

# SOA Best Practices and Design Patterns

## Keys to Successful Service-Oriented Architecture Implementation

---

**John Fronckowiak**

**4/3/2008**

A white paper exploring best practices for SOA.

## Executive Summary

There is no question that the successful implementation of Service-Oriented Architecture (SOA) relies on a careful and holistic approach to business planning. One of the most important tools in the evaluation, purchase, and ongoing use of SOA is the body of best practices that vendors, consultants, and customers have compiled. The promise of business agility, improved customer service, and competitive advantage with SOA is real. What varies most is the time, cost, and ease of SOA implementation. By learning from the experiences of those organizations that have been through the process and looking at the standard best practices of large-scale technology implementations, success can come earlier and more dramatically.

### Brief Overview of SOA

SOAs provide modular services that can be easily integrated throughout an enterprise. They are flexible and adaptable to the current information technology (IT) infrastructure and investments. SOA implementations continue their emergence in business as a mechanism for integrating organizational operations in new and different ways and for promoting reuse while leveraging the existing value of legacy systems.

### Benefits of SOA

In any business, the bottom line is the essential test of any technology. SOA can provide a significant return on investment (ROI) by integrating legacy and mixed technologies and maximizing the value of existing investments while minimizing risk. Promoting reuse through SOA also helps reduce overall development costs.

If services and their data are generic enough, they can be accessed through a variety of interfaces. Decoupling services from their presentation reduces expenses and decreases the overall development time. Further, SOA makes IT consider the dynamic operations of an organization, not just a set of static requirements, thereby exposing information and data sharing across the organization and focusing development on the best ways to improve overall operations.

### Challenges of SOA

Although SOA brings significant business benefits, there are challenges to their implementation. As SOA services are typically coarse-grained and loosely coupled, their operations exhibit more latency than more tightly coupled implementations. This can be a challenge when implementing with real-time requirements. SOA is designed to bring together legacy systems in heterogeneous IT environments. Standardization of naming, definitions, and identification can present implementation challenges. However, these challenges can be resolved by the implementation of identity and naming services.

Finally, SOA is designed to cut through an organization horizontally and vertically, which presents many cultural, cooperation, ownership, and budget issues. Strong leadership must be in place, and executive support must be clear and evident in order for any SOA implementation to be a success .

## Introduction to Best Practices

Best practices suggest that there is an overall commitment to increase organizational efficiency. These practices must be considered from the specific context of your organization. Although the notion of best practices is constantly evolving, it's clear that the following areas are critical:

- Vision and leadership
- Strategy and roadmap
- Policies and security
- Governance and acquisition
- Operations and implementation

## Vision and Leadership

The implementation of Service-Oriented Architecture (SOA) requires vision and leadership from the organization's executives. This leadership is extremely important in the face of cultural resistance, because users will be moving from a current comfort zone and will need to think and cooperate with greater cross-functionality. A clear, consistent, and repeated message of the benefits of the SOA from executive leadership makes it clear that they are committed to the SOA implementation.

An evangelist should be in place to spread the word on SOA benefits and to listen and respond to users' criticisms. Education is the key. The evangelist should promote the values in information sharing across the organization and show how doing so provides the ability to answer the bigger business questions. The evangelist must be someone who understands the business, understands the technologies, and has the clear and strong support of executive management.

SOA introduces a model that cuts across traditional organization lines and can produce new and valuable information and services. The SOA implementation should look horizontally and vertically through the organization to find value in the currently available resources. Remember, SOA can help you leverage legacy systems and get them communicating in ways they never have previously. Certainly, there will be differences to overcome, and it's important to provide the appropriate level of support for all stakeholders.

Making the business case for SOA is important for gaining universal organizational adoption, which is important both for executive management and for support staff. The business case should include the objectives, a consideration of alternatives, a financial analysis, a project plan, and a review of the project risks and assumptions. The business case will help the organization make the strategic case for the SOA implementation and, when well publicized, help to socialize the plan within the organization. Each of the sub organizations should be assigned clear deliverables and well defined success-factors to ensure that there is clear ownership and well-defined metrics on the deliverables. Each of these sub organizations should also be rewarded for ensuring success of their deliverables as well as the overall success of the project undertakings. Finally, the plan also serves as a tracking document and an objective measurement of project success.

## **Strategy and Roadmap**

The strategy and roadmap document explains the goals and objectives of the SOA implementation. It should be considered an evolving document that will change as you move through the acquisitions and implementation processes. The document should address the architectural view of the design and also include risk, security, and management considerations.

The SOA strategy should be published early in the SOA adoption process. This strategy document spells out the intentions and thought process of leadership and is essential for socializing the process throughout the greater organization. The strategy should clearly identify expected outcomes of the project as well as the measurements that will be used to determine whether outcomes are met.

SOA implementation will not transform the entire organization overnight, so it's important to plan for an incremental deployment. Ideally, begin with the low-hanging fruit: Choose a handful of services that will make a difference and that people will notice. Success breeds success—and confidence.

Resources that were typically stuck in organizational silos in the past will now be exposed and shared across the organization. This new structure will require new ways of accounting for the cost of those resources that the executive leadership must support.

The organization is not a static entity, and the SOA strategy should reflect that. Plan and design for change. New opportunities and challenges will arise, and the plan should be ready to rise to them and be flexible enough to incorporate them.

Again, as the SOA implementation cuts across organizational boundaries, terminology and definitions will be different. Creating a common organizational vocabulary is important for providing a shared understanding of the services and information that will be provided. This vocabulary will also help facilitate further information sharing and collaboration across the organization.

## **Policies and Security**

When the decision is made to implement SOA, a clear set of policies and serious security considerations are in order. The policies must clearly identify the model and architectures that will be implemented across the organization. Security and risk mitigation must also be assessed, and these considerations must feed back into the overall policy decisions.

It's important to adopt a set of technical standards across the organization. Recognizing that standards are constantly changing and evolving, it's important to use industry partners to track developments. Flexibility with changes and compliance with as many emerging standards as possible are key to maximizing your SOA investment.

SOA brings together loosely coupled services to provide greater business value. Many times, these services may be in a state of constant change and flux. It's important to implement a services directory, which catalogs the available services.

A top-down, bottom-up approach is critical to security. Leadership must articulate the overall security policies and requirements, and users and system owners must actively implement and promote those policies.

Disaster-recovery planning is essential for any project. Plan for and implement a business continuance plan. Because SOA cuts across business domains, Service Level Agreements (SLAs) are important for defining expectations for the ongoing availability of services.

Security should not be an afterthought; rather, it should be baked into the solution from the beginning. Make security and authentication part of the planning process to mitigate risk exposures. It's much easier to include security in the solution from the beginning than to add it after implementation is complete.

Often times, it is important to drive the security, risk, and compliance policies via a central team. These policies should then be referenced to from all implementations of SOA in your enterprise. This is the recommended way to enforce security and compliance requirements across your organization, no matter how large or complex it may get.

## **Governance and Acquisitions**

Governance and acquisitions must work closely for a successful SOA implementation. Implementing SOA for the lowest cost possible—just to get it done—will not work in the long run. Both the governance and the acquisitions process must be flexible to meet the market's frequent changes. Organizations must participate with standards bodies and with user communities (both internally and externally) to ensure that the most appropriate choices are made. Governance must embrace an open-standards, loosely coupled, modular service approach.

Experiment with pilot projects to gain an understanding of SOA. These projects provide the experience necessary to tackle larger projects and expose the cross-domain challenges that will be encountered. The SOA implementation plan must be built around incremental adoption, acquisitions, and implementation. This incremental approach will ease the transformation process, but take care to ensure that the current purchases will meet and integrate with future demands. This is where adherence to open standards and participation in standards bodies (and with users) will be of most benefit.

Implementing SOA is a different way of looking at the overall business. Use enterprise modeling to identify and document the business infrastructure and processes. Defining the scope and boundaries of business process is important, and this is where enterprise modeling will help.

Finally, SOA will increase reliance on the network as the transport and communication method of services. It's critical to have facilities in place that will monitor, measure, and analyze the network. Use network SLAs to define expectations. In addition, ownership of both the SOA service directory and of services must be clearly identified to keep the directory up to date.

## Operations and Implementation

Operations and implementation is where all the leadership, planning, and design are actualized. SOA implementation works best with an agile development approach, in which services are built incrementally but in a rapid fashion. It's important to recognize—and be prepared for—times when a service being moved to your SOA may be met with resistance and skepticism, which may result in lower quality for the very short term. With adoption and acceptance come increased effectiveness, greatly increasing the overall service quality.

As with acquisitions, SOA cannot be put in place overnight. Focus on the services that are the easiest to implement but provide the greatest business value. Use them as building blocks. When one is complete, build on that service to implement the next. Momentum is key: When you get started, ride your success and acceptance to the next phase of implementation. It is important to recognize the early successes of not only the projects, but even the involved players. This recognition and visibility early on will both encourage the already successful players, but also inspire those waiting on the sidelines to join the party.

Partnering and collaboration—internally and externally—in the implementation phase helps to mitigate risk. Such interaction also has the added benefits of maximizing investment, easing cross-domain boundary issues, and leveraging shared experiences.

A key benefit of SOA is reuse of services. It's often tempting to build something from scratch instead of reusing what's already available. This can often happen for two reasons. First, developers may not be aware that a similar service already exists. Therefore, it's important to maintain a directory of available services that is readily accessible and uses the common vocabulary adopted across the organization. Second, when designing and implementing services, their use outside traditional boundaries must be considered. Creating coarse-grained, modular services helps to promote their reuse in the organization. Organizations should not cringe from updating services that have already been deployed when there are additional needs demanding extension of functionality. For example, eventually, all the requirements should be factored into a single 'get Customer ' service rather than having multiple services that get different subsets of customer information.

Finally, just get started! Theory and talk only go so far. Begin with something, anything! Keep in mind, though, that those pioneering services may become widely adopted over time, so don't neglect them. SOA promotes reuse of and interdependence on services, so even those early efforts must be cared for and managed.

## Brief Overview of Design Patterns

*Design patterns* are reusable solutions to common software design problems. Design patterns speed up the development process through the implementation of tried-and-tested solutions. They can play an important role in SOA implementation, especially in the standardization of service design. Since their introduction in the late 1980s, numerous patterns have been recognized and documented. Many SOA

implementations use Web services. It is important for architects of SOA implementations to have an understanding of the four primary design patterns for Web services:

- **Adapter:** Promotes the reuse of existing technologies through wrappers, extending your existing investments
- **Façade:** Used to reduce the coupling between the client and the server components—an essential technique for creating the appropriate level of granularity
- **Proxy:** Provides an object surrogate, used to simplify the interaction between Web services components
- **Controller:** A key component of the Model-View-Controller (MVC) architecture, used as an intermediary between the user interface (UI) and the data

## Adapter

As previously discussed, promoting the reuse of existing technologies is essential for a successful—and profitable—SOA implementation. The Adapter design pattern, shown in Figure 1, allows otherwise-incompatible classes to work together by converting the interface of an existing class into an interface that clients expect.

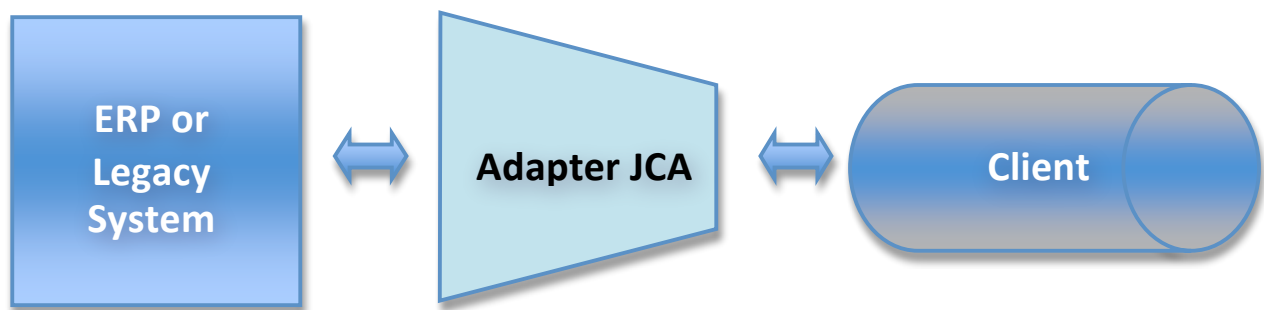


Figure 1. The Adapter design pattern

Organizations will look to reuse existing technologies in their SOA implementations; this is where the Adapter pattern is implemented. Typically, existing technologies provide interfaces that are incompatible with Web services. The Adapter pattern provides a bridge to the existing technology.

You don't have to start from scratch when designing Web services: The Adapter pattern can leverage your existing investment and quickly get you started on the road to service implementation. However, it's important to realize that not every application may be a useful service. It's important to be judicious in your design.

## Façade

Providing the appropriate level of granularity is essential to service design. Services that are too fine-grained can increase the overall network traffic as many service requests are made to perform an operation. More coarse-grained services can increase overall latency, but they help expose services that expose a business function.

The Façade pattern, shown in Figure 2, is often used to expose coarse-grained services. Instead of exposing the direct, one-to-one functionality of an existing software component or business function, the Façade pattern promotes encapsulation of these lower-level services to provide a single higher-level function.

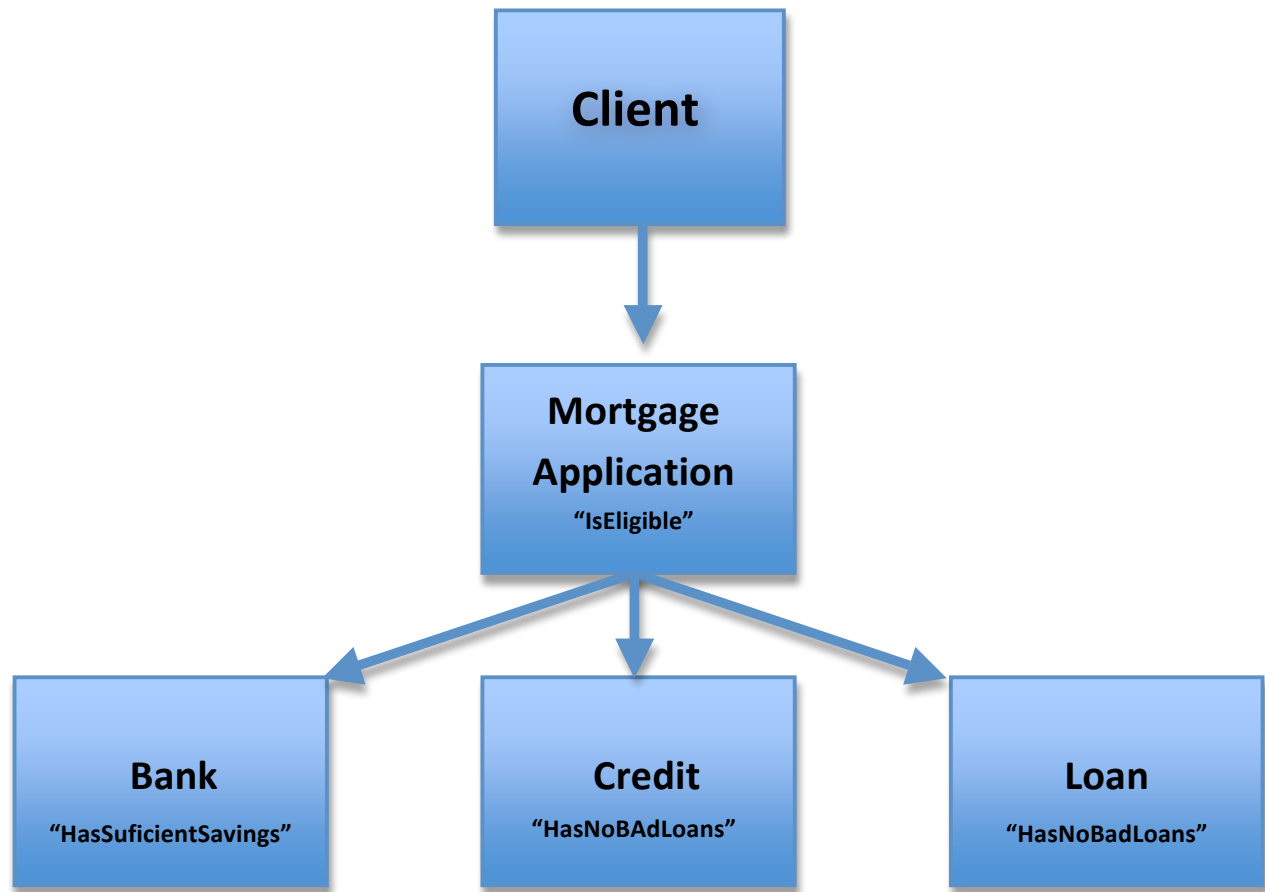


Figure 2. The Façade design pattern

The Façade pattern promotes consistent interfaces, abstracting clients from the implementation details of a service. Further, the pattern facilitates control and management of a service, providing a single entry point that simplifies elements such as security and transaction management.

### Proxy

The Proxy design pattern, shown in Figure 3, provides a surrogate or placeholder for another object. It can be used to simplify the interactions among services. The proxy can serve as a standardized interface for a collection of legacy back-end services. In other words, instead of providing a service for each individual back-end service, you can use the proxy to consolidate the messages into a single service, then dispatch the request to the appropriate back-end service, which simplifies the interaction with a collection of services.



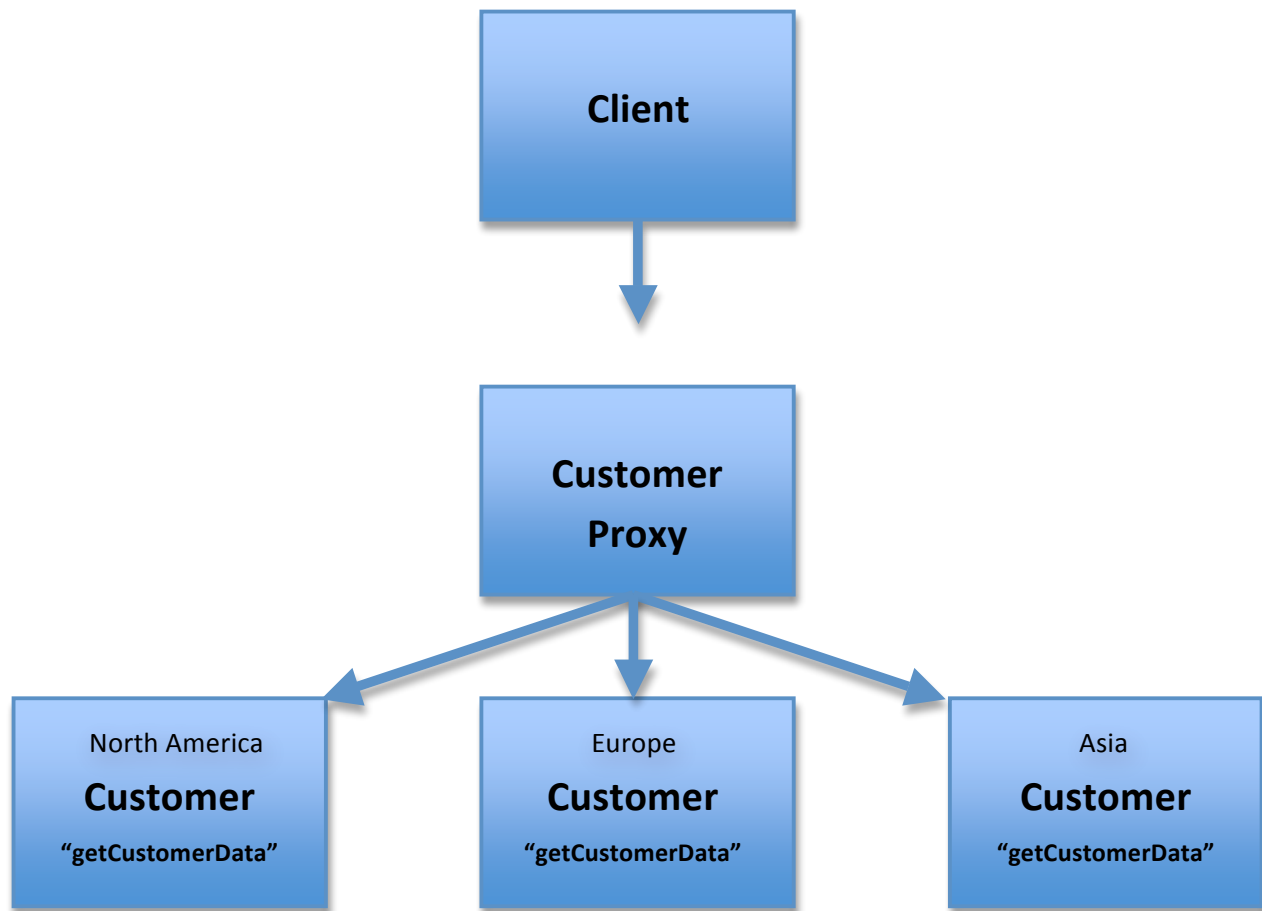


Figure 3. The Proxy design pattern

The Proxy design pattern can also be used for testing, especially when you are communicating with a third-party object that you do not control. The proxy would implement the same interface as the third-party service and can stand in its place during the testing process. Again, this simplifies the testing process.

### Controller

The Controller design pattern, shown in Figure 4, is probably best known from the MVC application architecture. In the MVC architecture, the *model* contains the data that the application requires; the *view* manages the user UIs; and the *controller* provides the logic and serves as the interface between the model and the view. The Controller design pattern is used to separate the presentation and data layers.

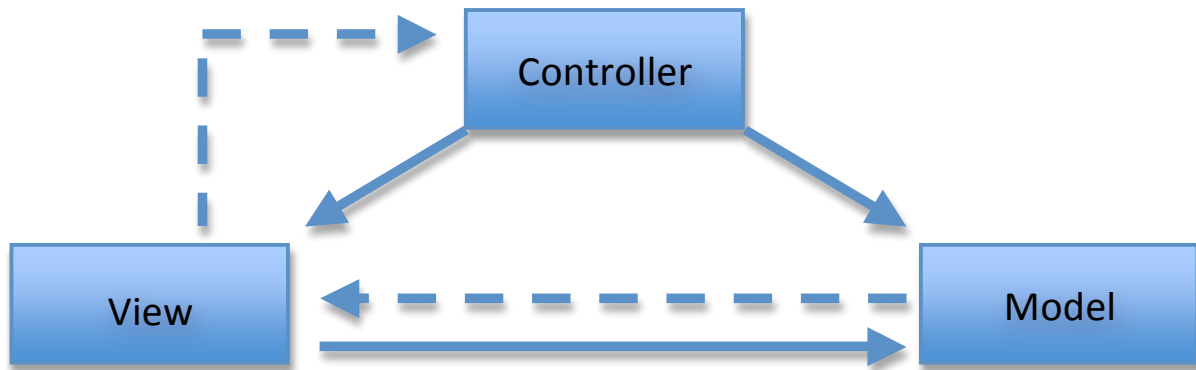


Figure 4. The Controller design pattern

The Controller design pattern can be used in SOA architectures to leverage existing application MVC design architectures and encapsulate the business logic of the service.

### Approaching Best Practices

SOA best practices are constantly evolving. However, efforts must be made in each of the areas discussed: vision and leadership, strategy and roadmap, policies and security, governance and acquisition, and operations and implementation. Having a champion who has a good understanding of SOA and can communicate that vision to all the stakeholders is essential to a successful implementation. Look for the low-hanging fruit as you begin your SOA implementation. Do something! Establish success with a project; learn from your mistakes as well as from your success. An incremental and agile approach will be essential.

### Getting Started with Design Patterns

Whether design patterns are a familiar tool or a new concept, an understanding of the four key design patterns—Adapter, Façade, Proxy, and Controller—will be essential to your SOA implementation. Begin with indentifying the services you need to implement, and then look to see how they fit into one of these patterns. Soon, you'll be able to easily recognize how a service can be implemented using these tried and true patterns, and that will give you an upper hand in the critical process of service standardization.

### Resources

This article is a starting point for understanding SOA best practices and design patterns. The following additional resources will help you gain a deeper understanding:

- Oracle's SOA Resource Center at <http://www.oracle.com/technologies/soa/center.html>