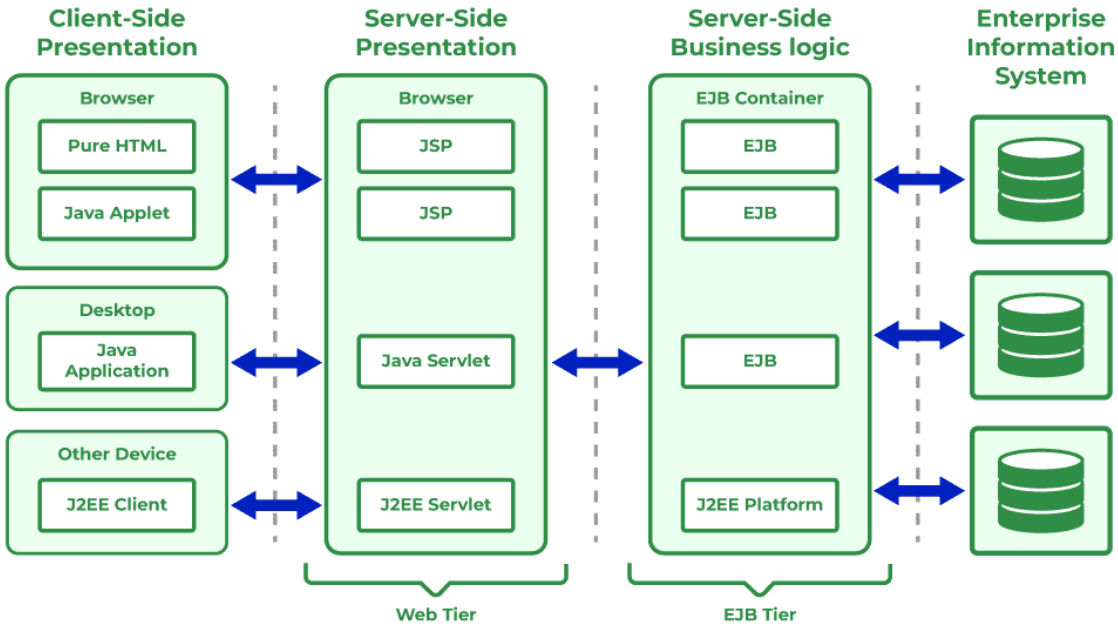
J2EE Architecture:

- J2EE stands for Java 2 Platform, Enterprise Edition.
- An architecture is a blueprint for an application.
- It contains collection of components/models,layers,protocols.
- J2EE is a set of **multi-tiered, network-based applications, such as those for web and business computing.**

J2EE is a four-tier architecture that consists of :

1. Client Tier/Presentation Tier
2. Web Tier(Presentation Layer)
3. Enterprise JavaBeans Tier/Business Tier
4. Server Tier

Diagram:



1. Client Tier/Presentation :

This is a frontend layer where users **interact with the application.** It can be a web browser, mobile app, or desktop application.

→ It uses technologies like HTML, CSS, JavaScript, etc.

* Standalone Java applications provide a dynamic interface to the middle tier

Responsibilities:

- User Interaction
- Sending Requests to Server
- Receiving and Displaying Responses
- Client-side validation
- Supports multiple platforms (web, mobile, desktop)

2. Web Tier(Presentation Layer):

It provides the internet functionality to the J2EE application.

The components on the web tier use HTTP to receive requests and send responses to the client that resides on the tier.

Responsibilities:

- Accepting requests from clients.
- Generating responses using JSP/Servlets.
- Passing business-related requests to the business tier.

Components of Web Tier: Two components that work on the web tier are mentioned below:

1. Servlets
2. Java Server Pages (JSP)

3. Enterprise JavaBeans Tier/Business Tier :

It contains business logic for J2EE applications. In this tier two or more EJB reside.

It is a core component for every J2EE Application.

It uses EJB to handle tasks like calculations, transactions and security, processes data and communicates with the database layer.

EJB tier provides key features such as concurrency, scalability, life cycle management, and fault-tolerant (Back-Up) to the J2EE applications.

Responsibilities:

- Contains business logic for J2EE applications
- Handles tasks like calculations, transactions, and security
- Processes data and communicates with the database layer
- Provides concurrency and scalability
- Manages application life cycle

4. Server Tier :

It provides the runtime environment for web components (Servlets, JSP) and business components (EJBs).

It acts as the backbone of J2EE applications by handling requests from clients and communicating with the database (EIS tier).

Data layer stores and retrieves data from a database.

Data layer ensures secure and efficient data storage.

Responsibilities:

- Hosts and manages application logic and business logic
- Runs the J2EE application server
- Provides runtime environment for Servlets, JSPs, and EJBs
- Handles client requests

J2EE Containers

Containers: It acts as Interface between components and application server services.

- It provides the run time environment.
- Containers are used to store components.
- Containers always provide security for components.

Components acts as Interface between applications and users.

Types of containers:

1. Application container
2. Applet container
3. Web container
4. Business logic container

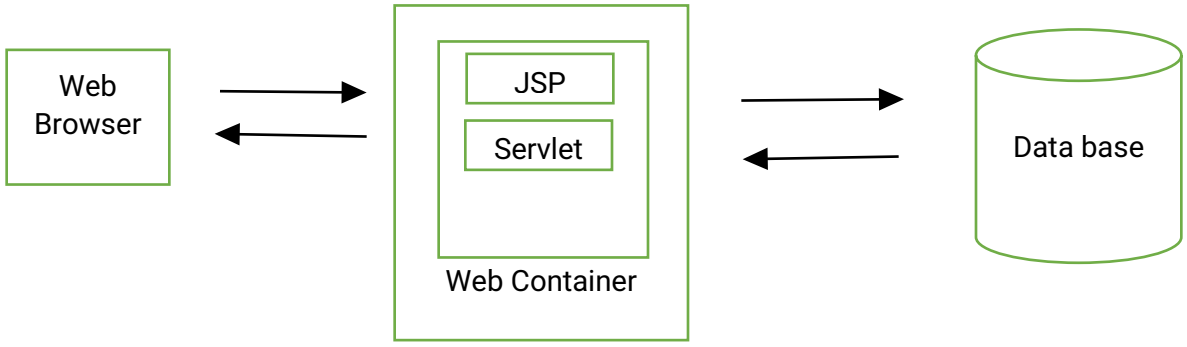
Web container: A web container or servlet container, host web applications which are composed of servlets and Jsp's.

- Web container contains JSP and servlets.
- The main functionality of web container is to process http requests and generate http response.

  Eg: AppacheTomcat, standalone web container.
- It provides the run time environment for execution of a program.
- The Web container provides servlets and APIs for JSP.
- It is a server side container.

Advantages:

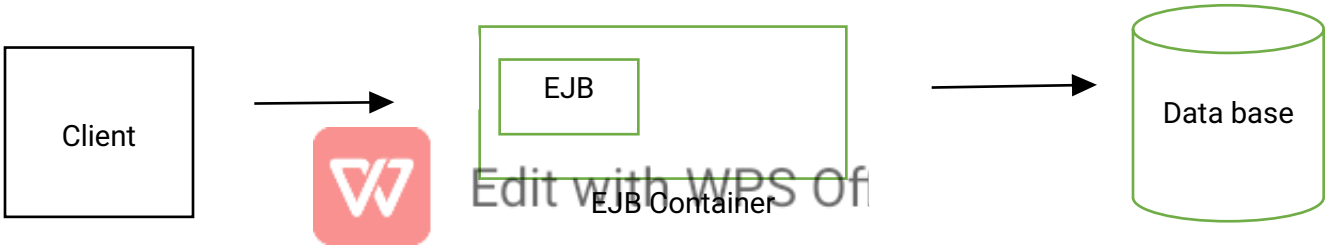
- To handle http request and response.
- Security.
- Managing data.



Web Server

Business logic container: A Business logic container hosts server side components that contain an EJB(Enterprise Java beads).

- It provides local and remote access to EJB.
- It manages the life cycle, state, transactions and security for EJB.
- An EJB container is a server platform used for controlling the execution of Enterprise Bean, and provides various services for Enterprise Bean. The EJB container provides APIs for EJB.
- It is a server side container.

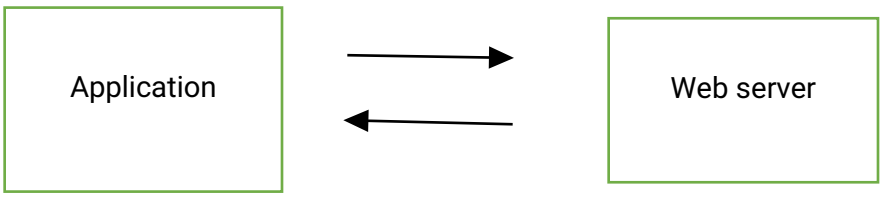


Application client

Container Web server

Application client container: This container runs on the client machine and hosts rich, standalone graphical user interface applications.

- It contains standard alone components.
- It provides access to other services on the server, such as EJB.
- Manages the execution of application client components. Application clients and their container run on the client.
- It is a client side container



Client

Applet container: The applet container runs applet within a client's web browser, using a java plug-in.

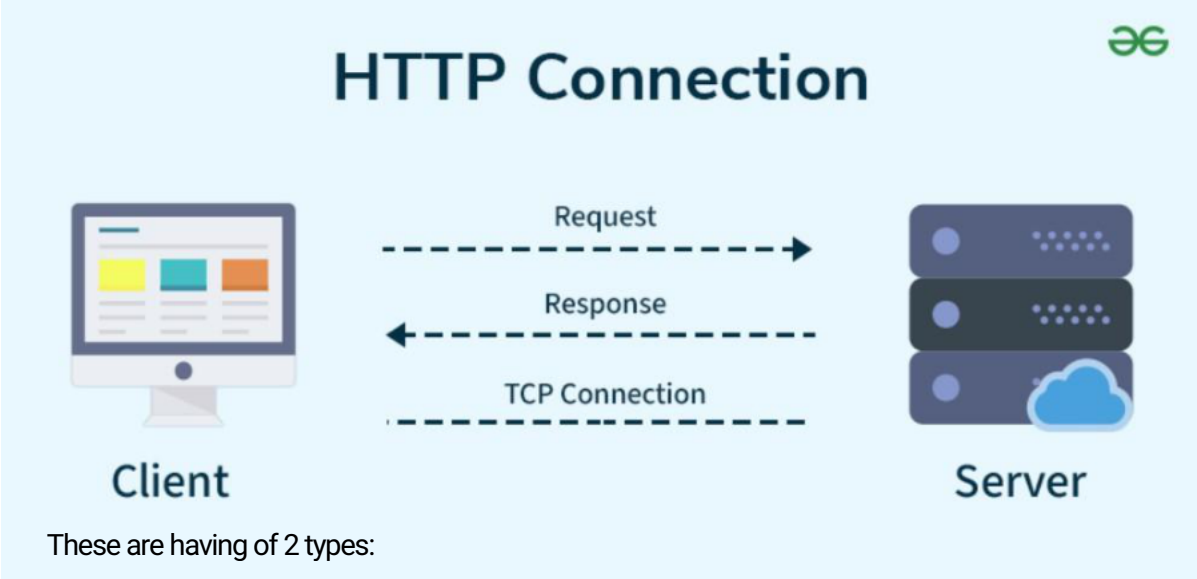
- It manages the applet's life cycle and execution.
- Contains HTML (Hypertext markup language) paper and small java programs.

- Contains embedded components program.
- It is a client side container.

HTTP PROTOCAL

HTTP Full Form – Hypertext Transfer Protocol

HTTP stands for Hypertext Transfer Protocol, and it's the system that allows communication between web browsers (like Google Chrome or Firefox) and websites. When you visit a website, your browser uses HTTP to send a request to the server hosting that site, and the server sends back the
data needed to display the page.



These are having of 2 types:

1.HTTP

2.HTTPS

1.HTTP: To handle HTTP request and HTTP response.HTTP provides the set of rules, or the language, that allows your browser and web servers to communicate and exchange data, such as text, images, and videos, ensuring websites load correctly.

An HTTP request is made from a client to a host located on the server in order to receive a resource needed to build the content.

An HTTP response is made by a server to a client. The aim of the response is to provide the client with the resource it requested, or inform the client that the action it requested has been carried out; or else to inform the client that an error occurred in processing its request.
Advantages of HTTP:

Simplicity and a lightweight design

Making it easy for developers and efficient for data transfer, along with support for a wide range of content, from text to media files

Disadvantages of HTTP:

- lack of security
- it does not encrypt data and exposes sensitive information
- Performance Issues

2.HTTPS: A secure version of HTTP, known as HTTPS (Hypertext Transfer Protocol Secure), encrypts
the data exchanged between your browser and the server, providing protection against eavesdropping and man-in-the-middle attacks.

HTTPS are of 2 types

- SSL(Secure socket layer)
- TLS(Transport layer security)

ADVANTAGES:

Data Encryption (Confidentiality):

- Data Encryption (Confidentiality): HTTPS encrypts traffic between the JDBC client and the database/web service endpoint.
- With HTTPS, data is encrypted in transit in both directions.
- Authentication & Trust: HTTPS uses SSL/TLS certificates to verify the server's identity.

HTTP Methods:

HTTP protocol methods also known as http verbs or request methods.

Define the action to be performed on a resourse.

The HTTP methods are:

1.get()

2.post()

3.put()

4.patch()

5.delete()

Methods	Description
get()	Used to retrieve data from server
post()	Used to send data to server
put()	It replaces the entire data at server side
patch()	It updates the data at server side
delete()	Delete data from server

Status Codes

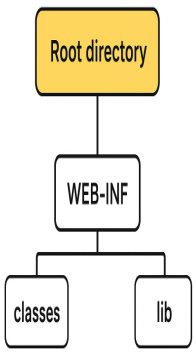
HTTP Protocol is used everywhere from the server page to each service communication, deploying service to monitoring service.

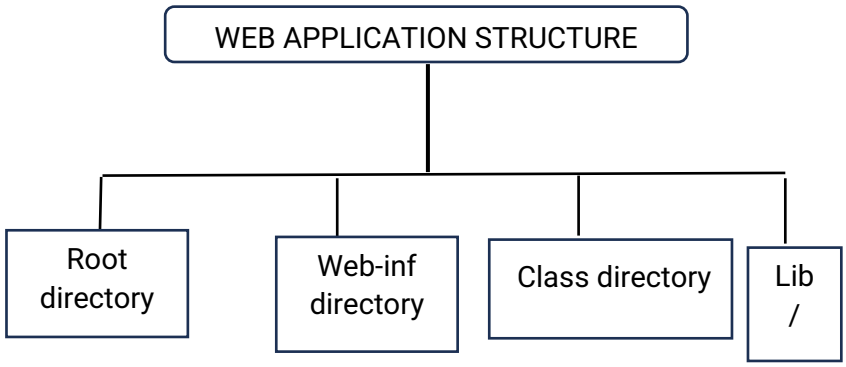
100 - continue data
200 - ok, successful
300 – Redirect
400 - client side error
404 - server not found
500 - server side error

Web application structure :-

* This is the blueprint of how a web application's components are organized and interact with each other.
* It also known as web application architecture.
* It defines the relationships between the user interface, databases, and business logic to ensure a seamless flow of data and efficient functionality.
* A WAR (Web Application Archive) file is a standard packaging format for Java web applications, containing all the necessary files for deployment on a webserver (or) application server.
* WAR files are supported by all javaEE Containers.

Diagram:-





* There are four types of directorys.
  1.Root Directory
  2.Web info-Directory
  3.Class Directory
  4.Lib Directory

1.Root Directory/Meta directory:

*This is the top-level directory of the web application within the WAR file.
*Here ,the data is Public access.
*It contains public-facing web resources like HTML files, JSP files, images, CSS, and JavaScript.
*It contains 2 types of resources.
1.static resource.
2.Dynamic resource.

1.Static resource:

*Static resources are files that are served to the client exactly as they are stored on the server, without any modification.

- Examples like:
  - HTML file(.html) o Cascading Style Sheets file(.css)
  - JavaScript files (.js)
  - Images, videos, and other media files (.png) o Other static files like plain text or PDF documents

2.Dynamic resource:

*Dynamic resources are content that is generated server in response to a client's request.

- Examples like:

*.jsp (JSP page)

*.jsf(JSF face) Diagram:

```
WEBAPP DIRECTORY (or) ROOT DIRECTORY--->(Public Access)
  ├── Static Resources
  │ ├── .html, .css
  │ ├── .js (Javascript)
  │ ├── .png, .jpg (Image)
  │ ├── .mp3 (Audio)
  │ └── .mp4(Video) │
  └── Dynamic Resources
     ├── .jsp (JSP page)
     └── .jsf (JSF Face)
```

2.WEB-INF Directory:

*It always under the root directory.

*A special, protected directory not directly accessible by web clients.

*Because it is private access.

*It is server side component and server configuration file.

* It contains :

configuration file,
filter file,
mapping file,
url file.

*Advantage:

*private access.

*security. Diagram:

```
WEB-INF DIRECTORY-->(Private Access ,Security)
  ├── web.xml (Configuration file)
  ├── filter file
  ├── mapping file
  └── url file
```

3.WEB-INF/classes/ Directory:

*Located within WEB-INF/, this directory contains all the compiled Java classes for your web application.

*Such as Servlets, JavaBeans, helps class and other utility classes.

Diagram:

```
├── classes directory
   └── .java (Compiled Java classes)
       .html
```

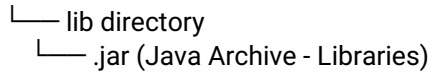
4.WEB-INF/lib/ Directory:

*Also located within WEB-INF/.

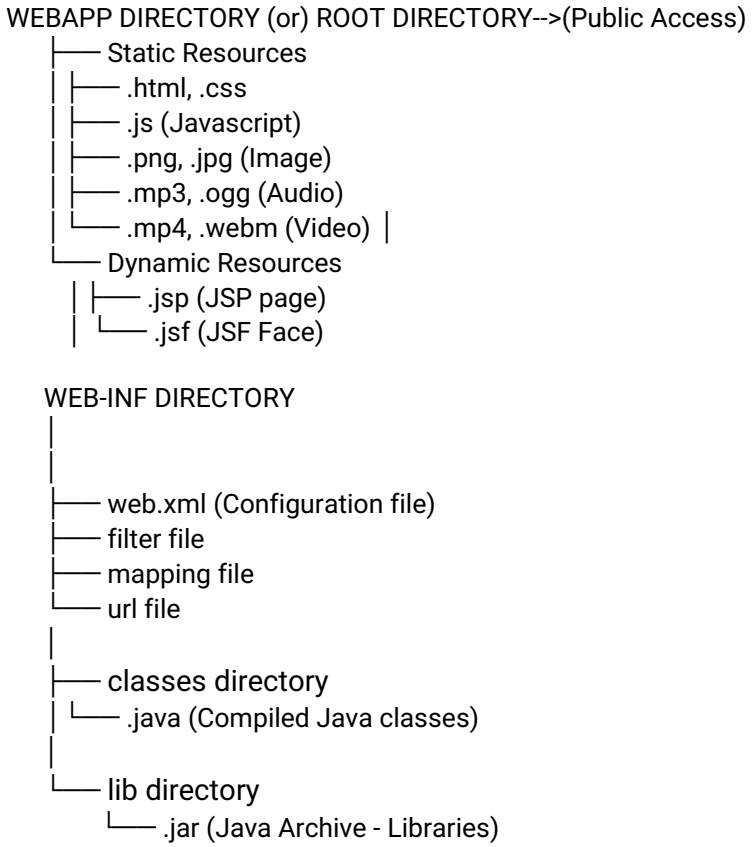
*This directory holds all the supporting JAR files (Java Archive files) .

*That your web application depends on, such as third-party libraries or internal utility JARs.

Diagram:



Diagramatic representation of Web application Structure:

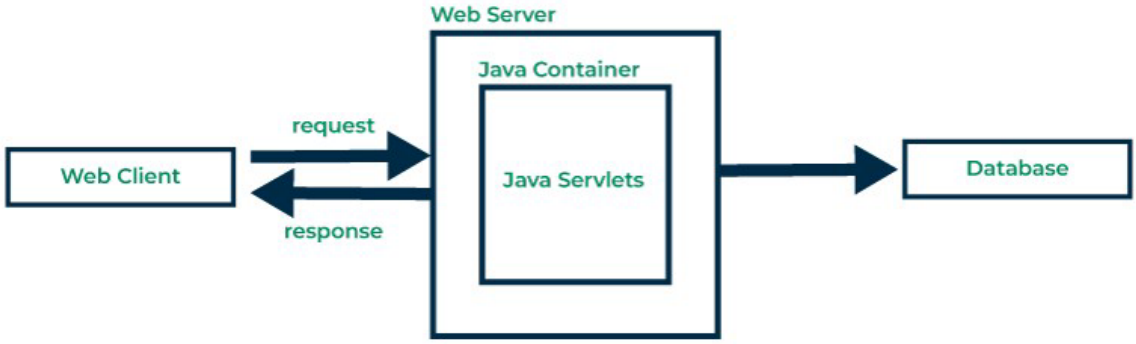


WEB CONTAINERS AND ARCHITECTURE MODELS:

WEB CONTAINER : Web container is a web server component that interacts with the Java Servlets. This processes the requests to the requested servlet while checking the required access rights of the URL.

* Architecture focuses on decisions that impact the entire system's scalability, performance, security, and maintainability.

Diagram:



Component Role:

Advantage:

1.Loads, initializes, and manages Servlets/JSPs.

2.Handles HTTP requests and sends HTTP responses

3.Provides security, session management, life cycle management, etc.

Example Web Containers:

1.Apache Tomcat

2.Jetty

3.GlassFish

J2ee architectures models:

*It is the blue print for web application.

*Blue print is required for better understanding

*There are some Architecture

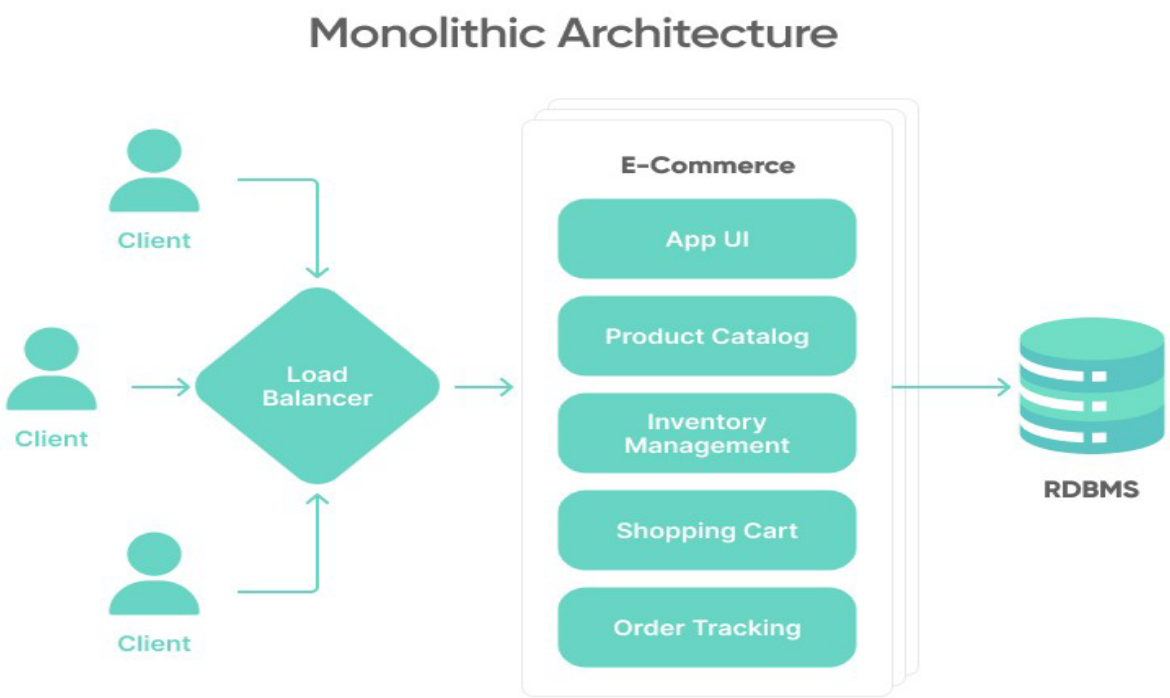
- Monolithic Architecture
- Micro Services Architecture
- Serverless Architecture
- Single-Page Application (SPA) Architecture

1.Monolithic Architecture:

*A monolithic architecture is a traditional software design approach where an entire application is built as a single, unified, and self-contained unit.

*It as more number of clients and only one database.

*All components, including the user interface, business logic, and data access layer, are tightly coupled within a single codebase and deployed together as one package.



Advantages:

- Single codebase makes development and maintenance easier.
- Low complexity due to fewer moving parts.
- Easier to debug .
- Low cost.

Disadvantages:

- Difficult to change individual components.
- Not well-suited frequent changes.
- It difficult to reuse code.

- Diffcult to handle for access data.

2.Microservices Architecture.

*Microservices architecture is a modern, web application architecture that breaks down a large, complex application into smaller, independent services.

*These services communicate with each other using APIs and are independently deployable.

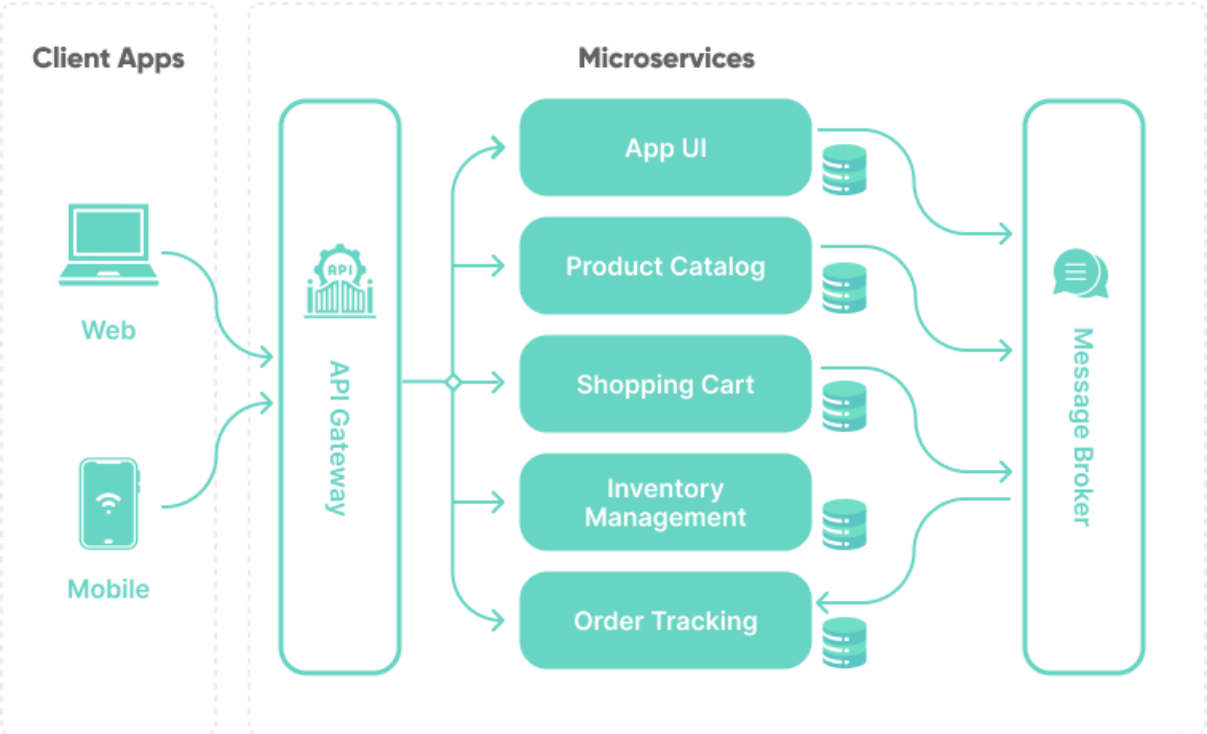
*In a microservices web application architecture, each service is designed to perform a specific task.

Disadvantages:

*high cost

*complex architecture

*easy access

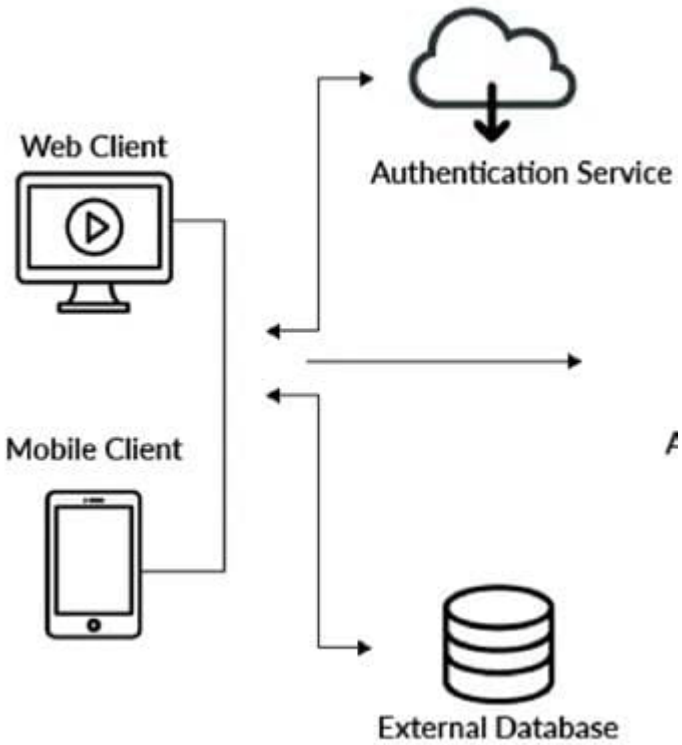


3.Serverless Architecture

*Serverless web application architecture leverages cloud-based services to deploy a web app without having to manage the underlying infrastructure.

*Here, third party is present that is cloud.

*that require high levels of scalability, such as e-commerce, gaming, and streaming applications.



Single-Page Application (SPA) Architecture.

4.Single-Page Application (SPA) Architecture:

*A single-page application (SPA) is a web application that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from the server.

*this perfroms on either client side or server side.

*server role is just connectivity passing and performs minimum access.

* The page is updated in real-time and is designed to provide a smoother, more responsive user experience.