GUIDE to USING OPEN-SOURCE SOFTWARE to DEVELOP WEB APPLICATIONS

Open Web Application Platform
White Paper
April 2009

Abstract

As enterprises struggle to develop and deliver new and more dynamic services to more people, they must do so with severe budget constraints. They need a Web infrastructure that can enable higher developer productivity at a lower cost. While many enterprises are looking to open-source software to meet these complex requirements, managing and supporting a range of disparate software solutions is a huge challenge for the IT staff. The Sun GlassFish Portfolio offers proven, integrated open-source software and enterprise-class support in the most complete, cost-efficient, open Web platform available.

Table of Contents

Executive Summary

Enterprises are under consistent pressure to deliver new functionality with fewer resources. They can no longer afford the luxury of having one infrastructure for Web applications and another for non-Web applications. Instead, enterprises require a robust infrastructure that can support all their needs at a lower cost while enabling higher developer productivity. More and more enterprises are moving to open-source software to meet these needs.

Enterprises that leverage open-source software typically pick the best product from the different categories to make up a complete Web application platform: a Web server (like Apache), an application server (like GlassFish or JBoss), scripting (like PHP and Ruby), an Enterprise Service Bus (like OpenESB or MuleSource) and a portal (like LifeRay). Adopting this approach has some significant benefits but also presents its own challenges:

- Cost/time to integrate the disparate projects together
- Ability to effectively patch and maintain the disparate projects
- Support of the product if/when problems arise and who can provide a fix for the product to address business critical issues

Sun announced the GlassFish Portfolio to enable enterprises to take advantage of open-source innovation in the Web application platform space while enjoying the assurance of enterprise-class support. The GlassFish Portfolio is the most complete, cost-efficient, open Web platform available. It includes software based on the work of leading open-source communities, including Apache, Tomcat, lighttpd, Perl, GlassFish, Squid, Liferay, and PHP—and Sun supports the entire Portfolio across Linux, OpenSolaris, and Windows operating systems.

Moving Beyond Static Web Applications

Because the innovation available through rich Internet applications has exploded in recent years, the Web experience in the consumer space has become far more dynamic and interactive. Enterprise employees and consumers alike have begun expecting more than static Web applications; they want rich Internet applications with functionality running in the browser or on mobile devices which directly leverage data in the back end. While such capabilities are becoming more common in the consumer space, enterprise applications often still have an older interface style and an infrastructure that makes it difficult to access information across data silos. Enterprises today want to bring this rich functionality inside their organizations.

At the same time, enterprises are struggling to deliver new services quickly and with less complexity, thereby increasing revenue and reducing costs. They must also decrease their risk by managing and protecting access to data and services and ensuring usage and access compliance. Finally, many businesses are looking for ways to reduce the acquisition costs of their software in order to increase their return on investment.

While there are a range of Web applications available that can increase business efficiency, capitalize on new revenue potential, and increase the capabilities of internal enterprise programs, they can also dramatically increase the complexity of rapidly growing corporate IT systems. And because traditional application servers lack the tools and functionality to develop rich Internet applications or to deliver existing applications over the Web, the need for a new tier—the Web platform—has emerged. This Web tier sits in front of the application tier and is responsible for taking output from the business logic and translating and formatting it into something that is consumable by someone through a browser.

As organizations continue to adopt and leverage the Web platform, the Web and application tiers will increasingly consolidate; business logic will get pushed up into the applications themselves, and much of the presentation layer will be run directly on the client device. In short, much of the consolidation of these tiers that is now taking place on the Internet will become more prevalent within the enterprise.

As enterprises adapt to more people accessing more services for more hours of each day in more ways, the responsibilities of enterprise IT staffs are becoming stretched well beyond installing and maintaining technologies. As Web application implementations grow, so does the risk. The labor-intensive management of Web applications may make projects too costly to be effective, or customer-facing applications may need continual adaptation to meet both corporate and customer demands. This is where open-source solutions can be advantageous for the enterprise.

Software Choices for the Web Platform

Over the last decade, companies have grown their IT infrastructures around two platform types:

- A formally supported proprietary middleware stack (BEA, IBM, Microsoft, Oracle, SAP, others)
- Free-access, open-source software (generally Linux, Apache, MySQL™, and PHP; referred to as the LAMP stack, although the components vary)

The former became the corporate standard and was used for most mission-critical applications, while open source largely became the choice for more immediate-need, less critical applications.

The Problem with Proprietary Middleware Stacks

A number of closed, proprietary Web and application software solutions are available in the Web platform space. While proprietary vendors focus on some compliance with industry standards, the platform that they provide locks customers into that solution. Additionally, proprietary enterprise software has often been prohibitively expensive, especially for departments, smaller projects, small/medium-sized businesses, and startups. The combination of high-cost, proprietary products and vendor lock-in frequently constrains businesses from embarking on new software initiatives—and the increasing consolidation of proprietary software vendors can easily result in higher prices.

Additionally, the capital expenditures and associated financial risk required to deploy proprietary products can either delay the profitability gains of new software initiatives or simply prevent enterprises from attempting innovative ideas to drive new revenue streams.

To ensure that problems can be addressed as quickly as possible and to reduce operational costs, enterprises can chose a comprehensive open-source platform backed by an established commercial entity that provides support and understands the interdependencies not only of that platform but of other third-party products that are already in the enterprise's IT infrastructure.

The Increasing Popularity of the Open-Source Web Platform

LAMP has popularized the open-source Web platform. Traditionally, LAMP solutions were used for small department-level applications; however, these applications have grown and the LAMP stack is frequently leveraged for even mission-critical applications in many enterprises.

While open source is becoming more popular as a way to lower the upfront cost of software and applications, enterprises should ensure that the open-source software they choose is backed by trusted commercial sponsors and/or a large, mature community of developers, and that the libraries they use are well-supported.

Leading open-source communities include a large number and diversity of contributors from a variety of vendors. Because some of those vendors provide commercial support and service for the software they are helping to develop, open-source software is typically less expensive to maintain because of price competition among those vendors. Additionally, enterprises that use open source are not locked into a single vendor or in danger of massive pricing changes due to a merger, so the threat of a severe price hike is non-existent.

However, enterprises should note that deploying a variety of open-source projects supported by a variety of communities can be a maintenance nightmare when, for example, a company's Web site crashes in the middle of the night. There are so many dependencies inherent in any Web application platform that to address the issue or even maintain the system can become a huge burden on an enterprise's IT staff.

Exploring Web Platform Options?

- **Think long-term.** Even when budgets are tight, think about the future when creating a Web application plan, and think in terms of how to best leverage existing applications and infrastructure.
- **Avoid vendor lock-in.** Consider a modular architectural strategy; if a component doesn't meet your needs or if its costs rise, developers can easily replace it.
- **Design your Web platform for worst-case success.** Ensure it can scale and that it can be modified and changed as changing needs and requirements dictate.
- **Consider the hidden costs.** Long-term licensing costs, maintenance, and poor performance can be a big hit to your bottom line.
- **Explore components that are not industry-specific.** By looking at applications that cut across industries, you'll have many more options to choose from.
- **Flexibility to leverage the newest capabilities.** It is to your competitive advantage to be able to introduce new services quickly. Make sure the software you choose doesn't inhibit your ability to integrate new features according to your business strategy.
- **Staff time.** The time your IT staff spends maintaining your enterprise infrastructure is time they can't spend developing new services.

Choosing the "Right" Open Source Web/Application Platform

When evaluating open-source software for an enterprise Web or application platform, the following are important considerations:

- **Size and activity level of the open-source community.** There is a lot happening in open source and the most popular projects (determined by downloads and community involvement numbers) are the ones that will likely survive and thrive, providing a steady stream of innovation
- **Open-source license (not just free).** Projects must have an open-source license (GPL, CDDL, Apache, etc.) that is endorsed by the Open Source Initiative (OSI—http://www.opensource.org/) This is the only way to determine if it is truly open source; otherwise, it is just another proprietary license
- **Active deployments.** Are there other customers using this in production? You probably don't want to be first unless there are compelling business reasons.

In the open-source Web application platform space, the projects that meet the criteria above are:

- **Web Server**—Apache, a Web server project launched in the early 1990s
- **Application Server**—GlassFish, an application server project based on the Java EE platform launched by Sun in 2005; JBoss, an application server from Red Hat; and Tomcat, an application server project that launched when Sun donated Java servlet reference implementation to the Apache Software Foundation in 1999
- **Enterprise Service Bus**—OpenESB, a Java technology-based enterprise service bus project founded by Sun in 2005; and Mule, a project founded in 2003
- **Portal**—Liferay Portal, the leading open-source portal on the market
- **Scripting languages**—PHP, Ruby, and Python
- **Database**—MySQL is the world's most popular open-source database

An Overview of Open-Source Projects and Our Recommendations

Project GlassFish™ (www.glassfish.org), launched when Sun open sourced its application server and the Java™ EE Reference Implementation, was Sun's first step toward open sourcing the entire Java platform. Less than a year after the initial launch, the GlassFish community delivered the first release of the GlassFish Application Server, a production-quality, Java EE-compliant application server, followed by a second release in 2007. Today, GlassFish is the leading open-source and open community platform for building and deploying next-generation applications and services. It includes more than 250,000 registered users, and the GlassFish application server has been downloaded more than 18 million times since 2006.

Because GlassFish is based on the Java EE platform, the industry standard for implementing next-generation Web applications, enterprises can easily transfer applications from more expensive, proprietary platforms like IBM Websphere and

Media Quote

"An integrated and supported open-source platform lowers the barriers and decreases the risk to large organizations who may very well be taking a conservative strategy to the economy. An easy-to-use cheaper platform should be well worth at least a serious look by organizations looking to lower infrastructure costs."

—Beth Gold-Bernstein
ebizq.net

Oracle Weblogic to GlassFish.

Like GlassFish, the current version of Red Hat's JBoss application server (www.jboss.org) is based on Java 2 Enterprise Edition (J2EE) and was released in 2006. Although JBoss has had some success in the past, we caution that there are issues around backward compatibility and features that are available in the free, open-source version but are not supported in the commercial release of the product.

In contrast, the GlassFish application server is backward-compatible; features released today will be supported in future freely available versions as well as future Sun-supported commercial versions of GlassFish Enterprise Server. Additionally, the freely available GlassFish application server is ready for production right out of the box. For these reasons, we recommend GlassFish application server over JBoss.

Tomcat application server (http://tomcat.apache.org/) is extremely popular with Java developers who only want to use servlets, but it doesn't support the full Java EE stack.

We believe that Tomcat and GlassFish provide the best choice for enterprise application servers and in most environments, both should be considered. In complex Web-tier architectures found in an increasing number of enterprises, a Java EE technology-based application server such as GlassFish is more appropriate than a Tomcat application server due to the complexity of the XML-based customizations that Tomcat requires.

Project OpenESB (http://www.open-esb.org), launched by Sun in 2005 as the home for the reference implementation of the Java Business Integration (JBI) specification, has become a leading open-source ESB. After delivering the reference implementation for the JBI specification (JSR-208), the community began building adapters and engines that plug into the standards-based platform as well as a rich set of tooling that makes developing applications easier. In 2008, Sun released GlassFish ESB, a combination of the GlassFish Enterprise Server, OpenESB components, and rich tooling with NetBeans™. GlassFish ESB provides Java EE developers a straightforward way to introduce SOA/ESB capabilities into their existing Java EE applications. The community is presently developing an even more modular platform, adding in OSGi support that will provide for a "sized-to-fit" deployment of just what is required. Since July of 2007, OpenESB has been downloaded more than 46,000 times.

Mule ESB (http://www.mulesource.org), another leading ESB, is a lightweight integration platform and service container that enables developers to connect applications. In contrast to OpenESB, Mule does not implement the JBI specification, instead relying on a proprietary Java technology-based interface for plugging components together. It also does not have rich tooling, but instead requires configuration through XML files. While Mule does not require an application server, which can lead to lower system requirements, that also means that the full facilities of Java EE are not seamlessly available.

We recommend OpenESB over Mule because OpenESB has a standards-based (OSGi) pluggable architecture that facilitates adding functionality from a variety of vendors. Without this pluggable architecture, the ESB can only be extended in a proprietary manner. In addition, OpenESB supports enhanced tooling that reduces complexity and increases developer productivity.

Sun's Open-Source Offerings for the Web Tier

Enterprises are looking for ways to make their existing IT environment more Web-centric. Most already use LAMP stacks, but need to integrate new capabilities into their existing system.

> For example, a large, U.S. financial institution that had multiple application platforms in its environment needed to get content out to customers and partners faster. The firm had developers on staff who were experienced with LAMP, and it was already running parts of LAMP in its environment. Recognizing that there are many versions of each LAMP component, the institution wanted to avoid the high costs of designing and maintaining multiple LAMP versions running across a variety of operating systems. The firm chose Sun GlassFish Web Stack software, a component of the Sun GlassFish Portfolio. GlassFish Web Stack enabled the company to reduce risk and simplify management by getting support from a single vendor.

Sun introduced the GlassFish Portfolio in response to enterprises' need to more fully leverage a complete Web platform. The most comprehensive, cost-efficient open Web application platform available, the GlassFish Portfolio includes all of the open-source projects recommended in the previous section of this white paper. It provides enterprise functionality and reliability with variable service level agreements (SLAs) to support smaller, low-cost deployments as well as extensive, mission-critical deployments—all at a simple, low, and predictable cost.

With the GlassFish Portfolio, enterprises developing Web applications can reduce total costs by more than 90% and improve application price/performance 7 times over alternative offerings. The Portfolio includes the following key products and features.

Sun GlassFish Enterprise Server

The performance record-setting open-source and open-community platform for building and deploying next-generation applications and services, GlassFish Enterprise Server is the Java EE reference implementation. Based on the work of the GlassFish community, it provides a focus on simple installation, development, and management of production environments with extreme scalability and reliability.

Analyst Quote:
According to Gartner, "An increasing number of organizations are considering supported open-source application infrastructure technologies because they don't require capital expenditure."

—Source: Gartner, *GlassFish Suite of Products Continues Sun Microsystems' Quest to Leverage Open-Source Technologies,* Jess Thompson, March 17, 2009.

Read the full report at:
http://mediaproducts.gartner.com/reprints/sunmicrosystems/volume1/article4/article4.html

Sun GlassFish Web Stack

A complete LAMP cross-platform portfolio of Web-tier technologies developed by several open-source communities (Apache, PHP, Ruby, MySQL, GlassFish, Tomcat, Lighttpd, Squid, and more), GlassFish Web Stack includes components compiled, pre-configured, and tested by Sun across multiple operating systems for optimal performance and compatibility.

Sun GlassFish Web Space Server

Based on the Liferay open-source portal project, GlassFish Web Space Server enables users to create their own Web spaces and define the access and functionality within their enterprise and social networks.

Sun GlassFish ESB

A Java technology-compliant, Web services-based, pluggable integration platform, Sun GlassFish ESB incorporates the Java Business Integration (JBI) standard to allow loosely coupled components to communicate with each other through standards-based messaging.

Message Queue

Enabling loosely-coupled applications to reliably exchange messages and cost-effectively scale, Message Queue allows applications to produce and consume messages at different rates without causing data traffic jams or system grid lock.

Update Center

Providing enterprise developers the ability to easily download, install, patch, and manage multiple implementations of GlassFish Portfolio components, Update Center also offers notification, access, and installation of available patch updates.

Enterprise Manager

Delivering an additional value-add to the application monitoring and management available within the administration console, Enterprise Manager reduces the risk of production problems by monitoring common causes of downtime and performance degradation.

The GlassFish Portfolio addresses the need for enterprises to build more with less and grow revenue and profits at a lower total cost of ownership and lower initial cost of development.

For example, a business with fewer than 1000 employees that runs 20 dual-CPU, dual-core x86 servers for the application server and 10 dual-CPU, dual-core x86 servers for the database server could expect to pay more than $3 million for a three-year contract using a proprietary application server and database. Sun offers a GlassFish & MySQL Enterprise Limited offering that costs just $240,000—a savings of more than $2.8 million.

The GlassFish Portfolio is available in both per-server pricing as well as Unlimited licensing.

- **Per-server Subscription**—an annual subscription based on the number of physical or virtual servers on which the software components of the GlassFish Portfolio are deployed. Smaller or initial business projects can benefit from the per-server pricing options, allowing installation and deployment flexibility with great cost savings.
- **Unlimited Subscription**—an annual subscription based on the number of company employees; unlimited pricing provides lower initial costs, more predictable future costs with no requirements for accounting, and no restrictions for architectural layouts or scale.

The Portfolio's flexible pricing ensures all levels of enterprises, from large Fortune 500 enterprises to smaller or medium-size businesses and start-ups, can leverage the platform to meet their specific needs.

How to Get Started with Sun's Open-Source Web Application Platform

1. If they haven't already, ask your developers to download and experiment with open-source Web platform components, including:
   a. Application server from www.glassfish.org
   b. HTTP server from www.apache.org
   c. Portal server from www.liferay.com
   d. Database from www.mysql.org
   e. ESB from www.open-esb.org
   f. Operating system from www.opensolaris.org (which includes industry-leading data management and virtualization capabilities)
2. If these products prove to be a good fit, or you need these products pre-integrated together, consider GlassFish Portfolio (www.sun.com/glassfish). If you start with any individual component, you can easily migrate to Sun's fully supported, commercial versions to simplify management, gain enterprise-level support, and ensure timely updates—all with flexible, subscription-based pricing.

Learn More

For more information about the Sun GlassFish Portfolio, visit sun.com/glassfish

Customer Case Studies

Pretium Telecom Builds Flexible, Cost-Effective SOA with GlassFish ESB

Pretium Telecom, a large provider of fixed telephony in the Netherlands, wanted to simplify its service-oriented architecture (SOA) to speed up the development of a new VoIP offering and to minimize costs. With the solution Pretium Telecom used at that moment, only 20%–30% of that solution's capabilities was used—and the unused components slowed performance, inflated hardware requirements, and added complexity to the user experience. Not only did this drive up the total cost of ownership, but the existing SOA's proprietary business-process-execution-language (BPEL) engine also limited interoperability, restricted choice, and impeded change. In addition, the new VoIP and broadband offering required additional hardware, which would significantly increase licensing costs. Another challenge was that all new business partners participating in the service chain—providing services such as modem delivery or home support—needed to run the proprietary BPEL Process Manager to interact with Pretium Telecom's SOA, adding another requirement and expense.

After evaluating possible solutions from Sun, Apache, and JBoss, Pretium Telecom and Yenlo concluded that replacing the current SOA Suite with Sun GlassFish ESB and Sun GlassFish Enterprise Server could provide the required functionality in a smaller, more agile, open-standards–based framework. In addition, the GlassFish ESB and Enterprise Server products offer a variety of support options from Sun, and having support services was crucial to Pretium Telecom. What finalized the company's decision to choose the Sun solution was the proof of concept. In less than two weeks, IT vendor Yenlo was able to rebuild the company's entire stack.

One month later, 8 - 10 employees from Pretium Telecom and Yenlo began to work on the SOA, which would initially support only the new VoIP and broadband services and later would replace the complete Oracle SOA offering. The team used the built-in BPEL engine in the GlassFish ESB to design an ordering system that includes 10 fully automated business processes to support tasks such as account setup, customer relationship management, invoicing, direct withdrawal from bank accounts, and marketing campaigns. In addition, the team took only one day to set up the new highly available production environment with four clusters that use in-memory replication and the SUSE 10.x operating system. To develop new applications, including a call-center program that runs on the GlassFish Enterprise Server, the team used the NetBeans IDE and Spring Framework. Developers also used built-in adapters in the GlassFish ESB and components of Metro Web Services—including JAX-WS, JAXB, WSIT, and XWS-Security—to create secure, XML-based Web services that can exchange information between Pretium Telecom’s systems, including an Oracle database, and six partner systems, some of which use Microsoft .NET.

“We are very excited about our new GlassFish ESB solution. It provides a combination of open-source development tools, which makes it possible for us to quickly adapt to changes and increase profits. And the development environment’s ease of use is remarkable. Our developers’ productivity has increased about 40%–50%,” says Ruud de Greef, chief information officer at Pretium Telecom.

Two notable differences between the Oracle and Sun solutions are the ease of clustering servers and establishing security. “Previously, it costed a lot of money to license clustered enterprise services, and setting up clusters was very difficult,” explains Joost Hofman, senior system engineer at Yenlo. “Today, we can cluster services in less than ten minutes.”

Adding security to applications and services also takes seconds. “This is a huge leap from the previous architecture,” adds Hofman.

Rapid resolution through the open-source community and Sun support services also boosts efficiency. “Sun provides very professional support services, and we have easy access to the engineers who can quickly answer our questions,” says de Greef.

Hardware costs are much lower than predicted, and throughput has increased. “The configuration of the GlassFish ESB is more efficient because we don’t have components that we don’t use,” de Greef notes. “This means we need less memory and less storage.” As a result, the new SOA costs approximately 50% less than the previous one.

In March 2009, Pretium Telecom launched its new SOA—which is protected by Sun Silver Support—three months earlier than projected. End users can access the SOA through a single Web-based application. Although the migration from the Oracle SOA Suite to the GlassFish technologies is still in progress, Pretium Telecom will completely replace the Oracle SOA Suite by October, 2009.

Initially, the new SOA is expected to support approximately 5,000 new customers each month. In general a relationship with a customer will last for many years. Pretium Telecom expects the new architecture to support at least 175,000 transactions a day in October 2009 once the replacement of the Oracle SOA stack is complete.

Telecom Company Reduces Operating Costs,

Increases Efficiency with GlassFish

After expanding rapidly over the past several years, a large North American telecom company found that its IT environment had become a mix of non-standard technologies and custom-developed applications. Integrating and maintaining this wide variety of systems resulted in higher operating costs, unreliable service, and longer application-development cycles.

To address this challenge, the firm wanted to move to a model that leverages industry standards and open-source solutions and decided to evaluate Sun open-source products. The benefits of GlassFish Enterprise Server to the company's retail operations soon became apparent, as GlassFish costs significantly less than IBM WebLogic server and, unlike JBoss, Sun offers a comprehensive, long-term open-source strategy that includes a complete software stack, middleware, and support services.

Because the GlassFish application server was very easy for the telco to download and test, the company felt comfortable with the solution within a few weeks and deployed it in its retail environment. Key elements in the company's decision were product performance, ease of getting the product to evaluate, and paying for the license and support at the time of deployment in production. The new solution is expected to reduce the telco's operational costs and development cycles, and to increase system availability and performance.

Learn More

For more information about the Sun GlassFish Portfolio, visit sun.com/glassfish.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN (9786) Web sun.com

Printed in USA 04/09 561266