



RED HAT CONSULTING

STRATEGIC MIGRATION PLANNING GUIDE

WEBLOGIC TO JBOSS ENTERPRISE APPLICATION PLATFORM

CONTENTS

1. EXECUTIVE SUMMARY	3
Pre-planning	3
The migration planning process	4
2. MIGRATION CONSIDERATIONS	4
Migration drivers	4
Potential migration scenarios	5
Configured services	7
Migration deployment scenarios	9
Hardware migration	14
Other considerations	19
3. THE STRATEGIC MIGRATION PROCESS	20
Migration process overview	20
Phase I: Server architecture and deployment model	21
Phase II: Application migration assessment	21
Phase III: Effort and risk assessment	23
Phase IV: Server and application migration plan	25
Phase V: Migration implementation	26
4. ENTERPRISE SERVICES	26
Middleware consulting services	26
Platform consulting services	27
Training	30
5. SUCCESSFULLY MIGRATED CUSTOMERS	32
AVIS Europe	32
GEICO (Government Employees Insurance Company)	34
Heidelberger Druckmaschinen A	36
Daiwa Securities America Inc.	39
Swedish National Police Board	42
Autotrader.co.uk	44
6. SUMMARY	47

1. EXECUTIVE SUMMARY

IT organizations are constantly faced with the challenge to produce high-quality solutions with a lower total cost of ownership (TCO). With the growing recognition that open source software provides quality, stable solutions, migrations of existing enterprise applications to products such as JBoss[®] Enterprise Application Platform have become increasingly popular.

JBoss Enterprise Application Platform provides a holistic solution that includes tools for such business challenges as business process management, enterprise application integration, and service-oriented architectures (SOA), enterprise portals, and data services solutions.

Migrating from proprietary technologies to JBoss Enterprise Application Platform offers many other advantages to IT organizations, including cost reduction versus proprietary JEE implementations, scalability, and the ability to extensively customize the server. Other advantages include:

- **Standards compliance:** Not only has JBoss been a strong supporter of open standards, it has been instrumental in driving a lot of the innovation around open standard enterprise Java, including JPA, JSF, EJB, and CDI. Basing your infrastructure on an open implementation of open standards removes a great deal of risk when choosing a platform for the future.
- **Cost:** Given that there are no up-front licensing costs, JBoss delivers lower overall TCO than Oracle WebLogic while providing mission-critical support via a subscription support model. A lack of up-front licensing costs combined with open standards compliance ensures that you are not locked into a single vendor with JBoss and gives you the flexibility to take back ownership of your IT environment.
- **Flexibility:** JBoss pioneered modular Enterprise Java and its second-generation Microcontainer model allows you to deploy and run only what you need, reducing operational overhead and increasing deployment flexibility.

JBoss uses the community version of its projects, following a “release early, release often” model that promotes innovation and is most responsive to user requirements. Appropriate versions of open source projects are subsequently integrated and put through a rigorous certification process to produce enterprise-class solutions, including JBoss Enterprise Application Platform.

A successful and smooth migration to JBoss requires a strategic level of up-front planning and a proper understanding of the issues and challenges involved in order to minimize migration risks and costs. In this document, we attempt to cover the most important items and provide readers with some of the resources and references necessary to be successful.

PRE-PLANNING

A thorough understanding of your migration environment is the critical first step to start realizing the benefits of moving to JBoss. Your organization’s motivations for undertaking a software platform migration should be carefully considered, as these may influence choices, opportunities, and trade-offs. Likewise, understanding your potential deployment scenarios will help you proactively identify any roadblocks and anticipate future needs.

THE MIGRATION PLANNING PROCESS

Red Hat has established a proven five-step process designed to identify migration opportunities, examine the risks associated with various migration scenarios, create a standard enterprise environment, and develop a comprehensive and strategic migration plan for the enterprise.

Through this process, your organization will:

1. Examine the existing application server and platform and determine the equivalent capabilities in a JBoss EAP ecosystem.
2. Examine proprietary features and native capabilities and determine the equivalent capabilities in JBoss EAP.
3. Measure organizational readiness and overall migration risk.
4. Develop a strategic migration plan, including a detailed roadmap and cost estimate.
5. Implement the strategic migration plan and employ implementation support strategies.

The details that follow are intended to provide insight into the considerations and processes required to move to the JBoss Enterprise Application Platform. We encourage you to share this with your team as you embark on your migration planning. Through these insights, we hope to arm you with the knowledge to successfully plan and execute your migration.

2. MIGRATION CONSIDERATIONS

When a company decides to migrate from one application server to another, there may be several possible drivers involved. Common drivers deal with cost, increasing development innovation, or combining datacenters with different technologies.

MIGRATION DRIVERS

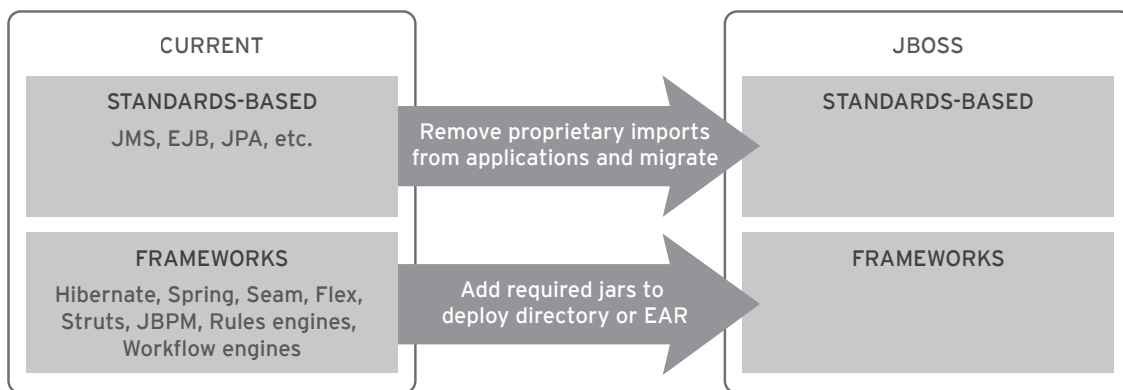
- Cost
 - Gain the ability to distribute server with application to customers
 - Eliminates per-CPU licensing
 - Allows the use of cheaper hardware and more robust clusters
 - Lowers support costs
- Expansion of server farms
- Conformance to standards
- Integration and customization
- Flexibility
- Failure of outsourcing
- Mergers and acquisitions
- Datacenter consolidation
- Adoption of rapid development techniques
- Desire for latest features and capabilities
- Performance
- Scalability

Whatever the particular drivers are, it is important to bear them in mind when planning migrations. Drivers help dictate priorities and methods when migrating. They are key to determining which migration scenario applies and how it should be deployed. For example, if a major driver is the desire to benefit from the latest features and capabilities, the migration project will need to delineate the features desired and determine which can be achieved by changing configurations and which will require updating source code.

POTENTIAL MIGRATION SCENARIOS

BASIC WEB APPLICATION MIGRATION

Migrating basic web applications that were developed according to the servlet specification and Java EE specifications is the easiest and most cost-effective way to start migrations. It can be an excellent measure of the effort involved. If the applications were developed with a standards-based Integrated Development Environment (IDE) such as Eclipse, this will go quite smoothly. Many proprietary IDEs embed links to proprietary libraries and thus can add additional challenges to migration.



Standard web applications include a deployment descriptor called “web.xml.” Forgoing behavior by application servers often leads to the use of syntax and notations that are not portable. Red Hat tries to document such examples and has a ready list of changes that are often needed when migrating from specific application servers to JBoss solutions.

Most application servers also include a second proprietary deployment descriptor for web applications to support value-added features that are not included in the standard specifications. This second descriptor is called weblogic.xml for WebLogic Server (WLS) and jboss-web.xml for JBoss. Red Hat often maintains XML style sheets that can be used to automatically transform other vendor descriptors to the JBoss-supported “jboss-web.xml,” where features and capabilities overlap.

Migrating a web application that is developed and deployed to WebLogic can be very simple or somewhat complicated depending on how proprietary the application architecture and dependencies are. Many web applications can simply be copied over and directly deployed to JBoss with no changes at all. Some of the common barriers to migrating web applications can include:

1. When deployed as an exploded directory, a .war extension for the directory may be required when deployed to JBoss. A similar naming convention in WebLogic is uncommon.
2. WebLogic web applications almost always bundle a weblogic.xml deployment descriptor. This descriptor can often include no relevant configuration information whatsoever and would therefore be ignored and not cause any harm under JBoss. When it provides a relevant configuration, it is very often comprised of simple items such as the context root of the web application or security settings for it, which are similarly provided in a jboss-web.xml file under JBoss.
3. Web applications may use libraries provided by WebLogic, including but not limited to JSP tag libraries and WebLogic helper classes. This is where the migration effort can potentially become substantial, depending on the exact nature of the library being used and what it takes to replace it with a standard open source alternative.

JAVA EE MIGRATION

Migrating Java EE applications is in theory no more difficult than migrating pure web applications. However, the purpose of Java EE specifications is to facilitate the integration of multiple frameworks with diverse functionalities. This means that there will in practice be many mini-migrations involved. We will discuss these more in the JSR migration section.

		APPLICATIONS TO MIGRATE				
		App 1	App 2	App 3	App 4	App 5
Applications	App 1		✓	✓		
	App 2				✓	
	App 3		✓		✓	
	App 4			✓		✓
	App 5					
Libraries	Jakarta Commons 3.2	✓	✓	✓	✓	
	Apache etc.	✓		✓		
	Hibernate 3.1				✓	
Framework	Hibernate 3.4	✓	✓	✓	✓	
	Spring 2.1		✓			
	Oracle DB 10g	✓	✓	✓	✓	
	etc.					

If proprietary IDEs are used in the development of Java EE applications, these will often have the generated classes extend and implement proprietary classes and interfaces. The JBoss MASS project provides a migration analysis tool called MAT, which scans the application's source code for proprietary libraries and imports. This provides invaluable information for estimating the amount of work that needs to be done for a Java EE migration.

The tool is hosted at <http://www.jboss.org/mass/MAT.html>.

JBoss MAT also produces detailed HTML reports covering server configuration, deployed applications, and class dependencies.

ADVANCED JSR MIGRATIONS

Java Specification Requests (JSRs) represent the Java community's effort to provide standards in additional areas such as:

- Portal
- ESB
- Rules engine
- Workflow engine
- Injection

There are different migration paths for each separate JSR. Depending on how complete the specification is, migration can be fairly simple or very involved. Generally, annotations are more standard than XML implementations.

CONFIGURATION MIGRATION

A WebLogic domain consists of an administration server and one or more clusters. To obtain information about a WebLogic domain configuration, only the domain directory on the administration server should be examined. Each cluster in the domain will then map to a JBoss cluster and each server used outside of a cluster would map to a JBoss server instance. The domain configuration is found in the config directory (under the domain directory) and the high level configuration is all within the config.xml file (in version 10.x of WLS).

CONFIGURED SERVICES

WebLogic Server has the concept of deployed applications and configured services. Services—in WLS terminology—can include JDBC services and JMS destinations, which are considered deployable artifacts in JBoss. This leads to two very distinct models:

With **WLS**, services such as JDBC and JMS are configured through the administration console by the WebLogic administrator. This is true in both production and development, though the developer may also be acting as the WLS administrator in a developer environment. While such service configuration ultimately depends on application needs, it is both physically and practically separate from the application itself. Application developers often document their service dependencies and administrators follow the document to set up the required services for the application to function properly.

With **JBoss**, each service (for example, a JDBC connection pool or a JMS queue) is configured in its own XML file. Two or more services of the same type can often be configured in the same XML file, if that is preferable. This gives users flexibility in their configuration choices. The same configuration practices administrators would use with WebLogic can be followed—configuring services based on provided documentation. But it is also possible for the development team to include these deployable XML artifacts with their application or even bundle them in the same folder or archive. This second model leads to a self-contained application that includes its own configuration dependencies.

The Migration Analysis Tool (MAT) provided by JBoss can automatically inspect your WebLogic environment and provide an inventory report of the configured services. Generated reports in HTML format help the architecture team map out service requirements and dependencies.

JDBC

In WebLogic version 10.x, each JDBC resource is configured by the insertion of a `<jdbc-system-resource>` XML snippet in the `config.xml` of the domain. An XML element called `<target>` will specify which server or cluster the JDBC connection pool is to be deployed to and the `<descriptor-file-name>` will point to the connection pool's configuration details. Configuration details for JDBC connection pools are typically found under the `jdbc` directory of the domain's config directory in WebLogic 10.x.

In JBoss, it is enough to create a datasource XML snippet for each connection pool which may be provided in its own `poolName-ds.xml` file or combined collectively in the same `*-ds.xml` file. The content of such a datasource XML file in JBoss will depend on the transactional behavior of the datasource as well as the connection details. Most—if not all—of these details can be found in the matching `*-jdbc.xml` file under the `config/jdbc` directory of the WebLogic domain. The database password will be encrypted and therefore such a migration cannot be fully automated. When the proper `*-ds.xml` files are prepared, they can simply be deployed to the matching JBoss server or cluster to provide an equivalent JBoss environment.

JMS

In version 10.x, high-level JMS configuration is provided and grouped as distinct JMS Servers. This includes configuration like the persistence store for destinations—which are set individually for each destination in JBoss. JMS resources in WebLogic are configured by the insertion of `<jms-system-resource>` XML snippets in the `config.xml` of the domain. An XML element called `<target>` will specify which server or cluster the JMS resource is to be deployed to and the `<descriptor-file-name>` will point to the JMS configuration details, including connection factories and actual destinations.

Configuration details for JMS resources are typically found under the `jms` directory of the domain's config directory in WebLogic 10.x. In JBoss, JMS connection factories are typically configured in the `deploy/jboss-messaging.sar/connection-factories-service.xml` file of a JBoss server, but can generally be configured as part of any service archive deployment. It is more typically the destinations themselves—the queues and topics—which need to be configured and deployed to support each application.

In JBoss, it is enough to create a JMS destination XML snippet for each queue or topic, which may be provided in its own `destination-service.xml` file or combined collectively in the same `*-service.xml` file. The content of such a destination XML file in JBoss will resemble that of any MBean deployment file, and the MBean code will be `QueueService` or `TopicService` as the case may require.

The destination configuration details can be found in the matching *-jms.xml file under the config/jms directory of the WebLogic domain. In WebLogic, each individual queue or topic is deployed to a server or cluster by specifying a <sub-deployment> XML element in the main config XML file. The corresponding destination XML file would be deployed to the JBoss server or cluster, possibly by dropping it in the server's deploy directory or one of the cluster members' farm directory.

CLUSTERING

While the concepts and the goal remain very similar, there can be significant differences in the way clustering is configured in WebLogic and JBoss.

WebLogic provides a very static view of clustering, both in terms of cluster membership and configuration details. There are limited configuration options and clustering largely remains a black box. This leads to a level of simplicity that works well for some clients but can cost others with more exceptional requirements. The cluster itself is statically defined on the administration server with provided addresses of each member.

In JBoss, a cluster is a very dynamic concept. It is defined by a cluster name and an address, which may be a multicast or unicast address. Any JBoss instance declaring an intention to join the cluster can do so by communication with that address and by providing the cluster name. This results in a very fluid cluster definition that works very well with dynamic provisioning scenarios. Furthermore, clustering in JBoss is built on top of JGroups, which itself is highly configurable and has configuration parameters that are exposed to cluster users. A wide variety of network topologies can be supported and the ability to fine-tune the cluster performance is virtually endless.

CACHE REPLICATION

In a cluster, a number of caches are provided and must be kept in sync to avoid data integrity issues. The most prominent ones are the HTTP session and entity bean caches. Once again, WebLogic takes a black-box approach to cache configuration while JBoss builds them on top of open stacks that are exposed and highly configurable.

JBoss Cache is used to provide a distributed transactional cache for entity beans and the HTTP sessions. A wide array of configuration options can help tailor the cache behavior to user requirements. Cutting-edge technology such as non-serialization-based replication through AOP is available through the POJO cache features. The communication protocol is also configurable, and synchronous replication, asynchronous replication, and invalidation are all supported.

MIGRATION DEPLOYMENT SCENARIOS

HIGH THROUGHPUT OR LOW LATENCY

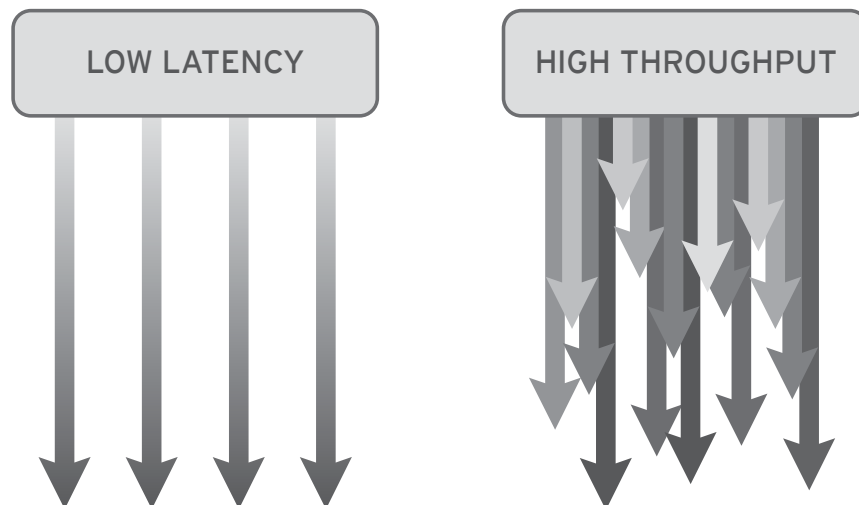
Servers deployed for high throughput are configured very differently than low-latency application servers. It is important to separate applications that require low latency from those that require high throughput. The configuration of timeouts, max threads, and other basic options are at the opposite end of the spectrum from each other and will not mix well.

When targeting high throughput, it is the overall performance of the system that is prioritized – client requests are not given individual attention. As an example, for a system with only four processor cores, one might determine that 200 concurrent requests mark the optimal point of parallel processing with an average processing time of one second. Anything below this number would be under-utilizing the system, while a higher number of concurrent requests results in wasteful context switching and hurts the overall throughput.

Configuration based on throughput would therefore limit concurrent requests to 200. In the event of 300 concurrent requests, we would queue up the remaining concurrent requests and process them as a batch consecutive to the first group. In this oversimplified analysis, requests that are part of the second batch will take almost twice as long to process as the first batch.

This runs counter to prioritizing low latency and the requests from the second batch may well be violating our service-level agreement (SLA). Should low latency be our priority, we could configure the system to accept 300 concurrent requests.

In our hypothetical example, it is conceivable that average processing time would then grow to 1.7 seconds. Requests from the second batch would complete in 1.7 seconds instead of 2 seconds and would therefore meet the SLA. The throughput, however, has gone down from 200 tps to 176.5 tps and even with a constant flow of 300 concurrent requests, the system will process fewer requests in the long-term.



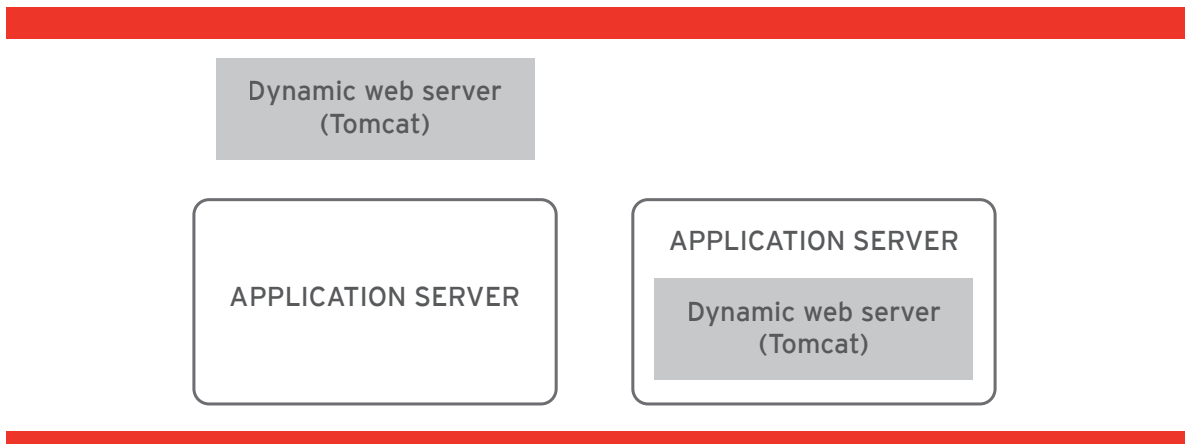
JBoss provides a robust platform for building out large-scale server farms that can support tens of thousands of concurrent requests.

PHYSICAL DEPLOYMENT MODELS

- **Physically separate machines for web and applications servers**

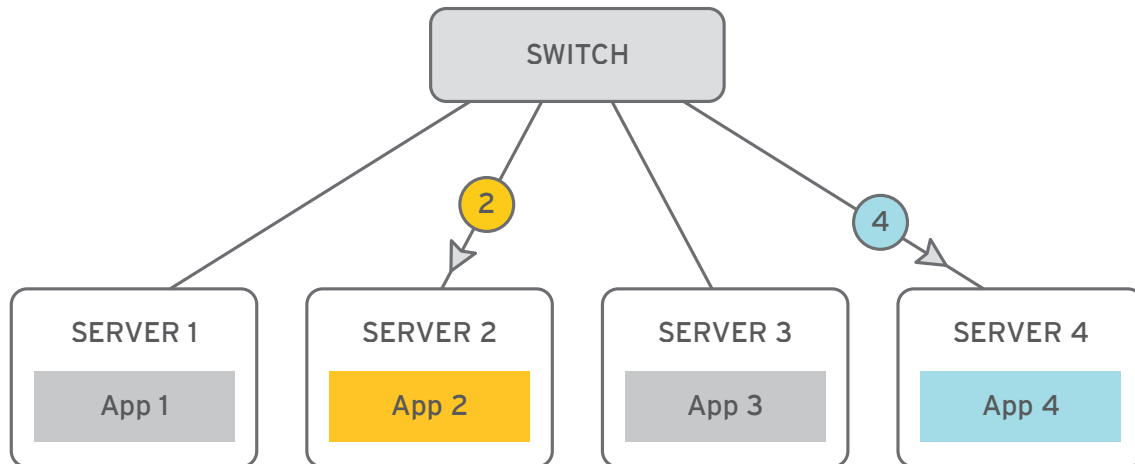
Some middleware server companies advocate using separate servers for web servers (servlet containers) and application servers. High license costs and a desire to avoid them in favor of simpler and cheaper servlet containers have often been at the root of such concerns.

JBoss embeds the web server in the application server, making for a much more efficient use of hardware while benefiting from an open source model that avoids many licensing concerns. As such, decisions to implement two physically distinct layers for presentation and business tiers should be reviewed in light of the lack of such licensing restrictions.



This enables JBoss to make more efficient use of hardware resources. Users have found that they can sometimes reduce their hardware requirements by up to 50 percent for the same load. This offers greater flexibility in selecting hardware and matching load. Of course there will still be occasions where separate physical tiers are recommended for various reasons, including security, isolated configuration, and better reliability.

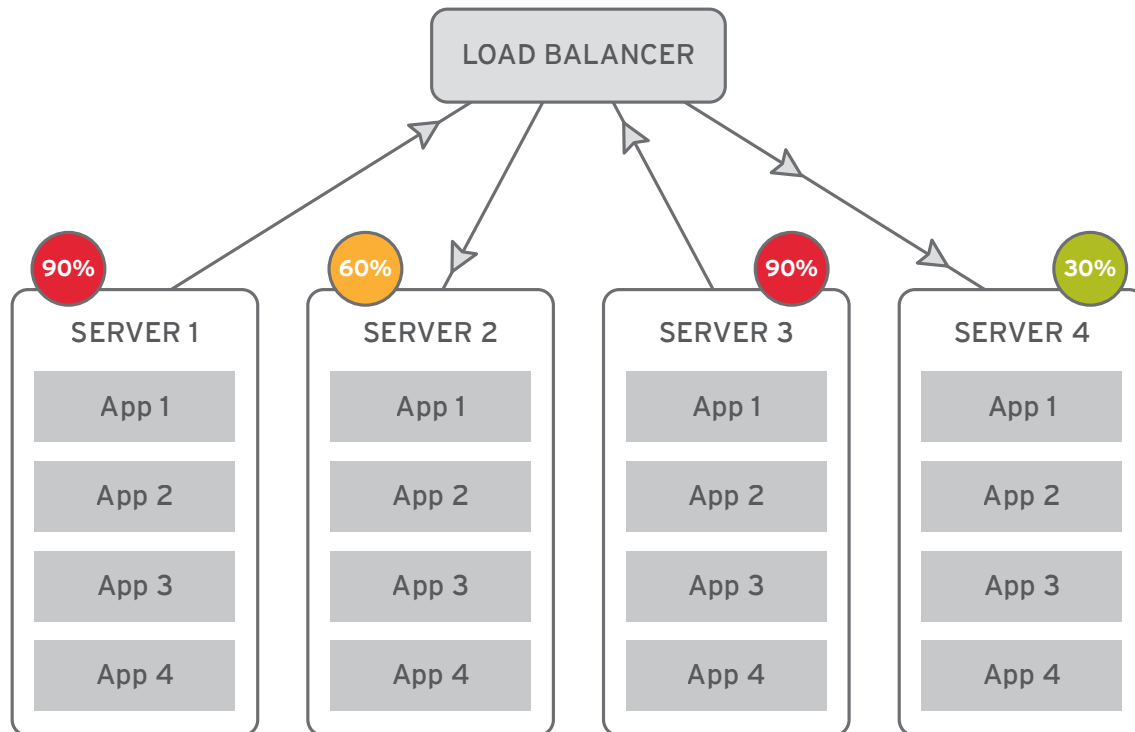
- Single application per application server



This deployment is used when:

- An application requires special configuration of the application server
- You are using a 32-bit operating system or JVM and need the maximum possible memory (less than 4GB per 32-bit JVM) for an application
- With 64-bit operating systems, JVMs, and more isolated configuration capabilities, the preferred deployment is multiple applications to a single server
- Governance or organizational issues that lead to preferring the single application per server model

- Multiple applications per application server



This is a more efficient deployment in many respects. Each application benefits from unused memory that would otherwise be allocated to each separate instance, with sufficient memory to minimize the latency at the application's peak usage. This would not necessarily coincide with peak usage time for other applications.

- Multiple application servers per machine

Multiple instances of JBoss can be deployed to a single machine. This is a common deployment model if you are using a 32-bit operating system on a multi-CPU system with more memory than can be allocated to a single JVM. It can also be used when there is limited large scale hardware, multiple applications requiring special configurations, or the need for failover.

Stateful versus stateless web applications

Stateful web applications use the container-managed HTTP session to store their state and as such require HTTP session replication to be configured on the application server.

JBoss offers various options for state replication. A straight-forward configuration of the clustered cache would result in replication of the state in all participating nodes. More advanced features would enable buddy replication, where nodes are joined with one or more buddies to keep a backup of their state in case of failover. This is particularly useful in large clusters where the replication of state from each node to all other nodes results in a ballooning of memory footprint and can become difficult to manage.

State replication can be configured to take place either synchronously or asynchronously. The former strategy offers reliability and the latter favors performance.

Stateless applications can reside on the same server as stateful applications and simply do not use the replication features. The replication includes constant overhead plus a variable portion that grows with each application state, but is independent of the number of stateless applications.

HARDWARE MIGRATION

There are times when an application server is in some way tied to the operating system or even hardware. The client may have happily decided or have been compelled to use an application platform and operating system or hardware from the same vendor. Under such circumstances, hardware migration will often accompany the software migration to JBoss.

Even when a hardware or operating system migration is not necessary, it may well be an option worth considering. There are arguments both in favor and against migrating the operating system and/or hardware along with the application platform:

ADVANTAGES:

- Share resources between the software and hardware migration, including management and technical personnel, QA environments and network resources used for testing.
- Save time and money by avoiding duplicating the overlapping testing effort. Platform migration already requires thorough testing and evaluation of the environment, tasks that need to be performed for a hardware migration as well.

DISADVANTAGES:

- While it is arguable whether the risk of the combined effort will be less or greater than the sum of the risk of the two separate efforts, it is undoubtedly larger than that of a single type of migration and may therefore surpass a desired maximum threshold.
- When two types of migrations are combined, there is less isolation in changes. Therefore, the root cause of a potential problem will be harder to discover. By the same token, it will be more difficult to attribute any performance gain/loss to one migration versus the other.

Regardless of whether the hardware migration is required or optional, when and if it is performed there is an opportunity to redesign the physical deployment architecture of the servers in a way that can lead to cost savings and various improvements.

There are four primary deployment scenarios that are common to migrations: consolidation, dispersion, aggregation, and cloud migration. These scenarios are not mutually exclusive and can be combined in a large-scale migration to achieve the right balance of functional and operational characteristics for specific workloads.

HARDWARE REQUIREMENTS

Whether it is because of a necessary hardware migration or changes in the software requirements, it may be required to step back and take a look at application constraints to determine the proper hardware. Some questions that should be asked include:

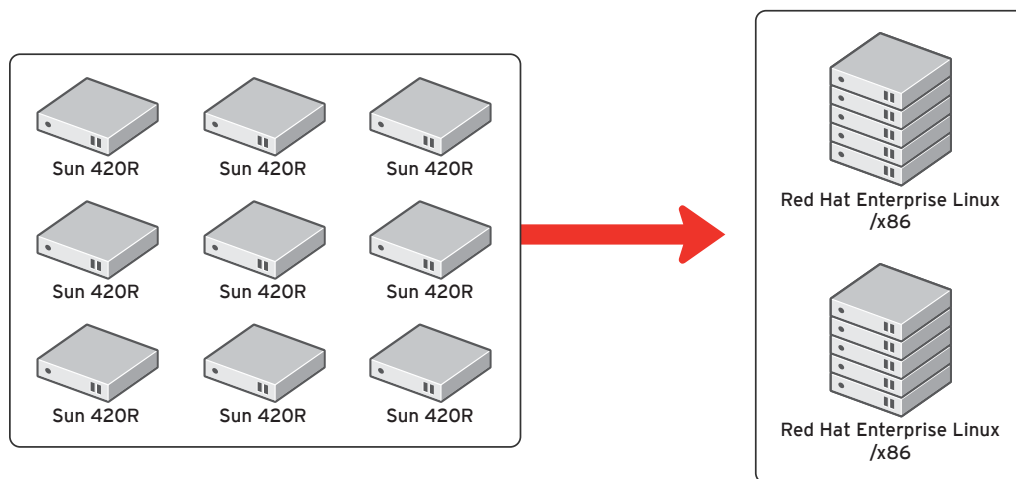
- Is the application mission-critical?
- What service-level agreement (SLA) must be met?
- How critical is the stored data and what type of storage and/or redundancy is required?
- What network topography suits the application and its communication needs?
- How much network bandwidth will be needed?
- What type of caching should be implemented on JBoss?
- What other optimization and tuning is appropriate for JBoss?
- Where is the optimal point in terms of latency versus throughput?
- How large is the required memory footprint and is it too large for a single JVM?

Certain performance expectations will have a direct impact on the choice of hardware and its deployment. For example, virtualization will often increase latency as a sacrifice to gain higher throughput and/or utilization. The same is also true with very large heaps that result in longer garbage collection. Certain advanced garbage collection strategies may alleviate that problem.

CONSOLIDATION

In the consolidation scenario, workloads on a large number of under-utilized systems are consolidated onto fewer systems. These new systems may use virtual machines running open standards-based operating systems to contain each workload. This type of scenario is common in environments where customers have made virtualization of systems a strategic directive. In this scenario, the customer utilizes the chosen virtualization technology to control access to system resources.

CONSOLIDATION SCENARIO



Advantages:

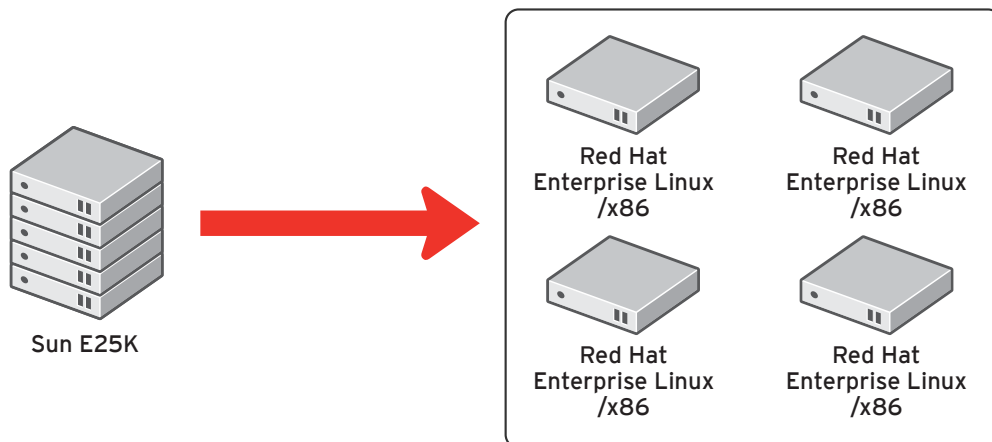
- Reduced hardware operational costs
- Reduced datacenter footprint
- Greater return on investment (ROI) from the chosen virtualization strategy

Disadvantages:

- Proprietary virtualization technologies can increase capital costs and create a new kind of vendor lock-in for the customer

DISPERSION

In the dispersion scenario, workloads on one or more large systems are distributed among a number of smaller systems running open standards-based operating systems. This type of scenario is common in environments where Enterprise Linux® has a growing footprint. Customers can distribute and scale hardware resources in smaller units across multiple datacenters. While 1U to 4U individual rackmount systems have traditionally been common in this scenario, the use of blades has been growing in recent years. Blade servers provide the customer similar advantages with lower operational costs.

DISPERSION SCENARIO**Advantages:**

- High performance from newer x86 hardware technologies
- Low capital cost to scale hardware resources
- High flexibility with deployment and re-deployment of resources

Disadvantages:

- When not properly planned, this scenario can result in high operational costs

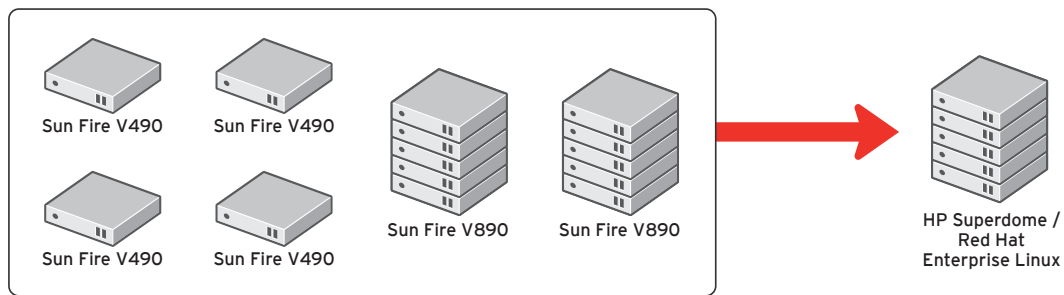
AGGREGATION

In the aggregation scenario, workloads for a large number of systems of various sizes are migrated into a single, large, fault-tolerant hardware platform where Enterprise Linux can be run. This type of scenario is common in environments where the customer already has a high investment in the specific hardware platform, and wishes to further leverage the platform to aggregate legacy platforms using Enterprise Linux. Customers have a choice of using hardware (LPARs, partitioning) or software (zVM, Xen virtualization) to control access to system resources.

Examples of these platforms include:

- IBM® System z® using Integrated Facilities for Linux (IFL) central processors
- HP® Superdome® (Intel Itanium-based)
- Fujitsu® Primequest® (Intel Itanium-based)

AGGREGATION SCENARIO



Advantages:

- Reduced hardware operational costs
- Reduced datacenter footprint
- Greater ROI derived from existing hardware platform

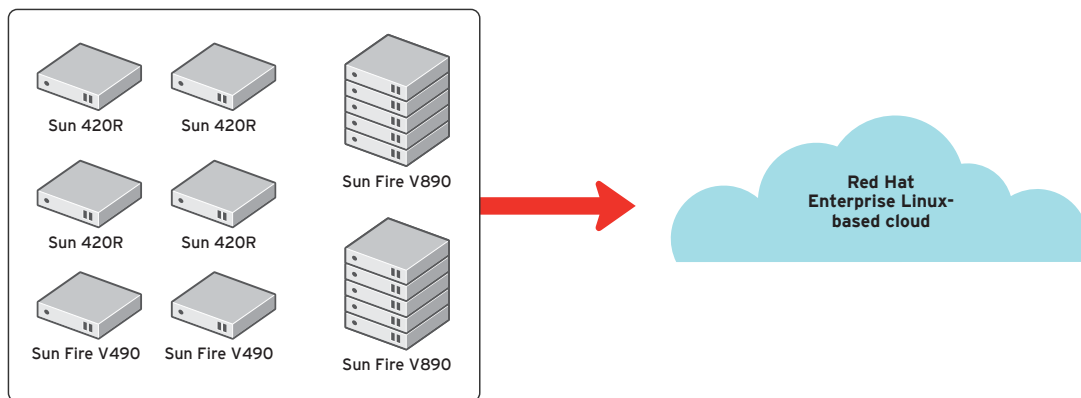
Disadvantages:

- Without prior investment in the platform, customer will incur a high capital hardware cost

CLOUD MIGRATION

In the cloud migration scenario, workloads on any number of systems are migrated to run on open standards-based operating systems in a cloud computing environment. This may be an internal cloud created by the customer, or an external cloud like those offered from Amazon or Rackspace. This type of scenario is very new to most customers, though a small number of customers are moving or have moved their entire operations into a cloud computing environment. Within the cloud, customers have a very high level of control over resources provided to individual workloads.

CLOUD MIGRATION SCENARIO



Advantages:

- Resources can be easily scaled up or down as needed for each workload
- Zero hardware costs (when using a public cloud)
- Low investment cost results in fast ROI (when using a public cloud)
- Higher hardware utilization providing better hardware ROI
- Simplified cloud environment provides for a lower operational cost

Disadvantages:

- Severe outage of cloud or connectivity can cause total loss of access to the operating environment (when using a public cloud)
- Critical data is stored and processed on systems not owned by the customer, so there may be issues of compliance and record-keeping (when using a public cloud)

OTHER CONSIDERATIONS

DIFFERENCES IN JAVA VIRTUAL MACHINE IMPLEMENTATIONS

In theory, various implementations of the JVM follow the same specifications and therefore an application or an application server that has been developed and tested against one JVM seamlessly works with other JVM implementations as well. While this is largely true, there are exceptions and circumstances where differences in implementations cause problems. These problems are sometimes the result of misinterpretation of the specification by a vendor, or at other times simply a defect that has not been uncovered and resolved. Most relate to assumptions made by developers that are not backed by the specification. When faced with uncertainty about the correct JVM behavior, more often than not, developers test the actual behavior and make assumptions based on their observations. Unfortunately such assumptions are only valid for the current JVM vendor and version and may not be part of the explicitly stated specifications. As such, a change in the JVM implementation may often negate such assumptions and lead to the misbehavior of the application.

CLASSLOADER STRATEGY

The JEE specifications have traditionally had little to say about classloading strategy, an unfortunate omission that has led to many challenges in JEE portability. Implementers of the JEE specification have made autonomous decisions on the default class loading behavior for applications that they host, though most have recently made such behavior configurable. Still, the difference in the default behavior alone is enough to cause much confusion and require days, if not weeks, of troubleshooting in search of the culprit.

For example a flat classloader model in JBoss can mean that two different web applications with the same class bundled in their library would actually share the same instance of that class. This is in contrast with the default classloader behavior in WebLogic where that identical class with load in two different classloaders would be considered two different classes.

With JBoss, the static context between the two classes of the two web applications will be shared—whereas in WebLogic they are distinct. This is a significant behavioral difference that can have a large impact, but be difficult to trace.

A related issue is the version mismatch of third-party libraries; applications often bundle the party libraries that are already shipped with the application server. However, depending on the classloading behavior, they may or may not actually be using their bundled version.

It is easy to forget that with the exception of a few cases, “parent-first” is the most common classloader delegation strategy. This means that developers will often bundle a certain version of hibernate, struts, or other commonly used library and use it with a certain expected behavior, unaware that a different version embedded in the application server may actually be the one in effect, overriding the child classloader. This can cause issues resulting from subtle functional differences in various versions of libraries that can be hard to troubleshoot.

3. THE STRATEGIC MIGRATION PROCESS

MIGRATION PROCESS OVERVIEW

PHASE	DESCRIPTION	DELIVERABLES	TYPICAL DURATION
Phase I: Server architecture and deployment model	This phase looks at the inventory of existing servers, network infrastructure, and applications. This creates a baseline of existing functionality and load capacity.	Inventory of application servers describing their purpose and general configuration; network configuration; hardware Inventory	2-4 weeks
Phase II: Application migration assessment	This phase assesses the individual applications for migration. This will examine dependencies on proprietary libraries and adherence to specifications.	Prioritized list of applications suitable for migration. Proprietary libraries used by applications. Specifications involved in migration.	2-12 weeks (highly variable, depending on number and complexity of applications)
Phase III: Effort and risk assessment	This phase looks at the application servers, their configuration, and services deployed. This will determine what functionality will need to be replaced on the servers. It will also assess the amount of effort required for each application and the risks involved in migrating them.	Listing of services configured by server. Clusters defined and the configuration of each cluster. Estimated effort in migrating each application. Identify major risk factors for each application.	4-6 weeks
Phase IV: Server and application migration plan	This phase will plan out the migration for the servers and the applications. This will use the information discovered in the previous phases to produce a roadmap for migrating the servers and the applications. This includes defining the open source alternatives to proprietary libraries and services.	Server migration plan detailing the services and configurations that must be migrated for each server/cluster. Application migration plan detailing library changes and code changes that will be required to migrate each application.	5-8 weeks
Phase V: Migration implementation	This phase involves the actual migration and successful deployment in the new environment. Red Hat Consulting offers a wide variety of workshops, training, and service offerings designed to help customers implement their strategic migration plan.	Successful migration	TBD

PHASE I: SERVER ARCHITECTURE AND DEPLOYMENT MODEL

This phase looks at the inventory of existing servers, network infrastructure, and applications. This creates a baseline of existing functionality and load capacity. This is a critical phase.

INFRASTRUCTURE ANALYSIS

The infrastructure includes the full ecosystem in which the application server operates.

- Datacenters: How many are there? What are their missions? Are they interconnected? Is there a failover strategy between datacenters?
- Hardware: Servers, load balancers, network routers (if clustering), database servers, file servers.
- Software: Load balancing, static web server, dynamic web server, GUI, etc.
- Ancillary services: This will include messaging, portal, AOP, injection, caching, etc.
- Development environment: What IDEs, how are they used, what methodology are they using?
- Applications: What applications run on the servers and which ones are business critical?

INFRASTRUCTURE ELEMENT	
Datacenter	one or multiple
Network	protocols, routers, DMZ,
Hardware servers	load balancers, database, application, web
Applications	web, Java EE, SOA
Development platforms	Eclipse, JBoss Developer Studio, WebLogic Workshop, etc.
Source control	CVS, SVN, GIT, ClearCase
Build	Ant, Maven

PHASE II: APPLICATION MIGRATION ASSESSMENT

In order to be aware of the full scope of migration, it is important to assess what technologies are currently used by the applications and their migration targets. The typical enterprise application makes use of several technologies for security, caching, clustering, and injection. These technologies are largely hidden from the average application developer and they might not be aware of them, but they can have a major impact on the performance and scalability of the application.

Failing to plan for the migration of these technologies can create serious challenges. These will generally crop up at the end of a migration project when performance testing is carried out. So it is vital to assess not only the technologies, but also the specific features used, in order to determine the most appropriate alternative.

COMMON APPLICATION SERVER MAPPING

COMPONENT	FEATURES	JBoss EAP
Monitoring and management	Remote control and configuration, threshold notification, etc.	JBoss admin console, standard JMX interface, threshold alerts
Web server	Servlet container, state replication	Embedded Tomcat, JSP and servlet spec implementation, coarse-grained and fine-grained in-memory replication
Messaging	Participate in transactions, persistence of messages, bridges outside cluster, JMS specifications	JBoss messaging, message distribution and rebalancing, persistent and transactional
Caching	Transactional, distributed, object graph support	JBoss cache, serialized and AOP-based replication
Clustering and replication	Layered replication, transaction-safe replication, fine-grained	JBoss cluster, configuration replication support using JGroups
Persistence	Supported databases, support for flat file systems, hybrid data stores, transaction support, and database feature support	JDBC connection pooling, Hibernate, security data stores, and transactional data sources
Security	Authentication, authorization, single sign-on, JAAS/JACC support, certificate support	JAAS compliant log-in modules, flat-file, database and LDAP support, SSO support
Aspect-oriented programming	Cross-cutting concerns	JBoss AOP
Injection	Standard injection support	Support for JEE injection standards
Presentation layer: JSP, JSF, Facelets, tag libraries	Compatibility and support for presentation standards	Support and implementation of common standards
Transaction manager	Reliable transactional behavior, distributed, support for latest technology	JBoss transactions, reliable transaction support, distributed, support for web service transaction standards

A key element of application migration will be the identification of proprietary libraries that may have been used in the development of the application. JBoss provides a Migration Analysis Tool (MAT) for assessing the amount of proprietary code that will have to be switched out for open source libraries. Details for using it can be found at:

<http://community.jboss.org/wiki/JBossMASS-MigrationAssesmentTool-GettingStarted>

More information is available from:

<http://www.jboss.org/mass/MAT.html>

The next step will be to identify required configurations and potential gaps between existing feature sets and open source capabilities. There are nearly always open source equivalents, or equivalents can be produced with relatively minor customizations.

Finally, an assessment should be made of the difficulty of migration for each application. One should plan to order the migration process from easiest to most difficult. Starting with the easiest applications will build knowledge and confidence in your team.

PHASE III: EFFORT AND RISK ASSESSMENT

After studying the deployment architecture and inventorying the applications and technologies that will be involved in the migration, we are in a position to estimate the effort and risk of the migration. Risk and effort assessment will include both technical and organizational components.

Using the technical analysis from the previous sections, we can begin the technical effort and risk assessment. This will focus on:

1. Technical analysis

- Scope
- Number of servers, datacenters, and applications to be migrated
- Technology gap analysis
- Features that don't immediately map to open source alternatives
- Conflicting configuration requirements
- Adherence to specifications
- Use of proprietary IDEs that embed proprietary libraries
- Library selection

2. Organizational analysis

Organizational factors will generally prevail over technical factors. Technical factors can be easily delineated and addressed. Organizational factors lie beneath the surface and can derail efforts. Seemingly small hurdles can become insurmountable if the organization is unprepared or unwilling to address them. Without a plan for addressing issues, the organization can quickly become focused on minor shortcomings and overlook major opportunities for migrating to open source.

The first step in addressing organizational risk factors is to analyze organizational issues and risks. This will provide a roadmap for preparing the organization for the migration. Organizational readiness factors include.

- Training and knowledge gaps
 - Is the staff knowledgeable about the technology or merely comfortable with existing tools?
 - Has the organization adopted a process for developing and deploying software?
- Workload factors
 - Will the current staff have time to perform current workload tasks and also participate in training and migration work?
 - Will there be sufficient hardware to deploy and test new servers before putting them into production?
- Cultural factors
 - Bottom-up versus top-down decision making
 - High quality versus low cost
 - Leading versus trailing edge
- Budget
 - CAPEX versus OPEX
 - TCO versus ROI

Taking these considerations into account for the migration plan will avoid unpleasant surprises and produce a successful migration.

A strength, weakness, opportunity and threats (SWOT) analysis can help define the organizational readiness. It basically helps the organization match up strengths and weakness, opportunities and threats, and then develop a plan to leverage strengths to overcome weaknesses.

SWOT ANALYSIS

<p>STRENGTHS</p> <ul style="list-style-type: none"> • IT staff trained in relevant standards and technologies • IT staff growing specific JBoss knowledge • Applications follow WAR and EAR formats • Use of open IDEs 	<p>WEAKNESSES</p> <ul style="list-style-type: none"> • Reduced budget • Reliance on proprietary integrated development environments
<p>OPPORTUNITIES</p> <ul style="list-style-type: none"> • Budget cuts • Licensing coming up for renewal • Need to expand capacity beyond the current licenses • Integration of multiple platforms due to acquisition or standardization 	<p>THREATS</p> <ul style="list-style-type: none"> • Non-standards based deployments • No budget for training • All resources are currently fully loaded • No budget for expanding capacity for migration

3. Migration risk assessment

There are risks common to all migrations. By understanding them and preparing for them upfront, we can minimize the risk posed to the migration effort. JBoss offers training, consulting, and tools for migrations.

Migration is a specialized skill that may not be worth developing internally. Even if the migration effort will be extensive enough to justify training an internal team to do it, the company may wish to utilize outside expertise.

RISK	LIKELIHOOD OF OCCURRENCE	POTENTIAL IMPACT	STRATEGY
Training budget	High	High	Formal classes and workshops can be provided to support migration efforts.
Staging hardware	Medium	Medium	It will be important to have hardware similar to the production environment to fully test configurations, applications, and installation.
Use of proprietary IDE tools	Medium	High	Proprietary tools from WebLogic and WebSphere will make heavy use of proprietary libraries. In their code generation, they will generate classes that extend or implement proprietary classes. We can use the JBoss MASS project and MAT in particular for identifying potential migration issues.

PHASE IV: SERVER AND APPLICATION MIGRATION PLAN

Once the risk and effort analysis has taken place, it is critical to begin thinking about how to design a standard operating environment (SOE) upon which to migrate. An SOE is an organization's standard implementation of the core operating system and middleware components. It can include the base operating system, the JEE container, custom configurations, standard applications used within an organization, software updates, and service packs.

Once an application set has been identified, a standardized configuration based on an SOE approach will be created for rapid and consistent deployment. An SOE configuration consists of a set of tested hardware, tested software, and configurations deployed within a JBoss environment. The SOE configuration will be fully aligned to your technical and business requirements, dramatically reduce deployment time, simplify maintenance, increase stability, and reduce support and management costs.

In some cases, multiple SOEs will be required and the next step in the migration plan is to determine how many different server configurations will be necessary to support all the applications after the migration. Multiple server configurations may be necessary because different applications require conflicting configurations, throughput versus latency, special deployments, or because the expected load requires special hardware for some applications. The key is to keep the number of configurations as small as possible while meeting the functional and non-functional goals of your environment.

Once the number of configurations has been determined, it will be important to know which applications will be deployed to which servers. From this point, configuration of the individual applications can begin. Individual applications can be configured to use different class loaders, interceptor stacks, caches, URLs, and many other configuration options.

Plan to provide training for the team to compensate for knowledge gaps that were identified in the earlier risk and effort assessment. With the migration of applications scheduled, training can be planned to coincide with the need for that knowledge in the migration effort. Arranging the training schedule so that the migration team first encounters the problem, works with it, and then gets training in the relevant technology can be a very efficient way of advancing the migration project.

Cost estimation and timeline finalization will be the next items to address. Cost estimation will focus on the following areas:

- Cost of staging hardware that will be used for testing servers and configurations
- Cost of developing extensions to existing open source software to fill gaps when moving off proprietary software
- Migrating applications can be grouped for cost estimation based on whether they need source code changes or just simple configuration changes
- Training costs
- Savings from reduced software costs
- Savings from re-deployed hardware

PHASE V: MIGRATION IMPLEMENTATION

Once a migration plan has been formed, one or more corresponding project plans can be designed and implemented. It is important for major milestones in the project plans to follow those of the migration plan and that training and resource availability conform to the migration plan.

Red Hat resources are available to ensure success in migration and include public and private training as well as development and production support and consulting.

4. ENTERPRISE SERVICES

In the current economic climate, it's critical to make the most of the technologies currently deployed while still looking for opportunities to carve out costs. Red Hat Enterprise Consulting Services provides the expertise and knowledge transfer to help your organization realize a faster time to value and improved migration experience.

ENTERPRISE-CLASS CONSULTING DELIVERED BY SUBJECT MATTER EXPERTS

Partnering with Red Hat Consulting to plan a middleware migration ensures success by combining proven best practices and methodologies with the experience and expertise of Red Hat consultants. With Red Hat, risks are mitigated better, implementation time is reduced, and as a result, the cost of the migration itself is lower. A Red Hat consultant will ensure that the migration team has the knowledge and support needed to complete the job with minimal disruption to IT operations.

Red Hat Consulting has a proven track record helping customers do more with less by fully utilizing the value of their subscriptions. Our global team of consultants includes architects and engineers who are JBoss product experts. Cumulatively, they have years of experience integrating JBoss Application Server into unique and varying environments—always ensuring maximum performance and value.

TRAINING TO IMPROVE PRODUCTIVITY AND PERFORMANCE

By investing in the expertise of your IT staff, you can help ensure optimal system performance, enhance productivity, and mitigate risk. Award-winning training offerings from Red Hat give your team the skills and confidence needed to maximize your open source implementation.

MIDDLEWARE CONSULTING SERVICES

By engaging Red Hat Consulting Services, you benefit from years of experience in the design and deployment of Java-based middleware on both open-source and closed-source software platforms. This expertise in closed-source middleware platforms helps Red Hat Consulting quickly identify and overcome the potential risks and hidden costs of porting applications within your portfolio to the JBoss platform. Red Hat Consulting has unique expertise with the JBoss Platform that helps ensure that your migration results in applications that are designed to work with future versions of the JBoss platform, as well as the current version.

Red Hat Consulting provides a wide range of services around migration to the JBoss Platform, including:

- **Migration assessments** - Review some or all of your current applications to identify costs, risks, and detailed estimates of the effort required to migrate your applications to the JBoss platform.
- **Application migrations** - Convert your existing applications from closed-source middleware systems to the open JBoss Platform to help reduce licensing costs, minimize vendor lock-in, and increase open standards compliance, where possible and cost-effective.
- **Architecture reviews and recommendations** - Provide ongoing guidance for your JBoss use and future application development. Help align your technical roadmap with the standards and features planned for future versions of the JBoss platform.

And because each project is unique, Red Hat Consulting will work with you to understand your current environment and the most appropriate scope of work to successfully migrate your applications to JBoss and maintain and enhance them in the future.

Red Hat Consulting uses developed tools, proven methodologies and experience gained in the field through numerous migrations to help guide clients through a smooth and safe process.

PLATFORM CONSULTING SERVICES

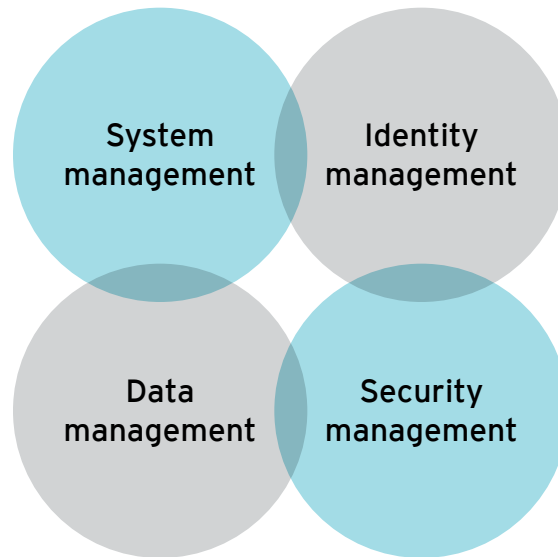
With all migration efforts, having a solid infrastructure that provides a scalable foundation is the first step. Red Hat infrastructure migration planning services provide a detailed evaluation of your IT environment and deliver strategic recommendations for simplifying your IT infrastructure as you migrate. The result? You can reduce IT costs while creating a scalable IT infrastructure.

Red Hat provides a foundation based on the standard operating environment (SOE) approach, in order to ensure a successful migration and a solid foundation for your organization's continued growth.

BENEFITS OF AN SOE

- **Simplified architecture:** One codebase that can be deployed on different branches and services. Support different platforms (workstations, servers, or mainframes) from the same build process.
- **Flexible and rapid deployment:** Grants the ability to take a server from bare-metal to fully-configured in less than ten minutes. Ensures identical configuration and the ability to compare machines from a centralized GUI interface, which is useful when searching for anomalies.
- **Security:** Enforce security policy across different machines and distributed datacenters.
- **Centralized management:** Manage different types of machines with different functionality remotely. Also includes the ability to delegate responsibility to regional or provincial management.

Centralized configuration management: Enforce configuration, schedule configuration updates, compare configurations, and query current configuration.

STANDARD OPERATING ENVIRONMENT DIMENSIONS

Systems management: Evaluates and documents current systems management infrastructure. Recommendations will be provided regarding the management of systems and software post-migration and how to incorporate Red Hat Enterprise Linux into existing change management processes and systems.

Service includes:

- Bare-metal and virtual platform provisioning
- Linux software build and deployment
- Monitoring and performance management

Identity management: Determines and documents current identity management policy. Recommendations are provided for integrating Enterprise Linux systems into existing authentication and authorization infrastructures or for migrating existing directory solutions to open source software.

Service includes:

- User and group management
- PKI infrastructure
- Policy creation and enforcement

Data management: Determines and documents availability requirements for migrated services. The architect will design a strategy for meeting those requirements with a mixture of storage and clustering technologies.

Service includes:

- High-availability clusters
- Distributed file systems
- Load-balancing solutions
- Disaster recovery
- Systems and data backup
- Data recovery
- Bare-metal recovery

Security management: Identifies and documents current corporate security practices and procedures for Linux as well as requirements for migrated services. A thorough understanding of the end-user requirements is necessary.

Service includes:

- Operating system hardening
- Emergency security errata patching
- Security auditing and reporting
- Compliance requirements and remedial action

Within each of the above areas, a gap analysis is performed to assess the existing infrastructure and processes that support the Enterprise Linux operating system versus the support of other operating systems within your IT environment. This analysis is conducted using industry-standard practices and industry-proven methodologies.

One of the additional benefits of these tasks is that Red Hat works side-by-side with your team members to provide hands-on mentoring, real-time knowledge share, and valuable guidance as your teams encounter issues or have questions.

Red Hat Consulting offers a comprehensive suite of end-to-end solutions to help your business realize the benefits of your investment faster – no matter where you are in your deployment cycle.

RED HAT CONSULTING SOLUTION	DESCRIPTION
Assessment	Combines proven best practices with the expertise of Red Hat Consultants to plan a safe, stable migration.
Quick Start	Accelerates project completion and time to value.
Implementation	Comprehensive installation, configuration, and deployment of new technologies.
Health Check	Validates installation and configuration of the technology to identify issues that impact your business.
Optimization	Troubleshoots and resolves issues, thus increasing business effectiveness and reducing costs.



If you're ready to begin your migration initiative, e-mail or call us, and we'll have a conversation to determine how we can best support you and your organization.

Tel: 1 (866) 273-3428 x44555 or redhatconsulting@redhat.com

TRAINING

When migrating platforms, it is critical to ensure that you have a skilled staff who can maximize performance beyond initial deployment. Hands-on training is suggested to teach organizations optimal management techniques, effective troubleshooting, and the ability to maintain improved efficiencies across the entire system. Training leads to rapid, successful deployments and ensures your staff has the skills and knowledge to keep your IT organization running smoothly.

Red Hat delivers a complete curriculum of JBoss Enterprise Middleware training offered in the classroom, at corporate on-sites, and in online settings. All JBoss courses are taught by JBoss Certified Instructors (JBCIs) and include a balance of essential concept-based lectures with real-world task- and project-oriented labs, ensuring your team will acquire the real-world skills that are required for success.

Regardless of experience level or training goals, Red Hat Training has the right course and training path that will build on and leverage existing industry experience.

Delivery methods: Red Hat performance-based training provides hands-on, real-world skills that IT professionals need to design, execute, and maintain successful open source infrastructures.

- **Classroom training:** Offered in more than 45 locations across North America
- **Virtual training:** Real-time training conducted by instructors in an online environment.
- **On-site training:** Training led by Red Hat certified instructors located at your company's facility

Learn more at redhat.com/explore/ways_to_train.

Certification: Certification helps measure your readiness and provides an entire ecosystem of experienced professionals to augment your migration strategy. The JBoss Certified Application Administrator (JBCAA) is the only performance-based certification in its category, continuing the Red Hat tradition of benchmarking the skills of IT professionals through hands-on skills assessments.

Learn more at redhat.com/certification/jbcaa.



Courses: The following table lists standard courses offered by Red Hat for JBoss Enterprise Middleware. Customized education opportunities are also available on a per-customer basis.

COURSES

JBoss Application Administration (JB336)

Red Hat's most popular JBoss course, JBoss Application Administration, focuses on installation and deployment issues with the JBoss Application Server as well as configuring the server for production usage.

JBoss Enterprise Application Development: (JB295)

Intended for entry to mid-level Java developers, the JBoss Enterprise Application Development course will expose students to JBoss Java EE frameworks, specifications, and interfaces (APIs).

Advanced JBoss Enterprise Development (JB325)

JBoss Enterprise Development dives into the JBoss Enterprise Application Platform (EAP) with an emphasis on advanced Java EE application programming interfaces (APIs).

JBoss Seam Development (JB311)

The JBoss Seam Development course teaches experienced Java developers how to efficiently use Seam to intelligently tie components together and manage increasingly complex IT systems.

JBoss Hibernate Technology (JB297)

The JBoss Hibernate Technology (JB297) course gives Java developers the knowledge and skills required to leverage the powerful Java Hibernate Application Stack.

JBoss Enterprise SOA (JB341)

The JBoss Enterprise SOA course provides developers with real-world examples, integration patterns, and strategies for integrating enterprise systems and legacy applications via a common, services-based architectural style.

To access a complete, interactive, or PDF/printer-friendly version of the complete JBoss Enterprise Middleware course catalog, please visit <https://www.redhat.com/training/catalog/>.

Red Hat training specialists can help identify your staff training requirements and what level of training is needed. Contact Red Hat at training_sales@redhat.com to craft a custom corporate training plan to meet the needs of your group.

5. SUCCESSFULLY MIGRATED CUSTOMERS

AVIS EUROPE

Industry: Automotive: Car rental

Geography: EMEA



Business challenge: To replace Avis' costly proprietary software with reliable open source solutions to reduce acquisition and ongoing costs for the organization's IT department

Migration path: Proprietary Sun hardware with Solaris and Weblogic-based application server platforms were migrated to a more cost-effective and flexible platform based on JBoss Enterprise Application Platform running on Windows Server 2003

Hardware: Wintel environment incorporating Dell hardware, Intel-based dual-CPU, quad-core machines with 8-16G of RAM, Windows Server 2003 32-bit and 64-bit

Software: JBoss Enterprise Application Platform, Hibernate framework

Benefits: Experienced significantly reduced TCO of at least 50 percent from reduced maintenance costs, as well as increased licensing flexibility and reconfigurable and decreased server load

"Avis has absolute trust in JBoss and its mission-critical enterprise applications. We are very satisfied with the results that we've achieved since migrating to JBoss solutions."

-Jens Utech, Director Revenue and Fleet Applications at Avis.

BACKGROUND

Avis Europe is a leading car rental company with a presence in Europe, the Middle East, Africa, and Asia. It serves over eight million customers per year via the Avis and Budget brands in over 3,800 locations. Avis holds leading positions in many markets in Western Europe, with approximately 80 percent of its revenues generated in the five major markets of France, Germany, Italy, Spain, and the UK.

BUSINESS CHALLENGE

Avis' revenue management system, undoubtedly the organization's predominant application affecting main business function, is used to conduct forecasting for business areas including future demand for car rental, the best placement of cars for rental services, what types of customers to target, and pricing decisions. Used in over 500 stations across Europe, the system needed to be upgraded and simplified to enable ease-of-use.

Avis' station application allowing agents to check in cars via handheld devices and the company's system for printing rental agreements both also needed to be optimized. The previous processes were conducted in-station using non-compliant, non-customer friendly, and incumbent systems.

In 2006, Avis recognized the need to replace these expensive, incumbent systems based on proprietary software with a reliable and affordable alternative. Though it had little experience with open source solutions, it decided to migrate its proprietary Solaris and combined BEA and Weblogic application server platform based on Sun hardware to the more cost-effective and flexible platform offered by open source solutions.

SOLUTION

Avis evaluated a number of open source solutions that could offer reduced TCO, expanded flexibility, and unmatched scalability. Avis required a Java EE-compliant application server on the Windows platform that offered support and a related monitoring and management infrastructure.

Ultimately, Avis selected a Wintel environment combining Dell hardware, Windows Server 2003, and JBoss Enterprise Application Server. The JBoss Enterprise Application Platform also incorporated the use of JBoss Operations Network for monitoring activities.

The solution ecosystem consists of Windows Server 2003 32-bit and 64-bit, JBoss Enterprise Application Server, and SQL Server 2005. The overall solution includes ten production servers used for the organizations revenue-management systems. Half of these solutions are based on JBoss Application Server, and all use the Hibernate framework.

The migration involved no integration with the incumbent platform, and was a full-system port. The process operated on a phased, three-year timeline with iterative releases occurring approximately every three months. No system integrators or value-added resellers were leveraged during the migration process, as all development was done in-house.

Avis used consultants to aid the smooth implementation of JBoss Operations Network Server monitoring and management. This allowed the company to rapidly incorporate the tool without having to invest heavily in administration and installation internally.

BENEFITS

Avis selected JBoss solutions for the offering of reduced TCO and increased performance, flexibility, and availability of resources with familiarity in the market.

One of the distinct advantages of the JBoss Enterprise Application Platform is that it enables ease-of-use for development. The solution integrates easily with IntelliJ IDEA and Avis' development cycle, and Avis employees are now able to run the JBoss platforms on their personal laptops.

The platform has enabled significant development efficiency and developers are more effective running JBoss in their processes and taking advantage of the full IDE integration and very rapid start-up times. The time required for start-up with the new JBoss-based system has been reduced considerably. Now, with JBoss Enterprise Application Platform, Avis can quickly and easily add new servers to its architecture. The new architecture has allowed Avis to buy, deploy, and re-deploy additional servers at very competitive prices.

In addition to reduced TCO amounting to savings of at least 50 percent of original costs, Avis also achieved enhanced performance from its newly implemented JBoss platform.

“It’s now definitely easier to scale a number of CPUs up and down in the current model and the system really brings us a lot of value, especially in terms of efficiency and effectiveness in the development process. We have saved at least 50 percent on TCO,” said Jens Utech, Director Revenue and Fleet Applications at Avis. “Thanks to JBoss’ integration with the development tools and the ability for the developer to work in very flexible ways, common development tasks around deploying, re-deploying, and generally working are three to four times faster with JBoss than any of the competitors.”

“Avis has absolute trust in JBoss and its mission-critical enterprise applications. We are very satisfied with the results and we are also impressed with JBoss support. We are provided with very professional, very prompt, and very helpful answers, as opposed to the usual vendor nonsense. We use the support portal website as well,” said Utech.

In addition to all of the valuable benefits delivered by JBoss solutions, Avis also found that many of its development staff perceived the use of JBoss and open source solutions in general as very positive. This has led to increased motivation and expanded recruiting for Avis’ internal staff.

GEICO (GOVERNMENT EMPLOYEES INSURANCE COMPANY)



Industry: Insurance

Geography: US

Business challenge: Existing proprietary middleware platform was complex to manage, not performing and scaling as expected, and expensive to maintain. The architecture team decided to investigate alternatives that could be deployed that would better meet their needs.

Migration path: Proprietary middleware platform to JBoss Enterprise Middleware

Software: JBoss Enterprise Application Platform: 28 bands (1 band = 32 CPUs), JBoss Technical Account Manager (TAM), Red Hat Consulting, Amentra

Hardware: 50 Dell servers

Benefits: Reduced the TCO by more than 30 percent, throughput gain of 3X with utilization down to a third of the current platform, overall resource utilization went from above 50 percent to under ten percent, which allowed significant room for scalability without having to acquire additional hardware.

BACKGROUND

GEICO (Government Employees Insurance Company) is the third-largest private passenger auto insurer in the United States based on the latest twelve months written premium. GEICO provides auto insurance coverage for nearly nine million policyholders and insures more than 14.4 million vehicles.

In addition to auto insurance, GEICO also offers customers insurance for their motorcycles and homes. Commercial auto insurance, boat, ATV, RV, personal umbrella protection, and life insurance are also available.

GEICO is a wholly-owned subsidiary of the Berkshire Hathaway group of companies, is rated A++ for financial stability by A.M. Best Company, and ranks at the top of several national customer satisfaction surveys. For more information about GEICO, go to www.geico.com.

BUSINESS CHALLENGE

GEICO's enterprise architecture team recognized they were facing several challenges with their existing proprietary middleware platform. The platform was complex to manage, not performing and scaling as expected, and expensive to maintain. The architecture team decided to investigate alternatives that could be deployed that would better meet their needs.

The GEICO IT team identified the following challenges with their existing proprietary solution:

- **Cost** - GEICO's license agreement was a "time bound licensing agreement" related to the number of proprietary application servers deployed during the time frame. Since GEICO experienced significant growth during this timeframe, the cost to "true up" and pay for the additional licenses was significant.
- **Performance** - When GEICO upgraded their standard Java Development Kit (JDK) from version 1.4 to 1.5 on their existing proprietary platform, they did not see any improvements in machine (CPU/Memory) usage or application response time. After eight weeks of performance testing and tuning, they were finally able to configure the upgraded proprietary platform to match the earlier version's performance. The upgrade was not only cumbersome but was also expensive since they had to engage external consultants to accomplish the upgrade.
- **Memory leaks** - The previous proprietary deployment also experienced unexplained memory leak(s). Developer load and memory testing returned misleading results unless the developer knew how to work around the leaks and complete certain types of tests.
- **Documentation and support** - GEICO found it challenging to identify and understand the Java API in the current proprietary environment due to lack of documentation. They also had challenges in acquiring tools to identify memory issues, debug leaks, etc. For every instance of a high-severity issue such as memory leak, external consultants needed to be engaged to identify and fix the problem.
- **Staging** - Due to these challenges, some of the GEICO development teams adopted JBoss technologies for their developer workstations and began building applications using JBoss. This dual-use strategy became complex and redundant for IT Operations as they needed to make configuration changes on both the proprietary and JBoss platforms.

GEICO conducted extensive research and identified Sun's GlassFish and JBoss Enterprise Middleware from Red Hat as potential solutions that were suitable for GEICO's applications and infrastructure. JBoss Enterprise Middleware was selected based on its market share and extensive support from Red Hat. GEICO conducted a proof-of-concept, installing JBoss Enterprise Application Platform on a cluster of servers (POC environment). Performance and load tests were conducted using various tools for a selected business application on both platforms.

The JBoss results from these tests were astonishing. A few highlights include:

- User page transition time decreased as much as 19 seconds using JBoss
- During the proof-of-concept, 1,749 additional business processes were created on the JBoss platform
- On the same hardware and environment, JBoss required 70 percent less CPU resources than the current platform
- Performance tuning with JBoss was accomplished in 40 man hours versus 1,440 man hours for the existing proprietary platform



GEICO also conducted multiple reference checks with organizations that were of similar size and industry. The reference checks were extremely positive about Red Hat and JBoss Enterprise Middleware.

SOLUTION

The solution consisted of subscriptions for JBoss Enterprise Application Platform (EAP) and the initial deployment environment consisted of 540 processors. An additional 350 were added at a later date. A plan was put together to aggressively migrate two out of three mission-critical applications in a time span of three months. GEICO also utilized a JBoss Technical Account Manager (TAM) who was dedicated to supporting GEICO's specific needs during their switch to JBoss.

BENEFITS

By implementing JBoss Enterprise Application Platform, GEICO was able to reduce the total cost of ownership by more than 30 percent.

When compared to the previous proprietary platform, GEICO also experienced throughput gains of 3x, and a two-thirds reduction in utilization. The overall resource utilization went from above 50 percent to under ten percent, which allowed significant room for scalability without having to acquire additional hardware.

One of the challenges for GEICO was the time-bound migration process. GEICO's middleware team was trained on JBoss for a week. With the support of Red Hat and Amentra (a Red Hat company), they successfully migrated the initial two applications and were able to migrate the third application as well. This was a clear demonstration of expertise in Red Hat Consulting services and the ability of GEICO's middleware team to adapt rapidly to the new JBoss environment.

GEICO had initial concerns about support, stability, and deploying open source software for its mission-critical applications, but the market maturity and the premium level of enterprise support offered by Red Hat made it very easy to make the change to an open source environment.

HEIDELBERGER DRUCKMASCHINEN AG



Industry: Printing and Publishing, Manufacturing and Mechanical Engineering

Geography: US, EMEA

Business challenge: Establish a new customer-specific portal to enhance customer relationships and provide internal business units with valuable customer insight; Evaluate and deploy an application server platform for its Java-based environment

Solution: With the help of Red Hat Consulting, Heidelberg built an interactive customer portal using JBoss Enterprise Portal Platform and migrated its proprietary application platform to JBoss Enterprise Application Platform

Software: JBoss Enterprise Portal Platform, JBoss Enterprise Application Platform, Oracle 10g database, SAP R/3, Autonomy search engine, Interwoven TeamSite 6.5 (CMS), Microsoft Windows Server 2003 64-bit

Hardware: Virtualized servers and F5 load balancer

Migration path: Proprietary application server and portal platform to JBoss Enterprise Application Platform and JBoss Enterprise Portal Platform

Benefits: Improved customer web presence with a central portal that increased services and information available; provided the ability to use applications based on open standards in order to interact better with customers, suppliers, and partners; reduced costs, gained excellent support, and continuous advancements of its portal software; and reduced vendor lock-in.



BACKGROUND

Heidelberger Druckmaschinen AG (Heidelberg), the world's leading solution provider for the print media industry, built an interactive customer portal, www.MyHeidelberg.com, using JBoss Enterprise Portal Platform to create an interactive site to enhance its customer relationships and provide a unified global brand presence. With the successful portal project, Heidelberg has since migrated its proprietary application platform to JBoss Enterprise Application Platform, in order to reduce costs, and increase the performance of its web applications.

Based in Heidelberg, Germany, with development and production sites in seven countries and around 250 sales offices across the globe, the company supports around 200,000 customers worldwide. Heidelberg's printing products and software enable its customers to print high volumes, quickly, at high quality and in color at an affordable price and convert the printed material into the desired format.

BUSINESS CHALLENGE

Heidelberg launched a corporate-wide web initiative to develop a strategic architecture and customer-centric web portal, to strengthen customer relationships, consolidate information and services, provide easier access to information, and increase revenues through improved communications with customers.

"Previously, we had a solid web presence for our customers, but it was more of an online brochure and lacked interactivity," said Mathias Berg, Director, Global Web Systems at Heidelberg.

Heidelberg also anticipates the ability to lower its costs by offering more self-service options, reducing the effort and cycle time for developing and deploying new services and applications, and simplifying support and operation of its applications and infrastructure. It also wanted the new portal to provide its customers with a central and secure repository for sharing data and communication.

"With MyHeidelberg.com, we have created a unified, central customer portal that can be rolled out and adapted to local markets, thus increasing our interaction and providing us with valuable customer insight," said Michael Neff, Chief Information Officer.

SOLUTION

In order to further its edge in the market, provide customers with more features, and enhance its web presence, Heidelberg decided to evaluate a number of proprietary and open source solutions for the customer portal project, MyHeidelberg.com. This was in addition to an application platform solution to set up and restructure its global web presence.

"In addition to JBoss, we evaluated proprietary and open source solutions, but we selected JBoss Enterprise Portal Platform on JBoss Enterprise Application Platform due to its ability to deliver an agile, reusable framework, coupled with the benefits of open source to reduce costs, and free us from vendor lock-in," said Berg, "Although we had limited open source experience, we made our decision confidently due to JBoss' established technology platform, flexible architecture, highly regarded enterprise support, and the expertise of Red Hat Consulting."

JBoss Enterprise Portal Platform, based on open standards, was selected for its ability to meet the key requirements of the project, including: the ability to integrate seamlessly into the existing infrastructure based on VMWare ESX 3.0 and Oracle 10g, and the content management systems: Windows Server 2003 and Interwoven TeamSite 6.5.

The MyHeidelberg.com customer portal needed to be flexible and reliable in order to adapt to Heidelberg's local website requirements, while at the same time, it had to scale for a steadily increasing number of users, which initially tallied more than 5,500 registered users and has been growing monthly by approximately five to ten percent.

At an early stage of the customer portal project, Heidelberg made the decision to work with Red Hat Consulting in order to facilitate knowledge transfer, have a closer relationship with JBoss, and to have access to the engineers and developers behind the open source technology. Heidelberg's IT teams in the U.S. and Germany worked with Red Hat consultants to enable a smooth integration of JBoss technology into the existing global infrastructure.

"We had a great experience with Red Hat Consulting," said Berg. "The Red Hat Consultants worked closely with our IT teams in the U.S. and Germany to ensure a smooth integration of JBoss technology into the existing global infrastructure, by mentoring our team, demonstrating product features, providing documentation, and ensuring our team was prepared for success after the engagement. We view Red Hat Consulting as a trusted advisor to our team."

Through back-end integration with ERP and CRM systems, Heidelberg enables portal users to securely interact with their data, such as service information, and view and retrieve brochures, product specs, and account data easily through one gateway. Additionally, the JBoss Enterprise Portal Platform runs on Windows Server 2003 and collaborates with a content management system based on Interwoven TeamSite 6.5.

Heidelberg decided to expand its collaboration with JBoss when it decided to migrate its complete internet presence, www.heidelberg.com, from a proprietary application platform to JBoss Enterprise Application Platform, restructuring its entire global web presence.

"Based on our successful deployment of JBoss Enterprise Portal for the MyHeidelberg.com project, and knowing that we had a stable, reliable, and fully supported solution, we quickly and confidently made the decision to migrate our global enterprise web applications from our previous proprietary solution to JBoss Enterprise Application Platform," said Berg.

BENEFITS

Heidelberg's use of JBoss for its customer portal and web application server has resulted in improved customer web presence with a central portal, reduced costs, excellent support, and continuous advancements of its portal software, as well as reduced vendor lock-in.

"All over the world, the name Heidelberg is synonymous with state-of-the-art technology, top quality, and closeness to the customer, so we set the benchmark high for ourselves and we expect the same level of quality from all of our vendors," said Neff. "JBoss absolutely delivered all we needed by providing high availability, performance, and scalability. Due to the Red Hat subscription model, we realized significant cost savings, faster ROI, and freedom from vendor lock-in."

The introduction of JBoss has helped simplify IT administration. With only one portal infrastructure to manage, one platform, and one brand for the company's web presence, this simplicity strengthens the brand of Heidelberg. The new self-service capabilities provided by the MyHeidelberg.com portal give customers fast, convenient access to critical information whenever they need it, increasing customer satisfaction and the credibility of Heidelberg as a valuable business partner.

“In addition to the enhancements for our customers, the MyHeidelberg.com portal based on JBoss Enterprise Portal, offers valuable customer insight and behavioral information that is being leveraged internally by our marketing, sales, and product development teams, which will allow us to make informed and targeted business decisions and increase company sales,” said Berg.

The cost of developing and deploying new services and applications has been reduced now that Heidelberg has a simple way to integrate content and information into the portal. The IT team can index content and expose it through the search capabilities and can manage single sign-on. Also, JBoss has drastically reduced costs, as the subscription model has freed Heidelberg from the cost-constraints associated with proprietary software licensing and support.

“We were especially impressed with the performance and product features of JBoss and the quality of support, advice, and knowledge contributed by our Red Hat consultants,” said Neff, “The JBoss Enterprise Portal and JBoss Enterprise Application Platforms went into production and performed extremely well, and this is a great benefit for our customers and internal users, who can now access their data securely and without restrictions.”

“JBoss was always willing to work with us, as partners, to ensure the success of the project,” said Berg, “A prime example is when we identified the need for a unique feature for our customer portal, With any other vendor, we would have had to develop a custom feature with no vendor support. But with JBoss, we discussed our need, and they worked with us develop a new feature into the product, that will be fully supported by JBoss. This kind of commitment and performance is something that I have never seen before, from any technology vendor.”

DAIWA SECURITIES AMERICA INC.

Industry: Financial Services

Geography: New York

Business challenge: Build an utterly reliable, high-performance portal using open source to speed application development while reducing costs

Migration path: Proprietary application server and portal solutions to JBoss Enterprise Application and Portal Platforms.

Software: JBoss Enterprise Portal Platform, JBoss Enterprise Application Platform and JBoss Rules Framework. Databases: Sybase SQL (transactional), Sybase IQ (data warehousing for production environment), Microsoft SQL (training data), Alfresco Software for enterprise content management (ECM).

Hardware: Intel dual-core Intel Xeon HP servers

Benefits: Established a reliable platform with no unplanned downtime, speed time-to-market by cutting application development time in half, dramatically improved application performance, and saved \$300,000 in licensing and hardware costs

Daiwa

"If our portal were to go down today, it would be catastrophic. JBoss Enterprise Application and Portal platforms have been extremely reliable and stable. JBoss is a worry-free solution that helps us sleep better at night."

–Steve Dunstan, Enterprise Systems Architect and Vice President,
Information Technology, Daiwa Securities America Inc.

BACKGROUND

Daiwa Securities America Inc. is a privately held, wholly owned subsidiary of Daiwa America Corporation, owned by Daiwa Securities Group Inc., one of Japan's big three securities brokerages. The subsidiary focuses on sales and trading of Japanese and U.S. equities and fixed-income instruments, financial futures, and investment banking. For the year ending March 31, 2008, Daiwa Securities America posted net operating revenues of \$2 billion.

BUSINESS CHALLENGE

In 2006, Daiwa Securities America was having trouble keeping its internal portal up and running. The portal was the lifeblood of the firm's operations. Because Daiwa's 250 employees depended on it to access the applications they needed to do their jobs, the frequent crashes created serious business and technical problems. "It got to the point where we couldn't keep the server up for more than a week at a time," said Tom Cordova, senior vice president, Information Technology, Daiwa Securities America Inc.

The portal, called DSAweb, supported 120 applications ranging from a "Compliance Dashboard" to back-office integration systems. "When it crashed, we'd be down for as long as 90 minutes at a stretch. Our users wouldn't be able to do what they needed to keep our business running. You can't get more mission-critical than that," said Cordova. The problem resided in the popular industry-standard application server that Daiwa had used to build the portal.

The company prides itself on being at the bleeding edge of technology, and had pushed the platform to its limits. But when the maker of the application server released a new version with much-needed functionality, it was loaded with bugs, frustrating Daiwa application developers and delaying application time-to-market. Even more significantly, Daiwa couldn't get adequate support. "The vendor suggested we needed to purchase additional hardware and software licenses to solve the problems we were having," said Cordova. "We thought that was outrageous. That's what pushed us to find another solution."

SOLUTION

Daiwa already had some experience with JBoss products in other in-house development projects. Cordova was impressed with what he saw and downloaded the free version of the JBoss.org Portal project. It took only a short trial before he decided to build the new portal using JBoss. Because of the mission-critical nature of DSAweb, Cordova purchased a subscription to the JBoss Enterprise Portal Platform to benefit from the stability and reliability of an enterprise class platform that included support, patches, and updates as well as a multi-year maintenance policy.

First, Daiwa successfully migrated the 120 DSAweb applications – most of them database-driven – from the previous application server to JBoss. This monumental task was accomplished within just 12 months by developing a template using Velocity, an open source templating tool, that described how each application “mined” the database and which then regenerated the applications on JBoss Portal. In the same timeframe, Daiwa also developed and brought online 80 new applications – something that would not have been possible using the old platform.

“With JBoss, we’re a much more agile company,” said Steve Dunstan, enterprise systems architect and senior vice president, Information Technology, Daiwa Securities America Inc. “Today, if the business needs a new application, we can deliver it immediately. That makes everyone happy.”

Daiwa depends on the breadth and depth of the JBoss Enterprise Middleware portfolio to meet its business objectives. In addition to the JBoss Enterprise Portal Platform, the JBoss Rules Framework allows Daiwa to embed alerts into portal applications that notify employees of unfinished trade allocations, unusually large trades, or customers nearing their credit thresholds. Daiwa also leverages the caching functionality in the JBoss Enterprise Application Platform to seamlessly and securely share information across applications, and with Alfresco’s open source content management system, provide employees with single sign-on access to all the applications they need. In addition, Daiwa uses the message queuing functionality in the JBoss Enterprise Application Platform with Informatica PowerCenter to asynchronously send messages to all distributed applications in its service oriented architecture (SOA).

BENEFITS

With JBoss, Daiwa’s IT team now focuses on helping users be more productive. Additionally, the organization can be more competitive rather than simply being in technical “fire fighting” mode.

Reliability is the bottom line. “If our portal were to go down today, it would be catastrophic,” said Dunstan. “The JBoss platform has been extremely reliable and stable. We have had no unscheduled downtime.”

Reduced cost and improved performance have also been big wins. Daiwa saved \$300,000 by avoiding software licensing and hardware expenses that would have been incurred by sticking with the previous solution. “That’s a huge savings for us,” said Cordova. Applications are also easier and less costly to maintain, and when bugs are found, Daiwa can quickly propagate the fixes. Application performance has dramatically improved too. Developers can compile and roll out new applications in seconds, rather than the 15 minutes required by the previous platform.

Employees are more productive because the portal applications load faster – in just one second compared to the ten seconds required with the old solution. Applications are also easier-to-use and require less employee training because the interfaces and functionality are consistent across all applications.

Developer productivity has also increased dramatically. The ability to develop applications using open source code allows developers to create new applications in half the time. “And shorter development cycles translate into a faster time-to-market,” said Dunstan. “We wouldn’t have been able to grow these services and capabilities for the business without accelerating our development cycle.”

The transparency of open source code and Red Hat’s dedication to open standards simplifies developers’ work as well. “When we have problems, we like to work them out quickly in-house,” said Dunstan. “We can troubleshoot issues more easily with open source code.”

When Daiwa does need outside help, JBoss support from Red Hat has been outstanding. “When you contact JBoss technical support, you get to the right person immediately – sometimes even the person who wrote the code you’re using,” said Dunstan. He constantly marvels at the fact that he has achieved 100 percent resolution on all support issues with Red Hat. “With other enterprise vendors, I feel lucky if I get 40 percent of my issues resolved,” he said.

Daiwa is starting a new project to develop Web 2.0 applications and will use a clustered server environment, which it plans to monitor using JBoss Operations Network.

In moving to JBoss, Daiwa has decreased its risk while getting very big returns. “JBoss is a worry-free solution that helps us sleep better at night,” said Dunstan.

SWEDISH NATIONAL POLICE BOARD

Industry: Government

Geography: Sweden

Opportunity: To replace costly proprietary software with reliable open source solutions to reduce acquisition and ongoing costs for the organization’s IT department.

Migration path: Proprietary software and hardware, including Hewlett Packard PA-RISC chip-architecture, HP-UX Unix operating system, Oracle database, and BEA WebLogic Server to open source solutions, including JBoss Enterprise Application Platform

Software: JBoss Enterprise Application Platform, MySQL Enterprise Server 5

Hardware: HP Blade System C-class servers with 300 AMD Opteron Dual Core CPUs

Benefits: Experienced reduced TCO, increased reliability, enhanced performance, freedom from vendor lock-in, and expects estimated cost savings of approximately €20 million Euros over the next five years



BACKGROUND

Swedish National Police Board (SNPB) is the law enforcement agency for the country of Sweden. The Police Board consists of the 21 police authorities in Sweden, in addition to the National Laboratory of Forensic Science. The Police Board employs 25,000 staff, including 16,900 policemen and 535 IT department workers.

BUSINESS CHALLENGE

The Police Board uses approximately 500 separate IT applications, 70 percent of which are developed in-house by the SNPB IT department. The applications are used to support the functions of the police force, covering a wide range of activities, including incident reporting, investigations, traffic surveillance, forensics, human resources, and accounting.

The IT department at SNPB has historically used a number of propriety solutions to provide the required services for the organization’s underlying IT infrastructure. Faced with the large costs associated with licensing, support, and maintenance for its proprietary solutions, the Police Board searched for ways to reduce acquisition and ongoing costs for its IT department.

The long-term goal of the Police Board is to replace all the proprietary software running on its servers with open source solutions. In addition to cost savings on licenses, support, and maintenance, the Police Board would benefit from the open standards, freedom of choice, increased competition between vendors, minimized vendor lock-in, reduced TCO, and the increased ROI that is associated with open source solutions.

SOLUTION

As it explored open source solution alternatives to its costly proprietary software, the SNPB IT department embarked on a pilot proof-of-concept project to test the viability of using open source solutions within its organization. Looking to replace its existing BEA WebLogic application server with an open source alternative, the SNPB saw only one realistic option, JBoss Enterprise Application Platform.

“JBoss Enterprise Application Platform was able to deliver the performance that was required and had the professional support necessary to enable us to confidently deploy the software in a production environment,” said Per-Ola Sjöswärd, Executive IT Strategist at the Swedish National Police Board. “The IT department required very little convincing to its suitability as it is well known in the industry as the leading open source application server. Additionally, training was not an issue, as many of the in-house developers had been using JBoss technology before. From a development perspective, JBoss fit well into our existing setup, as all of its in-house development is done in Java.”

The pilot scheme for SNPB’s open source architecture was the migration of the police photo database, named “PICTURE.” The PICTURE database fulfills three roles at the Police Board. First, it is used to store and search for passport photos – all citizens applying for a passport in Sweden must have their photo taken in a police station, which is then stored on the central database. Second, the PICTURE database is used to store and archive photos taken by police officers with digital cameras. Finally, the database is used as a common service for any other police application which requires access to photographs. As the system uses web services, the application can potentially be accessed by a large range of devices, such as mobile devices. In the near future, police officers will be able to use smartphones and PDAs to access the photo database to verify identification while they are out on duty.

The existing PICTURE system was built using entirely proprietary software and hardware, including Hewlett Packard PA-RISC chip-architecture, the HP-UX Unix operating system, Oracle database, and BEA WebLogic Server. The SNPB’s pilot was built by replacing all of these proprietary solutions with open source alternatives. The new architecture utilizes commodity x86-architecture, SUSE Linux Enterprise Server, MySQL, and the JBoss Enterprise Application Platform, which replaced the BEA WebLogic server. The JBoss Enterprise Application Platform performs a vital role in the new infrastructure, running the PICTURE application, handling user requests, and accessing the picture database.

System integrator Red Pill provided assistance for the project, with integration and training services. Additionally, the Police Board relies on JBoss for ongoing support and updates through a subscription to JBoss Operations Network.

BENEFITS

During its evaluation of the potential benefits of open source alternatives for the PICTURE system, the Police Board calculated the costs associated with the potential solutions for its IT system. Taking into consideration cost savings from volume discounts, the TCO of the proprietary solution for three years was estimated at €260,000. The team similarly calculated the cost of its proposed open source and commodity x86-architecture solution. At full list price, without allowances for volume discounts for the open source products, the TCO for the open source alternative over three years was valued at €70,300 – a massive 73 percent cost savings when compared to its current proprietary solution.

The SNPB has estimated that switching to open source solutions and commodity x86-architecture for all new IT systems will save approximately €20 million Euros in IT costs over the next five years. With its IT cost savings, the SNPB now has the opportunity to purchase 400 new police cars or hire 70 new system developers during the same five-year time frame.

In addition to costs savings, SNPB's new open source system has also delivered performance advantages over its former proprietary solution. The average load for the PICTURE system is 3,500 new passport images per day.

AUTOTRADER.CO.UK

Industry: Publishing

Geography: United Kingdom

Business challenge: To replace inefficient, incumbent systems with a reliable and affordable standardized system

Migration path: A number of different platforms, including Tomcat, to one standardized platform based on Red Hat Enterprise Linux and JBoss Enterprise Middleware

Hardware: x86 commodity hardware

Software: Red Hat Enterprise Linux, JBoss Enterprise Application Platform including JBoss Hibernate

Benefits: Gained increased knowledge management, expanded stability, and reduced operational costs

"We are more than happy with JBoss Enterprise Application Platform. Our 60-strong development team is happy with the implementation and the operation teams understand it, laying strong foundations for the future."

- Peter Hanlon, chief technical architect, Trader Media Group

BACKGROUND

AutoTrader.co.uk is the UK's number one motoring website, with over 10.3 million monthly unique users, who carry out over 106 million searches on new and used vehicles*. The brand is building a competitive position across all effective channels including online, mobile, and magazine. (*Source: Auto Trader Audit, January 2009).

AutoTrader.co.uk is owned by Trader Media Group, one of Europe's largest specialist multi-media groups and provider of market leading websites and its associated magazines, including Top Marques, Bike Trader, Truck & Plant Trader, Ad Trader, Motorhome and Caravan Trader, Farmers Trader – all with the largest reach in their sectors, through web visitors and magazine readership.

Trader Media Group operates in the UK, Ireland, Italy, and South Africa. The Group is looking to expand on its winning formula, giving trade customers the opportunity to choose all effective channels to market; building the service and relationship with dealers and manufacturers; and strengthening its portfolio of automotive and specialist classified titles.

BUSINESS CHALLENGE

Previously, Trader Media Group was using a variety of different platforms to support various applications across the AutoTrader website, posing a challenge to maintain in terms of cost, efficiency, and, in particular, knowledge management.

Trader Media Group recognized the need to replace these inefficient, incumbent systems with a reliable and affordable alternative. The objective was to standardize its system on one stack, streamlining the maintenance and support required for applications to run on AutoTrader.co.uk.

From a business point of view, there were significant costs associated with running multiple platforms. From an operational perspective, it also wasn't logical for the operations team to undergo a steep learning curve to understand various aspects of the platform in order to address specific issues that arose.

SOLUTION

Trader Media Group evaluated a number of open source and proprietary solutions that could offer the flexibility and stability that its critical platform required. The team felt that proprietary solutions didn't offer the full value at the cost-effective price that the organization desired. Trader Media Group needed a platform that could run Enterprise JavaBeans (EJB) and Hibernate on an enterprise level and at an affordable price. It was a natural choice to select JBoss Enterprise Application Platform, which integrates a clustered Java EE application server with next-generation application frameworks such as Hibernate.

The migration to standardize the AutoTrader website on JBoss Enterprise Application Platform began in 2006, and because AutoTrader.co.uk was a live site, the process was deliberately gradual. It took two years to complete the migration in total, with around 120 applications being migrated onto JBoss Enterprise Application Platform.

The decision to undertake a gradual, low-risk migration was based on cost, resources, and the need to maintain a competitive edge. The AutoTrader IT team had a strategy of migrating the different applications and elements of the site and schedule of when specific projects were to take place. The alternative, which would be to migrate the site in one go, would have been a resource-draining project for the whole team over a four-month period, stifling the continuous innovation and development of the site that helps maintain its position as the leader in the market.

The solution deployed included Red Hat Enterprise Linux as the operating platform and JBoss Enterprise Application Platform for messaging and web services, which includes JBoss Hibernate, a high-performance object/relational persistence and query service. The ecosystem also includes core search technology from Endeca.

The Trader Media Group team developed a virtualized image, which means that it can quickly deploy new servers when needed. The combination of Red Hat and JBoss solutions allows Trader Media Group to buy, deploy, and re-deploy additional servers at very competitive prices.

BENEFITS

Standardizing on JBoss has resolved the issues Trader Media Group had with running different platforms. The combined Red Hat Enterprise Linux and JBoss solution has given Trader Media Group a more resilient, high-performing, and stable platform and has ensured that any problems can be resolved quickly by the team.



Trader Media Group's key consideration is supportability. It is crucial that the operations team can understand and support the software. The fact that Red Hat offers a combination of technology spanning the operating system and middleware layers—one complete solution from one reliable vendor—offers value for its internal IT staff.

With Red Hat solutions, knowledge management has been simplified considerably, and because the new platform is open source, it has helped to build a strong community where staff can discuss issues and knowledge can be exchanged.

The resulting stability of the business-critical AutoTrader website is also recognized as an important benefit of the Red Hat solution. The availability, stability, and speed of the AutoTrader website is key to the company maintaining its leadership in the market. Users need to know that they can rely on a fast, accessible site that is available 24 hours per day, seven days per week.

Peter Hanlon, chief technical architect, Trader Media Group, said, "In Red Hat and JBoss we have acquired an integrated platform that includes all the core components required, such as messaging and web services, resulting in an enterprise-level infrastructure, out of the box. We are more than happy with JBoss Enterprise Application Platform running on Red Hat Enterprise Linux. Our 60-strong development team is happy with the implementation and the operation teams understand it, laying strong foundations for the future."

As a leading publication competing in a highly dynamic market, AutoTrader is constantly looking for new ways to both improve users' online experience and to carve out costs from its publishing model. With a solid open source architecture now in place, and a vibrant open source development community established within Trader Media Group's in-house development team, the future evolution of AutoTrader.co.uk will be built on the reliable Red Hat Enterprise Linux and JBoss Enterprise Application Platform.

6. SUMMARY

Every migration project, no matter the size or scope, requires detailed planning to ensure success. Understanding the risks, savings, and cost structure of a migration project is critical if you are to accurately project net improvements and realize actual return on your IT investment.

The considerations and processes detailed in this guide are designed to help you identify migration opportunities, examine the risks associated with various migration scenarios, create a standard build, and help develop a comprehensive strategic migration plan.

Prior to formal planning, an organization must acknowledge the motivations behind the migration, as well as understand the advantages and disadvantages to each potential migration scenario. Lacking this understanding, organizations may be unprepared for decisions and trade-offs that must be made throughout the planning process. Once motivations are clear, organizations should step through each of the five phases of the strategic migration process detailed in this guide. Those phases are:

1. Examine existing middleware architecture and determine the equivalent capabilities in the JBoss Enterprise Application Platform.
2. Examine third-party functional and business applications and determine compatibility or alternatives for the JBoss platform.
3. Measure organizational readiness and overall migration risk.
4. Develop a strategic migration plan, including a detailed road map and cost estimate.
5. Implement the strategic migration plan and employ implementation support strategies.

With this guide and additional Red Hat services, any organization will be armed with the necessary tools for planning and implementing a successful migration. And by combining the technology, training, and mentoring from one source, you will experience reduced development complexity and risk and see the value of your investment faster.

When you are ready to embark on your middleware migration, we encourage you to give us a call to discuss how Red Hat can help you make the right decisions from the start, reduce risk, and accelerate the impact of your deployed technology.



CONTACT RED HAT TO GET A PERSONALIZED QUOTE FOR THIS RED HAT SOLUTION

www.redhat.com/consulting
redhatconsulting@redhat.com

NORTH AMERICA
1-888-REDHAT1
www.redhat.com

LATIN AMERICA
+54 11 4341 6200
www.latam.redhat.com
info-latam@redhat.com

**EUROPE, MIDDLE EAST
AND AFRICA**
00800 7334 2835
www.europe.redhat.com
europe@redhat.com

Turkey: 00800 448 820 640
Israel: 1809 449 548
UAE: 80004449549

ASIA PACIFIC
+65 6490 4200
www.apac.redhat.com
apac@redhat.com

ASEAN: 800 448 1430
Australia and New Zealand:
1800 733 428
Greater China: 800 810 2100
India: +91 22 3987 8888
Japan: 0120 266 086
Korea: 080 708 0880

