

Interoperating J2EE and
Microsoft .Net Applications –
What Standards to Adopt?

An Oracle White Paper
October 2006

Interoperating J2EE & Microsoft .NET Applications

INTRODUCTION

Many developers and project teams today need to create applications and solutions that effectively use both J2EE and .NET based components. While there are a number of different approaches that can be taken to address this need, there are extensive sets of standards around Web Services that provide a flexible and adaptable mechanism for accomplishing this effectively.

Oracle Fusion Middleware provides broad and deep support for these standards, but more importantly focuses on delivering real world interoperability by ensuring that Oracle products and Microsoft products do truly interoperate through these standards. There are many mutual customers with which we have worked and gathered best practices and experience that have been folded back into our products as well as creating many developer materials and examples that help developers and architects implement J2EE and .NET projects.

By focusing on common standards, the most flexible and adaptable approach to interoperability is addressed, while Oracle's focus on actual product interoperability through testing and supporting mutual customers ensures real world interoperability.

EVOLVING REQUIREMENTS

Traditionally, we would choose the same tool and language in which an application was originally developed in order to extend that application or make modifications to it. However, there is an increasing need to be able to cross existing technology stacks as we look to develop newer SOA-based processes, composite applications, and improved user interface implementations. This need is driven both by the fact that we need to leverage already existing functionality that was implemented in both J2EE and .NET, as well as the fact that different development teams will continue to choose the tools with which they are most familiar. Heterogeneity exists today and will likely to continue to exist in our IT environments.

Some examples of projects where J2EE and .NET need to be interoperable are:

- Using an orchestration engine that provides BPEL support in order to create an integrated process that interacts with services that have been implemented in J2EE and .NET. The BPEL engine itself was likely implemented in J2EE.

- Building a new .NET based application that incorporates functionality implemented in J2EE. The mirror situation is just as likely, building a new application using Java Server Faces and J2EE that incorporates .NET-based functions.
- Creating web Portals that include portlets and other functionality created with both J2EE and .NET.
- Implementing a smart document solution where the document is a Microsoft Office document that calls BPEL-based processes or J2EE-based services.

As we implement these types of projects involving enterprise services and SOA, interoperability requirements go beyond simple data exchange to also require secure and reliable communications, support for both synchronous and asynchronous messaging, appropriate authorization and authentication, and overall management.

STANDARDS

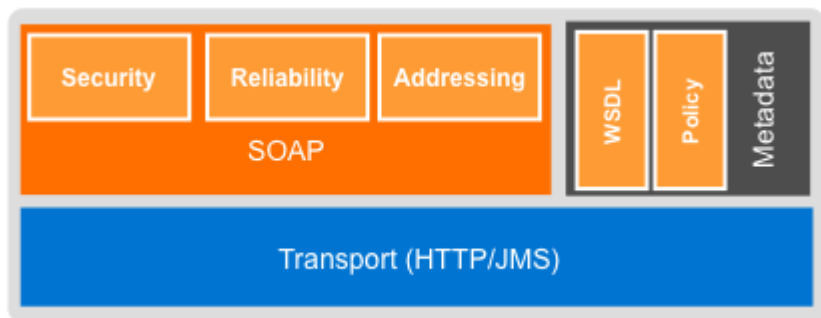


Figure 1. Key Areas Addressed by Web Service Standards

While there are a variety of mechanisms for achieving some level of interoperability between J2EE and .NET applications, Web Service standards provide the most flexible and least-custom approach for interoperability. Some of these other mechanisms include:

- Web Communication Protocol are used to facilitate exchanges: HTTP, HTTPS, and even Mails and instant messaging
- HTML and XML (Ajax for example) also facilitate the exchange of data between J2EE and .NET applications
- WSRP can also be used to interoperate portlets and portals that may be built on J2EE and .NET

However, the broadest, most comprehensive set of standard that make the integration of J2EE and .Net possible are related to Web Services:

- WSDL: provides a common description of a service. WSDL binding describes how a messaging protocol – e.g. SOAP – is bound to the service.

There are four styles for describing the binding: RPC/encoded, RPC/literal, Document/encoded, Document/literal.

- SOAP: common messaging protocol. Most serious applications have a need for using attachments when carrying large amount of data, either binary or other XML documents. SwA (SOAP with Attachments) and SOAP Message Transmission Optimization Mechanism (MTOM) address this need.
- BPEL: provides a description of processes which orchestrate across individual web services. This provides great abstraction of the process definition from the technology infrastructure used to implement individual web services (i.e. J2EE, .NET, legacy, etc)
- WS Addressing, WS-Security, WS-ReliableMessaging, WS-Policy: major standards that enable enterprise implementation of services and SOA.

WS-Addressing provides a standard for describing the mechanisms by which the information needed to interact reliably with asynchronous Web services can be exchanged. For example, a BPEL process which needs to asynchronously interact with .NET-based web services, and needs to know how to call back a .NET web services, when the process has completed, or vice-versa. In the long term, this promises seamless interoperability, even for asynchronous services, between clients and services implemented on different technology stacks.¹

The main purpose of WS-Addressing is to incorporate message-addressing information into SOAP messages (for example, where the provider should send the response). SOAP is an envelope-encoding specification that represents Web service messages in a transport neutral format. However, SOAP itself does not provide any features that identify endpoints. The usual endpoints, such as message destination, fault destination, and message intermediary, are delegated up to the transport layer. Combining WS-Addressing with SOAP creates a complete messaging specification. WS-Addressing specifies that address information be stored in SOAP headers in an independent manner, instead of embedding that information into the payload of the message itself. WS-Addressing is complemented by two other specifications, WS-Addressing SOAP Binding and WS-Addressing WSDL Binding, which specify how to represent the WS-Addressing properties into SOAP and WSDL respectively.²

The main goal behind WS-Security is to enhance SOAP messaging by providing message integrity, confidentiality, and authentication. WS-Security also serves as the base standard to other WS-* protocols, such as WS-SecureConversation, WS-Trust, and WS-Federation. The key function of WS-Security is to provide a general purpose mechanism to associate security tokens with SOAP messages. In this scenario, security-related information is carried in the message itself, instead of in

^{1,2,3} Web Services and SOA – Practical Interoperability Approaches, WS-Security and WS-Addressing Explained, Clemens Utschig, Heidi Buelow, and Jesus Rodriguez, September 2006

external artifacts. This is known as message-based security and can be used in conjunction with a transport layer security such as SSL.³

WS-Policy provides a general purpose model and syntax to describe and communicate the policies of a Web service. WS-Policy assertions express the capabilities and constraints of a particular Web service. WS-PolicyAttachments defines several methods for associating the WS-Policy expressions with Web services (i.e., WSDL). Web Services Security Policy Language (WS-SecurityPolicy) specification indicates the policy assertions that apply to Web Services Security: SOAP Message Security, WS-Trust, and WS-SecureConversation.

The challenges in using these standards for interoperability are: 1) mutual agreement on how to use these standards, i.e. Document/literal vs. RPC/encoded 2) choosing standards that are well supported in both the J2EE and .NET stacks 3) ensuring common implementation of those standards 4) using tools and techniques appropriately to achieve the best level of interoperability. For example: up to four different WS-Addressing versions are commonly used – WS-Addressing March 2003, March 2004, August 2004 versions, WS-Addressing 1.0 (W3C approved).

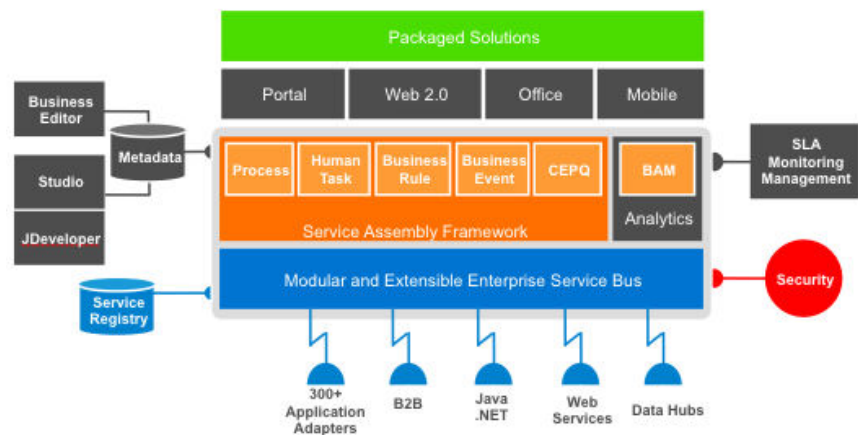


Figure 2: Oracle Provides a Comprehensive Approach to .NET Interoperability

ORACLE'S PRODUCT STRATEGY

Oracle has a comprehensive approach to providing real world interoperability between J2EE and .NET. Oracle Fusion Middleware goes beyond the “we can both claim to support the same standards” level of interoperability with a strong commitment to working with Microsoft in the development of common standards, implementing extensive support for a wide set of standards within the product to ensure developer options, actual testing of our products with Microsoft products to confirm support, and working with many real customers who have successfully implemented actual projects.

Oracle also has joint support agreements with Microsoft so that mutual customers are well supported. Oracle also closely follows new developments from Microsoft such as the impending release of key updated products like – Microsoft Vista,

Microsoft Office 2007, Windows Presentation Foundation, Windows Communication Foundation, Windows Workflow Foundation, etc – to ensure interoperability stays current.

Active Involvement in Standards Development

Oracle actively participates in industry standard bodies from core standards groups like the W3C and the JCP to business oriented standards groups like OASIS and OAG. The implementation of open standards enables customers to realize significant cost savings and business benefits.

Deep and Broad Product Support

Microsoft provides a wide array of products and technologies, of which .NET is a core foundation, and Oracle Fusion Middleware provides extensive support for leveraging these products across a wide range of middleware capabilities.

- *Microsoft .Net and Visual Studio .Net*: Developing applications faster and easier while allowing organizations to effectively leverage both .Net and Java tools and frameworks.
- *Microsoft Window Server System Products*: Effectively working with products such as Microsoft SQL Server, Microsoft BizTalk, Microsoft SharePoint, etc. to create an enterprise-wide Service-Oriented Architecture (SOA).
- *Microsoft Security and Directory Services*: Enabling an unified, enterprise-wide security infrastructure that ensures compliance, simple user management, and effective policy enforcement.
- *Microsoft Office*: Enabling the effective utilization of Microsoft Office – Word, Excel, PowerPoint, Outlook, InfoPath –with enterprise applications, Web Services, and custom applications where there are benefits of using Microsoft Office in the context of enterprise applications and solutions.

Oracle Fusion Middleware effectively works with Microsoft products in each of these areas, which when combined with Oracle Fusion Middleware’s extensive support for a wide range of applications, industry standards, and other vendor products enables the delivery of applications and solutions that accommodate the wide heterogeneity typically found in most enterprises.

Real World Testing

There is a strong focus on ensuring that Web Services/Protocols standards that are jointly supported by Oracle and Microsoft do actually deliver interoperability. Oracle Application Server 10g is tested with Microsoft .NET products to confirm this interoperability. Oracle conducts ongoing tests as part of our development process to ensure this interoperability. In addition, Oracle participates in open industry interoperability tests called “PlugFests”; Microsoft has often hosted these at their facilities.

Best Practices

Oracle has worked with many customers in actual implementations involving J2EE and .NET, from which many best practices have been incorporated back into the products as collected in materials to support developers and architects. Among these materials, Oracle has produced a book called “SOA Best Practices – BPEL Cookbook” that provides advanced BPEL concepts and best practices for development, deployment, and administration from the architects implementing them in real-world applications, as well as a Developers Guide to Microsoft Interoperability.

CONCLUSION

Developers and project teams often need to create applications and solutions that effectively use both J2EE and .NET based components. Taking advantage of the extensive sets of standards around Web Services provides a flexible and adaptable mechanism for accomplishing this effectively.

Oracle is actively involved in the development and ratification of these standards in conjunction with other major vendors in the industry, and is in the forefront in the actual implementation of these standards within Oracle products as well as successful adoption in real world situations with customers and partners.

The focus is very much on practical, successful interoperability between J2EE and .NET across a range of commonly encountered use cases. This success is built on a foundation of broad and deep product capabilities combined with best practices and architectural expertise gained in actual implementations.



White Paper Title

October 2006

Author:

Contributing Authors: Frank Knifsend, Regis Louis, Nilesh Junnarkar, Tugdual Grall, Heidi Buelow, Clemens Utschig, Sachin A. Agarwal, Jesus Rodriguez

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

oracle.com

Copyright © 2006, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.