J2EE, .NET, or Open Source Software – a Matter of Choice or an Integration Challenge?

The Middleware Platform Choice and How It Affects Your Business

Many IT departments are now faced with the decision whether to continue building applications on a J2EE architecture from SUN Microsystems, endorse the .NET architecture from Microsoft, look for less expensive alternative such as Open Source Software or accept the reality of multiple architectures and hope that internal IT can achieve required level of interoperability.

This is a major decision which will affect companies for many years to come, not only in terms of the efficiency of IT, but might possibly become a key factor for the productivity of the company as a whole.

Companies need to anticipate significant implications, reaching far beyond the core Middleware technology. The impact includes operating system choices, development languages and tools, availability of skilled resources, time to market and IT direction as a whole. This decision will also impact purchasing decision of new packaged applications, interoperability with legacy systems as well as with business partners.

While both the J2EE and .NET platforms cover many technologies and are marketed as a one-stop shopping experience to IT management, for many companies, however, this will not be a simple binary choice. In larger companies, coexistence of both platforms is becoming a reality, whereby choices are being made on a case-by-case basis. In addition, Open Source Software has become more and more of a comprehensive platform for application development and deployment. It now includes functionality that can replace products provided by the traditional J2EE vendors as well as Microsoft.

As a result, companies have to worry about integration of new business applications that are built and deployed on the different platforms. However, instead of doing integration “after the fact”, a more prudent integration approach is emerging in form of the Service Oriented Architecture (SOA). A SOA based on Web Services can bring both platforms together. The SOA ensures that services encapsulate platform dependent implementations and standardized protocols make the services accessible across the platforms.

This seminar provides an in-depth comparison of the two platforms, cutting through the standard marketing materials offered by both camps. It also separates promises from reality and helps attendees to distinguish between features that are available today versus those yet to be implemented. The seminar also addresses the complex issues of “co-opetition” and interoperability between the platforms, from taking a high-level architectural view to the feasibility of a low-level interoperability at the component level.

Get the Competitive Advantage at a Special Seminar

Don’t just take our word for it, take us up on it by coming to a seminar we’re offering on J2EE, Microsoft .NET and Open Source Software and what it means for your business. You’ll discover that the more you know about the potential of these two competing technologies, the better equipped you will be to make the right choices between them, and how to combine and integrate them with the rest of your application environment.

This special seminar conducted by ISG starts with a review of the requirements for building and deploying enterprise applications. This provides the participants with the necessary context to understand the essential concepts, features and services of J2EE and .NET. Both platforms are then compared in-depth, covering all major criteria that can determine the success – or failure - of any enterprise application. This comparison is the main focus of the seminar and it is done from a practitioner's perspective - a key differentiator of ISG's seminar.

Benefits of Attending

- Differentiate between the major approaches employed by J2EE and .NET in building enterprise applications.
- Learn how J2EE and .NET can transform your approach to building, integrating and deploying enterprise applications, giving you a competitive advantage.
- Understand how the two technologies address the scalability and reliability challenges and which platform is best suited to your environment.
- Evaluate the state and maturity level of the competing platforms.
- Discover the platforms breadth of support for XML and Web Services and what the key differentiators are.
- Learn about the different options that J2EE and .NET offer for the integration of your new component-based applications with your back-end legacy and packaged applications.
- Understand the impact that the platform selection will have on your IT organization, including migration from previous platforms, skill requirements, technology choices and vendor lock-in.
- Determine a strategy for integrating both platforms at different levels – from interoperability to shared architectures.
- Get an overview of how the Open Source Software movement impacts the platform choices.
- Understand the Total Cost of Ownership (TCO).

Who Should Attend

- IT professionals who would like to migrate application development to a component paradigm.
- IT Managers and IT Strategists with a technical background who need to see how J2EE and .NET fit into their enterprise.
- Architects and developers who want a detailed look at the two competing architectures.

Prerequisite: This class does not require attendees to be knowledgeable in any specific technology; however, a general understanding of distributed applications and object-oriented concepts will be beneficial

Seminar Outline

1. **Introduction: ISG Overview and Agenda**

2. **Setting the Stage: From Middleware to Web Services**

   c) Middleware Evolution

   c) From Objects to Components to Services

   c) Service Oriented Architecture (SOA)

   c) Web Services

3. **Requirements for Development and Integration of Enterprise Applications**

   a) Development vs. integration

   b) High-level application architecture

   - Multi-tier application architecture

   c) Platform requirements

   - Presentation layer
   - Business logic layer
   - Data access layer
   - Integration layer
   - Common services layer
     - Horizontal middleware services
     - Horizontal business services

4. **J2EE Overview**

   a) Introduction to Java 2 Enterprise Edition (J2EE) - overview

   b) Servlets and Java Server Pages (JSP)

   c) Java Server Faces (JSF)

   d) Java Database Connectivity (JDBC)

   e) Java Remote Method Invocation (RMI)

   f) Java Naming and Directory Interface (JNDI)

   g) Java Messaging Service (JMS)

   h) Java Transaction Service (JTS)

   i) J2EE Connector Architecture (J2CA)

   j) Enterprise Java Beans (EJB)

   k) Web Services in J2EE

5. **.NET Overview**

a) Introduction to COM

b) COM+ overview

c) The .NET Framework

- Common Language Runtime
- Supported programming languages
- ASP.NET
- Web Services in .NET
- ADO.NET
- Remote objects
- Interoperability between COM and .NET

6. **Comparing and integrating J2EE and .NET**

a) Basic concepts
- Language and OS Support
- JVM vs. CRL
- Java Community Process (JCP) vs. Microsoft Shared Development Process (SDP)

b) Core platform capabilities
- Performance
- Scalability (clustering, load balancing, resource management, activation policies, network connection pooling)
- Availability / failover
- Security
- Naming
- Communication models (synchronous, asynchronous, event driven)
- Persistence
- Data access
- State and session management
- Distributed transaction support
- Management and monitoring

c) Comparing platform tiers

- Presentation tier
- Portal integration
- Legacy application integration
- Packaged application integration

d) Building a SOA with Web services

- Mapping SOA to Web Services
- Support for XML processing
- SOAP protocol support
- Interface definition (WSDL), client vs. server implementation
- Registration (UDDI and other registries)

e) Levels of integration

- Client/Server integration

- Interoperability based on Web Services
- Common Frameworks
- Common Architecture
- Cross-platform development tools

f) Impact on the IT organization

- Platform maturity
- Portability and vendor lock-in
- Development tools
- Ease of use (development, deployment)
- Skill requirements and skill availability
- Alternative platforms
- Migration from previous platforms
- Total Cost of Ownership (TCO)

g) Conclusions

- When to use J2EE?
- When to use .NET?
- When to use both?

7. J2EE and .NET vs. Open Source Software (OSS)

a) What's wrong with J2EE and .NET?

b) The Open Source Software (OSS) model

c) The OSS landscape

d) Pros and cons of OSS

8. Future Outlook

a) Market penetration

b) Future Middleware Platform – trends, directions and outlook