

An Oracle White Paper in Enterprise Architecture  
January 2010

# SOA Anti-Patterns: How *Not* to Do Service-Oriented Architecture

## Contents

Introduction .....	2
Anti-Pattern: From Application Silos to SOA Silos .....	3
Anti-Pattern: Build It and They Will Come.....	3
Anti-Pattern: Over-Engineer Reference Architectures.....	4
Anti-Pattern: Undefined Baseline for Business ROI .....	4
Anti-Pattern: Web Service Sprawl.....	5
Anti-Pattern: Armchair Architecture from the Ivory Tower .....	5
Anti-Pattern: Scattered SOA Strategy Entangled with IT Strategy .....	6
Anti-Pattern: Expecting a Free Ride on the SOA Train .....	6
Anti-Pattern: Where's the Money? or SOA Equals EAI 2.0.....	7
Summary .....	8
Conclusion .....	9

## Introduction

Early expectations for Service-Oriented Architecture (SOA) were high. But eager and well-intentioned early adopters faced unforeseen obstacles that inhibited the realization of SOA's purported benefits. It turns out that to deploy SOA successfully requires as much organizational as technical skill. This is especially apparent when attempting to implement SOA at an enterprise level. Many who have tried to architect an enterprise-class SOA solution have been surprised mid-project to find that they were not empowered to make enterprise-wide decisions.

As a means of increasing business and IT benefits, the latest SOA conversations urge architects to return to original SOA principles, such as genuine enterprise-class shared services. But to do this, SOA architects face significant organizational and political challenges as they attempt to implement SOA principles and practices beyond the confines of their departmental silos. Even the term *Enterprise Service* can trigger a turf war that can cripple a project before it can even get started.

Despite the inherent organizational and technical politics, SOA, after all, is an architecture, a set of principles and practices that can be applied at any level – within a department, or across the enterprise. Our conclusion is that SOA can deliver the greatest value at the enterprise level. That potential is reason enough to take steps to overcome the organizational hurdles to enterprise-level SOA. And, this is where SOA architects and IT management have a lot to learn from the emerging methodology and practices of Enterprise Architecture.

Enterprise Architecture and Service-Oriented Architecture are not mutually exclusive concepts. In fact, recognizing the abundant common ground between Enterprise Architects and SOA architects is an important first step in clearing any hurdles. Enterprise Architects bear the responsibility for optimizing and leveraging IT investments and processes across the enterprise. In doing so they must balance centralized governance for standards and consistency while maintaining enough autonomy to promote innovation and agility. Similarly, SOA architects must consider their own tough SOA-based enterprise policy decisions, weighing performance against flexibility, dependencies against layers of abstraction, and giving careful consideration to component granularity, among other concerns.

This paper is intended to provide insight into how to achieve next-generation architecture by applying SOA principles and practices at the enterprise level. In order to sharply define these practices and principles, this paper will focus on how *not* to do SOA by illustrating common anti-patterns that result from a narrow, siloed approach.

## Anti-Pattern: From Application Silos to SOA Silos

The good news/bad news is that silos are commonplace. Although silos enable focus and allow a quick claim to jumping on the SOA bandwagon, they also perpetuate redundant, unnecessary work; inconsistency (in business rules, integration, service granularity, and service contracts); overhead (from duplicative testing, documentation, training, and tuning); and risky lifecycle management.

SOA has value at many levels, but the lack of an enterprise or strategic view for services will limit SOA's full potential. For many organizations, application silos have been replaced by service silos, in which each application development team and/or each large object (LOB) develops its own set of services. The result is "service sprawl," a situation that prevents the realization of the economies associated with service reuse.

Silos defeat many core SOA architectural principles. For example, one such principle demands explicit contracts between service producers and consumers. It is difficult to have a contract among parties that do not communicate. So, while many have used SOA/Web services for loosely agreed-upon transaction connectivity, this practice misses the more strategic point about services that can be shared across the enterprise.

Technology alone cannot solve the SOA silo problem. In order to avoid the SOA silo trap and achieve service reuse, services, architectural principles, and other artifacts must be designed and shared through an enterprise-wide architectural communication process.

### Solution:

Embrace SOA by institutionalizing the elimination of uncooperative behavior that runs counter to enterprise SOA principles. To do this, establish a center of excellence or an architectural review/governance board that will ensure communication and interaction across teams.

## Anti-Pattern: Build It and They Will Come

Poor communication can bring a quick end to any attempt to build shared services. Imagine the reaction among team members when they learn of unannounced changes to consumer requirements, or that a new group is involved in the production of a service already under consideration. Similar reactions can result when centralized groups choose to alter service capabilities, such as security or functionality, without complete buy-in from their constituencies.

Such cases threaten the potential for service sharing among different departments, thrust project teams into unexpected situations, and, worst of all, damage trust among the groups involved.

### Solution:

The architectural review board must clearly define the architectural roles and communication responsibilities for the various layers of IT execution and relevant artifacts. For Enterprise SOA, publishing corporate and divisional IT practices, principles, and policies is a common and effective best practice.

## Anti-Pattern: Over-Engineer Reference Architectures

Playing with a new toy is fun. Playing with a lot of new toys is even more fun. Architects can sometimes focus so much on the “what” and “how” of various tools and technologies that they lose sight of the “why” and “when” of the specific business requirements. For example, when considering how an enterprise service bus interacts with a registry, it is just as critical to consider why the two tools are needed and when they should be used in the system deployment in order to maximize their value.

It is a wonderful résumé-building exercise to think through and idealize an elegant reference architecture. But architects should guard against the temptation of including something just because they can, because they have paid for it, or because someone in authority insists. The goal is not a service-oriented reference “anarchy-itecture!”

### **Solution:**

Architects should keep in mind that it is acceptable -- even desirable -- for reference architectures to evolve along with business requirements. An over-engineered reference architecture can disrupt focus, consume time and budget, and increase project risk. The more prudent and practical strategy is to adopt a just-enough, just-in-time architecture that reflects business requirements and available IT resources.

## Anti-Pattern: Undefined Baseline for Business ROI

Using new technologies and architectures has a nasty way of getting away from you. Early architectural decisions are commonly reversed, which can unleash a cascade of re-engineering. Before you know it, there you are with no time, no budget, and no usable work. With SOA, many have fallen into this trap. SOA’s many architectural nuances—from integration to application design, service definitions, performance, and security—will challenge effective planning until they are mastered. However, this challenge is more about project management than about SOA technologies. When the time comes to make a strong business case for additional budget and time, it is important to point back to as many material successes as you can. Unless each project milestone can establish a value-based result, you will have a difficult and complicated story to tell.

### **Solution:**

Carefully evaluate your SOA maturity level and your project management skills. Plan and budget projects with value-based milestones, and include the right contingencies—including asking for help from outside experts. Then formalize IT project management and accountability with your business units. A sharp focus on business milestones will help you avoid the draining exercise of repeatedly reselling SOA to the business and across IT.

## Anti-Pattern: Web Service Sprawl

The ability to create Web services has been greatly simplified and even commoditized, thanks to advancements from tool vendors and services available across the internet. Nowadays, creating a Web service interface and the supporting classes is just a matter of clicking a particular module and—voilà—you have a Web service.

The downside of this convenience is that developers get carried away and create Web interfaces just because they can, regardless of need. It is common to see registries, repositories, and directories filled with hundreds or even thousands of services that are never used. The ease with which services can be created only exacerbates the lifecycle burden and cost to document, test, manage, maintain, secure, audit, and continuously regenerate them.

### **Solution:**

Be deliberate about your Web services and remember that creating an enterprise service is different from exposing a Web service. Use an architectural review board, establish a service governance discipline, and deliberately decide on your services based on their service contracts.

## Anti-Pattern: Armchair Architecture from the Ivory Tower

Everything is possible on a whiteboard. In the pursuit of architectural excellence, creativity blooms, scope creeps, senior technical people are given the benefit of the doubt, and everyone gets their say. Whether it is the eagerness of the architects, the pressure of a completion date, or the lack of a lab environment where folks can get their hands dirty, those dry-erase scribbles are often adopted as mandatory patterns and standards without the benefit of empirical testing. To further complicate matters, architecture teams often broadcast their early decisions in order to allow others to begin work.

Adopting a less than rigorous architecture can produce an avalanche of negative consequences, from the need for simple reworking to a full-on catastrophic failure. To add rigor to your vetting process, consider research and experimentation tests for performance impacts, security breaches, deployment alternatives, network latency, server scalability, and auditability—to name a few.

### **Solution:**

Include an architectural vetting and communication process that targets specific audiences. For example, to publish early directions, communicate a roadmap of reference architectures, progressing from the less rigorous to those that are more demanding. Ensure that the right conclusions can be drawn from preliminary conversations in order to avoid suboptimal design, disjointed software development, and squandered resources.

## Anti-Pattern: Scattered SOA Strategy Entangled with IT Strategy

A static snapshot of the IT landscape reveals that it is teeming with projects and initiatives. One group might be exploring business process management (BPM) while another is experimenting with server virtualization and yet another is in the last stages of deploying new enterprise services—all while the new corporate IT strategy is getting its final touches.

So how is an emerging or strategic initiative such as SOA affected by projects in process, with their imminent deadlines and go-live schedules? While it is true that SOA evangelists have a corporate duty to get the word out about proper SOA education, planning, and execution, they must also guard against being dragged in at the last minute to apply a SOA quick fix or offer a full-scale rebuild. Under these circumstances the chances for effective SOA execution are low, and SOA may be marginalized, or worse.

### **Solution:**

Everyone involved should recognize the potential risks in introducing new technologies and reference architectures midway through a project. Nevertheless, when asked to do this, SOA evangelists and architects should demand absolute clarity about expectations and requirements. They must then be just as clear about the specifics and limitations of the proposed reference architecture, and about the benefits that can be achieved given available time and skills. For example, a quick fix solution may preclude reusable enterprise services. The reality is that there is no steady state in modern IT, and that green field projects are a luxury. When doing SOA right is not an option, establishing realistic expectations is essential.

## Anti-Pattern: Expecting a Free Ride on the SOA Train

SOA's wide popularity is a given. It is safe to say that nearly every global IT organization understands SOA's value and has started down the SOA path. But organization-specific SOA principles, processes, and standards are also widely under-resourced. IT groups that are loath to invest in adequate training, in architecting foundational components, or in writing a strategic plan to drive widespread adoption are common. As a result, companies take shortsighted shortcuts, roll out severely stunted SOA, or end up missing the boat completely.

### **Solution:**

Companies should evaluate their true goals for SOA. If the goals are to realize the full benefits of Enterprise SOA, it will be necessary to invest in architectural and process leadership. Frameworks are available to aid in advancing an organization's SOA and EA maturity levels. Adopt SOA deliberately, and be prepared to invest the time and effort needed to see the results across pragmatic time horizons.

## Anti-Pattern: Where's the Money? or SOA Equals EAI 2.0

Many companies think SOA is defined as “Web services managed with an enterprise service bus.” If that definition is accurate, then SOA has done nothing more than provide a new way to do what we already could with Enterprise Application Integration (EAI). What hype are we perpetuating by calling this SOA? If it is hype we are after, maybe it would be trendier to call it EAI 2.0.

Of course, there is nothing wrong with using SOA technologies that manage JBOWS (“just a bunch of Web services”) and an enterprise service bus (ESB) as an EAI tool. In fact, doing so may allow those tools to operate with better performance and connectivity options, and simpler business rules management. However, as described throughout this paper, SOA offers much more than transactional integration.

### **Solution:**

It is important to clearly understand the benefits you seek, and to apply the appropriate technologies. While some may exercise liberties in referring to certain technology components as SOA, accuracy demands crystal-clear explanations and expectations about which strategies and technologies drive which benefits. Don't get caught having to explain missing enterprise SOA benefits by providing EAI 2.0 as an example.

## Summary

The following list summarizes how to defeat the anti-patterns discussed in this paper:

### *Application Silos to SOA Silos*

- ❖ Doing SOA *right* is not just about technology. It also requires optimal cross-team communications.

### *Build It and They Will Come*

- ❖ Establish a SOA architecture and governance review board for policy setting, alignment, education, communication, and even building code.

### *Over-Engineer Reference Architectures*

- ❖ It's OK to build an evolutionary just-enough, just-in-time reference architecture.

### *Undefined Baseline for Business ROI*

- ❖ Embrace formal project management, and tie project milestones to business objectives.

### *Web Service Sprawl*

- ❖ Create services only where and when they are needed. Target areas of greatest ROI, and avoid the service sprawl headache.

### *Armchair Architecture from the Ivory Tower*

- ❖ Always provide usable reference architectures. Be clear about their capabilities and limitations and how they should be used. Validate your architecture rigorously.

### *Scattered SOA Strategy Entangled with IT Strategy*

- ❖ Get and set expectations with absolute clarity when adding SOA to a project midstream.

### *Expecting a Free Ride on the SOA Train*

- ❖ To achieve enterprise-class SOA, embrace a maturity framework, build a shareable enterprise foundation, and invest in enterprise-wide training and adoption.

### *Where's the Money? or SOA Equals EAI 2.0*

- ❖ Set expectations clearly about which SOA strategies and technologies drive which benefits.
- ❖ Effective SOA realization requires an appreciation that SOA is more than an ESB deployment, and that it is not simply a matter of using Web services for interfacing between systems.



## Conclusion

While there is little doubt that SOA technology has been widely implemented, it is apparent that architects and IT have not realized its full potential. The examples presented in this paper show that the successful implementation of SOA principles and practices requires technical and organizational skills in equal measure, and to add an enterprise architecture perspective to a SOA project. The enterprise architecture perspective, by definition, aligns with enterprise business initiatives and consequently provides the foundation for IT to establish an enterprise-wide IT discipline. Establishing an empowered enterprise architecture methodology, process, and governance board will facilitate achieving that discipline. This is where SOA benefits from enterprise architecture.

As a global technology leader, Oracle has had a decade of experience across myriad industries, helping customers realize the business and IT benefits of disciplined IT approaches. Oracle offers many resources, including expertise, architectural and organizational processes, and a portfolio of tools and reference architectures, to help companies of all sizes along the SOA maturity curve.

We hope you will embrace the opportunity to reinvigorate your SOA initiatives with true enterprise-wide initiatives in order to achieve the architectural and organizational excellence (enterprise shared services and service contracts; abstracted processes, data, and business objects) that defines SOA. In addition, the Enterprise Architecture and SOA practitioner communities offer a wealth of knowledge, lessons learned, and best practices. Seeking help from these communities in your SOA efforts allows you to turn someone else's 20/20 hindsight into your insight.



*An Oracle White Paper  
in Enterprise Architecture:*

SOA Anti-Patterns:  
How *Not* to Do Service-Oriented Architecture

January 2010

Author: Peter Heller

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, 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.

0109