

Evaluating Oracle WebLogic Server and JBoss Application Server for Government Programs

*An Oracle White Paper
Updated December 2008*

Evaluating Oracle WebLogic Server and JBoss Application Server for Government Programs

Executive Overview	3
Introduction	3
Components of a Business Case for Software Projects.....	4
Evaluate Conformance to Government Directives	4
Evaluate Published Performance Benchmarks	5
Evaluate Native Operations, Administration, and Management Capabilities	6
Evaluate the Impact of the Organization and Governance Structure.....	8
Processes	9
Tools	10
Assets	11
People	12
Evaluate the Requirements: Open Systems, Open Standards, or Open Source?	13
Evaluate the Requirements: Open Source Tools or Open Source Functionality?.....	15
Evaluate the Role of the Application Server Within a Service-Oriented Architecture.....	16
Conclusion.....	17

Oracle WebLogic Server and JBoss Application Server in Government Programs

EXECUTIVE OVERVIEW

Large government organizations are evaluating how to measure the impact of and maximize the return on investment (ROI) from commercial off-the-shelf (COTS) and open source software within IT programs. Specifically, they are seeking guidance on how to define evaluation criteria for Oracle WebLogic Server and the open source JBoss Application Server. This white paper presents evaluation strategies designed to help large government organizations, individual program managers, and technical leaders create a business case that will rationalize enterprise application server infrastructure decisions.

In evaluating whether Oracle WebLogic Server or JBoss Application Server is the correct choice for a given program, government organizations must ultimately create a business case that will drive the software infrastructure decisions for individual programs. A business case should be based on cost, schedule, performance, and risk across the program lifecycle.

INTRODUCTION

As they migrate toward delivering Net-centric capabilities, government organizations are developing strategies for optimizing use of IT software infrastructure, maximizing ROI, and allocating realized capital to mission-critical capabilities.

The largest initial software cost in a project proposal is often the licensing cost for COTS software. It is easy to believe that open source software, with its lack of right-to-use license costs, would represent a simple shortcut to reducing the cost of system acquisition. In fact, the reality is that open source software, in many cases, can have a significantly higher total cost of ownership (TCO) over a project's lifecycle. Although open source software certainly has a role to play in modern information systems within government and defense organizations, decisions on when and where to use open source software should be based on a full understanding of the lifecycle costs and not simply on the cost of acquisition. Government and defense organizations must determine the TCO of COTS licensed software versus open source software at different architectural levels, including the operating environment, application, and middleware levels.

This white paper reviews evaluation criteria for determining the TCO and ROI of both the COTS and open source options. This information can be used to make investment decisions on a project-by-project basis. Because many government organizations are evaluating Oracle WebLogic Server and JBoss Application Server as the software infrastructure for government programs, this paper focuses on

these options. However, the majority of the evaluation criteria discussed here are useful for other COTS-versus-open-source decision-making processes. Organizations, program managers, and technical leaders can then create a business case to rationalize application server infrastructure decisions.

COMPONENTS OF A BUSINESS CASE FOR SOFTWARE PROJECTS

How do government organizations evaluate whether Oracle WebLogic Server or JBoss Application Server is the correct choice for a given program? These organizations must ultimately create a business case that will drive the software infrastructure decisions for individual programs. A business case should be based on cost, schedule, performance, and risk across the program lifecycle where the following conditions apply:

- *Cost* includes the total outlay required to operate the program from inception through deployment, routine maintenance, and upgrades, as well as the comparative cost of decommissioning and replacement.
- *Schedule* is the program timetable as measured against requirement milestones.
- *Performance* is the degree to which the program fulfills its stated requirements.
- *Risk* is the degree of probability that a program will fail to meet stated cost, schedule, and performance requirements.

Government organizations must build the business case to address these considerations across the entire program lifecycle, not just at inception. They will then be well equipped to develop a business case that properly assesses and measures the impact of software infrastructure choices on overall program success.

EVALUATE CONFORMANCE TO GOVERNMENT DIRECTIVES

In creating a business case for government programs, it is important to develop a risk mitigation strategy with respect to cost, schedule, and performance. As part of the strategy, organizations should assess how Oracle WebLogic Server and JBoss Application Server adhere to assurances of quality and security, including how they conform to government directives such as the National Information Assurance Partnership (NIAP) Common Criteria of the National Institute of Standards and Technology (NIST) and National Security Agency (NSA), Federal Information Processing Standards (FIPS), and the Homeland Security Presidential Directive (HSPD-12). In many cases, standards apply in multiple nations—for instance, certification standards under the Common Criteria Recognition Arrangement (CCRA) are recognized in countries including Australia, Canada, France, Germany, Japan, the Netherlands, New Zealand, Spain, South Korea, the U.K., and the United States. These countries authorize certificates globally; another 12 countries recognize and use products and deployments based on common criteria certifications.

Given DoD Directive 8500.2 E.3.2.5.1, evaluate Oracle WebLogic Server and JBoss Application Server as validated and recognized by the NIST and the National Security Agency (NSA) under the NIAP for conformance to international standards and national security accreditation.

For example, according to Department of Defense (DoD) directive 8500.2 E3.2.5.1, “If an approved U.S. Government protection profile exists for a particular technology area and there are validated products available for use that match the protection profile description, then acquisition is restricted to those products; or to products that vendors, prior to purchase, submit for evaluation and validation to a security target written against the approved protection profile.”¹

Oracle ensures that each new version of Oracle WebLogic Server conforms to the common criteria specifications for application servers and is responsible for all changes to the code and its maintenance thereafter. Oracle WebLogic Server 7.0 Service Pack 6 was granted NIST/NIAP Evaluation Assurance Level 2 (EAL2) Common Criteria certification in January 2006, and Oracle WebLogic Server 8.1 Service Pack 5 was granted NIST/NIAP EAL2 Common Criteria certification in April 2006. NIST/NIAP EAL2 certification is currently under way on Oracle WebLogic Server 9.1. As of December 2, 2008, no open source application servers are independently certified for conformance to the NIST/NIAP Common Criteria for IT Security Evaluation (ISO Standard 15408) at EAL2 or EAL4.²

Given DoD Directive 8500.2 E.3.2.5.1, government organizations should evaluate Oracle WebLogic Server and JBoss Application Server as validated and recognized by the NIST and the National Security Agency (NSA) under the NIAP for conformance to international standards and national security accreditation. They should also review each product’s road map for continuing to certify successive versions.

EVALUATE PUBLISHED PERFORMANCE BENCHMARKS

Evaluate published independent performance benchmarks; these provide a relative measure of performance that can be assessed against program performance requirements.

Another consideration in the development of a risk mitigation plan is published performance benchmarks. Government organizations should evaluate published independent performance benchmarks, which provide a relative measure of performance that can be assessed against program performance requirements. Nonprofit organizations such as the Standard Performance Evaluation Corporation (SPEC) exist to establish, maintain, and endorse a standardized set of relevant benchmarks.³

SPEC members include vendors such as Oracle and IBM as well as open source providers such as Red Hat (JBoss) whose application server software supports mission-critical applications within government organizations.

The SPEC maintains the SPECjAppServer2004 benchmark, “a multi-tier benchmark for measuring the performance of Java 2 Enterprise Edition (J2EE)

¹ Department of Defense Instruction 8500.2, February 6, 2003.

² The Common Criteria Evaluation and Validation Scheme (www.niap-ccavs.org/cc-scheme/vpl/).

³ Standard Performance Evaluation Corporation. Competitive claims reflect results published on www.spec.org (www.spec.org/osg/jAppServer2004) as of November 17, 2008.

technology-based application servers.”⁴ A performance benchmark does not guarantee equivalent performance within a program; application server performance is dependent on environmental factors such as hardware, tuning, application code quality, network bandwidth, and architecture. However, the existence of a peer comparison benchmark does provide government organizations with a relative measure of performance.

For testing completed through October 2008, SPECjAppServer2004 shows that Oracle WebLogic Server is the fastest application server in the industry—in both single-node and dual-node tests. JBoss Application Server has not publicly released performance benchmarks through SPECJAppServer2004 testing.

For testing completed through October 2008, SPECjAppServer2004 shows that Oracle WebLogic Server is the fastest application server in the industry—in both single-node and dual-node tests.⁵ JBoss Application Server has not publicly released performance benchmarks through SPECJAppServer2004 testing.

EVALUATE NATIVE OPERATIONS, ADMINISTRATION, AND MANAGEMENT CAPABILITIES

Government programs generally consist of several lifecycle phases: planning, implementation, and sustainment. Increasingly, government organizations are faced with limited visibility into program expenditures and ever-shrinking budgets. As a result, they often lack the necessary insight to effectively reduce program cost and maximize investment.

Evaluate the built-in operations, administration, and management (OA&M) capabilities for both Oracle WebLogic Server and JBoss Application Server that will drive down operations support and sustainment costs. If capabilities do not exist, programs should account for the cost, schedule, risk, and performance impact associated with building and sustaining, or acquiring, such capabilities.

On average, hard costs—such as up-front license fees, support contracts, and hardware—account for 10 percent to 20 percent of the cost of a software project. The remainder is soft costs that include facilities, power, cooling, and labor for the development, deployment, and maintenance of the system.

Labor is easily the largest cost factor of a software development project and is affected by factors such as usability, learning curve, quality of tooling, and the need to modify or build required infrastructure functionality. Developer productivity, technical support, quality of documentation, ease of software deployments, system administration, and system reliability all contribute to program costs. Even small improvements in these areas can have profound impacts on the total cost of the program—particularly if those improvements are focused on highly inefficient and pervasive phases of typical project lifecycles.

Software labor costs must be evaluated based on the lifecycle phase where they are expended. In general, the majority of software labor costs are for maintenance and minor feature enhancements.

At the October 2006 Gartner Symposium/ITxpo, Gartner Managing Vice President and Fellow Daryl Plummer said that many organizations spend as much

⁴ Standard Performance Evaluation Corporation. Competitive claims reflect results published on www.spec.org (www.spec.org/osg/jAppServer2004) as of November 17, 2008.

⁵ www.spec.org/jAppServer2004/results/res2008q4/jAppServer2004-20080924-00116.html and www.spec.org/jAppServer2004/results/res2008q3/jAppServer2004-20080826-00110.html.

as 90 percent of their IT budgets on “standing still.”⁶ This number has increased from March 2005, when Gartner analyst Ray Paquet estimated that 70 percent of IT budgets were spent on maintenance and operations.⁷ These numbers are in line with February 2006 Forrester estimates showing that in the U.S. and Europe, the Middle East, and Africa (EMEA) government organizations typically spend 80 percent of their budgets on maintenance operations⁸—the largest expense across a program lifecycle. Conversely, license costs associated with COTS products typically represent a small fraction of expenses.

Inflated operations support and sustainment costs can be attributed to several factors, including process, tooling, asset, and human considerations as well as a lack of long-term organization and governance planning. Government organizations continue to allocate ever-increasing resources to sustaining existing software infrastructure and find themselves limited in their ability to introduce new capabilities. Instead, analyzing historical spending and resource allocation across program lifecycles could help organizations make appropriate decisions—decisions that will serve as the foundation for a long-term strategy.

Although it is easy to focus on obvious and predictable hard costs such as software licensing and support, doing so can be detrimental to the lifecycle cost of a program. Government organizations should avoid making technology decisions based on short-term savings in hard costs that can dramatically increase the soft costs that represent the other 80 percent to 90 percent of total program expenditures. The best value to the government comes through optimizing developer productivity, providing reliable and scalable infrastructure, and reducing program soft costs.

Government organizations should evaluate the built-in operations, administration, and management (OA&M) capabilities for both Oracle WebLogic Server and JBoss Application Server that will drive down operations support and sustainment costs. If capabilities do not exist, programs should account for the cost, schedule, risk, and performance impact associated with building and sustaining, or acquiring, such capabilities.

Government organizations should avoid making technology decisions based on short-term savings in hard costs that can dramatically increase the soft costs that represent the other 80 percent to 90 percent of total program expenditures. The best value to the government comes through optimizing developer productivity, providing reliable and scalable infrastructure, and reducing program soft costs.

⁶ ZDNet, <http://blogs.zdnet.com/BTL/?p=3741>.

⁷ “Effortless Systems Management” white paper, BMC Software, 2005.

⁸ Forrester Research’s Gene Leganza, February 2006.

EVALUATE THE IMPACT OF THE ORGANIZATION AND GOVERNANCE STRUCTURE

Understand the impact of Oracle WebLogic Server and JBoss Application Server on organization and governance strategy and implementation. Specifically, these considerations must be framed within the context of organization and governance processes, tools, assets, and people.

A required resource investment is the establishment of an organization and governance strategy. Such a strategy may be developed bottom-up from within the program or inherited from within the enterprise. This strategy should span all program phases and encompass assets, people, processes, and tools. It should be dynamic, addressing current as well as future program requirements. The development of a robust, dynamic strategy will help minimize variability and maximize visibility across programs. Ultimately, the implementation of a dynamic organization and governance structure will allow predictability across program phases and drive reduced cost and risk and shorter schedules.

How is program complexity derived?

Before discussing the impact of Oracle WebLogic Server and JBoss Application Server on organization and governance strategy, it is important to define how program complexity is derived. Program complexity increases cost, risk, schedule, and performance impacts across all phases of a program.

For government IT programs, predicting program complexity requires understanding how operational baselines are defined—across multiple operational layers (network, hardware, software, application) within multiple security enclaves (unclassified, secret, SCI, coalition) and within multiple deployment environments (development, test, preproduction, production). An example, representing a single operational baseline, might be deployment of a command-and-control (C2) application across multiple operational layers within the secret security domain in the production environment and support processes, tools, assets, and personnel. The extent to which programs can create predictability with respect to assets, people, processes, and tools within these three layers is dependent on the maturity and effectiveness of the organization and governance structure. Operational predictability enables programs to reduce operations support and sustainment costs.

For example, how predictable is the process of acquiring, provisioning, managing, monitoring, and sustaining Oracle WebLogic Server hosting a command-and-control (C2) application on enterprise-class servers on the Global Information Grid (GIG) within the secret security domain in the production fielded environment? How predictable are the people and skills required to support this activity? What tools are needed to implement this process? What assets, including licenses and intellectual property, must be managed as part of the process? Finally, are the processes, assets, people, and tools the same when acquiring, provisioning, managing, monitoring, and sustaining a JBoss Application Server hosting a C2 application on low-end workstation-class servers on the GIG within the unclassified security domain in the preproduction environment?

What is the impact of Oracle WebLogic Server and JBoss Application Server on the organization and governance strategy?

Evaluate Oracle WebLogic Server and JBoss Application Server against defined design-time and runtime process-enablement capabilities natively provided by the products. If process-enablement capabilities are not available, organizations should account for the cost of building and sustaining or acquiring such capabilities as part of the business case.

As government organizations consider Oracle WebLogic Server and JBoss Application Server for use within programs, it is important to understand the impact of these products on organization and governance strategy and implementation. Specifically, these considerations must be framed within the context of organization and governance processes, tools, assets, and people.

Processes

Government organizations are building organization and governance models to support service-oriented architecture (SOA) implementations that include best practices for standards compliance, enterprise change management, functionality road maps, reuse enforcement, organizational structure, skills development, and funding models. With respect to application servers, it is important to understand the lower-level processes that must be defined to serve as foundational components for enterprise organization and governance best practices.

In regard to application servers, organization and governance processes are generally defined at two levels: design-time and runtime. Continual optimization of these processes will result in a concrete set of best practices and standards that can be leveraged by development, testing, and operations teams. Examples of design-time processes include multilevel provisioning, inventory management, versioning, configuration management, patch management, and security management. Design-time processes are intended to define the road map for application server infrastructure services and create repeatable processes for provisioning and managing applications and the application server infrastructure. Design-time processes enable government organizations to streamline lifecycle management through predictable processes while reducing operations support and sustainment costs.

Examples of runtime processes include performance monitoring, fault management, metrics collection, reporting, logging, and auditing. Runtime processes are intended to provide a real-time and historical view of application server and application runtime behavior. These views provide valuable visibility into the fielded system and an opportunity for process optimization, reducing long-term program lifecycle costs and risk while accelerating the program performance and schedule.

Government organizations should evaluate Oracle WebLogic Server and JBoss Application Server against defined design-time and runtime process-enablement capabilities natively provided by the products. If process-enablement capabilities are not available, organizations should account for the cost of building and sustaining or acquiring such capabilities as part of the business case.

In addition to the evaluation of public performance benchmarks and OA&M features, evaluate Oracle WebLogic Server and JBoss Application Server tooling in terms of features that provide visibility into operations cycles and optimize development cycles.

Tools

Oracle WebLogic Server or JBoss Application Server can exist as part of an overall tooling strategy to support program implementation. With respect to application servers, it is important for government organizations to recognize tooling evaluation strategies that can realize capital over the lifecycle of the program. As previously discussed, proper tooling implementation enables organizations to maximize visibility and minimize variability in operations support and sustainment, hence reducing the program maintenance funding requirements over time.

In addition to the evaluation of public performance benchmarks and OA&M features, government organizations should evaluate Oracle WebLogic Server and JBoss Application Server in terms of features that provide visibility into operations cycles and optimize development cycles. Features that provide visibility into operations cycles include intuitive administrator interfaces and configuration, autotuning, autorestart, real-time and historical management statistics, response automation, thread analysis, zero downtime, update management, and diagnostics.

Optimizing development cycles can reduce the time to deliver capabilities, reduce required development resources, increase code quality, and reduce development testing—driving cost savings, reducing risk, and improving the schedule and performance over the lifecycle of a program. Features that optimize development cycles include graphical development tools, popular development framework support, visual interface design, and code testing facilities.

Use the following questions to develop evaluation plans for the operations visibility features of Oracle WebLogic Server and JBoss Application Server.

What is the current resource investment in the following?

- Tracking user clicks
- Tracking access to the system (who, what, when)
- Dynamically modifying application server configurations
- Tuning the application server
- Troubleshooting application outages
- Understanding how applications are being used
- Identifying and troubleshooting performance
- Creating custom, location-based application views
- Diagnosing application issues without unified statistics
- Identifying, applying, and managing software patches
- Manually restarting application servers
- Troubleshooting application and server threads due to application or service downtime

For each item, Oracle WebLogic Server provides capabilities that range from Java Virtual Machine (JVM) monitoring and unified statistics through a Web-based management console to memory-leak and thread analysis tooling for performance troubleshooting, automatic restarts with the Oracle WebLogic Server Node Manager utility, and patch management through Oracle Customer Support.

PushToTest, a software publishing and global services company specializing in Web-enabled test automation technology, applied its collective set of tools to Oracle WebLogic Server and JBoss Application Server as part of an application server comparison study designed to measure developer productivity. PushToTest showed that Oracle WebLogic Server 8.1 and Oracle WebLogic Server 9.0 were the easiest, fastest, and most straightforward, requiring the least amount of time and expertise for software architects and developers to build services.⁹

In addition to evaluating public performance benchmarks and OA&M features, government organizations should evaluate Oracle WebLogic Server and JBoss Application Server tooling in terms of features that provide visibility into operations cycles and optimize development cycles.

Assets

A significant consideration in government organizations is the control and management of program and enterprise assets. Enterprise assets include physical artifacts or intellectual property that give governments a competitive, strategic differentiator over adversaries or enable improved collaboration, as well as pooling and reuse of existing resources, among cooperative agencies and departments. The selection of Oracle WebLogic Server or JBoss Application Server can potentially solidify or compromise such competitive differentiators.

Government organizations must evaluate critical asset considerations for Oracle WebLogic Server and JBoss Application Server.

Specifically, when leveraging Oracle WebLogic Server or JBoss Application Server, government organizations need to identify risk mitigation strategies for program assets, including licensing. They may also request product enhancements or fixes for Oracle WebLogic Server, which will then be developed, tested, delivered, and certified by Oracle; integrated into the product source code; and supported in future product versions. The delivery and management of Oracle WebLogic Server enhancements enables government organizations to operate with minimal risk.

With respect to JBoss Application Server, if a system integrator or a government organization modifies the underlying source code during implementation, what are the ramifications and potential risks? What would be the licensing impact if the packaged solution were delivered to another government organization? Does the modified code need to be contributed back to the open source community, consistent with the GNU Lesser General Public License (LGPL)? If so, what is the

⁹ www.pushtotest.com/Downloads/kits/soakit.html.

Evaluate and understand Oracle WebLogic Server and JBoss Application Server with respect to their impact on operational baseline variability, intellectual property control, licensing, and third-party asset management.

impact on the organization's or program's intellectual property? If the code is returned to the open source community, what assurances exist that the code will be included in future versions of the product? If the code is not included, what is the impact on the program that must manage the code, including integration testing and validation, against future versions of the product? Would such a code base still be considered an open platform if it were strictly controlled and used by the government—in short, is it still open source? With this approach, government organizations could assume a higher risk profile; such risk must be evaluated as part of the overall business case.

Another issue to consider is the impact of the underlying application source code's being modifiable. JBoss Application Server source code is modifiable; the government program is then responsible for managing any changes to JBoss source code. This presents additional complexity for supporting multiple development, test, and operational baselines. On the other hand, Oracle WebLogic Server source code is not modifiable. All changes to Oracle WebLogic Server source code are implemented, tested, and managed by Oracle for government programs' baseline continuity.

Another example explores the impact of making physical artifacts or intellectual property assets public. "The majority of JBoss software is licensed under the LGPL, or GNU Lesser General Public License. Users may distribute changed versions of JBoss products to others, but if you distribute such changed versions, you are required to share those changes with the rest of the community by publishing that changed source code under the LGPL."¹⁰ Oracle WebLogic Server licenses, on the other hand, are purchased and become the property of the purchasing organization.

Government organizations should evaluate and understand both Oracle WebLogic Server and JBoss Application Server with respect to their impacts on operational baseline variability, intellectual property control, licensing, and third-party asset management.

People

Another consideration in government organizations is the people responsible for executing a program. If an application server is implemented as a service enablement step within the rollout of SOA initiatives, the organization will have to be aligned to provide proper support for the implementation of application server tooling and the execution of organization and governance processes.

Specifically, when evaluating Oracle WebLogic Server and JBoss Application Server, government organizations should determine and factor into the business case the impact of required organizational restructuring, education, training, and staff augmentation. Organizations should also determine the impact of product

Determine and factor into the business case the impact of required organizational restructuring, education, training, support, and staff augmentation.

¹⁰ JBoss Professional Open Source Licensing, www.jboss.com/company/aboutopensource.

documentation quality, product communities, and product support on the overall cost, schedule, risk, and performance of the program.

Government organizations can develop a plan for evaluating the impact of personnel and productivity on a program by asking the following questions:

What is the program impact if...

- The organization has to restructure to provide additional personnel to support multiple baselines?
- Staff members have to be trained to use new assets?
- The organization has to augment existing staff?
- The documentation and product support community is insufficient?
- Having multiple support levels (developers, administrators, operators, testers, and the like) is not supported?
- The organization has been directed to collaborate per contractual obligations with an outside contractor?

Oracle WebLogic Server provides robust support for developers, administrators, operators, and testers. Through Oracle University, Oracle provides targeted training for these user groups, with courses on topics such as SOA, system administration, and application server development. Additionally, Oracle manages thousands of pages of public product documentation¹¹ targeting all phases and aspects of the application lifecycle.

Again, government organizations should determine and factor into the business case the impact of required organizational restructuring, education, training, support, and staff augmentation.

EVALUATE THE REQUIREMENTS: OPEN SYSTEMS, OPEN STANDARDS, OR OPEN SOURCE?

The terms *open systems*, *open standards*, and *open source* are often used interchangeably to describe the same set of principles. In reality, the three terms describe different concepts, and government organizations should be diligent in evaluating the requirements for individual programs, making architecture and product decisions based on the appropriate requirements.

Open systems are systems that contain runtime interfaces and application programming interfaces (APIs) that promote and allow interoperability. The runtime interfaces should be defined, conform to a public standard, and be accessible. The internal implementation of the interface should be abstracted; it can be proprietary or conform to an open standard. Open systems do not necessarily provide portability or compatibility—only interoperability. An open system can be implemented with either Oracle WebLogic Server or JBoss Application Server.

Qualify program requirements for open systems, open standards, and/or open source. As the requirement for open standards is validated, evaluate the maturity of the specific open standards required for the program. Also evaluate the benefits of the capability provided by the specification versus the impact of the proprietary implementation. To understand the role of Oracle and Red Hat in the introduction and development of the standard, do research on target open standards.

¹¹ Oracle's BEA product documentation, <http://edocs.bea.com/>.

Open standards refers to a set of publicly available standards that have reached a satisfactory state of maturity for implementation. The focus of open standards is to provide a measure of compatibility, generally facilitating portability, across products. This maturity is defined consistent with the transition of an open standard from specification to community-accepted standard. Open standards often contain both required and optional components; the degree to which an open standard is implemented within an application server determines the compatibility or portability of an application that leverages the standard across multiple containers. There are many standards organizations, often moderated by not-for-profit consortia or open industry organizations, including vendors such as Oracle and professional open source companies such as Red Hat (JBoss). Oracle actively participates in more than 100 standards organizations. In the context of SOA and Java standards, the most notable standards organizations include the Organization for the Advancement of Structured Information Standards (OASIS), Web Services Interoperability Organization (WS-I), World Wide Web Consortium (W3C), and the Java Community Process (JCP).

Although open standards are implemented in both Oracle WebLogic Server and JBoss Application Server, the implementation of open standards does not guarantee portability, for several reasons. Application servers might not implement the same open standard or the same version of a standard. Different vendors or professional open source companies might implement all or none of the optional features of an open standard. Finally, specifications might be represented as open standards that are, in fact, proprietary implementations.

Open source refers to a community-based approach to software development in which public access is provided to a product's source material, including code and documentation. Open source application servers, such as JBoss Application Server, are generally an integrated collection of multiple open source projects. In contrast, vendor products, such as Oracle WebLogic Server, do not provide access to the product source, instead providing only API-level access to published, public APIs that can be leveraged by application developers to interact with product internals. Fundamentally, with vendor products, the only entity that can physically change the core product is the vendor. With open source, any individual can change the core product within a specific implementation; individuals might or might not donate the changes back to the open source community, depending on modified source distribution in accordance with LGPL licensing.

Government organizations should consider whether or not the open source community as currently constituted is motivated to be responsive to the needs of the public sector on a large scale. Does the open source community have a dedicated engineering staff behind the software to make urgently required changes when new capabilities are required or flaws are detected?

Under the open source model, a government organization must either wait for someone in the open source community to develop new functionality (and trust that they implement the standard in a way that meets government needs) or must

pay an integrator to make the modifications for it. Once again, the costs traditionally absorbed by a commercial vendor would be transferred to the government. Commercial software, on the other hand, is driven by market needs. As a large customer, the government can influence technology road maps and the priority of enhancements and bug fixes without directly taking ownership of the software in perpetuity. Open source software, by definition, has no technology road map and forces government to take ownership of open source code. Over time, that code becomes closed and proprietary, because it is dedicated only to the projects owned by that particular government organization.

Thus, the selection of open source software does not guarantee support for current open standards, nor does it guarantee interoperability as an open system. A program can create a closed system, as well as a system that is not portable, with open source. The implementation of an open system is based on the degree to which open standards, as required by the program, are implemented as interfaces by a product and leveraged by a hosted application.

Government organizations should first qualify program requirements for open systems, open standards, and/or open source. As the requirement for open standards is validated, organizations should evaluate the maturity of the specific open standards required for the program. Standards in the draft or specification phase have not been widely accepted; implementation may vary widely. Implementations using these specifications can be proprietary. In such cases, organizations must evaluate the benefits of the capability provided by the specification versus the impact of the proprietary implementation. Governance models and enterprise repository requirements should also be considered, along with project portfolio management. Researching target open standards provides understanding of the role of Oracle and Red Hat in the introduction and development of the standard. Further, the open standards compliance of Oracle WebLogic Server and JBoss Application Server should be verified by the program. With this information, government organizations can make an educated decision with respect to the implementation of open systems, open standards, and open source.

EVALUATE THE REQUIREMENTS: OPEN SOURCE TOOLS OR OPEN SOURCE FUNCTIONALITY?

Increasingly, COTS vendors are adopting blended strategies for application development and deployment. This approach combines popular standards-based open source projects with commercial software. These open source capabilities are then maintained and enhanced as part of the vendor's technology investments, and the enhancements are rolled back into the open source community. A good example of this is the broad adoption of the Eclipse open source integrated development environment (IDE) as the tooling environment for commercial software products. The hard-won skills, organizational culture, and intensely competitive business models of major enterprise software companies continue to advance the art of blended-source software development at an unprecedented pace.

The selection of open source software does not guarantee support for current open standards, nor does it guarantee interoperability as an open system.

Identify open source projects to be implemented as part of the technical requirements, and determine the support for these projects available in Oracle WebLogic Server. This approach provides increased flexibility for developing a business case for Oracle WebLogic Server or JBoss Application Server.

As a result, government organizations now have access to popular open source technologies with the support of commercial vendors.

Oracle has led the movement toward a blended strategy for application development and deployment. Accordingly, Oracle has donated projects to the open source community, including Eclipse, Apache Beehive, and OpenJPA. Oracle has also included support for popular open source frameworks and projects within Oracle WebLogic Server, including Spring Framework, Kodo JPA/JDO, and Hibernate.

Government organizations should identify open source projects to be implemented as part of the technical requirements and then determine the support that Oracle WebLogic Server provides for these projects. This approach provides increased flexibility for developing a business case for Oracle WebLogic Server or JBoss Application Server.

EVALUATE THE ROLE OF THE APPLICATION SERVER WITHIN A SERVICE-ORIENTED ARCHITECTURE

Government organizations are rapidly adopting application servers as enablement tooling, making it possible to facilitate conformance to the Net-Centric Checklist (NCC).¹² Programs are leveraging NCC design tenets, particularly accessibility and interoperability, in combination with node design guidance from efforts such as Net-Centric Enterprise Solutions for Interoperability (NESI)¹³ to implement services on application server nodes. These nodes expose basic information and shared business services, either natively or through integration with existing legacy systems.

However, as government organizations look to fulfill the tenets of the NCC and the Office of the Secretary of Defense (OSD) NII¹⁴ metadata strategy, they should evaluate tooling beyond the scope of the application server. Regardless of COTS or open source, the application server provides an initial enablement step for designing, building, and exposing services. The ability to implement services beyond simple information and business services adds significant complexity.

As organizations mature beyond the implementation of simple information services, the SOA Layered Architectural Model can be used as a conceptual framework for driving tooling requirements. In addition to providing information services, this model consists of shared business services, presentation services, composite applications, and a service infrastructure. The application server hosts information and access services that typically represent the data and functionality of the existing enterprise or community of interest. Beyond such information and access services, organizations should look to leverage additional products to

¹² Office of the Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, Net-Centric Checklist, Version 2.1.3, May 12, 2004.

¹³ Net-Centric Enterprise Solutions for Interoperability documentation, v1.3.0, June 16, 2006.

¹⁴ Office of the Secretary of Defense Networks and Information Integration.

Do not implement Oracle WebLogic Server or JBoss Application Server without evaluating and understanding the role of the application server within a larger SOA and the program road map toward Net-centricity. Understand the impact of integrating both application server infrastructure and hosted services within a community of interest or an enterprise SOA.

simplify designing, building, securing, exposing, configuring, and managing shared business and presentation services as well as composite services and applications. To that end, they should evaluate workflow engines, portals, business process management tools, enterprise service buses, enterprise repositories, and registries.

Government organizations should not implement Oracle WebLogic Server or JBoss Application Server without evaluating and understanding the role of the application server within a larger SOA and the program road map toward Net-centricity. Organizations should understand the impact of integrating both application server infrastructure and hosted services within a community of interest or an enterprise SOA.

CONCLUSION

Government organizations must maximize the ROI and the impact of their software investments. As a result, there is a debate on whether to use COTS or open source software within IT programs. Specifically, organizations seek guidance when defining evaluation criteria for Oracle WebLogic Server and JBoss Application Server.

Government organizations should develop a comprehensive business case that includes the strategies presented here for evaluating Oracle WebLogic Server and JBoss Application Server. These strategies—including evaluation of government directive conformance, public performance benchmarks, OA&M native capabilities, the impact of application servers on organizational and governance structure, and the role of the application server in a SOA—can be combined with clarification and understanding of open systems, open standards, and open source tools and functionality requirements to help government organizations, individual program managers, and technical leadership create a business case that will rationalize the decision on the application server infrastructure.



Evaluating Oracle WebLogic Server and JBoss Application Server for Government Programs
Updated December 2008

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 © 2007, 2008, Oracle and/or its affiliates. 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 is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. 0408