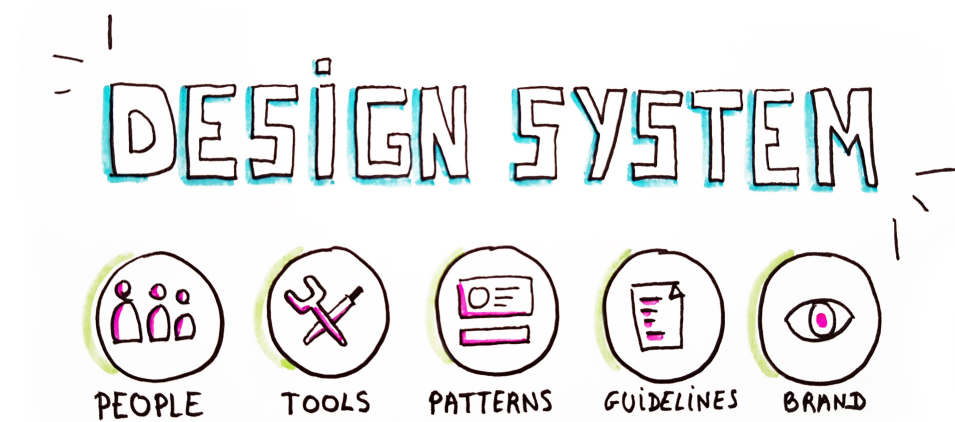# System Design



## Client Side Architecture

### Types of Client

Logical manipulation has to be done on data before returning to client then, The logic may reside on server side or client side. On this basis Client are classified into two types

1. Thin Client
2. Thick Client

- Thin Client

When Logic that has to be performed on data reside on Server Side then it is classified as Thin Client. It depends on Cloud platform for its features.

Eg: Netflix

- Thick Client

When The logic that has to be performed on data resides on server side then it is known as Thick Client. It's function can also run offline. and runs at least some features locally on your device.
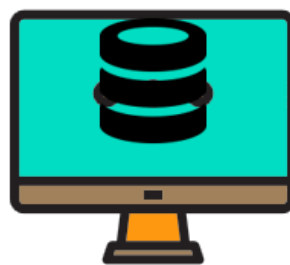
Eg: Google Docs  (It perform calculations and computational related stuff using local resources but data is stored in database).

## Database Architecture (Tier 1, Tier 2, Tier 3)

There are Mainly three types of Database architectures

### Tier 1 Architecture

This is the most simplest of all the database, server and client all reside at a same machine.
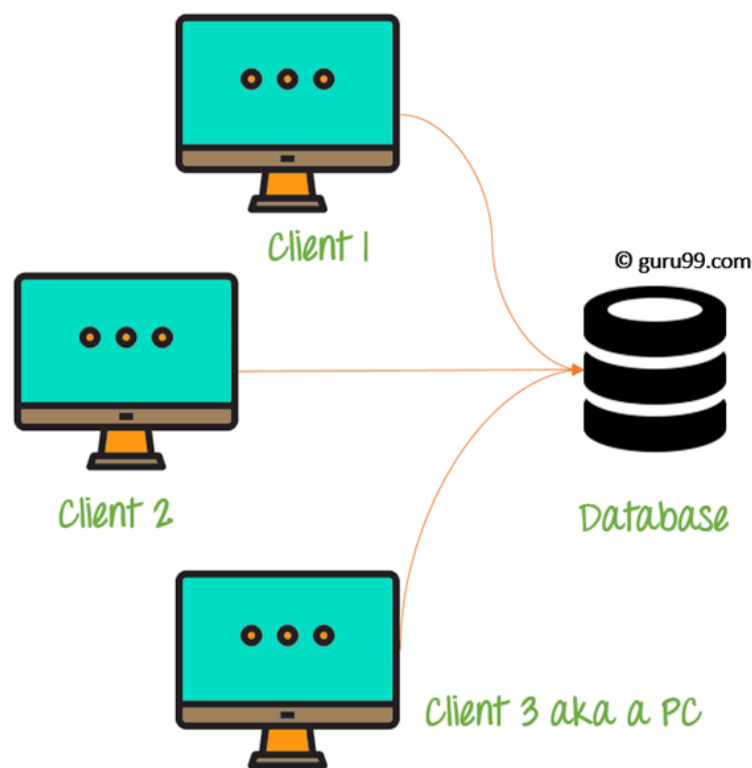


Single Tier Architecture

These could be installed and run on a single machine without any external resources.

Eg: when we install an SQL database to practice SQL quires.

## Tier 2 Architecture

In this architecture the presentation layer runs on client side while database resides at an external resource.
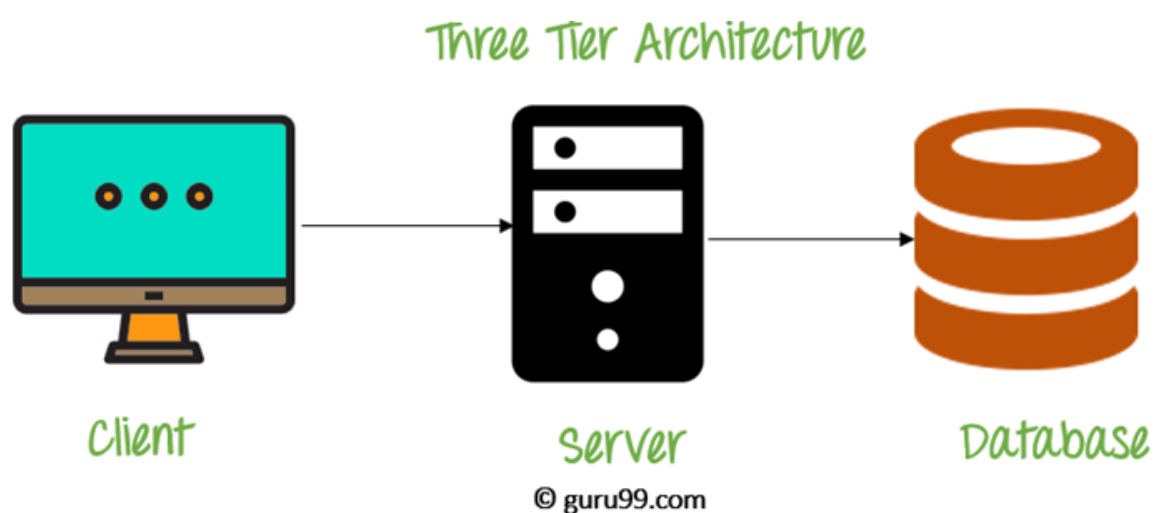


It has better security and scalability scope as compared to Tier 1. Here logic can reside either on client side or side depending on service.

## Tier 3 Architecture

A three tier architecture is most scalable among the three. It has three layers

- Presentation Layer (PC, Tablet or any sort of UI Screen)
- Application layer (server)
- Database Server



Here Logic resides at server side and data is stored at database. This architecture can integrate data from different resources

In some cases even Tier 3 architecture isn't sufficient so load balance, caching layer, etc are introduced
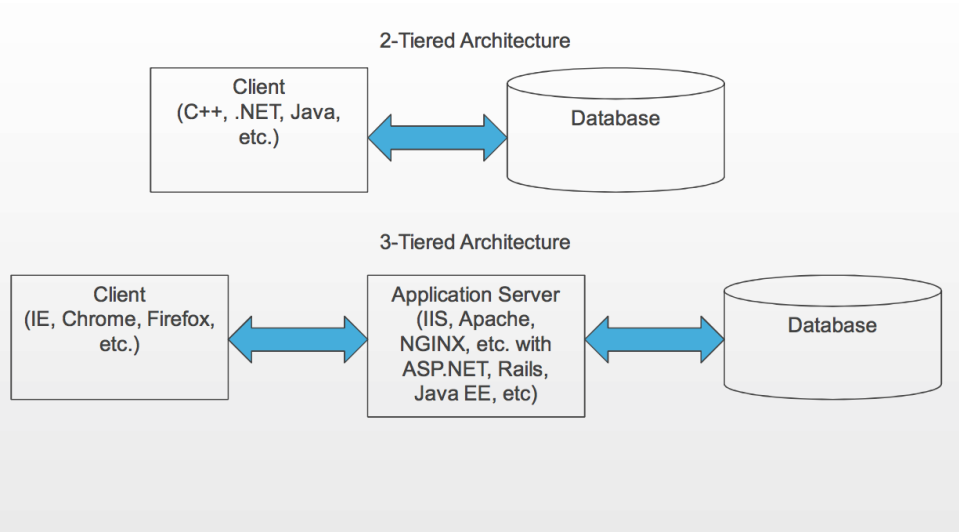
# How Internet Works ?

## What Happens when you hit a web address into web Browser

- Firstly the web address is converted into an IP using DNS lookup then your browser look up in local browser cache if it didn't find a copy in browser's cache then it will look up at OS cache.

- If it didn't finds a cached version then it will look up at ISP. If its not present even there then it will give recursive calls to several other bigger ISPs and then lastly it will call TLD ( Top Level Domain).

- Once the IP address if found then client sends a HTTP request to server to send a copy of website using TCP/IP.

- The Server then respond with a "200"  response code. and send contents of website in packets to client.

**Request Methods**

| Aa Method | ☰ Description |
|-----------|----------------|
| Untitled  |                |
| GET       | Used to get contents from server Eg: playing a new unloaded YouTube Video |
| PUT       | Replaces an existing target with a new value Eg: Changing password |
| DELETE    | Removes an existing target Eg: Delete a tweet or a Instagram post |
| POST      | Used to send data to server Eg: Submit button in google forms |

**HTTP Response Codes**

| Aa Response Code | ☰ Description |
|-------------------|----------------|
| Untitled          |                |
| 1XX               | Request is still in progress |
| 2XX               | OK |
| 3XX               | Redirect, Request to some other server |
| 4XX               | Request Invalid (Client Side Issue) |
| 5XX               | Server Side Issue (Server Down etc) |

# HTTP Status Codes

**Level 200 (Success)**

**200 : OK**

**201 : Created**

**203 : Non-Authoritative Information**

**204 : No Content**

**Level 400**

**400 : Bad Request**

**401 : Unauthorized**

**403 : Forbidden**

**404 : Not Found**

**409 : Conflict**

**Level 500**

**500 : Internal Server Error**

**503 : Service Unavailable**

**501 : Not Implemented**

**504 : Gateway Timeout**

**599 : Network timeout**

**502 : Bad Gateway**

**Author:  Uzair Ali**