J2EE Architecture - A Review

Achilleus Almeida

Abstract: Java™ 2 Platform, Enterprise Edition (J2EE) is a set of specifications, for developing enterprise-level applications created by the JCP (Java Community Process). In simple terms it provides a standard for developing multitier, enterprise applications. Advancements in technology further enhanced the need for scalable, efficient and faster solutions for information management. The JEE technology is best suited when it comes to meeting these requirements. J2EE architecture has various layers working in unison. These include the client, presentation, business logic and data access layer. The client tier acts as the interaction point between the client and the application. The presentation layer is where an interface is generated dynamically using different components. These components may include Servlet's and JavaServer Pages (JSPs), or standalone Java applications. The middle tier is also called as the server tier. Enterprise beans and Web Services encapsulate distributable, reusable business logic for the application in the next tier which also happens to be known as the server tier. The forth tier handles connections with the respective databases.

Index Terms: Business logic layer, Client layer, Dynamic Applications, Data Access Layer, Enterprise Applications, Presentation layer, Programming model etc.

1 INTRODUCTION

THIS paper tries to depict the use of J2EE architecture to design and implement a dynamic web based application. There is a growing need for faster, more efficient, and larger-scale information management solutions due to the current economy and the technologies in use now days. These challenges can be successfully tackled by using the J2EE specification. This is because it provides a programming model that improves productivity, development and standardizes the platform for hosting enterprise applications. IT also ensures the portability of developed applications with an extensive test suite. Enterprise applications are portable between application servers supporting the J2EE specification. In other words, J2EE further enhances the "write once, run anywhere" promise of Java. J2EE adopts the layered thought. J2EE is used to develop n-tier applications by breaking apart the different layers in the two-tier (client-server) architecture into multiple tiers This symbolizes the division of the application logic according to their function into components. Depending on which tier in the multitier Java EE environment the application components belong the various application components that make up a Java EE application are installed on different machines. The different tiers of the J2EE architecture are described further in this paper.

2 HISTORY OF THE J2EE ARCHITECTURE

Originally Sun Microsystems developed the J2EE specification. Beginning with J2EE 1.3, the specification was developed under the Java Community method. The Java EE 5 specifications were developed under JSR 244 and also the final release was made on May 11, 2006. With the web revolution, Java step by step evolved into a language for client-side development with capabilities like applets and JavaBeans. Along the way, many Java API's, like JDBC, were developed to deal with market desires for generic access and usage of resources usually needed by enterprise software applications. The main reasons why java is preferred when it comes to the development of enterprise software are its simplicity, rich set of API's, platform independent architecture and also its nature to handle the web.

• Achilleus Almeida is currently pursuing masters degree program in Computer Applications at Vivekanand Education Society's Institute Of Technology, India, PH-+91-9920217989.

3 THE MULTI-LAYERED ARCHIETECTURE OF J2EE

3.1 The Client Layer

The user accesses the application through the client layer. When choosing a presentation layer technology for any n-tiered design, Java developers are typically left with 2 choices: JSP (Java Server Pages) or a Swing/AWT (Abstract Windowing Toolkit) solution. JSP offers Java developers the flexibility to make dynamic content that's extraordinarily straightforward to distribute. However, developers should surrender a particular degree of control over however their application can run when delivered in various browsers. Swing offers developers the flexibility to manage most aspects of an application's behavior, however needs that users install a Java runtime. An application may require the following J2EE components in the client layer:

- Web Browsers
- Applets

3.1.1 Web Browsers:

The user's web browser downloads static or dynamic Hypertext Markup Language (HTML), Wireless Markup Language (WML), or Extensible Markup Language (XML) web pages from the web tier. Dynamic web pages are generated by servlets or JSP pages running in the web tier.

3.1.2 Applets:

An applet is a small client application written in the Java programming language that executes in the Java VM installed in the web browser. However, client systems will likely need Java Plug-in and possibly a security policy file so the applet can successfully execute in the web browser.

3.2 The Presentation Layer

The presentation layer is where the user interface is dynamically generated. An application may require the following J2EE components in the presentation layer:

- Servlet's
- JSP's
- Static Content

3.2.1 Servlet's:

It is small Java program that runs within a Web Server. The requests from web clients are received from and responded to by servlet's usually by making use of the HTTP (HyperText

Transfer Protocol)

3.2.2 JSP's:
One way of developing dynamic web applications which are platform independent is by making use of JSP's (JavaServer Pages).JSP's are based on Java Servlet Technology.

3.2.3 Static Content:
This refers to text as well as other multimedia content on a web page that do not change based on a per-request basis Eg: text, images, audio, video, css etc

3.3 The Business Logic Layer
Components (also known as "EJB (Enterprise Java Bean)" components) that contain business rules and other business functions are encapsulated within this layer. These components are of the following types:

- Session Beans
- Entity Beans
- Message-Driven Beans

3.3.1 Entity Beans: The type of bean which is used to manage data which is persistent, to perform complex business logic and can be uniquely identified by making use of a primary key is known as an entity bean

3.3.2 Session Beans: A bean which can be invoked by a client over local, remote or web service client views, programmatically and encapsulates business logic is known as a session bean.

Types of Session Beans:
- Stateful
- Stateless
- Singleton.

Stateful Session Beans:
Values of the instance variables of an object are stored in its state. The instance variable in a stateful session bean depicts the state of a unique client-bean session.

Stateless Session Beans: A conversational state (interaction between client and its bean) with the client is not maintained by a stateless session bean.

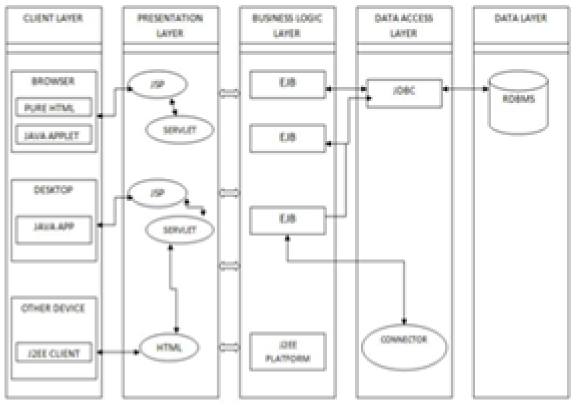
Singleton session bean: A bean which is instantiated once per application and exists throughout the lifecycle of the developed application is called a singleton session bean.

3.3.3 Message-Driven Beans: Java EE applications are allowed to process messages asynchronously by making use of message driven beans. This bean behaves usually like a JMS (Java Message Service) listener. It is functionally similar to an event listener. The only difference is that instead of receiving events it receives JMS messages.

3.4 The Data Access Layer
In order to fire queries, return results and connect to databases JDBC (Java database connectivity) is used. These features are a part of the data access layer. The definition of business data of the application program is the responsibility of this layer. Simulation Of real world entities along with the business flow of the organization is also carried out at this layer. As this layer encompasses all these features we can say that it is the foundation of an enterprise application.

4 ILLUSTRATION OF J2EE ARCHITECTURE

6.1 Figures and Tables



5 ADVANTAGES OF J2EE
- J2ee provides for a simplified architecture and development methodology because it provides us with the capabilities of dynamic as well as component-based assembly/deployment.
- Demand variations can be met easily because J2ee provides us with the property of scalability through load-balancing, transaction support as well as DB connection pooling.
- Using J2EE it is easy to carry out integration with existing information systems as it provides us with various APIs which assist us in the integration process
- The is a wide variety of choices when it comes to servers, tools and components as there are a number of tools (IDE) server choices which can easy handle J2EE
- It supports a wide range of security requirements and hence we say that it makes use of a flexible security model.

6 END SECTION

CONCLUSION
J2EE may be a very comprehensive platform and initially the range of technologies and API's may appear intimidating. But by building our knowledge of J2EE piece by piece, technology by technology, one will soon be in a decent position to start out planning and building J2EE systems. The component technology of JSP, Servlet, and EJB etc. supported by the J2EE platform and MVC style pattern can alter the developing process and improve the software performance. The Java EE platform provides everything you need to design, build, test, and deploy distributed multi-tiered applications.

REFERENCES

[1] J2EE AND MVC ARCHITECTURE- Manish Bhatt, Banaras Hindu University (BHU), Varanasi (UP) Journal of Global Research Computer Science & Technology Researcher Forum (JGRCST) Vol-I, Issue-II, July 2014 ISSN: 2349 - 5170

[2] The J2EE 1.4 Tutorial, http:// java.sun.com/j2ee/1.4/docs/tutorial/doc/ 2015

[3] JAVA J2EE Tutorial: http://www.tutorialspoint.com/listtutorials/java/j2ee/1 2016

[4] IntroductiontoJ2EE: http://www.javaranch.com/journal/2002/10/J2EE.html 2002

[5] J2EE History: http://www.j2eebrain.com/java-J2ee-j2ee-history.html 2009

[6] J2EE AND MODEL-VIEW-CONTROLLER ARCHITECTURE – A REVIEW: Naveen Malik, Naeem Akhtar, Pankaj Sharma, Rahul, Hardeep Rohilla Dronacharya College of Engineering, Khentawas, Farukhnagar, Gurgaon. Journal Of Harmonized Research in Engineering 1(2), 2013, 110-114

[7] W3 Schools: Java Introduction http://www.w3schools.in/java-tutorial/intro/ 2016

[8] SunOne: Overview of clients. https://docs.oracle.com/cd/E19651-01/817-2150-10/dcoverview.html 2003

[9] Just Objects http://justobjects.org/cowcatcher/browse/j2ee/slides/j2ee-overview/j2ee/j2ee-over/slide.0.10.html 2014