# INTRODUCTION TO SOA GATEWAYS: BEST PRACTICES, BENEFITS & REQUIREMENTS

*Learn best practices and common deployment scenarios of SOA Gateways and why they are an essential component of a secure, robust and scalable SOA deployment.*

## I. OVERVIEW

Modern web services-based Service Oriented Architecture (SOA) enables service consumers and producers to exchange messages over ubiquitous standards such as XML and SOAP. Typically, companies embark on SOA projects for system-to-system integration within their corporate domains as well as with external trading partners. The success of a SOA deployment is measured by the level of reuse of producer services. A well thought out SOA is flexible, loosely coupled and nimble and enables consumer applications to rapidly integrate with published producer services. The driving force of a successful SOA to reuse producer services is usually at odds with security. Integration flexibility and security typically pull in diametrically opposite directions. Integration pulls corporations towards "opening up" internal systems for other systems to call into whereas security pulls a company towards "locking down" business information through access control and data privacy. For a successful SOA deployment, ease-of-system integration without compromising security is paramount. Overly strict security models can eliminate the ease-of-integration advantage and an overly open SOA deployment with little or no consideration for security is a recipe for disaster, especially where sensitive and valuable corporate data is involved. SOA Gateway products such as Forum Sentry (a FIPS and DoD Certified hardware appliance) and Radware AppXML provide the necessary balance between rapid integration and security that results in a successful SOA deployment.

A SOA Gateway is a core infrastructure component of a SOA deployment with the ability to integrate services securely. Typically deployed as a hardware appliance, a SOA Gateway seamlessly controls access to services, protects information through data-level encryption, ensures the integrity of a message through signatures, and controls corporate information flow. This article covers SOA Gateway deployment best practices, benefits and requirements with a focus on service virtualization, message privacy and integrity, and message control and auditing.

## II. BEST PRACTICE: ENABLE SERVICE VIRTUALIZATION AND CONTROL

*Description:* Service Virtualization - the most important best practice of a SOA - is the ability to create a virtual service from one or more Web Services Description Language (WSDL) files generated by producer applications such as application servers, RDBMS, CRM and ERP systems. As shown in Figure 1 below, service virtualization across multiple producer systems is accomplished through an intermediary SOA Gateway that sits between the producer and the consumer. Instead of directly importing services from a variety of producer applications, consumer applications can import virtualized services from an intermediary SOA Gateway that aggregates and consolidates multiple back-end producer services into a single WSDL and a single entry point. Service virtualization enables enterprises to expose only specific business services required by a consumer rather than exposing all services from producer applications. On requesting a WSDL from the intermediary SOA Gateway, the consumer is presented with a set of virtual services that it is allowed to access. In the sample deployment shown in Figure 1 below, the "Internal" consumer is authorized to use services A-C and D,E from the producer Application Server and Business Application respectively. However, the "External" consumer, perhaps a supplier, is only allowed to use services B and F. Thus, a SOA Gateway acts as a central point of control that cloaks and safeguards producer services through access control policies and through virtualization only shows the services available to a consumer.

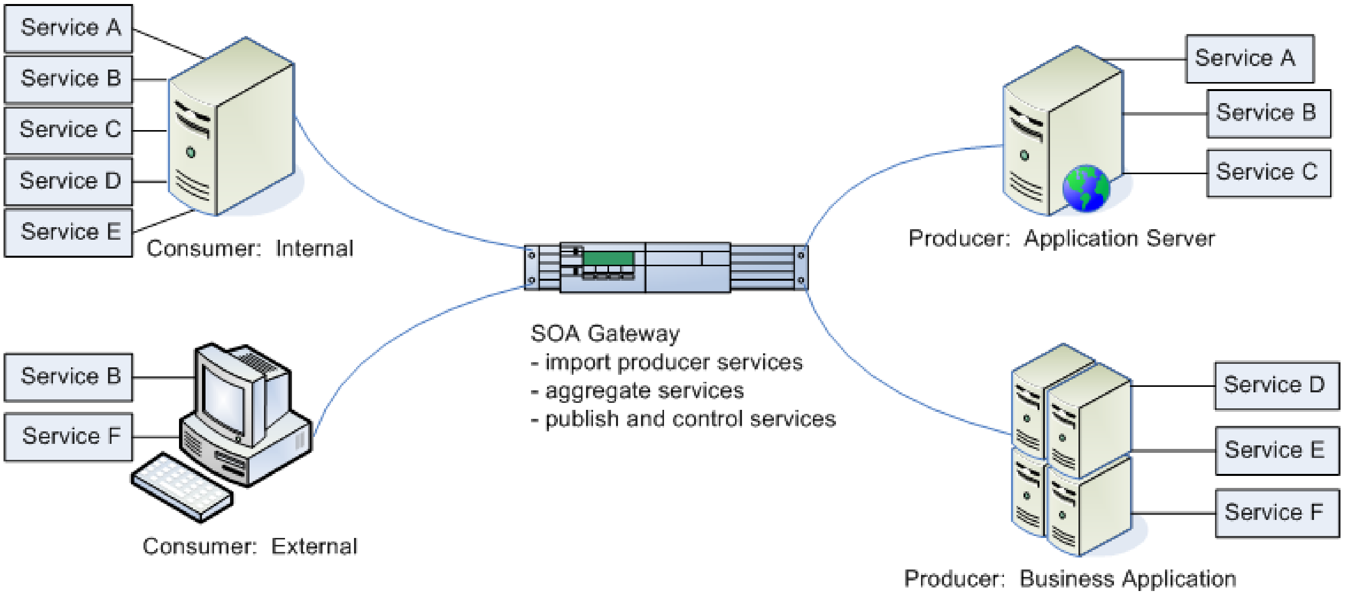

*Figure 1: Enabling Service Virtualization through a SOA Gateway*

*Benefits:* Enabling Service Virtualization by using SOA Gateways has significant benefits including:

- *Consistency:* Virtualization enables service selection across multiple WSDLs and exposes only authorized services to clients as a coherent single WSDL.
- *Security:* The virtual WSDL can selectively expose some of the services of the original WSDLs. The WSDL endpoints are cloaked with only the SOA Gateway endpoints being exposed. The SOA Gateway endpoints are protected through credential-based access control.
- *Productivity:* Service virtualization improves productivity by enabling mixing services across different producer services without having to copy and paste parts of the desired WSDLs into new WSDL files. It allows a customer to be able generate a library of all the services supported by its organization and only expose the ones required for a particular customer.

*Requirements:* SOA Gateways require industry-hardened WSDL and schema parsing to import, aggregate and publish complex WSDLs generated from a variety of application servers, RDBMs, and business applications. For enterprise-class service virtualization, a SOA Gateway is required to have the following essential features:

- *Integration with existing Identity Management Systems:* A SOA Gateway must have deep authentication and authorization-level integration with existing Identity Management Systems such as CA SiteMinder, Sun Access Manager, and HP Select Access for service access control.
- *Identity Bridging*: A SOA Gateway must have sophisticated identity token processing capabilities to intercept, process and convert credentials among a variety of protocol formats (Basic Auth, SSL Mutual Auth) and content formats (SAML, WS-Tokens, X.509 Tokens) for Authentication and Authorization decisions.
- *Manage complex WSDLs*: Ability to parse, merge and administer multiple compound WSDLs and schemas and avoid namespace collisions while aggregating WSDLs from multiple systems.

Deploying Service Virtualization with granular access control is crucial for a scalable SOA. The vices of free-for-all services can quickly set chaos within a SOA deployment. A SOA Gateway can control and manage an increasingly complex SOA deployment through service virtualization. Identity integration goes hand-in-hand with service virtualization. Leveraging existing identity infrastructure with service-level access control is the most crucial best practice of a SOA deployment. The quality of a SOA Gateway integration with identity systems along with the gateway's performance optimization features - such as intelligent credential caching - will determine the overall performance of a SOA deployment.

## III. BEST PRACTICE: ENFORCE DATA-LEVEL PRIVACY AND INTEGRITY

*Description:* Data-level privacy and integrity is the cornerstone of enterprise-class SOA. With SOAP/XML encryption and signature, confidentiality and integrity remain "always on" by being independent of transport protocols. With security now living within the SOAP/XML messages, it does not matter if the transport pipe – HTTP, FTP, JMS – between Web service consumers, producers, or intermediaries is SSL enabled.

The protocol independent nature of SOAP/XML-based messaging within a SOA decouples the messages from protocol security such as SSL. In a SOA, messages traversing the enterprise may do so over non-secure protocols such as FTP and JMS. To ensure that a message retains its privacy and integrity, security has to live within the message through data-level encryption and signatures. By using WS-Security (an OASIS security standard), messages can be granularly encrypted/decrypted and signed/verified for ensuring privacy and integrity at the data level, independent of protocol security.

Deploying a SOA Gateway to handle data-level encryption/decryption and signature/verification is a widely used industry best practice for ensuring strong privacy and integrity. Figure 2 below shows a typical SOA Gateway deployment for verifying and decrypting inbound messages from a consumer that has first encrypted and then signed the message before sending it out to the producer services. The message is intercepted by the SOA Gateway that performs a signature verification on the message to ensure that it has not been tampered with and then decrypts the message before sending it to the back-end producer application. The SOA Gateway takes away the burden of signature verification and decryption from the producers and provides a central location for maintaining security policies.

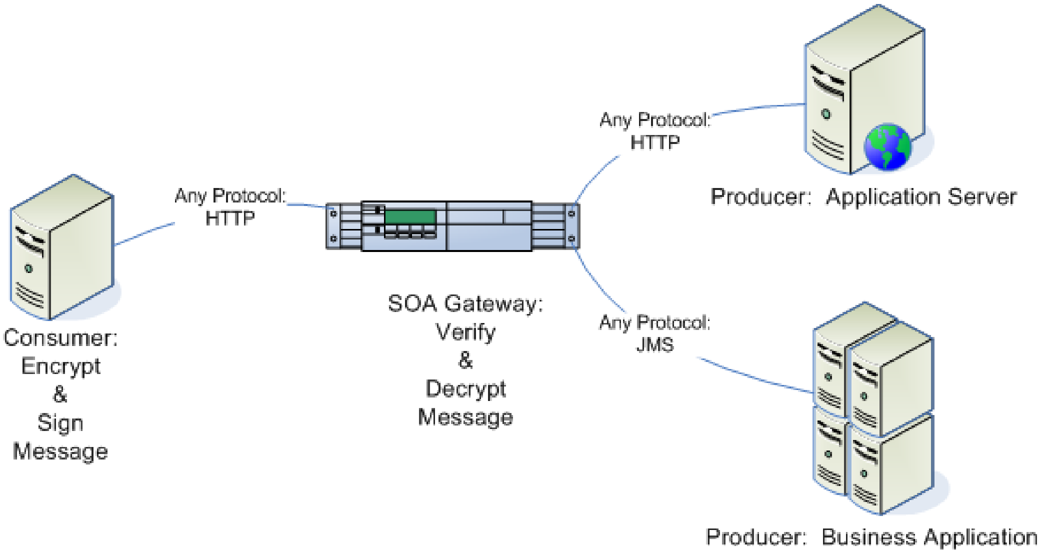

*Figure 2: Enabling Data-level message privacy and integrity.*

*Benefits:* Enabling Data-level privacy and integrity using SOA Gateways has significant benefits including:

- *Consistency*: By centralizing security policies and separating the security policies from the producer services, SOA Gateways provide a high-degree of policy consistency and control. Service implementations become independent of security policies that are centrally managed on the the

SOA Gateway.

- *Security*: Data-level security provides "data-at-rest" security over protocol-level security (SSL). Highly sensitive data within a message can be secured using strong cryptography whereas public data can be left as clear text. Data-level security is always-on, even when the message is at rest where it is most vulnerable. SOA Gateways provide protection of messages in-flight through protocol security as well as security for messages at rest through granular data-level security.
- *Productivity*: By centralizing security policies to a SOA Gateway, developers building producer services need not worry about writing code for security policies. For example, as shown in Figure 2, developers can focus on building core functionality rather than writing code that verifies and decrypts messages - functionality that is handled by the SOA Gateway. With no coding required for such centralized security functions, the overall productivity of the SOA project increases significantly.

*Requirements:* SOA Gateways have to be secure, robust, performant and highly interoperable in an enterprise-class environment where many internal and external systems are exchanging messages. For deploying privacy and integrity, a SOA Gateway must meet the following requirements:

- *Extensive Standards Support*: A message encrypted has to be eventually decrypted. Similarly, when an application signs a message, at some point, this message has to be verified to check for its integrity. WS-Security provides extensive standards support to ensure that when a system performs a security operation, the inverse operation can be performed by the receiving entity. A SOA Gateway intercepts secured messages from a variety of consumers and has to perform inverse security functions on the messages. Without an extensive standards-based implementation, the SOA Gateway would fail to process messages from a broad array of consumers.
- *Robust PKI Management*: Encryption/Decryption and Signature/Verification are based on Public Key Infrastructure (PKI). A SOA Gateway is required to support Key Life-cycle Management such as Public-Private Key Generation, Enrollment and Revocation. For hardened security, SOA Gateways also use Hardware Security Modules (HSMs) to protect private keys used for message signature and decryption. Solid PKI Management coupled with HSMs is a requirement for SOA Gateways deployed at the edge in a corporate DMZ. In highly senstive SOA deployments within, for example, the Federal Government and Financial institutions, only SOA Gateways that have been independently certified by non-commercial agencies should be considered. Hardware appliance-level FIPS and DoD/PKI are two certifications that are strongly recommended.
- *Performance Acceleration*: Cryptographic operations necessary for encryption/decryption and signature/verification are computationally intensive operations and cannot scale with general purpose processors, specialized cryptographic processors are necessary. With a SOA Gateway managing such operations for many service producers and consumers, using hardware accelerators is required for a scalable architecture that does not add latency to message processing.

Requiring developers to intertwine security policy code with their services is a failing long-term strategy. It puts tremendous maintenance burdens on the service implementation and exposes SOA to security risks as well as interoperability pitfalls. Separating security policies from a service implementation and centralizing such policies using an intermediary SOA Gateway ensures that the security holes are not inadvertently introduced by developers and the SOA deployment can be built up rapidly by decoupling security processing tasks from the producer services.

## IV. BEST PRACTICE: CONTROL AND AUDIT INFORMATION FLOW

*Description:* In a well-built SOA, business information flows effortlessly between applications. The applications are not only internal applications - one of the key benefits of SOA is that external partner applications can also be readily integrated with internal corporate systems for process automation. This ease of information flow facilitated by a SOA puts significant message control and audit requirements on the infrastructure including message filtering and message archiving requirements. Message filtering ensures that nothing malicious enters a corporation through SOAP/XML channels.

Message filtering also prevents "information leak" by ensuring that classified corporate information such as financial and customer information or top secret military/defense information is not stolen or inadvertently sent over outbound SOAP/XML messages. In addition to message filtering, corporations are mandated by regulatory requirements to archive and store business transactions for auditing purposes. Regardless of regulatory mandates, it is a widely instituted best practice to control inbound and outbound messages through filtering as well as archiving selected messages for auditing purposes. As shown in Figure 3 below, a SOA Gateway provides filtering and archiving functionality by intercepting SOAP/XML messages sent between consumers and producers. The SOA Gateway intercepts and filters messages by performing deep-content inspection of the SOAP/XML messages. Selected messages can then be archived to an external RDBMS as a complete message or by extracting parts of the messages.

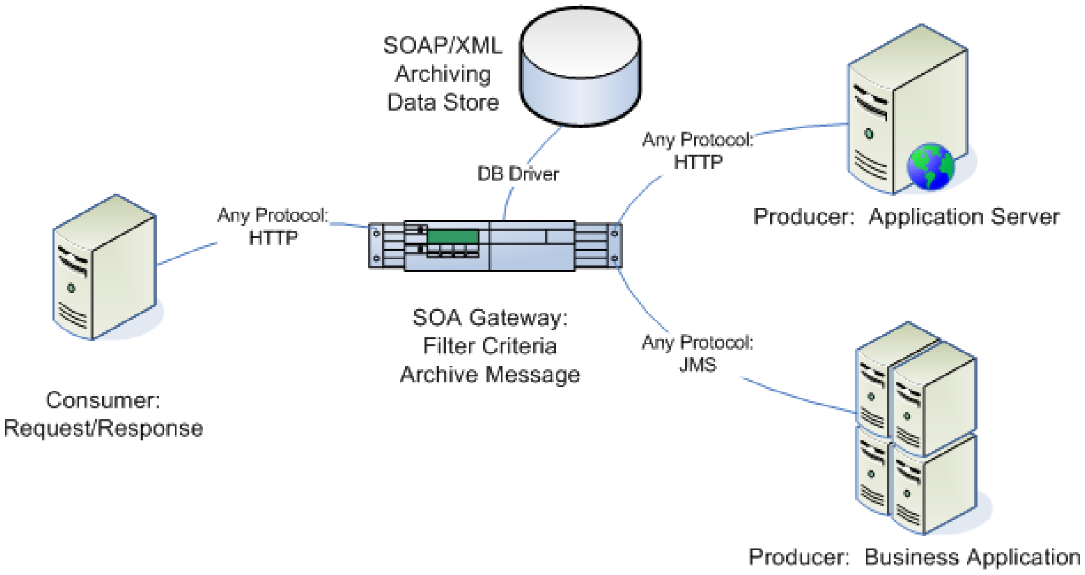

*Figure 3: Enabling information control through filtering and archiving.*

*Benefits*: Enabling information control through a SOA Gateways has significant benefits including:

- *Consistency*: By centralizing message filtering and archiving policies, a consistent set of rules can be enforced across a SOA deployment for allowing or denying messages in and out of a corporate network based on message payload. Messages can also be archived consistently based on business data within a message.
- *Security*: SOA Gateways act as SOAP/XML Firewalls that inspect inbound messages for malware and only let clean messages through. Through deep content inspection, SOA Gateways provide security by ensuring that malicious content is kept out and sensitive information is prevented from leaving corporate domains.
- *Productivity*: Deploying a SOA Gateway removes the burden of building filtering and archiving functionality into consumer and producer applications. With simple, code-free policy authoring for message control and archiving, SOA Gateway administrators can rapidly author policies freeing developers to focus on building business functionality.

*Requirements*: SOA Gateways have to filter, control and archive messages between consumers and producers seamlessly for messages traversing a SOA. For deploying filtering and archiving, a SOA Gateway must meet the following requirements:

- *Bi-directional Message Control*: Both inbound and outbound SOAP/XML messages have to be handled by the SOA Gateway. Typically, inbound messages are checked for malware, validated and then archived to an off-board RDBMS. Outbound messages are usually checked for sensitive information before they are permitted out of the corporate domain.

- *Granular Message Capture*: SOA Gateways should have the ability to capture full messages and store them in an external database and also have the ability to select any element-level information from a SOAP/XML message and store this information in the database as well.
- *Flexible Configuration*: SOA Gateways should provide a high degree of flexibility in configuring policies for filtering and archiving messages. Message should be selected for filtering and archiving on any criteria driven from message content. SOA Gateways should be able to archive messages on arrival, at any stage of processing such as encryption, signature or message enrichment and then archive processed messages again so that multiple snapshots along the message processing may be captured for auditing and analysis. One common best practice is to sign and store all inbound and outbound messages for preserving a detailed message trail.

SOAP/XML messages can contain complex content including binary attachments that should be scanned before allowing the messages to reach corporate back office systems that handle essential business transactions. Similar to email systems that check for message size and content, SOA Gateways should be deployed in a network to ensure that only good messages are permitted in and that sensitive information that is not authorized to leave a corporation are prevented from leaking out of corporate boundaries. Messages should be signed and stored through a robust archiving process for ongoing auditing and data analysis.

## V. CONCLUSIONS

A SOA Gateway is a core infrastructure component of a SOA with the ability to integrate services securely. Typically deployed as a hardware appliance, a SOA Gateway seamlessly controls access to services, protects information through data-level encryption, ensures the integrity of a message through signatures, and controls corporate information flow. A SOA Gateway enables SOAP/XML messaging across a variety of transport protocol such as HTTP, MQ Series, FTP, or JMS with protocol-mixing amongst the protocols. In most deployments, SOA professionals choose to use SOA Gateways as hardware-based network appliances primarily for the following reasons:

- *Ease of Deployment*: SOA Gateways are easy to install in a network as a snap-on appliance. Appliance-based SOA Gateways eliminate the need to install software packages and operating system patches and enable technologist to focus on configuring business policies.
- *Centralized Policy Management*: SOA Gateways provide centralize policy management removing the burden of security policies from service developers. Keeping such policies separate from developers, enables better on-going policy management, security and developer efficiency.
- *Superior Performance*: Hardware-based SOA Gateways process SOAP/XML messages faster than software based solutions. For resource-intensive processing such as message encryption-decryption, signature-verification, filtering, transformation and access control, hardware-based SOA Gateways provide exceptional message throughput for enterprise-class scalability.

SOA Gateways add significant productivity to project teams by providing code-free configuration of policies and by freeing up development teams to focus on building core business functionality instead of developing code for security policies, message control and service aggregation. Enterprises serious about cutting costs and increasing revenue through service reuse and rapid system-to-system integration must deploy a robust, secure and flexible SOA Gateway for efficient message processing.