



# **J2EE Architecture**

---

Dr.R.Saranya



# History

---

- Initially two tier architecture (client server applications)
- Client is responsible for data access applying business logic and presentation of data
- Only service provided by Server was that of database server.

# Two Tier Application Architecture





# History

---

- Two – Tier – Drawbacks :
  - Easy to deploy but difficult to enhance or upgrade.
  - It makes reuse of business and presentation logic difficult
  - Not scalable and not suited for internet



# History

---

- Three – Tier Architecture

Layers are

- 1) Client layer**
- 2) Business layer**
- 3) Data layer**



# Three – Tier Architecture

---

## **Client layer:**

- It is also called as *Presentation layer* which contains UI part of our application.
- E.g. Designing registration form which contains text box, label, button etc.

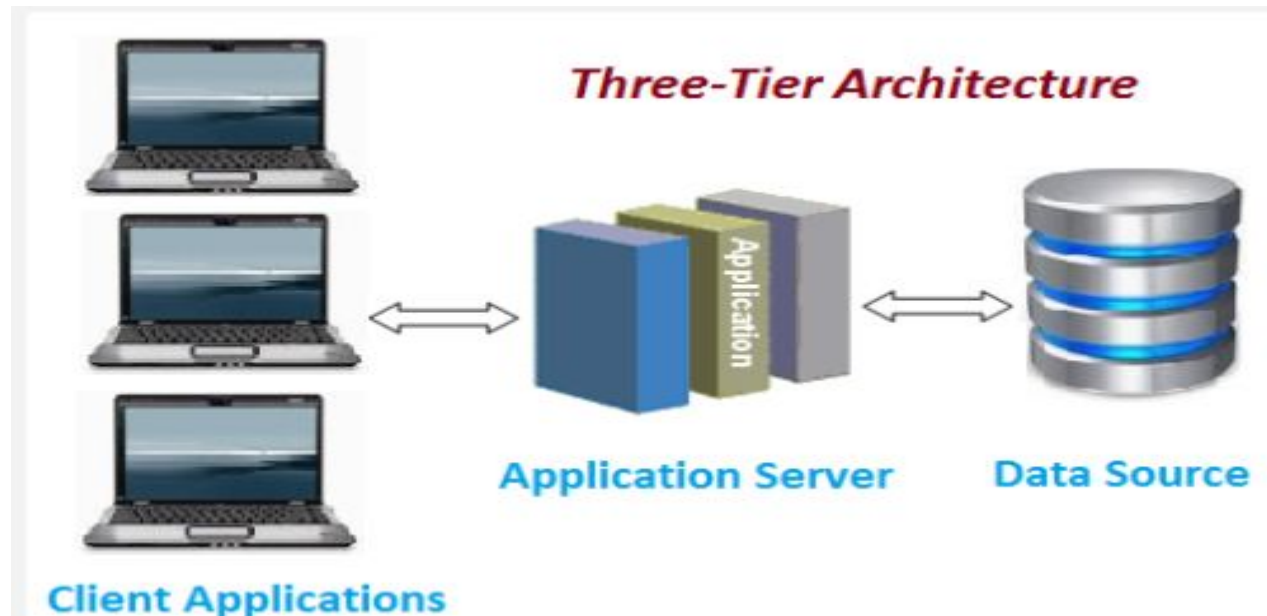
## **Business layer:**

- In this layer all business logic written like validation of data, calculations, data insertion etc.
- This layer is also called the intermediary layer helps to make communication faster between client and data layer.

## **Data layer:**

Data Access Layer contains methods to connect with database and to perform insert, update, delete, get data from database based on our input data.

# Three – Tier Architecture





# Three – Tier Architecture

---

## **Advantages**

- High performance, lightweight persistent objects
- Scalability – Each tier can scale horizontally
- High degree of flexibility in deployment platform and configuration
- Better Re-use
- Improve Data Integrity and Security

## **Disadvantages**

- Increase Complexity/Effort



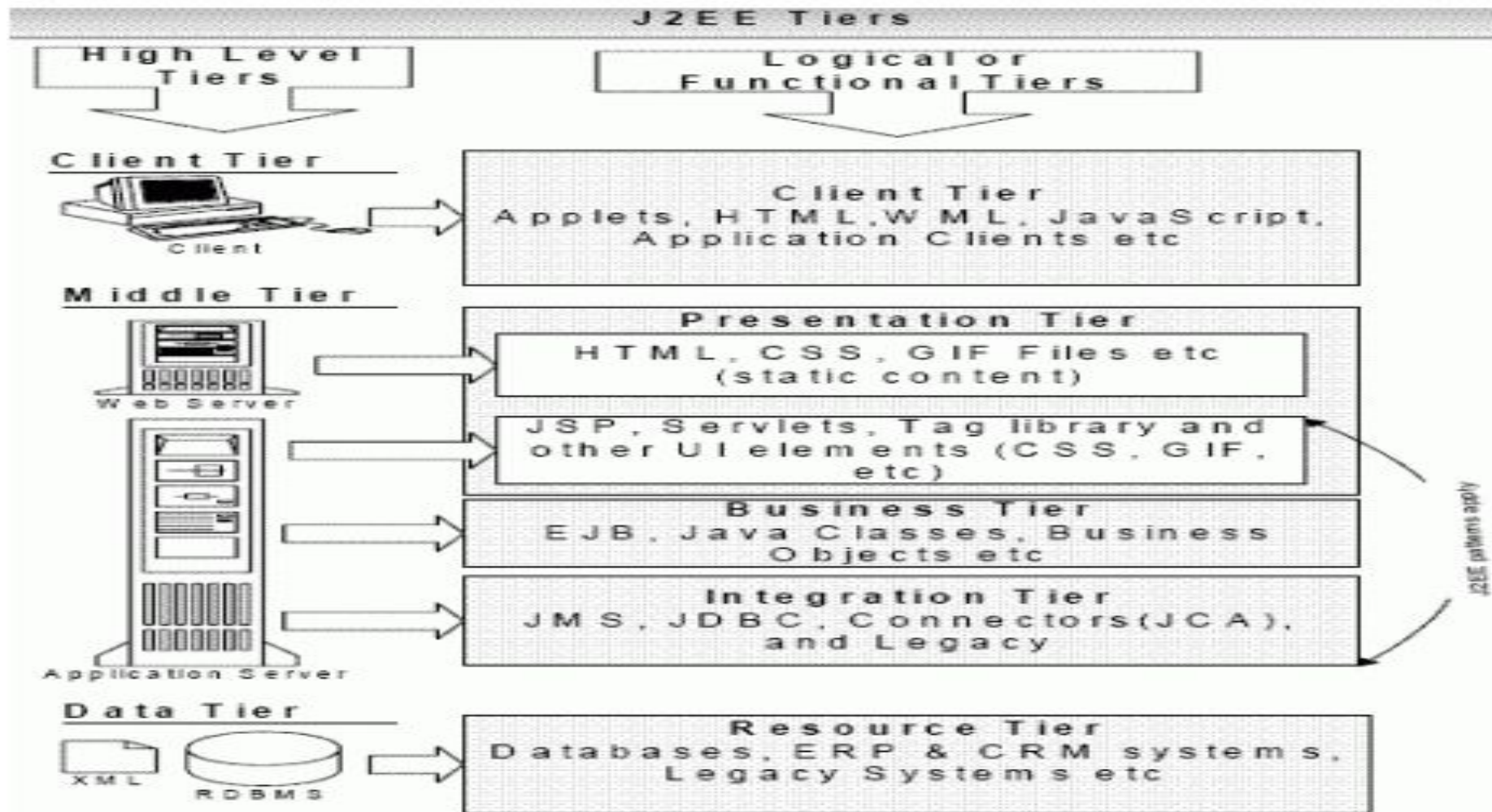


# Java 2 Platform Enterprise Edition(J2EE) Architecture

---

- The J2EE platform uses a **multi-tiered or n-tiered** distributed application model for web based, enterprise applications.
- It is a public specification that embodies several technologies
- Current version is 1.3
- **Advantage of a 3-tiered or n-tiered application - force separation** among presentation logic, business logic and database logic. **Key benefits:**
  - (i) **Manageability**
  - (ii) **Scalability:** More hardware can be added and allows clustering (i.e. horizontal scaling).
  - (iii) **Maintainability.**
  - (iv) **Availability:** Clustering and load balancing can provide availability.
  - (v) **Extensibility**

# J2EE Architecture





# J2EE Architecture

---

- **Client tier** represents Web browser, a Java or other application, Applet, WAP phone etc. The client tier makes requests to the Web server who will be serving the request by either returning static content if it is present in the Web server or forwards the request to either Servlet or JSP in the application server for either static or dynamic content.
- **Presentation tier** encapsulates the presentation logic required to serve clients. A Servlet or JSP in the presentation tier intercepts client requests, manages logons, sessions, accesses the business services, and finally constructs a response, which gets delivered to client.



# J2EE Architecture

---

- **Business tier** provides the business services. This tier contains the business logic and the business data. All the business logic is centralized into this tier as opposed to 2-tier systems where the business logic is scattered between the front end and the backend. The benefit of having a centralized business tier is that same business logic can support different types of clients like browser, WAP, other stand-alone applications etc.
- **Integration tier** is responsible for communicating with external resources such as databases, legacy systems, ERP systems, messaging systems like MQSeries etc. The components in this tier use JDBC, JMS, J2EE Connector Architecture (JCA) and some proprietary middleware to access the resource tier.

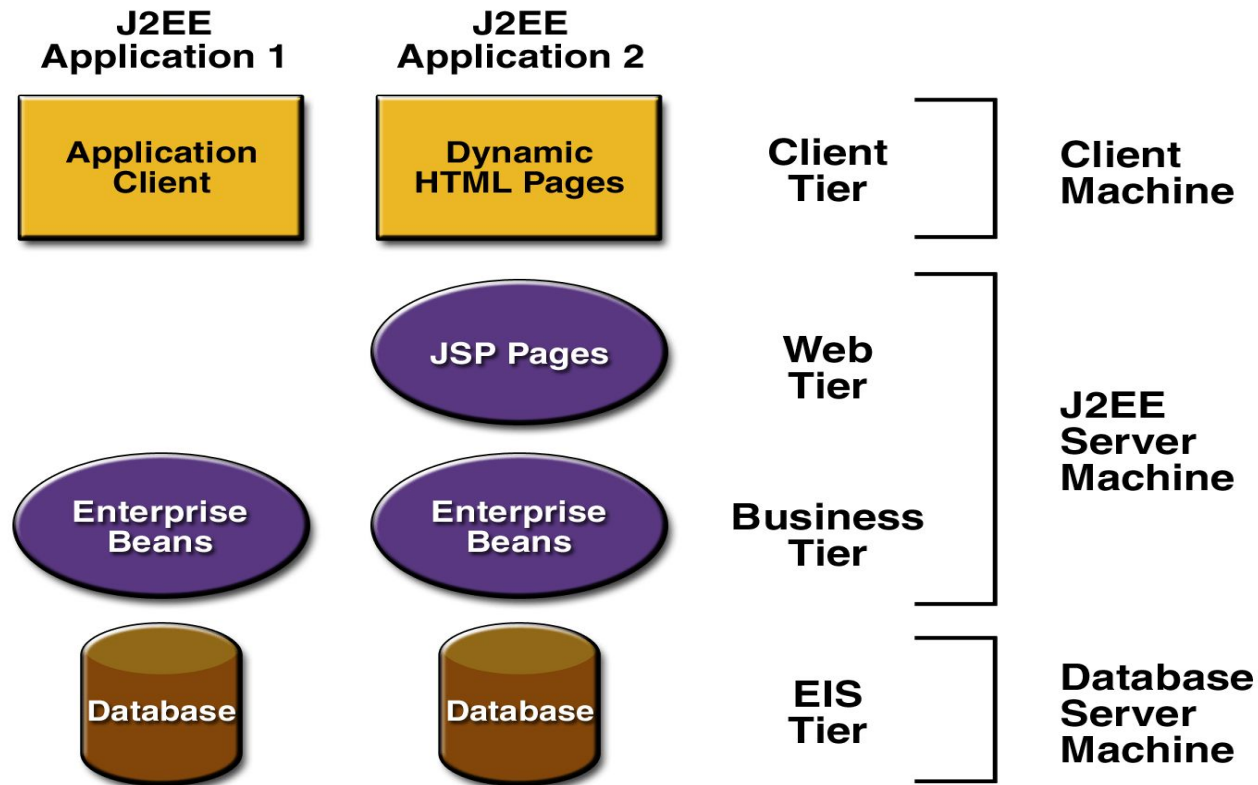


# J2EE Architecture

---

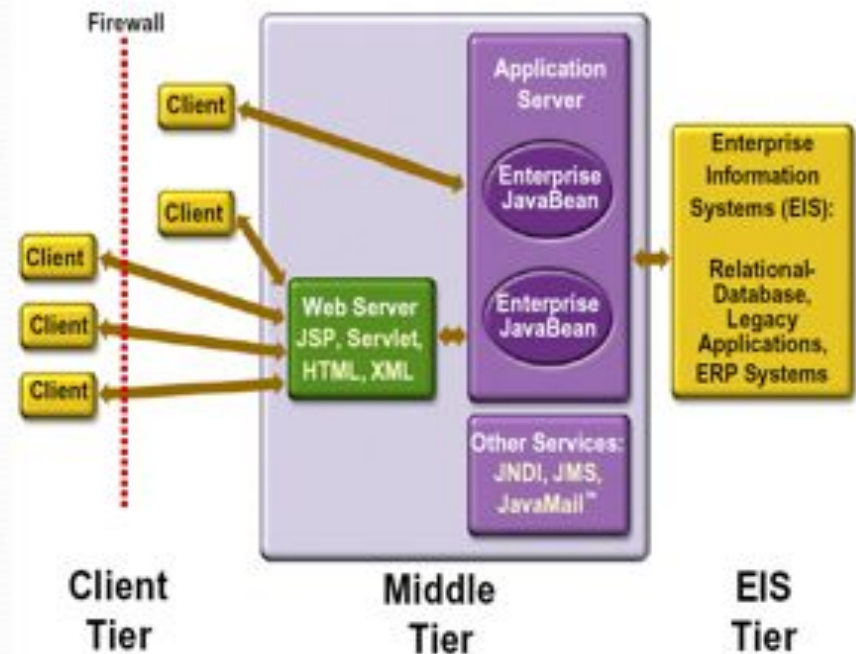
- **Resource tier** is the external resource such as a database, ERP system, Mainframe system etc responsible for storing the data. This tier is also known as Data Tier or EIS (Enterprise Information System) Tier.

# J2EE Architecture



# J2EE Architecture

Three-tiered applications that run in this way extend the standard two-tiered client and server model by placing a multithreaded application server between the client application and back-end storage





# J2EE Features and Goals

---

- Component based model
- Container provided services
- Highly scaleable
- Simplified Architecture
- Flexible security model
- Robustness
- Scalability
- Simplicity
- Maintainability
- Testability
- Reusability





# J2EE Components & Containers

---

- **Primary Technologies**

- Servlets
- JavaServer Pages (JSP)
- Enterprise JavaBeans (EJB)

**Container** - The application server maintains control and provides services through an interface or framework known as a *container*



# J2EE Containers

---

- Three of these are server-side containers:
  - The server itself, which provides the J2EE runtime environment and the other two containers
  - An EJB container to manage EJB components
  - A Web container to manage servlets and JSP pages
- The other two container types are client-side:
  - An application container for stand-alone GUIs, console
  - An applet container, meaning a browser, usually with the Java Plug-in

# J2EE Containers

