SOALEADERS

Putting the SOA Infrastructure Together: Lessons from SOA Leaders

Maximizing the Value and Success of SOA

An SOA Leaders Council Whitepaper

Contents

Introduction

Today, enterprises need to be more agile than ever. Narrow windows of market opportunity and increased global competition mandate a faster response to business challenges and the ability to quickly evolve to meet new requirements. It's vital to their success that organizations leverage existing investments in applications and infrastructure, while also supporting new opportunities to interact with customers, partners, and suppliers. An XML-based Service Oriented Architecture (SOA) enables enterprises to easily create high level services that encapsulate business processes and functions, and add new services or upgrade existing services quickly and efficiently to address new business requirements.

The success of SOA depends upon the broad interoperability and re-use of services. SOA principles, XML and Web services standards enable the enterprise to deploy best-of-breed infrastructure solutions that are integrated easily with each other – coupling the best problem solvers into an integrated and efficient enterprise infrastructure – a backbone for the SOA.

Unlike proprietary, single-vendor solutions that lock enterprises into an infrastructure that lacks flexibility moving forward, employing best-of-breed components of a SOA infrastructure has several benefits. In addition to enabling the evolution of SOA as business requirements demand, a best-of-breed infrastructure reduces risk and allows organizations to take advantage of technology innovation from industry leaders focused on XML-based SOA and Web services. With best-of-breed technologies, the enterprise is ensured that its vendors develop the most scalable and useful implementations of infrastructure – avoiding lock-in, lackluster features and performance, and delays. Often a single-source vendor must make resource tradeoffs between iterations of the different distinct technologies – and these trade-offs lead to release delays that can be avoided with multiple vendors applying their resources in parallel. The key to working with multiple vendors is that the vendors demonstrate a commitment to integration supported through native and standard interfaces with existing and evolving infrastructures.

This whitepaper describes a practical approach to implementing an enterprise SOA. The paper is informed by face-to-face meetings between over 500 SOA Leaders – Enterprise Architects, IT managers and others that have faced the challenge and realized the benefits of creating a SOA. The goal of this paper is to clarify the benefits of each major category of SOA Infrastructure and provide enterprises with a framework to determine what infrastructure is needed at each point to achieve short- and long-term SOA goals.

Infrastructure Requirements for SOA

The reality is that enterprise IT infrastructure is already heterogeneous across operating systems, applications, system software, and application infrastructure. A successful approach to SOA should embrace the heterogeneous nature of IT and must wrap and reuse existing IT assets, not remove and replace them. Thus a standards-based approach to SOA is critical, ensuring interoperability of all SOA components and freeing IT to pursue a best-of-breed approach to vendor selection. This approach maximizes the functionality of each infrastructure system without compromising integration or introducing excess latency.

It also isn't practical to assume that an enterprise SOA will be created whole, overnight. The reality is that SOA can be implemented incrementally and still show value. Indeed, given current budgetary pressures and the need to demonstrate business value, an incremental approach is preferred. However, this requires careful consideration of the required SOA infrastructure components necessary to ensure scalability, performance, governance and management as the SOA expands.

Design, Creation and Testing of Services vs. Run-time Service Operations

SOA infrastructure components fall into two related domains: *design, creation and testing* (development) and *run-time* (production or run-time). *Design, creation and testing* includes processes and technologies critical to building, discovering and deploying web services within an SOA. *Run-time* includes processes, components and technologies critical to ensuring reliable operations, performance, and security.

Design, Creation and Testing components may include:

- Web services containers
- Integrated Service Development environments (ISE)
- Discovery Service
- Integrated Development Environments (IDE)
- Legacy Integration
- Governance & Service Lifecycle Management
- Service Test Tools
- Orchestration Servers
- Presentation Service

Run-time components may include:

- XML Gateways - XML Message Mediators for security, integration and acceleration
- Web Services Management (WSM)
- Message Queues such as ESBs
- Business Activity Monitors

Universal Infrastructure Requirements

There are basic universal requirements for a SOA infrastructure. These include:

- **Interoperability with Standards/ Support -** SOA is not a new idea, but it has gained new life due to the emergence of standards-based XML Web services. XML-based Web services have gained broad support in the vendor community and among SOA Leaders, and provide easy application and data interoperability.

- **Availability with Flexibility -** Process and services should be isolated from hardware and communication failures and allow for the modification of processes, relationships between applications, rules, data mappings, etc. without excessive effort or downtime. Interactions should be possible with services located both within the enterprise and externally.

- **Heterogeneity that Scales-** Service oriented architecture consisting of existing legacy applications with services interfaces, new services and composite services must ensure system-wide performance and accommodate on-demand changes.

- **Secure Visibility, Control and Governance -** Auditing, managing and monitoring of processes and services deployed while ensuring trusted interactions and message integrity among a highly distributed SOA. Infrastructure must ensure consistent policy enforcement and lifecycle management.

Incremental SOA Adoption Demands Scalable Infrastructure

Enterprises should take an incremental approach to SOA, pursuing a strategy that leverages best-of-breed solutions that lay a foundation for the organization's evolving business needs.

Expose Initial XML-based Web Services Interfaces

Fundamentally, XML-based Web services increase the utility of existing applications by making them more accessible, re-usable and cost-effective to integrate. In order to take advantage of the benefits of SOA, enterprises need to service-enable applications, either by updating to a Web service-enabled version, adding a service front-end from an ESB, or virtualizing through a Gateway. This is often an efficient mechanism to demonstrate the value of XML-based Web services and build support, momentum and budget for deeper SOA initiatives.

- Gateways, Web services management and ESBs can insulate developers from the complexity that can arise when creating and operating services while also creating Web services "front ends" to existing systems.

- Registry provides knowledge of the services an enterprise has, what they do and conditions of use and reuse. It also supports governance and service lifecycle management

- Testing ensures new service interfaces perform as expected

Service Oriented Business Development

It's vital for enterprises to align IT services with business processes, and in order to do that organizations must look at key business processes and align IT services to support them. The components of a best-of-breed SOA infrastructure provide the foundation as organizations begin to re-implement services.

- Registry ensures consistent governance and enables service reuse

- Web services management delivers visibility across messages, proactively ensures operational health and services for reporting

- Gateways optimize end to end performance and provide message level security, while mediating between heterogeneous message formats, identity credentials, standards, versions and different transport protocols between service delivery platforms, partners and applications

- ESBs mediate between message addresses (location), data encoding formats, specific communication protocols, and provide reliable guaranteed delivery semantics for messages and business processes collaborating on a shared bus.

- Testing ensures that new service-based applications are secure and perform properly

Enterprise SOA

Once services are in place, enterprises looking to reap the full benefits of SOA should pursue innovation and new combinations from those services. A variety of SOA infrastructure components enable organizations to leverage the investment in XML-based Web services.

- Registry provides much-needed real-time information about service behavior

- Web services management provides dynamic action based on service attributes

- Gateways enable virtualization, security and mediation across busses, platforms, systems and organizations

- ESBs deliver reliable messaging and business process orchestration

- Testing solutions test new services and identify potential downstream and upstream dependencies

SOA Design, Creation and Testing Infrastructure

Web Service Containers, IDE, and ISE

Web Services Containers

Web Services Containers support the creation, integration and deployment of services through publication of WSDLs for Java, .NET, C++ and other applications and systems**.** These products and technologies are considered mandatory, "must-haves" for a web-services implementation and form a critical foundation of the standards-based SOA infrastructure. Typically, "Web service-enabling" existing applications and systems is a first step toward creating a SOA.

Integrated Development Environment (IDE)

Development of web services imposes additional referential needs upon the developer. An Integrated Development Environment (IDE) enables the drag and drop development of web services, providing the developer with a variety of services to choose from for greater productivity. A good IDE also encourages better reuse and can eventually lead to composable web services.

Most packaged applications and application servers now ship with Web services APIs, and there are many options for creating Web services. Key requirements for a Web services container include:

- *Rigorous Standards Support*
  At a minimum, the Web services container should support SOAP 1.2 & 1.2, WSDL 1.1 and related XML standards.

- *Exceptional Interoperability*
  Look for full support for WS-Interoperability Basic Profile, which ensures basic interoperability between different Web service containers. But also check for interoperability test results at collaborative organizations like SOAPBuilders.

- *Security & Reliability*
  Most business process require some level of security. Look for support for WS-Security, as well as support for authentication and authorization.

- *Ease of Deployment, Configuration and Integration*
  Make sure the solution has tools that support your existing developers IDEs, such as Eclipse, to make it easy to create, test and configure services, then publish them to a registry

Integrated Services Environments (ISEs)

Once services are defined and stood up on an application server, packaged application or bus, integration architects and developers need the ability to assemble business processes based on a series of Web service interactions and business logic. This development activity requires new process modeling tools (diagrammatic) as well as tools for building partner collaborations (a variant of business process for interactions with parties external to an enterprise). For these tasks, and for more routine tasks such as building XSLT based XML-to-XML transformation maps, developers will need an Integrated Services Environment or ISE. ISEs are different from IDEs in that they are not code development environments, but rather, services and process development environments. ISE will also need to be integrated with registries and repositories to publish and control these SOA artifacts.

Orchestration

Orchestration technology supports the SOA vision of composable web-services based applications. By composing new functionality from an abundance of existing web services, these tools enable the creation of new mission functionality and new applications and web services that are based on services and data from existing applications. Orchestration technology brings together many components into a web services system that can be comprised of other web services, databases, legacy applications and ESB-based processes.

Registry & Repository

Discovery

The ROI for SOA is reuse: The ability to discover data and services is paramount.

Although discovery services may be optional for small implementations, they are mandatory for large scale deployments, and a vital component of a SOA foundation layer. Universal Description, Discovery and Integration (UDDI)-based registries are in common use today to support "service discovery" and house important information and metadata that describe characteristics of the service.

Governance

The creation of web services that comply with organization policies and directives is essential to the successful implementation of interoperable web services. Governance lays out the policies that should be adhered to when building web services. Governance products assist developers by ensuring developed services adhere to the organizational policies. The SOA Backbone should include technology that makes searching, reviewing and updating governance policies intuitive and widely available.

Lifecycle Management

Enterprises need to certify and approve services for business and IT standards and deployment readiness. SOA participants need to be able to share, find and use services. Services need to scale for availability, reliability, integrity and overall quality. These requirements create the need for a new SOA business services lifecycle that guides the control and the quality of an enterprise's SOA.

A registry supports the business services lifecycle by enablement and publishing of SOA information like functional service descriptions and policies and advanced functionality for security, scalability, and reliability. For enablement and publishing, it provides mappings of SOA and Web services resources and publishing. In design and runtime discovery, it provides a secure services information browser, change notification, and UDDI V3 standard discovery and data access. For management, it provides replication, mapping of Quality of Service (QoS) management information, and advanced business service classification management.

SOA Test Tools

Test tools need to be implemented to inject realistic usage scenarios onto the infrastructure and insure a successful deployment. Services impose specialized testing challenges for the SOA, and Service Test Tools rapidly test web services for functionality, performance, and scalability. In addition, test tools can also validate functionality, boundary conditions, performance, scalability, and security for smoother, more responsive deployments. Because reuse, service access and service availability are fundamental to achieving a robust service oriented architecture, automated regression testing is a necessary process in order to ensure secure, reliable, complaint services.

The testing process and enabling technology should be capable of detecting errors early in the development process and assisting development to prevent these errors from happening. This requires a detailed and specific understanding of the system in question, as well as the flexibility to address both known usage scenarios as well as unusual or unknown usage scenarios. Most tested systems fail because the system or a component of the system was exercised in a manner that it was not designed (e.g. passing unexpected values). Untested or lightly tested code that gets executed in a way that the developers never intended is usually the primary culprit for a security vulnerability. Furthermore, in an environment which organizations are exposing legacy systems via web services, there is a huge potential for a component to be exercised in a manner in which it was not originally designed. Automated test tools deliver a comprehensive and quick framework to eliminate potential errors and vulnerabilities from the code base.

The process of enforcing a security policy, however, can be tedious. Manual code reviews for security can be costly and lengthy, often requiring repeat visits from third-party experts to ensure that previous vulnerabilities were patched and that no new vulnerabilities have been opened. A best-of-breed AEP (Automated Error Prevention) solution allows enterprises to create and enforce a security policy that is customized and extensible, and adds a transparent layer of security testing and verification throughout the development process.

SOA Run-time Infrastructure

The introduction and architecture of run-time infrastructure is crucial to short-term results and long-term success of SOA. The issues range from mediating heterogeneity (there is considerable room for alternatives within standards and interoperable implementations), reliable message deliver, system-wide performance monitoring and system-wide security. While the first web services exposed can have a basic run-time infrastructure – perhaps provided by the web services container itself – once there are multiple services, connections and the start of a SOA, enterprises find they need robust run-time infrastructures to maximize the performance, security and reliability of their SOA.

In fact, many enterprises actually need layered run-time infrastructures – XML Gateways, management nodes, and messaging systems. It is vital to have expert development of any of these run-time systems as their reliability impacts the performance of the entire SOA.

XML Gateways -
XML Message Processing Horsepower

At its core, XML Gateways deliver deep, application-centric, XML message processing based on a set of declarative policies. As a result, XML Gateways create a reliable, secure, optimized and controlled environment in the network for loosely coupled systems. XML Web services require unparalleled throughput in order to support complex integrated application systems. The result of a robust, best-of-breed solution is the most secure, efficient, and flexible message processing performance, end-to-end.

Security

Security is a critical run-time component of SOA implementations, ensuring information isn't compromised by unauthorized individuals. Security considerations from threats to message integrity to appropriate access controls existing between services and messages within an application server and those remotely accessed. From the very first service to a robust SOA, ensuring security while maintaining utility and availability is crucial, as is the ability to ensure system performance while proving that the system protects private and confidential data.

Integration

XML-based Web Service implementations potentially expose business logic programmers to the variety of platforms and supporting infrastructures deployed within an organization. XML Gateways route messages between heterogeneous systems based on message contents, qualify of service or routing tables.

A SOA infrastructure must address the issues of integrating XML-based Web Services with support systems where the information in these systems is required for policy or control decisions. Traditional application development environments hide many issues related to integration of supporting infrastructures, such as attribute repositories or Identity and Access Management. SOA requires XML Gateways to abstract those complexities away from service development.

Acceleration and Performance Optimization

XML Gateways significantly accelerate XML-based SOAs by improving performance of the entire system. XML messages frequently require expensive operations such as XML parsing, XSL transformations, authentication, schema validation, signature validation, encryption, and message or attachment compression. These functions consume significant amounts of system resources, potentially overburdening systems and degrading performance. Offloading these functions to Gateway appliances optimizes performance and minimizes impact on system resources.

SOA Management

The implementation of a loosely coupled web services environment brings system management and monitoring requirements that are beyond the scope of conventional ESM and NSM solutions. Web Services Management (WSM) enables enterprises to control Web services in real-time, while monitoring how well the Web services are operating and whether they are meeting performance commitments. Ensuring the operational health of the system by setting and maintaining appropriate service levels, detecting and handling exceptional conditions, and addressing heightened security requirements are the primary capabilities provided by SOA management. Additional management capabilities include online upgrades and redirects, routing, logging, monitoring and reporting.

Two types of management that are of paramount concern to organizations looking to gain business value from service-oriented systems include:

- Service Level Management: Concerns related to performance, security and availability of the system.
- Exception Management: Concerns related to errors and exceptions that must be recorded and resolved, preferably automatically.

Service Level Management

The management system monitors the complete SOA system, including external application components—such as a partner's Web service that is consumed by the system—and actively manages it to ensure acceptable performance and availability. By managing the system according to service level agreements, or SLAs—formal guidelines for performance levels—the management system enables an organization to identify and mitigate performance issues before they can impact business. Detailed service level data enables the IT team to fine-tune the system for better performance.

Exception Management

The number and variety of application components, combined with the evolutionary nature of distributed applications lead to errors and unexpected conditions that mandate a new brand of exception-handling capabilities. SOA management can solve this by catching all types of exceptions midstream —from simple data entry errors to complex business conditions—and resolving them in real-time, before they result in lost revenues.

Message Queues and Enterprise Service Busses (ESB)

The infrastructure often requires a messaging system to handle the increased flow of traffic generated on a SOA infrastructure. An ESB guarantees delivery of messages and helps to mitigate the effects of resource failure within applications collaborating on a bus. ESBs are used when the number of services and their interaction grows and/or when guaranteed delivery of messages to a service or within an orchestration process is of utmost importance. ESBs also provide for broadcast style communications, allowing multiple services to be notified in parallel of certain events.

It isn't necessary to re-host applications to enable them to interact through an ESB. In many cases, this would be impractical, because the applications are configured and deployed within their own managed environments, such as application servers. In addition to Java™ 2 Platform, Enterprise Edition (J2EE) and Microsoft .NET applications, ESBs routinely connect packaged applications and data sources such as relational databases.

The ESB represents each service using a common interface model, regardless of the nature of the resource on which a service is based. The interaction model—how a service is invoked—is event driven and invoked through messages (typically in XML or binary wrapped with XML metadata).

Best-of-breed ESBs leverage messaging technology, building on queuing and publish/subscribe mechanisms, to reliably and securely route messages between services within the ESB. Beyond basic messaging patterns, ESBs deliver process- and content-based routing and orchestration are introduced as mediation services.

Continuous Availability of ESB

At the heart of an ESB is the messaging server which actually delivers the messages between nodes in a distributed architecture. These servers form the basis of the SOA infrastructure and provide the store and forward semantics needed to deliver different quality of service– best effort delivery, reliable delivery and guaranteed delivery. To insure the backbone is never off-line, best-of-breed ESBs and Enterprise Messaging systems need to provide some mechanism to insure continuous availability – such that no message or transaction is lost – regardless of the type of failure and that the ESB is always responsive to service requests.

Business Activity Monitoring

As enterprises invest in SOA, their goal is to implement business processes. That's why it is imperative that the processes are visible to management and that business decisions can be taken to rapidly change processes that are not performing as anticipated. This class of work can be supported through simple Web dashboards if the state of the business processes and partner collaborations are accessible in real-time. Best of breed orchestration servers and partner collaboration servers provide this basic information. Combined with an ESB as the transport for broadcasting business events (changes in the process or collaboration state), these orchestration and collaboration servers enable firms to monitor their business activity.

Integrating Into an Infrastructure System

Loose coupling differentiates SOA - inside the service, no information is assumed as to the purpose, technical nature or business nature of the service consumer. The infrastructure backbone supporting SOA is no different. Each piece of the infrastructure is operated by different people, crossing roles and responsibilities that were once isolated. Selecting a best-of-breed solution built upon common communication standards enables organizations to mediate between the disparate roles and responsibilities while improving collaboration.

SOA combines services into solutions by essentially becoming a collection of services integrated using various technologies. Because a SOA architecture creates one unit of work from many moving parts, it's important for enterprises to build the architecture as a best-of-breed platform that is interoperability centric.

Design and run-time infrastructures naturally touch each other and provide seamless information flow between development and production. Registry, Repository, ESB, Orchestration and Web Services Containers also have a role in acceleration and performance optimization. Registry and Repository store run-time statistics about services, while Web Services Containers execute the business logic and Orchestration servers inform the routing of messages. ESBs deliver multi-casting and reliable messaging and can maintain a parsed version of a message throughout a business process execution. Similarly, Web Services Management and XML Gateways deliver benefits in development through powerful debugging interfaces, simplified service development and testing.

Examples:

- *Reactivity and Parasoft* – Reactivity enables secure, fast and highly interoperable XML message transport. Parasoft generates wide arrays of messages and scenarios to test the policies being enforced through Reactivity, and also the security and reliability of the back-end services themselves.

- *Systinet and Sonic* – Systinet's Registry provides a home for Sonic to find and publish Web services produced and consumed by processes executing within the ESB. Through UDDI interoperability the Sonic Workbench allows process developers to find services and drag&drop them into their process and B2B collaboration models.

- *Reactivity, AmberPoint and Systinet* – Systinet stores all the information about services, and that information is updated by AmberPoint. AmberPoint gets information about services and messages from AmberPoint agents and Reactivity Gateways – ensuring complete end-to-end understanding of the operation and performance of a service. AmberPoint can use this runtime data to proactively manage the system to ensure performance and availability.

- *Sonic and Parasoft* – Enterprises select Parasoft's best-of-breed SOA Test to ensure that their service oriented architecture (SOA) and enterprise service bus (ESB) processes are secure, compliant and reliable. Parasoft SOA Test bridges the gap between development and QA by promoting a workflow that shares the same testing assets throughout the development environment. This workflow increases the speed at which SOA and ESBs are deployed and significantly reduces the re-work loop between development and QA.

- *Sonic and AmberPoint* – Sonic is working with AmberPoint to allow AmberPoint agents to track and monitor any ESB endpoint as if it were a SOAP interaction. This will allow ESB administrators to turn on tracking for service interactions between Sonic ESB and over 200 legacy and packaged application systems, all the leading relational databases, Sonic's business process execution engines and its B2B collaboration servers. Through this collaboration AmberPoint can provide enterprises with a pan-SOA view of all interactions – Web services and otherwise.

Putting it All Together - The Best-of-breed Benefit

Selecting best-of-breed SOA solutions creates competitive advantage for enterprises. Organizations that choose best-of-breed components of a SOA infrastructure leverage focused development efforts, superior performance and infrastructure flexibility not available with a single-vendor solution. Loosely coupling the enabling infrastructure supporting SOA enables replacement or enhancement of a component without disrupting the entire infrastructure and system.

Web Services Management - AmberPoint

AmberPoint provides the management layer that is critical for realizing the return on investment from distributed, heterogeneous service-oriented systems. AmberPoint's solutions provide comprehensive management and security capabilities without requiring any changes to the services themselves.

Customers such as British Telecommunications, H&R Block, Motorola and the U.S. Department of Defense have chosen AmberPoint for its comprehensive capabilities, its non-invasive approach and its native support for Java and .NET.

Web Services Testing - Parasoft

Parasoft is the world's leading provider of Automated Error Prevention software solutions. The privately held company develops and markets proprietary development tools and solutions to help companies eliminate and prevent software errors. Parasoft's award winning AEP products and services help customers accelerate time to market, dramatically reduce development expenses, enhance total software quality, and increase end-user satisfaction.

XML Gateways – Reactivity

Reactivity is the leading provider of the core infrastructure for networks to secure, manage and optimize XML traffic. Reactivity appliances maximize security and speed of XML Web services while reducing the costs of deployment and ongoing operations. Reactivity delivered the market's first integrated security, integration and acceleration XML appliances and continues to deliver the most application fluent infrastructure for XML based connections and Service Oriented Architectures (SOAs). Today, Reactivity Gateways are the production foundation of the most valuable XML services in the world. Reactivity continues to lead the market with core functionality for federated identity integration and enablement and the only multi-mode processing for optimized network performance on any XML message of any size, with any policy and across any transport.

ESB and Orchestration - Sonic

Sonic Software is the inventor and leading provider of the enterprise service bus (ESB). Sonic enterprise integration and messaging products deliver flexibility, scalability and continuous availability to SOA environments through a number of patent-pending innovations, including Dynamic Routing™ and Continuous Availability Architecture™.

Registry, Governance & Service Lifecycle Management - Systinet

Systinet provides the foundation for SOA governance and lifecycle management, making IT simpler, faster and standards-based. With its suite of award-winning and proven products, Systinet enables organizations to rapidly leverage and reuse their existing applications and data assets, provide interoperability among heterogeneous systems, and better align business processes with IT. Systinet's products enable, publish, discover and manage SOA business services, and make it easy to build secure and reliable Web services with Java and C++ applications.

SOA QuickStart – Architecture Assessment and Roadmap - ThoughtWorks

ThoughtWorks helps global 1000 organizations realize meaning value through business agility. The SOA QuickStart helps organizations evolve their existing enterprise architecture to be more a scaleable, flexible, and agile enabling. ThoughtWorks approach focuses on identifying an organization's highest priority business imperatives and then collaboratively develops a practical roadmap that balances architecture and other IT investments with the value produced. Additionally, ThoughtWorks extensive experience delivering enterprise transforming solutions ensures our clients avoid the all too common 'architects dream, developers nightmare' scenario found in many architecture initiatives.

About the SOA Leaders Council

The SOA Leaders Council is the largest peer-to-peer community of SOA visionaries and implementers. It comprises IT innovators bound by their common interest in the practical realities of SOA implementations for the global enterprise. Community members are CTOs, VPs of Architecture, Security Architects and Directors of Application Development who are willing to share their experiences and expertise for the benefit of the broader community.

Every SOA Leader is invited to regional chapter meetings and participation in online forums based upon their current role leading SOA adoption within their enterprise and their insights on requirements, best practices/processes and infrastructure technologies.

"The SOA Leaders Council is a great opportunity to trade ideas with others who are serious about SOA. It's an open forum to bounce ideas off each other, see if others are facing the same challenges and hear what leading analysts have to say." Kevin Rice, Enterprise Architect, Allstate Insurance Company

"I really appreciate the lessons and contributions of other company's experiences. SOA Leaders enables us to collaborate and compare our experiences." Karen Kaminski, Harley Davidson

"The most valuable information for our initiatives are the best practices derived from the experiences of our peers. SOA Leaders is unmatched in the quality of the members and dialogue," said Eric Norman, Director of Strategic Applications Development, Intercontinental Hotels Group.

SOA Leaders Council members are employed by end-user organizations implementing SOA are actively involved in the architecture, design and deployment of the SOA, and are ready to share questions and lessons learned. For more information about becoming a member of the SOA Leaders Council, please visit, http://www.soaleaders.org/join.html and complete the short application.

SOA Leaders is supported by its technology benefactors – AmberPoint, Parasoft, Reactivity, Systinet and ThoughtWorks.