J2EE Introduction

TAKE SOME EXPERIENCE WITH HIGHER TECHNOLOGY

Editions of Java

There are mainly 3 editions available in Java:

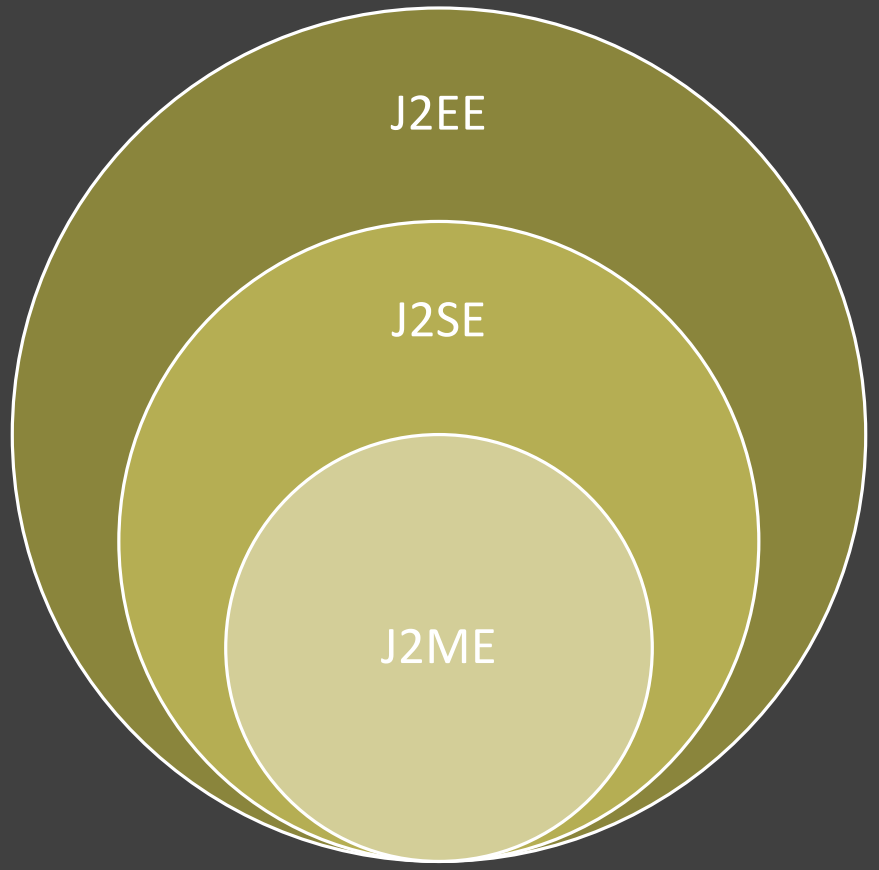
- J2SE
- J2EE
- J2ME

J2SE works as a basic core technology

J2EE works for higher level, scalable, component base technology

J2ME works for PDAs (Mobile Phones, Pagers, Remote Control etc.)

Editions as per their work



What is J2EE?

J2EE is an open source.

It reduces the cost of developing multi – tier applications.

It provides a number of APIs to develop the different types of applications, such as JDBC used for making connection between the java programs and database, JSP and servlets to create the dynamic web pages, EJB for creating the business logic. RMI to create the remote objects, client and server based application etc.

So it provides distributed computing framework and with this feature it reduces the complexity of the application.

J2EE platforms also include concept of container to provide the interface between the clients, server and the database.

It defines a flexible standard that can be built on either a single computer or deployed across several servers, each providing a specific set of J2EE supporting services:

Now a days J2EE provides many frameworks such as JSF, spring, struts, hibernate to develop very attractive, secure, user – friendly, database driven enterprise application.

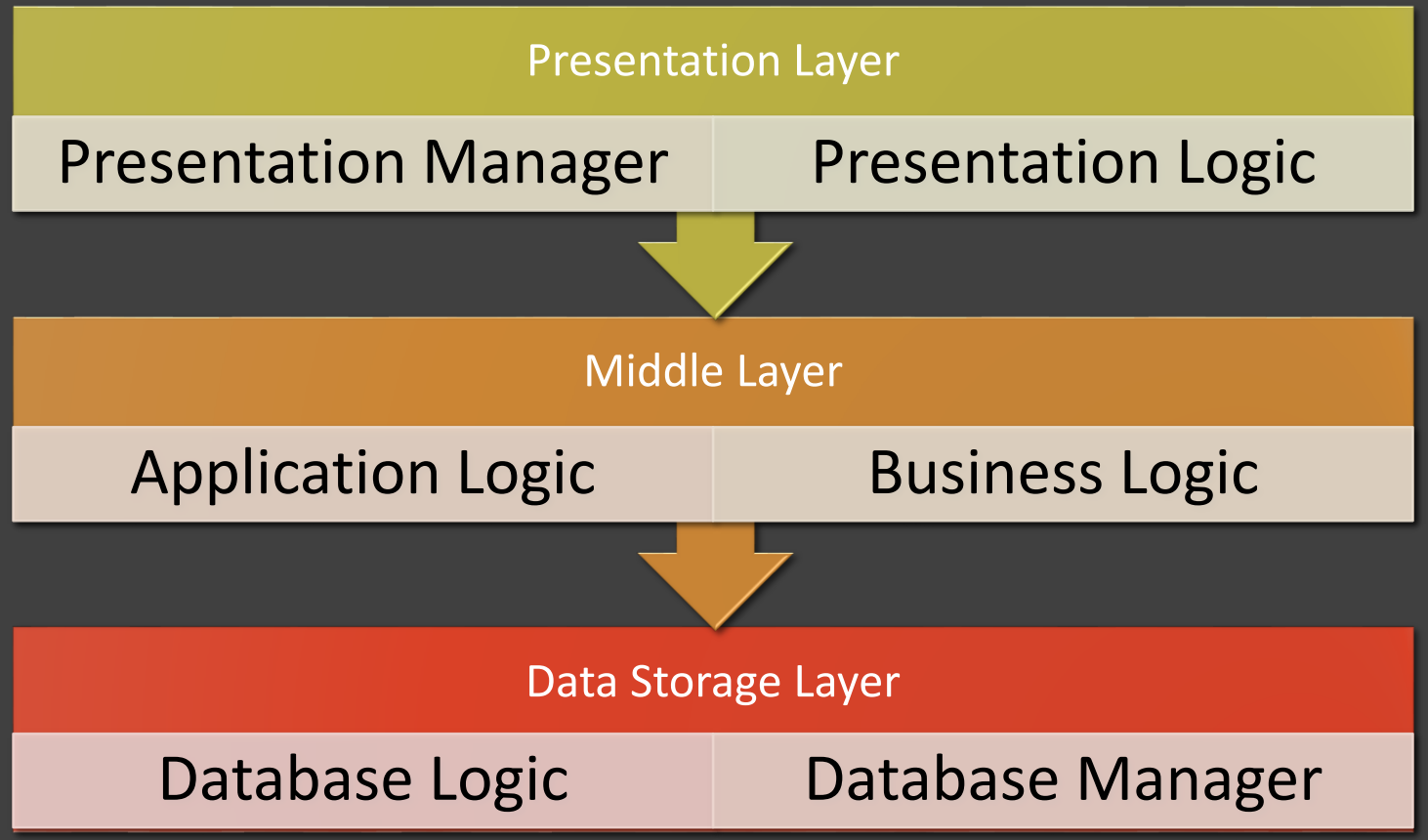
J2EE Platform

If you want to develop an enterprise application, there are 6 logical layers, which is related to logic of the client tier, middle tier and database tier.

It defines which layer belongs to which tier.

- Presentation Manager
- Presentation Logic
- Application Logic
- Business Logic
- Database Logic
- Database Manager

Software layers



Presentation Manager

The presentation manager defines the user interface.

It always resides on the client tier.

It manages the information displayed to the user.

Presentation Logic

The presentation logic defines the navigation system of the user interface, how and what will be displayed to the user.

It may reside with client tier or business tier or database tier based on thin client and thick client and application tier.

Application Logic

Application logic defines actual application logic with it.

Application logic can be connectivity, validations etc.

It may reside with the client tier or business tier or database tier based on thin client and thick client and application tier.

Business Logic

The business logic layer contains the business rules of the application.

It should be shared with the whole application.

It may reside with the business tier or database tier based on thin client and thick client and application tier.

Database Logic

The database logic defines the table structure and the relationship between the tables.

It also includes all the constraints of the table.

It always resides with database tier.

Database Manager

The Database Manager stores the persistent data.

It always resides with database tier.

Single Tier Architecture

Simple software applications are written to run on a single computer.

User inputs its verification, business logic and data accessed, all these could be found together on single computer, this kind of architecture is called as **Single Tier Architecture.**

Because all logic application services, the presentation, business rules and the data access layers exist in a single computer layer.

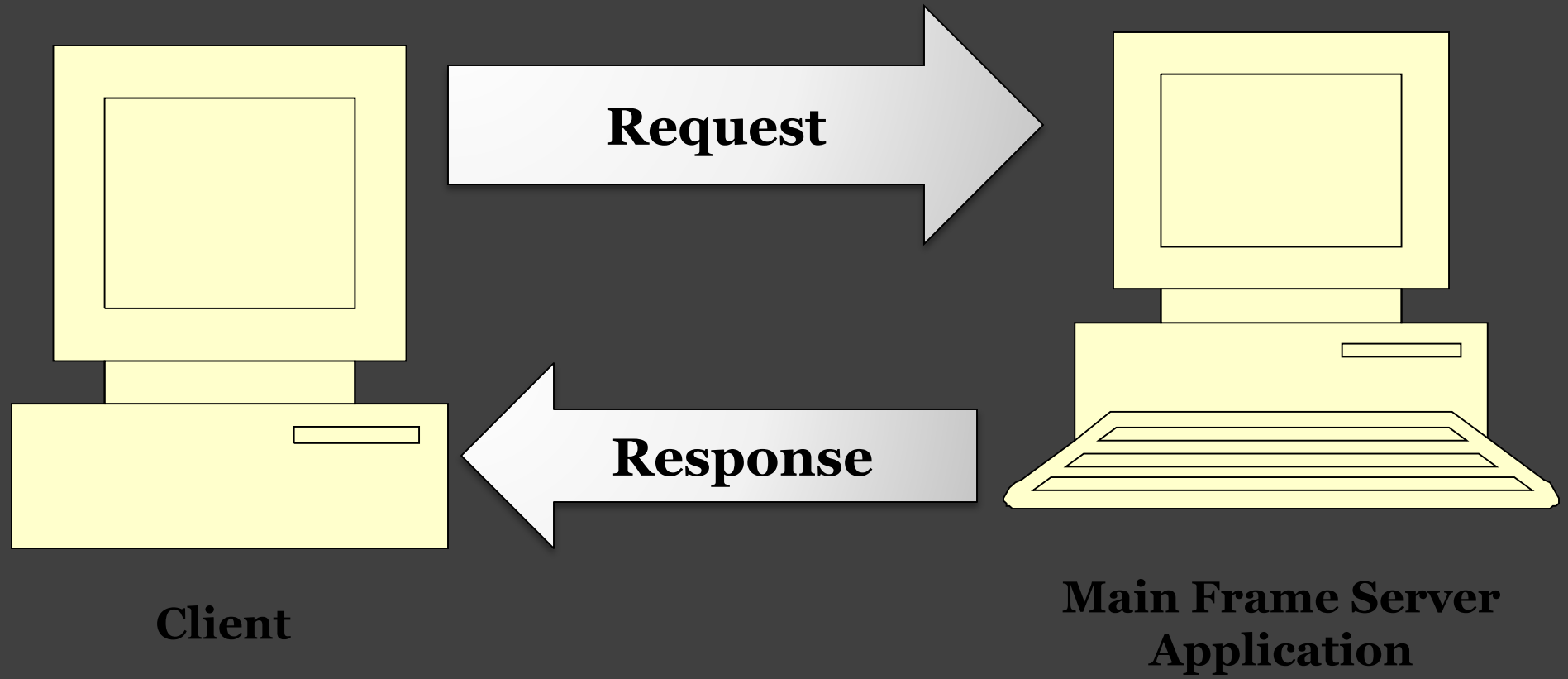
Advantages:

- Single tier system is relatively easy to manage.
- Data consistency is simple because data is stored at only one single location.

Disadvantages:

- However in the world now a days single storage location is not sufficient because of the changing business needs.
- With the single tier application we can not share the data in the large amount; it can also not handle multiple users.
- Because of many such reason two tier architecture is required.

Single Tier Architecture



Two Tier Architecture

Application which is divided into two separate tiers, client machine and database server machine is called as **Two tier architecture application**.

The application includes the presentation and business logic.

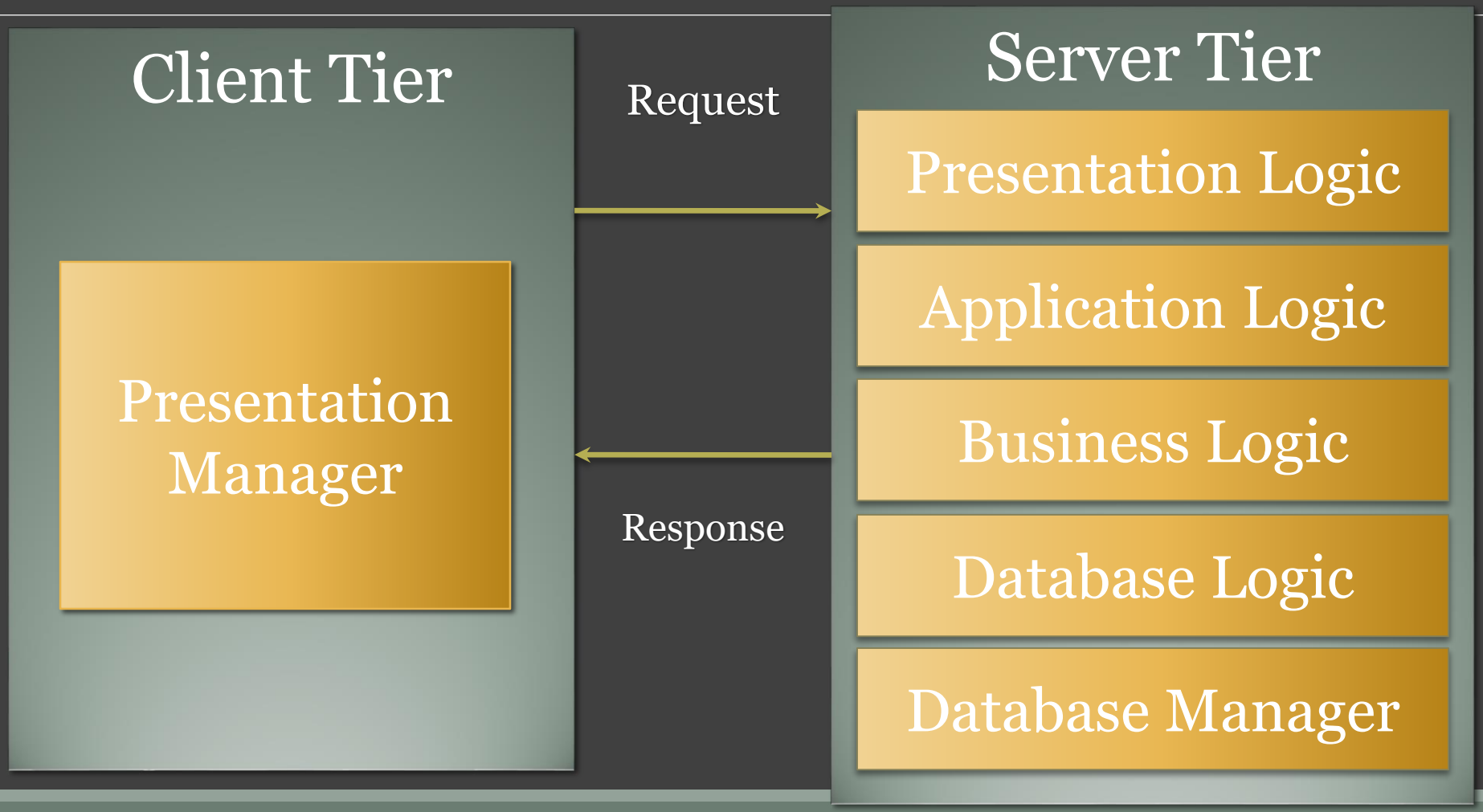
Data is accessed by connecting client machine to a database which is lying on another machine.

Thin Client

With the two tier architecture, if the presentation manager resides only with the client tier then the client is called as thin client.

Other presentation logic, application logic, business logic, database logic and database manager reside with the server side (database side).

Thin Client

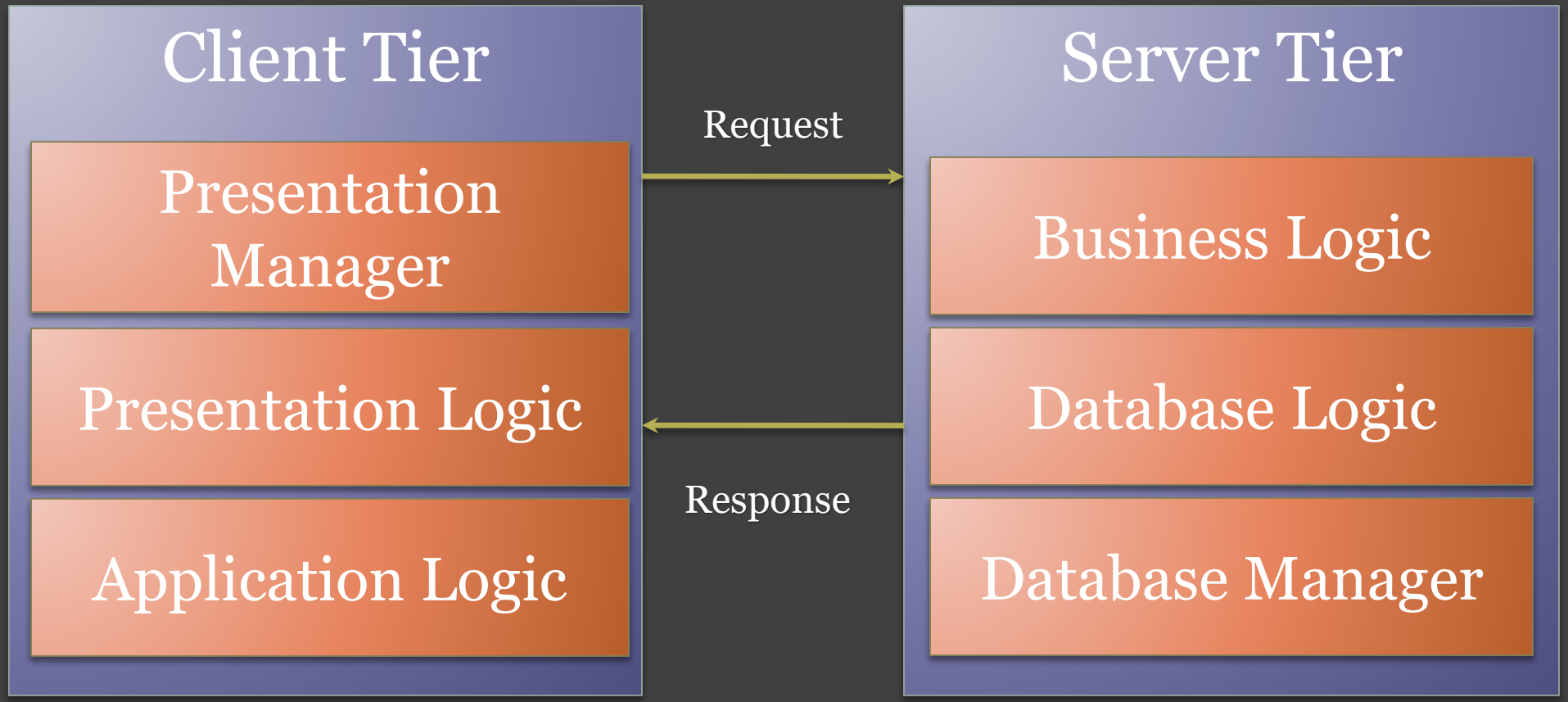

Client Tier
Presentation Manager
Request
Response
Server Tier
Presentation Logic
Application Logic
Business Logic
Database Logic
Database Manager

Thick Client

With the two tier architecture if the presentation manager, presentation logic, application logic reside with the client tier then the client is called as thick client.

Others like business logic, database logic and database manager reside with the server side (database side).

Thick Client

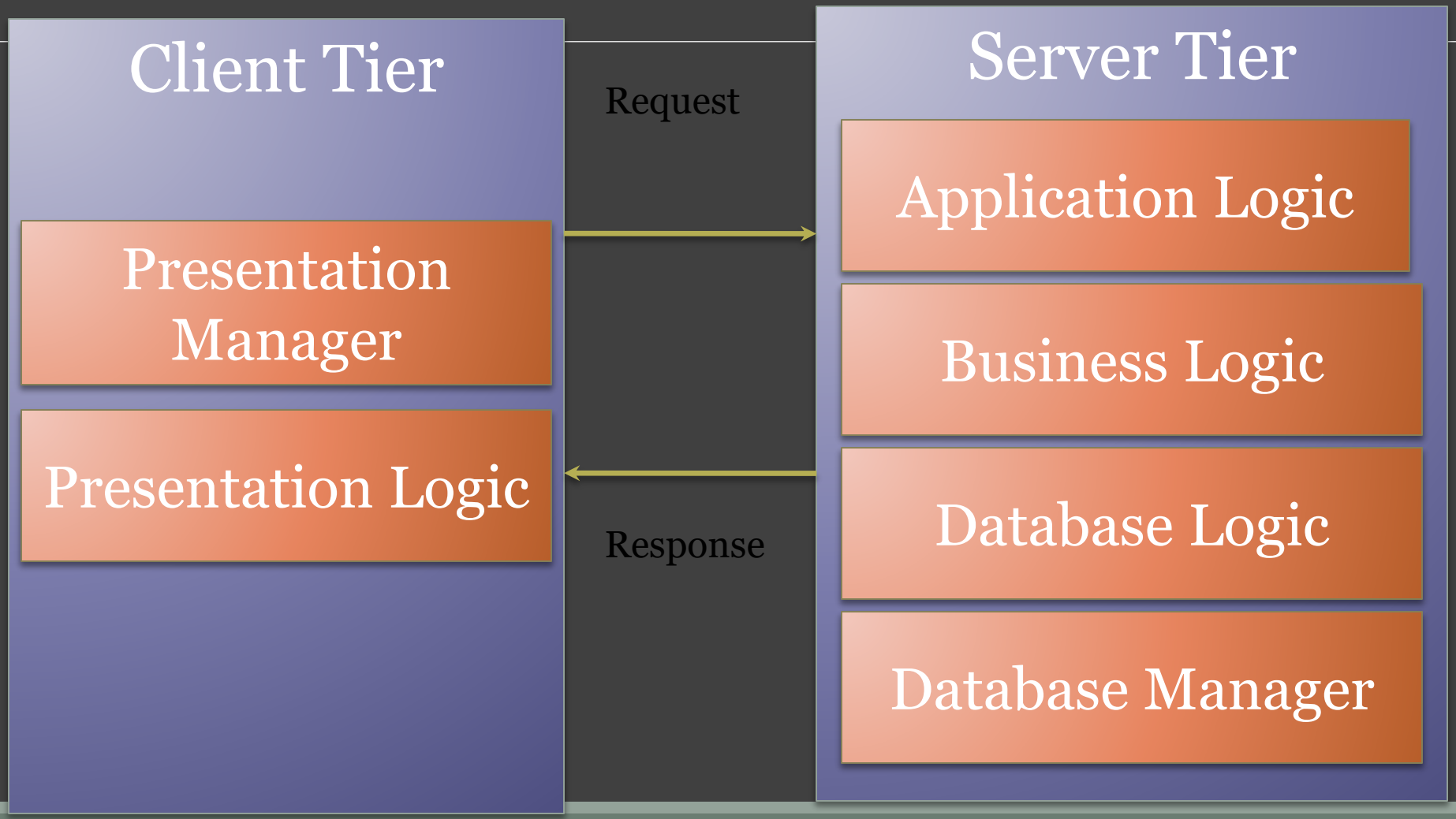


Normal client

With the two tier architecture if the presentation manager and presentation logic reside with the client tier then the client is called as normal client.

Others like application logic, business logic, database logic and database manager reside with the server side (database side).

Normal Client



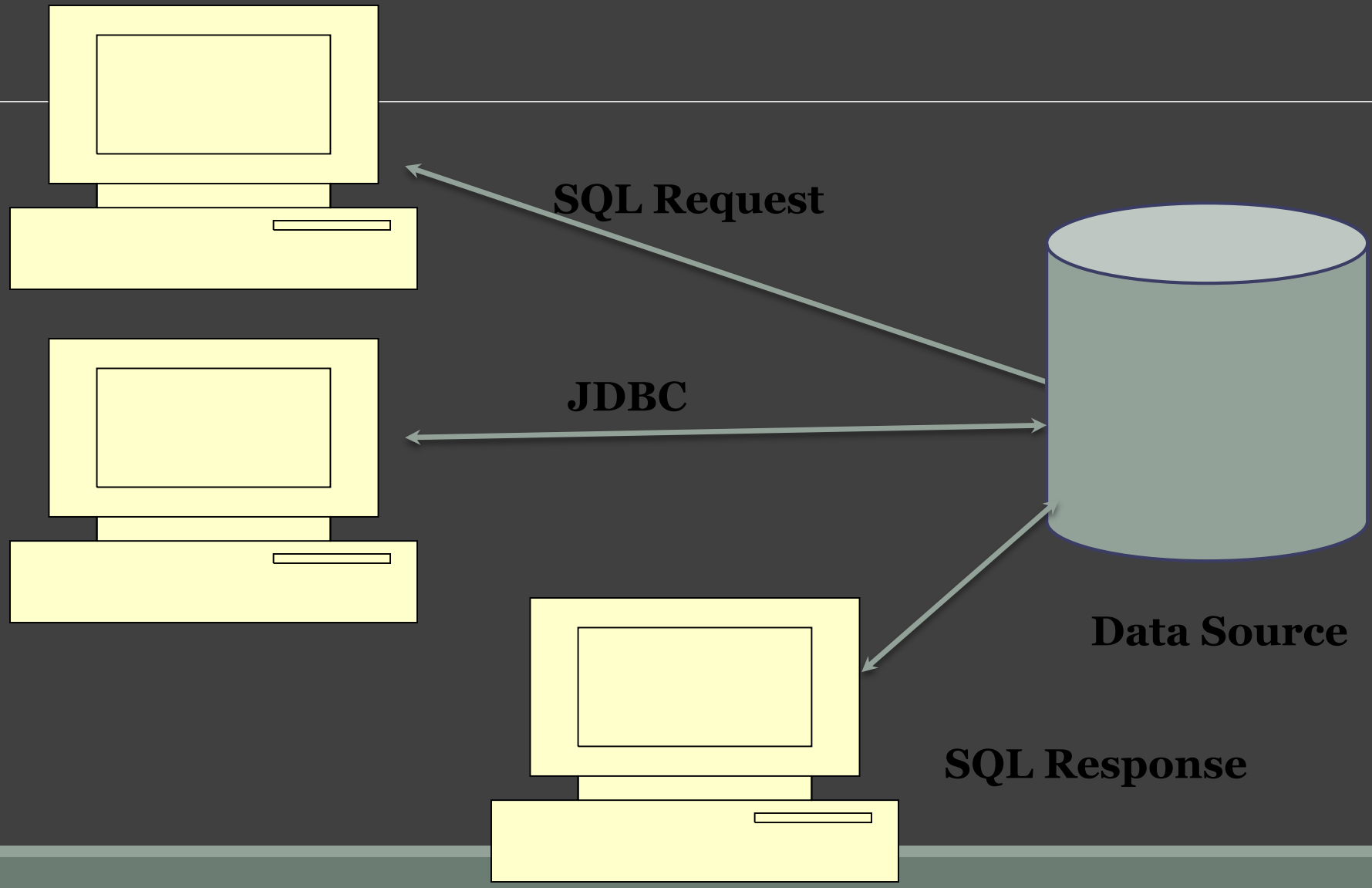
Advantages:

- Any changes made in data access logic will not affect the presentation and business logic.
- With two tier architecture it is easy to develop an application.

Disadvantages:

- One of the disadvantages of two tier architecture is that the application is expected to support a limited number of users.
- The reason is that each client requires its own connection and each connection requires CPU and memory.
- As the number of connections increases, the database performance degrades.

Two tier architecture



Three Tier Architecture

Application which is divided into three tier client tier, middle tier and database (Enterprise Information System) tier is known as **three tier architecture.**

Logic physically separates the business rules.

The presentation layer runs on client machine, application and business logic runs on J2EE Server and database logic there with database layer.

Thin Client

With the three tier architecture if the presentation manager resides only with the client tier then the client is called as thin client.

Presentation logic, application logic and business logic are with the business tier and database logic and database manager are with EIS tier.

Thick Client

With the three tier architecture if presentation manager, presentation logic, application logic reside with the client tier then the client is called as thick client.

Business logic is only with the business tier.

The database logic and database manager are with EIS tier.

Advantages:

- It improves scalability since the application servers can be deployed on many machines.
- The database no longer requires a connection from every client – it only requires connections from a smaller number of application servers.
- It provides better reusability because the same logic can be initiated from many clients or applications.
- It provides security because client does not have direct access to the database.

Disadvantages:

- It increases the complexity because to develop the three tier application is more difficult than developing a two tier application.

Introduction to Web Server

Web server is basically used to handle the Http request and Http response.

In J2EE web server is Apache Http Tomcat server to handle the client request and send the response to the client.

It also contains the dynamic web pages such as JSP, servlets and XML to handle such requests and responses.

HTTP

A browser can send request and other information.

That information can be parameters either by embedding them in the URL or by sending a data stream with the request.

This suggests that an HTTP request can be interpreted as a database query and query results can be used to build an HTML document dynamically.

HTTP consists of set of commands written as lines of ordinary ASCII values.

When we use a web browser, we do not enter HTTP command directly instead when we type URL or click a hyperlink, the browser actions into the HTTP commands that request the document from the server specified in URL.

HTTP as a state less protocol

HTTP protocol is known as stateless protocol because when client requests to the web server and once server has sent response to the client, it forgets client's identity, next time when client again makes a request to the web server at the time web server cannot recognize that client it means web server does not have any information related to client request and response which are made in the past.

But sometimes it is necessary to keep conversation state with the client across the multiple requests.

For that we can use some mechanism that store and maintain conversation between client and server.

For ex, HTTP SESSION, COOKIE etc.

N – Tier Architecture

An application which is divided in to more than three tiers can be called N – Tier architecture.

In N – tier architecture it is not decided how many tiers can be there, it depends on computing and network hardware on which application is deployed.

Basically it is divided into four application layers; client tier, web tier, business (EJB) tier and database tier.

Client Tier

Client tier consists of the user interface for user request and print the response.

Client tier runs on client machine.

It basically uses the browser or applet as client side application.

Web Tier

Web tier consists of the JSP and servlet dynamic web pages to handle the HTTP specific request logons, sessions, accesses the business services, and finally constructs a response and send it back to the client.

Business tier

Business tier consists of the business logic for the J2EE application.

For example, the EJB.

The benefit of having a centralized business tier is that same business logic can support different types of clients like browser, other standalone applications etc.

EIS Tier

EIS Tier consists of the DBMS/RDBMS.

It handles the users SQL request and generates appropriate response based on queries.

It is responsible for communicating with external resources such as legacy systems, ERP systems; messaging systems like MQSeries etc.

It also stores all persistent data in the database.

J2EE is based on N – tier or multi tier architecture applications.

J2EE makes easy to develop the Enterprise Application based on 2, 3 or more application layers.

Here it also proves that J2EE is distributing computing framework and multi – tiered application.

Advantages

Separation of user interface logic and business logic is done.

Business logic resides on small number of centralized machines.

Easy to maintain, to manage, to scale, loosely coupled etc.

Additional features can be easily added.

Disadvantages

It is having more complex structures and difficult to setup and maintain all the separated layers.

Enterprise Architecture of J2EE

Java to enterprise edition is basically developed for commercial projects and web solutions.

Business solutions for commercial project is solved using Multitiered Architecture.

The J2EE platform uses a multi – tiered distributed application model for enterprise applications.

By dividing application logic into the various components according to its task or function and the various application components that are gathered as a J2EE application are installed on

Different machines depending on the tier in the multi – tiered J2EE environment to which the application component belongs.

Following tiers are available in J2EE:

- Client – tier -> Client Machine
- Web – tier -> J2EE server.
- Business – tier -> J2EE server.
- Enterprise Information System (EIS) – tier -> EIS server (DB).

J2EE architecture is divided into 3 or 4 tier, it is known as Multi – tiered architecture.

But actually those 4 tiers are divided into the 3 locations:

- Client Machine
- J2EE Server
- EIS Server

Editions of JAVA

Mainly there are 3 editions in Java:

- J2SE: stands for Java 2 Standard Edition and is normally used for developing desktop applications.
- J2EE: stands for Java 2 Enterprise Edition for applications which run on servers, for example websites.
- J2ME: stands for Java 2 Micro Edition for applications which run on small scale devices like cell phones, for example games.

What is J2EE?

J2EE is an open standard which is provided by Sun Microsystems for applications which run on servers.

It provides multi – tiered architecture for commercial applications.

It includes J2SE + most of the other Java technologies including JavaMail, Activation, Servlets, JSF, JMS, EJB, and others.

Most of the APIs are component – oriented.

They are used to provide interfaces for business oriented components for enterprise and distributed internet applications.

J2EE is a multitier architecture consisting of the client tier, web tier enterprise java beans tier and enterprise information system.

Two or more tiers can physically reside on the same JVM although each tier provides a different type of functionality as required by the J2EE application.

A J2EE application accesses only those tiers whose functionality is required by the J2EE application.

Client Tier

The Client tier consists of programs that prompt the user for input and then convert into request.

The request is then forwarded to software on a component that processes the request and returns results to the client program.

Web Tier

The Web Tier provides internet functionality to a J2EE application.

Components that operate on the web tier use HTTP to receive requests from and send requests to clients that could reside on any tier.

EJB Tier

The EJB tier consists of the business logic for J2EE applications.

It is here where one or more EJBs reside each encoded with business rules that are called upon indirectly by clients.

The EJBs tier is the keystone to every J2EE application because it enable multiple instances of an application to concurrently access business logic and data so as not to slow down the performances.

EIS Tier

The EIS links a J2EE application to resources and legacy systems that are available on the corporate backbone network.

It is on the EIS where a J2EE application directly interfaces with DBMS.

J2EE APIs

The J2EE platform provides a set of APIs to develop the different types of applications.

It uses the concept of distributed computing to develop the enterprise application.

Java Database Connectivity(JDBC)

This API is useful to connect our java database application with any relational database.

The java application can access data from well known databases like ORACLE, MySQL and MSACCESS.

Remote Method Invocation over Internet Inter ORB Protocol (RMI)

This API is useful to access object methods running on different machines.

Enterprise Java Beans (EJB)

This API is useful to define server side components.

The J2EE supports components based application development using EJB.

Java Servlets (JS)

This java servlets API provides object oriented abstractions for building dynamic web applications.

Java Server Pages (JSP)

The Java server pages API provides easy way for building dynamic web applications.

JSF (Java Server Faces)

It is an API for building component based user interface for web application.

It is a part of Java EE.

It is also web MVC framework.

Java Messaging Service (JMS)

JMS provides a java API for message passing and publish and subscribe the types of message oriented middleware services.

Java Naming Directory Interface (JNDI)

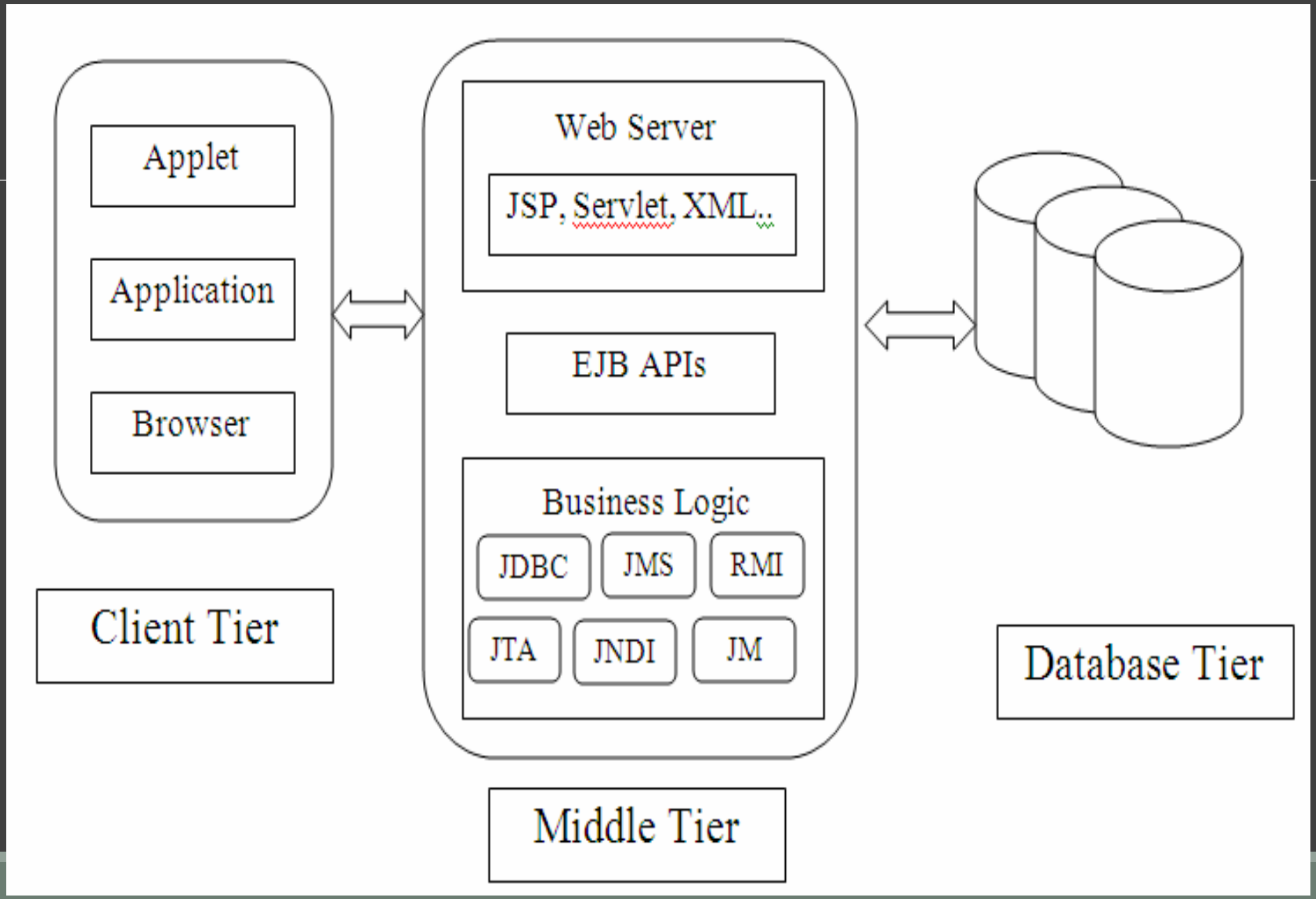
The JNDI API standardizes access to different types of naming and directory services.

Java Transaction API (JTA)

This API is for implementing distributed transaction application.

Java Mail (JM)

This API provides a platform independent and protocol independent framework to build java based mail application.


Applet
Application
Browser
Client Tier
Web Server
JSP, Servlet, XML..
EJB APIs
Business Logic
JDBC
JMS
RMI
JTA
JNDI
JM
Middle Tier
Database Tier

Containers

There are two types of clients; one is “Thin Client” and other is “Thick Client”.

Generally, thin client in multi – tiered architecture and it is hard to manage.

But using component based platform independent architecture of J2EE we can easily write J2EE application because reusable components are stored at business logic and easily shared within the whole application.

For all these components server provides services in the form of containers.

Because of these containers we need not write services for each component included in our application.

“Containers are interface between a component and client (Lower Level Code) or platform oriented functionality which supports component. Container provides communication platform between client and components.”

Types of container

There are four basic containers in J2EE.

In that two server containers and two client containers are listed below.

- Applet Container
- Application Container
- EJB Container
- Web Container

Applet Container

Applet container is a client container which is used to manage an execution of the applet on the browser.

Application container

Application container is a client container which is used to manage application client and their components.

EJB container

EJB container is a server container which is used to manage enterprise beans components.

Web container

Web container is also a server container which is used to manage execution of JSP pages and servlet components.