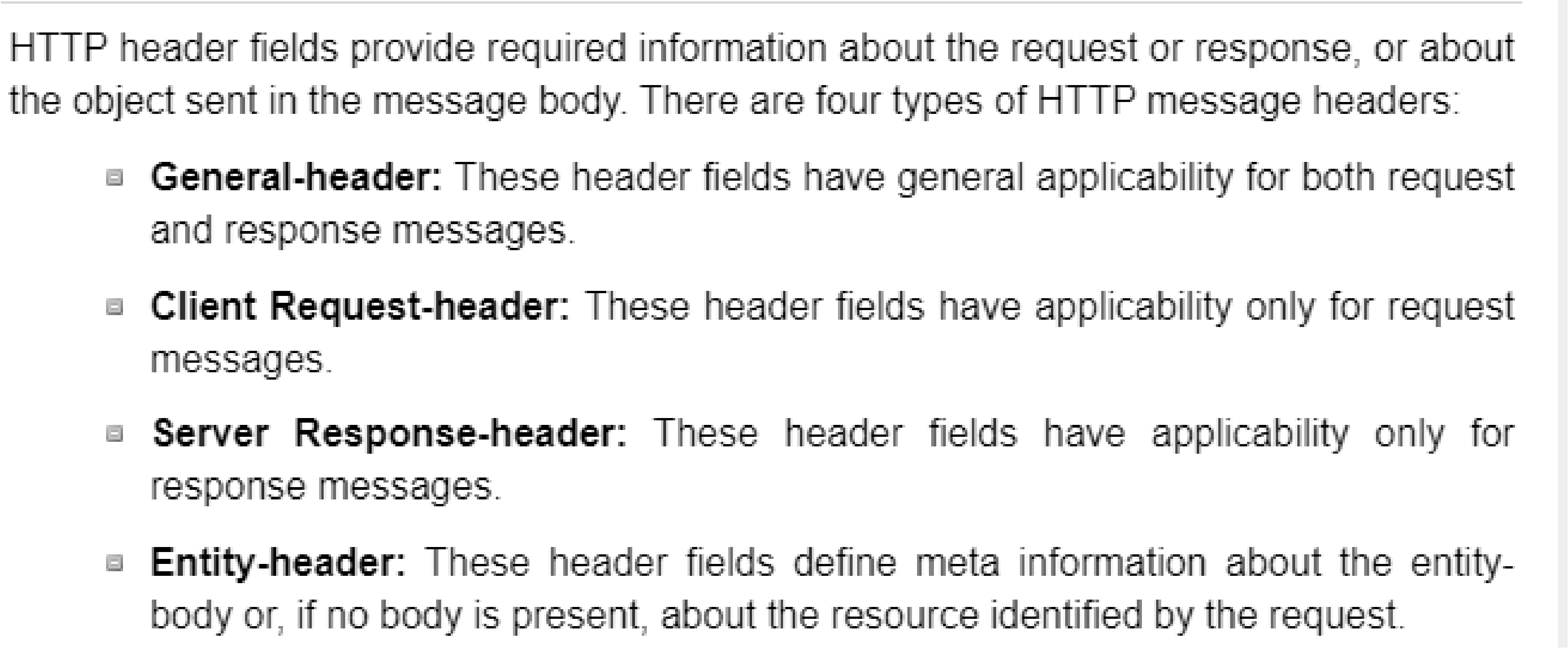
Response Message: The response message is sent by the server to the client that consi
a status line, headers, and sometimes a body.



Messages



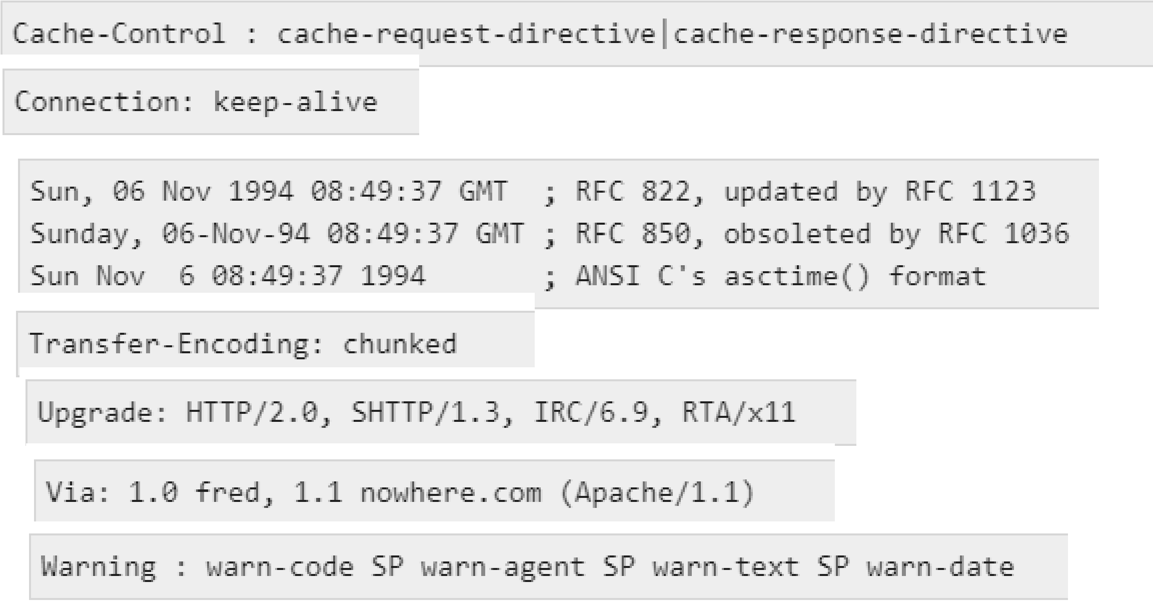
HTTP - Header Fields



HTTP header fields provide required information about the request or response, or about the object sent in the message body. There are four types of HTTP message headers:

- **General-header:** These header fields have general applicability for both request and response messages.
- **Client Request-header:** These header fields have applicability only for request messages.
- **Server Response-header:** These header fields have applicability only for response messages.
- **Entity-header:** These header fields define meta information about the entity-body or, if no body is present, about the resource identified by the request.

General Headers



```
Cache-Control : cache-request-directive|cache-response-directive
```

```
Connection: keep-alive
```

```
Sun, 06 Nov 1994 08:49:37 GMT  ; RFC 822, updated by RFC 1123
Sunday, 06-Nov-94 08:49:37 GMT ; RFC 850, obsoleted by RFC 1036
Sun Nov  6 08:49:37 1994       ; ANSI C's asctime() format
```

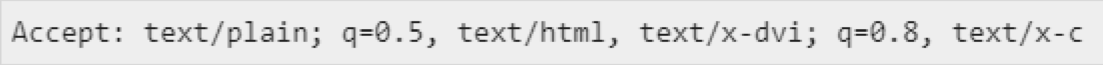
```
Transfer-Encoding: chunked
```

```
Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9, RTA/x11
```

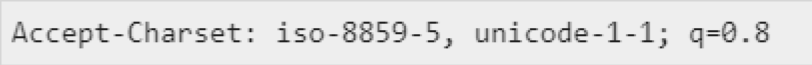
```
Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)
```

```
Warning : warn-code SP warn-agent SP warn-text SP warn-date
```

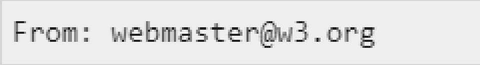
Client Request Headers



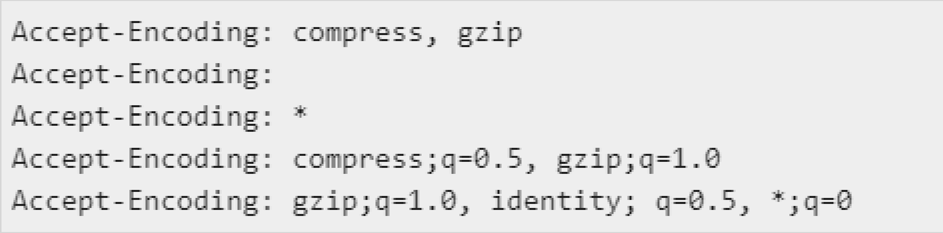
```
Accept: text/plain; q=0.5, text/html, text/x-dvi; q=0.8, text/x-c
```



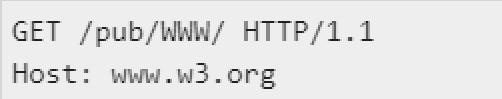
```
Accept-Charset: iso-8859-5, unicode-1-1; q=0.8
```



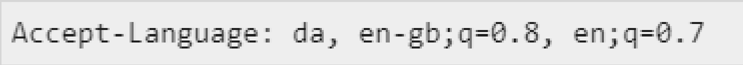
```
From: webmaster@w3.org
```



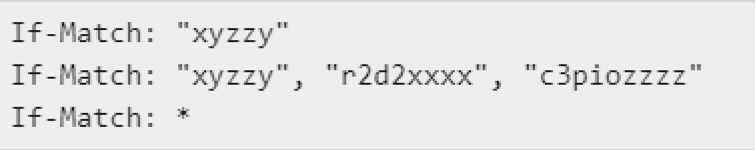
```
Accept-Encoding: compress, gzip
Accept-Encoding:
Accept-Encoding: *
Accept-Encoding: compress;q=0.5, gzip;q=1.0
Accept-Encoding: gzip;q=1.0, identity; q=0.5, *;q=0
```



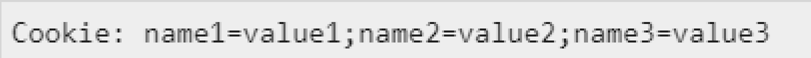
```
GET /pub/WWW/ HTTP/1.1
Host: www.w3.org
```



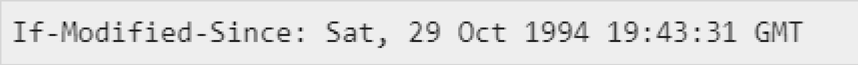
```
Accept-Language: da, en-gb;q=0.8, en;q=0.7
```



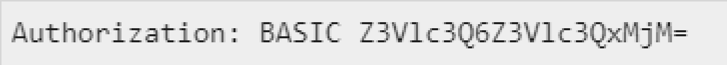
```
If-Match: "xyzzy"
If-Match: "xyzzy", "r2d2xxxx", "c3piozzzz"
If-Match: *
```



```
Cookie: name1=value1;name2=value2;name3=value3
```

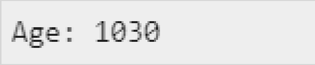


```
If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT
```

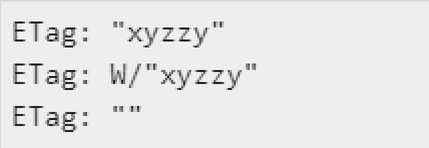


```
Authorization: BASIC Z3Vlc3Q6Z3Vlc3QxMjM=
```

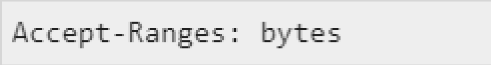
Server Response Headers



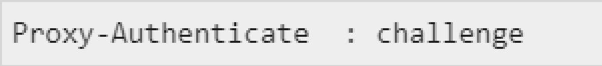
```
Age: 1030
```



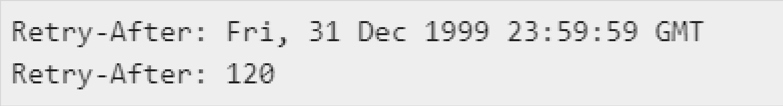
```
ETag: "xyzzy"
ETag: W/"xyzzy"
ETag: ""
```



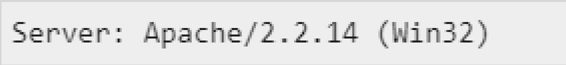
```
Accept-Ranges: bytes
```



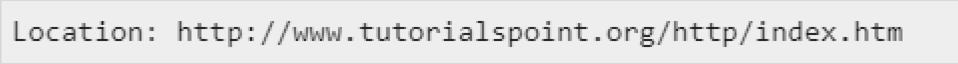
```
Proxy-Authenticate  : challenge
```



```
Retry-After: Fri, 31 Dec 1999 23:59:59 GMT
Retry-After: 120
```



```
Server: Apache/2.2.14 (Win32)
```



```
Location: http://www.tutorialspoint.org/http/index.htm
```

Examples of Request Message

Now let's put it all together to form an HTTP request to fetch **hello.htm** page from the web server running on tutorialspoint.com

```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Here we are not sending any request data to the server because we are fetching a plain HTML page from the server. Connection is a general-header, and the rest of the headers are request headers. The following example shows how to send form data to the server using request message body:

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

licenseID=string&content=string&/paramsXML=string
```

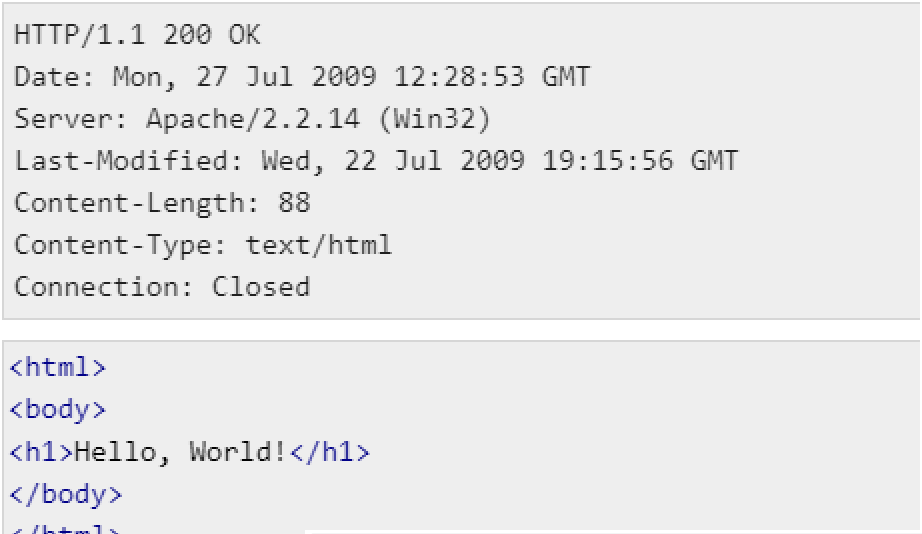
Here the given URL */cgi-bin/process.cgi* will be used to process the passed data and accordingly, a response will be returned. Here **content-type** tells the server that the passed data is a simple web form data and **length** will be the actual length of the data put in the message body. The following example shows how you can pass plain XML to your web server:

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://clearforest.com/">string</string>
```

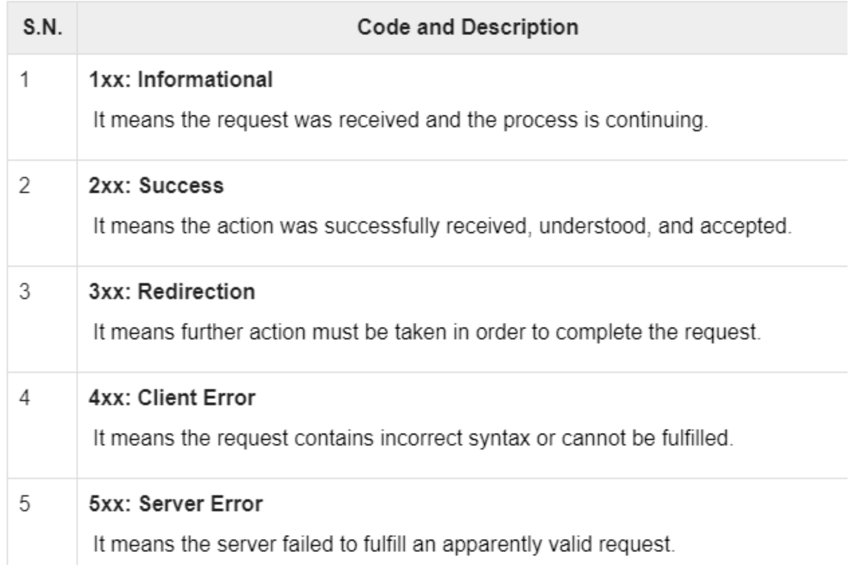
Examples of Response Message

Now let's put it all together to form an HTTP response for a request to fetch the **hello.htm** page from the web server running on tutorialspoint.com



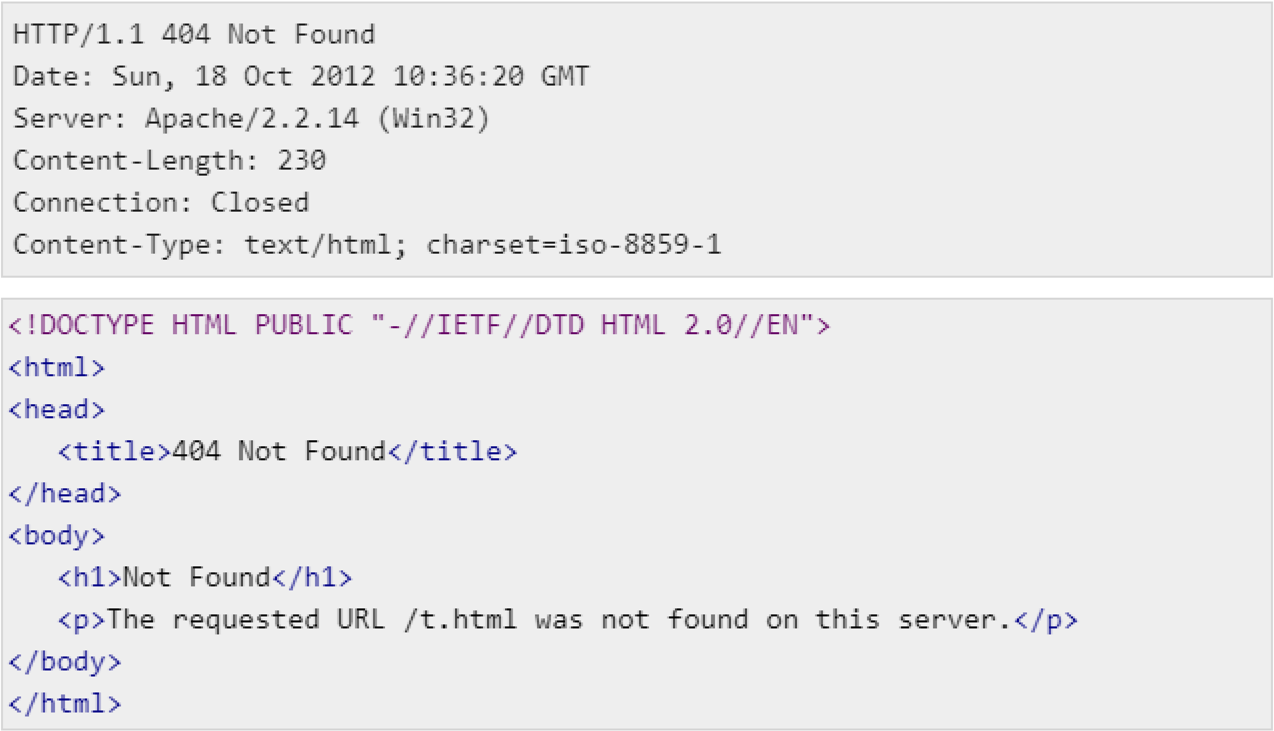
```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

```
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```



S.N.	Code and Description
1	**1xx: Informational** It means the request was received and the process is continuing.
2	**2xx: Success** It means the action was successfully received, understood, and accepted.
3	**3xx: Redirection** It means further action must be taken in order to complete the request.
4	**4xx: Client Error** It means the request contains incorrect syntax or cannot be fulfilled.
5	**5xx: Server Error** It means the server failed to fulfill an apparently valid request.

The following example shows an HTTP response message displaying error condition when the web server could not find the requested page:



```
HTTP/1.1 404 Not Found
Date: Sun, 18 Oct 2012 10:36:20 GMT
Server: Apache/2.2.14 (Win32)
Content-Length: 230
Connection: Closed
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
   <title>404 Not Found</title>
</head>
<body>
   <h1>Not Found</h1>
   <p>The requested URL /t.html was not found on this server.</p>
</body>
</html>
```

FTP

- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
- It is also used for downloading the files to computer from other servers.

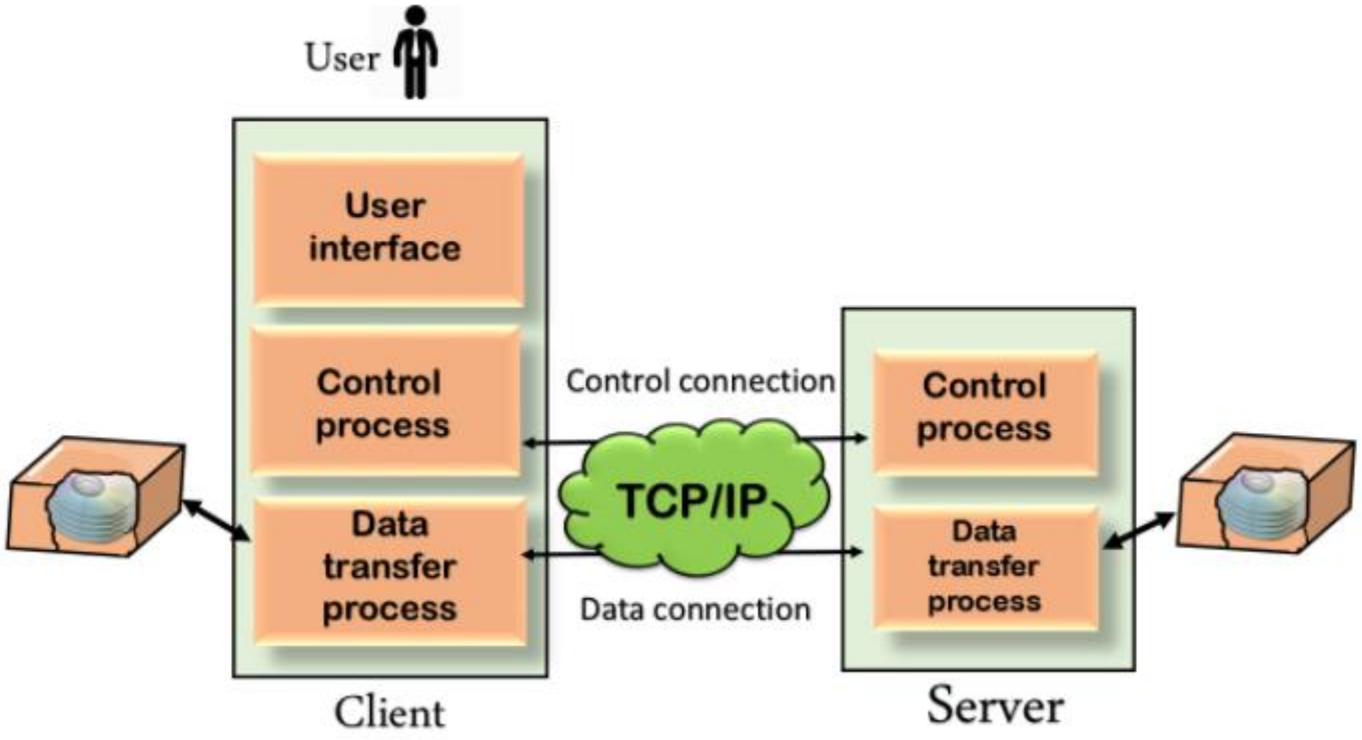
Objectives of FTP

- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

Why FTP?

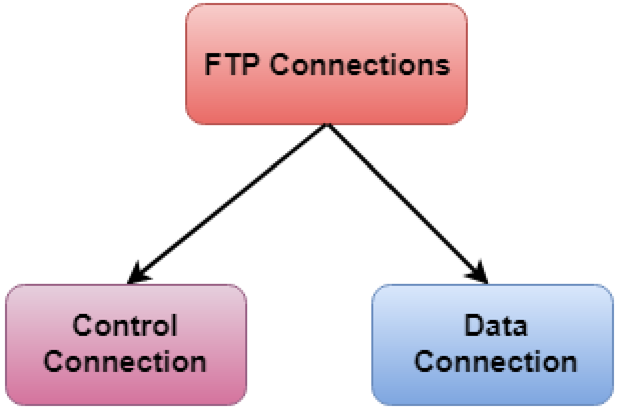
Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. FTP protocol overcomes these problems by establishing two connections between hosts. One connection is used for data transfer, and another connection is used for the control connection.

Mechanism of FTP



> The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The server has two components: the server control process and the server data transfer process.

There are two types of connections in FTP:



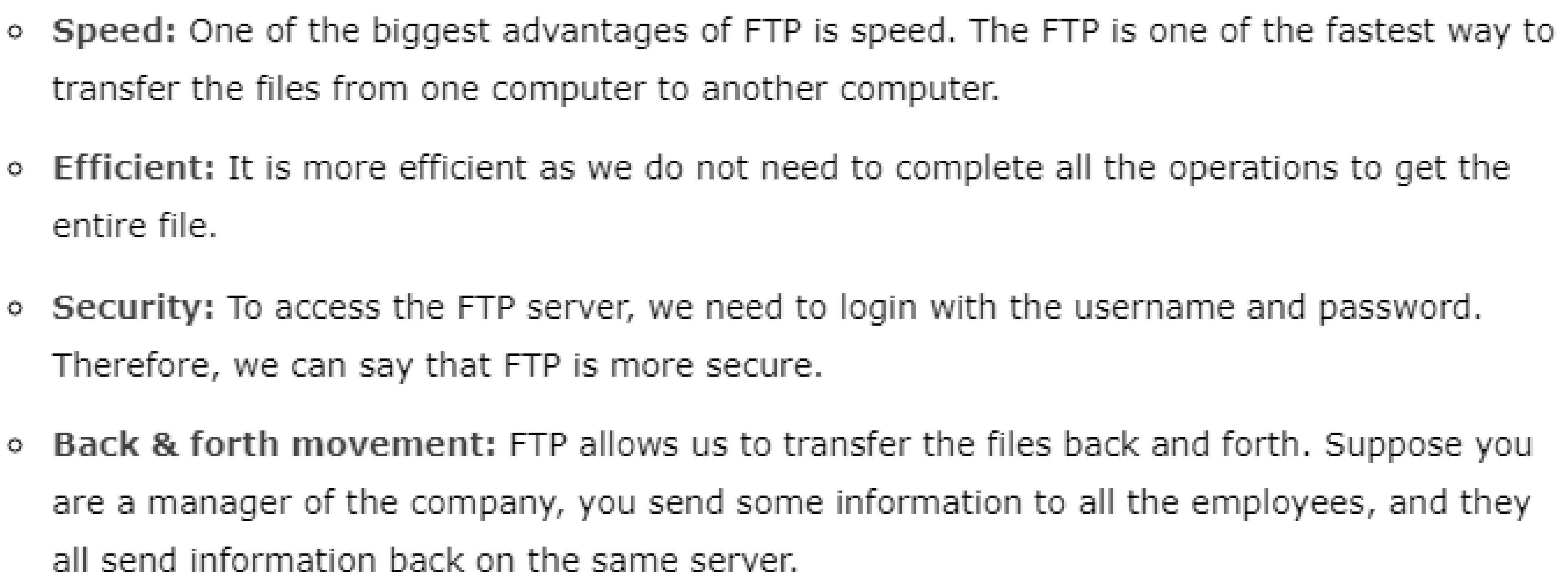
- **Control Connection:** The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time. The control connection is made between the control processes. The control connection remains connected during the entire interactive FTP session.

- **Data Connection:** The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes. The data connection opens when a command comes for transferring the files and closes when the file is transferred.

FTP Clients

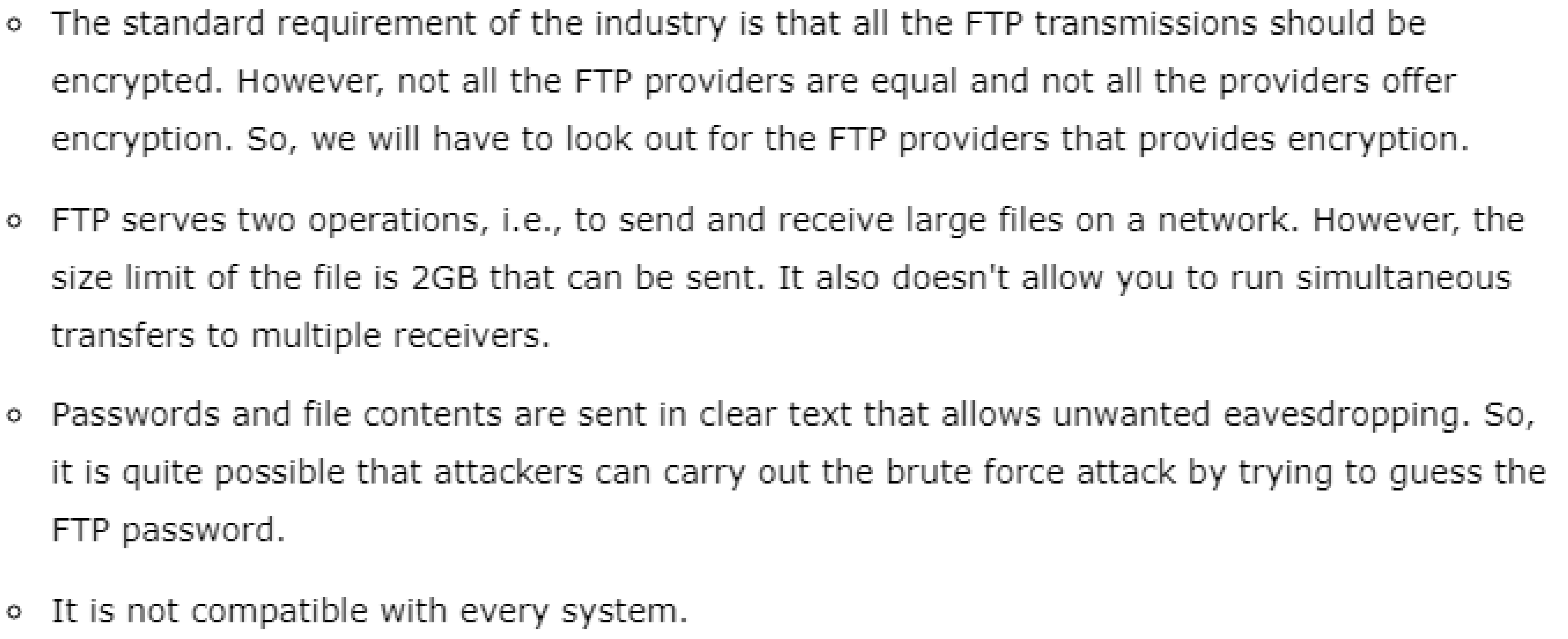
- FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.
- It allows a user to connect to a remote host and upload or download the files.
- It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.
- The FTP program is also available as a built-in component in a Web browser. This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

Advantages of FTP:



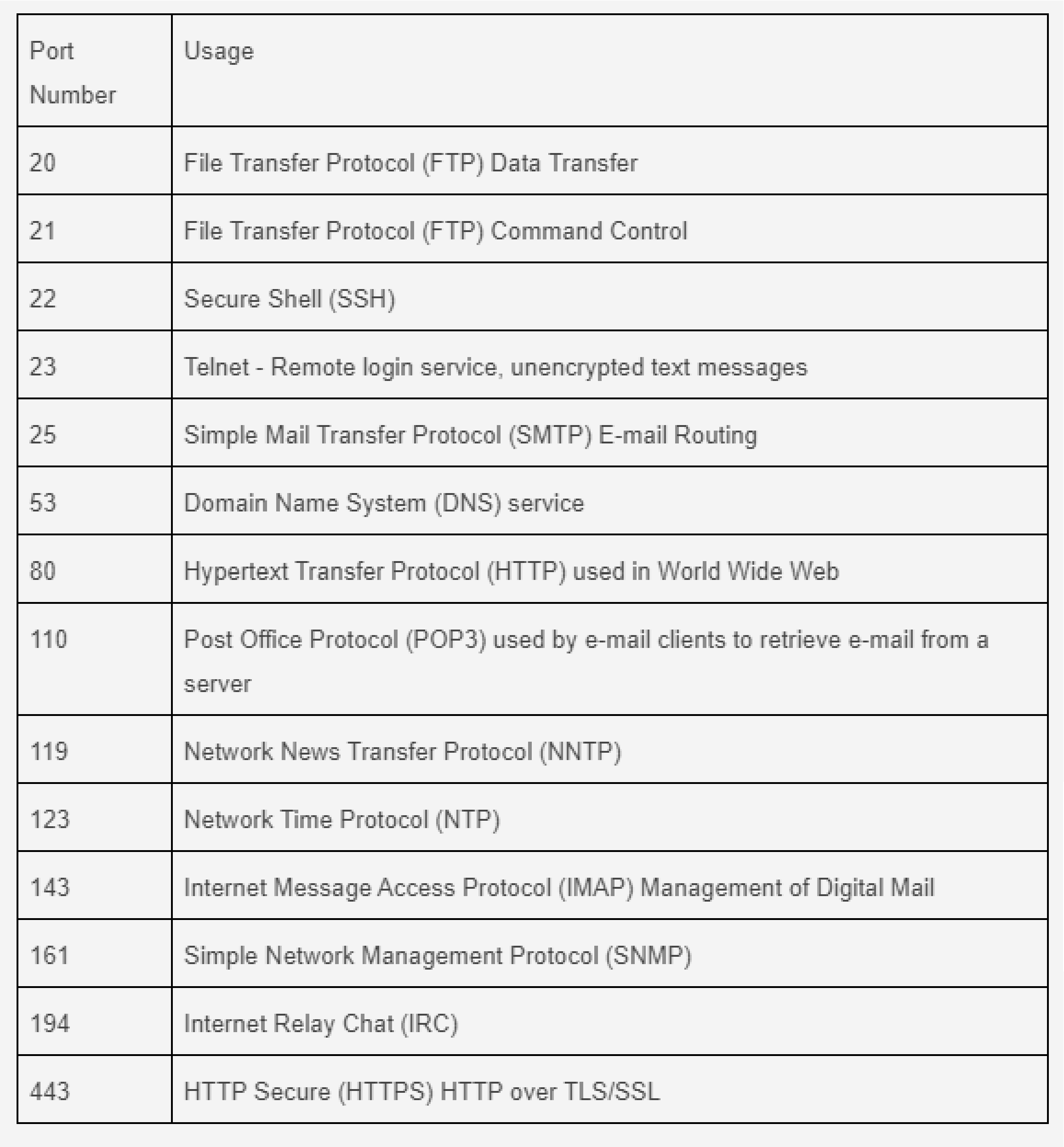
- **Speed:** One of the biggest advantages of FTP is speed. The FTP is one of the fastest way to transfer the files from one computer to another computer.
- **Efficient:** It is more efficient as we do not need to complete all the operations to get the entire file.
- **Security:** To access the FTP server, we need to login with the username and password. Therefore, we can say that FTP is more secure.
- **Back & forth movement:** FTP allows us to transfer the files back and forth. Suppose you are a manager of the company, you send some information to all the employees, and they all send information back on the same server.

Disadvantages of FTP:



- The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provides encryption.
- FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.
- Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.
- It is not compatible with every system.

14 common network ports you should know



Port Number	Usage
20	File Transfer Protocol (FTP) Data Transfer
21	File Transfer Protocol (FTP) Command Control
22	Secure Shell (SSH)
23	Telnet - Remote login service, unencrypted text messages
25	Simple Mail Transfer Protocol (SMTP) E-mail Routing
53	Domain Name System (DNS) service
80	Hypertext Transfer Protocol (HTTP) used in World Wide Web
110	Post Office Protocol (POP3) used by e-mail clients to retrieve e-mail from a server
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol (NTP)
143	Internet Message Access Protocol (IMAP) Management of Digital Mail
161	Simple Network Management Protocol (SNMP)
194	Internet Relay Chat (IRC)
443	HTTP Secure (HTTPS) HTTP over TLS/SSL

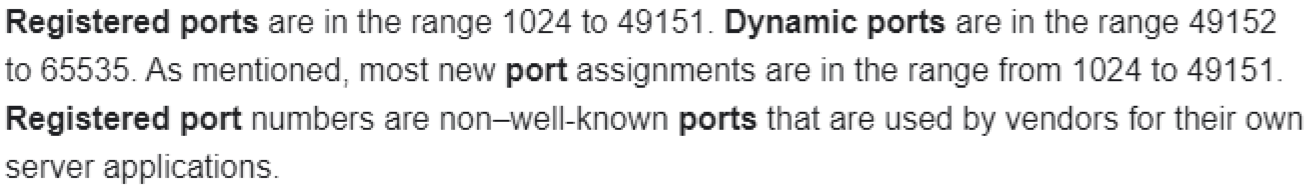
The physical ports on your computer allow communicate with peripheral devices such as your keyboard and mouse and to connect with internet devices via Ethernet cables.

Witin computer networking, ports serve a similar purpose. When a computer system seeks to connect to another computer, the port serves as a communication endpoint. It is also possible for different services running on the same computer to expose various ports and communicate with one another using these ports. In simple terms, if a software application or service needs to communicate with others, it will expose a port.

Ports are identified with positive 16-bit unsigned integers, ranging from 0 to 65535.

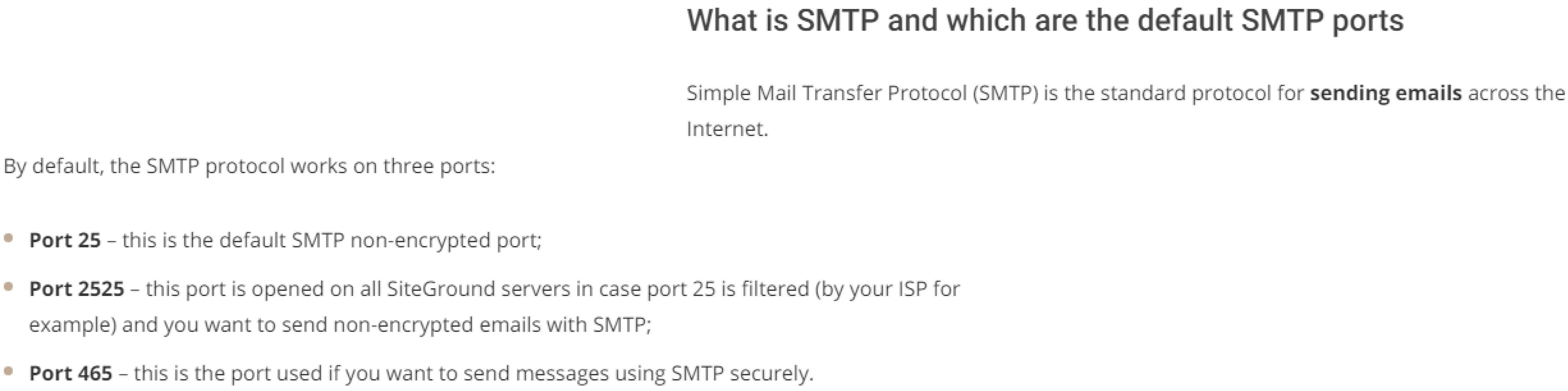
Port numbers are divided into three ranges: well-known ports, registered ports, and dynamic or private ports.

Well-known ports (also known as system ports) are numbered from 0 through 1023.



Registered ports are in the range 1024 to 49151. **Dynamic ports** are in the range 49152 to 65535. As mentioned, most new **port** assignments are in the range from 1024 to 49151. **Registered port** numbers are non–well-known **ports** that are used by vendors for their own server applications.

Email Protocols - POP3, SMTP and IMAP



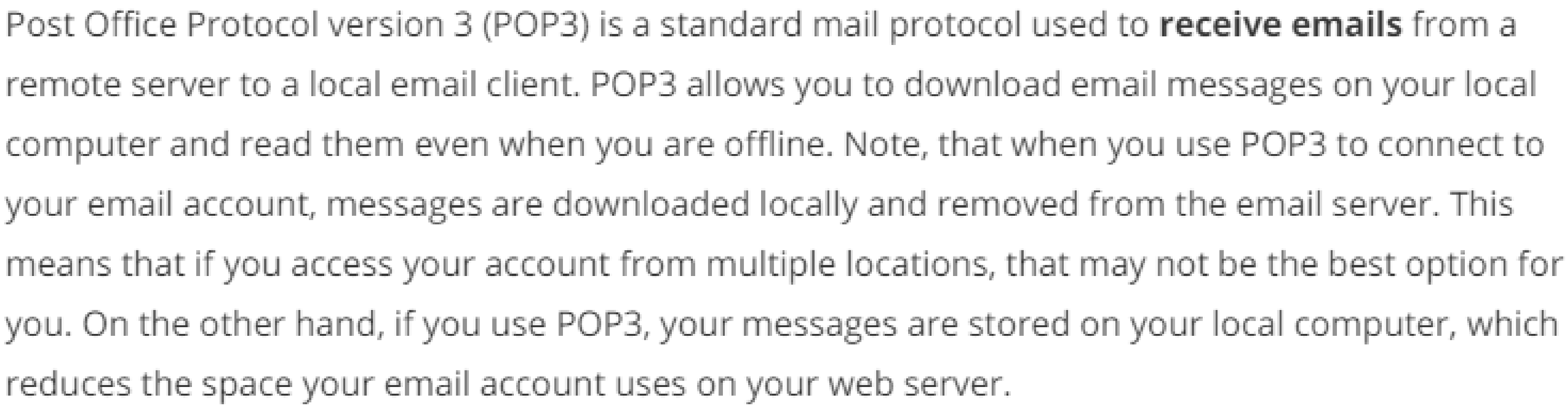
What is SMTP and which are the default SMTP ports

Simple Mail Transfer Protocol (SMTP) is the standard protocol for **sending emails** across the Internet.

By default, the SMTP protocol works on three ports:

- **Port 25** – this is the default SMTP non-encrypted port;
- **Port 2525** – this port is opened on all SiteGround servers in case port 25 is filtered (by your ISP for example) and you want to send non-encrypted emails with SMTP;
- **Port 465** – this is the port used if you want to send messages using SMTP securely.

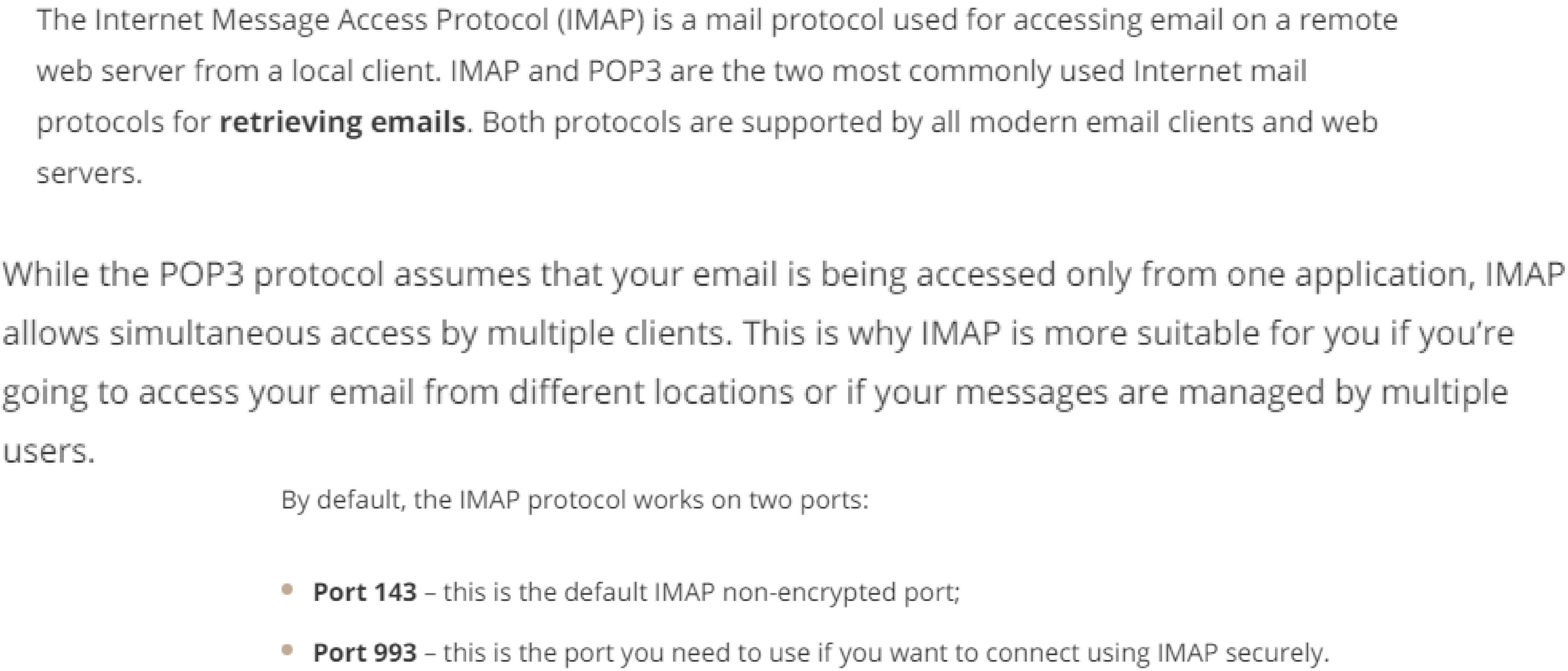
What is POP3 and which are the default POP3 ports



Post Office Protocol version 3 (POP3) is a standard mail protocol used to **receive emails** from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline. Note, that when you use POP3 to connect to your email account, messages are downloaded locally and removed from the email server. This means that if you access your account from multiple locations, that may not be the best option for you. On the other hand, if you use POP3, your messages are stored on your local computer, which reduces the space your email account uses on your web server.

By default, the POP3 protocol works on two ports:
•Port 110 – this is the default POP3 non-encrypted port;
•Port 995 – this is the port you need to use if you want to connect using POP3 securely.

What is IMAP and which are the default IMAP ports



The Internet Message Access Protocol (IMAP) is a mail protocol used for accessing email on a remote web server from a local client. IMAP and POP3 are the two most commonly used Internet mail protocols for **retrieving emails**. Both protocols are supported by all modern email clients and web servers.

While the POP3 protocol assumes that your email is being accessed only from one application, IMAP allows simultaneous access by multiple clients. This is why IMAP is more suitable for you if you're going to access your email from different locations or if your messages are managed by multiple users.

By default, the IMAP protocol works on two ports:

- **Port 143** – this is the default IMAP non-encrypted port;
- **Port 993** – this is the port you need to use if you want to connect using IMAP securely.

1. What is DNS?
2. What is dynamic web page?

11. Explain HTTP request message in brief.