Professional training and consulting in Enterprise JavaBeans (EJB) and Java 2 Platform, Enterprise Edition (J2EE) Technologies

The Business Benefits

of EJB and J2EE Technologies over COM+ and Windows DNA

By Ed Roman and Rickard Öberg
December 1999

Prepared for Sun Microsystems, Inc.

www.middleware-company.com

Table of Contents

About This Whitepaper

This whitepaper highlights the business advantages of the J2EE architecture over Windows DNA. For a technical analysis of these two architectures, please see the accompanying whitepaper, *The Technical Benefits of EJB and J2EE Technologies over COM+ and Windows DNA.*

This whitepaper was primarily written by two authors, however we would like to thank our independent review panel that has given us feedback, contributions, and technical suggestions. This includes Ian McCallion, Robert Orfali, Gopalan Suresh Raj, Anne Thomas, Karl Avedal, Jonas Wallenius, Doug Hibberd, Scott Brittain, Chip Wilson, Adam Berman, and Floyd Marinescu.

Executive Summary

As the Internet becomes a primary mechanism for conducting business in the 21st century, the scalability, maintainability, versatility, and availability of commerce systems become chief concerns. IT departments will not be able to successfully adapt to changes in technology and business if commerce systems are not designed with enough flexibility. The failure of a commerce site could mean millions of dollars in lost revenues, which is unacceptable in today's fast-paced economy. A robust, well-designed server-side platform provides the infrastructure and tools to build flexible and reliable commerce solutions that meet business' core needs and enables enterprises to leverage IT as a source of competitive advantage.

Corporations have two prominent server-side platform choices today. The first is Sun Microsystems' Java 2 Platform, Enterprise Edition (J2EE), which includes the Enterprise JavaBeans (EJB) server-side component architecture. The second is Microsoft's Windows Distributed interNet Applications Architecture (Windows DNA), which includes the COM+ server-side component architecture.

The key business benefits that J2EE offers over Windows DNA include industry buy-in and choice of vendors, a faster time to market through more advanced middleware services, a greater level of protection of past and future IT investments, a faster response to change, and a higher potential for overall system scalability and reliability.

Industry buy-in and choice of vendors. J2EE is an open architecture that represents the combined expertise of a collaborative industry effort, and has wide industry support from key middleware vendors. The vendor-neutrality of the J2EE standard removes barriers to entry for new vendors who wish to compete in the marketplace. The result is that customers can select products from an ecosystem of server-side platforms, tools, and components. If a customer designs a J2EE application well and remains loyal to the standard, then new infrastructure and platform technology can be adopted with minimal changes to applications. In comparison, Microsoft's Windows DNA is based on proprietary technology, raising barriers to entry that prevent competition. Customers that choose Microsoft technology are bound to a specific middleware implementation; applications cannot be ported to another middleware platform that emerges as optimal for a particular business' specialized needs.

Faster time to market. Advanced J2EE platform-based products handle many more middleware tasks on behalf of the developer than the Windows DNA platform handles. This enables developers to focus more on the business problem at hand, rather than focusing on middleware development. Applications are developed more quickly, resulting in a faster time to market. See the accompanying technical whitepaper for more information, entitled *The Technical Benefits of EJB and J2EE Technologies over COM+ and Windows DNA*.

Greater utilization of existing IT assets. J2EE layers on top of existing systems, which allows a corporation to fully leverage past investments. Almost any existing investment can be leveraged, such as one using BEA *Tuxedo*, IBM *CICS*, IBM *Encina*, Inprise *VisiBroker*, or the Sun-Netscape Alliance's *Netscape Application Server* (future: *iPlanet*). Customers can remain with their existing middleware vendors, and re-use existing applications, operating systems, hardware, developer skill sets, and vendor relationships. This reduces an IT shop's deployment costs and risk levels. In comparison, companies adopting Microsoft's Windows DNA will have a two-vendor bridged solution, increasing risk levels and vendor dependencies.

Greater protection of future IT assets, and a faster response to change. In today's world of Internet commerce, an IT department's business requirements are in a constant state of flux. New levels of functionality, scalability, and availability are always in demand. To make matters worse, mergers or acquisitions often result in complex heterogeneous deployment environments. A server-side platform is needed that enables customers to adapt to these changes. J2EE gives customers this adaptability. A well-designed J2EE application can be deployed into a heterogeneous environment of middleware, hardware, or operating system products with minimal modifications. This enables customers to choose their desired service level without throwing away applications, protecting future IT assets. In comparison, a Windows DNA-based application will not run on heterogeneous systems, and customers cannot switch vendor products without substantially rewriting applications. This introduces a higher level of risk because customers cannot respond as quickly to changes.

Choice of proven, scalable, and reliable platforms. J2EE customers can choose from a wide array of operating systems and hardware that can scale to a very large number of processors. This enables customers to increase the scalability levels of the underlying platform quickly and easily over time and as requirements emerge. J2EE customers can also choose operating environments with a proven history of high availability, which helps lower risk levels. In comparison, Microsoft's Windows DNA limits the underlying platform to Windows 2000. Windows 2000 supports fewer processors than traditional operating systems, does not provide robust data caching, and does not yet support middle-tier load-balancing which is needed for a highly scalable system. Microsoft also has yet to show levels of availability offered by traditional operating system vendors.

Introduction

Corporations are struggling to achieve scalable, maintainable, and highly available commerce systems to meet the demands of their customers and partners. Any commerce site, be it business-to-business, business-to-consumer, or consumer-to-consumer, faces these same challenges. A robust server-side platform, consisting of a runtime environment, middleware services, development tools and management tools, can be harnessed to achieve enterprise-class deployments of commerce systems.

In the past, corporations were forced to invent a server-side platform in-house. Most corporations do not have the expertise to build such a platform, and those that do have a maintenance nightmare, since they seldom have a core competency in middleware. Over time, the maintenance costs of a custom middleware solution increase exponentially, and it becomes necessary to train new hires on the proprietary architecture and evolve the architecture as new needs arise. This is unacceptable for most companies, and is especially unacceptable for e-commerce startups that deliver solutions on the order of months or even weeks; they cannot spend the necessary man-years to build and maintain a complex middleware solution.

To enable corporations to solve business problems rather than invent middleware, the modern approach to server-side Internet computing *partitions* the task of constructing a deployment into separate *roles*. Rather than build an entire deployment from scratch, a corporation can purchase a server-side platform from a middleware vendor, re-usable components from an application vendor, and development and management tools from a tools vendor. This results in a speedier time to market (development and maintenance time decrease), as well as best-of-breed deployments (the aforementioned vendors employ top experts to develop their particular products). Greater efficiency is achieved through specialization, and the corporation is free to focus on solving the business problem at hand.

Currently, corporations have two choices of server-side platform:

A single vendor's proprietary architecture. A proprietary architecture is one that is designed by a single vendor and locks the customer in via proprietary Application Programming Interfaces (APIs). This restricts the customer's future choices, and is especially disastrous if the server-side architecture is not suitable for enterprise-class deployments. In the past, all server-side architectures were proprietary, so customers had no choice but to adopt proprietary architectures. Today, Microsoft Corporation is the only contender that still has a proprietary server-side architecture, the Windows Distributed interNet Applications Architecture (Windows DNA), which runs exclusively on the Windows 2000 operating system.

A vendor-independent architectural standard. The vast majority of middleware vendors are adopting open standards that eliminate proprietary solutions. This enables customers to write vendor-independent applications that will run on a wide variety of middleware platforms. Sun Microsystems has led an industry collaboration to build the Java 2 Platform, Enterprise Edition (J2EE), a widely adopted, vendor-independent architectural standard for constructing server-side commerce systems that run on a wide variety of operating systems.

This whitepaper explains the business benefits that Sun Microsystems' J2EE platform offers over Microsoft's Windows DNA. We will focus on Sun Microsystems' EJB standard and Microsoft's COM+, which are the server-side component models of their respective platforms.

J2EE Overview

The Java 2 Platform, Enterprise Edition (J2EE) was designed to simplify complex problems with the development, deployment, and management of multi-tier enterprise solutions. J2EE is an open industry standard, and is the result of an ongoing industry collaboration led by Sun Microsystems. This collaborative process represents the collective expertise of the largest enterprise computing vendors (such as IBM, BEA, and Oracle), and ensures their participation and adoption. Because J2EE is an open standard, it promotes competition among vendor products and tools, and thereby promotes best-of-breed products in the middleware space, the tool space, and the off-the-shelf component space.

A well-designed J2EE application can be independent of middleware, operating system, and hardware. Middleware independence is possible because J2EE is a specification, not a middleware implementation. Sun Microsystems enforces middleware independence by requiring middleware vendors to pass a compatibility test suite before receiving the J2EE certification logo. Operating system and hardware independence is possible because J2EE is built on the platform-independent Java virtual machine.

J2EE is an evolution of several enterprise Java technologies, such as Java Database Connectivity (JDBC), and Java Remote Method Invocation (RMI). The cornerstone of J2EE is Enterprise JavaBeans (EJB) technology, a standard for building server-side components.

Windows DNA Overview

Like J2EE, Microsoft's Windows DNA was also designed to simplify complex problems with the development, deployment, and management of multi-tier enterprise solutions. However, the most significant difference between the two platforms is that Windows DNA is a proprietary product designed and supported by a single vendor, whereas J2EE is an open industry standard supported by all key middleware vendors, each of which is providing an implementation of the standard.

A Windows DNA application is tightly bound to Microsoft's Windows DNA implementation. There is no open specification for Windows DNA, which prevents other vendors from providing Windows DNA implementations. Windows DNA applications are also tightly bound to Microsoft's Windows 2000 operating system, and cannot be realistically deployed on other platforms. Because of this, hardware selection is limited as well.

Windows DNA has evolved from the middleware services provided in Windows NT, such as the Component Object Model (COM), Distributed COM (DCOM), Microsoft Transaction Server (MTS), and Microsoft Message Queue (MSMQ). The cornerstone of Windows DNA is COM+, a standard for building server-side components.

The Business Benefits of EJB and J2EE Technologies over COM+ and Windows DNA

The remainder of this whitepaper describes the business benefits of EJB and J2EE over COM+ and Windows DNA.

Industry Support

With any commerce technology, buy-in from key industry players is essential for a thriving marketplace. J2EE is a widely endorsed industry standard produced through a collaborative initiative led by Sun Microsystems, with participation from the key players in the enterprise computing space. These players include IBM Corporation, BEA Systems, Oracle Corporation, and many others.

The inclusion of multiple vendors in the specification process ensures that vendors will be able to successfully layer the J2EE platform on top of established and proven existing solutions. As a result, today there are over twenty-five (25) different server-side platforms, supporting EJB and other J2EE APIs that are evolving into full J2EE platform-enabled products.

The open collaborative specification process of J2EE also reflects the collective knowledge and expertise of a variety of enterprise computing vendors. This means that product solutions will likely be available to fit any business' needs. Corporations can mix and match compatible J2EE platform-based products from different vendors to form best-of-breed solutions that are customized to specific enterprise requirements.

By way of comparison, Microsoft's Windows DNA architecture is defined and implemented by a single vendor – Microsoft. By choosing a proprietary architecture that does not have widespread industry support, customers must limit their choices to a single vendor's implementation. This prevents customers from combining different vendors' products in the future; products that may be optimized for a business' specific needs.

Preservation of Existing IT Investments

As corporations are forced to adapt to new business requirements, it becomes essential to leverage investments in existing enterprise information systems, rather than rewrite entire business solutions from scratch. A server-side platform is needed that provides mechanisms to build on existing systems in an evolutionary (rather than a revolutionary) manner.

J2EE layers on top of existing systems, allowing a corporation to fully leverage past investments. Almost any existing investment can be leveraged, such as one using BEA Tuxedo, IBM CICS, IBM Encina, Inprise VisiBroker, or Netscape Application Server. This is possible because J2EE has widespread industry support and buy-in from key enterprise computing vendors. Each of these vendors is providing migration paths for existing customers to enter the portable J2EE space without abandoning existing investments. Existing operating systems and hardware can also be preserved, because J2EE platform-based products are available on almost all operating systems and hardware configurations.

The J2EE preservation of existing IT assets has a threefold benefit:

Applications can be leveraged. Existing code can be re-used in a J2EE solution, enabling an application to be layered rather than rewritten. This reduces development time, speeds time to market, reduces development costs, and keeps projects on a reasonable budget. It also reduces risk, because a corporation can use a proven solution that has worked in the past, rather than tearing out existing systems and rewriting them from scratch.

Existing products and license agreements can be leveraged. If a corporation has already purchased license agreements or maintenance agreements for legacy technology, such as middleware, operating systems, or hardware, those licenses can be re-used in a J2EE environment, lowering project costs and reducing training time.

Fewer vendor dependencies. Since the existing vendor is the one providing a J2EE platform-enabled product, there is only one vendor in the equation. There is no need to rely on two different vendors, with a bridge between the two vendors' products. This reduces the application's complexity as well as the overall risk level.

By way of comparison, Microsoft's Windows DNA requires customers to purchase and install Windows 2000, as well as likely purchase new hardware. This does not provide customers the opportunity to make use of license or maintenance agreements with the majority of existing vendors, increasing project costs.

It should be noted that Windows DNA does provide connectivity to existing systems. However, a customer using such connectivity is now dependent on two vendors, Microsoft and the existing vendor. A single-vendor solution is generally much more manageable and reliable than a two-vendor bridged solution.

Preservation of Future IT Investments, and Adaptability to Change

Today, a corporate IT department's nimbleness and adaptability is a source of competitive advantage for the corporation in the marketplace. By rapidly evolving applications to respond to changing market conditions, a corporation can leapfrog the competition through faster time to market, or quickly respond to steps taken by its competitors. By designing commerce systems to be adaptable to increased user load or transactional needs, an IT department can scale a deployment as demanded by the marketplace.

J2EE enables corporations to be highly adaptable to change. Because every J2EE platform-based product is guaranteed to support the minimum J2EE services, a corporation can build a well-written J2EE compliant application once, and make environment decisions at deployment time. If the underlying platform's functionality, scalability, or availability ever proves to be insufficient due to new market demands, the customer has choice. If the customer designs a J2EE application well and remains loyal to the standard, the system can be scaled over time by replacing underlying middleware, operating system, or hardware without significantly modifying applications. This level of protection for future IT investments is a compelling value proposition. Response time to market needs is minimized, enabling the corporation to move at speeds demanded by the Internet.

By way of comparison, Windows DNA does not offer the J2EE platform's level of adaptability to change. A customer adopting Windows DNA is tied to Microsoft's middleware and operating system implementations. If functionality, scalability, or availability is not satisfactory, the customer is out of luck. The underlying middleware or operating system cannot be changed without significant changes to applications. There is a limited menu of hardware options available as well. This may cause significant delays in a corporation's ability to respond to changing market conditions.

Time to Market

When developing a commerce solution in today's marketplace, a few months of time is an eternity. Missing a small window of opportunity is the difference between a corporation that is first to market, and a corporation that must play catch-up for years.

One way to speed time to market is to choose a server-side platform that allows rapid application development. Both J2EE and Windows DNA offer corporations the ability to externalize their common server-side plumbing tasks to a middleware vendor. This enables developers to exclusively focus on creating business logic, lowering development time.

However, advanced J2EE platform-based products provide a greater range of middleware services than Windows DNA provides. Some of these differences are fundamental to the J2EE architecture, and some are made available because J2EE enables vendors to differentiate through various features while remaining compatible with the platform. Examples of tricky middleware services offered by advanced J2EE vendors that are not found in Windows DNA include the following:

State-management services enable developers to write less code and not worry about managing state, resulting in a higher degree of rapid application development.

Persistence services enable developers to write applications without coding data access logic, resulting in leaner, database-independent applications that are easier to build and maintain.

Distributed shared data object caching services enable developers to craft high-performing systems, enhancing the overall deployment's scalability significantly.

By choosing a J2EE platform-based product that implements these services, a corporation can realize a significantly faster time to market, and a higher scalability level than with a Windows DNA-based solution. For a complete description of these differences, please see the accompanying whitepaper, *The Technical Benefits of EJB and J2EE Technologies over COM+ and Windows DNA.*

The J2EE platform also offers significant time-to-market advantages to vendors of off-the-shelf e-commerce applications, such as SAP, Peoplesoft, Trilogy, Oracle, Tradex, EC Cubed, Integral, I-mind, or Intershop. A vendor that chooses the J2EE platform can build portable applications that are deployable on any of its customers' target platforms, including Windows, UNIX, and legacy systems, as well as any J2EE platform-based product. A vendor choosing Microsoft's platform can only support the subset of customers that are deployed on Windows. J2EE offers the ability to write applications that are independent of specific technologies, such as databases, operating systems, or hardware; this is necessary for a vendor that has a broad, heterogeneous customer base. J2EE offers a greater number of underlying platform services than Windows DNA, resulting in speedier application development and a faster time-to-market for software vendors.

Platform Maturity

The server-side component services in Windows DNA and COM+ have been evolving since Microsoft Transaction Server (MTS) was released in December 1996. J2EE and EJB products have been evolving since the spring of 1998.

This may seem to depict Windows DNA as the mature platform. However, in reality today's high-end J2EE platform-based products contain fundamental transactional logic that has been in production for many years. Vendors providing commercial implementations of the J2EE platform (such as IBM and BEA) have long, proven histories of delivering middleware products that enable high-volume transactional systems. Rather than construct a server-side platform from scratch (as Microsoft had done with Microsoft Transaction Server), many vendors implementing the J2EE platform are developing a J2EE platform-based portability layer over proven existing technologies. This includes CORBA products, transaction processing monitors, databases, and/or application servers that were formerly proprietary. The technology behind many J2EE platform-based products is running the bulk of today's commerce transactions, and was in production long before Microsoft entered the server-side commerce space. Examples of J2EE product vendors that are building on proven technology include:

IBM's *WebSphere* builds on logic from IBM's *MQSeries*, *CICS,* and *Encina* products, which are driving the transactions in most major mission-critical systems today.

BEA's *WebLogic* has been on the market since Microsoft first offered their MTS product in 1996. WebLogic provides connectivity to *M3* and *Tuxedo*, enabling IT departments to leverage transaction processing monitor technology with a strong history of reliability.

Oracle's *Oracle 8i* contains support for database services, server-side components, and Web services in a single physical tier. *Oracle 8i* builds on Oracle Corporation's market-leading database technology and online transaction processing technology.

Netscape's *Netscape Application Server* (future: Sun-Netscape Alliance's *iPlanet*), based on the *Kiva* application server, has been available since 1996 and is powering many large commerce sites today.

IONA's *iPortal Application Server* is based on *OrbixWeb*, a CORBA product that has a long deployment history and provides a strong lineup of advanced middleware services.

Inprise's *Inprise Application Server* is based on *Visibroker*, a CORBA product that has been largely adopted by the industry and deployed at many large corporate sites.

Thus, because an IT department can leverage existing and proven technologies that reside behind a J2EE platform-based portability layer, an IT department can have the best of both worlds: a non-proprietary, high-end transaction processing system. This reduces an IT department's risk factors substantially.

Scalability

Corporations must choose a server-side platform that will provide unmatched scalability to accommodate the legions of new customers that will perform commerce on their systems.

Microsoft's Windows 2000 Datacenter Server, Microsoft's most scalable offering slated for February of 2000, is reported to scale to 16 processors per machine, or 32 processors through OEMs. By clustering several Windows 2000 machines, one can theoretically scale into the hundreds of processors. To achieve that level of scalability, the underlying server-platform must have the functionality to fairly spread client requests across machines. Microsoft's *component load-balancing* technology is intended to spread such requests across COM+ business components.

Unfortunately, Microsoft has recently announced[1] that component load-balancing technology has been removed from the February 2000 release of Windows 2000 due to lack of functionality. This means that a middle tier involving COM+ is limited to a single physical machine, which does not provide the scalability required to service very large commerce sites.

By way of comparison, J2EE platform-based applications can be deployed into a variety of operating systems. As an example, J2EE platform-based applications can be deployed into high-end UNIX and mainframe systems, which can support from 64 to 256 processors per server machine. Vendors in the J2EE space offer a wide variety of load-balancing strategies that eliminate bottlenecks in systems, and allow for a federated deployment involving many servers. Deployments can reach into the thousands of processors, resulting in a highly scalable system that can meet the needs of tomorrow's commerce applications.

This scalability example illustrates the danger with choosing a proprietary architecture such as Windows DNA. Customers do not have a choice but to wait for their vendor to implement a desired feature. With an open architecture like J2EE, a customer enjoys a choice of products while preserving their investment in application development. If the J2EE platform-based application is designed correctly, the customer can switch between J2EE platform-based product vendors to address scalability or functionality needs with minimal porting costs.

High Availability

A server-side platform must be highly available 24x7x365 to service the needs of a corporation's customers and partners. Because the Internet is global and ubiquitous, even planned downtime during the evening can result in significant lost business. Unplanned downtime during peak business hours can be disastrous, and will become amplified even more as the Internet becomes a primary vehicle for conducting business.

The J2EE platform is deployable onto reliable operating environments that support high availability. While some J2EE deployments will be on Windows, customers also have the choice of deploying J2EE solutions on more robust operating system environments, such as Sun's Solaris Operating Environment or IBM OS/390. The most robust operating system environments can reach levels of 99.999% availability, or as little as 5 minutes of downtime per year[2]. This level of availability is ideal for businesses building mission-critical commerce systems.

Deployments based upon Microsoft technology have historically been unable to reach this level of availability. Windows DNA is tied to the Windows operating system, which has been criticized in the past for unreliability. Microsoft has been touting Windows 2000 as highly available, but whether Microsoft can approach the availability levels offered by traditional enterprise operating systems remains to be seen. Although some early adopters will embrace Windows 2000, mainstream customers with mission-critical commerce sites are likely to tread a more conservative path and take a “wait and see” approach.

Total Cost of Ownership

When purchasing a server-side platform, one must consider the total cost of ownership. This is defined as the price of the server platform, the cost to develop and evolve a solution on that platform, maintenance fees, lost deployment time due to scalability or instability problems, and lost customers due to platform shortcomings. A low total cost of ownership is a necessity for any high-volume, mission-critical commerce site.

A wide variety of J2EE platform-based products are available for purchase, with price points varying dramatically, enabling a corporation to choose the platform that meets its budget and desired service level. Costs range from completely free (open source J2EE platform-based products, such as Bullsoft's *JOnAS*), to $2500 per machine (Valto's *Ejipt*), to $35,000 per CPU[1] (IBM's *WebSphere Enterprise Edition*). Initial pricing data from Microsoft indicates a price point of $3999 per machine for Windows 2000 Advanced Server, plus $1999 for an Internet Connector yielding unlimited authenticated Internet users[2].

Thus, Microsoft's current offering price point falls in the low to mid range of vendors implementing the J2EE platform. Microsoft has not announced pricing for Windows 2000 Datacenter Server, nor Windows 2000 AppCenter Server (slated for release in mid 2000), which are Microsoft's more scalable offerings. So for now, no price comparison can be drawn between Microsoft's high-end platform and the most scalable J2EE platforms. In the future, due to competition we expect that over time each of the middleware vendors, including Microsoft, will reach similar price points based upon service levels offered.

Regardless of the outcome of price point wars, the J2EE architecture does promote a significantly lower total cost of ownership than Windows DNA. This is because an IT shop can fully utilize existing investments in applications, middleware technology, and operating systems, which reduces the funds needed for a new commerce solution compared to a Windows DNA solution. J2EE platform-based application development costs, both initial and ongoing, are also significantly lower than Windows DNA-based application development costs. The reason is J2EE platform-based products can implicitly handle a variety of tricky middleware on behalf of the programmer that Windows DNA cannot handle (see accompanying whitepaper, *The Technical Benefits of EJB and J2EE Technologies over COM+ and Windows DNA*, for further details). An application developer writing to the J2EE standard can spend more time focusing on application's business logic, resulting in increased productivity.

Choice vs. Single Vendor Lock-In

One large difference between Windows DNA and J2EE is that Microsoft's solutions are proprietary, eliminating competition by raising barriers of entry (resulting in a monopoly), whereas J2EE lowers barriers of entry to competition due to open standards (resulting in a free market).

The advantage of competition is that price points are reduced, best-of-breed products emerge, and customers have choice and flexibility rather than being locked into one vendor’s solutions, reducing a customer's risk levels significantly. The disadvantage of competition is that an open standard requires extra work to insure compatibility. Open standards such as SQL and CORBA have been plagued in the past for lack of this very insurance, and as a result, proprietary products have emerged.

Sun Microsystems has responded to this challenge by including a deliverable with J2EE, called the *J2EE Compatibility Test Suite*. A middleware vendor must successfully pass the test suite to receive the J2EE certification logo. This ensures that J2EE application code can be developed once, compiled, and then deployed into any J2EE platform-based product. This also promotes choice and avoids vendor lock-in.

Indeed, the very notion of a test suite has already been used successfully by Sun Microsystems to ensure compatibility with the Java 2 Platform, Standard Edition (J2SE). Today, one can create a J2SE-based application once and then deploy that application into any compliant Java Virtual Machine. This works very well today, due to the J2SE test suite. This same concept holds true with J2EE; the J2EE compatibility test suite is the key to bridging incompatibilities between server-side platforms.

Finally, the reader should note that because of the intense competition in the J2EE space, vendors will provide facilities that go beyond the J2EE standard. It is then up to developers to use judgement in deciding when it is appropriate to use such features.

Support for Heterogeneous Environments

Many large corporations have a heterogeneous server-side deployment environment, with a wide variety of operating systems, hardware, and middleware products. A heterogeneous environment can also emerge as the result of several distinct projects at an organization, or perhaps due to a merger or acquisition with another corporation that has a different deployment environment. Portability and interoperability across heterogeneous enterprise systems is essential in such scenarios.

J2EE enables development of portable applications that can be deployed into heterogeneous environments. J2EE-based applications are independent of any particular operating system, middleware product, or hardware. As such, well-designed J2EE-based applications can be developed once and deployed into a variety of platforms, which is critically important in the typically heterogeneous corporate computing environment. The J2EE standard also enables corporations to purchase J2EE-compliant third-party components off the shelf and deploy them into a heterogeneous environment, saving development costs over building a complete solution in-house.

By way of comparison, Windows DNA applications (including COM+ components) can only be deployed on Windows 2000, and cannot be deployed onto existing systems. This proposition is hard to swallow for any corporation that has a heterogeneous environment, because applications are not portable across the heterogeneous enterprise. Furthermore, by choosing Windows DNA, a corporation is forced to inject a new operating system into an existing heterogeneous environment, which results in an even more complex system. Components purchased off the shelf can only be deployed in a Windows 2000 platform, restricting their usage as well.

Interoperability, or the communication of different systems, is also critical in a heterogeneous environment. Two different J2EE systems, each an installation of the same J2EE product, can communicate quite well today. Two different vendors' products can also communicate, but transaction and security information will not be interoperable across vendor products due to a lack of standardization. A new version of the J2EE specification will standardize a protocol that will enable full interoperability between J2EE-based platforms, and will not affect existing application code. Microsoft is not expected to comply with this protocol, so Microsoft's server-side platform will remain proprietary and non-interoperable with other vendors' products, while the J2EE platform-based products will become interoperable across heterogeneous environments.

Maintainability

Often times, corporations judge the cost of a deployment in the span of a single project. In reality a deployment's total cost includes the maintainability of the developed solution over time.

Applications developed using J2EE are far more maintainable than their Windows DNA counterparts. The reason lies in the functionality of the base platform. J2EE platform-based products support a wide variety of component types and middleware services, which enables developers to focus on the business problem at hand, rather than on middleware. Developers can harness the underlying platform to build flexible, maintainable enterprise systems that can be adapted to new middleware, databases, or database schemas without significant modification of a well-designed J2EE application.

In comparison, Windows DNA does not offer the same breadth of middleware functionality offered by advanced J2EE platform-based products. More code is required in the overall system, which increases application complexity, and does not accurately separate middleware plumbing from application business logic. The end application is then more difficult to maintain, especially as development teams change and/or turn-over as compared to a J2EE platform-based equivalent solution. For a full technical critique, please see the accompanying whitepaper, *The Technical Benefits of EJB and J2EE Technologies over COM+ and Windows DNA.*

Conclusions and Recommendations

In this whitepaper, we have explored the business benefits of J2EE over Windows DNA. The J2EE platform offers the following significant advantages over Windows DNA that are critical for mainstream businesses:

Industry support and choice. J2EE is the result of a collaboration supported by all major industry players, and reflects the combined expertise of seasoned middleware experts. Customers can choose J2EE products that have evolved from proven existing systems, and can mix and match products that are the most appropriate for the business problem at hand. By way of comparison, Windows DNA is a Microsoft-only solution, which locks customers in to a specific vendor implementation. Windows DNA does not have an open specification that enables a variety of vendors to offer solutions optimized for specific business problems.

Protection of past IT investments. With J2EE, a corporation can fully utilize existing middleware, operating systems, hardware, vendor relationships, license agreements, and developer skills, lowering project costs significantly compared to Windows DNA.

Protection of future IT investments. As business requirements change, well-written J2EE platform-based applications can be redeployed into different middleware products, reliable operating systems, hardware, and tools. This means customers can adopt new technology while preserving investments, reducing project risk factors considerably and resulting in a competitive edge. With Windows DNA, customers cannot easily adopt new technology, because they are bound to Microsoft's implementations.

Time to market, and future maintainability. J2EE platform handles many more complex middleware tasks than Windows DNA. This reduces a corporation's time to market as well as maintenance costs. Software vendors also experience a faster time to market with J2EE, because J2EE enables a vendor to author database-independent code, and easily support any customer's deployment environment.

Support for heterogeneous environments. Heterogeneous deployment environments can result from different projects being combined, from a merger, or from an acquisition. J2EE platform-based applications adapt well to heterogeneous environments, because they are largely independent of underlying middleware, operating system, or hardware. Windows DNA applications cannot be deployed into most heterogeneous environments because Windows DNA requires the Windows 2000 operating system. Microsoft does provide connectivity with existing systems, however this results in an even more heterogeneous system. There are now more vendors in the equation, increasing the complexity of the application, and increasing the number of risk variables in projects.

High availability and platform maturity. Customers embracing J2EE can lower their project's risks by choosing operating systems with a more proven reliability history than Windows. Customers can reduce their risks even further by selecting a vendor that builds in proven existing middleware technologies, such as IBM *Encina* or BEA *Tuxedo*.

Scalability. Customers can combine the most scalable middleware with advanced caching and state management, the most scalable operating system with high processor support, and the most scalable hardware with generous resources to reach a deployment configuration that offers much greater scalability than Windows DNA-based deployments offer today.

The Middleware Company is a group of distributed object professionals that provides expert-level training, consulting, and advice in EJB and J2EE Technologies. Services offered include:

- On-site training of development staff through interactive discussions and lab exercises
- On-site development of application prototypes
- Guidance when making an application server purchase decision
- A J2EE quickstart package, including each of the above services, designed to get a corporation up-and-running with J2EE in a matter of weeks
- Development of full-scale enterprise applications
- Business and technical whitepaper development

For further information about our services, please visit our Web site at http://www.middleware-company.com

References

[1] Source: Microsoft Corporation

[2] Source: IBM Corporation

[3] Source: Flashline.com application server comparison matrix

[4] Source: Microsoft Corporation

The Middleware Company believes the information contained in this document is accurate as of its publication date; such information is subject to change without notice. The Middleware Company is not responsible for any inadvertent errors.