A Role Of J2EE And .Net In Enterprise Level Application Development

Gourav Rawal
Sunil Patil

Abstract

Enterprise level application development means eBusiness term in ology used to describe the information systems and applications that support and drive business processes using web technologies. eBusiness means any business process that relies on an automated information system. This is mostly done with web-based technologies. eBusiness methods enable companies to link their internal and external data processing systems more efficiently and flexibly to work more closely with suppliers and satisfy the needs and expectations of their customers. Applications of eBusiness are based on N tier architecture. For developing such types of applications there are two approaches – J2EE and .NET. The main objective of this paper is to present the concept of eBusiness and N tier architecture as well as to present the comparison of J2EE and .NET technology for developing the enterprise level application development (eBusiness) on specific criteria.

Keywords: eBusiness, N tier architecture, J2EE, .NET

1 Introduction : Nowadays eBusiness system has been very popular since Internet and World Wide Web came out. eBusiness may be defined as the application of information and communication technologies in support of all the activities of business. Electronic business methods enable companies to link their internal and external data processing systems more efficiently and flexibly, to work more closely with suppliers and partners, and to better satisfy the needs and expectations of their customers. eBusiness focuses more strategic focus with an emphasis on the functions that occur using electronic capabilities. eBusiness involves business processes spanning the entire value chain : electronic purchasing and supply chain management, processing orders electronically, handling customer service, and cooperating with business partners. eBusiness software solutions allow the integration of intra and inter firm business processes. eBusiness can be conducted using the Web, the Internet, intranets, extranets or some combination of these [1].

The activities of eBusiness can be divided into three categories based on end-users of transactions On the Internet: Intra-business, Business-to-Consumer, and Business-to-Business. Intra-business activity share company information and computing resources between employees on the Internet. Business-to-Business activity improves inter-organizational partnerships and relationships. Business-to-Consumer activity provides the services to the consumers who are out of organization.

Java, especially J2EE (Java 2 Enterprise Edition) or ASP (Active Server Pages) has been widely used technology to develop server side web systems for e-Business. J2EE is the one edition of Java that is platform independent and object oriented language – Java is the product of *Sun Microsystems*. Therefore, J2EE is a good alternative for building e-Business systems for both a development and server side in Linux and Windows operating systems. Generally, the applications of J2EE are built in Windows Operating system and can be published into any operating systems. Microsoft Corporation provides ASP for building e-Business systems. ASP applications could be integrated with the codes in Visual Basic provided by Microsoft Corporation as the products. Therefore, ASP applications are developed and published only in Windows Operating Systems. In this paper, we present the concept of two most widely used e-Business development approaches J2EE and .NET. Also compare both these approach on the basis of several factors, introduces the N-tier architecture which is the fundamental architecture of e-Business system.

2. e-Business Architecture (N-Tier) The traditional client server architecture is based on distribution of functions between two types of independent and autonomous processes: Server and Client. A client is any process that requests specific services from the server process. A server is a process that provides requested services for the client. Client and Server processes can reside in same computer or in different computers linked by a network. In case of two tier client/server model, the application processing is done separately for database queries and updates and for business logic processing and user interface presentation. The network binds the back-end of an application to the front end and both tiers can be present on the same hardware. In two tier architecture, if the client/server application has a number of business rules required to be processed, then those rules can reside at either the client or at the server. In case of two tier architecture, as the application development is done on the client side, maintenance cost of application, as well as client tools etc is expensive. In case of two tier architecture, since actual processing of data takes on the remote client, the data has to be transported over the network. This leads to the increased network stress. Applications are loaded on individual PC i.e. each application is bound to an individual PC. For this reason, the application logic cannot be reused [2]

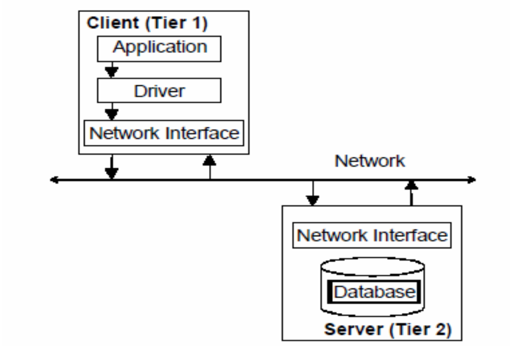

Fig.1 Two-Tier Client/Server Model

The N tier architecture provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers, developers only have to modify or add a specific layer, rather than have to rewrite the entire application over. The N tier architecture splits the solution process into different projects based on the business requirements. The main advantage of using N tier architecture is that the complexity associated with the business and the process is reduced and it is easy to work with. The elements of performance, scalability and future development issues need to be considered when deciding on the architecture of the application depending on the priorities required [3].

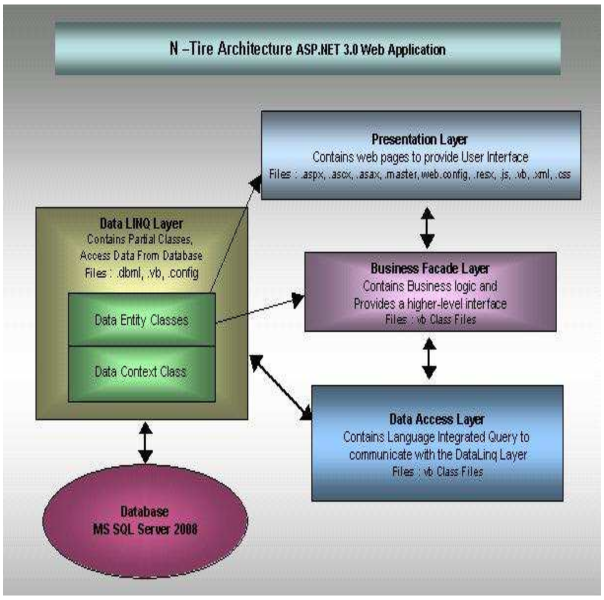

Fig 2. N tier Architecture

The N tier application has 3 tiers - presentation tier, business tier and data tier. Each tier or layer interacts with the layer directly below and has specific function to perform. Presentation layer is responsible for displaying user interface to the programmer end user. Programmers use this layer for designing purposes and to get the data back and forth. The business layer works as a mediator to transfer the data from presentation layer. In the 3 tier architecture the data access layer is not made to interact with the presentation layer. The architecture in ASP.NET includes using SqlConnector or OleDb objects to retrieve, update and delete data from databases and passing the data retrieved to the presentation layer in DataReader or DataSet object. The data layer gets the data from the business layer and sends it to the database or gets the data from the database and sends it to the business layer. There are many business benefits to N tier architecture. For example, a small business can begin running all tiers on a single machine. As traffic and business increases, each tier can be expanded and moved to its own machine and then clustered. This is an example to show how the N tier architecture improves scalability and supports cost-efficient application building. The N tier architecture makes application more robust because there is no single point of failure. Tiers function with relative independence. The

N-tier model also makes applications more readable and reusable. It reduces the amount of code. Custom tags and EJBs are easier to port to readable applications in well maintained templates. The N-tier architecture offers innovations in the standard Client-Server technology that spawned the Internet itself. These are used to develop dynamic websites, web application or web services[4].

3 J2EE: Java is a programming language that has been developed specifically for the distributed environment of the Internet. Java can create applications that are distributed between clients and servers in a network. Java source code is compiled into bytecode. A Java interpreter then executes this byte code form at Java interpreters and runtime environment run on a Java Virtual Machines. Java is one of the most well suited languages for working on the World Wide Web and the client server model is the primary model for working on distributed networks. Java API has a set of built in libraries as bytecodes. Java 2 platform Enterprise Edition (J2EE) defines the standard for N-tier architecture [5]. J2EE is a technology that aims to simplify the design and implementation of enterprise applications[6].
The main reason for the evolution of enterprise application frameworks are as follows:

- Need to abstract new technologies especially improvements on web technologies.
- Need to handle complex low level details inherent in enterprise applications like security, transaction processing and multithreading.
- Evolution and popularity of widely accepted concepts like N-tier architectures and component based software development.

J2EE has the extended API from Java platform standard edition. It is based on the J2EE components for simplifying the development cycle by providing the details of application behavior. It also includes the features of Java platform standard edition such as platform independency and object-oriented languages. Apart from these J2EE includes many technologies such as Enterprise Java Beans (EJB), Remote Method Invocation (RMI), Java Naming and Directory Interface (JNDI), Java Database Connectivity (JDBC), Java Messaging Service (JMS), Java Servlets and Java Server Pages (JSP) and XML[7]. The J2EE architecture is a multi-tiered architecture. The J2EE architecture consists of the following tiers[8].

Client tier – The client interacts with the user and displays information from the system to the user. The J2EE platform supports different types of clients, including HTML clients, Java Applets and Java applications.

- **Web tier –** The web tier generates presentation logic and accepts user responses from the presentation clients, which are HTML clients, Java applets and other web clients. Based on the received client request, the presentation generates the appropriate response to a client request that it receives. In the J2EE platform, servlets and JSPs in a web container implement this tier.
- **Business tier –** Business tier handles the core business logic of the application. The business tier provides the necessary interfaces to the underlying business service components. The business components are implemented as EJB components.

- **EIS tier –** EIS tier is responsible for the enterprise inform at ion system s, including database system s, transaction processing system s, l egacy system s and enterprise resource planning system s.

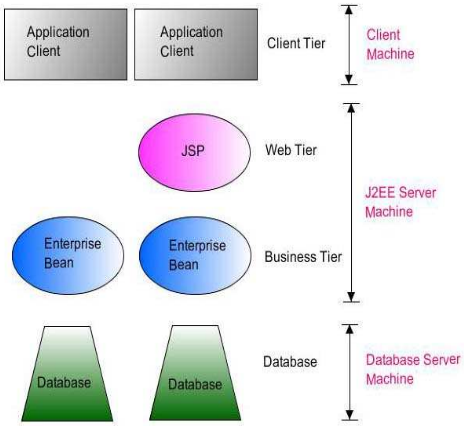

Fig 3 J2EE Application Model

J2EE has been very popular to im plem ent e-Business application because it support platform independent concept and has high performance com paring the CGI system s and Perl. M icrosoft Corporation's ASP.NET is another com petitor for developing e-Business applications but only for M icrosoft W indow s system with IIS w eb server. Therefore, J2EE has been m ost popular tool for developing e-Business system in the current m arket.

4. NET : M icrosoft Corporation is very popular for developin g the w indow s based applications in the personal com puter m arket. M icrosoft not only focused on the W indow s server m arket but also focused on its com ponent such COM Com ponent is sim ilar to object and independent, but it provides a function to adapt w ith and interface of operator, property and event. M icroso ft ASP.NET and other languages of V isual Studio have been used for developing e-Busin ess applications on Internet and W orld-W ide-W eb. How ever, the M icrosoft applications m ainly depend on W indow s operating system, therefore M icrosoft lost the huge m arket of e-Business system against J2EE. Therefore, M icrosoft presented .NET. W ith .NE T fram ew ork, M icrosoft can com pete w ith and over J2EE in e-Business application s and w eb services m arkets. The .NET fram ew ork supports m ulti-language facility. The .NET fram ew ork supports VB, C++, C#, J# languages. The code w ritten in any one of these languages is converted into M SIL (M icrosoft Interm ediate Language Code). .N et sim plifies the developm ent of robust client/server and m ulti (N-Tier) applic ations [9].
M icrosoft .NET offers language independence and lang uage interoperability. One of them ost intriguing and fundam ental aspects of .NET platform. A single .NET com ponent can be w ritten, for exam ple, partially in VB.NET, the .NET version of V isual Basic, and C#.NET. The source code is translated into M SIL. This M SIL code is language neutral and is analogous to Java byte code.
The Com m on Language Runtim e (CLR) executes program on w eb server. The CLR activates objects, perform s security checks, execut es them and handles garbage

collection. The CLR is sets of framework base classes followed by an additional layer of data and XML classes plus another layer of classes intended for web services and web forms and window forms.[10]

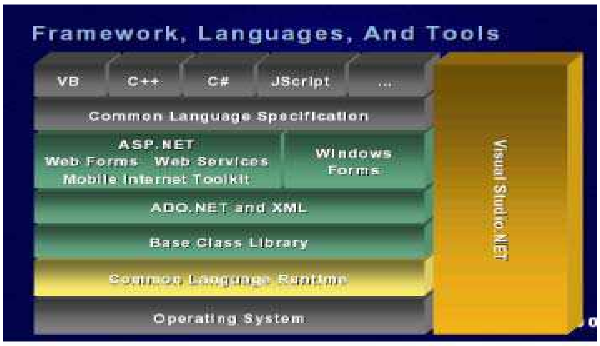

Fig4.NET Framework

5.J2EE vs .NET Language Support- J2EE promotes Java-centric computing and all the components deployed into a J2EE deployment must be written in the Java language. J2EE is the enterprise edition of Java. J2EE provides enterprise development and deployment with the enterprise APIs like JDBC, JNDI, Servlets, JSP, EJB etc. The JDBC APIs are used to connect a Java source code to database. The JNDI APIs are used to access distributed objects. The Servlet APIs are used to handle the HTTP request and response between client and server. The JSP are used to create a dynamic page to integrate the presentation logic with HTML documents. The RMI APIs are used to execute the methods of the remote objects on networks. Thus, on the basis of these features we can say J2EE is a good solution of Business systems. .NET is for platform independent applications as J2EE. Besides, existing programming languages such as C++, VB, ASP, C# and J# can interoperate each other in .NET framework. These languages have compilers that support .NET framework Common Runtime Library.[11]

Platform Independency– Java is a platform independent programming language with Java Virtual Machine provided by Sun Microsystems. The Java source codes are compiled into Java bytecodes. The Java Bytecodes could be run on any platform like Linux or Window environment, which the platform has the Java Virtual Machine. .NET framework is to achieve platform independency but works only on windows environment.

Security– In the world of Internet, security is a big issue. J2EE support two types of security control: Programmatic Access Control and Declarative Access Control. To disallow an operation based on role or identification is called the declarative access control. But sometimes it may be required to decide whether or not an operation can be allowed based on the data being manipulated. This can be handled by programmatic access control. A platform base JVM supports byte code verification and policy based security.

.NET also supports two types of security: Code Access Security and Role Based Security. The Code Access Security mechanism performs security checks before code is executed by analyzing the metadata and comparing the access requests to the system settings and during JIT compilation. The Role Based Security is similar to what is used in Windows today. One can create roles and assign permissions to them.

Performance: In .NET, intermediate language (IL) is just in time compiled to native code and executed. In J2EE, the Java byte code is interpreted by the JVM. Since interpreting is slower than executing native code, applications written on .NET perform better than J2EE applications.

6. Conclusion: In this paper, we describe the concept of N tier architecture of e-Business. J2EE and .NET is the two most popular approaches for developing the applications on N tier architecture. J2EE specification provided by Sun Microsystems and is more opened because anyone can implement J2EE application server that match the specification. On the other hand, .NET platform is product of Microsoft Corporation. Therefore it is completely dependable on Microsoft products.

References

1. http://en.wikipedia.org/wiki/Electronic_business
2. An Introduction to Client Server Computing, Subhash Chandra Yadav, Sanjay Kumar Singh, New Age International Publishers
3. http://computechnocrat.blogspot.in/2011/04/n-tier-architecture-in-aspnet.html
4. http://www.exforsys.com/tutorials/client-server/n-tier-client-server-architecture/1.html
5. "Java 2 Platform, Enterprise Edition (J2EE)", http://java.sun.com/j2ee/, Sun
6. Microsystems, Inc. (2003)
7. http://www.exforsys.com/tutorials/j2ee/j2ee-overview.html
8. J2EE Professional Projects, Shadab Siddiqui and Pallavi Jain
9. Java 2 Platform, Enterprise Edition Platform and Component Specification by Bill Shannon, Mark Hapner, Vlada Matena, James Davidson, Eduardo Pelegri-Llopart, Larry Cable
10. .NET Framework Essentials, 3rd Edition By Hoang Lam, Thuan L Thai
11. Programming ASP.NET, 2nd Edition by Dan Hurwitz, Jesse Liberty
12. "Got Dot NET (.NET Framework Website)", http://www.gotdotnet.com/team/lang, Microsoft Corporation (2003)