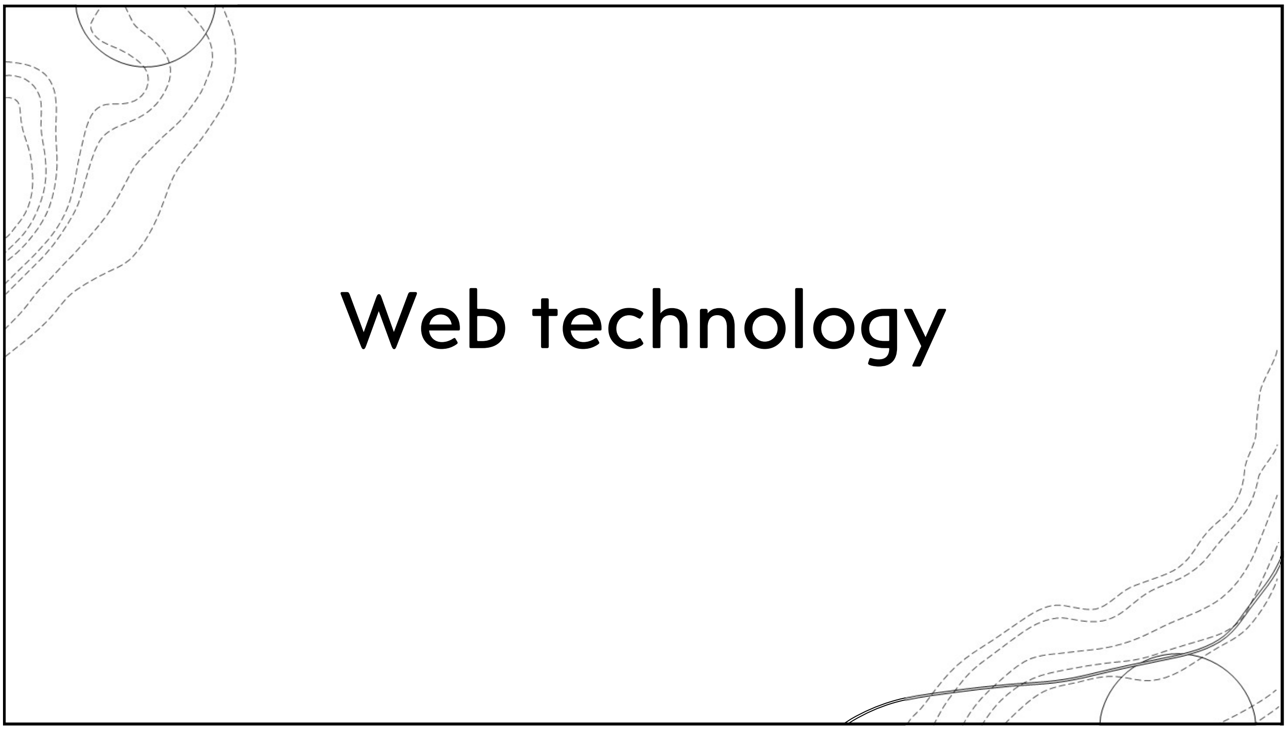
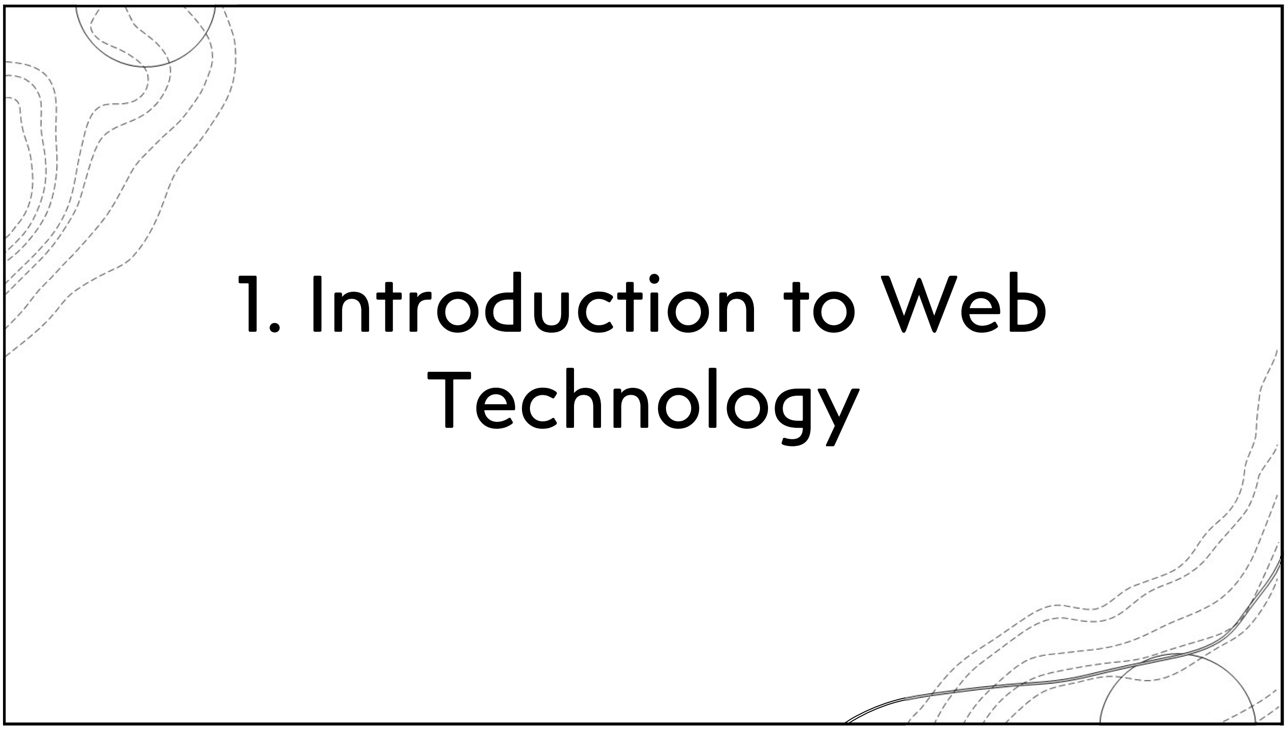
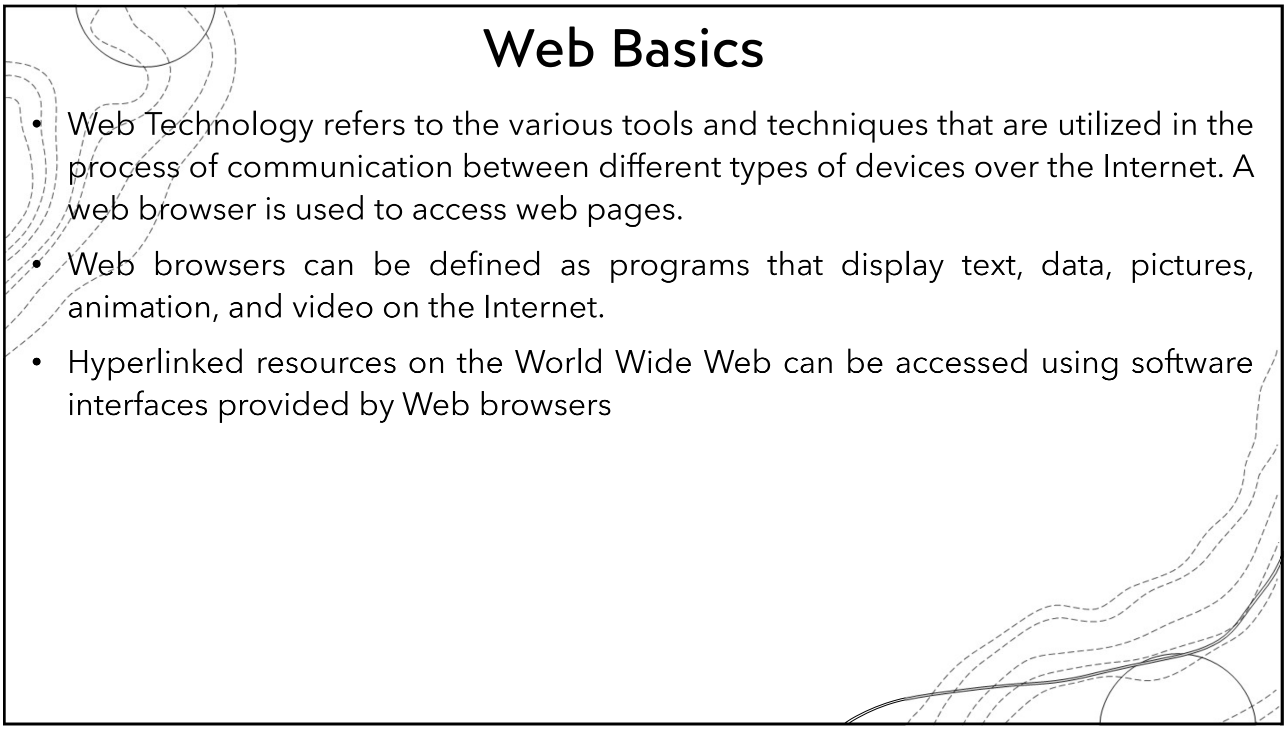


Web technology

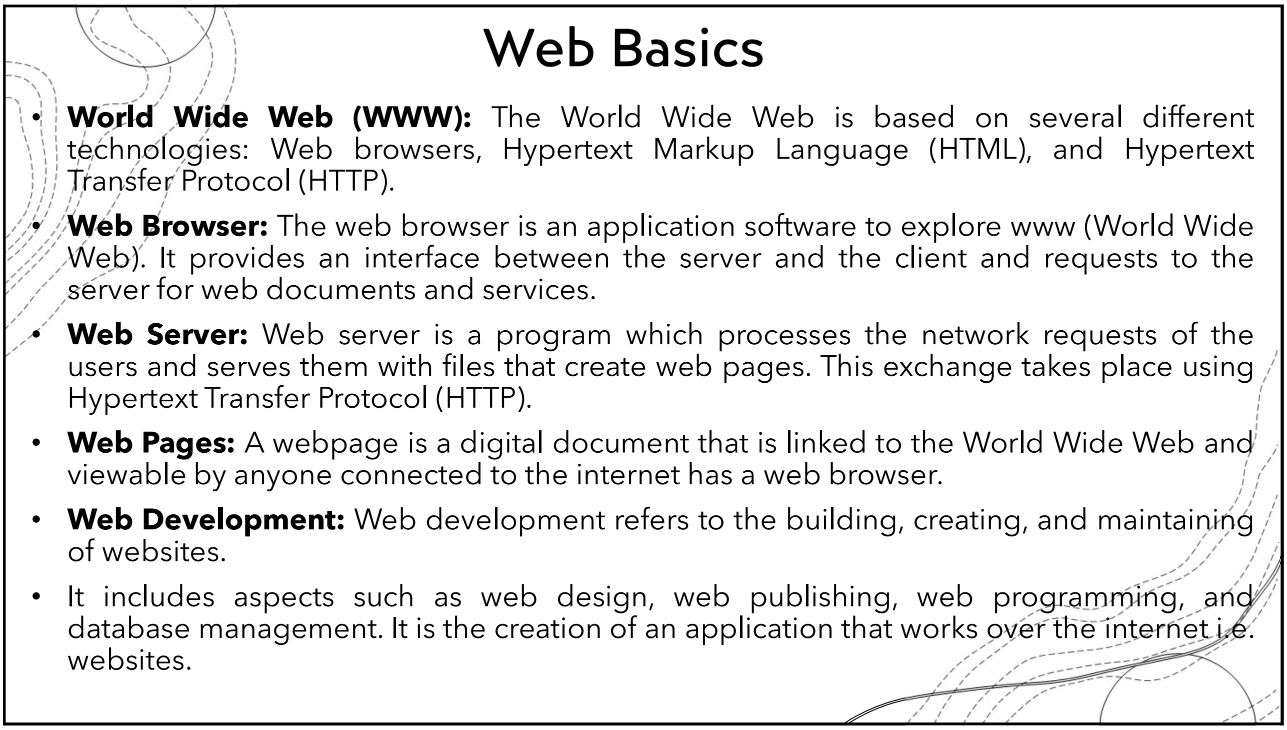


1. Introduction to Web Technology



Web Basics

- 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. A web browser is used to access web pages.
- Web browsers can be defined as programs that display text, data, pictures, animation, and video on the Internet.
- Hyperlinked resources on the World Wide Web can be accessed using software interfaces provided by Web browsers



Web Basics

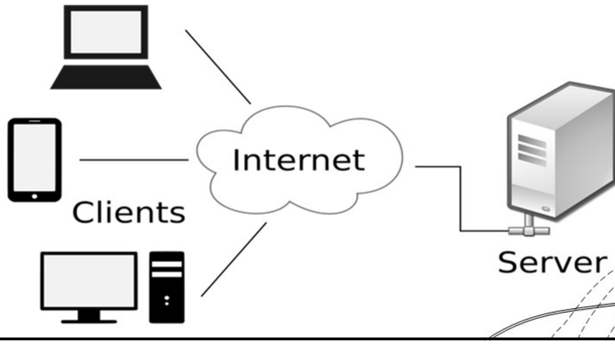
- **World Wide Web (WWW):** The World Wide Web is based on several different technologies: Web browsers, Hypertext Markup Language (HTML), and Hypertext Transfer Protocol (HTTP).
- **Web Browser:** The web browser is an application software to explore www (World Wide Web). It provides an interface between 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.
- It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

Tier Technology

- Tier technology refers to the architectural approach used in software development to separate the different components or layers of an application.
- This separation helps in organizing the codebase, improving maintainability, scalability, and facilitating collaboration among developers.
- In the context of software architecture, there are several variations and approaches to tiered architectures, each with its own characteristics and benefits. Here are some common types of tier technology:

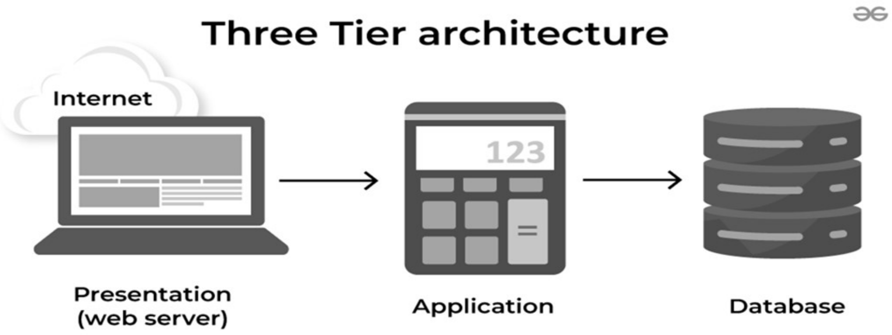
Two-Tier Technology

- Also known as client-server architecture.
- Consists of two main tiers: client and server.
- Clients handle the presentation layer and user interaction, while servers manage the application logic and data storage.
- Often used for simple applications where both the client and server components are tightly coupled.



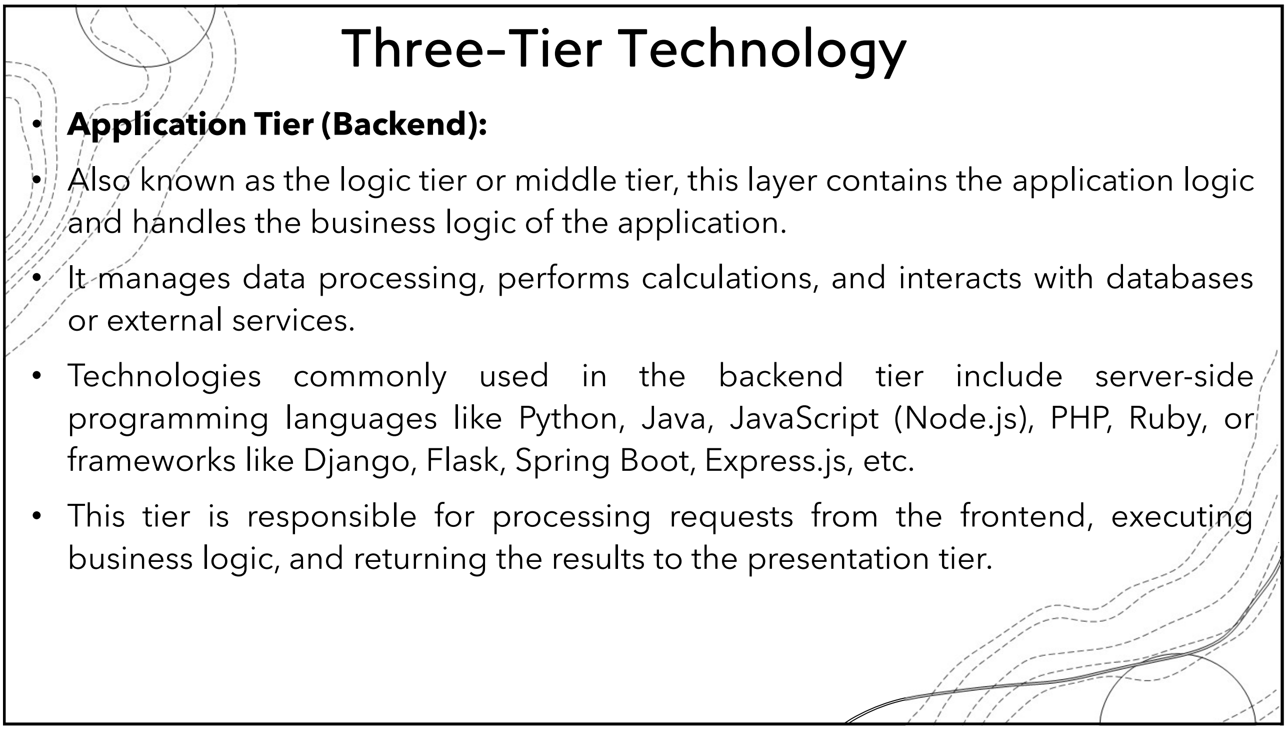
Three-Tier Technology

- Three-Tier Architecture is an is an well established software application design pattern which will organizes the application in the three logical and physical computing tiers as following:
- Presentation Tier
- Application Tier
- Data Tier



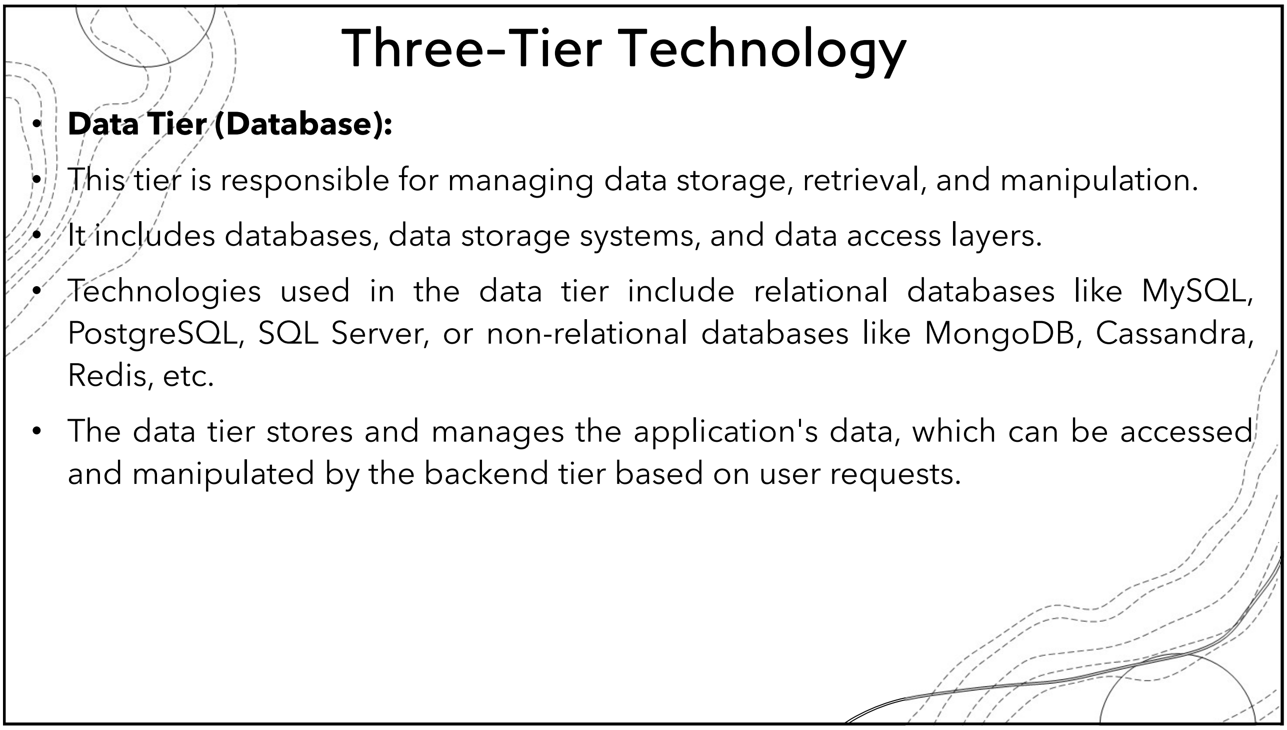
Three-Tier Technology

- **Presentation Tier (Frontend):**
- This tier is responsible for presenting the user interface to the end-users.
- It includes technologies such as HTML, CSS, JavaScript, and frontend frameworks like React.js, Angular, or Vue.js.
- The presentation tier interacts directly with the user and sends requests to the backend tier for data retrieval or processing.



Three-Tier Technology

- **Application Tier (Backend):**
- Also known as the logic tier or middle tier, this layer contains the application logic and handles the business logic of the application.
- It manages data processing, performs calculations, and interacts with databases or external services.
- Technologies commonly used in the backend tier include server-side programming languages like Python, Java, JavaScript (Node.js), PHP, Ruby, or frameworks like Django, Flask, Spring Boot, Express.js, etc.
- This tier is responsible for processing requests from the frontend, executing business logic, and returning the results to the presentation tier.



Three-Tier Technology

- **Data Tier (Database):**
- This tier is responsible for managing data storage, retrieval, and manipulation.
- It includes databases, data storage systems, and data access layers.
- Technologies used in the data tier include relational databases like MySQL, PostgreSQL, SQL Server, or non-relational databases like MongoDB, Cassandra, Redis, etc.
- The data tier stores and manages the application's data, which can be accessed and manipulated by the backend tier based on user requests.

Multi-Tier Architecture

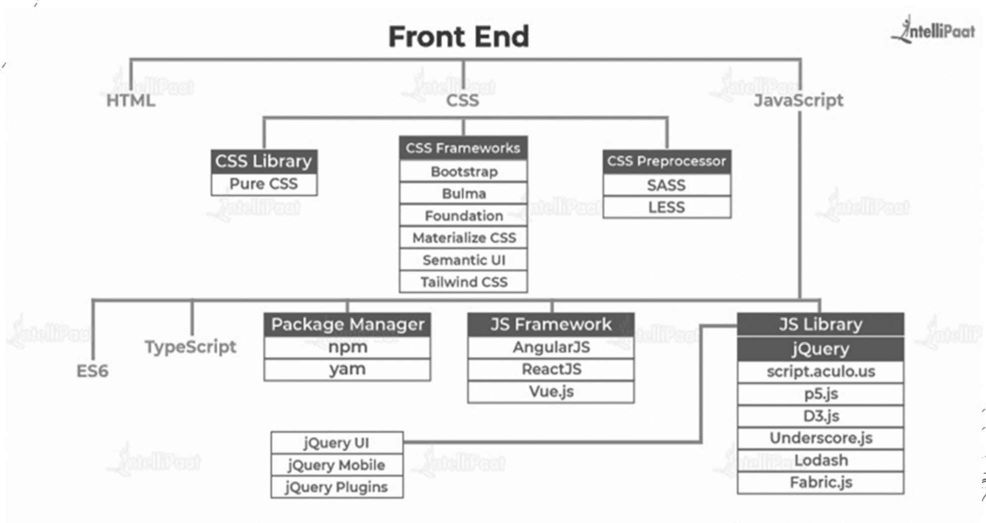
- Extends the three-tier architecture by adding additional tiers for specific purposes.
- Common additional tiers include caching tiers for performance optimization, load balancing tiers for distributing incoming requests, and service tiers for managing reusable business logic or microservices.

N-Tier Architecture

- Generalizes the concept of multi-tier architecture by allowing an arbitrary number of tiers.
- Each tier performs a specific set of functions, and communication typically occurs between adjacent tiers.
- Provides flexibility and scalability but can introduce complexity, especially in large systems.

Web Development can be Classified into Two Ways:

- **Frontend Development:** The part of a website that the user interacts directly is termed as front end. It is also referred to as the 'client side' of the application.



Front End

HTML

CSS

JavaScript

CSS Library: Pure CSS

CSS Frameworks: Bootstrap, Bulma, Foundation, Materialize CSS, Semantic UI, Tailwind CSS

CSS Preprocessor: SASS, LESS

ES6

TypeScript

Package Manager: npm, yam

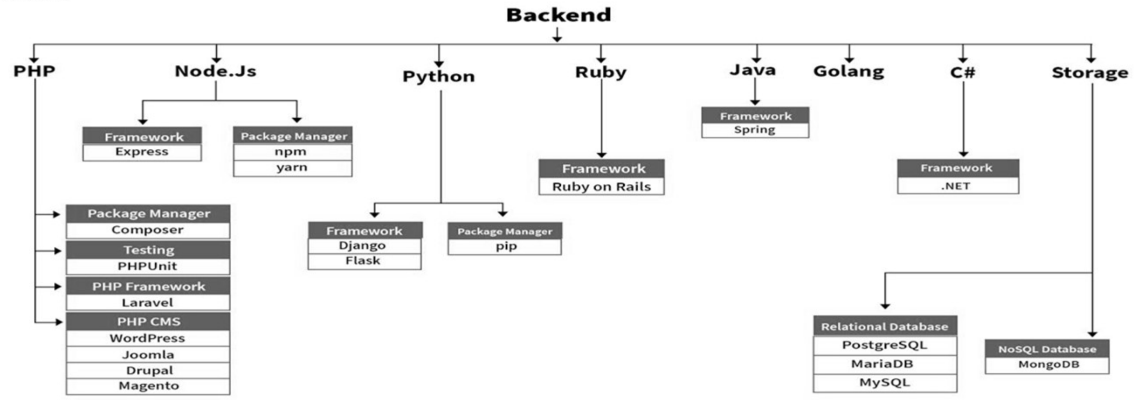
JS Framework: AngularJS, ReactJS, Vue.js

JS Library: jQuery, script.aculo.us, p5.js, D3.js, Underscore.js, Lodash, Fabric.js

jQuery UI, jQuery Mobile, jQuery Plugins

Web Development can be Classified into Two Ways:

- **Backend Development:** Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.



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

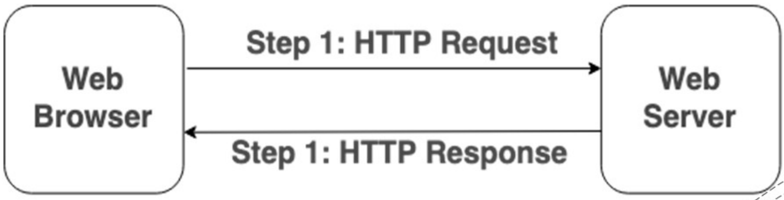
Golang

C#: Framework: .NET

Storage: Relational Database: PostgreSQL, MariaDB, MySQL; NoSQL Database: MongoDB

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.



Static Web Page

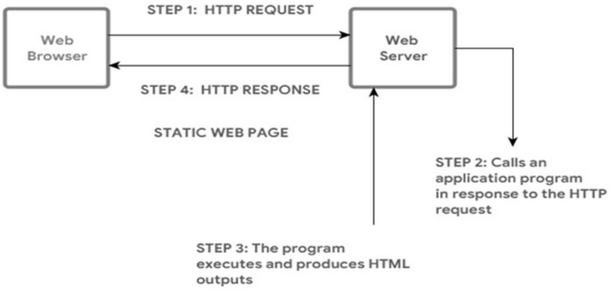
Dynamic web pages

Dynamic Web Pages are written in languages such as CGI, AJAX, ASP, ASP.NET, etc.

In dynamic web pages, the Content of pages is different for different visitors.

It takes more time to load than the static web page.

Dynamic web pages are used where the information is changed frequently, for example, stock prices, weather information, etc.


STEP 1: HTTP REQUEST
Web Browser
Web Server
STEP 4: HTTP RESPONSE
STATIC WEB PAGE
STEP 2: Calls an application program in response to the HTTP request
STEP 3: The program executes and produces HTML outputs

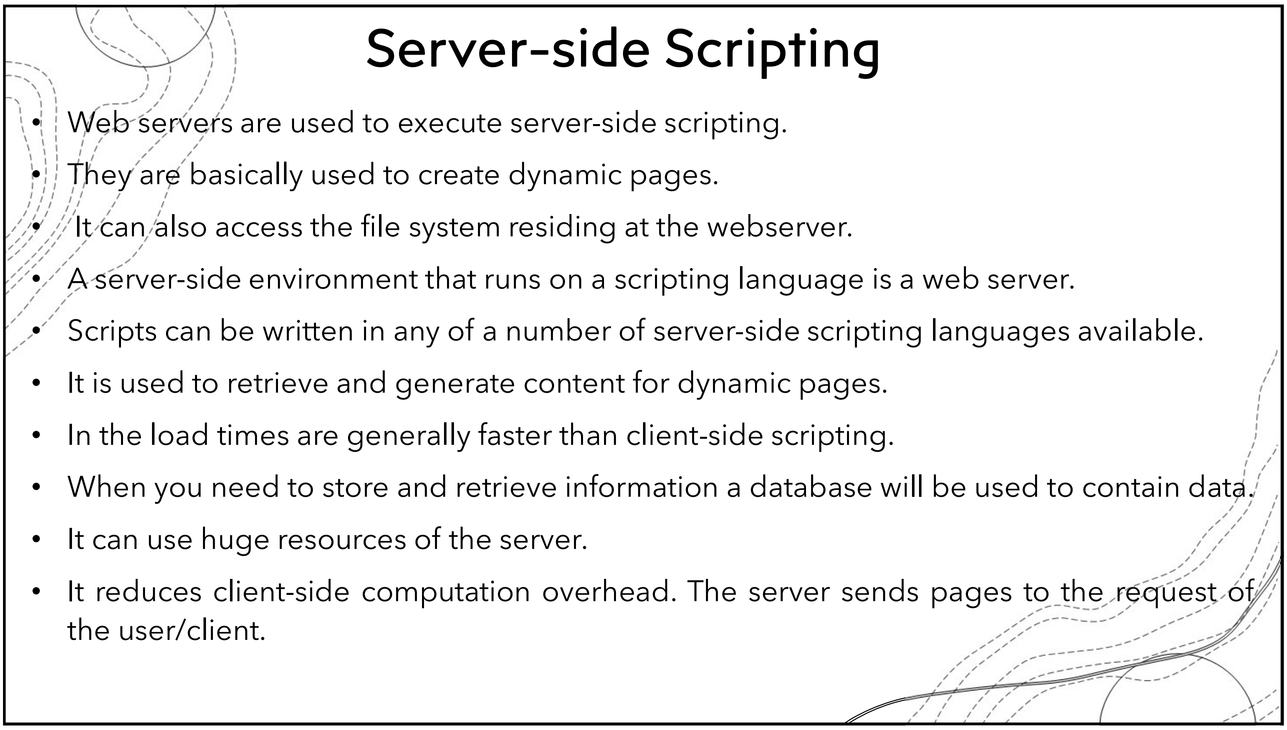
Dynamic web page

Static web page vs dynamic web page

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.
7.	Static web pages does not contain any application program .	Dynamic web pages contains application program for different services.
8.	Static web pages require less work and cost in designing them.	Dynamic web pages require comparatively more work and cost in designing them.

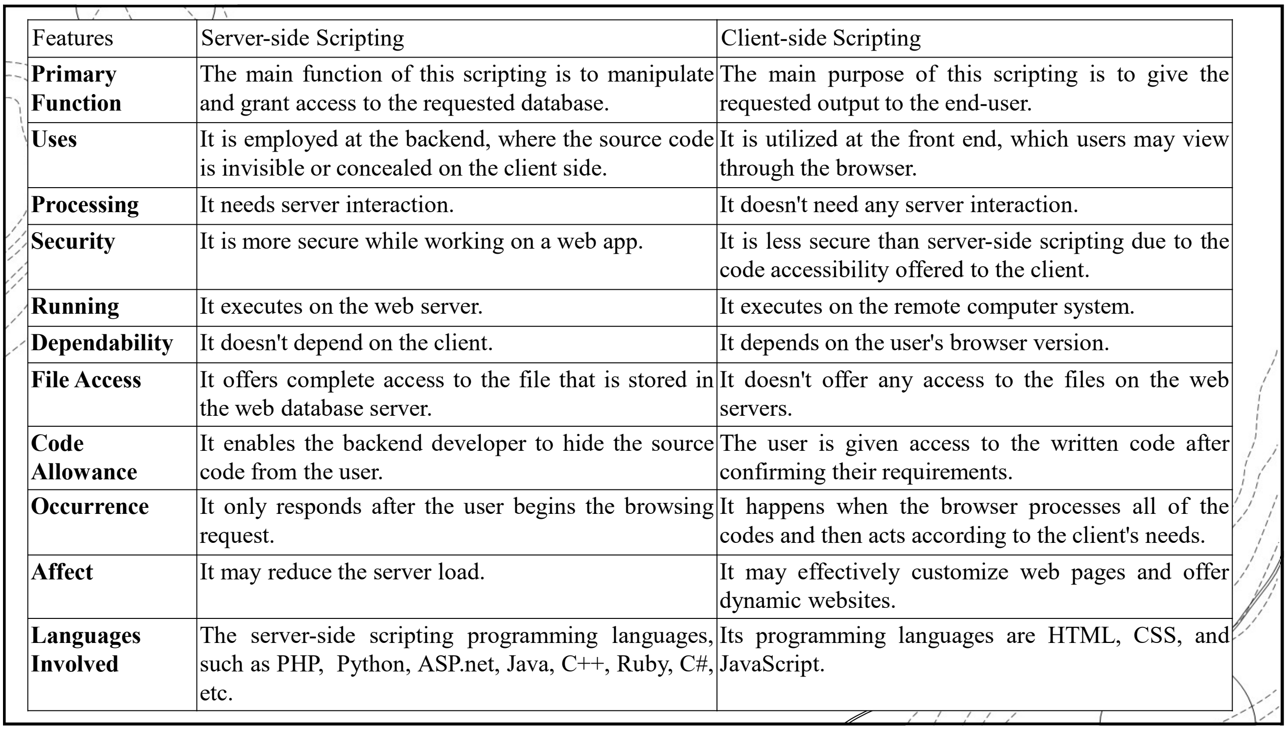
1.Client-side Scripting

- Web browsers execute client-side scripting.
- It is used when browsers have all code.
- Source code is used to transfer from webserver to user's computer over the internet and run directly on browsers.
- It is also used for validations and functionality for user events.
- It allows for more interactivity.
- It usually performs several actions without going to the user.
- It cannot be basically used to connect to databases on a web server.
- These scripts cannot access the file system that resides in the web browser.
- Pages are altered on basis of the user's choice.
- It can also be used to create "cookies" that store data on the user's computer.



Server-side Scripting

- Web servers are used to execute server-side scripting.
- They are basically used to create dynamic pages.
- It can also access the file system residing at the webserver.
- A server-side environment that runs on a scripting language is a web server.
- Scripts can be written in any of a number of server-side scripting languages available.
- It is used to retrieve and generate content for dynamic pages.
- In the load times are generally faster than client-side scripting.
- When you need to store and retrieve information a database will be used to contain data.
- It can use huge resources of the server.
- It reduces client-side computation overhead. The server sends pages to the request of the user/client.



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, Python, ASP.net, Java, C++, Ruby, C#, etc.	Its programming languages are HTML, CSS, and JavaScript.

Internet(web) 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.

Need of protocols

- It may be that the sender and receiver of data are parts of different networks, located in different parts of the world having different data transfer rates.
- So, we need protocols to manage the flow control of data, and access control of the link being shared in the communication channel.
- Suppose there is a sender X who has a data transmission rate of 10 Mbps. And, there is a receiver Y who has a data receiving rate of 5Mbps.
- Since the rate of receiving the data is slow so some data will be lost during transmission.
- In order to avoid this, receiver Y needs to inform sender X about the speed mismatch so that sender X can adjust its transmission rate.
- Similarly, the access control decides the node which will access the link shared in the communication channel at a particular instant in time.
- If not the transmitted data will collide if many computers send data simultaneously through the same link resulting in the corruption or loss of data.

Types of protocols

Internet Protocols are of different types having different uses. These are mentioned below:

- TCP/IP(Transmission Control Protocol/ Internet Protocol)
- SMTP(Simple Mail Transfer Protocol)
- PPP(Point-to-Point Protocol)
- FTP (File Transfer Protocol)
- SFTP(Secure File Transfer Protocol)
- HTTP(Hyper Text Transfer Protocol)
- HTTPS(HyperText Transfer Protocol Secure)
- TELNET(Terminal Network)
- POP3(Post Office Protocol 3)
- IPv4
- IPv6
- ICMP
- UDP
- IMAP
- SSH
- Gopher

TCP

TCP (Transmission Control Protocol) is a method used on the internet to ensure that data is sent and received accurately and in the correct order.

Key Points about TCP:

Connection-Oriented:
TCP creates a connection between two devices (like your computer and a website) before starting to send data. This connection ensures that both devices are ready to communicate.

Reliable Data Transfer:
TCP makes sure that all data sent reaches its destination correctly. If any data is lost or gets mixed up, TCP resends it.

Error Checking:
TCP checks for errors in the data being sent. If it finds an error, it corrects it by resending the affected data.

Flow Control:
TCP controls the flow of data so that the sender does not send data too quickly for the receiver to handle.

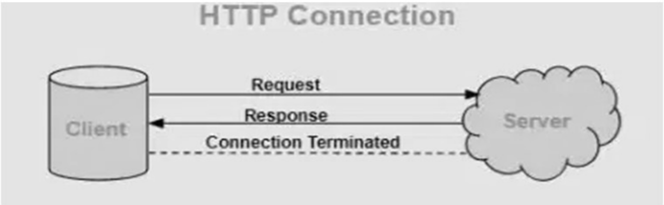
Congestion Control:
TCP adjusts the rate of data transmission if it detects that the network is getting too busy, to prevent data loss and delays.

Http (HyperText Transfer protocol)

- HTTP stands for HyperText Transfer Protocol.
- Tim Berner invents it. HyperText is the type of text that is specially coded with the help of some standard coding language called HyperText Markup Language (HTML).
- HTTP/2 is the new version of HTTP.
- HTTP/3 is the latest version of HTTP, which is published in 2022.

- The protocol used to transfer hypertext between two computers is known as HyperText Transfer Protocol.
- HTTP provides a standard between a web browser and a web server to establish communication.
- It is a set of rules for transferring data from one computer to another.
- Data such as text, images, and other multimedia files are shared on the World Wide Web.
- Whenever a web user opens their web browser, the user indirectly uses HTTP.
- It is an application protocol that is used for distributed, collaborative, hypermedia information systems.

Http (HyperText Transfer protocol)



Http (HyperText Transfer protocol)

- HTTP is IP based communication protocol that is used to deliver data from server to client or vice-versa.
- The server processes a request, which is raised by the client, and also server and client know each other only during the current bid and response period.
- Any type of content can be exchanged as long as the server and client are compatible with it.
- Once data is exchanged, servers and clients are no longer connected.
- It is a request and response protocol based on client and server requirements.
- It is a connection-less protocol because after the connection is closed, the server does not remember anything about the client and the client does not remember anything about the server.
- It is a stateless protocol because both client and server do not expect anything from each other but they are still able to communicate.

Http (HyperText Transfer protocol)

Advantages of HTTP

- Memory usage and CPU usage are low because of fewer simultaneous connections.
- Since there are few TCP connections hence network congestion is less.
- Since handshaking is done at the initial connection stage, then latency is reduced because there is no further need for handshaking for subsequent requests.
- The error can be reported without closing the connection.
- HTTP allows HTTP pipe-lining of requests or responses.

Http (HyperText Transfer protocol)

Disadvantages of HTTP

- HTTP requires high power to establish communication and transfer data.
- HTTP is less secure because it does not use any encryption method like HTTPS and use TLS to encrypt regular HTTP requests and response.
- HTTP is not optimized for cellular phones and it is too gabby.
- HTTP does not offer a genuine exchange of data because it is less secure.
- The client does not close the connection until it receives complete data from the server; hence, the server needs to wait for data completion and cannot be available for other clients during this time.

Http (HyperText Transfer protocol)

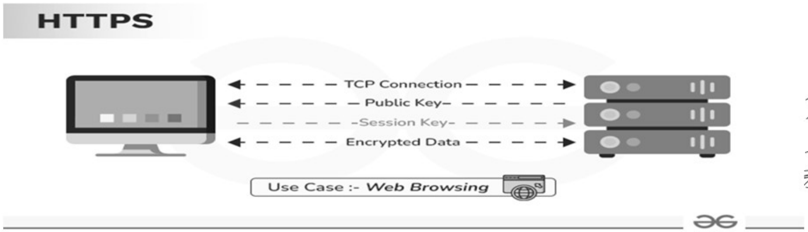
Cookies in HTTP

- An HTTP cookie (web cookie, browser cookie) is a little piece of data that a server transmits to a user’s web browser.
- When making subsequent queries, the browser may keep the cookie and transmit it back to the same server.
- An HTTP cookie is typically used, for example, to maintain a user’s login state, to determine whether two requests originate from the same browser.
- For the stateless HTTP protocol, it retains stateful information.

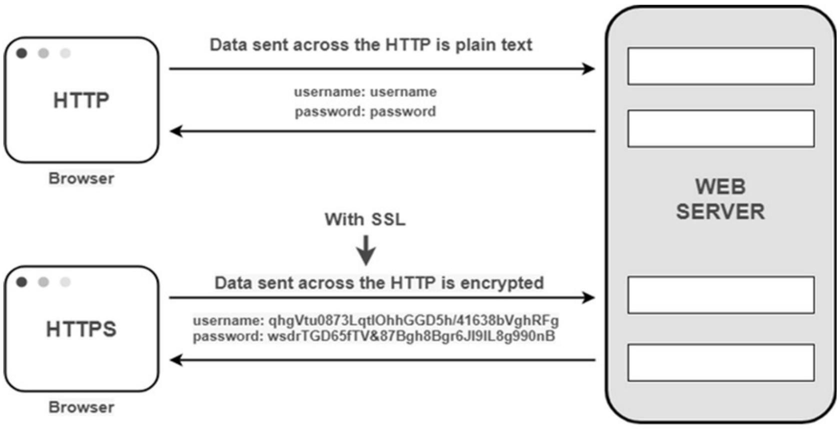
Hypertext Transfer Protocol Secure

HTTPS stands for HyperText Transfer Protocol Secure. It is the most common protocol for sending data between a web browser and a website.

- Hypertext Transfer Protocol Secure is a protocol that is used to communicate between the user browser and the website. It also helps in the transfer of data.
- It is the secure variant of HTTP.
- To make the data transfer more secure, it is encrypted.
- Encryption is required to ensure security while transmitting sensitive information like passwords, contact information, etc.



Hypertext Transfer Protocol Secure


HTTP
Browser
Data sent across the HTTP is plain text
username: username
password: password
WEB SERVER
With SSL
HTTPS
Browser
Data sent across the HTTP is encrypted
username: qhgVtu0873LqtlOhhGGD5h/41638bVghRFg
password: wsdrTGD65fTV&87Bgh8Bgr6JI9IL8g990nB

HTTP vs HTTPS

HTTP	HTTPS
HTTP stands for HyperText Transfer Protocol.	HTTPS stands for HyperText Transfer Protocol Secure.
URL begins with "http://".	URL starts with "https://".
HTTP Works at the Application Layer.	HTTPS works at Transport Layer.
HTTP speed is faster than HTTPS.	HTTPS speed is slower than HTTP.

Advantage of HTTPS
Secure Communication: HTTPS establishes a secure communication link between the communicating system by providing encryption during transmission.

Data Integrity: By encrypting the data, HTTPS ensures data integrity. This implies that even if the data is compromised at any point, the hackers won't be able to read or modify the data being exchanged.

Privacy and Security: HTTPS prevents attackers from accessing the data being exchanged passively, thereby protecting the privacy and security of the users.

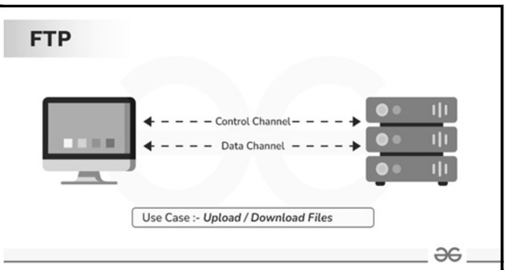
Faster Performance: TTPS encrypts the data and reduces its size. Smaller size accounts for faster data transmission in the case of HTTPS.

File Transfer Protocol

File Transfer Protocol(FTP) is an application layer protocol that moves files between local and remote file systems. It runs on top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.

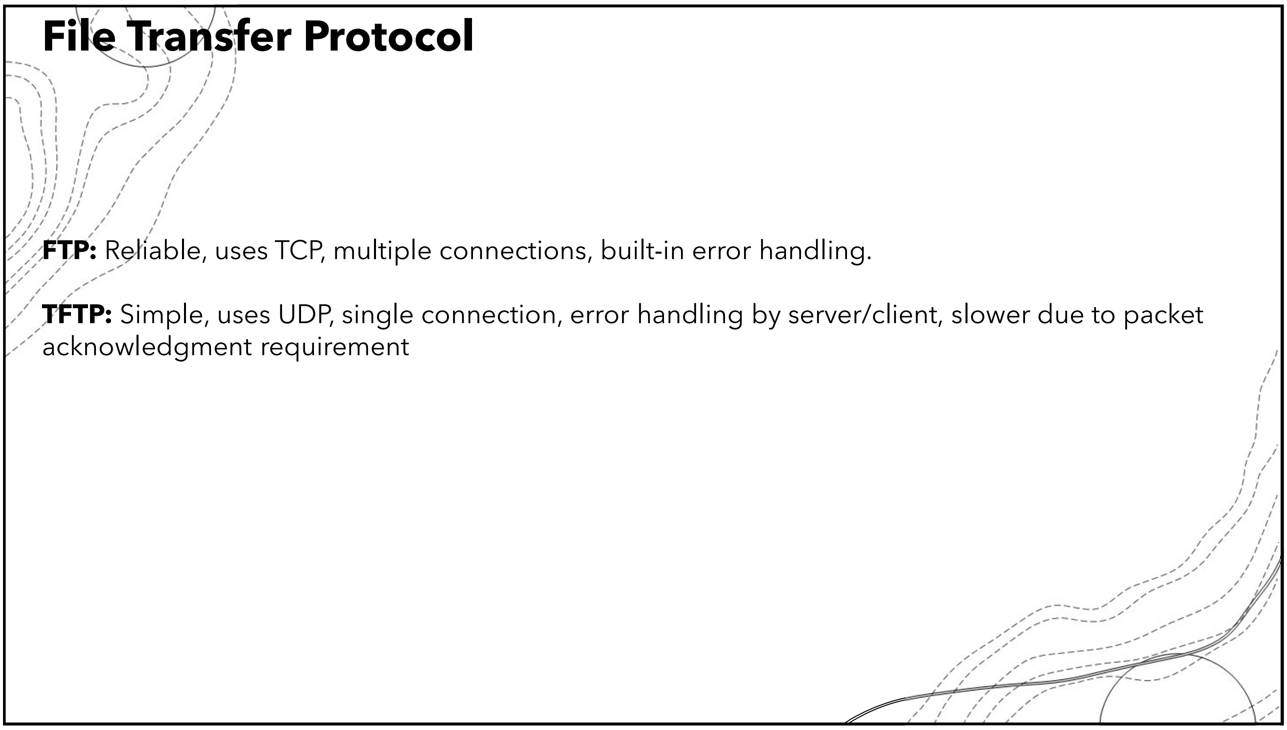
- FTP is a standard communication protocol.
- There are various other protocols like HTTP which are used to transfer files between computers, but they lack clarity and focus as compared to FTP.
- Moreover, the systems involved in connection are heterogeneous, i.e. they differ in operating systems, directories, structures, character sets, etc the FTP shields the user from these differences and transfers data efficiently and reliably.
- FTP can transfer ASCII, or image files.
- The ASCII is the default file share format, in this, each character is encoded by NVT ASCII.

File Transfer Protocol



Characteristics of FTP

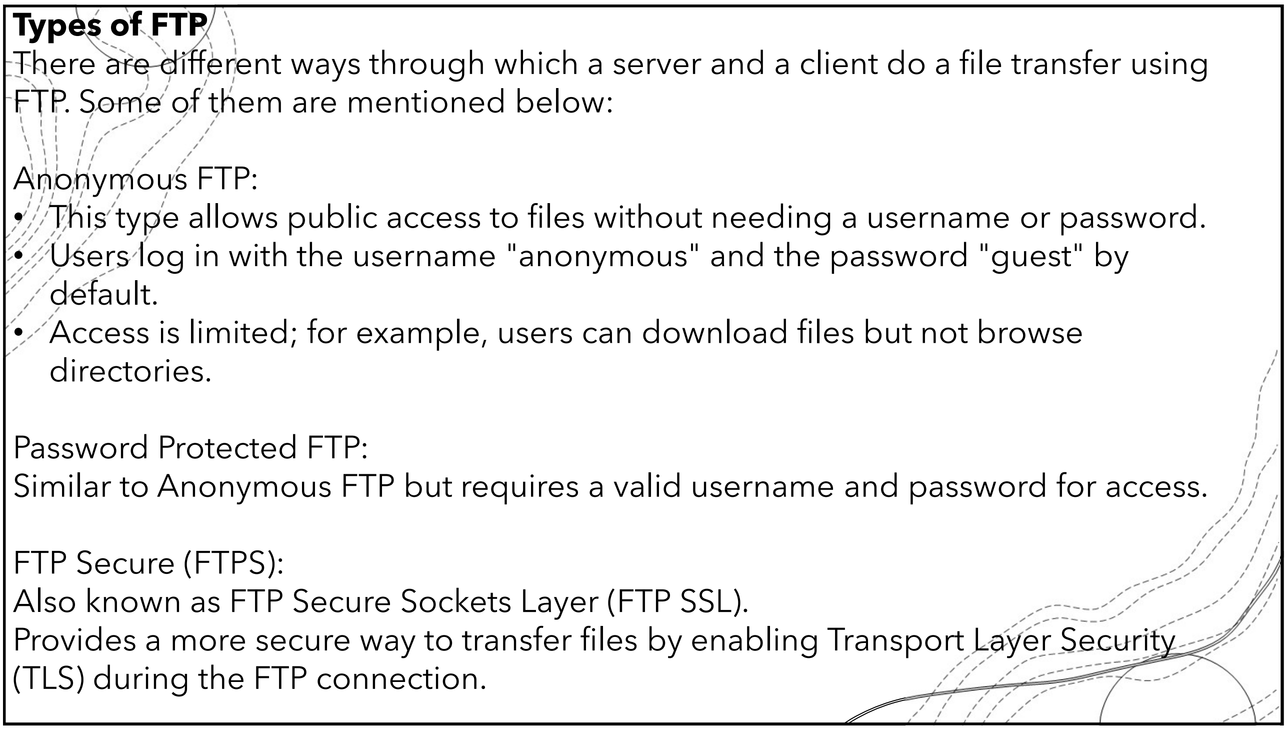
- FTP uses TCP as a transport layer protocol.
- It is good for simple file transfers, such as during boot time.
- Errors in the transmission (lost packets, checksum errors) must be handled by the TFTP server.
- It uses only one connection through well-known port 69.
- TFTP uses a simple lock-step protocol (each data packet needs to be acknowledged). Thus the throughput is limited.



File Transfer Protocol

FTP: Reliable, uses TCP, multiple connections, built-in error handling.

TFTP: Simple, uses UDP, single connection, error handling by server/client, slower due to packet acknowledgment requirement



Types of FTP

There are different ways through which a server and a client do a file transfer using FTP. Some of them are mentioned below:

Anonymous FTP:

- This type allows public access to files without needing a username or password.
- Users log in with the username "anonymous" and the password "guest" by default.
- Access is limited; for example, users can download files but not browse directories.

Password Protected FTP:

Similar to Anonymous FTP but requires a valid username and password for access.

FTP Secure (FTPS):

Also known as FTP Secure Sockets Layer (FTP SSL).

Provides a more secure way to transfer files by enabling Transport Layer Security (TLS) during the FTP connection.

File Transfer Protocol

Types of Connection in FTP

1. Control Connection
2. Data Connection

File Transfer Protocol

Control Connection

- For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of a control connection.
- The control connection is initiated on port number 21.

Data connection

- For sending the actual file, FTP makes use of a data connection. A data connection is initiated on port number 20.
- FTP sends the control information out-of-band as it uses a separate control connection.
- Some protocols send their request and response header lines and the data in the same TCP connection.
- For this reason, they are said to send their control information in-band. HTTP and SMTP are such examples.

Free and Open Source Software

- Free and Open Source Software (FOSS) refers to software that is freely available for anyone to use, modify, and distribute.
- This software is built on principles of collaboration, transparency, and freedom, and it plays a crucial role in the development of the technology ecosystem.

1. Freedom:

Freedom to Use: Users can run the software for any purpose.

Freedom to Study: Users can examine how the software works and modify it to suit their needs.

Freedom to Distribute: Users can share the software with others.

Freedom to Improve: Users can improve the software and release their improvements to the public.

Free and Open Source Software

2. Open Source:

Source Code Availability: The source code, the human-readable form of the software, is made available to anyone. This contrasts with proprietary software, where the source code is typically kept secret.

Collaborative Development: Developers from around the world can contribute to the software, enhancing its features, fixing bugs, and improving security.

Advantages of FOSS:
1. Cost-Effective:
FOSS is generally free to use, reducing costs for individuals, businesses, and educational institutions.

2.Security and Transparency:
With the source code available for review, security vulnerabilities can be identified and fixed more quickly.
Users can verify that the software does not contain malicious code.

Free and Open Source Software

Community Support:

FOSS projects often have vibrant communities that provide support, documentation, and user forums.
Flexibility and Customization:

Users can modify the software to meet their specific needs, allowing for greater customization compared to proprietary software.
No Vendor Lock-In:

Users are not dependent on a single vendor for updates or support, giving them more control over their software environment.

Free and Open Source Software

Examples of FOSS:
Operating Systems: Linux (e.g., Ubuntu, Fedora), FreeBSD
Web Browsers: Mozilla Firefox, Chromium
Office Suites: LibreOffice, Apache OpenOffice
Programming Languages: Python, Ruby, PHP
Content Management Systems: WordPress, Joomla, Drupal

Licenses:
FOSS comes with licenses that explain what you can do with the software. Some common ones are:

GNU General Public License (GPL): Keeps the software free for everyone.

MIT License: Very open and allows almost anything.

Apache License: Similar to the MIT License and includes patent rights.

Free Software

Characteristics:

Freedom: Users have the freedom to run, modify, and distribute the software.
Source Code: Source code is available to the users.
Licensing: Usually licensed under the GNU General Public License (GPL) or similar licenses.

Advantages:

User Freedom: Users can use the software for any purpose, study how it works, modify it, and share it.
Community Support: Large communities often support free software, offering extensive resources and help.
Transparency: Users can inspect the code to ensure there are no malicious elements.
Cost: Generally free of cost.

Disadvantages:
Support: Professional support may be lacking or inconsistent, relying on community support.
Usability: Might not be as user-friendly as commercial alternatives.
Compatibility: Can have compatibility issues with proprietary formats or systems.
Sustainability: Funding and sustained development can be uncertain.

Open Source Software Characteristics:

Source Code: Source code is available for anyone to view, modify, and distribute.
Collaboration: Encourages collaborative development.
Licensing: Licensed under open-source licenses like the MIT License, Apache License, or GPL.

Advantages:
Flexibility: Users can modify the software to fit their needs.
Security: Transparency allows for quicker identification and fixing of security vulnerabilities.
Community Collaboration: Diverse contributions can lead to robust and innovative solutions.
Cost: Often available at no cost or at a lower cost than proprietary software.

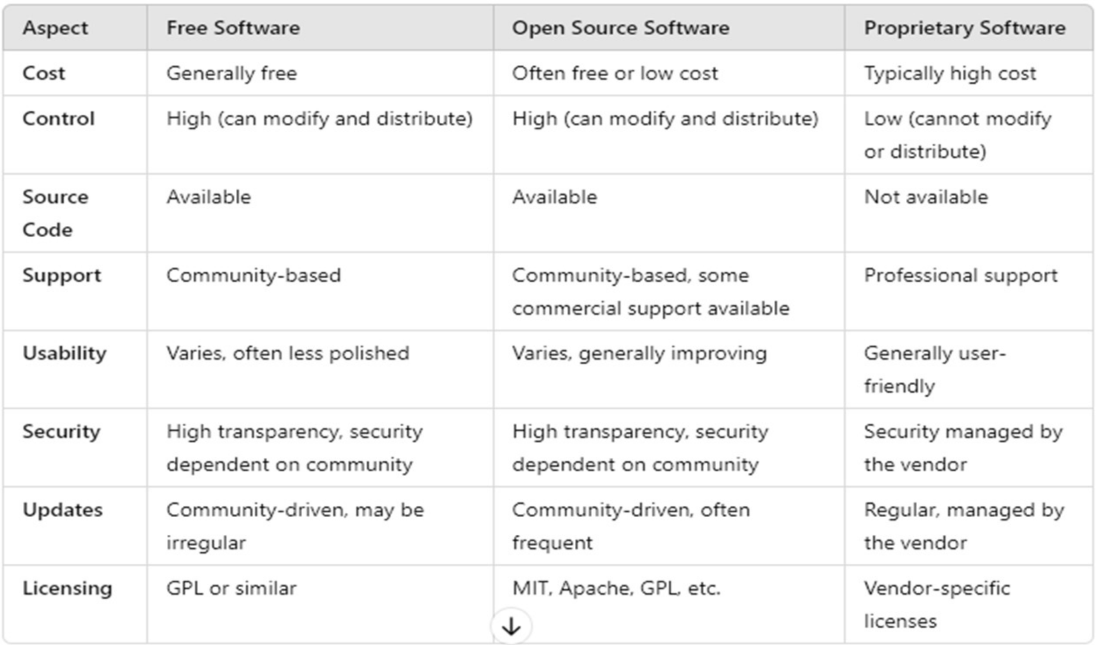
Disadvantages:
Support: Professional support might be limited compared to commercial software.
Documentation: Documentation might not be as thorough or professionally produced.
Quality Control: Varying quality of contributions can lead to inconsistent software quality.
Adoption: May face resistance in adoption, especially in enterprises accustomed to proprietary solutions.

Proprietary Software
Characteristics:

Restrictions: Usage, modification, and distribution are restricted by the license agreement.
Source Code: Source code is not available to the users.
Control: Developed and maintained by a single company or entity.

Advantages:
Professional Support: Usually comes with dedicated, professional support services.
User-Friendly: Often designed with a focus on user experience, making it more intuitive.
Integration: Generally well-integrated with other proprietary systems and software.
Reliability: Often thoroughly tested and maintained, ensuring a higher level of reliability.

Disadvantages:
Cost: Can be expensive, both for initial purchase and ongoing maintenance or subscription fees.
Lack of Control: Users cannot modify the software to suit their specific needs.
Dependency: Creates dependency on the vendor for updates, support, and continued use.
Transparency: Users cannot inspect the code, which could hide malicious code or privacy issues.



Aspect	Free Software	Open Source Software	Proprietary Software
Cost	Generally free	Often free or low cost	Typically high cost
Control	High (can modify and distribute)	High (can modify and distribute)	Low (cannot modify or distribute)
Source Code	Available	Available	Not available
Support	Community-based	Community-based, some commercial support available	Professional support
Usability	Varies, often less polished	Varies, generally improving	Generally user-friendly
Security	High transparency, security dependent on community	High transparency, security dependent on community	Security managed by the vendor
Updates	Community-driven, may be irregular	Community-driven, often frequent	Regular, managed by the vendor
Licensing	GPL or similar	MIT, Apache, GPL, etc.	Vendor-specific licenses

Licensing is a critical aspect of software distribution and usage, defining the terms under which software can be used, modified, and distributed. Software licenses broadly fall into two categories: commercial licenses and open-source licenses.

Commercial License

Types of Commercial Licenses:
Perpetual License: The user pays a one-time fee for indefinite use of the software.
Subscription License: The user pays a recurring fee (monthly or yearly) to use the software.
Site License: Allows an organization to use the software on multiple computers within a specific location.
User License: Licenses the software for a specific number of users.
Concurrent User License: Limits the number of users who can use the software simultaneously.

Examples of Commercial Software Licenses:

- **Microsoft Office License:** Provides usage rights for Microsoft Office software, often with varying levels of access and features based on the type of license purchased.
- **Adobe Creative Cloud Subscription:** Allows users to access Adobe's suite of software through a subscription model.

Open Source License

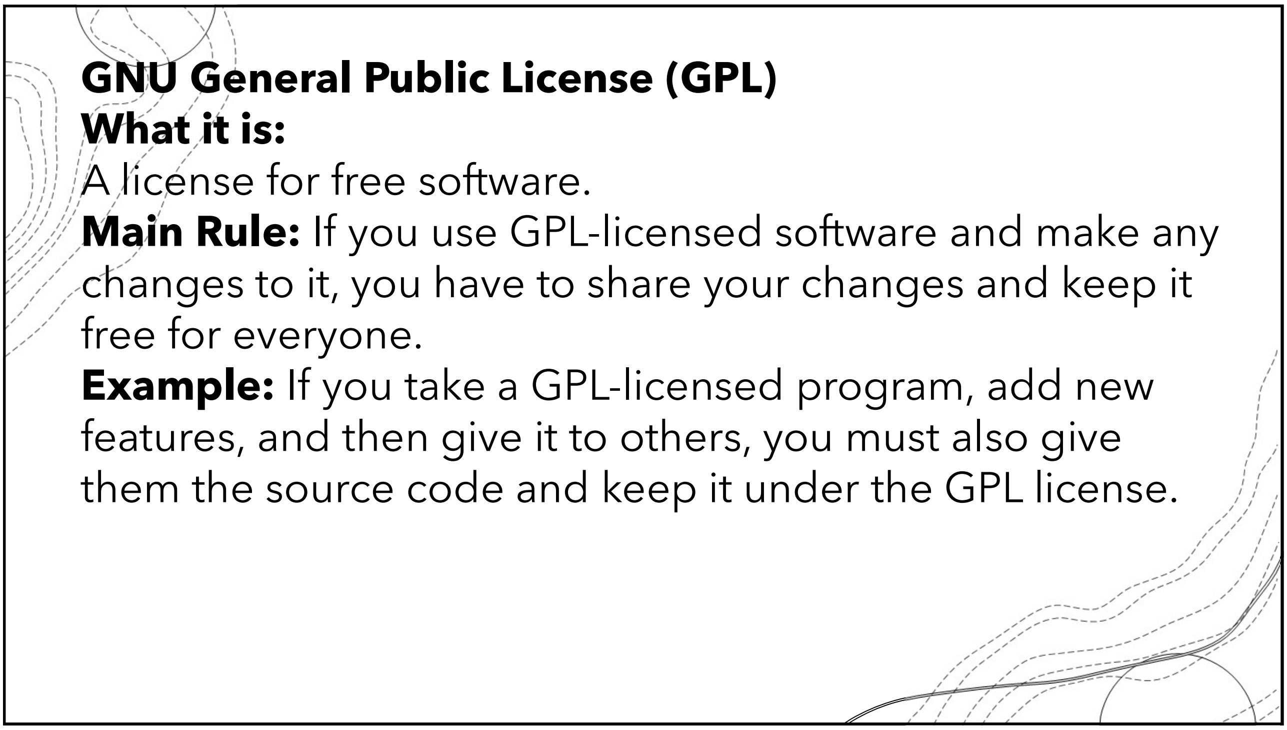
Types of Open Source Licenses:

1.Permissive Licenses: Allow users to do almost anything with the software, including using it in proprietary software. Examples include:

- **MIT License:** A very permissive license with minimal requirements.
- **Apache License 2.0:** Permissive but includes a patent grant and provides protection from patent litigation.

2.Copyleft Licenses: Require that any distributed modified versions of the software also be open-source and distributed under the same license. Examples include:

- **GNU General Public License (GPL):** Ensures that the software and its derivatives remain free and open.
- **GNU Lesser General Public License (LGPL):** Similar to the GPL but allows linking with proprietary code under certain conditions.



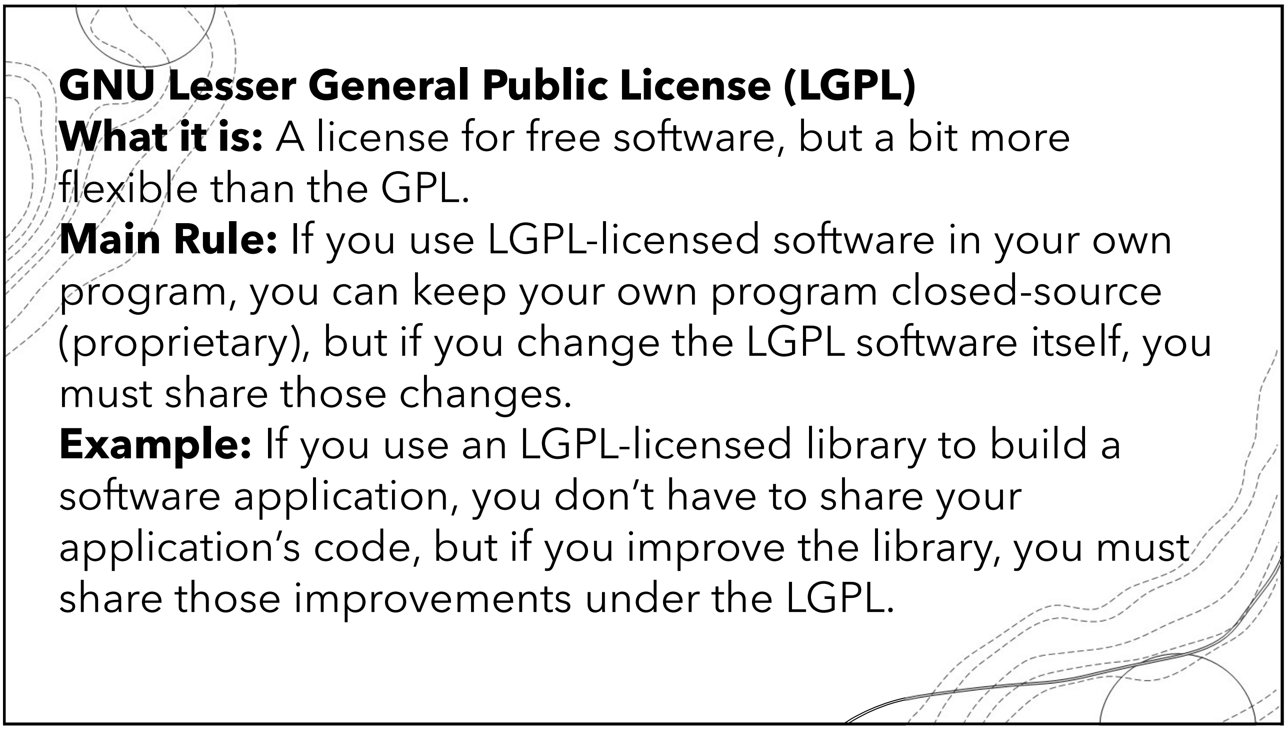
GNU General Public License (GPL)

What it is:

A license for free software.

Main Rule: If you use GPL-licensed software and make any changes to it, you have to share your changes and keep it free for everyone.

Example: If you take a GPL-licensed program, add new features, and then give it to others, you must also give them the source code and keep it under the GPL license.

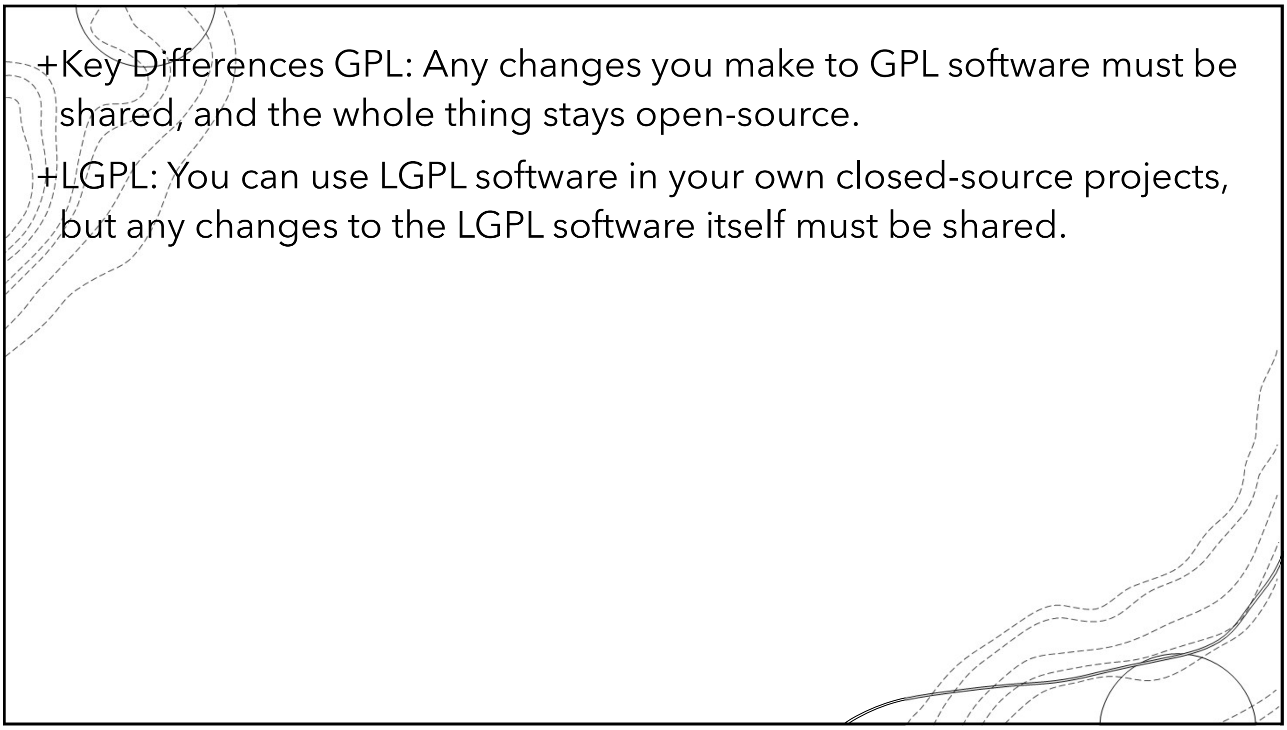


GNU Lesser General Public License (LGPL)

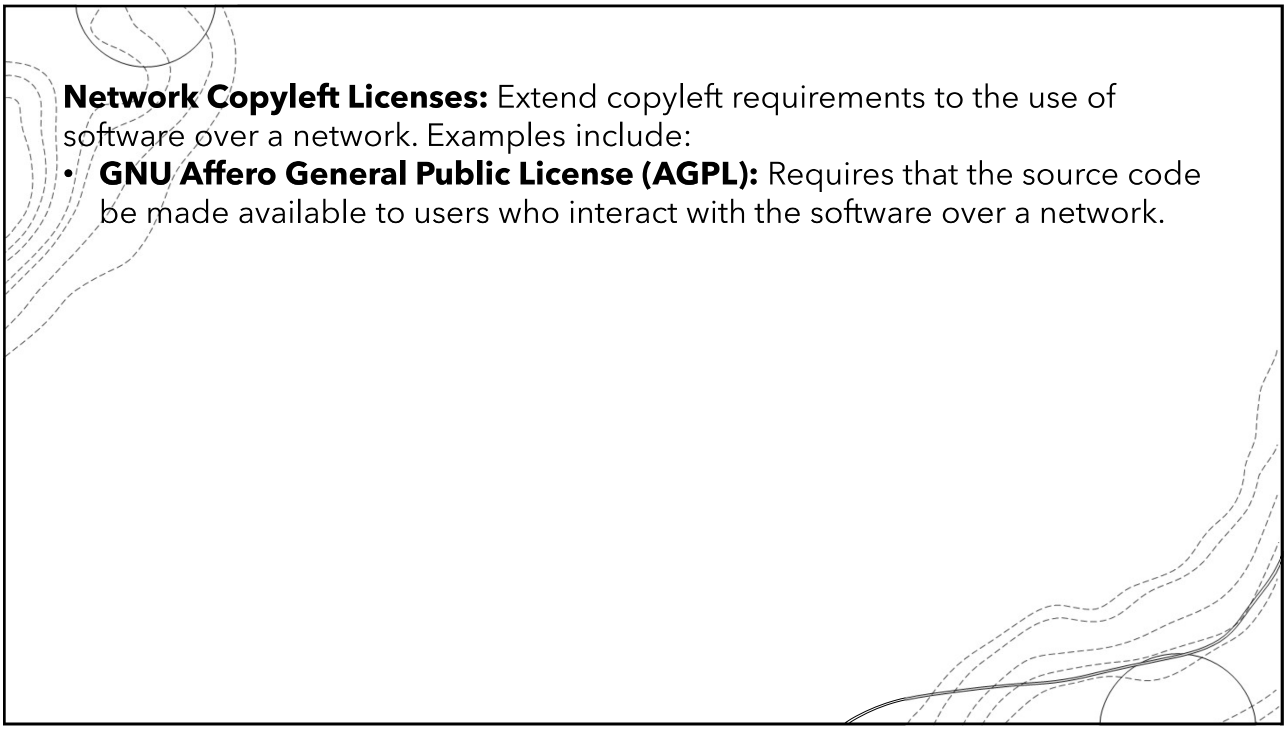
What it is: A license for free software, but a bit more flexible than the GPL.

Main Rule: If you use LGPL-licensed software in your own program, you can keep your own program closed-source (proprietary), but if you change the LGPL software itself, you must share those changes.

Example: If you use an LGPL-licensed library to build a software application, you don’t have to share your application’s code, but if you improve the library, you must share those improvements under the LGPL.



- +Key Differences GPL: Any changes you make to GPL software must be shared, and the whole thing stays open-source.
- +LGPL: You can use LGPL software in your own closed-source projects, but any changes to the LGPL software itself must be shared.



Network Copyleft Licenses: Extend copyleft requirements to the use of software over a network. Examples include:

- **GNU Affero General Public License (AGPL):** Requires that the source code be made available to users who interact with the software over a network.