

White Paper: Sarbanes-Oxley in SOA



Table of Contents

| | |
|--|---|
| INTRODUCTION | 1 |
| WHY SOA MATTERS..... | 1 |
| THE CHALLENGE OF SOA COMPLIANCE | 1 |
| Identity Management Requirements for SOA..... | 2 |
| Access Control Requirements for SOA | 2 |
| Transmission Control Requirements for SOA | 3 |
| THE NEED FOR SOA GOVERNANCE | 3 |
| Identity and Trust Control Framework..... | 3 |
| Declarative Policy Definition Environment | 4 |
| Automated Policy Provisioning, Coordination, and Contracting | 5 |
| Compliance Verification Framework..... | 5 |
| SUMMARY | 6 |
| ABOUT LAYER 7 TECHNOLOGIES | 6 |
| CONTACT LAYER 7 TECHNOLOGIES | 7 |
| LEGAL INFORMATION | 7 |

INTRODUCTION

Service-Oriented Architecture (SOA) is transforming business integration. Business processes no longer just involve humans accessing machines. Technologies like XML and Web services are making machine-to-machine interactions commonplace where business processes stretch across many cooperating and coordinated computers. Applications instantiated as reusable and dynamically callable services make IT-enabled business processes flexible, agile, and efficient—but at a cost. Business processes built on SOAs make identity and access management complicated; yet they are more important than ever.

The financial controls and reporting required by the Sarbanes-Oxley Act of 2002 (SOX) forces companies to rethink how they govern their IT processes. In particular, section 404 requires every publicly registered company to demonstrate the effectiveness of their internal control structures and reporting procedures. This requires identity and access infrastructure that can both control and validate user-machine interactions as well as SOA-based machine-machine interactions. In this White Paper, you will learn how to extend SOX compliance to SOA.

WHY SOA MATTERS

SOA is an integration framework for dynamically interconnecting loosely-coupled software components into on-demand business processes. SOA allows software functionality to be delivered as network callable services that can be composed into processes in a flexible, agile manner. What this means most often today is that one software system can programmatically call another by sending XML messages, usually via Web services technology.

An example of a SOA-based business service 401k delivery: A 401k provider will push the functionality in their consumer-facing application in the form of a Web service out to their employer-customers. This allows that employer to treat the Web services interface as an API, allowing them to integrate that service into their internal benefits portal. Essentially, the 401k provider has “pushed” the 401k functionality inside the “internal” application of their employer-customer inside their corporate environment. Employees can access all benefits information including their 401k through this portal even though only the regulated 401k provider can actually handle investment requests.

THE CHALLENGE OF SOA COMPLIANCE

In the post-Enron era, many regulations have been put in place to force transparency and accountability for financial, personal, and confidential information. These compliance regulations (HIPAA, Basel II, USA Patriot Act, etc) have elevated the need for companies to know and control who has access to confidential or sensitive data to track when unusual activities occur. In all cases, these new control and disclosure requirements create new demands for IT Governance.

Sarbanes-Oxley is no different. The Sarbanes-Oxley Act of 2002 (SOX), Section 404, requires effective and demonstrable internal control structures and reporting procedures for financial information in all publicly registered companies. This requires:

1. A framework for controlling and auditing who accesses financial information.
2. A framework for controlling and auditing what financial information is accessed.
3. A framework for ensuring that financial information is not compromised during transmission.

In the client-server or Web environment, this can in large measure be addressed by a user-centric identity and access control infrastructure plus some kind of point-to-point SSL tunneling between clients/browsers and servers. The situation in SOA, however, is more complex. Identities belong to machines and SOA interactions are not just point-to-point. In fact, machines in an SOA can act as intermediaries for other service requestors, creating challenges for managing and tracing identity across long-lived, multi-hop transactions. For the same reason, ensuring message integrity across a multi-hop transmission is complicated. Point-to-point SSL will not be enough.

Identity Management Requirements for SOA

The problem of controlling who accesses financial information can be addressed through identity management. Several vendors offer products specifically designed to manage user identities and their associated credentials. However all the products are human centric. Extending identity infrastructure to SOA requires an ability to provision identities for machines, certify those identities with some non-repudiable credentialing mechanism like PKI, and establish some evidence trail for authentications even when the authentications span identity domains.

Access Control Requirements for SOA

Similarly, the problem of controlling what information is accessed at what time can be reduced to managing an identity's access entitlements. In the client/server world, this is addressed by an Access Management system that can control access to a specific server or file resource. In the Web world, this is typically addressed by a Web Single Sign-on solution that restricts access by URL. However, managing access entitlements for SOA is more complex. Access needs to be controlled at a fine-grained service or sub-service operation level. This requires an ability to inspect every XML and SOAP message to resolve a service or operation address and then perform an authorization decision based on the requestor's identity (or identity membership in a group), or the request's time-of-day, IP range, metering parameter, content, or security policy conformance (for instance, does the requestor's message include a signature?).

Transmission Control Requirements for SOA

In the Web world, protecting content from compromise during transmission can be easily handled by SSL (secure socket layer). SSL is natively supported by all Web servers and browsers, providing confidentiality, integrity, and protection against transmission threats like man-in-the-middle and replay attacks. However, protecting content during transmission in SOA is more problematic. As discussed previously, SOA security cannot be handled by point-to-point transport security like SSL (in the case of HTTP) because Web services can transact across multiple intermediaries and transports. Consequently, privacy and integrity must be rolled into a Web services message. This requires an ability to selectively apply and enforce encryption (for confidentiality) and a digital signature (for integrity) to a whole message (SOAP envelope), message part (SOAP element/XML field), or message attachment, depending on where the financial information is carried in a Web services transmission. Other requirements include the ability to protect against transmission compromise from things like message spoofing (man-in-the-middle attacks) or session hijacking (replay attacks), and there needs to be a way of negotiating cryptography keys ahead of each security session.

THE NEED FOR SOA GOVERNANCE

SOA Governance is predicated on an ability to establish, control, and verify security between SOA consumers, providers, and intermediaries participating in a transaction. This requires an ability to certify and validate SOA consumer identities even in transactions that span federated departments and partners. It also requires an ability to define, provision, and execute policy across SOA consumers and providers in a consistent way. Accomplishing this in a way that preserves the fidelity of a policy from its definition to execution, while also assuring architectural flexibility, demands very specific capabilities and features from a SOA Governance framework.

Lastly, a SOA Governance framework must result in contractible and verifiable behavior between SOA consumers and providers. That requires an ability for SOA providers and consumers to contractually agree on policy terms for their transaction in a way that is both non-repudiable and verifiable using either real-time policing (monitoring for contracted policy violations) or forensic audit. In an ideal SOA Governance framework, violations to contracted policy will result in alerts or fully automated remediation.

Identity and Trust Control Framework

The cornerstone of an effective SOA Governance framework is the ability to trust identities; all secure SOA transactions depend on identity trust between SOA consumers and providers. For a SOA provider to trust an SOA consumer's identity requires every service consumer to clearly establish its identity in a certifiable and non-reputable way to every service provider. This also requires every SOA provider to have a reciprocal mechanism for authenticating an SOA consumer's identity by validating its identity certification credentials. In scenarios where the SOA consumer and SOA provider lie in the same identity domain, this task is made simple because a common identity certification directory will exist. However, in scenarios where SOA consumers and providers reside in different identity domains, this task is complicated since no

common identity store will exist. In these scenarios, an SOA Governance framework must be able to establish trust across identity domains so that an identity certified in one domain can be trusted in another. This kind of transitive trust depends in practice on the ability of an SOA provider to establish trust with an identity validation intermediate that lies in the consumer's domain, and for that intermediate to be able to electronically vouch for the identity of the consumer.

Therefore, any framework for governing trust between an SOA consumer and provider must include not only a mechanism for certifying and authenticating an identity inside a single domain but also between domains. In practice, this requires a facility for establishing trust between identity domains as well as exchanging evidence for identity authentication (i.e. federating identity authentication) across the resulting trusted domains. Hence, a total trust framework for SOA governance includes the ability to:

- o certify identity,
- o validate identity certification credentials,
- o establish trust between domains, and
- o federate identity authentication operations.

Declarative Policy Definition Environment

Once a mechanism exists to establish trust and validate identity between SOA consumers and services, there needs to be a clear, uncontestable way of defining SOA security policy. Since each SOA business process can have a distinct security relationship between the service consumer and provider, there needs to be a way of defining SOA security policy tailored to each service consumer and provider relationship. This requires an ability to declaratively define identity-specific security policies. However, security policies are not monolithic; they are assembled from atomic security assertions that express security preferences.

Defining security preferences inside an SOA security policy requires an ability to assemble a policy statement from these atomic security assertions. It also requires an ability to prescribe how these policy statements or expressions are to be processed. Variability based on the successful execution of a security assertion, or specific message content, or real-time event, needs to be accommodated. Adding the ability for a policy to branch or change based on a transaction dependency makes security adaptive and integrations flexible.

An ideal SOA security policy authoring environment, therefore, would include a facility for constructing complex policy statements and execution instructions from atomic security assertions; would test and validate policies to ensure that they do not violate corporate security rules; would allow policies to be templated and inherited to improve consistency; and would provide lifecycle controls so that policy statements can be versioned and work flowed between multiple authors.

Automated Policy Provisioning, Coordination, and Contracting

SOA Governance is predicated on the ability to define, provision, execute, and verify trust and policy instructions in SOA. However, since SOA is fundamentally an integration framework, these trust and policy instructions need to be agreed upon or contracted by service consumers and providers in advance of the first data exchange. The most straightforward mechanism for accomplishing this involves a service provider publishing a policy contract that can be consumed by a service consumer in some legally verifiable way. Compliance can then be checked against the policy contract. This could be accomplished through out-of-band developer mechanisms or in-band automated mechanisms that allow SOA providers and consumers to dynamically negotiate a service policy contract in-band. Clearly, the fastest and least error-prone would involve some in-band automation mechanism. Several products exist to satisfy this need for SOA.

Compliance Verification Framework

Once technologies are in place to control identity and trust, once security policies are declaratively defined, and once those policies are provisioned and coordinated between SOA consumers and providers, a runtime compliance verification framework must exist to complete the SOA Governance picture. This requires an ability to police, audit, alert, and potentially update security policy dynamically. Just as a security camera or security guard is required to maintain compliance to physical security policy, automated monitoring of SOA security policy is required to fully establish SOA policy compliance. We need to make sure that the security policy we have employed does indeed accomplish the goals we set for the level of trust and security required for the assets that might be at risk so that if a breach does happen, there is an automated response to remedy the problem.

Policing

The central requirement of a compliance verification framework is a system for policing SOA behavior. This requires an ability to monitor how services are performing in real-time by inspecting the data entering and leaving an SOA provider. With visibility, a service provider's behavior can be checked or validated against a predefined policy prescription. Violations against a policy can then be captured and used to generate remedial events like an alert, service shutdown, or active update to the service provider's policy.

Audit

Sarbanes-Oxley requires organizations to log IT transactions to provide audibility and non-repudiation. Audit in an SOA transaction could involve tracking any number of activities and incidents. However, first and foremost, audit in SOA has to be able to provide evidence that a particular identity accessed a specific service resource, that the service consumer's request satisfied the service provider's security policies (communication integrity, privacy, data cleanliness, etc.), and that the service provider's response satisfied security and performance contracts established with the service consumer (particularly if an SLA is specified in the policy). Secure logging of what happened, when, by whom, and under what terms in an SOA communication therefore underpins any forensic audibility of a transaction.

Alerting

Generating notifications or alert events from policy violations are essential for effective compliance. When SOA transactions violate security behavior contracted between an SOA service consumer and provider, some alert notification must occur for compliance verification to be effective. Alerts can be represented as alarms directed to a human administrator. Alternatively, they can be real-time electronic events that in turn are used to trigger an automated remediation event like service shutdown or policy change.

Remediation

Real-time policing and forensic audit are essential for identifying trust and policy violations in SOA. Violations in contracted behavior between a service consumer and provider necessitate immediate remediation for effective SOA Governance. A facility for generating real-time alerts is clearly the first step towards remediation. With an alert, a human operator can be notified of a policy violation and take corrective action. Alternatively, an alert can be used to trigger an event that can directly affect policy or operation in an SOA transaction. This would ensure that an SOA transaction could dynamically adapt to trust or policy violations with automated remediation. This should be the ultimate goal in any compliance verification framework since it most closely adheres to the SOA precept of flexible, just-in-time integration.

SUMMARY

SOA is transforming IT. Sarbanes-Oxley is similarly transforming how IT is governed. Ensuring that SOA fits within the new governance imperative requires a rethink of basic concepts around trust, policy creation, policy implementation, and policy verification. By piecing together a few key technologies, enterprises can satisfy compliance requirements set-down by Sarbanes-Oxley legislation.

ABOUT LAYER 7 TECHNOLOGIES

Layer 7 Technologies helps enterprises accomplish secure and cost-effective business integration using XML and Web services. Layer 7 Technologies' SecureSpan™ product line is the first technology that addresses security and governance across a Web services deployment without expensive and inflexible programming. With the SecureSpan™, customers realize lowered integration costs, increased security reliability, and the ability to future-proof their Web services investments. Contact Layer 7 Technologies or visit www.layer7tech.com for more information.

CONTACT LAYER 7 TECHNOLOGIES

Layer 7 Technologies welcomes your questions, comments, and general feedback.

Email:

info@layer7tech.com

Web Site:

www.layer7tech.com

Phone:

604-681-9377

1-800-681-9377 (toll free)

Fax:

604-681-9387

Address:

Head Office:

Layer 7 Technologies

Suite 501 - 858 Beatty Street

Vancouver, B.C.

V6B 1C1 Canada

LEGAL INFORMATION

Copyright © 2005 by Layer 7 Technologies, Inc. (www.layer7tech.com). Contents confidential. All rights reserved. SecureSpan™ is a registered trademark of Layer 7 Technologies, Inc. All other mentioned trade names and/or trademarks are the property of their respective owners.