

# The SOA Magazine

## Feature Article



### SOA Engineering Focal Points

by Ted Barbusinski

Published: June 16, 2008 (SOA Magazine Issue XIX: June 2008)

[Download this article as a PDF document.](#)

*Abstract: This article, the second in a two-part series, examines why strategic SOA investments so often suffer from poor technical implementation and diminished ROI. In the first article [REF-1], we established common SOA engineering misconceptions that are often the root cause of poor SOA technical implementations. Within this second part, we will examine key SOA engineering focal points and success factors that are essential to successful SOA solution implementations.*

*As with the first article, the goal here is to empower IT leadership, management, and enterprise architects with the knowledge necessary to apply high-level guidance to SOA engineering teams and to position the SOA investment for successful technical implementation and maximum ROI.*

*(Note: The information presented here reflects insights and knowledge derived directly from engineering teams designing and implementing large-scale SOA solutions within global IT environments. [REF-1])*

### Introduction

SOA engineering perceptions are often shaped by vendor product and marketing literature and industry pundits reacting to new industry developments. The driving motivations behind much of this information are market forces and competitive struggles between market leaders. By their nature, these forums of discussion offer little focus on vendor-neutral, SOA engineering fundamentals and success factors that are essential to successful implementations. These engineering focal points are the topic of this article.

Since we could fill volumes with SOA engineering issues and guidelines, we will scope the following discussion to a brief look at the primary focal points, along with some insights relative to these concerns. Each point is introduced and defined in terms of real-world engineering concerns, and includes a set of action items.

### Focal Point #1:

#### "Delineate Service Infrastructure as a Distinct Subset of the Service Architecture"

The term "service infrastructure" is somewhat subjective since different engineering teams might define its boundaries differently. Yet, it is critical to delineate this clearly because "service infrastructure" must address a separate and distinct problem domain with respect to the overall service architecture.

Since vendor product stacks often blur the lines between architecture and infrastructure and often weakly address this distinct problem domain, it falls to the SOA architect to clearly identify these boundaries and ensure the associated problem domain is properly addressed.

We have found it helpful to associate service infrastructure with those elements of service architecture that directly facilitate service producer and consumer interaction while ensuring the following:

- scalable, high performance messaging
- scalable, high performance XML and cryptographic processing
- scalable, high performance, adaptive message and service endpoint security
- efficient, high-performance execution of composite services
- efficient "service mediation" services (virtualization, routing, transformation, SLA, etc)
- high service availability
- client abstraction and independence from SOA design and interaction complexity
- consistency in cross-platform, client-service interaction
- consistency in service exception behavior
- efficient hooks for governance, auditing, metrics monitoring.

Notice that virtually everything within this problem domain is focused upon or heavily impacts the client-side design-time and real-time service interaction experience. If service clients have problems with consuming services, the value of an SOA investment can decline dramatically. Delineating service infrastructure with a focus on service interaction helps ensure the associated problem domain is properly addressed.

#### *Action Items*

- Look at the overall proposed SOA solution architecture and delineate the subset of that architecture that comprises the service infrastructure.
- Clearly define the boundaries of responsibility for SOA infrastructure relative to the overall SOA architecture.
- Define in some detail how the proposed infrastructure design addresses the technical challenges in its problem domain.

#### **Focal Point #2:**

#### **"Keep Your Service Infrastructure Lean and Mean"**

As your SOA design unfolds, it pays to never forget that everything you place inline between a service and its consumers introduces some overhead and subsequently some performance degradation. Successful SOA implementations never lose sight of this fact and strike an intelligent and prudent balance between the need for "service mediation" capabilities and the constant requirement to ensure a quality service client experience.

#### *Action Items*

Examine your overall proposed SOA infrastructure and ask the following questions:

- How many policies must be parsed and executed between a service client and endpoint (e.g. ESB service façade policy, governance monitoring policy, service endpoint policy, etc)?
- How many authentication cycles occur between a client and a service endpoint (e.g. ESB authentication, LDAP authentication, endpoint authentication, etc)?
- How many LDAP and/or IAM interactions must take place between a client and service endpoint?
- How many layers of software must a client request pass through before it reaches the service endpoint?
- How many cryptographic cycles occur between a service client and a service endpoint (e.g. "first mile" security, "last mile" security, etc)?
- How would composite services (services calling services) or composite apps (apps calling multiple services) perform through all the layers between a service client and a service endpoint?

Questions like these can help shed some light on potential infrastructure-bloat and inefficiencies that lead to poor service consumer experiences.

**Focal Point #3:  
"Establish Core Service Producer Foundations"**

By nature, services will be developed and deployed on a wide variety of IT platforms based on the whims of developers and the priorities and interests of various IT fiefdoms. This natural scattering and proliferation of service hosting platforms throughout an IT enterprise is detrimental to good SOA engineering. There are significant architectural efficiencies to be gained from consolidating service development and deployment within a core set of integrated, scalable service hosting platforms supporting a wide range of services.

These efficiencies can be realized in the following areas:

- security architecture
- governance infrastructure
- composite service infrastructure
- information source and B2B integration infrastructure
- trust infrastructure
- administration and operations infrastructure
- network and datacenter infrastructure
- development skills and environments

*Action Items*

- Define core service hosting foundations for process, data, integration / utility and presentation services. Catalog where services are currently hosted.
- Examine whether it's practical to consolidate them onto the established core service hosting foundations.
- Define a service migration strategy to consolidate services on core hosting platforms where practical.
- Establish governance policies that ensure new services are developed and hosted on core platforms whenever possible.

**Focal Point #4:  
"Adopt a Holistic and Strategic Perspective of SOA Engineering"**

No aspect of a service architecture should ever be designed in isolation. A mature SOA must support a strategic vision that spans the IT enterprise and eventually interacts with most key IT assets. As such, a service-oriented solution is much more than the sum of its parts. Each solution element must be designed with a deep understanding of its interactions, integrations, performance impact, security impact, operational impact, and its role in supporting the overall strategic SOA vision. Additionally, SOA design decisions should always be prioritized favoring broad, long-term strategic goals over near-term, narrow concerns.

*Action Items*

- Define and document an overall SOA vision that articulates the strategic business goals the SOA engineering solution must support and enable.
- Balance this strategic vision against engineering constraints and realities and ensure it is the guiding light of the overall SOA solution architecture.
- Validate every SOA solution element or subsystem against this strategic vision.
- Establish policies ensuring SOA expansion occurs in support of strategic goals rather than near-term, isolated pressures.

### **Focal Point #5:**

#### **"Always Design for Operational Scalability"**

This is an area of SOA engineering that's frequently overlooked, yet it can have a huge impact on long-term ROI. Operational scalability is the ability of a service-oriented solution architecture to establish and maintain highly efficient and adaptive, cost effective day-to-day operations as the solution grows and scales with time. It also represents the ability of the architecture itself to be efficiently re-factored to accommodate change and dynamic business requirements.

For example, let's assume business drivers call for a security change requiring that two fields within a service response document be encrypted. Let's further assume the service is secured via WS-Security and has five active clients on five different platforms. This security change requires that all five platforms execute a WS-Security and XML Encryption-compliant decryption cycle on the two response document fields. In the absence of good operational design, this simple change might require the following:

- Five different system administrators on five different platforms must be notified.
- Five administrators must use five different platform security tools to comply with the new security change.
- Possible disruption of production software on five different platforms must be resolved.
- WS-Security compatibility issues on multiple client platforms may need to be resolved.
- IT must invest in five different system administrators skilled in the nuances of WS-Security and WS-Policy.

There is little of anything efficient, cost effective or adaptive about this scenario. Operational efficiency and scalability is a key SOA engineering success factor that must be designed into the overall, end-to-end solution from the very start.

#### *Action Items*

- Envision a scaled SOA architecture / infrastructure (2 to 3 year maturity model).
- Apply business use cases to this model that demand a high degree of functional and operational agility.
- Look at all details necessary to respond with agility to these use cases.
- Identify areas of concern relative to SOA operational efficiency and address them within the proposed architecture.

### **Focal Point #6:**

#### **"Design for the Service Interaction Problem Domain"**

In our first article, we introduced the concept of the "Service Interaction Problem Domain," which encompasses all issues associated with heterogeneous client platforms, applications and development environments interacting with services.

These include the following:

- SOA performance engineering
- user identity propagation
- user credential availability and propagation
- SAML incompatibilities
- Kerberos integration
- single sign-on
- platform trust relationships
- client-side WS-Policy enforcement
- WS-Security incompatibilities

- client service proxy limitations
- client security complexity abstraction
- federated identity support and overhead.
- identity and authorization management (IAM) integration and overhead.
- portal client integration

Although these are some of the most difficult challenges in SOA engineering, they do not enjoy a high degree of support from the SOA vendor community. Attaining ROI begins with making services accessible, consistent, simple and responsive for heterogeneous client platforms and development environments. Successful SOA engineering requires accommodating these architectural challenges within all phases of service delivery.

#### *Action Items*

- Examine all the heterogeneous service client platforms in the IT landscape.
- Build an understanding of the unique service consumer challenges faced by each potential service client platform.
- Take into account any IAM solution that may be in place and the impact it might have on service interaction.
- Design your service infrastructure to provide common, consistent service interaction models across heterogeneous client platforms while accommodating specific platform limitations and the design concerns defined above.

#### **Focal Point #7:**

#### **"Define and Implement an End-to-End SOA Security Strategy"**

Perhaps nothing has the potential to make or break the success of a service-oriented solution quite like its security architecture. Although the intricacies of a security architecture engineering could fill volumes, there are a handful of considerations worth highlighting:

#### *Understand the Scope of the Problem*

SOA vendor security products sometimes leave the impression that SOA security is mostly a service-side problem. It's important to acknowledge that strategic SOA security engineering begins at the IT perimeter where single sign-on frameworks convert user IDs and credentials into platform-specific tokens (thereby impacting the ability of service client applications on those platforms to authenticate to a service security framework). Security concerns extend to portal frameworks where identity and credential mapping can impact service client authentication and further extend to IAM platforms that may govern user security contexts, federated identities, authentication tokens and more.

They span client-service proxy objects and their ability and limitations in conforming to service policy requirements and also span through network infrastructure and service endpoints to the back-end systems they interact with, and so on.

#### *Understand the Performance Challenge*

It's important to understand that SOA security architecture elements inevitably degrade service performance. The only question is how much? It's crucial to design every facet of a security architecture from the ground up, with performance in mind.

#### *Design for Operational Efficiency and Scalability*

Never lose sight of the fact that service security is both a client and service-side issue. A security architecture must therefore efficiently address the security issues of both service producers and consumers and support efficient, agile re-factoring of security behavior for both clients and services as requirements change in support of business dynamics.

#### *Action Items*

Build and document an understanding of how and where the SOA security architecture integrates with and / or complements the following:

- IT perimeter single sign-on solutions
- identity and authorization management (IAM) infrastructure
- service client platforms
- portal infrastructure
- network infrastructure
- back-end information systems

Develop and document an end-to-end security strategy that spans these IT assets, accommodates their unique concerns, and aligns with strategic IT and SOA business goals. Furthermore, design SOA security subsystems in support of this overall strategy.

### **Focal Point #8:**

#### **"Invest in Expertise and Experience"**

Nothing will impact the success of your SOA investment more than the people chosen to architect and implement the solution. Some of the areas of expertise necessary to drive a successful service-oriented solution architecture can be summarized as follows:

- Enterprise Architecture (EA) – SOA solution engineering is a challenge that is scoped at the enterprise architecture level, as service-oriented solutions integrate, interact with, and support most key platforms within an IT enterprise. SOA architects require deep expertise and experience in EA solution engineering to understand and address the myriad EA-level challenges intrinsic to service-oriented architecture and infrastructure.
- SOA Engineering – Expertise and experience in large-scale, secure, high-performance, high availability service-oriented solution engineering requires a deep knowledge of common SOA challenges, risks, and pitfalls, as well as proven SOA design patterns.
- Security Engineering – This represents expertise in security engineering principals and experience in SOA security architecture engineering in high threat and high security environments.
- SOA Strategic Transition Experience – When transitioning a large IT enterprise to a services-based value delivery model over time, it is helpful to have the experience of those who have already been through this lifecycle.
- SOA Vendor Market Knowledge – A solid working knowledge of SOA market vendor product offerings is very useful to assess the capabilities, limitations, and strengths of any products being considered.
- Distributed Application and Service Development Experience – Having someone with a background in developing diverse, large-scale, distributed Web-based business solutions and services on various IT platforms is also very helpful.

Larger SOA initiatives typically blend transitional external talent with in-house skill-sets to ensure formal knowledge transfer over time. In selecting personnel to drive service-oriented solution architecture, it is important to remember that an SOA initiative is a significant strategic investment that will likely impact the ability of an IT organization to deliver value to the business for years to come.

#### *Action Items*

Balance the natural desire to cut costs on personnel with the realization that SOA engineering is a challenging strategic investment with long-term consequences.

### **Focal Point #9:**

#### **"Plan for Governance Infrastructure Integration Early On"**

SOA Governance is more than a set of policies, procedures and authorities. Governance monitoring and enforcement requires architectural support and often this support is treated as an afterthought (which is a mistake). The integration of governance infrastructure should be part of fundamental SOA engineering planning efforts at their earliest stages.

Governance infrastructure has design-time and runtime components. Poor engineering on the design time component will impact service reusability, security, deployment, and operational efficiency, whereas poor runtime engineering impacts performance and the value of governance metrics. All of these issues end up impacting the attainable ROI of an SOA initiative.

### *Action Items*

Governance infrastructure has a close relationship with SOA security architectures and often integrates with SOA security elements. Consider viewing governance infrastructure as an integral part of your SOA security architecture and design it into your foundational SOA security strategy.

### **Conclusion**

It goes without saying that SOA engineering, as a subject matter, cannot be adequately addressed by a set articles or whitepapers. It is also self evident that SOA management must rely heavily on the expertise and experience of SOA architects to yield a successful technical implementation. However, the insights offered in these discussions can provide a valuable starting point for gauging the directions, and ultimately the technical success, of any SOA engineering solution.

### **References**

[REF-1] Vektrel Solution Template Library and Knowledge Base, SOA Architectural Domain

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL



[Home](#) [Past Issues](#) [Contributors](#) [What is SOA?](#) [SOA Glossary](#) [SOA School](#) [SOA Books](#) [About](#) [Legal](#)

Copyright © 2006-2008  
SOA Systems Inc.