Introducing JEE and Eclipse

J2EE now called as JEE(since version 5). JEE used to develop Enterprise applications.

J2EE released in 1999 but major changes done in version 5 released in 2006.

Using latest JEE can easily develop multi-tier distributed applications.

J2EE just focused on core services and left the easier tasks to the external frameworks such as MVC and persistant frameworks. but JEE brought many of these frameworks in the core services and supporting annotation. These services made lot easier the application development.

By using IDE(Integrated Development Environment) we can develop applications fast.Ex:Eclipse supports JEE and supports build, unit testing and version control.

JEE has some Technologies and each Technolgy has many specifications and different versions and each performs specific task.

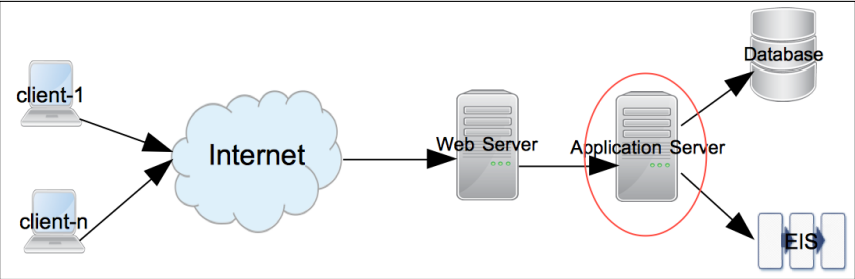
As this book was written the version of the JEE is 7.

JEE has 3 groups

1. Presentation Layer
2. Business Layer
3. Enterprise integration Layer

All these includes APIs(Application Programming Interface).

JEE web application flow



Request starts from client such as browser application or desktop application requesting services from a remote application.

Request will be received by the Web Server such as Apache Web Server, IIS, Nginx.

If the request is for static content then is handled by one or more Web Servers.

If the request is for dynamic content then that should be handled by Application Server.

JEE servers handles the dynamic request.

Most of the JEE Technology specification API execute in the Application Server.

Examples of JEE Application Servers are WebLogic, WebSphere, GlassFish, JBoss.

Most of the Complex JEE applications access database or Enterprise Integration Server(EIS) for data and to process it.

Respose is returned from the Application Server to the Web Server and then to the clients.

**Following is the brief description of some specifications that are available in the different layers.**

**The Presentation Layer**

The Technologies and Specifications that are present in this group receive the request from web server and send back the response usually in HTML format. It is also possible to return only data from presentation layer. Example, in JavaScript Object Notation(JSON) or eXtensible Markup language (XML) which consumed by Asynchronous JavaScript and XML (AJAX) that calls to update only part of the page, instead of rendering the entire HTML page.

Classes in the Presentation Layer are mostly executed in a Web Container

Web Container is a part of the Application Server that handles the web requrests. Example: TomCat is a Web Container.

**Some specifications if the Presentaion Layer Group**

**Java Servlet**

Java Servlets are Java Software Components for Server Side.

Used to process a request and send back the response in web applications.

Servlets are useful to handle request that do not generate large HTML response.

Servlets are usually used as controllers in MVC(Model View Controller) Forwarding/Redirecting requests or generating non-HTML resposes such as PDF’s.

To generate HTML response from servlet, we need to embed the HTML code(as Java String) in Java code. Therefore it is not the convenient way to generate large HTML response.

JEE7 has Servlet 3.1 API.

**Java Server Pages**

Like Servlets JSP’s are also Server Side Java Software Components used to process web requests.

JSP’s are great to handle requests that generate large HTML responses.

In JSP pages Java code or JSP tags can be mixed with other HTML code, such as HTML tags, JavaScript and CSS.

Since Java code is embedded in larger HTML code it is easier than(Servlets) to generate large HTML respose from JSP pages.

JEE7 has the JSP specification 2.3

**Java Server Faces**

JSF used to create user interface on the server side by incorporating the MVC design pattern in its implementation.

Example Task - If we have a page that posts form data from a browser, we can have JSF to save that data in a Java Bean so that it can be used to subsequent response to the same or different requests.

JSF also makes easier to handle UI events on the server side and specify page navigation in an application.

JSF codes are written in JSP using custom JSP tags for JSF

JEE has JSF 2.2

**The Business Layer**

We write code in the Business Layer for the business logic of an application

The request to Business Layer may come from Presentation Layer or directly from client application or sometimes from the Web Services.

Classes in this layer are executed in the application container part of the JEE Server.

GlassFish and WebSphere are the examples of web container and also an application container.

**Enterprise Java Beans**

EJB’s are Java Classes where we can write our business logic. Its not mandatory to use EJB’s to write business logic but EJB’s provide some essential services that are required in enterprise applications. Example Services are Security, Transaction Management, Component Lookup, Object Pooling.

We can distribute EJB’s across multiple servers and let the application container(EJB container) take care of Component Lookup(Searching Component) and Component Pooling(for scalability). This can improve scalability of the application.

There are Two Types of EJB’s

1. **Session Beans:** Session Beans are called directly by clients or middle tier objects.
2. **Message Driven Beans:** Message Driven Beans are called in response to Java Messaging Service(JMS) events.

JMS and Message Driven Beans can be used to handle asynchronous requests.

Example Scenario : The client puts a request in the messaging queue and does not wait for the immediate response. The server application gets the request message directly by using JMS API’s or using MDB. it process the request and put the response in the different queue to which the client would listen and get the response.

JEE7 contains EJB specification 3.2 and JMS specification 2.0

**The Enterprise Integration Layer**

API’s in this layer used to interact with external to JEE application systems in enterprise.

Accessing database and API’s comes under this layer.

**Java DataBase Connectivity(JDBC)**

JDBC is a specification to access database. Using JDBC we can execute SQL statements and get results on differrent databases using common API’s.

Database specific driver sits between the SQL call and the database. Driver translates the JDBC calls to database vendor specific API calls.

JDBC can be used directly in both Presentation and Business layer.

We should seperate database calls from both UI and business code.usually this is done by creating Data Access Objects(DAO) which encapsulate logic to access database.

JEE7 containes JDBC specification 4.0

**The Java Persistant API (JPA)**

One of the problems with using JDBC API’s directly we have to constantly map the data between Java Objects and data in columns of rows in relational database.

Frameworks such as Hibernate and Spring have made this process easier by using Object Relational Mapping(ORM). ORM is included in the JEE in the form of Java Persistant API(JPA). JPA gives the flexibility to map the objects to the tables in a relational database and execute queries with or without using Structured Query Language

JEE7 has JPA specification 2.1

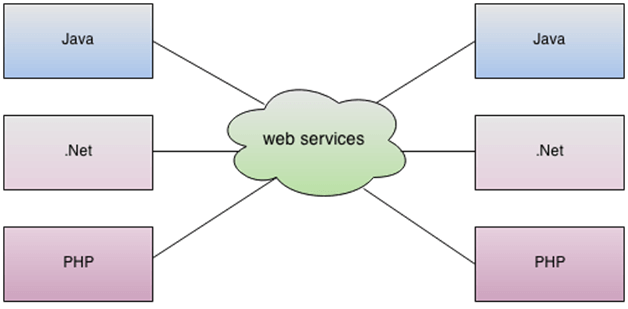
**Java Connector Architecture(JCA)**

JCA API’s used to communicate with the Enterprise Integration Systems such as SAP, Salesforce.

Just like drivers to communicate in between JDBC API’s and relational database we have JCA adapters in between JCA calls and EIS. Most EIS applications provide REST API’s, which are light weight and easy to use. So REST can replace JCA in some cases.but advantage of using JCA is we get transaction and pooling support from JEE application server.

**Web Services**

Web Services are remote application components. It is a collection of standards or protocols for exchanging information between two devices or application. So web service is a language independent way of communication.



**There are Two Standards of Web Services**

1. **Simple Object Access Protocol (SOAP)**
2. **Representational State Tranfer (REST)**

JEE provides many specifications to simplify development and consumption of both type of web services for example, JAX-WS (Java API for XML – web services) and JAX-RS (Java API for RESTful Webservices)

**Eclipse IDE**

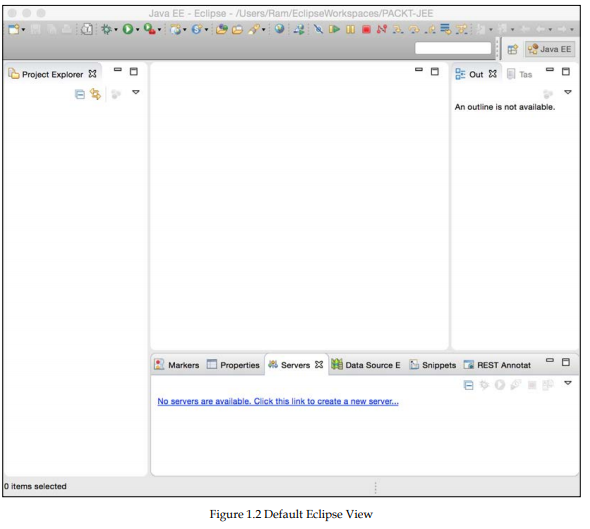
Eclipse has great editor features and has many integration points with JEE technlogies.

Eclipse is an open source IDE for developing applications in different programming languages.

We can develop many different types of Java Applications.

Support for many languages is added as plugins.

Eclipse can also interact with the external systems during the development process such as source control like SVN and Github. Build tools such as Apache Ant and Maven. File explorer for remote systems using FTP. Managing services such as Tomcat, GlassFish. And also database explorer.



**Terms used with respect to eclipse**