securing services-based IT architectures with CA SiteMinder®
Web Services Security
# Table of Contents

**EXECUTIVE SUMMARY**  
3

**SECTION 1: Challenge**  
4  
The challenge of security management for services

**SECTION 2: Opportunity**  
5  
The journey to service-oriented architecture
  
  Understanding SOA security  
  
  SOA security layers  
  
  What’s missing from SOA/WS security?

**SECTION 3: Benefits**  
11  
CA SiteMinder Web Services Security addresses security management challenges
  
  Key benefits  
  
  CA SiteMinder Web Services Security product architecture  
  
  CA SiteMinder Web Services Security key features

**SECTION 4:**  
15  
Adding value at every layer of the SOA environment  
  
  Cross-industry applicability

**SECTION 5:**  
18  
Conclusion
Web Security Management Solution from CA Technologies

executive summary

Challenge
Service-oriented architectures and Web services (SOA/WS) are emerging as the next major wave of application architectures for IT-intensive enterprises. Organizations are looking to SOA/WS to improve the speed, flexibility, and cost of building and deploying applications for both internal and external uses. However, as with all new IT architectures, and in particular those that are highly distributed, security management can be a significant challenge. Without a proper architecture, application security is often created in silos. This leads to increased risk of information leakage, cost of security administration and complexity of complying with IT impacting regulations.

Opportunity
With centralized security management enabled by CA SiteMinder® Web Services Security (WSS), organizations can manage the security of their enterprise SOA/WS deployments no matter how many Web services or different infrastructural technologies are deployed. Providing centralized, policy-based security as an integral part of the SOA/WS service infrastructure helps enable the externalization of security from the services themselves. This eases the administrative burden and cost of providing consistent and reliable enterprise security for SOA/WS.

Benefits
CA SiteMinder WSS provides identity-based Web services security—authentication, authorization, and audit (AAA). This solution can thus reduce the time and cost devoted to service development and maintenance and help reduce IT risk through greater control and monitoring. In addition, when CA SiteMinder WSS is used in conjunction with CA SiteMinder®, the combined solution provides a broad Web security solution. This solution secures both traditional Web applications/portals and XML-based Web services—leveraging the same agent and policy server-based architecture.
Section 1: Challenge

The challenge of security management for services

Service-oriented architectures (SOA) have emerged as the newest and often best approach to build and deploy IT applications. Typically embracing Internet standards through the use of XML-based Web services (WS), the SOA approach holds the promise of greater IT flexibility and agility. This is because it enables organizations to publish their services for the multitudes of potential internal and external service consumers. This service-based approach fundamentally changes the way applications are designed and constructed. It can support an infinite variety of business processes, but simultaneously challenges the way that IT organizations govern, manage and secure applications and data.

Just as with the first arrival of secured Web applications and portals in the 1990s, the arrival of SOA/WS-based applications creates a number of IT and security management challenges that must be addressed before they can be deployed at scale. Given that SOA/WS can be deployed for internal use, external use, or a mix of both, who gets access to what matters intrinsically for SOA/WS. In short, SOA/WS need the equivalent functionality that has become standard with websites and portals—firewalls and Web access management (WAM) systems; leveraging the general approach and philosophy that is so proven for website security, but adapting it in particular to service orientation and XML-based Web services.

Security management for SOA/WS does not require a reinvention of the wheel, as security requirements with SOA/WS are largely consistent with those of web-based applications. For example, with both websites and SOA/WS, it is important to keep private data confidential, and make sure that messages have not been tampered with (integrity). It is equally important to discern the identity of the requester (authentication), decide the level of entitlement that the requesting application or user is granted (authorization), and track what has and is happening from a security point of view in the environment (auditing/reporting).

A typical, first phase SOA/WS deployment today often combines a traditional portal deployment on the front end for the human user and a Web service call on the back end acting on behalf of that user at Web services either hosted internally or by partners externally. Given this scenario, many organizations want to preserve the identity and security context seamlessly in all steps of this application. In effect, organizations want the user’s session initiated the traditional way through a user login to be carried over to one or more Web service transactions on the back end. Along these same lines, since many of these Web service hops often involve separate internal or external security domains, trust needs to be enabled and enforced (through federation) across these security domains in a standards-based and scalable way.

CA SiteMinder Web Services Security was developed by CA Technologies to address these issues by providing an identity-centric SOA/WS security software product that secures access to services by inspecting the security information contained in the XML documents submitted by the service consumers. Leveraging a core set of SOA/WS standards, CA SiteMinder WSS uses centralized security policies bound to user identities to provide authentication, authorization, federation, session management, and security auditing services. CA SiteMinder WSS fits into a heterogeneous SOA/WS
Section 2: Opportunity
The journey to service-oriented architecture

It is estimated that the majority of large organizations around the world have either started to use Service-oriented Architectures/Web services (SOA/WS) or are planning to do so in the near future. The move to cloud-based deployments only adds increased emphasis to the use of SOA/WS for application integration. The attraction of SOA/WS largely rests on its ability to increase application development and deployment speed, reusability and flexibility while reducing IT costs. Extending gains already realized by leveraging traditional Web portals and Web applications, SOA/WS takes the model of cross-domain applications focused on serving human users and generalizes this concept to computer-driven applications that may or may not be acting under the direct control of a person. SOA/WS directly leverage the benefit of the Internet and Internet technology to provide application integration flexibility no matter whether the service consumer resides on the Internet or Intranet/Extranet of the enterprise. The SOA/WS approach both eases internal application integration, while leveraging standards to open up the same services to the world at large, whether they are customers, partners, or other third-party organizations.

Using IT to enable and speed these third-party relationships is not a new phenomenon. Fixed format data structures like electronic data interchange (EDI) have traditionally been used to send data back and forth between trading partners. Yet, in this new generation of truly integrated global business processes, EDI is a highly constrained communications system that is not open to the infinite types of communications and transactions that organizations need. However, EDI provides a useful example of what is possible and can be considered the first generation of relatively wide-scale, cross-organizational digital information exchange. What is needed is a set of open, standards-based interfaces that any organization can use to integrate business systems in a secure, reliable fashion.

As with all new technologies, there are challenges with SOA/WS that must be addressed before this technology can be used widely. Given the massive scale and flexibility inherent to SOA/WS, any solution deployed must be reliable, available, scalable, manageable, and secure while helping to ensure that the environment can be effectively monitored. These key IT management issues require an evolution in thinking. While this paper focuses on the new security management challenges that SOA/WS bring to the forefront, it is important to remember that security is only one part of the IT management challenges raised by SOA/WS.

Historically the bad guys were attackers from outside of your organization, trying to launch attacks like denial-of-service, message spoofing and DNS poisoning to impact the ability of the application to
function. This is no longer a good assumption, as insiders are increasingly both the consumers of an SOA/WS application and a legitimate threat vector, stealing sensitive data and bringing down business processes, regardless of the insider’s true intent. At best, the concept of insiders and outsiders gets extremely vague when thought of from an SOA/WS perspective.

Because of this vagueness, the traditional approach of deploying security in multiple layers, where different products and processes have secured the network, data center, applications and end points is no longer sufficient. In an environment where services are requested by a requestor that can be either internal or external to the organization (or by an inside application acting on behalf on an outsider), security services like authentication and authorization are more important than ever. SOA/WS enable a new generation of open, integrated and accessible applications, but also require a consistently enforced set of usage policies that can scale to provide management services at a scale and granularity beyond what has been seen in the IT world to date.

Another security management risk for SOA/WS is the practice of building security logic (confirming the identity of the requester and what they can access, for example) directly and uniquely into each service as opposed to providing this as a shared security service. As SOA/WS deployments continue to scale, this tendency toward building security silos is increasingly impractical for organizations that have services that could easily number into the hundreds and potentially thousands. Redundant security silos are not only expensive to build and maintain, but also increase risk and make regulatory compliance more challenging to attain.

Ultimately, it gets down to accountability and control. As SOA/WS-based applications proliferate, organizations must figure out how to provide at least the same level (and hopefully better) of security that is available for the current generation client/server and web-based applications. This is further complicated due to increasingly stringent regulatory requirements, which directly impacts IT and requires corporate executives to vouch for the sanctity of transactions and related financial reporting and to provide protection of private personal information.

Understanding SOA security

In trying to understand the security requirements inherent in an SOA/WS-based environment, it makes sense to look at how enterprises dealt with (and are still dealing with) the movement to web-based applications over the last ten years or so. The first generation of Web applications built security directly into the applications themselves. These so called security silos involved implementing a user directory, access control lists (ACLs), and sets of access policies for each application. Basically, each application in a fully siloed architecture literally handled its own user authentication, authorization, and auditing (AAA) at some level.

As organizations moved beyond having only a handful of Web applications, this silo-based approach to Web application security didn’t scale and ultimately proved to be insecure and costly to manage. Thus, in the 1990s, a whole new class of security applications was introduced to enable applications to externalize authentication, authorization, audit, and user administrative functions into a centrally managed, highly scalable security infrastructure that could be used by all the Web applications in the enterprise. At the same time, a standard technology for user directories called LDAP started to proliferate to provide the centralized repositories critical to scaling this externalized security infrastructure.
There are many parallels between Web applications and SOA/WS-based applications, including the fact that both can be deployed on an intranet (for enterprise use), an extranet (for business partners), or even the public Internet (for consumers). The main difference is that the user in an SOA world can be another machine talking the language of XML, WSDL and SOAP, as opposed to a person seeing a Web page rendered in a browser. But many of the security challenges are basically the same and can be mitigated using a similar security management approach.

Before jumping into possible solutions, let’s take a more detailed look at the security requirements of SOA/WS. While being mostly consistent with the security requirements of traditional Web applications, there are some differences that will also be highlighted.

Security requirements for SOA/WS-based applications include:

- **Threat/Malware prevention.** XML traffic is no different than Web traffic or email traffic in that it can be used to carry a malicious payload to its destination. As is best practice with other traffic types, there is a need to screen all the incoming XML traffic at the edge/DMZ to make sure there is no malware or other targeted attacks on business services, including viruses, denial of services, spoofed messages, etc.

- **Authentication.** Who is the other party that is trying to access a service? Regardless of whether the other party is a computer process or other Web service, before anything can be done, the identity of requester needs to be confirmed. No one just lets anyone into a high profile Web application without a positive authentication. SOA/WS-based applications should be no different.

- **Authorization.** Once the service consumer is authenticated, what can it do with the organization’s Web services? What services are they allowed to access? What data can be accessed and what transactions and business functions can be used? Just as users get entitlements to use certain functions in a Web portal, the Web service provider needs to grant similar entitlements on behalf of a service consumer, whether they are from the inside or from the outside.

- **Auditing and reporting.** Given serious regulatory requirements to log every material transaction and closely monitor business operations in case of a data breach or other problem, the SOA/WS environment must provide the ability to track each transaction and reconstitute business activities in a forensically sound way. Similarly, it’s critical to be able to provide enterprise-wide reports of activity.

- **Identity administration.** Organizations need to manage identities, credentials and entitlements for SOA/WS-based applications, just as they do today in traditional IT architectures. Since Web services often act on behalf of users or other applications or technology processes, single sign-on and the provisioning of credentials and access rights are critical to allowing the environment to scale securely.

- **Enterprise manageability/centralized policy management.** With the sheer number of potential services available through an SOA/WS-based approach, how can an organization get an enterprise-wide view of what is going on with potentially hundreds or thousands of distinct SOA-based applications running? Moreover, it’s critical to be able to build and enforce centralized security policy that can change quickly depending on business requirements, without impacting or changing the underlying business service.
• **Session management.** Similar to Web application single sign-on (Web SSO), Web services can be part of business processes where sessions need to be maintained across multiple Web services for an entire transaction. This can be thought of as a type of SSO for Web services.

• **Support of heterogeneous infrastructure.** A key advantage of Web and now Web services-based applications is that specific hardware, network or applications are not required as long as they adhere to a standard set of interchange technologies. Web services can be deployed in many different ways and no doubt will be in many large organizations. So the ability to protect them consistently given this heterogeneous world is critical.

• **Performance, reliability, availability and scalability.** Having all of these aspects of an enterprise-class computing environment goes without saying. Many Web applications need to be able to scale to the millions or ten millions of users with five 9s uptime. Likewise, SOA-based applications, where reusability is a key benefit, may have an order of magnitude higher level of usage with the same five 9’s availability requirements. Moreover, the interdependent nature of SOA/WS-based applications means that an issue in one component service could adversely impact many other services.

• **Standards support.** SOA/WS are driven by standards (such as XML, WSDL, SOAP, etc.), including a set of security standards (WS-Security and others), which need to be supported as a means of providing the requisite interoperability that enables eased deployment and management both for internal and externally facing services.

The above security requirements must be delivered in a flexible, enterprise-class environment that enables an organization to achieve the promise of SOA/WS. Given that many large organizations will ultimately have thousands of SOA-based Web services comprised of many different, self-contained components, the idea of building security capabilities into each component is not practical. Thus, SOA security (just as Web access management before it) needs to be delivered as a centralized infrastructure or service to maintain the highest level of flexibility and efficiency.

**SOA security layers**

Security for SOA/WS can be deployed in a variety of places depending on the application architecture. SOA/WS security is often implemented on the edge (or perimeter) of the network, within a SOA platform, or in a SOA application container as depicted in the diagram below. Given that to date there has been little integration between these disparate security areas, this has resulted in a tremendous amount of duplication in functionality. Thus enterprises have often had to manage similar security policies at different parts of the SOA/WS architecture.

Managing these multiple security policies can be problematic for a number of reasons. It’s more resource intensive, can result in security gaps, and also may duplicate similar defenses. Best practices dictate a layered defense for SOA Security, but those layers must be consistent, coordinated and managed within a centralized policy.
Ultimately, a SOA/WS security solution should support the application developers without burdening them with details of how each component service should be secured. But at the same time, a centralized and structured way of enforcing organizational policy across all deployed Web services that helps ensure proper end-to-end reporting is also critical. It is that balance that is driving many organizations to look to a SOA/WS security system that can provide the needed flexibility, while offering world-class centralized management. Let’s look into each layer in a bit more detail.

**Edge (perimeter) security.** Offered through hardware or software form factors that reside within the demilitarized network zone of an organization, these edge-based systems (also commonly known as XML security gateways or XML firewalls) are focused on being the first line of defense for SOA/WS applications. These systems are usually deployed as reverse proxies for XML traffic so all inbound messages are inspected and processed to help ensure security policy compliance.

These XML security gateways check for XML-based malware and other threats in inbound traffic, including viruses and denial of service attacks. Protocol translation can also happen at the perimeter to enable compatibility with deployed applications and other standards.
**SOA platform security.** Given the large number of services that are deployed in a large enterprise, many have implemented a SOA/WS platform that acts as an intermediary to connect, mediate and manage the available services. SOA/WS developers have the option of using some of the integrated security capabilities within the SOA platforms, but at the risk of both duplicating defenses, potentially leaving security gaps, and creating security silos to add to management and compliance challenges in the future.

The SOA platforms tend to use SOA/WS security standards (including WS-*) to be able to issue entitlements and support federation between different systems either internal or external to the organization.

**SOA container security.** SOA/WS applications are deployed within containers, which typically are built using either the Java J2EE or the Microsoft’s .NET specification. Since SOA/WS are standards-based, the development environment isn’t material to the deployment of the services themselves, but it does make a difference when trying to secure the environment. As with the SOA platforms, J2EE and .NET offer certain security capabilities that can be built directly into the application at the developer’s discretion, but have the same risks of duplicating functions, creating security silos, which adds to management and compliance challenges, and potentially leaving security gaps within the application.

**What’s missing from SOA/WS security?**

As mentioned above, duplicating security functions across the different SOA domains (Edge, Platform and Container) clearly is inefficient and requires significant additional management and developer resources that result in increased IT costs. Besides the overlap, it is difficult to implement a consistent SOA security policy across all layers and all of the disparate applications running in the environment.

A parallel can be drawn to Web access management, where initially there were many disparate levels of security implemented (edge, container, within the application), which were then consolidated into a common security infrastructure to both increase the level of security and decrease the amount of time and resources needed to secure those applications. The good news is that this problem was solved in the web-based application domain and many of those same techniques are directly applicable in the world of SOA/WS.

SOA-based applications will likely follow the same evolutionary path as web-based applications before it. This sets the stage for a new generation of SOA/WS security solutions to appear to enforce centralized security across all layers of the application, bringing together the best of both worlds. Today’s demanding SOA/WS applications require effective security at each layer, while using a common management interface, consistent policy enforcement, and integrated reporting for audit and compliance across the entire SOA ecosystem.
Section 3: Benefits

CA SiteMinder Web Services Security addresses security management challenges

CA SiteMinder WSS is well positioned to offer end-to-end security for SOA/WS by providing centralized policy management, policy enforcement for different security layers and central auditing to an enterprise SOA/WS deployment. By abstracting security from the services themselves, CA SiteMinder WSS helps you to significantly reduce the administrative burdens and other costs associated with providing security for SOA/Web services.

CA SiteMinder WSS inspects the security information contained in XML documents submitted by service consumers and uses this information to determine access. It provides enterprise-level functionality for SOA/Web services that are exposed internally and externally, helping to keep XML threats out while simultaneously controlling access for legitimate service consumers. Like Web access management before it, CA SiteMinder WSS largely abstracts security from the sphere of the application developer, thus enabling the developer to focus on the application logic and the security professional to focus on security and risk mitigation.

CA SiteMinder WSS brings a shared-services security vision to previously disparate SOA security silos. Built on top of a centralized policy server, every message is checked to enforce authentication and authorization policies. Additionally, inbound and outbound messages can be transformed and secured depending on the organization’s policy. With agents running on the major application servers, within leading SOA containers, CA SiteMinder WSS enables an end-to-end model to secure SOA/Web services from the edge to the container.

Key benefits

- **Consistent security.** As opposed to disparate security implemented in many places without common security policies, CA SiteMinder WSS provides a single point for access control and audit, consistently enforcing your organization’s security policies.

- **Reduced development costs.** Developers no longer have to build security into the respective components of their SOA/WS applications. Externalizing security provides significant developer efficiencies and results in faster time-to-market of business services.

- **Simplified security policy creation.** New Web services focused user interface simplifies security policy creation using a WSDL file. This administrative UI can also connect to multiple policy servers so you can manage all of your components from a single shared administration server.

- **Centralized auditing of SOA security policy.** Security policies implemented on centralized policy servers are checked at each stage of the transaction to help ensure that proper controls are implemented at every step of the transaction process. This also allows for central reporting to address auditing and compliance requirements.
• **Session management and single sign-on.** Centralized management of security also enables single sign-on (SSO) where once authenticated, Web service requests don’t need to be reauthenticated as the transactions move through multiple service steps (whether provided by the organization or by a third party) that make up a typical business process. Sessions can be configured to be valid for certain durations, providing more flexibility.

• **Reliability and high-availability for Web services.** Web services are always available and so is CA SiteMinder WSS, providing unparalleled reliability and uptime for even the most industrial strength, 24 hours a day, 7 days a week business processes.

• **Leverages standards in an open, platform-neutral environment.** CA SiteMinder WSS supports applicable Web services standards, including XML, SOAP, REST, WSDL, SSL, WS-Security, XML encryption and XML Signature.

• **Can make use of existing Web access management environment.** Built on the same policy server and agent-based architecture as CA Technologies industry leading Web access management offering, CA SiteMinder, CA SiteMinder WSS can leverage the same deployment environment as CA SiteMinder. Thus in combination, providing comprehensive Web security for both websites and applications and Web services.

Externalizing SOA security functions into a common infrastructure dramatically reduces development costs and provides a single point of access control and administration for the hundreds (or even thousands) of distinct services that will come into service at most large enterprises. CA SiteMinder WSS provides comprehensive security functionality from the edge to SOA containers.

**CA SiteMinder Web Services Security product architecture**

**Policy server—policy decision point (PDP).** The CA SiteMinder WSS policy server provides the policy decision point (PDP) for CA SiteMinder WSS and is the centerpiece of the centralized, policy-based management platform. The policy server was built on top of the policy server in CA SiteMinder, adding additional features designed to support XML-specific processing and security standards. The policy server can use the CA SiteMinder WSS SOA agents and an optional XML security gateway as policy-enforcement points (PEPs) for Web services wherever they are hosted.
CA SiteMinder Web Services Security and its PEP/PDP/PAP-based architecture.

CA SiteMinder WSS reference deployment architecture. CA SiteMinder WSS can be deployed in a highly distributed architecture that provides a combination of distributed policy enforcement points (via optional XML security gateways and via SOA agents) and centralized policy server based policy decision points.

1. Web service requests coming from outside into your network can be secured by an optional XML security gateway running in the DMZ. Alternatively a user may also access the Portal Server, which in turn makes a Web service request to a Web service hosted behind the DMZ.

2. Web services deployed within an enterprise can also make requests to each other as part of a particular business process. This is secured by SOA Agents as part of the last mile of SOA/WS security.

3. The common central policy server secures both Web service traffic and website traffic when CA SiteMinder and CA SiteMinder WSS are used together.
Built on an extensible and scalable architecture, security services can be added and enhanced as the security and management needs for Web services evolve. Integrating with industry standard LDAP directories, relational database systems, and mainframe identity stores for centralized management of user identity and entitlement information, you have the utmost in flexibility to implement CA SiteMinder WSS to meet your business requirements and extend your existing IT infrastructure, not vice-versa.

**SOA agents—policy enforcement points (PEPs).** CA SiteMinder WSS offers different policy enforcement points to help ensure end-to-end security for the entire SOA/WS enterprise infrastructure. Agents are available for the leading .NET and J2EE containers. New SOA Agents are regularly being developed such as those for additional ESB and SOA platforms.

**SOA security manager SDK—for custom built PEPs.** This Java API enables partners and customers to write custom SOA Agents for their environment. This open API allows CA partners and customers to extend their existing integrations with SOA Security Manager, in which SOA platforms, XML Firewalls or other appliances use CA SiteMinder WSS to provide a centrally managed authentication and authorization environment.

**CA SiteMinder Web Services Security key features**

CA SiteMinder WSS brings many important features to the market, including:

- **Centralized SOA security policy management** Implementing a shared services model, CA SiteMinder WSS externalizes security from the underlying Web service, providing the ability to consistently enforce security policy at all layers of the Web service—including the edge/perimeter, on the SOA platform, and within the SOA container.

- **Identity-aware Web services.** CA SiteMinder WSS binds the XML flow to a user identity (whether that user is a human or another application), helping to ensure that proper authentication, authorization and entitlements are maintained throughout the transaction.

- **Secure single sign-on and synchronized session management.** CA SiteMinder WSS manages session state and eliminates reauthentication of XML messages during multistep and federated transactions across multiple component services and organizational boundaries.

- **Credential mapping.** CA SiteMinder WSS not only authenticates and authorizes Web service requests, but also supports the generation of a new security token for that same requester, mapping identity in one security token to another security token—generally acting like a Security Token Service. Additionally, CA SiteMinder SMSESSION tokens can be mapped to standards-based WS-Security SAML Assertions to provider further openness and interoperability.

- **Creation of security policies using WSDL.** A new web-based user interface simplifies security policy creation off of the WSDL file. The WSDL file can be loaded from a file or from a URL location. Once loaded, the product’s UI displays all the Web service operations and makes it very easy for the administrator to secure different Web service operations with one or more authentication schemes.
• **Support for federation.** By supporting the WS-Security standard for security information contained in the XML/SOAP documents, inter-enterprise transactions can be managed across security domains from a single authentication. In fact, a typical use case for CA SiteMinder WSS is to provide a Web service based authentication service that can be leveraged as the enterprise’s shared authentication service.

• **Dynamic authorization based on XML content in the request.** As part of the authorization process, a security policy can be created to dynamically compare XML content against user attributes stored in user store.

• **Deployment flexibility for enforcement points.** CA SiteMinder WSS provides agents for the leading J2EE and .NET containers and is fully interoperable with solutions from other SOA vendors, including .NET, J2EE, and leading vendors including IBM, Microsoft, Oracle and many others.

• **Standards-compliant.** CA SiteMinder WSS supports all of the important Web services standards, helping to ensure interoperability and future-proofing, including XML, SOAP, REST, WS-Security (SAML, Username, X509), XML Signature, XML encryption, WSDL and SSL.

• **Extends the proven CA SiteMinder platform.** CA SiteMinder WSS provides seamless integration with the CA SiteMinder Web security platform, leveraging the same policy store and offering single sign-on to CA SiteMinder protected applications and Web services.

---

**Section 4:**

**Adding value at every layer of the SOA environment**

CA SiteMinder WSS adds value to every stage of the SOA/WS environment. As illustrated in the following use cases, the true value of a secure SOA/WS environment can only be achieved by leveraging a centralized security policy and centralized logging and reporting that also provides comprehensive, enterprise-scale security services.
**Key Benefits of this Use Case**

- The user only needs to authenticate to the bank portal, and the rest of the transaction is not visible to them. However, the user’s context is maintained at every step.

- The bank portal and the first-step (internal) Web service are secured by a single policy-based service enabled by a combination of CA SiteMinder WSS and CA SiteMinder. This saves development and security administration time and money.

- Each application/service is protected to the last mile (using agents), not receiving protection from some distant security service that may or may not be used. With CA SiteMinder WSS, there is no way to go around the security it provides.
• The credential mapping capability allows security context to be mapped to standards based security tokens, such as WS-Security SAML in this case, to complete the transaction. The Web services or portal itself didn’t need to worry about credential mapping as the security system provided by CA SiteMinder WSS took care of that. The use of security standards is particularly important, as in this case, when secure integration with third-party services are desired.

**Cross-industry applicability**

Nearly an infinite number of business scenarios can potentially be improved through the use of SOA/WS and the security that makes flexible, cross-enterprise digital information exchange possible. A few further brief scenarios are provided here to give you enough information to envision how SOA/WS might apply to your organization. In each case, you can see the importance of how a centralized, policy-based security system, such as CA SiteMinder WSS, dramatically streamlines the protection of these critical applications.

• **Healthcare.** A SOA/WS application can be used to provide real-time referrals and authorizations for appointments with specialists. A Web portal used by the primary care physician can send a Web services request on the back-end to the referral Web service to verify whether the referral is allowed or not based on information from the physician and the health plan in which the patient is enrolled. A consistent security policy can be enforced at all stages of the transaction even though each side uses a different security solution—due to the usage of common standards that facilitate interoperability. In this case, CA SiteMinder WSS can be used on either side of this scenario. However, if it is assumed that it is used to protect the referral service, not only can CA SiteMinder WSS review the validity of the security information in the request for authentication purposes, but it can also take part in the authorization decision at multiple levels.

• **Financial.** A thick application running on a trader’s desktop can call multiple Web services using multiple protocols and formats to perform currency and options trading services. CA SiteMinder WSS, acting as the shared authentication service, can provide a WS-Security SAML Assertion to the desktop client, which can be reused to get access to these and other secured Web services, whether hosted inside or outside the organization. CA SiteMinder WSS can also be used to secure any of these Web services, particularly and most likely those that are hosted internally, at the last mile of the service itself.

• **Shipping.** A shipping company can expose real-time shipment tracking information through a Web service for integration with their customers’ particular applications.

• **Manufacturing.** A global auto manufacturer can roll out innovative informational services on an ongoing basis directly to their end customers in their cars, whether on a fee for service basis or as part of value-added product bundles. In this way, access to services can be determined in part by the identity of the car itself and can include ongoing monitoring of the car’s performance, need for servicing, and the provisioning of premium services that were not purchased at the time of original acquisition or that weren’t available at that time. When you imagine all of the services that might be useful while traveling in a car, you can see that there are many services that an SOA/WS-based approach might provide. CA SiteMinder WSS could take on the important role of protecting these various services from misuse or direct attack.
Section 5: Conclusion

Service-oriented architectures and Web services are emerging as the next major wave of application architecture. SOA/WS have as their goal to improve the speed, flexibility and cost of building and deploying applications for both internal and external audiences. However, security strategies and architectures need to be planned in advance, or organizations will be at risk of repeating the mistakes of the past with security constantly playing a game of catch up and being deployed as a collage of technologies and processes.

It is not sufficient to address the SOA/WS security issues that are discussed in this paper based upon the traditional approach of deploying many inconsistent, incompatible and overlapping layers of security. Security must be architected into the environment as an infrastructure service, enabling flexible and cost-effective deployment, from edge to container.

The good news is that the security issues we face for SOA/WS-based applications are very similar to those we dealt with as traditional web-based applications became prevalent. Organizations need to centrally manage the security of their enterprise SOA/WS deployments, just as they do today for their websites and portals, no matter how many Web services or different infrastructural technologies are deployed. This can be accomplished by providing centralized, policy-based security as an integral part of the SOA/WS infrastructure, enabling the abstraction of security from the services themselves.

CA SiteMinder WSS extends the proven CA SiteMinder architecture to provide the industry’s most comprehensive SOA/WS security platform, providing both identity-based Web services security—authentication, authorization, and audit (AAA). In addition, CA SiteMinder WSS when used in conjunction with CA SiteMinder, provides a comprehensive Web security solution that secures both traditional Web applications/portals and XML-based Web services, leveraging the same agent, proxy, and policy server-based architecture.

To learn more about CA SiteMinder Web Services Security, please visit [ca.com/siteminder-wss](http://ca.com/siteminder-wss)