

# Foreword

**F**ROM its inception, the Java™ platform has been focused on the Web. Now a new generation of Web applications—Web services—will play an important role in the way enterprises implement their next generation of services. The Java™ 2, Enterprise Edition (J2EE) platform will be the platform of choice for those enterprises that realize the key to Web services success is to combine the interoperability of Web standards with the portability of the Java platform. Today's J2EE applications are proof that the J2EE platform is the benchmark for both interoperability and portability of enterprise applications.

The world of the Web is bigger than any single operating system. Whether using the J2EE platform, a PHP hypertext preprocessor, the Perl scripting language, or any other open approach, developers are demanding technologies that give them freedom to apply their knowledge broadly across computing environments. J2EE developers routinely apply their development skills across operating systems, machine architectures, and application servers. For developers, portability means more than the ability to run the same service on multiple operating systems; it means using the tools they prefer and having the freedom to apply their skills across Solaris, Linux, Windows, AIX, and HP-UX.

The WS-I Basic Profile 1.0 is a proof point that the J2EE vendor community takes Web services interoperability very seriously. By default, J2EE 1.4 Web service clients and Web service endpoints will conform to the WS-I Basic Profile 1.0, taking the guess work out of getting Web services to talk to each other. With its full support of the WS-I Basic Profile, the J2EE platform is the most interoperable Web services platform ever.

J2EE applications use a mix of relational databases, JMS, and enterprise connectors to deliver sophisticated composite functionality. Web services joins the J2EE platform's already rich arsenal of integration strategies. Each integration technology has its strong points. Broadly, Web services should be used when maximum interoperability is desired. J2EE Connectors should be used when closely-coupled integration with an external system is needed. RMI-IIOP should be used when a rich, object-oriented distributed object model is needed. JMS should be used when a reliable, store-and-forward, publish/unsubscribe messaging system is needed to integrate asynchronous services. Many applications will use combinations of these integration technologies. For instance, many Web ser-

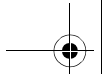
vices will be implemented as J2EE applications, where these application are business logic tiers that use Connectors to access existing Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM) systems.

J2EE developers are deploying Web services right now. For over a year now, J2EE vendors have been providing production support. Developers have been gaining experience with Java Web services through the Java Web Services Developer Pack. The JAX-RPC technology has been a Java standard since June of 2002; the J2EE 1.4 platform is the first release that makes JAX-RPC a required part of the platform. On November 12, 2003, the J2EE 1.4 platform was unanimously approved by the Java Community Process. Vendors will soon be competing vigorously to deliver the first 1.4-level products.

J2EE developers who are new to Web services can quickly pick up these Web service technologies by leveraging their existing knowledge. The J2EE 1.4 platform not only delivers comprehensive support for Simple Object Access Protocol (SOAP) 1.1 and Web Services Description Language (WSDL) 1.1, it tightly integrates these technologies with the existing servlet and Enterprise JavaBeans (EJB) component models. Both EJB components and servlets now support the full capabilities of JAX-RPC for building Web services. Developers can select between enterprise beans and servlets based on their component approach. Those developers that desire a more formal business component with declarative transaction support and method-level authorization can select enterprise beans, while those that like a less formal, HTTP request-handler style can choose servlets.

With each version of the J2EE platform, Sun Microsystems, Inc. has released the Java BluePrints, Sun's recommended practices for using the latest features of the J2EE technology. Java BluePrints includes a published book, a Web site (<http://java.sun.com/blueprints>), and a reference application. For the J2EE 1.4 platform, BluePrints focuses on Web services, providing detailed design considerations for taking advantage of the interoperability, portability, security, and maintainability features of the latest platform. Java BluePrints presents this essential information in the practical context of building an application—in this case, the Java™ Adventure Builder Reference application, which has been specifically architected to illustrate the practical use of Web services in the enterprise. This reference application highlights features of the J2EE 1.4 platform's Web services, such as service endpoint design, client design, and XML processing, just as Java™ Pet Store showcased features of the J2EE 1.3 platform, such as container-managed persistence, message-driven beans, and servlet filters.

Since Web services is a major extension of the Web computing model, developers need to understand the Web computing model technology itself as well as



how to effectively implement a Web service with the J2EE technologies. Java BluePrints provides an effective way to combine both into one easy-to-follow, step-by-step examination of the architecture and implementation of the Java Adventure Builder Reference application. The application has been written to shed light on implementation details of the Java BluePrints guidelines and J2EE design patterns.

Java BluePrints offers guidelines that will empower developers with a new dimension of flexibility—the ability to develop Web services without the walls imposed by operating systems and application server vendors. Once again, this version of the Java Blueprints is an essential source of implementation and architectural information for J2EE developers who will be using the J2EE 1.4 platform as their Web services platform.

*Mark Hapner, Distinguished Engineer  
Lead Architect, J2EE platform  
Sun Microsystems  
January, 2004*

