J2EE

- Sandip Upadhyay

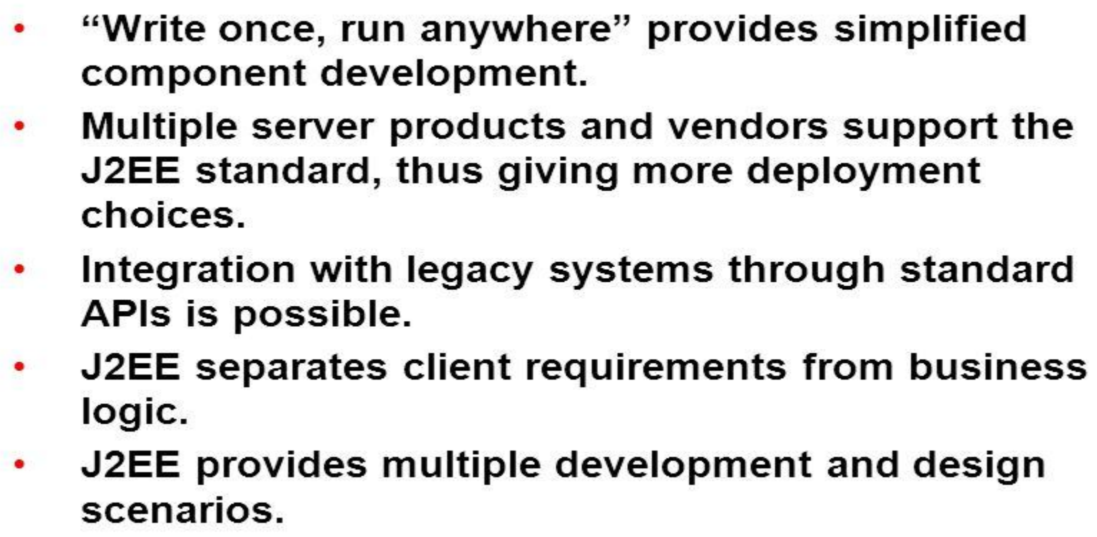
Sandip Upadhyay

What is J2EE?

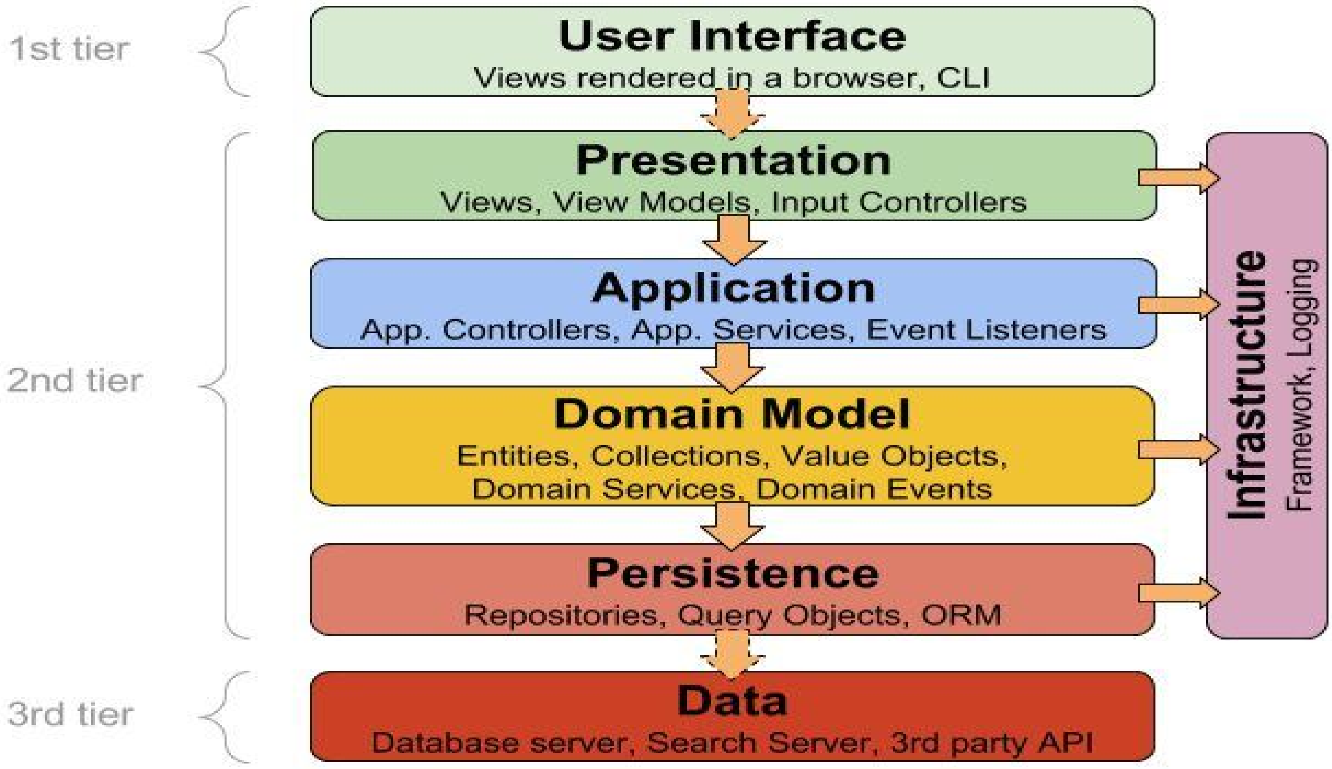
- J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying **Web-based enterprise** applications online.
- The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multi tiered, Web-based applications.

  core java is the basic java programming and is called as the j2Se(Java standard edition.)This is basically used to develop standalone programs.
- Based on this sun had released the J2EE(Java enterprise edition.)This is basically used to develop the enterprise applications and also web applications.J2EE consists of Servlets, Jsp, Jdbc, EJb.

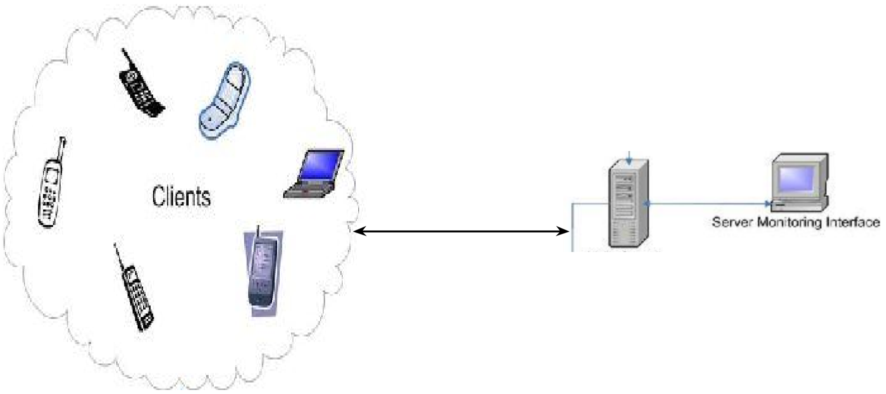
- Advantage:

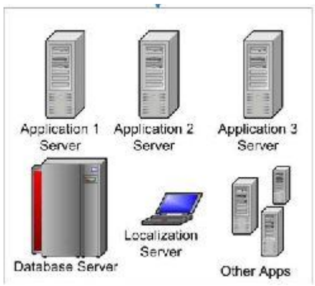


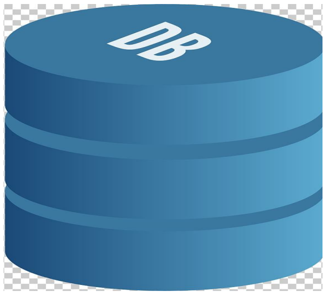
- “Write once, run anywhere” provides simplified component development.
- Multiple server products and vendors support the J2EE standard, thus giving more deployment choices.
- Integration with legacy systems through standard APIs is possible.
- J2EE separates client requirements from business logic.
- J2EE provides multiple development and design scenarios.


1st tier
User Interface
Views rendered in a browser, CLI
2nd tier
Presentation
Views, View Models, Input Controllers
Application
App. Controllers, App. Services, Event Listeners
Domain Model
Entities, Collections, Value Objects,
Domain Services, Domain Events
Persistence
Repositories, Query Objects, ORM
Infrastructure
Framework, Logging
3rd tier
Data
Database server, Search Server, 3rd party API

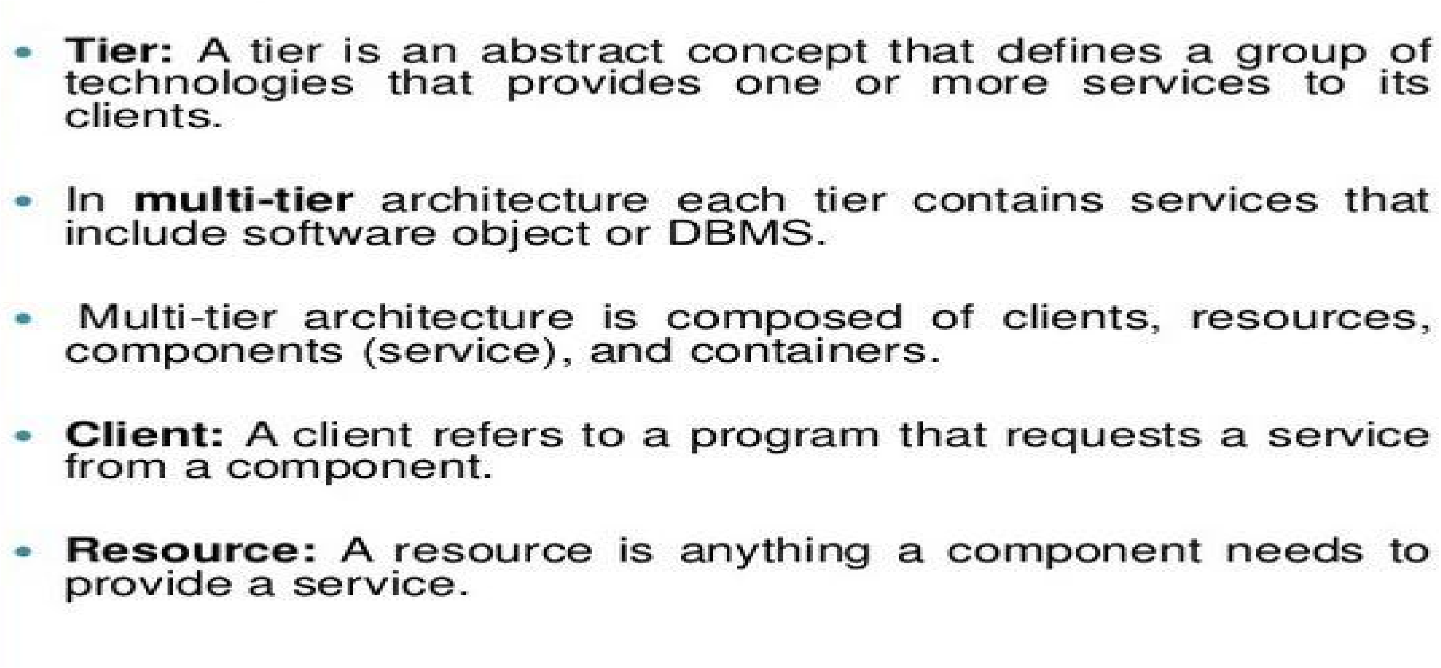
Sandip Upadhyay


Clients
Server Monitoring Interface


Application 1 Server
Application 2 Server
Application 3 Server
Database Server
Localization Server
Other Apps


DB

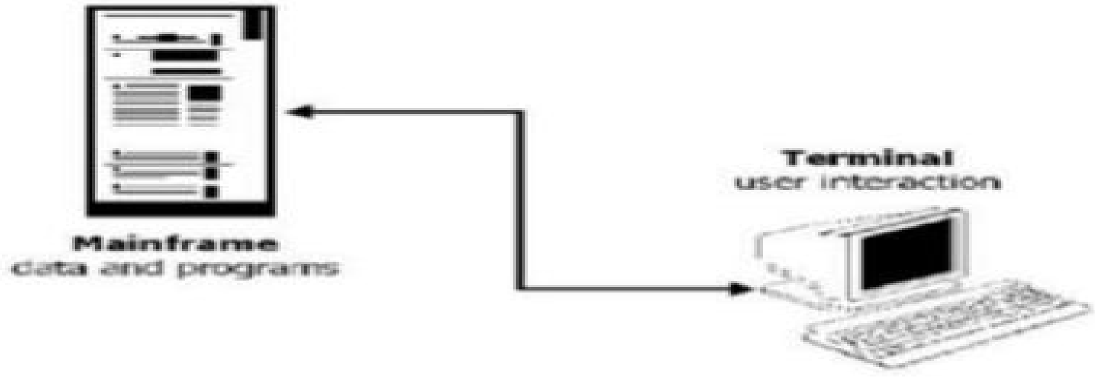
What is Tier?



- **Tier:** A tier is an abstract concept that defines a group of technologies that provides one or more services to its clients.
- In **multi-tier** architecture each tier contains services that include software object or DBMS.
- Multi-tier architecture is composed of clients, resources, components (service), and containers.
- **Client:** A client refers to a program that requests a service from a component.
- **Resource:** A resource is anything a component needs to provide a service.

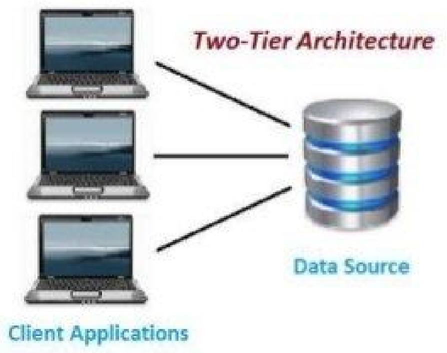
Single Tier Architecture

- Time of Huge Main-Frame computers
- Single Computer manage all processes
- All Resources Attached to the same computer
- Access via dumb Terminals

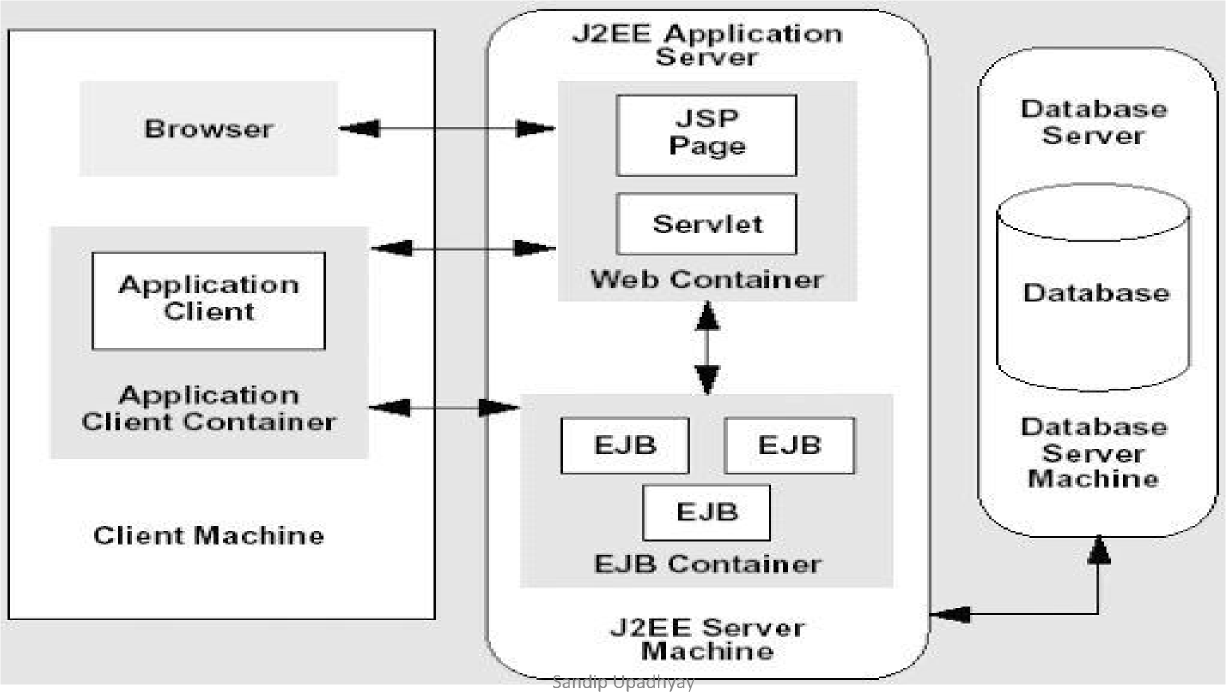


2-TIER ARCHITECTURE

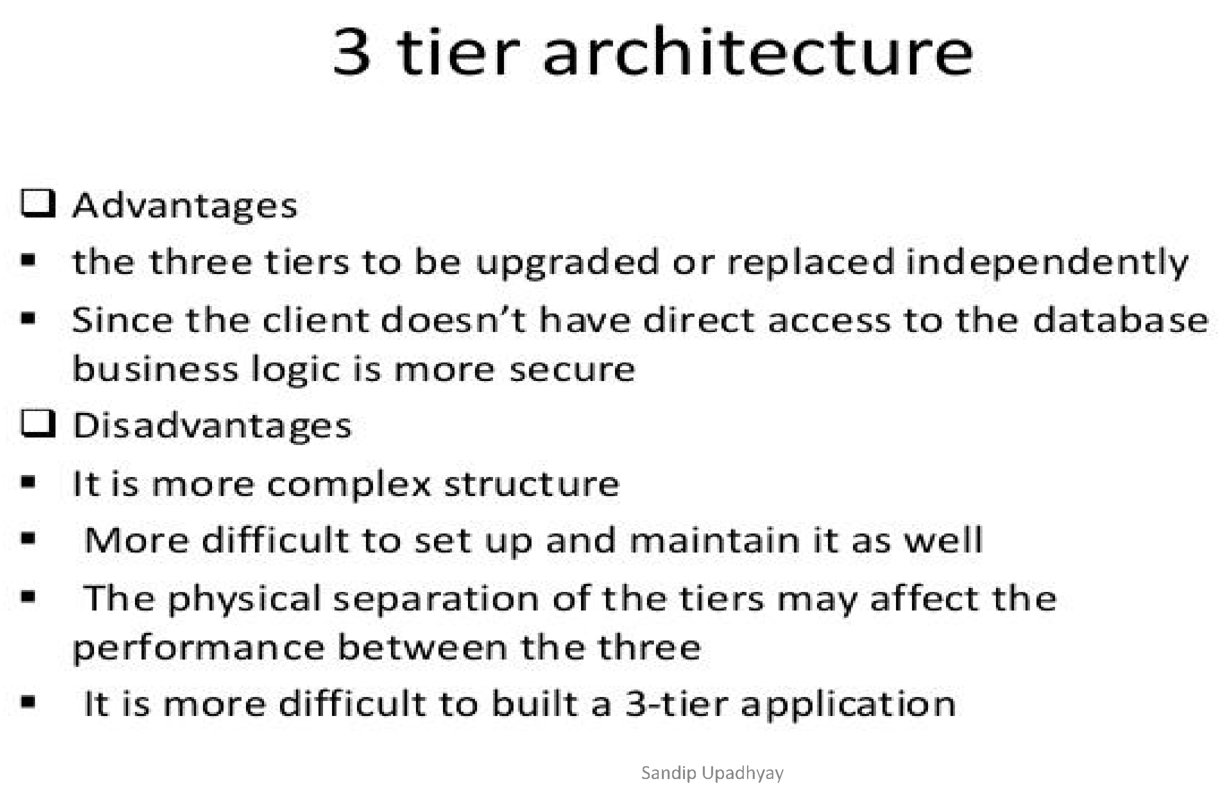
- It is client-server architecture
- Direct communication
- Run faster(tight coupled)



3 – Tier Architecture



Sandip Upadhyay



3 tier architecture

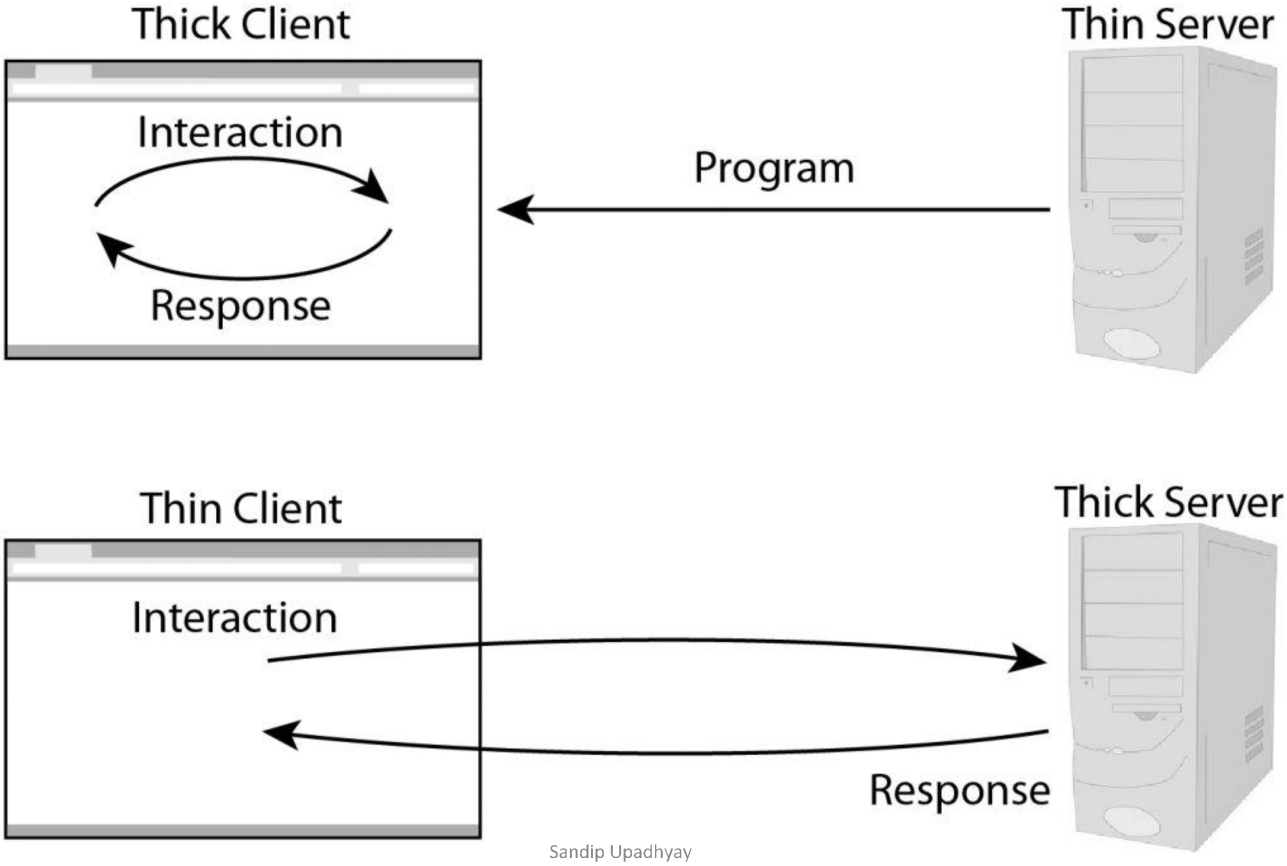
❑ Advantages

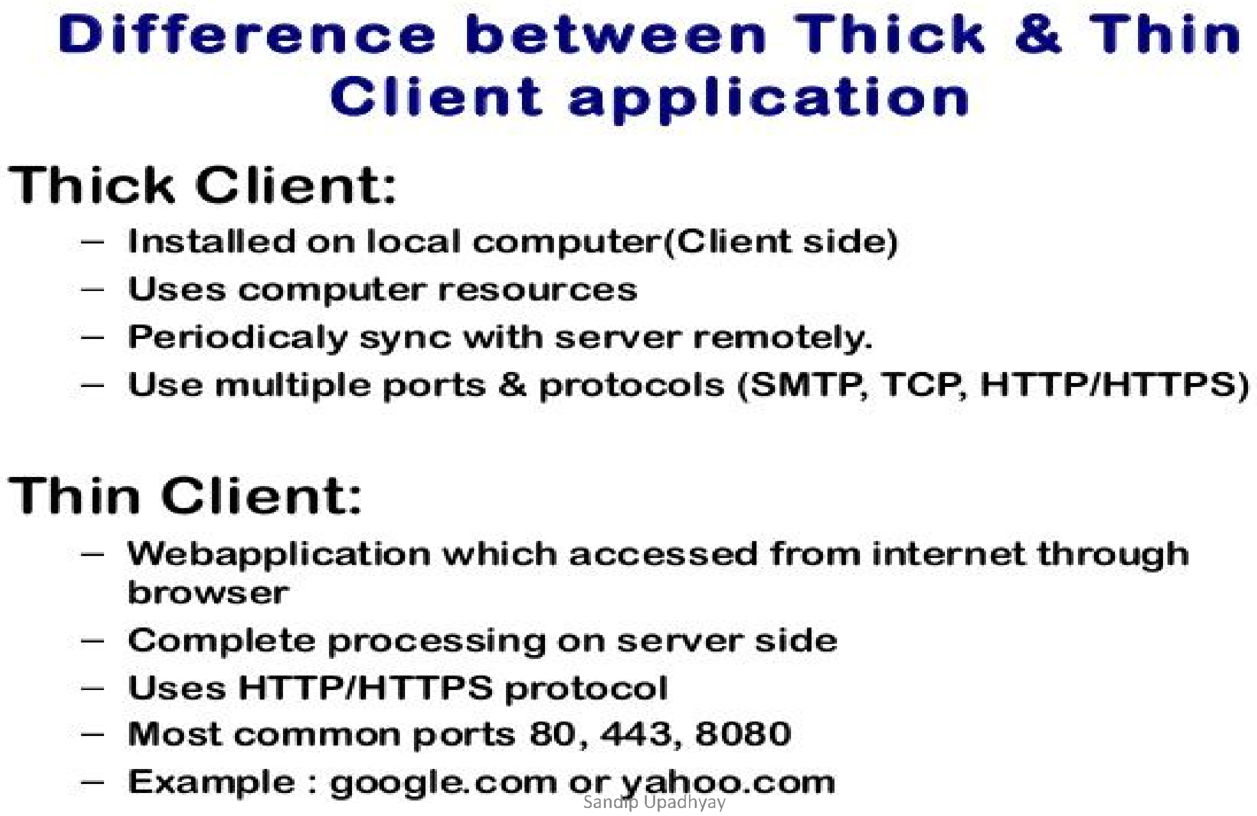
- the three tiers to be upgraded or replaced independently
- Since the client doesn’t have direct access to the database business logic is more secure

❑ Disadvantages

- It is more complex structure
- More difficult to set up and maintain it as well
- The physical separation of the tiers may affect the performance between the three
- It is more difficult to built a 3-tier application

Sandip Upadhyay


Thick Client
Thin Server
Interaction
Response
Program
Thin Client
Thick Server
Interaction
Response



Difference between Thick & Thin Client application

Thick Client:

- Installed on local computer(Client side)
- Uses computer resources
- Periodicaly sync with server remotely.
- Use multiple ports & protocols (SMTP, TCP, HTTP/HTTPS)

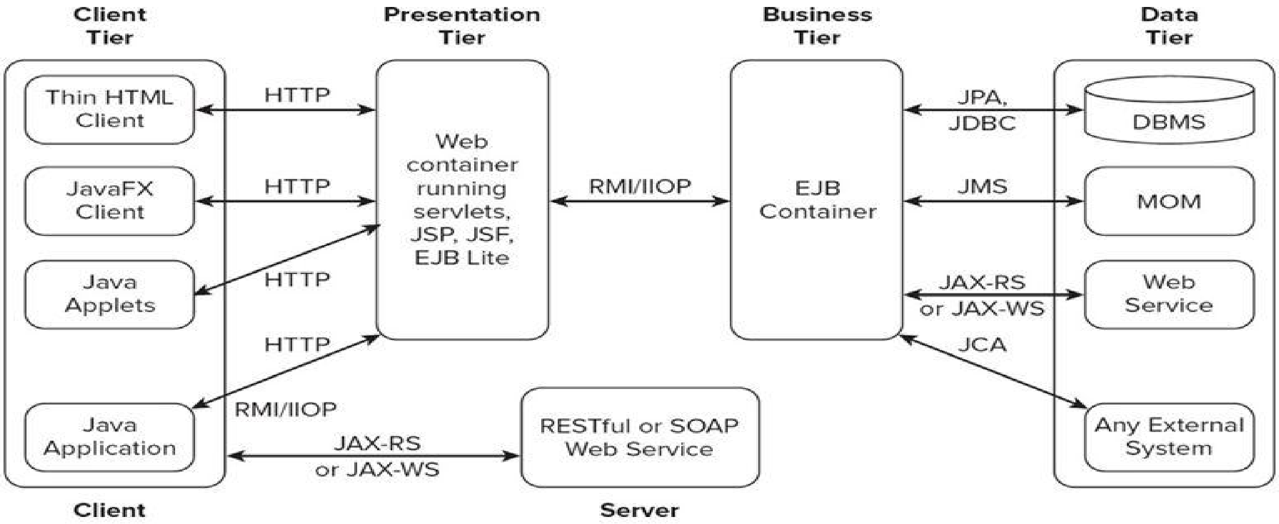
Thin Client:

- Webapplication which accessed from internet through browser
- Complete processing on server side
- Uses HTTP/HTTPS protocol
- Most common ports 80, 443, 8080
- Example : google.com or yahoo.com

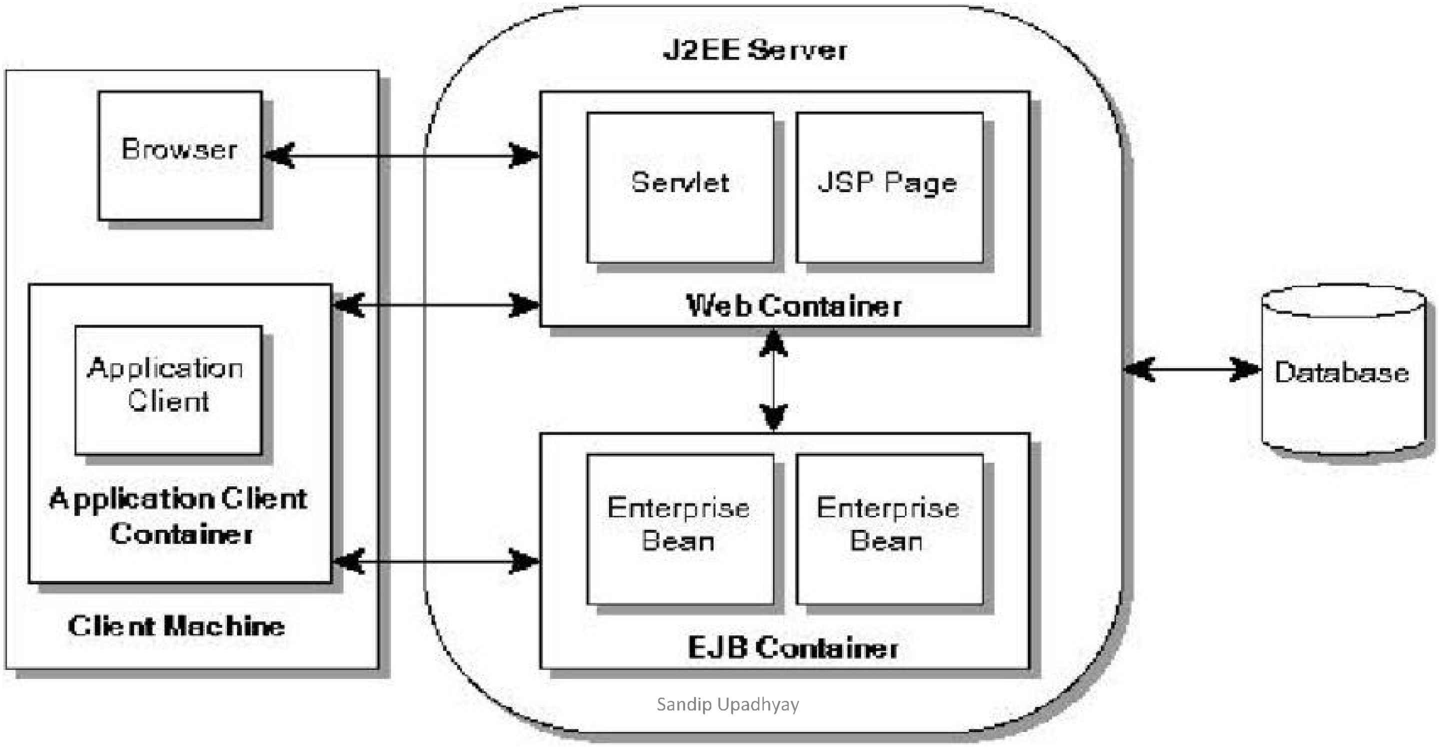
J2EE API

- Database: JDBC
- Directory:JNDI
- Transaction JTA,JTS
- Mail: Java Mail
- Messaging: JMS
- Communication: RMI
- Server-Side Component: EJB
- Presentation-Side: Servlet, JSP

J2EE APIs – The big picture



J2EE Architecture

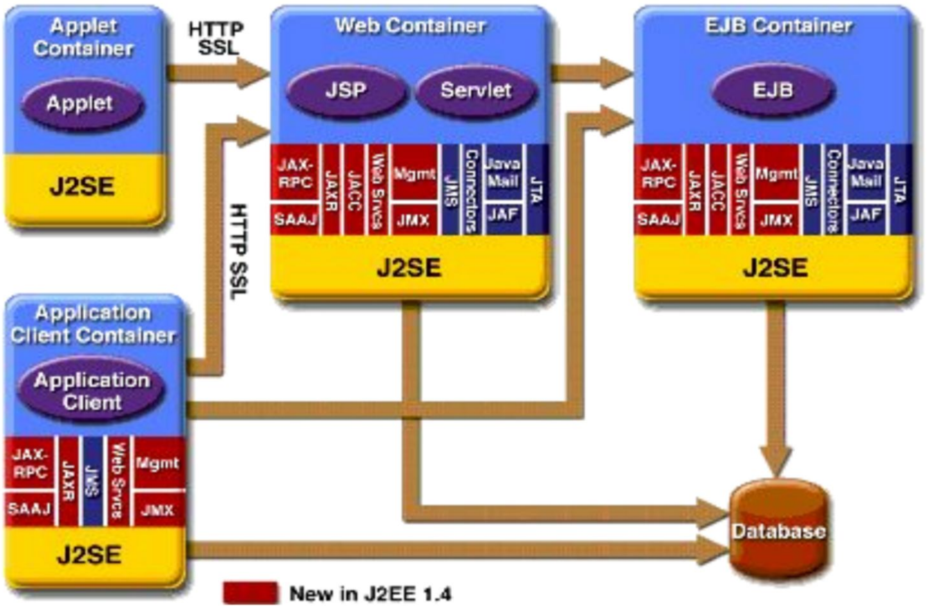


Sandip Upadhyay

What is Container ?

- A **container** is a component which can contain other components inside itself.
- It is also an instance of a subclass of **java**.
- **Container** extends **java**. awt. Component so **containers** are themselves components.

Types of container



Advantages:

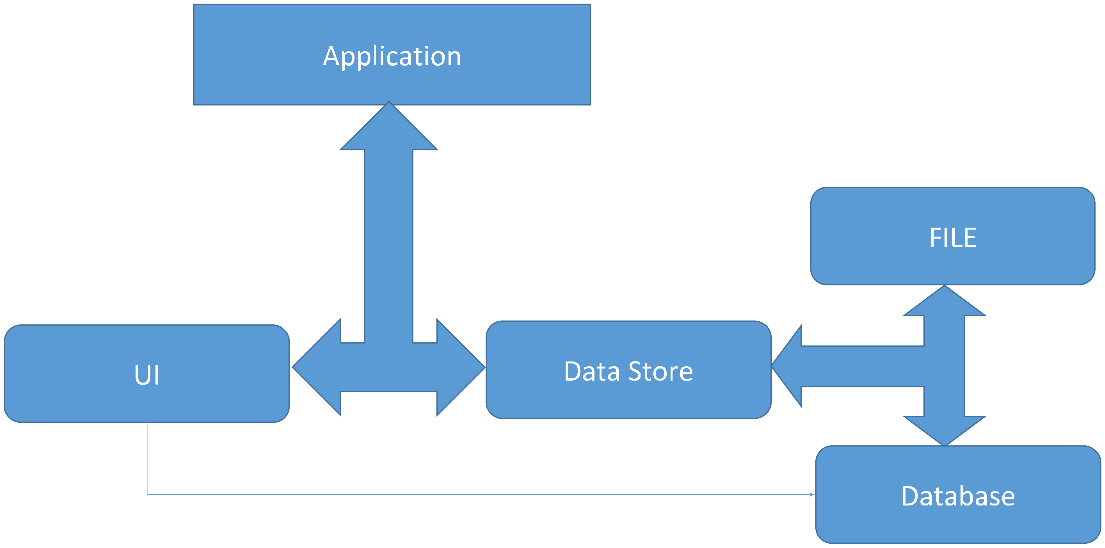
Improved maintainability
Consistency
Interoperability
Flexible
Scalability
Easy
Reusable

Dis-Advantage:

-Expensive
-Require more time
-Require more physical resouces

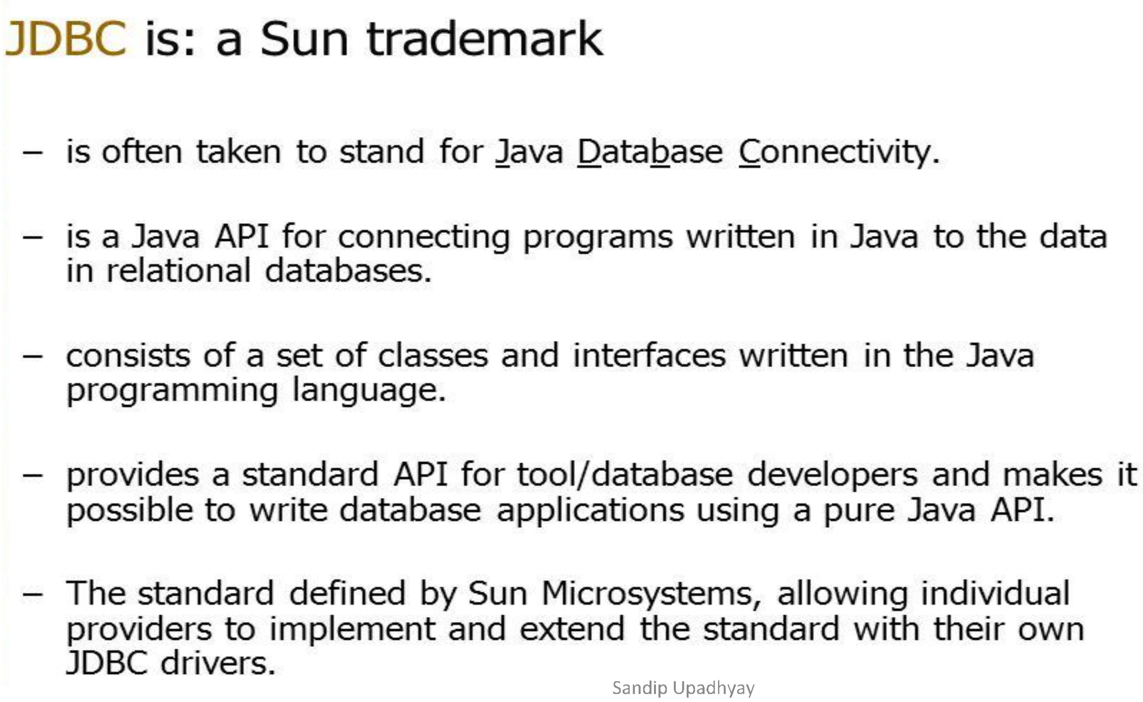
Tomcat as a web container

- Apache Tomcat is a long-lived, open source Java servlet container that implements several core Java enterprise specs, namely the Java Servlet, JavaServer Pages (JSP), and WebSockets APIs.
- An Apache Software Foundation project, Tomcat was first released in 1998, just four years after Java itself. Tomcat started as a reference implementation for the first Java Servlet API and the JSP spec. While it's no longer the reference implementation for either of these technologies, Tomcat remains the most widely used Java server, boasting a well-tested and proven core engine with good extensibility



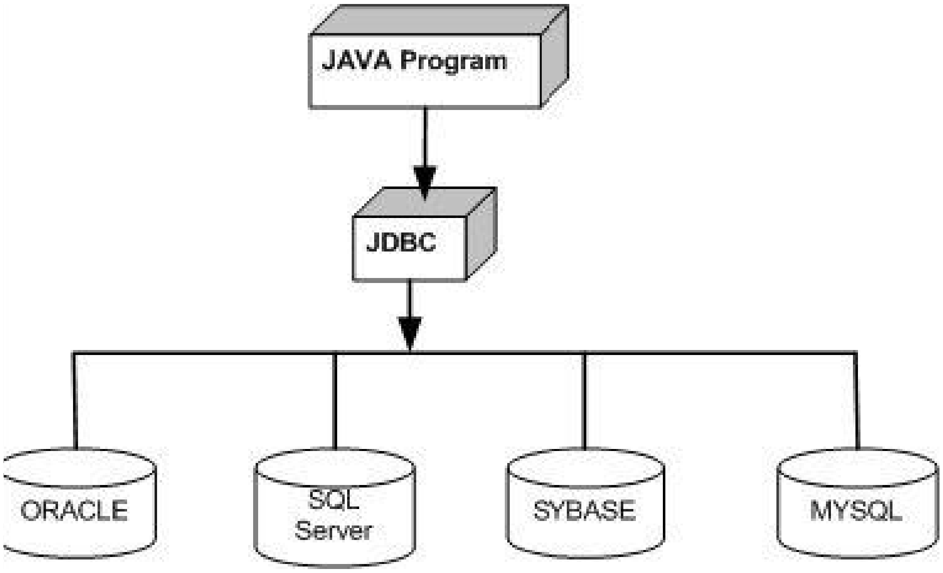
Sandip Upadhyay

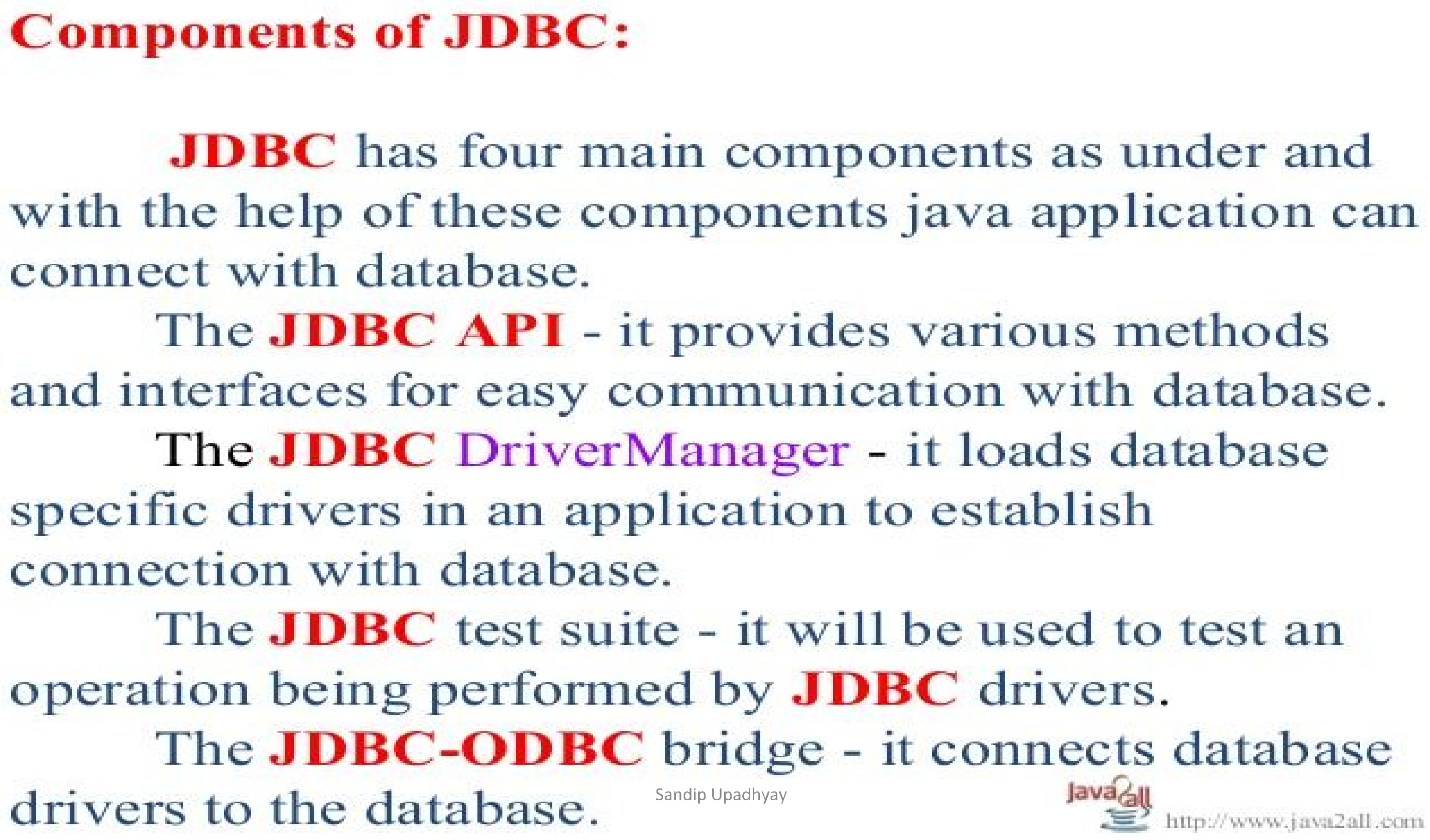
JDBC (Java Database connection)



JDBC is: a Sun trademark

- is often taken to stand for Java Database Connectivity.
- is a Java API for connecting programs written in Java to the data in relational databases.
- consists of a set of classes and interfaces written in the Java programming language.
- provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.
- The standard defined by Sun Microsystems, allowing individual providers to implement and extend the standard with their own JDBC drivers.


JAVA Program
JDBC
ORACLE
SQL Server
SYBASE
MYSQL



Components of JDBC:

JDBC has four main components as under and with the help of these components java application can connect with database.

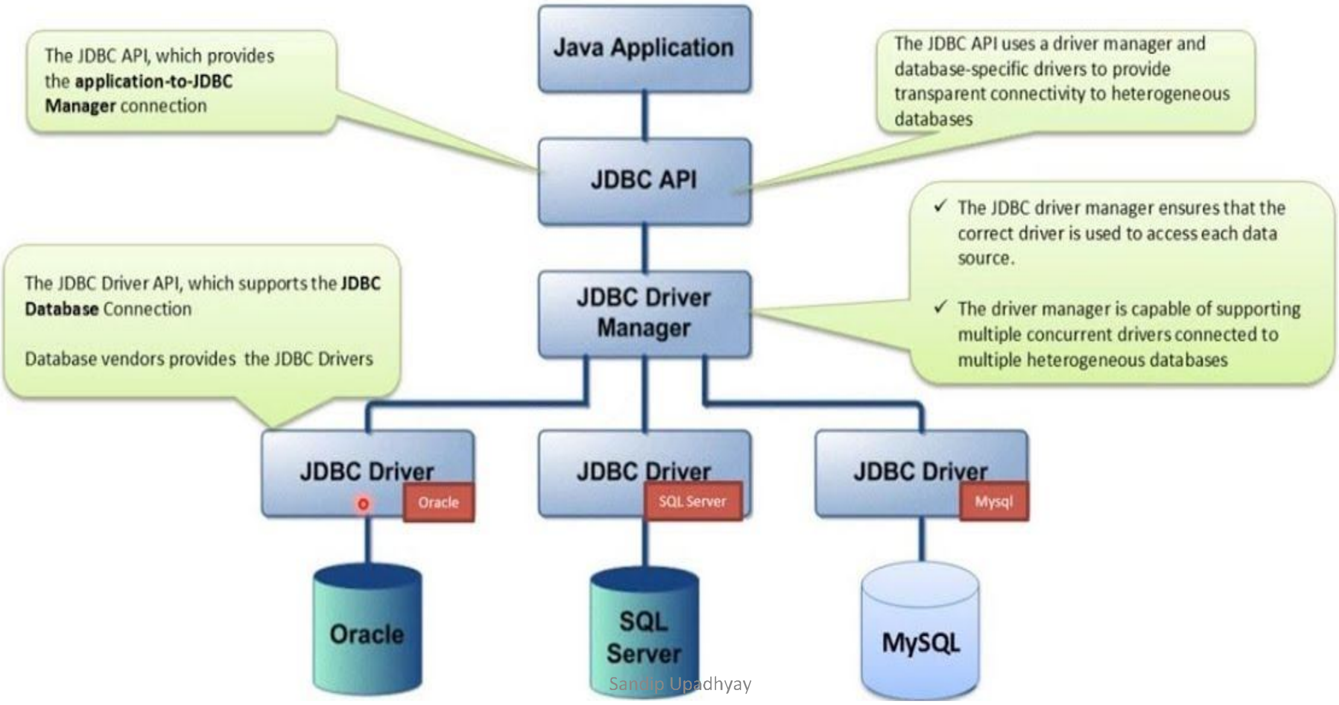
The JDBC API - it provides various methods and interfaces for easy communication with database.

The JDBC DriverManager - it loads database specific drivers in an application to establish connection with database.

The JDBC test suite - it will be used to test an operation being performed by JDBC drivers.

The JDBC-ODBC bridge - it connects database drivers to the database.

JDBC Architecture



The JDBC API, which provides the **application-to-JDBC Manager** connection

The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases

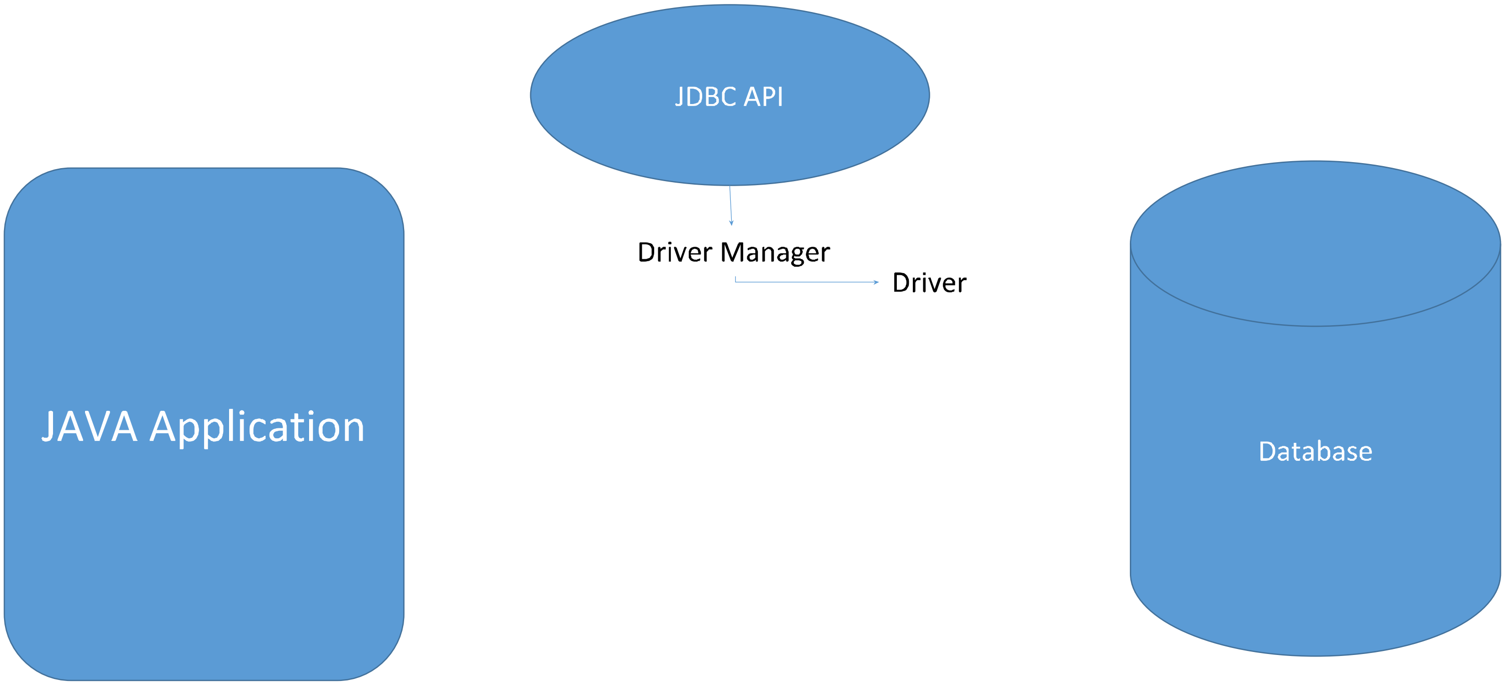
- ✓ The JDBC driver manager ensures that the correct driver is used to access each data source.
- ✓ The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases

The JDBC Driver API, which supports the **JDBC Database** Connection

Database vendors provides the JDBC Drivers

Driver Manager class

- This class manages a list of database drivers.
- Driver: this interface handles the communications with database server.
- Connection: this interface with all methods for connecting a database. The connection object represent communication context.
- Statement: this interface is used to submit the sql statements to database.
- ResultSet: this objects hold data retrieved from a database after you execute an sql query using statement object.
- Sql Exception: this class handles any errors that occur in a database application.



JAVA Application

JDBC API

Driver Manager

Driver

Database

- JDBC Data Types :

- JDBC Exception Handling:

Try and Catch