GANPAT UNIVERSITY
FOET-UVPCE

2CEIT6PE1: WEB TECHNOLOGY

UNIT 1

INTRODUCTION OF WEB TECHNOLOGY

Unit 1: INTRODUCTION Of WEB TECHNOLOGY

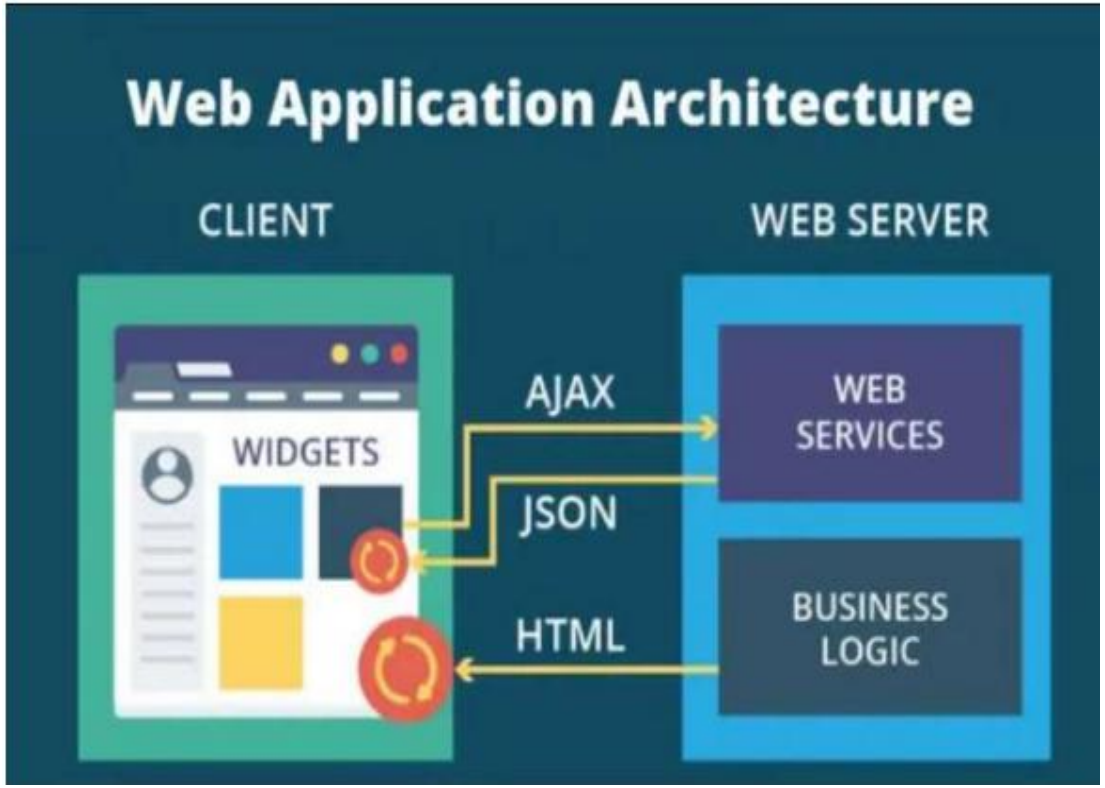
Contents:

- Overview
- Web Application Architecture Components
- Layers of Modern Web Application Architecture
- Web Application Architecture Diagram
- Web service and API
- Types of Web Application Architecture

Web Application Architecture

Web Application Architecture:

The web application architecture describes the interactions between high level of components on web like applications, databases and middleware systems.



Example:

User hits button after typing URL in the address bar of a web browser, it requests for that particular web address.

The server sends files to the browser as a response to the request made.

The browser then executes those files to show the requested page.

Web Application Architecture Components

Web application architecture comprises 3 core components:

Web Browser: The browser or the client-side component or the front-end component is the key component that interacts with the user, receives the input and manages the presentation logic while controlling user interactions with the application. User inputs are validated as well, if required.

Web Server: The web server also known as the backend component or the server-side component handles the business logic and processes the user requests by routing the requests to the right component and managing the entire application operations. It can run and oversee requests from a wide variety of clients.

Database Server: The database server provides the required data for the application. It handles data-related tasks. In a multi-tiered architecture, database servers can manage business logic with the help of stored procedures.

.

Web Application database Architecture

A Database store a lot of critical information to access data quickly and securely.

Types of DBMS Architecture:

One- Tier Architecture

Two- Tier Architecture

Three- Tier Architecture

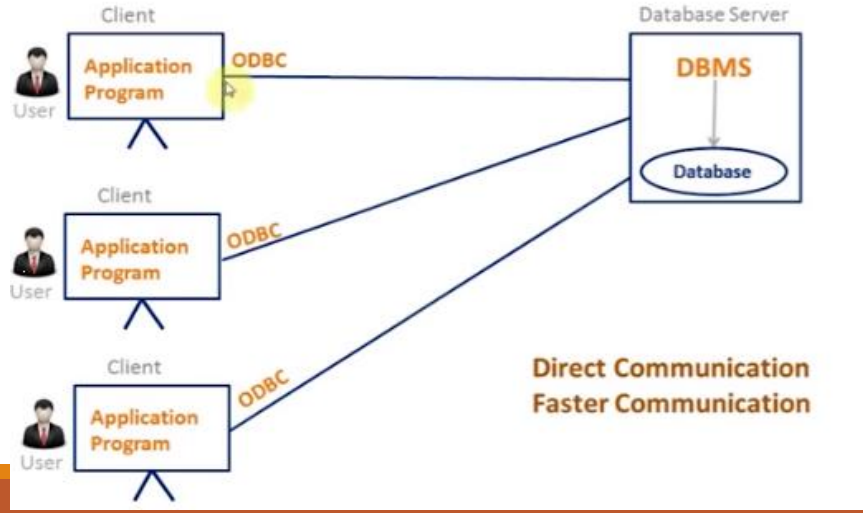
.

Web Application database Architecture

One-Tier Architecture: In One-Tier Architecture the database is directly available to the user, the user can directly use it. The client, server, and the Database are all present on the same machine.

Two-Tier Architecture: The two-tier architecture is similar to a basic **client-server** model. The application at the client end directly communicates with the database at the server side. APIs like ODBC and JDBC are used for this interaction. The server side is responsible for providing query processing and transaction management functionalities. On the client side, the user interfaces and application programs are run. The application on the client side establishes a connection with the server side in order to communicate with the DBMS.

EG. Desktop based Applications like railway ticket booking



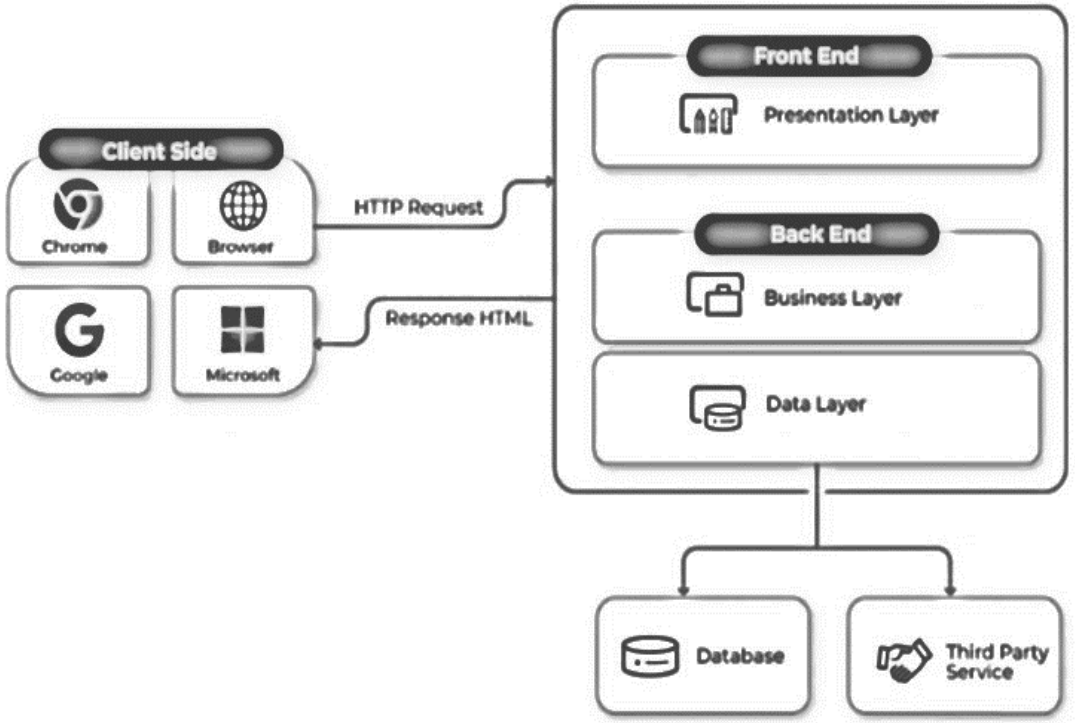
Web Application database Architecture

Three-Tier Architecture:

There are three layers of a 3-Tier architecture:

- Presentation layer / Client Layer
- Application Layer / Business Layer
- Data Layer

.



Standard web Application Architecture

Web Application database Architecture

Three-Tier Architecture: There is another layer between the client and the server. The client does not directly communicate with the server. Instead, it interacts with an application server which further communicates with the database system and then the query processing and transaction management takes place. This intermediate layer acts as a medium for the exchange of partially processed data between server and client.

- The 3-tier web application architecture diagram is more secure as the client does not directly access the data.
- Deploying application servers on multiple machines provides higher scalability, better performance and better re-use.
- User can abstract the core business from the database server to perform load balancing efficiently.
- Data integrity is improved as all data goes through the application server, which decides how data should be accessed and by whom.

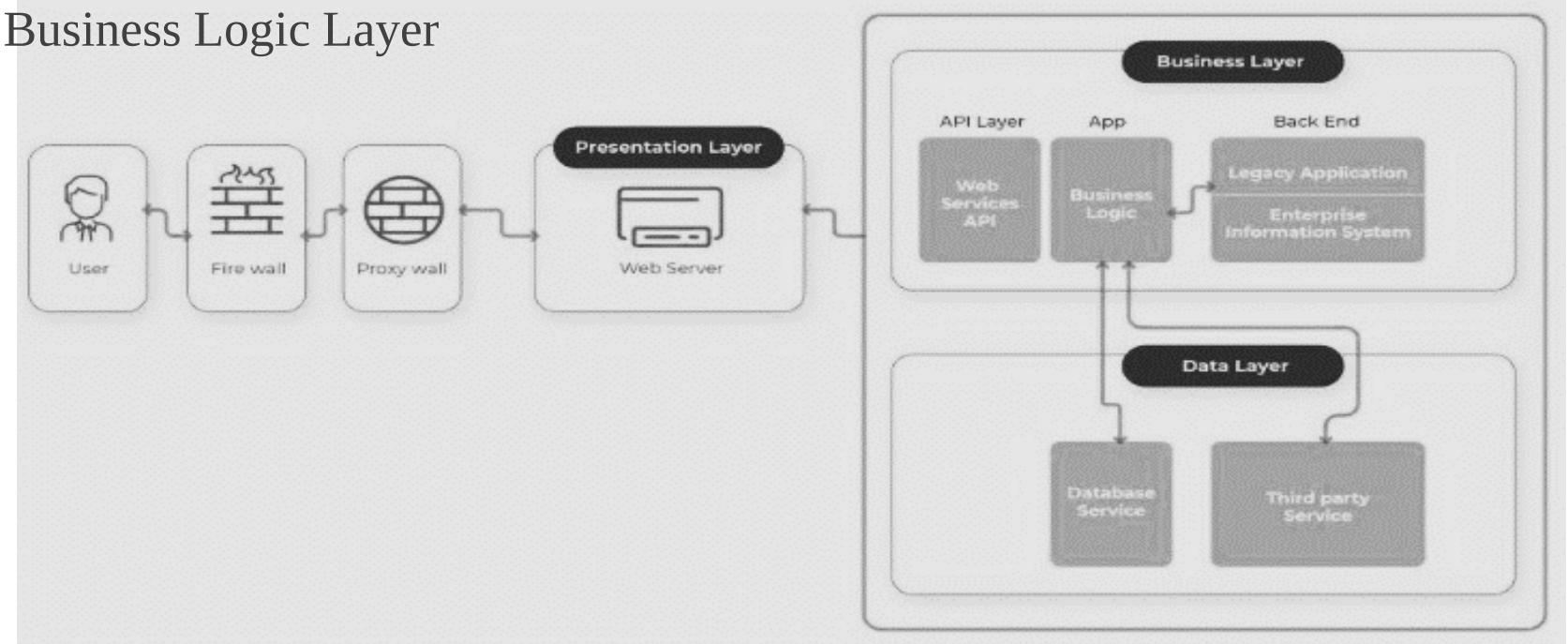
This type of architecture is used in the case of large web applications.

.

Web Application Architecture

The three layers of a web application architecture diagram:

1. Presentation layer / Client Layer
2. Application Layer / Business Logic Layer
3. Data Layer



Modern Web Application Architecture

Web Application Architecture

Presentation Layer: Client side components

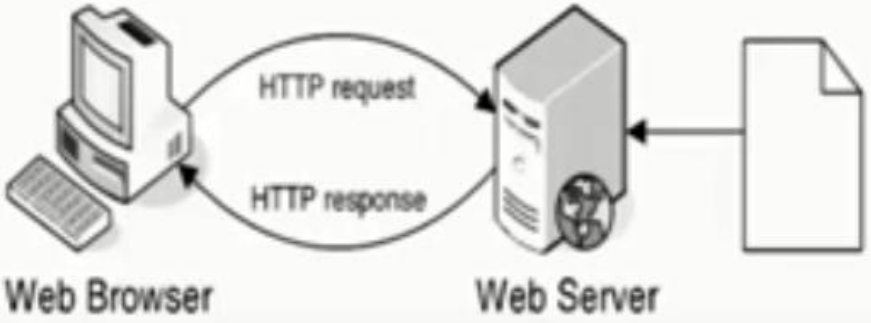
The client-side component of a web application architecture diagram enables users to interact with the server and the backend service via a browser.

The code resides in the browser, receives requests, and presents the user with the required information.

This is where UI/UX design, dashboards, notifications, configurational settings, layout, and interactive elements come into the picture.

Web Application Architecture

How static Web Pages are processed?



simple HTTP request

```
GET / HTTP/1.1
Host: www.example.com
```

simple HTTP response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
    <title>Example Web Page</title>
</head>
<body>
    <p>This is a sample web page</p>
</body>
</html>
```

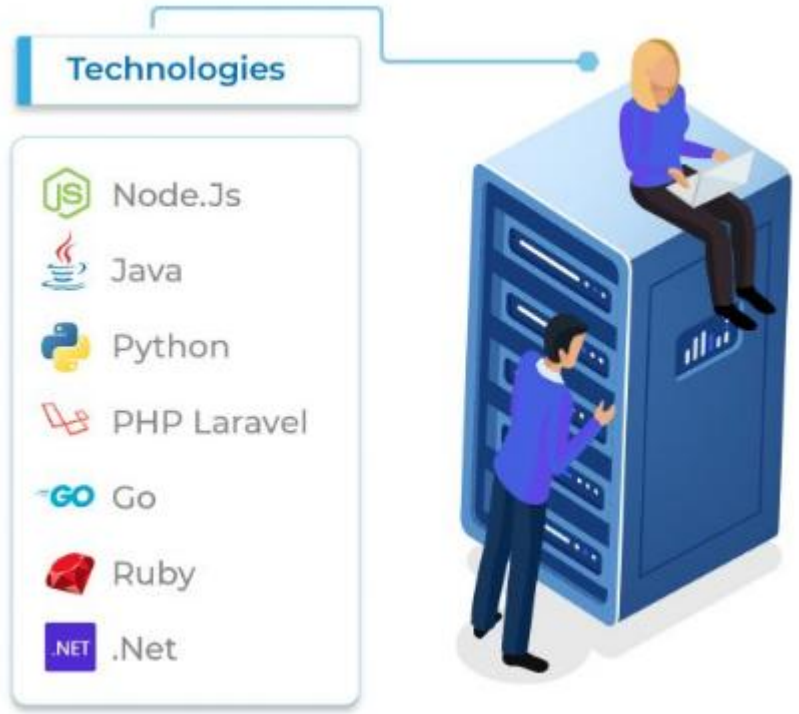
Web Application Architecture

Application Layer: Server side components

The server-side component is the key component of the web application architecture diagram that receives user requests, performs business logic, and delivers the required data to the front-end systems. It contains servers, databases, web services, etc.

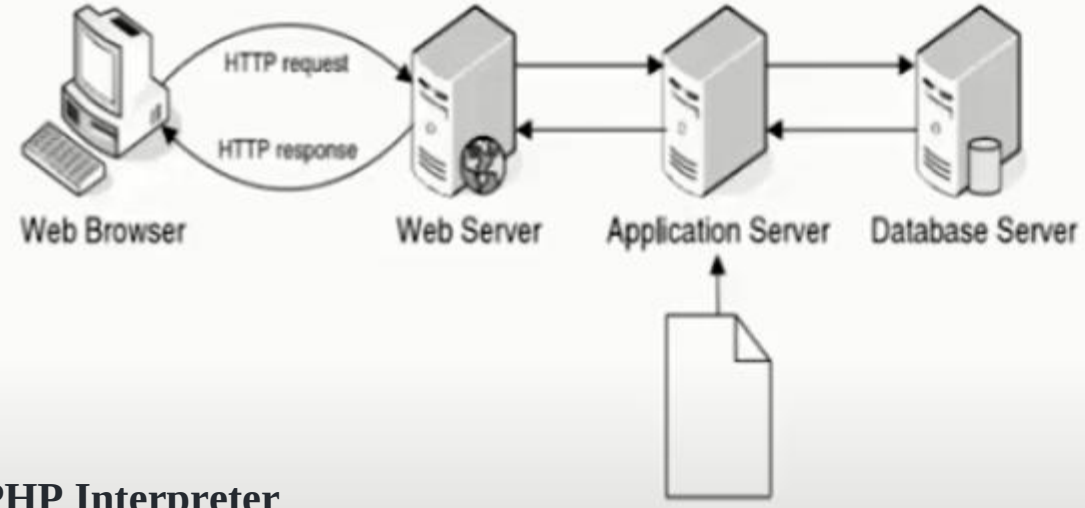
Examples of Application Server:

- **Apache**
- **NGINX**



Web Application Architecture

How dynamic Web Pages are processed?



PHP Interpreter

Parsing: Interpreter first parses the script to break it down into smaller components, such as tokens and syntax trees.

Compilation: Parsed code is compiled into opcodes (low-level instructions)

Execution: Compiled opcodes are executed by the PHP engine.

simple HTTP request

```
GET / HTTP/1.1
Host: www.example.com
```

simple HTTP response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
    <title>Example Web Page</title>
</head>
<body>
    <p>This is a sample web page</p>
</body>
</html>
```

Web Application Architecture

Application Layer: Application Programming Interface (API)

Application Programming Interface (API) enables developers to access specific data and functions of software. It works as mediator that allows apps to communicate with each other. It comprises protocols, tools, and subroutine definitions required to build apps.

Example: Gmail, Google Map

- With APIs, developers don’t have to create everything from scratch but use existing functions exposed as an API to increase productivity and gain faster time to market.
- By reducing development efforts, APIs significantly reduce development costs.
- It also improves collaboration and connectivity across the ecosystem while enhancing customer experience.

Web Application Architecture

Application Layer: Application Programming Interface (API)

Types of APIs:

RESTful API: Representational State Transfer API in lightweight JSON format. It is highly scalable, dependable and delivers fast performance, making it the most popular API.

SOAP: Simple Object Access Protocol uses XML for data transmission. It requires more bandwidth and advanced security

XML-RPC: Extensible Markup Language – Remote Procedure Calls use specific XML format for data transmission

JSON-RPC: It uses JSON format for data transmission

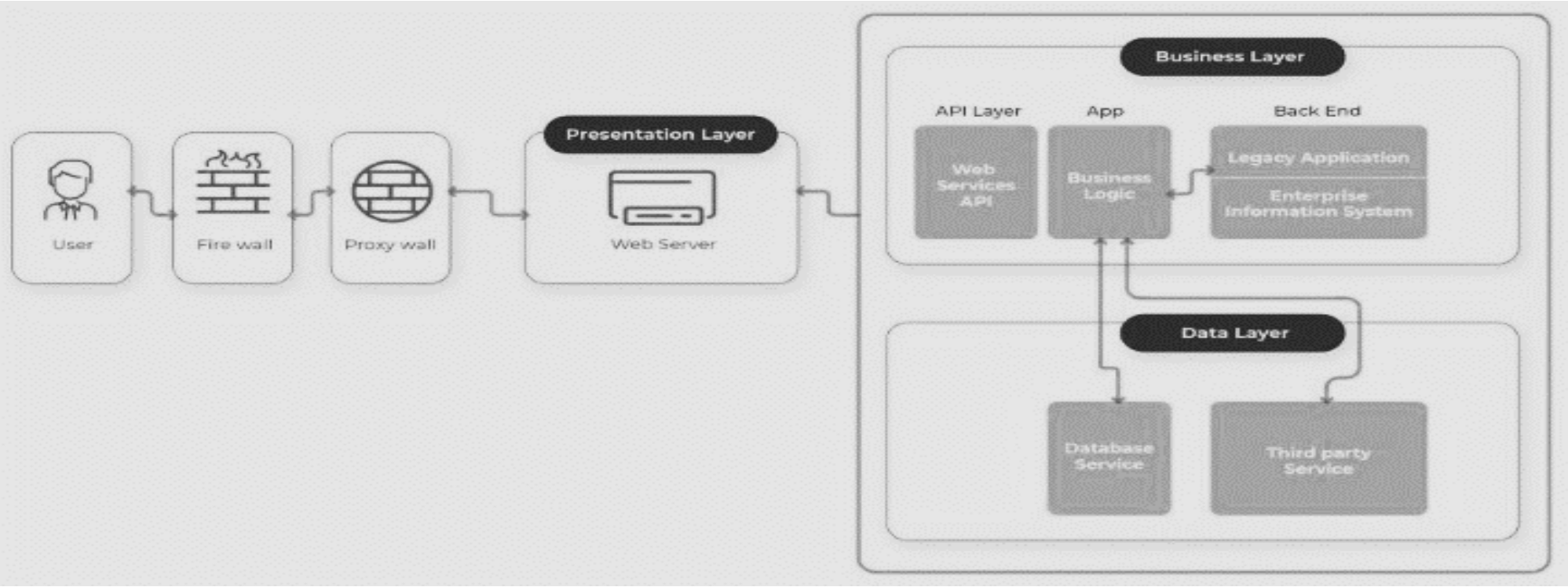
Web Application Architecture

Web service	API
Web service is a collection of open source protocols and standards used for exchanging data between systems. (interacting two systems over a network)	API(Application Programming Interface) is a software interface that allows two applications to interact with each other without any user involvement.
Web service is used for REST, SOAP and XML-RPC for communication.	API is used for any style of communication.
supports only HTTP protocol	supports HTTP/HTTPS protocol
Exposed in XML	Exposed in com object, DLL or .H files in C/C++, JAR files in Java, XML, JSON, etc..

All web services are APIs, but all APIs are not web service.
Web services might not perform all the operations that API would perform.
Web services always needs a network for its operation whereas API doesn't need

API Example: Google Map, Sign in, MakemyTrip

Web Application Architecture



Web Application Architecture

Data Layer: Database

- A database is a key component of a web application that stores and manages information for a web app.
- Using a function, User can search, filter and sort information based on request and present the required info to the end user.
- There are four aspects to work with database of a web application: size, speed, scalability, and structure.
- For structured data, SQL-based databases are good choice, it suits financial apps wherein data integrity is a key requirement.
- To handle unstructured data, NoSQL is a good option. It suits apps wherein the nature of incoming data is not predictable. Key Value databases associate each value with a key and suit databases that store user profiles, reviews, blog comments, etc.
- **Cloud storage** is about storing the data in the cloud and accessing it over the Internet. A cloud service provider provides the storage infrastructure on a pay-per-use subscription model. Eg. Amazon S3 by AWS, Azure by Microsoft, Google Cloud storage

Types of Web Application Architecture

Single-Page Applications (SPAs) :– Instead of loading completely new pages from the server each time for a user action, single page web applications allows for a dynamic interaction by providing updated content to the current page.

Micro services :- By cutting a large software application into loosely coupled units, which communicate with each other through APIs. These are small and lightweight services that execute a single functionality where developers working with the Micro services Architecture are free to pick up a technology stack of choice. It makes developing the application simpler and quicker.

Serverless Architectures :- An application developer consults a third-party cloud infrastructure services provider for outsourcing server as well as infrastructure management.

MVC – Model View Controller :- The MVC architecture splits the app logic into three components based on the functionality. Model: Data storage model, View: Components that the user can view and Controller: Intermediate component between the model and view components.

References

- https://www.clickittech.com/devops/web-application-architecture/
- https://hackr.io/blog/web-application-architecture-definition-models-types-and-more