Introduction to JEE Standards

IGATE
Speed.Agility.Imagination

## The Java 2 Platform

- **Platform introduced June, 1999**
- **JSE – Java Standard Edition**
  - Java for the desktop / workstation
  - http://java.sun.com/javase/index.jsp
- **JME – Java Micro Edition**
  - Java for the consumer device
  - http://java.sun.com/javame/index.jsp
- **JEE - Java Enterprise Edition**
  - Java for the server
  - http://java.sun.com/javaee/index.jsp





J2SE 1.4 just shipped in February!

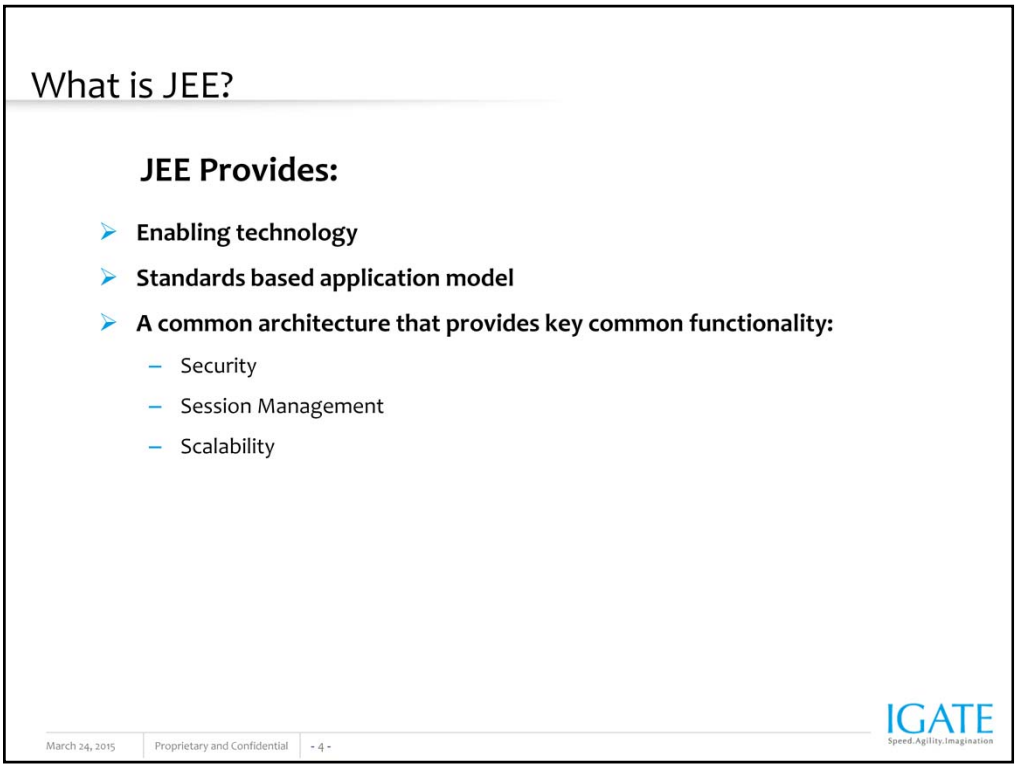
## What is JEE?

**What is Java Enterprise Edition?**

***"The Java platform, Enterprise Edition reduces the cost and complexity of developing ... multi-tier services, resulting in services that can be rapidly deployed and easily enhanced"***



IGATE
Speed.Agility.Imagination

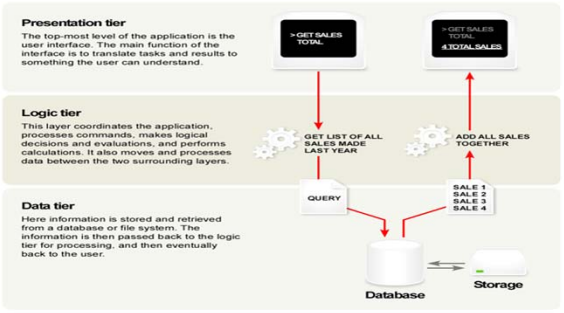

What is JEE?
JEE Provides:
Enabling technology
Standards based application model
A common architecture that provides key common functionality:
Security
Session Management
Scalability
March 24, 2015
Proprietary and Confidential
- 4 -
IGATE
Speed.Agility.Imagination

## What is JEE?

### What does "multi-tier" mean?

- In software engineering, multi-tier architecture (often referred to as *n-tier architecture*) is a client-server architecture in which an application is executed by more than one distinct software agent.
- For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of "multi-tier architecture" refers to *three-tier architecture*

## Three tier application



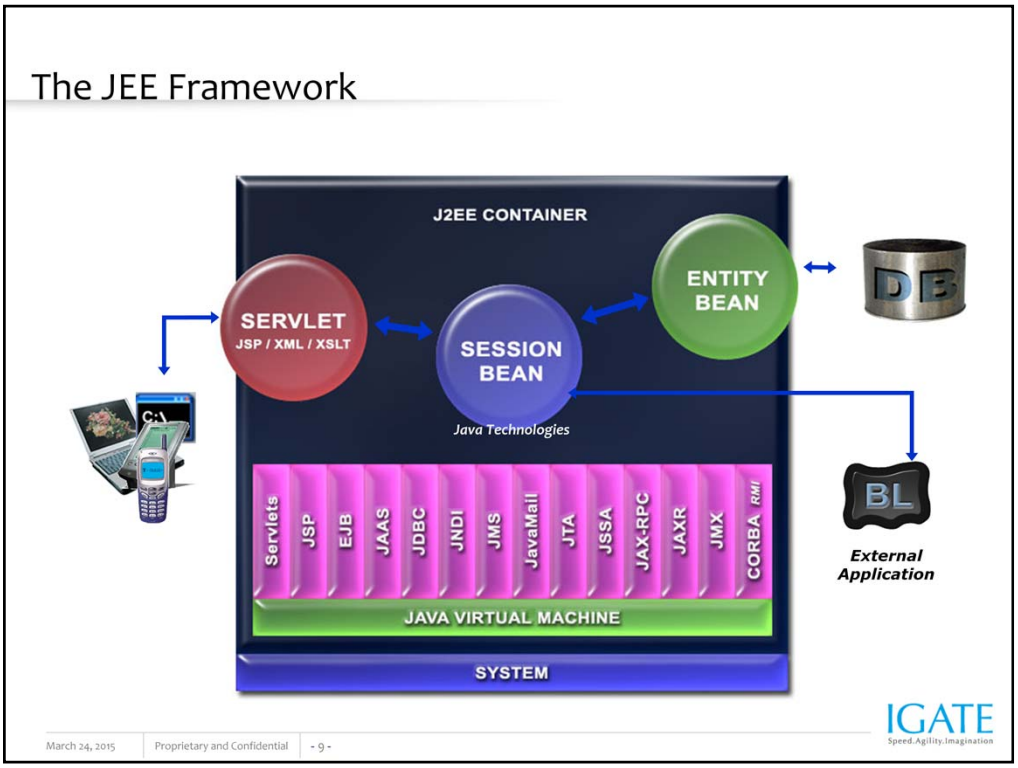

- Thin clients make easy upgrades
- Centralized application server management is easier to manage
- Model allows for horizontal scalability

## Three tier application

- **Presentation Tier:**
  - This is the top most level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network
- **Application Tier/Logic Tier/Business Logic Tier**
  - The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing

## Three tier application

- **Data Tier**
  - This tier consists of Database Servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.


The JEE Framework
J2EE CONTAINER
SERVLET
JSP / XML / XSLT
SESSION BEAN
ENTITY BEAN
DB
Java Technologies
BL
External Application
Servlets
JSP
EJB
JAAS
JDBC
JNDI
JMS
JavaMail
JTA
JSSA
JAX-RPC
JAXR
JMX
CORBA
RMI
JAVA VIRTUAL MACHINE
SYSTEM
March 24, 2015
Proprietary and Confidential
- 9 -
IGATE
Speed.Agility.Imagination

## Some of the JEE Technologies

- Java Servlets – Web-server side web-programs
- JSP- Web-server side programs , on top of servlets
- EJB –Application server side enterprise components
- JMS- Java Message Service
- JDBC-Backend database connectivity
- JNDI - Provides naming and directory functionality
- JTA / JTS –Provides transaction API and service
- JavaMail – API that models email system
- JAAS – Provides for security infrastructure
- JAXP – provides for XML support

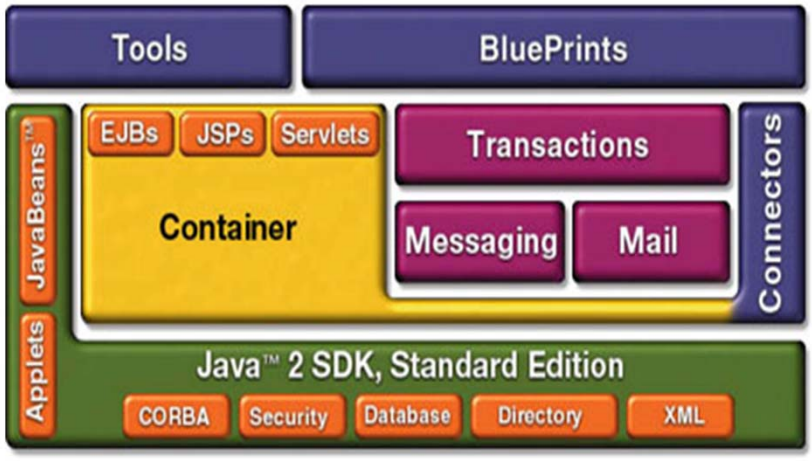
## Some of additional Technologies

- **RMI – Provides for invocation of methods on remote Java objects. Java RMI is included with Java SE**
- **CORBA –It is the open standard for distributed computing across heterogeneous platforms and languages**



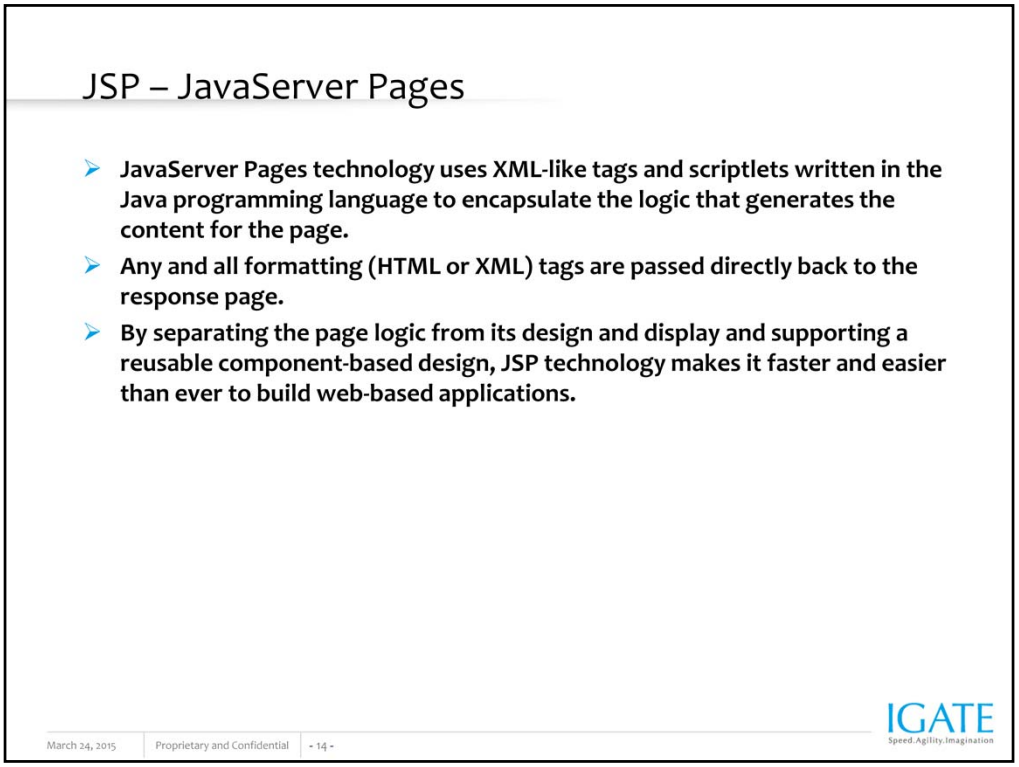
IGATE
Speed.Agility.Imagination

## JEE Components

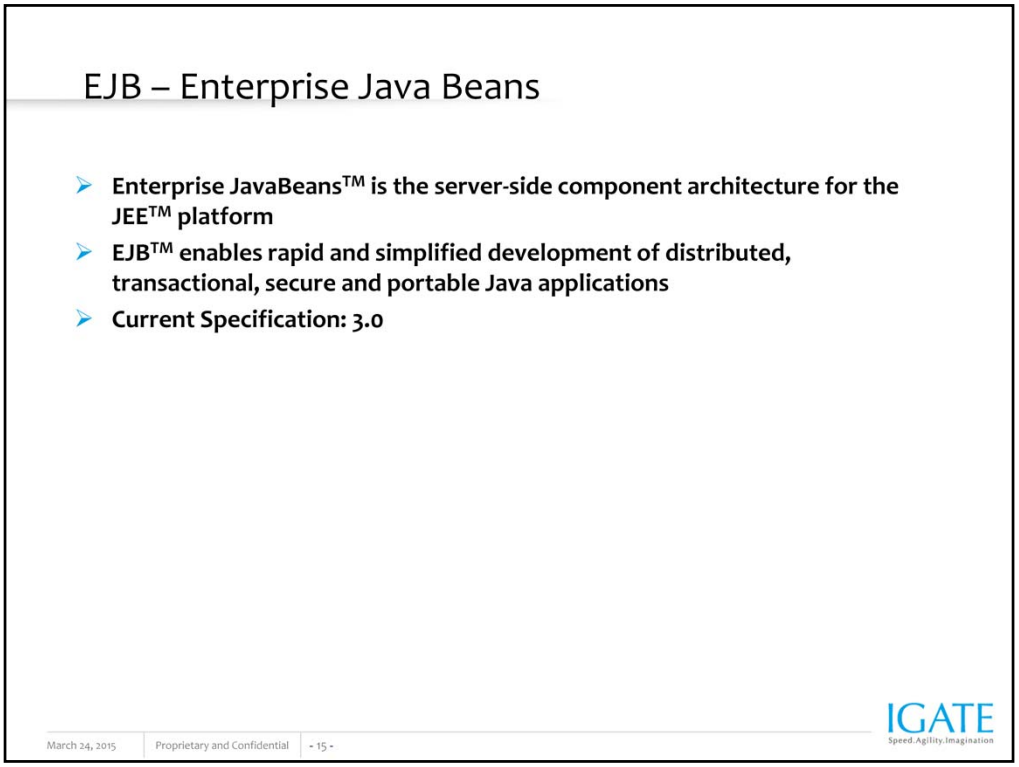


## Java Servlets

- Servlets are the Java platform technology of choice for extending and enhancing web servers.
- Servlets provide a component-based, platform-independent method for building web-based applications, without the performance limitations of CGI programs.
- Servlets have access to the entire family of Java APIs, including the JDBC™ API to access enterprise databases.
- Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection




Programs developed in Procedural languages such as C become too difficult reuse and maintain as the size of the application grows. So, a new approach called "OOP" was introduced, which amongst other things offered the concept of Objects. C++ became one of the most common languages (but not the first language) to offer OOP features, and got popularity. Later, need was felt by Sun Microsystem for a language, that follows the paths of C and C++, but with many more features (discussed on other slides). That gave birth to a language called "Java"




Java is completely object oriented. In C++, being more compatible to C, allows code to exist outside classes too, but in Java, every line of code has to belong to some or other class. Thus it is closer to true object oriented language.

Java is simpler than C++, since concepts of pointers or multiple inheritance is not there.

## EJB – Enterprise Java Beans

- **Enterprise Java Beans are components that are deployed into containers**
- **The container provides services**
  - Loading / Initialization Transactions
  - Persistence Communication with EJB clients
  - Enterprise Naming Context (JNDI name space)
- **Types of Enterprise Beans**
  - Entity Beans
    - Container Managed Persistence (CMP)
    - Bean Managed Persistence (BMP)
  - Session Beans
    - Stateless Session Beans
    - Statefull Session Beans.
  - Message Beans

## EJB – Entity Beans

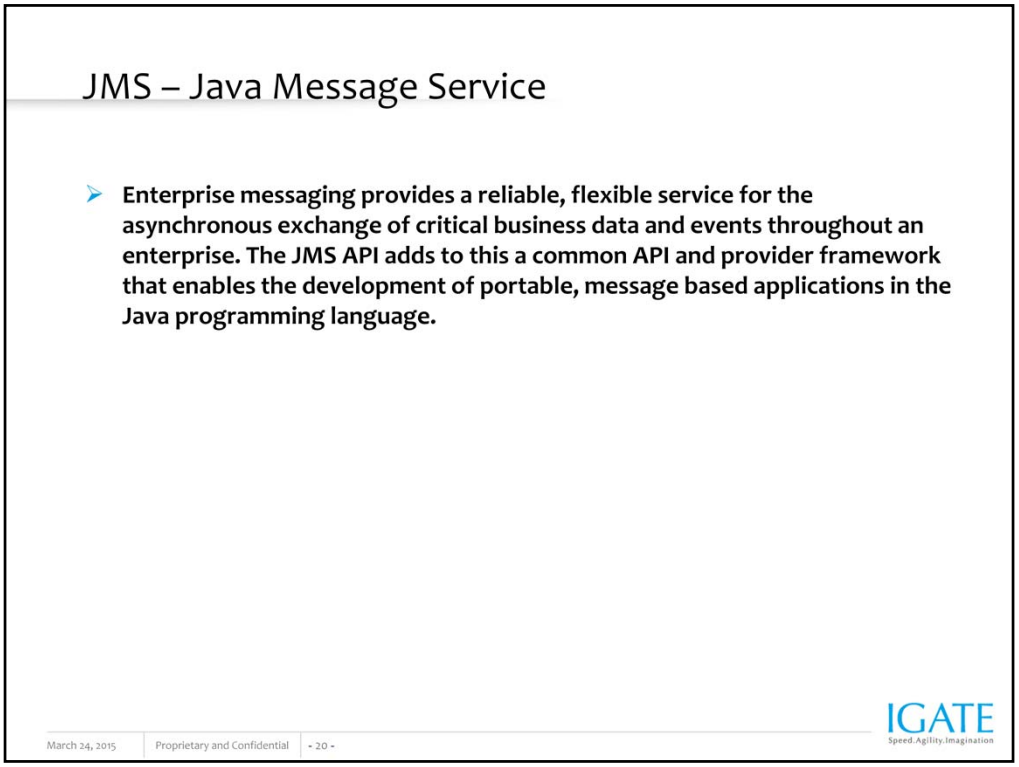
- **Entity beans are classes that map to individual entities – typically, an Entity bean references a row in a database table, providing an object representation of that database object.**
- **Entity beans provide an abstraction layer so that working with the entity is not specific to the storage mechanism for that entity.**
- **Container Managed Persistence (CMP)**
  - The EJB container automatically persists the EJB objects, usually to a relational database where each type of object is represented as a table, and each instance of the object is a row in that table
- **Bean Managed Persistence (BMP)**
  - The EJB container calls bean methods when it is appropriate for the bean to load, save or update data, enforcing transactions without transaction code written by the bean developer

## EJB – Session Beans

- Session beans perform work for a client application
- Stateful – A stateful bean maintains a conversational state with a client. The client perceives that it is only talking to one bean, and that bean maintains information between calls
- Stateless – A stateless bean maintains no client information between method calls – the container can substitute beans as necessary between method calls

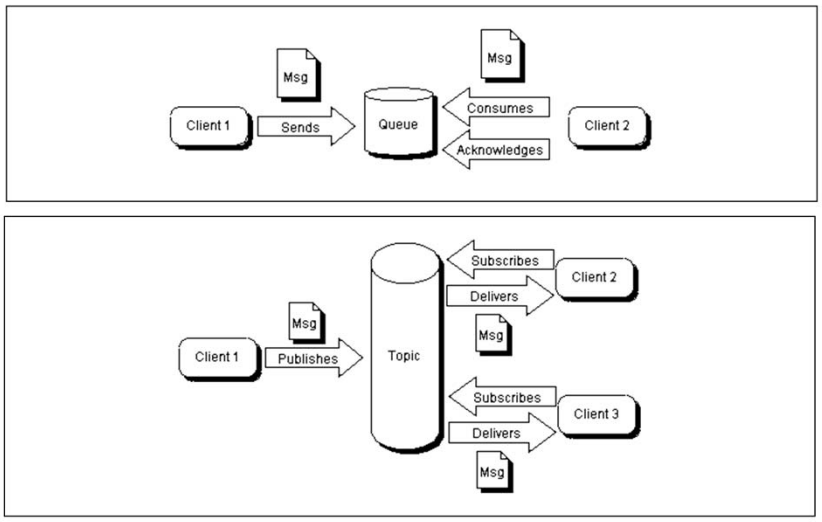
## EJB – Message Beans

- **Message beans are classes that receive asynchronous notification from a Java Message Service server**




It should be mentioned to the participant that current version of Java is 1.5, although not covered as part of this training.

## JMS – Java Message Service

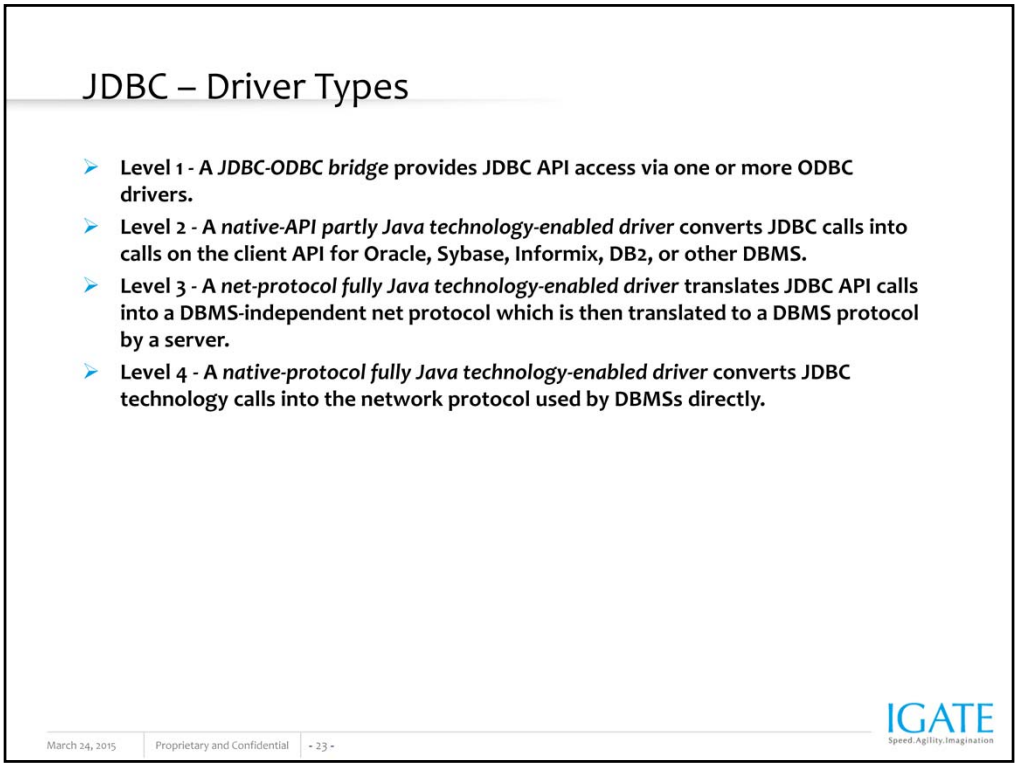





IGATE
Speed.Agility.Imagination

## JDBC – Data Access API

- **JDBC™ technology is an API that lets you access virtually any tabular data source from the Java™ programming language.**
  - Cross-DBMS connectivity to a wide range of SQL databases
  - Access to other tabular data sources, such as spreadsheets or flat files.

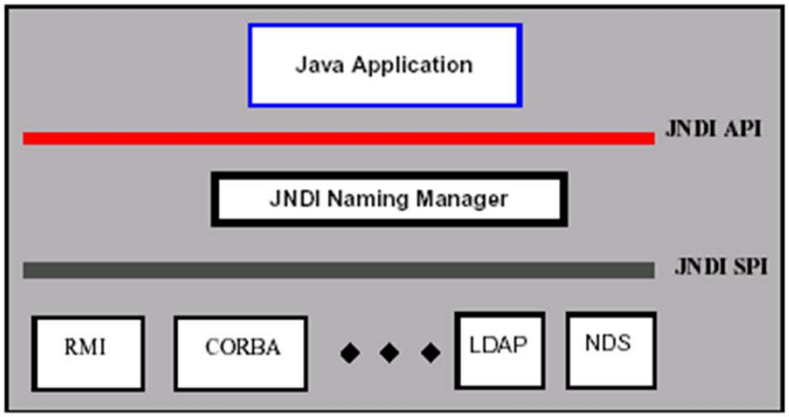



It should be noted that JRE is re-distributable but JDK is not re-distributable. JRE can be bundled along with your own software that you intend to sell commercially too. However, one can't bundle JDK along with one's own software. JDK is freely downloadable, but not freely redistributable.

## JNDI – Java Naming and Directory Interface

- JNDI is an API specified in Java[tm] that provides naming and directory functionality to applications written in Java. It is designed especially for Java by using Java's object model.
- Using JNDI, Java applications can store and retrieve named Java objects of any type.
- JNDI provides methods for performing standard directory operations, such as associating attributes with objects and searching for objects using their attributes.
- JNDI allows Java applications to take advantage of information in a variety of existing naming and directory services, such as LDAP, NDS, DNS, and NIS(YP), and allows Java applications to coexist with legacy applications and systems.

## JNDI - Layers


Java Application
JNDI API
JNDI Naming Manager
JNDI SPI
RMI
CORBA
LDAP
NDS

## JTA / JTS – Transactions

- The Java Transaction API (JTA) and the Java Transaction Service (JTS) allow J2EE application servers to take the burden of transaction management off of the component developer.
- Developers can define the transactional properties of Enterprise JavaBeans™ technology based components during design or deployment using declarative statements in the deployment descriptor.
- The application server takes over the transaction management responsibilities.

## JavaMail

The JavaMailTM 1.2 API provides a set of abstract classes that model a mail system.

The API provides a platform independent and protocol independent framework to build Java technology-based mail and messaging applications.

J2EE contains JAF – JavaBeans Activation Framework since it is required by JavaMail

Supports common mail protocols

- IMAP
- POP
- SMTP
- MIME

## JAAS – Java Authentication and Authorization Service

- ***Authentication*** **of users, to reliably and securely determine who is currently executing Java code, regardless of whether the code is running as an application, an applet, a bean, or a servlet; and**
- ***Authorization*** **of users to ensure they have the access control rights (permissions) required to do the actions performed.**



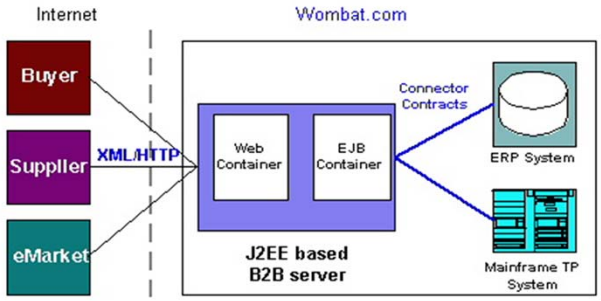
IGATE
Speed.Agility.Imagination

## JAXP for XML

- The Java™ API for XML Processing ("JAXP") supports processing of XML documents using DOM, SAX, and XSLT.
- The portability and extensibility of both XML and Java make them the ideal choice for the flexibility and wide availability requirements of this new web.
- J2EE 1.3 includes JAXP 1.1 support, as well as Servlet Filters and XML JSP™ documents.

## J2EE Connectors

- The J2EE Connector architecture defines a standard architecture for connecting the J2EE platform to heterogeneous EISs (Enterprise Information Systems)
- Examples of EISs include ERP, mainframe transaction processing, database systems, and legacy applications not written in the Java programming language






IGATE Speed.Agility.Imagination

## Remote Method Invocation (RMI)

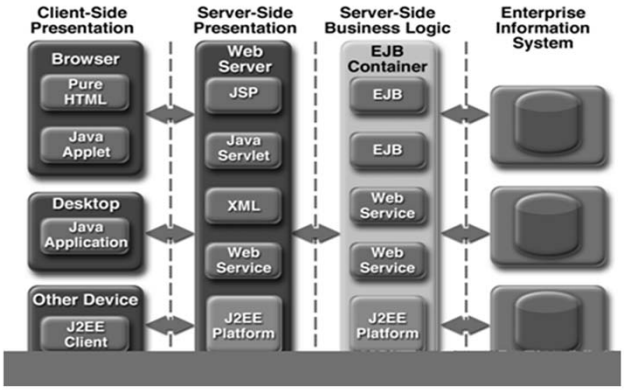
- Java RMI is included with Java SE
- Enables the programmer to create distributed Java technology-based to Java technology-based applications
- In these distributed applications the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts
- RMI uses object serialization to marshal and unmarshal parameters and does not truncate types, supporting true object-oriented polymorphism

## Common Object Request Broker Architecture (CORBA)

- CORBA technology is the open standard for distributed computing across heterogeneous platforms and languages
- CORBA complements the Java platform by providing a distributed objects framework, services to support that framework, and interoperability with other languages
- CORBA standards provide the proven, interoperable infrastructure to the Java platform.
- IIOP (Internet Inter-ORB Protocol) manages the communication between the object components that power the system

## The Platform for Enterprise Solutions

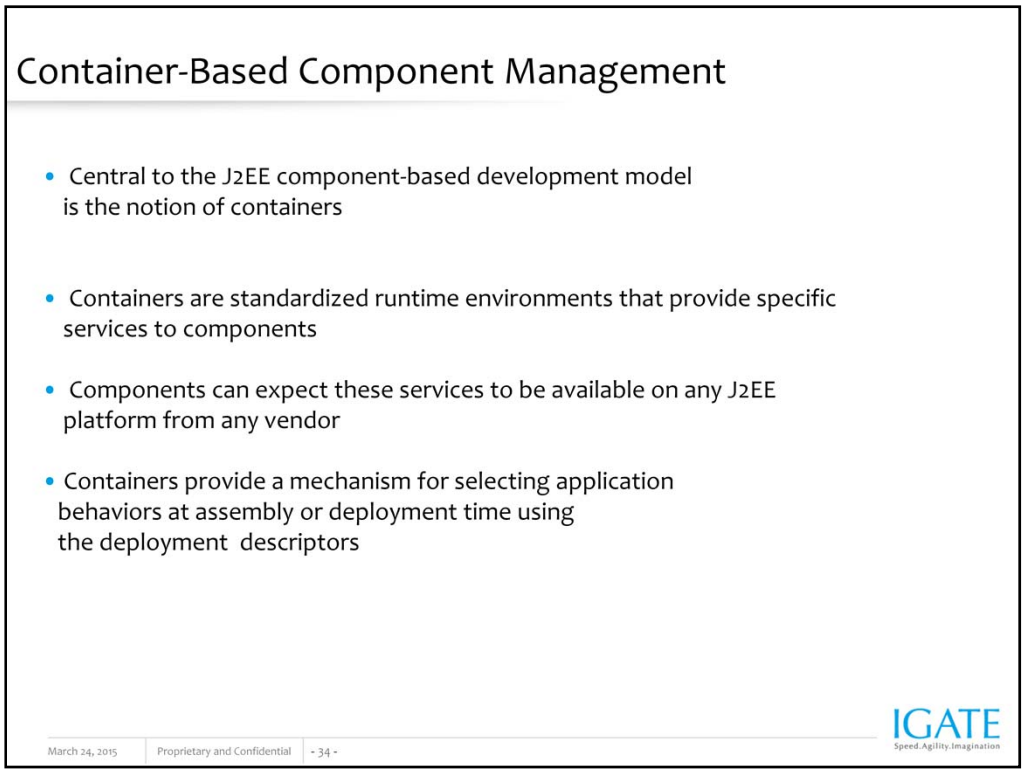
- Provides server-side and client-side support for developing distributed, multitier applications



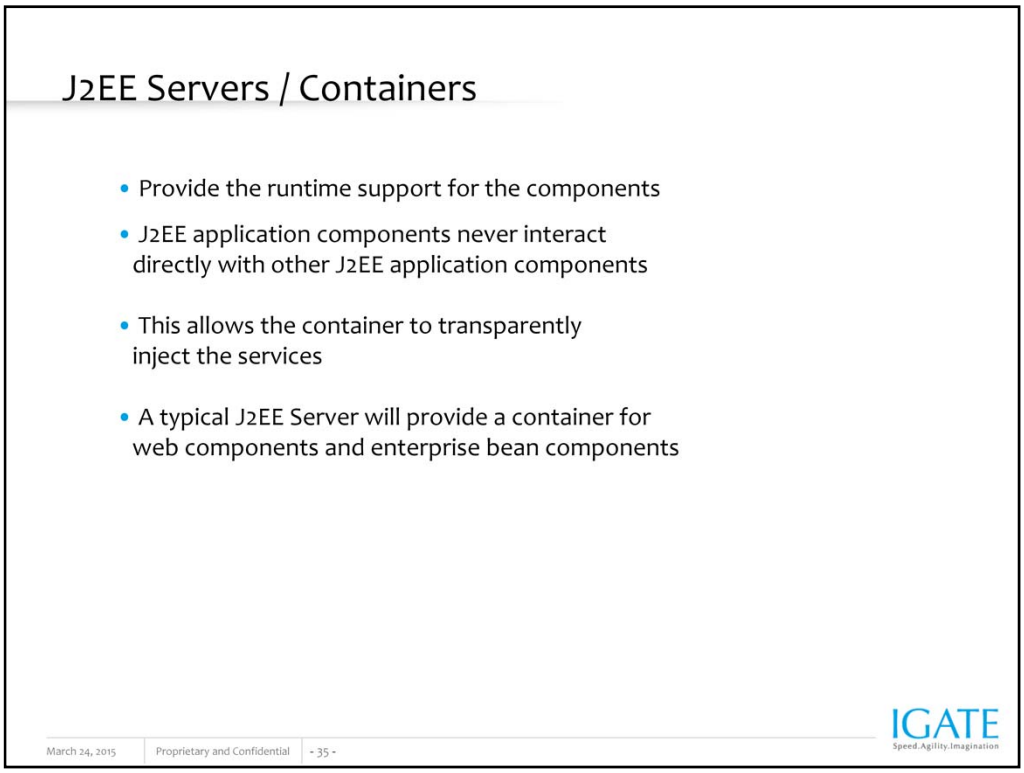



IGATE
Speed.Agility.Imagination

Multitier Model As illustrated, the J2EE platform provides a multitier distributed application model. This means that the various parts of an application can run on different devices. The J2EE architecture defines a client tier, a middle tier (consisting of one or more subtiers), and a back-end tier. The client tier supports a variety of client types, both outside and inside of corporate firewalls. The middle tier supports client services through Web containers in the Web tier and supports business logic component services through Enterprise JavaBeanstm (EJBtm) containers in the EJB tier. On the back end, the enterprise information systems in the EIS tier are accessible by way of standard APIs.



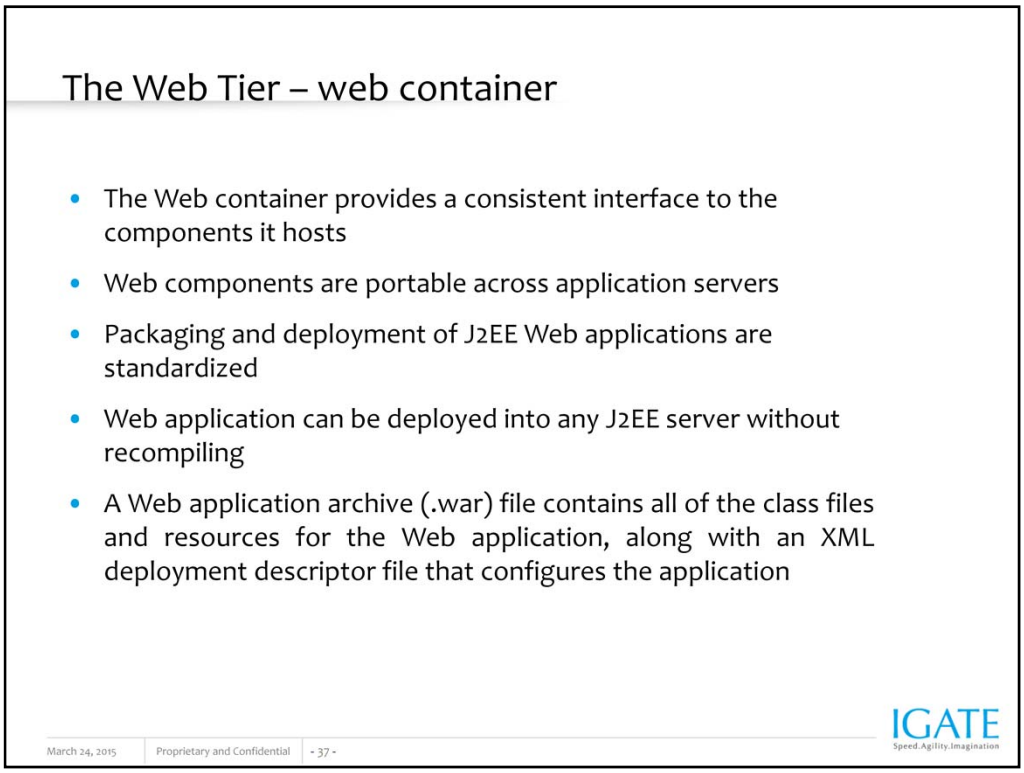

For example, all J2EE Web containers provide runtime support for responding to client requests, performing request-time processing (such as invoking JSP pages or servlet behavior), and returning results to the client. In addition, they provide APIs to support user session management. All EJB containers provide automated support for transaction and life cycle management of EJB components, as well as bean lookup and other services. Containers also provide standardized access to enterprise information systems; for example, providing access to relational data through the JDBC API.




Containers provide the runtime support for J2EE application components.
Containers provide a federated view of the underlying J2EE APIs to the application components.
J2EE application components never interact directly with other J2EE application components.
They use the protocols and methods of the container for interacting with each other and with platform services
Thus mediating between the application components and the J2EE services allows the container to transparently inject the services defined by the components' deployment descriptors, such as declarative transaction management, security checks, resource pooling, and state management
The container tools must understand the file formats for the packaging of application components for deployment.

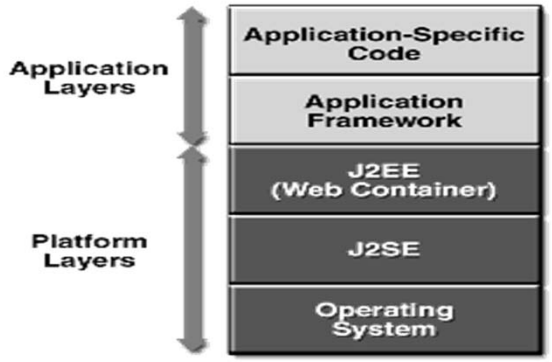
## The Web Tier – web container

- A J2EE Web application runs inside a J2EE server's Web container
- The Web container manages
  - each component's lifecycle
  - dispatches service requests to application components
  - provides standard interfaces to context data such as session state and information about the current request




A Web component is a software entity that provides a response to a request. A Web component typically generates the user interface for a Web-based application. The J2EE platform specifies two types of Web components: servlets and JavaServer PagesTM (JSPTM) pages.

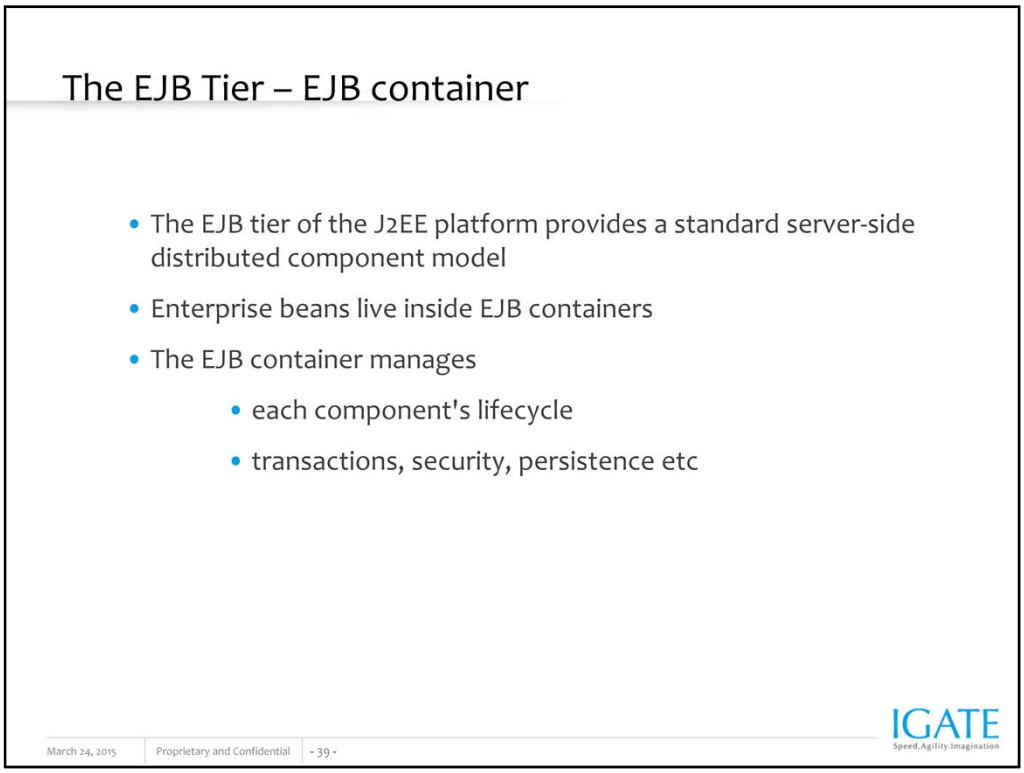
## Web-Tier Application Structure



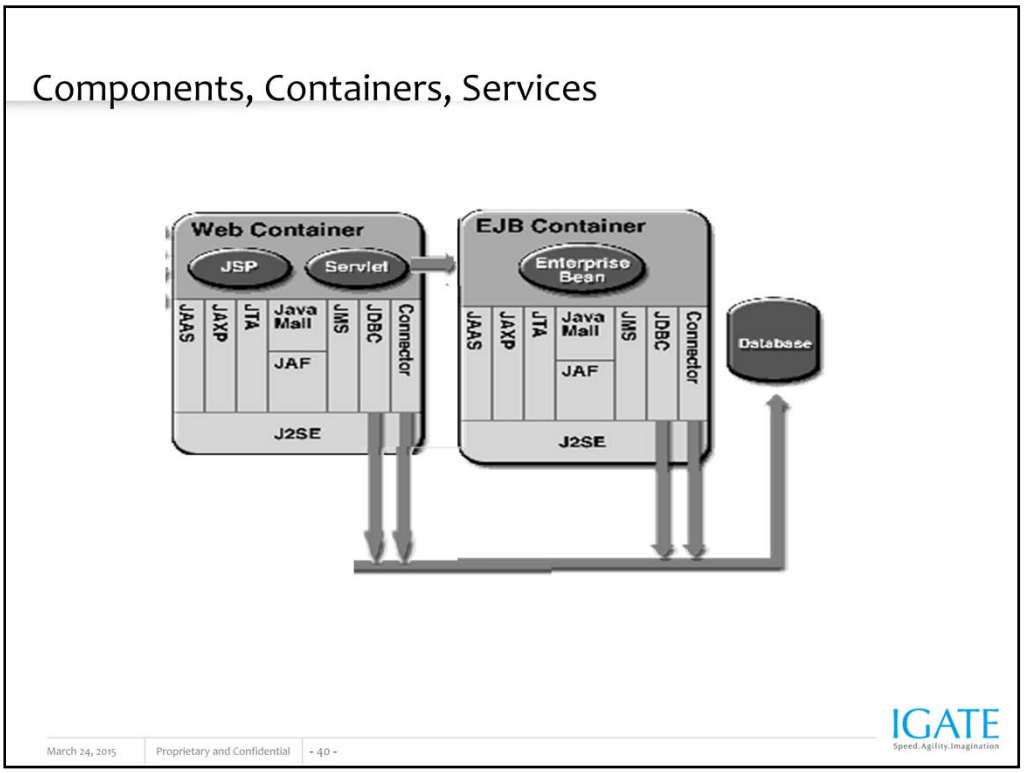

In the Web tier, a J2EE Web container provides services related to serving Web requests



A Web component is a software entity that provides a response to a request. A Web component typically generates the user interface for a Web-based application. The J2EE platform specifies two types of Web components: servlets and JavaServer PagesTM (JSPTM) pages.

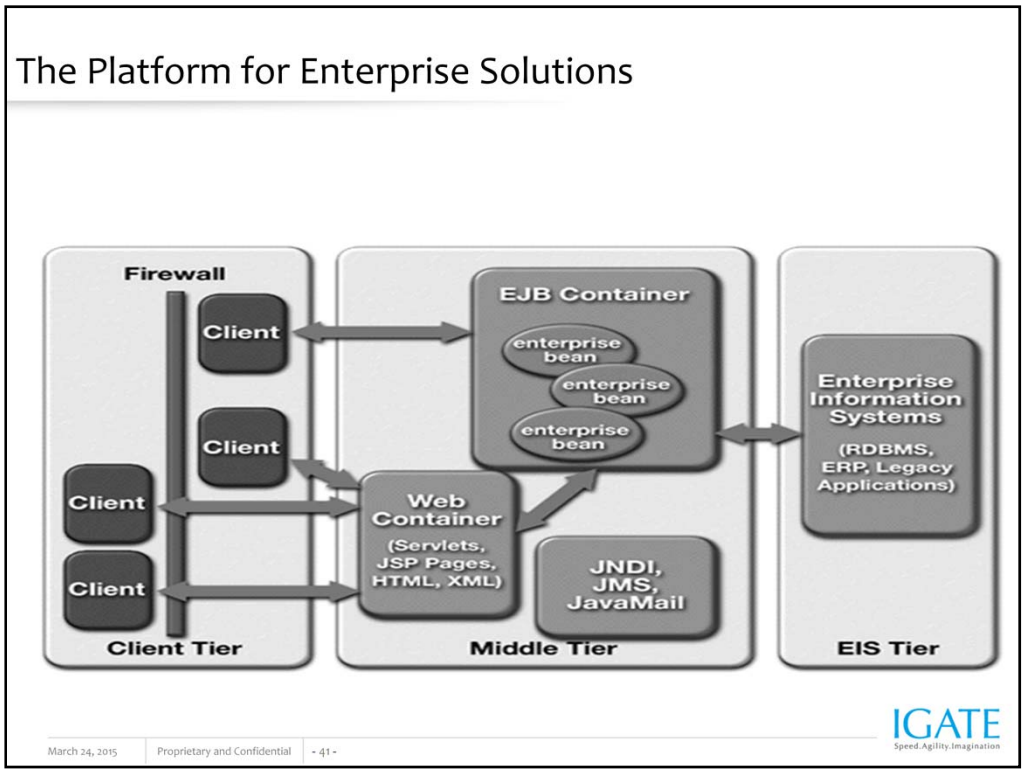



Apart from the two major benefits of platform independence and ease of learning of the language, here are some other aspects that are quite standard in Java application servers.
Support CORBA and COM in most cases through the Enterprise Javabeans specification.
Feature Servlet support - implies that business logic need not be centralized at the server. You can have a number of servlets - mini applications that are spawned by the application server - handle your processing in different machines. Your databases could exist on entirely different machines.




The Enterprise JavaBeans architecture is a server-side technology for developing and deploying components containing the business logic of an enterprise application. Enterprise JavaBeans components, also referred to as enterprise beans, are scalable, transactional, and multi-user secure. There are three types of enterprise beans: session beans, entity beans, and message-driven beans.

In a multitier J2EE application, the Enterprise JavaBeans (EJB) tier hosts application-specific business logic and provides system-level services such as transaction management, concurrency control, and security. Enterprise JavaBeans technology provides a distributed component model that enables developers to focus on solving business problems while relying on the J2EE platform to handle complex system-level issues. This separation of concerns allows rapid development of scalable, accessible, robust, and highly secure applications. In the J2EE programming model, EJB components are a fundamental link between presentation components hosted by the Web tier and business-critical data and systems maintained in the enterprise information system tier.




Multitier Model As illustrated, the J2EE platform provides a multitier distributed application model. This means that the various parts of an application can run on different devices. The J2EE architecture defines a client tier, a middle tier (consisting of one or more subtiers), and a back-end tier. The client tier supports a variety of client types, both outside and inside of corporate firewalls. The middle tier supports client services through Web containers in the Web tier and supports business logic component services through Enterprise JavaBeanstm (EJBtm) containers in the EJB tier. On the back end, the enterprise information systems in the EIS tier are accessible by way of standard APIs.

## References

- J2EE Tutorial - http://java.sun.com/j2ee/tutorial/1_3-fcs
- J2EE Developers Guide - http://java.sun.com/j2ee/sdk_1.2.1/techdocs/guides/ejb/html/DevGuideTOC.html
- JNDI - http://java.sun.com/products/jndi/tutorial/

## References

- JMS - http://java.sun.com/products/jms/tutorial/
- JDBC - http://java.sun.com/docs/books/tutorial/jdbc
- JSP - http://java.sun.com/products/jsp/docs.html
- JAXP - http://java.sun.com/xml/jaxp/dist/1.1/docs/tutorial
- JNI - http://java.sun.com/docs/books/tutorial/information/download.html
  http://java.sun.com/javase/6/docs/technotes/guides/jni/spec/jniTOC.html

Thank You

IGATE
Speed.Agility.Imagination