Web Services: .Net vs. J2EE

Joseph Williams, Ph.D.
Sun Microsystems, Inc.
500 Eldorado Blvd, Broomfield, CO, 80021 USA
joseph.williams@sun.com

Abstract—The ubiquity of web services argues for the application of the right tool for the right job at the right time for the right organizational context. .Net and J2EE present different views on solving many similar problems and some uniquely dissimilar ones. It isn’t an either/or proposition with respect to .Net or to J2EE. Rather, it is a problem of tools selection.

Web services are of interest in large part because they may well provide a common communications infrastructure that will be supported by all of the major application vendors and development shops. There are still a number of major shortcoming to web services that need addressing before we can anoint web services as a lingua franca (security and workflow management are just two obstacles) and it is not at all clear at this point that SOAP, XML, HTTP, UDDI, and WSDL will be the ultimate enabling platforms for web services (alternatively, consider JINI or REST). However, there is considerable momentum behind web services so, although web services may not yet be a mature technology, they have clearly “arrived.” Unlike previous attempts at interoperability (e.g., DCOM, CORBA), web services take a “loosely coupled” approach that is appealing at many levels.

Development and deployment of web services requires no specific underlying technology platform. However, the 600-pound gorillas on the scene are Microsoft’s .Net framework and the Java Community Process’ (JCP) J2EE specification. Since Sun Microsystems invented the Java programming language and is one of its most vocal proponents the web services platform debate is often viewed as a Sun vs. Microsoft, J2EE vs. .NET confrontation. Indeed, there is some philosophical and business animus but it does not detract from the core commitments both companies (and, indeed, virtually all companies) have to making web services successful as common communications infrastructure. In point of fact, the realization of web services as a common communications infrastructure requires cooperation between the J2EE and .NET factions.

.NET is a Microsoft-centric approach to web services. It runs on a single platform (Windows) but it does natively support multiple languages (although each is a “value-added” purchase for the .NET development platform and C# is still the language of choice). J2EE, on the other hand, is a platform independent solution deployed in a single language (Java), although it does have support for other languages. J2EE is the more mature and proven technology. Microsoft has made a massive investment in its .NET framework and, in many respects, has staked the credibility and future of the company on .NET. Both J2EE and .NET present different challenges and opportunities in the application environment.

The development platform that should be adopted will depend on a number of factors, including the development/business environments, the exigencies being addressed, and the enterprise’s objectives for the project. As with any tools choice, J2EE vs. .NET is an “it depends” proposition.

In the final analysis, however, J2EE has a lot more going for it than does Microsoft’s .NET. By most accounts J2EE is the platform of choice for 65-70% of web services deployments. It has broad support from the major vendors (e.g., SAP, Oracle, IBM, HP, PeopleSoft, BEA) and provides the ISVs with the greatest platform flexibility. J2ME is also emerging as the standard for wireless and mobile deployments and analysts such as IDC have speculated that web services could realize their greatest potential in the mobility arena.