.NET. WINDOWS. J2EE. LINUX. UNIX/MAC. THE WAR.

written by Manoj Khanna | February 3, 2003
.NET stands up against J2EE

Windows cheaper than Linux

Arguments supporting both platforms

- Regardless of which platform you pick, new developers will need to be

  trained (Java training for J2EE, OO training for .NET)

- You can build web services today using both platforms
- Both platforms offer a low system cost, such as jBoss/Linux/Cobalt for

  J2EE, or Windows/Win32 hardware for .NET.

- Both platforms offer a single-vendor solution.
- The scalability of both solutions are theoretically unlimited.

Arguments for .NET and against J2EE

- .NET has Microsoft's A-team marketing it
- .NET released their web services story before J2EE did, and thus has

  some mind-share

- .NET has a better story for shared context today than

J2EE

- .NET has an awesome tool story with Visual Studio.NET
- .NET has a simpler programming model, enabling rank-and-file

  developers to be productive without shooting themselves in the foot
- .NET gives you language neutrality when developing new eBusiness

  applications, whereas J2EE makes you treat other languages as separate

  applications
- .NET benefits from being strongly interweaved with the underlying

  operating system

Arguments for J2EE and against .NET

- J2EE is being marketed by an entire industry
- J2EE is a proven platform, with a few new web services APIs. .NET is a

  rewrite and introduces risk as with any first-generation technology
- Only J2EE lets you deploy web services today
- Existing J2EE code will translate into a J2EE web services system

  without major rewrites. Not true for Windows DNA code

ported to .NET.

- .NET web services are not interoperable with current industry

  standards. Their BizTalk framework has proprietary SOAP extensions and does

  not support ebXML.

- J2EE is a more advanced programming model, appropriate for

  well-trained developers who want to build more advanced object models and

  take advantage of performance features

- J2EE lets you take advantage of existing hardware you may have

- J2EE gives you platform neutrality, including Windows. You also get

  good (but not free) portability. This isolates you from heterogeneous

  deployment environments.

- J2EE has a better legacy integration story through the Java Connector

  Architecture (JCA)

- J2EE lets you use any operating system you prefer, such as Windows,

  UNIX, or mainframe. Developers can use the environment

they are most

productive in.

- J2EE lets you use Java, which is better than C# due to market-share

  and maturity. According to Gartner, there are 2.5 million Java developers.

  IDC predicts this will grow to 4 million by 2003. 78% universities teach

  Java, and 50% of universities require Java.

- We would not want to use any language other than C# or Java for

  development of new mission-critical solutions, such as a hacked

  object-oriented version of C, VB, or COBOL.

- We are finding most ISVs and consulting companies going with J2EE

  because they cannot control their customers' target platforms. We believe

  this application availability will result in J2EE beginning to dominate more

  and more as time goes on.

In conclusion, while both platforms will have their own

market-share, we

feel most customers will reap greater wins with J2EE. We feel the advantages

outweigh those offered by Microsoft.NET. That is our preferred architecture,

and we stand behind it.

As we know the field is the real test and these head to head comparisons

will be more believable by the end of 2003 when the real cost of people and

maintenance and interoperability and up time and down time.