

# Unlocking the Mainframe: Modernizing Legacy Systems to a Service-Oriented Architecture

*An Oracle White Paper*  
*June 2008*

# Unlocking the Mainframe: Modernizing Legacy Systems to a Service-Oriented Architecture

Executive Overview .....	3
Introduction .....	3
Drivers for Modernization .....	5
The Need for Agility and Adaptability .....	5
A Shrinking Workforce .....	5
High Cost of Ownership .....	6
Mainframe-Specific Concerns .....	6
Strategies for Modernization .....	7
The Benefits of Incremental Adoption .....	9
Incorporating Legacy Systems .....	10
Providing a Flexible, Consistent Architecture .....	11
Courses of Action .....	12
Conclusion .....	13

# Unlocking the Mainframe: Modernizing Legacy Systems to a Service-Oriented Architecture

---

## *Legacy system*

Wikipedia defines a legacy system as an existing computer system or application that continues to be used because the user does not want to replace or redesign it. Although this definition can apply to a number of different computer systems, it can also refer to a system that is out of date or difficult to replace due to the high cost or risk associated with doing so.

---

## *Service-oriented architecture (SOA)*

Wikipedia defines SOA as a software architectural concept where services support the requirements of software users. In an SOA environment, network nodes make resources available to other network participants as independent services, which participants can access in a standardized way through well-defined service interfaces.

## EXECUTIVE OVERVIEW

In recent years, the much-reported “democratization of IT”—as delivered by the PC platform and other midtier environments—has led to a plethora of choices for users; however, the mainframe has suffered as a result. With a dwindling ecosystem and a host of siloed applications, mainframe environments are becoming an increasing source of concern to organizations worried that one day the business logic trapped within them will be inaccessible to the enterprise at large. With the average Fortune 100 company maintaining approximately 35 million lines of legacy code, you can see what a sizable problem this presents.

Enter service-oriented architecture (SOA). Functioning as both a software development environment and a delivery framework, SOA provides a mechanism for defining business services and operating models—and in so doing provides a technological infrastructure that IT organizations can use to provide the agility and adaptability they need to meet enterprise business requirements and to modernize their legacy environments.

## INTRODUCTION

Traditionally, IT organizations have been highly centralized, similar to mainframe environments. Roles, responsibilities, skills, and budgets have all focused on discrete projects addressing specific business activities. As a result, vertically integrated, siloed applications were built and supported. This architecture, however, by its very nature represents a “closed” system—and one that results in isolated resources with limited flexibility. SOA offers a means of unlocking that data and making it available to the rest of the enterprise.

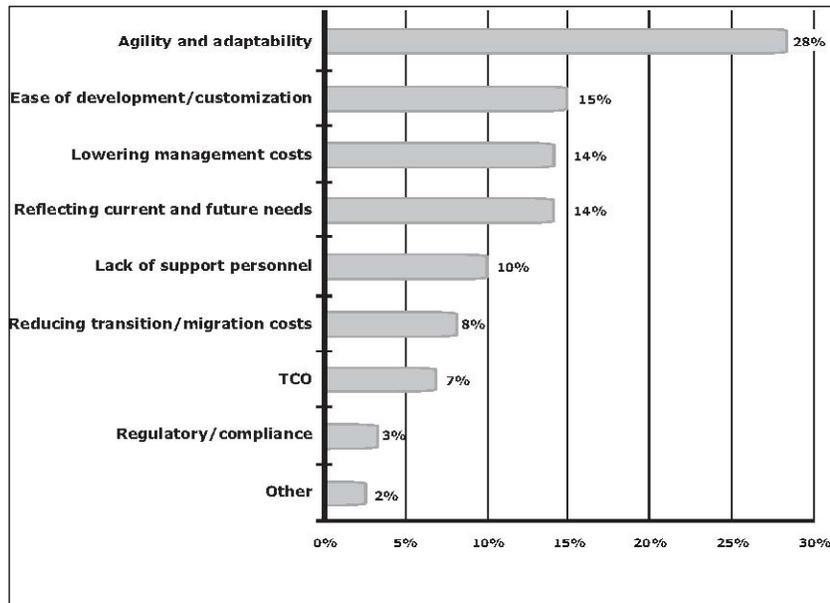
Successful execution of an SOA strategy, combined with the release of the business value contained in legacy systems, can bring enormous benefits to the enterprise by

- Improving productivity, agility, and speed for both business and IT; leveraging existing software assets to create new applications; and improving time to market
- Allowing IT to deliver services faster and align more closely with business, enabling business processes to be modified without rewriting code

- Providing interoperability with modern Web-based systems, thus extending the practical life of legacy routines and processes
- Lowering the overall IT cost structure by sharing business services across multiple applications
- Decreasing complexity via a reduction in redundancy across the enterprise

Although the benefits of this kind of modernization effort can be enormous, they do not come without risk. For starters, the business climate confronting modern enterprises is one of almost mind-numbing complexity. Constantly evolving business strategies and the resulting tactical implementation requirements, regulatory demands, and cost containment requirements must all be considered in developing an SOA initiative. What’s more, a multitude of heterogeneous business needs—coupled with an equally heterogeneous and complex IT environment—make a coherent analysis of business and technology requirements a must in designing a cost-effective technology platform that can accommodate future growth, according to Forrester Research.<sup>1</sup>

An environment of this complexity demands the right platform, a solid grasp of industry best practices, and the expertise to make it happen. Even more important, it requires a careful analysis and plan of execution before doing anything. Oracle can provide the platform, the best practices, and the expertise to achieve these objectives.



The results of a 2004 Meta Group survey clearly show that agility and adaptability top the list of business drivers prompting the modernization of legacy systems.

<sup>1</sup> Jost Hoppermann, “Large-Scale Banking Platform Renewal Case Study: Drivers,” Forrester Research (March 2005).

## DRIVERS FOR MODERNIZATION

Any drive to modernize a legacy application code base should have at its core four primary goals:

- To identify and manage the core business processes it contains
- To transform a typically monolithic application into an agile, adaptable environment that enables the organization to leverage business logic in new applications while maintaining consistency in business rules and correctness
- To separate business logic from technology, making it easy for applications to evolve to meet organizations' changing needs
- To promote the reuse of existing software assets

### The Need for Agility and Adaptability

Every enterprise wants to get the greatest business value possible out of its existing applications. However, with change a constant in today's business environment, agility and adaptability are an imperative—and they can be difficult to achieve in legacy mainframe environments, because most were developed in a time and place where change was the exception, not the rule.

But time marches on, and in today's environment business performance, quality, and return on investment need to improve, while costs, risks associated with change, and time to market must all be reduced. To achieve these goals, IT must improve its alignment with business by reducing complexity, optimizing asset usage, and helping the business achieve the agility and stability it so badly needs.

### A Shrinking Workforce

According to the book *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*,<sup>2</sup> the average Fortune 100 company maintains about 35 million lines of legacy code—a bit of a problem when you consider that many such legacy applications were written before many of the people currently using them were born, giving rise to the very real fear that one day the system will break, and there will be no one to fix it. And indeed, the aging population of programmers with mainframe skills is leading to a shrinking mainframe workforce, meaning that over the long term, the skills available for working with, updating, and fixing this platform will disappear.

When that happens, legacy systems will fulfill one of the definitions of legacy environments: they will resist being modified or cease to evolve to meet changing business needs. The opportunity to leverage this code portfolio will vanish forever.

---

<sup>2</sup> Robert C. Seacord, et al., *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*, (Addison-Wesley, 2003).

When IT organizations increase their agility and adaptability, they

- Simplify access to corporate data
  - Modernize their interfaces
- Expose business processes embedded within applications

## High Cost of Ownership

With IBM increasingly dominating the mainframe market and innovation by third parties diminishing due to shrinking margins, mainframes have become much more costly to own and maintain, leaving customers feeling trapped. If margins continue to dwindle for independent software vendors, innovation on the mainframe platform could grind to a halt completely.

## Mainframe-Specific Concerns

When considering the drivers for modernization, you must also take into account the factors inhibiting modernization. For mainframe environments, they can be significant, demonstrating how daunting a task migrating to a Windows or UNIX environment can be for large organizations with significant mainframe investments.

Table 1 shows the results of a survey Gartner Research conducted in which members of a mainframe-focused audience were asked what they believed to be the biggest inhibitors to growth in mainframe use within their organizations.

What is the largest inhibitor to the growth in usage of the mainframe in your organization?	2004	2003
Hardware costs	3%	0%
IBM software costs	19%	12%
Third-party software costs	29%	52%
Portfolio of third-party applications	10%	11%
Graying of the skill set / availability of trained mainframe personnel	8%	5%
Sole-source concerns	1%	1%
Perceived complexity of the mainframe	5%	5%
Management perception that the mainframe is outdated	25%	15%

**Table 1: The biggest inhibitors to growth of mainframe use, as identified by mainframe loyalists**

**Source:** Gartner Research, "Survey Results Reveal Perceived Role and Use of the IBM Mainframe," March 2, 2005.

In the same survey session, Gartner also asked respondents to provide a three-year outlook for their mainframe environments. The results of that survey are shown in Table 2, with the mainframe environment being divided into three general segments:

- Under 500 MIPS (million instructions per second)
- 500 to 1,000 MIPS
- More than 1,000 MIPS

<b>Which of the following most closely describes the outlook for your mainframe environment for the next three years?</b>	<b>2004</b>	<b>2003</b>	<b>2002</b>
We will completely eliminate our mainframe environment.	9%	5%	4%
The number of MIPS we have installed will decline from current levels.	9%	6%	14%
The number of MIPS we have will remain steady.	21%	15%	20%
We will grow our MIPS primarily due to new applications.	10%	10%	6%
We will grow our MIPS primarily due to the growth of legacy applications.	21%	20%	19%
Will grow our MIPS through the growth of both legacy and new applications.	29%	44%	37%

**Table 2: Note that 70 percent of those who responded to the survey had 500 MIPS or more installed—in other words, the mainframe loyalists.**

**Source:** Gartner Research, “Survey Results Reveal Perceived Role and Use of the IBM Mainframe,” March 2, 2005.

According to reports issued by Gartner in 2004 and 2005, it believes that mainframe environments at the low end of the scale (that is, those with 500 MIPS or fewer) are more likely to complete successful migrations to the Windows or UNIX platforms than their larger counterparts, which in general face significantly greater challenges to migration given the level of complexity involved.

It’s important, however, not to confuse modernization with migration, because migration is just one option for addressing the problem. It is entirely possible to modernize environments of 500 MIPS and beyond, and still leverage the assets of those environments in an SOA. Keep in mind, however, that one size will not fit all; thus, those seeking a silver bullet will be disappointed.

### **STRATEGIES FOR MODERNIZATION**

Any attempt to modernize mission-critical legacy applications by converting to an SOA must begin with a series of questions that seek to define the criteria for initiating such a project. The answers to these questions will serve as the foundation for the modernization effort. The following questions serve as a good starting point for any organization considering modernizing its legacy application infrastructure.<sup>3</sup>

---

<sup>3</sup> “Incremental SOA for Your Mainframe Environment,” Blue Titan (November 2005).

- **Is there a quantifiable business benefit?** Oftentimes, important organizational constituents fail to see SOA as anything more than “just IT architecture”—a problem because it’s difficult to demonstrate the business value of “architecture.” Thus, it’s important that early projects deliver tangible, measurable business benefits to all constituents. If this occurs, the momentum behind the SOA will increase exponentially. Ideally, initial SOA projects should be justifiable on cost savings alone.
- **Is there a high likelihood of future service sharing?** People become aware of the value of an SOA when they can leverage services created in earlier projects for subsequent projects. Careful planning in early projects should give rise to services that are likely to be shared by future projects.
- **Is the project low risk?** The more important a project is to the enterprise, the more risk it entails, regardless of its architecture. For this reason, early modernization projects should not be mission-critical ones—though they should still deliver tangible business benefits and be highly visible. Mission-critical applications can and should be built on the SOA—but not until after the organization has learned from early mistakes and risks have been reduced. A good example of an early project would be one that would deliver a major impact from a business perspective but would be relatively straightforward from a technology perspective.
- **Is the project high visibility?** Early projects should be both high visibility and low risk. Internally focused initiatives, such as portals and intranets, can be highly visible throughout the organization while carrying low business risk.
- **Is there a well-defined problem and clearly stated criteria for success?** By defining in advance both the problem and the criteria for success, an organization greatly increases the likelihood that its SOA project will succeed.

By answering these questions, organizations establish a basic foundation for their modernization efforts. To minimize risk and overcome organizational obstacles, the IT organization must now organize tasks incrementally so that it can win the enterprise over to the modernization effort one success at a time.

### **Overcoming Skepticism and Organizational Challenges**

Representing a unique confluence of forces, SOA promotes reuse in a context that makes sense for both business and IT (because both the architecture and the programming paradigm support reuse). In addition, it’s proved to be a powerful mechanism for defining business services and operating models. According to a Forrester Research report,<sup>4</sup> the result is that SOA gives IT a structure to deliver against actual business requirements and lets IT begin to actually adapt to the business on an ongoing basis.

---

<sup>4</sup> Phil Murphy, “Got Legacy? Four Fates Await Your Applications,” Forrester Research (January 2006).

Based on technologies that have been evolving for a number of years—including CORBA, Oracle Tuxedo, IBM MQSeries (now IBM WebSphere MQ), and other platforms—SOA incorporates an established set of best practices that can be adopted by mainstream, risk-averse enterprises. What’s more, SOA-enabling technology is becoming standardized, and thus cheaper, and packaged application vendors such as SAP are aggressively SOA enabling their products. In short, SOA is moving into the mainstream.

Whatever modernization strategy an enterprise adopts, it will face significant organizational challenges—and those challenges will increase in direct correlation to the size of the MIPS installation. For that reason, Oracle recommends that organizations keep the following principles in mind as they initiate their modernization efforts:

**Be forewarned: An enormous number of steps will be required to prepare for and navigate the process of converting to an SOA. Thus, organizations should be prepared for an incremental implementation.**

- **Adoption should be incremental.** The best way to fight internal resistance is to carry out modernization projects in very small steps. Even if all of your successes are small, they allow momentum to build. Slowly but surely such victories will demonstrate the value of the new approach. Proceeding in this manner also enables project teams to gain experience over time, empowering them to take on projects of increasing importance as they move forward.
- **Legacy systems should be incorporated into the new architecture.** Any modernization effort should maximize existing software, while transforming monolithic legacy IT environments into more-agile ones in which enterprises can leverage business logic and continue to support business rules and consistency.
- **A flexible, consistent architecture must be in place.** An SOA requires a unified, scalable, and extensive rock solid infrastructure for application development and integration. That architecture must also be flexible enough that project teams can operate with minimal design constraints, while still maintaining interoperability and ensuring that assets can be shared. A consistent architectural approach ensures ongoing evolution of the application environment and interoperability downstream.
- **A scalable governance model is required.** Without a solid governance model at its core, an SOA will not be successful.

### **The Benefits of Incremental Adoption**

When you consider the benefits of modernizing legacy application infrastructures, it’s not surprising that many organizations want to start reaping these rewards as soon as possible. However, it’s a mistake to attempt to dictate a global architectural change all at once; that almost certainly means ignoring or failing to understand the size and complexity of the project at hand. When a tectonic shift in an enterprise’s architecture is being considered, there are simply too many dependencies to consider for a global change strategy to be feasible.

Instead, a project-by-project approach is required—one that will require rigorous attention and management. Without such an approach, IT organizations risk “architectural creep”: what happens when teams start trying to get their own projects to market as quickly as possible and fail to adhere to the architectural imperatives integral to the long-term success of the overall SOA project.

To avoid this sort of architectural creep, a robust, scalable infrastructure is essential, and the correct tools must be in place to support the efforts of the project teams. Those tools must allow development efforts to be repeatable, and they must be flexible enough to allow the teams to do their jobs, while still providing sufficient rigor and discipline to support the overall objective.

Employing a widely used Java 2 Platform, Enterprise Edition application server and renowned for its easy set-up and clustering capabilities, the Oracle application infrastructure allows the architectural rigor required for this type of migration effort, according to *Software Development* magazine.<sup>5</sup> It is also standards compliant and features enterprise-grade security, deployment, and management services.

### **Incorporating Legacy Systems**

Today’s legacy systems—particularly mainframes—serve as home to critical business processes and contain significant business value. That value would increase exponentially if those processes could be shared across the enterprise.

Thus, if enterprises are to maximize the value of their SOAs, legacy systems must play a part. SOA can even enable incremental (service-by-service) replacement of legacy applications—a feat it accomplishes by isolating user-facing applications from the implementation of relevant services.

The legacy mainframe will play a key role in any migration. Because it’s familiar to the enterprise—and security issues are well documented and understood, and rigorous testing processes typically exist—the mainframe serves as the natural reference environment for the migration. (And if testing processes do not exist, they must be created. Once in place, they will isolate the tester from many of the complexities of the business processes associated with the application.)

Once the reference environment has been established, the next step in the modernization and migration project is pinpointing the principal project’s business goals. To do so, the organization needs to

- Identify the project’s scope and complexity
- Identify what needs to be migrated
- Determine the migration sequence

---

<sup>5</sup> Chris Minnick, “A Special Guide to Application Servers,” *Software Development* (August 2004).

- Provide “early warning” of requirements for specialist tools or contractors
- Provide ongoing application intelligence

### **Providing a Flexible, Consistent Architecture**

The internet represents a shared network of linked Web sites connected by a number of distributed routers. To be of most value to an enterprise, an SOA needs to be constructed in much the same way—that is, as a shared network of Web services linked by distributed intermediaries. In other words, according to the Blue Titan white paper “Incremental SOA for Your Mainframe Environment,”<sup>6</sup> it needs to be a *services network*. This means that a software infrastructure must be in place to enable reliable, manageable, secure, and high-performance communication between the consumers and providers of services.

Such an approach enables an enterprise to achieve its near-term project goals, while building toward an enterprisewide SOA. Thus, assets created during one project need to be easily reusable in new projects—which means that to be successful, an SOA implementation must adhere to flexible, coherent architectural principles that are consistent with an organization’s long-term objectives.

The need for both flexibility and consistency cannot be overemphasized. Project teams need to be able to complete their tasks with minimal constraints, while interoperability across the heterogeneous environments of modern enterprises must be maintained. What’s more, complying with standards and governance processes in an SOA will make consistency essential.

To achieve these goals, an SOA infrastructure must be diverse in all aspects: it should be able to reconcile a multitude of protocols, standards, and message formats. This is what makes an SOA a flexible, consistent, and practical approach. If an enterprise employs the appropriate infrastructure and tools (such as the Oracle application infrastructure) to support these initial projects, it will have already made significant inroads in creating a services network.

There is, however, no one-size-fits-all SOA solution. Thus, an organization must design its modernization strategy based on objective criteria, asking questions such as

- Is there a way to quantify which enterprises represent the most likely candidates for modernization with relatively lower risk?
- Is it possible to promulgate modernization strategies that will provide a reasonable assurance of success while minimizing risk for a broad range of MIPS installations?
- How can organizational challenges be overcome to achieve the maximum benefit of an SOA?

---

<sup>6</sup> “Incremental SOA for Your Mainframe Environment,” Blue Titan (November 2005).

## Courses of Action

When it comes to modernizing IT infrastructures, enterprises have four options:

- **Leave it alone.** No one can afford to change everything. That's why when funds are limited, it's essential to focus on the changes that will have the most positive impact. Yielding to inertia, in contrast, leaves enterprises unresponsive to rapidly changing business conditions.
- **Replace it.** Outsourcing, rewriting, or replacing legacy applications with packaged software represents a potentially expensive and high-risk task for even relatively small MIPS installations. The bottom line is that when this occurs, an organization is replacing a known quantity with an unknown new environment.
- **Retire it.** Although frequently overlooked, retiring legacy applications remains an option, albeit a risky one. After all, it takes time and money to terminate any application, because abrupt termination can spell disaster. On the other hand, an organization could be running an application simply because it's always run that application—consuming resources and generating reports that no one reads.
- **Modernize it.** Oracle believes that some form of modernization represents the optimum balance of risk and reward for the typical large enterprise. Although enterprises can achieve this goal in a variety of ways, this paper focuses on two identified by Forrester Research and referenced in the Blue Titan white paper cited earlier:
  - **Platform migration.** Also known as a native migration, this is the least invasive migration strategy. As such, it represents a good initial step for the highly risk averse. One good example of this type of migration is MetaWare Technologies' Phoenix solution, which allows client companies to move an application (typically COBOL) from a mainframe platform such as a customer information control system (CICS) to a Windows/Intel (Wintel), UNIX, or Linux platform. These mainframe applications can then be rehosted onto Oracle Tuxedo, which provides an equivalent set of CICS APIs. Oracle Tuxedo represents a standalone environment, which if used in conjunction with Oracle WebLogic Server, provides additional features and functionality to mainframe applications. In this way, it allows these applications to participate in SOA architectures and introduces the ability to build new composite applications. Given the necessity to retain and maintain application security and data integrity, organizations with greater than 500 MIPS installed are likely to favor this option. Anecdotal evidence reported by third-party vendors such as MetaWare and Acucorp suggests that the ideal target for this type of platform migration will be in the range of 2,000 to 3,000 MIPS.
  - **Language migration.** This significantly more invasive strategy poses the highest level of risk because it usually involves not one but three

migrations—that is, from the language, the platform, and the database. This solution, however, also offers the greatest potential benefits in modernizing to an SOA. Based on the Gartner statistics quoted earlier, this type of migration will likely represent a viable option for organizations with fewer than 500 MIPS installed. As SOA successes continue to accrue, however, this MIPS target will likely increase.

## **CONCLUSION**

For mainframe customers facing a narrowing window of opportunity to leverage their enormous investments in mainframe IT environments, SOA provides an opportunity for true synthesis—providing a solution that’s greater than the sum of its parts.

Despite the fact that organizations have been using related pioneering technologies such as CORBA, Oracle Tuxedo, and IBM MQSeries (now IBM WebSphere MQ) for more than a decade, the perceived newness of SOA has slowed such projects’ adoption. Once organizations begin to experience success with real-world SOA projects, however, adoption rates should accelerate significantly. Combine this with packaged software application vendors aggressively SOA enabling their applications, and SOA should soon be firmly established in the technology mainstream. And Oracle will be there with the platform, the best practices, and the expertise to ensure successful SOA initiatives.



Unlocking the Mainframe: Modernizing Legacy Systems to a Service-Oriented Architecture  
June 2008

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2008, Oracle and/or its affiliates. All rights reserved.  
This document is provided for information purposes only and the contents hereof are subject to change without notice.  
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. 0408