Migration and Security in SOA

Abstract

It is estimated that 90% of most external attacks on applications take advantage of known vulnerabilities and misconfigured systems. While it is unlikely that one could develop 100% secure applications, it may be advisable to analyze vulnerabilities, threats & risks and implement robust security and access-control mechanisms to tackle some of the known, anticipated security threats specifically in SOA initiatives that are increasingly beginning to expose the once secure 'legacy functionality'. Such an approach would not only improve overall system security but also lead to reduced costs (incident response costs, application outage costs, cost of fixing, reputation damage costs, etc.) through increased efficiency and better customer satisfaction levels.

This white paper provides comprehensive guidance on integrating security and access-control best practices into your SOA and WSOA initiatives. It includes the review of topics such as:

- The different Access control models
- A meta-model for WSOA (Web service-oriented architecture)
- Goals of SOA Security
- SOA Security implementation models
- Industry standards for SOA Security and
- SOII (Service-Oriented Information Integration), standards for SOII

Table of Contents

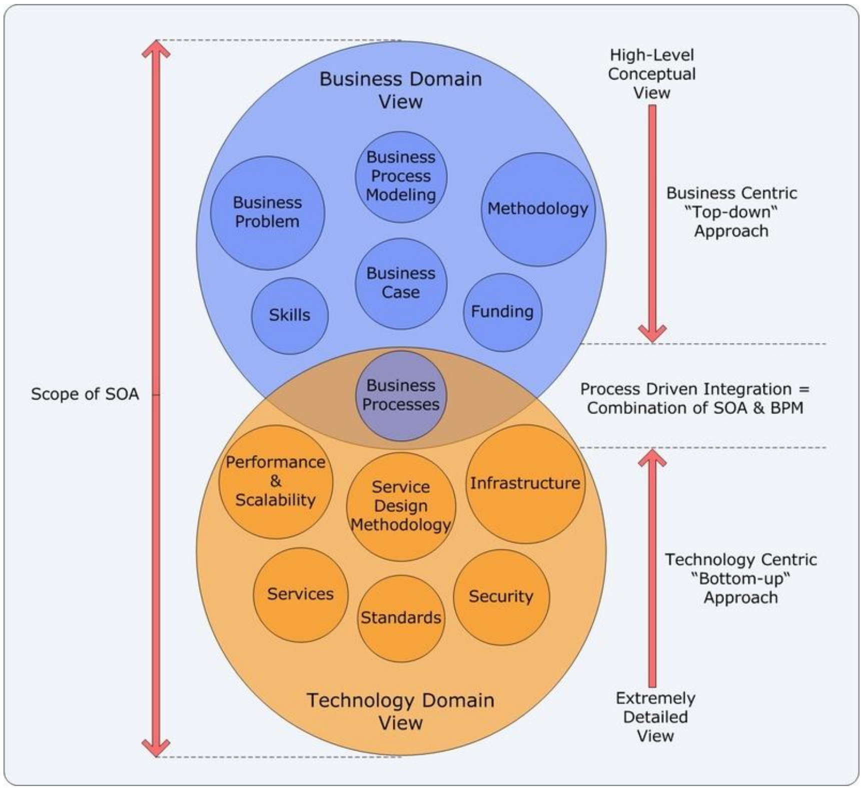
1. Migration

In this section, we briefly describe why enterprises may choose to migrate to newer systems and how SOA and WSOA could help in such migration. Two approaches are discussed along with high level descriptions for SOA. Finally, we discuss some questions that enterprises may need answers to, prior to embarking on migration to SOA or more specifically WSOA.

Migrate to what?

Currently, we find reuse driven migration / modernization finding greater acceptance by enterprises world-wide, as it offers several alternatives for modernizing legacy applications. Web service-oriented architecture (WSOA) is probably the best candidate for most enterprises to align their business processes with the supporting IT for migration. It is an accepted fact today that Web service technologies provide a promising way of implementing service-oriented architecture (SOA).

Normally, WSOA is not built from scratch but the functionality of existing systems and their components are leveraged using web services. There are two approaches to do so.


Business Domain View
Business Problem
Business Process Modeling
Methodology
Business Case
Skills
Funding
Business Processes
Performance & Scalability
Service Design Methodology
Infrastructure
Services
Standards
Security
Technology Domain View
Scope of SOA
High-Level Conceptual View
Business Centric "Top-down" Approach
Process Driven Integration = Combination of SOA & BPM
Technology Centric "Bottom-up" Approach
Extremely Detailed View

Bottom up approach

Bottom up approach starts from the existing software systems and eases conventional application integration. In an integration process, the composition of web services of heterogeneous core software systems is done using process execution languages like Business Process Execution Language (BPEL). This approach is useful in scenarios where an organization envisages reusing functionality provided by legacy systems.

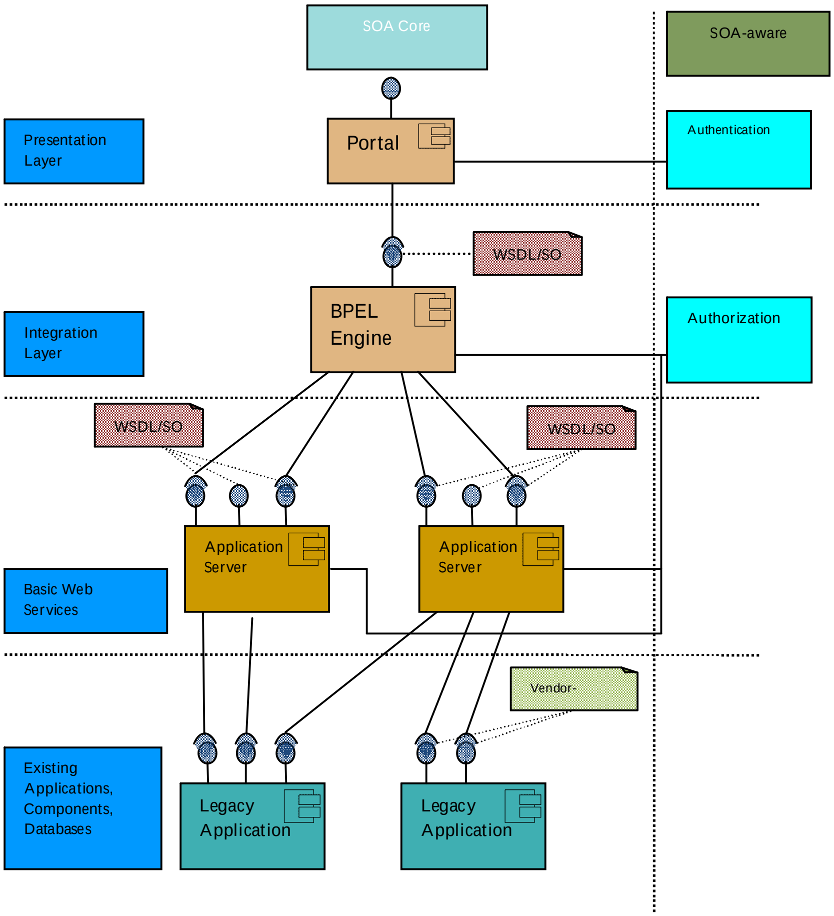
Top down approach

The Top down approach begins with business processes and converges at model-driven mapping downwards to basic and composite web services. They facilitate business analysts to perform the so-called programming-in-the-large, the system-independent orchestration of business-related (web) services along business processes.

Web Service-Oriented Architecture

The basic WSOA layers are shown in the diagram below which describes an integrated framework of web services as a web portal facilitating single sign on among various web services. In the bottom-most layer, existing applications can be seen to be wrapped into (basic) web services. Application servers do the wrapping by applying design patterns: proxy or façade [Erich 1998]. Web services are composed at the integration layer using BPEL; web portals are used for integration of the human users utilizing existing web technology such as web browsers. Closer alignment of business processes with their supporting IT is a key driver for WSOA. Further layers are introduced on top of the existing applications so that the focus on IT changes from the (internal) view of systems and applications towards operated and quality assured IT services. Standardized interfaces are used to define these services in the basic web services and the integration layer. The standardized interfaces facilitate the traditional integration process, particularly in heterogeneous environments. In addition, the standardized interfaces also allow flexible service reuse in various business processes.

SOA is not strictly layered in the web service context and one or more intermediaries, like BPEL engines, enable access to web services; web services can also be accessed directly from the presentation layer in the form of portlets via the web portal. As shown in the figure, a BPEL composition of web services and a basic web service are accessed using WSDL/SOAP interfaces.



How Secure are WSOAs?

The previous section introduces the core of WSOA, but there would possibly remain some fundamental questions about securing WSOA across enterprises of which access control is an important issue as it facilitates ensuring that a user only has access to the resources necessary to perform its respective task. These questions include but are not limited to:

- How can access control be handled in a highly distributed and service-oriented environment?
- How can a policy decision point (PDP) be defined when accessing between existing applications to business services?
- How can access control mechanisms cope with internal Identity management (IdM) structures of different legacy systems?
- How can the alignment be achieved for different IdM access control mechanisms inside applications?

These and other aspects of security are addressed in the sections below.

2. Web Service & Access Control

This section answers some of the questions on security by describing an access control Meta-model for web-service oriented architecture. It may be useful to briefly visit the basics of web services and composite web services before detailing the access control model.

Web service

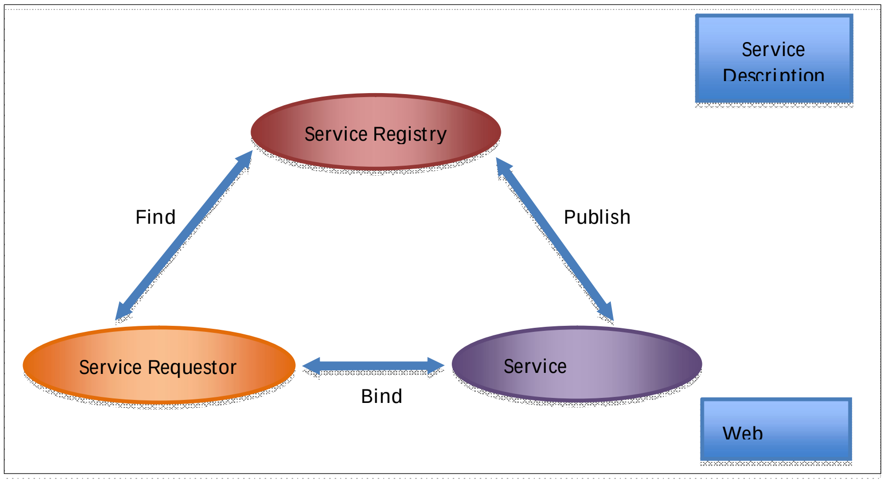
A Web service is a distributed software component accessible through an application interface that provides information and functionality to an application rather than to a human user.

OR

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

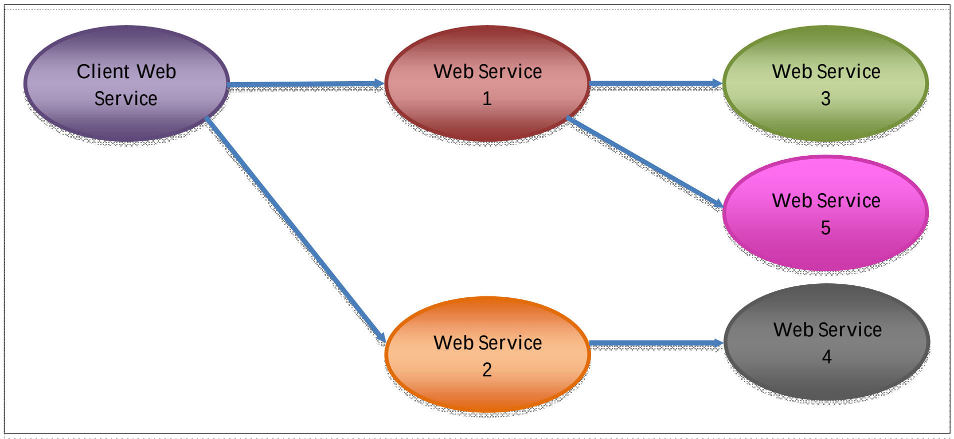
Web service description and definition

A service is defined and advertised by its provider for service users/consumers. A service description uses the Web Service Description Language (WSDL) and this definition is published to a directory of services. A potential service consumer queries the directory service to discover a service that meets its needs.



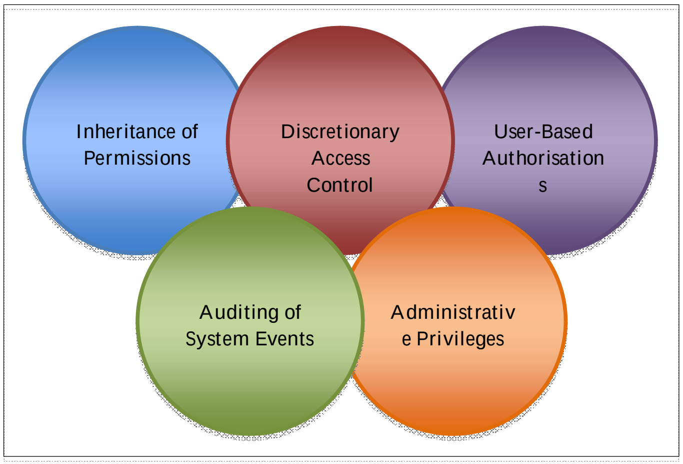
Why a Composite Web Service?

Web services are distributed software systems that support interoperable machine-to-machine interactions over a network. WSDL describes their functionalities that can then be invoked by other systems through message-based interactions. However, in certain scenarios, a desired functional or non-functional requirement may not be met by a single Web service. But appropriately integrating and composing a set of available services may possibly fulfill the requirements which emphasize the need for a composite web service.



Access Control Models

In typical access control models, the "action" is always reduced to basic system operations like read, write, delete, execute etc. In WSOA however, the most atomic object to restrict access to is a web service operation which stimulates a functionality provided by the web service.


Inheritance of Permissions
Discretionary Access Control
User-Based Authorisation s
Auditing of System Events
Administrativ e Privileges

In WSOA, explicit service composition takes place when a web service calls other different web service operations and returns a combined result (a service may have to access information from different sources and it may not be achieved by a single service). The access restriction to the composed service is at least the sum of the restrictions of all underlying operations it is composed of and that are invoked mandatorily. This allows checking authorization at an earlier stage i.e. the BPEL-composed web service, thereby limiting unnecessary calls ending in rollback operations if particular permissions for invoked basic web service operations are missing.

Access Control Meta-model for web-service oriented architecture

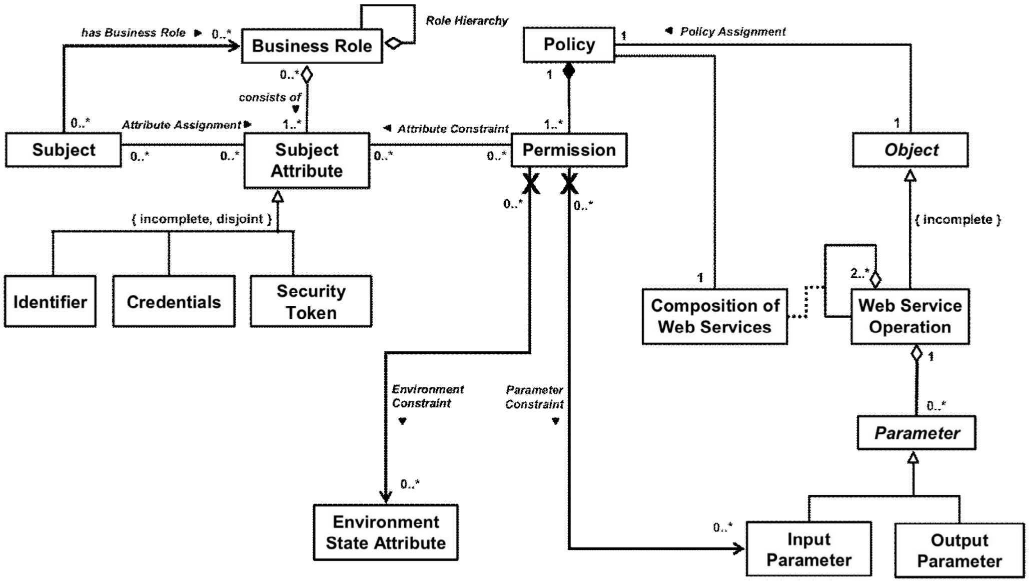
A metamodel for access control in web service-oriented architecture (WSOA) is shown in the diagram below. It is based on a conceptual model in the UML 2.0 metamodeling approach to define the sets and relations used to enforce access control. This is an enhancement of the combination of hierarchical Role Based Access Control (RBAC) and Attribute Based Access Control (ABAC).

Policy is the central element of this model. A policy is the composition of ***permissions***. The traditional Subject / Object relation of a single service usage context can be defined with a permission. In this meta-model, denial of access is the default behavior and it has to be modeled explicitly when a permission is granted.

One Object (related via the Policy towards which it is aggregated) and a set of Subject Attributes can be combined using permission with the possibility to have constraints considering the Object's associated Input Parameters and the Environment State (like date, time or any other attribute related to neither Subject nor Object). There are some special Subject Attributes that are explicitly modeled as the subject's Identifier, the Credentials and a Security Token (which is of temporary validity, i.e. refers to a session context).

In WSOA, Subjects can either be human users or self-acting services. In reality, it is possible to have a 1:n relation between a human user and a Subject (i.e. a user having more than one identity). This is not explicitly modeled, as it is not relevant for the definition of access control. Subjects are characterized by a defined number of Subject Attributes. Therefore, users having multiple identities will also have different and independent subjects.

Subjects in a business perspective act in the context of a Business Role. In this model, a defined amount of finer-grained Subject Attributes give us the concept of a Business Role. Similarly, Role Hierarchies can be defined as well, together finally mapping to a set of Subject Attributes.



SOA is the reuse of existing services

To achieve the goal of SOA, to reuse the existing services in different contexts, the concept of Permission is used i.e.

- Each permission covers one service usage context.
- The Policy is the composition of all permissions of an object using boolean “OR” concatenation.
- Input Parameters of the Web Service Operation are placed into focus.

Two relations form the Permission towards the Object

1. A direct one towards the input parameter (not backwards navigable) following the idea that a parameter does not need to know if its value is evaluated for access control
2. An indirect one via the Policy

Service composition is one goal of service-oriented architecture

A Composition of Web Services consists of multiple invocations of other Web Service Operations in a specific order. It has a web service interface like the basic web services consisting of operations. It cannot be determined if the service interface is a composite or a basic web service. But service composition relates to access control in respect that there should be the possibility to pre-verify authorization at the layer of composite web services to be able to stop execution in case of missing authorization at an early stage. The composition aspect, modeled as an association class of the aggregation of Web Service Operations in figure 2, enables the authorization verification even at a composition level. For all Web Service Operations that are obligatorily invoked by the Composition, their Policies have to be added to the (overall) Policy of the Composition of Web Service using Boolean “AND” concatenation.

3. Security & Access Control

Considering the "big-picture" of SOA Security, it may be important to understand different aspects of security, role of AAA (Authentication, Authorization and Auditing) in SOA Security, encryption, industry standard specifications etc. This section introduces the key goals of SOA security, nature of security threats and finally some practical SOA security implementations in the industry. Considering web service is the most widely accepted approach to implement SOA, the technical aspects in this section focus more on web service security.

Goals of SOA Security

As the adoption of SOA increases, the boundary of services "grows" beyond internal applications. To achieve true re-usability, it may be required for organizations to expose services to third-parties, partners or even end-customers over insecure networks such as the Internet. Services are organizational assets and exposing them without appropriate security considerations poses a big threat to the organization in the form of un-authorized access, misuse of services, over-use of services and also hacker vulnerability.

To address the above risks, the main goals of SOA security are:

- Authentication – Allowing access only to the intended application that invokes the service. In traditional security approaches, this is the standard "credential" based security such as a login ID/password pair, certificates etc.

- Authorization – Controlling access only to a defined set of services and/or operations within a service. This is the classical "role" based security to restrict access to a subset of functions

- Auditing/Monitoring – Recording all invocations of a service to address the 5 W's of security – Who, What, When, Where & Why. This is crucial to identify an attack and trace the attacker. Also, auditing constitutes a "digital" record of all activities within the SOA infrastructure

- Federation – When a service requires authentication against another external system, federation is used. Federation is an extension of authentication that helps the service provider to establish trust between the provider's security domain and an external domain. So the external provider "trusts" the request and considers it authenticated without expecting an additional credential.

- Integrity – Goal of integrity checking is to ensure that the XML data entering in the form of a web service request is not corrupted

- Policy – The idea of a policy in SOA security is the capability of the service provider to specify web service's conditions under which the service is provided. For example, the condition may require that the request to a web service be encrypted.

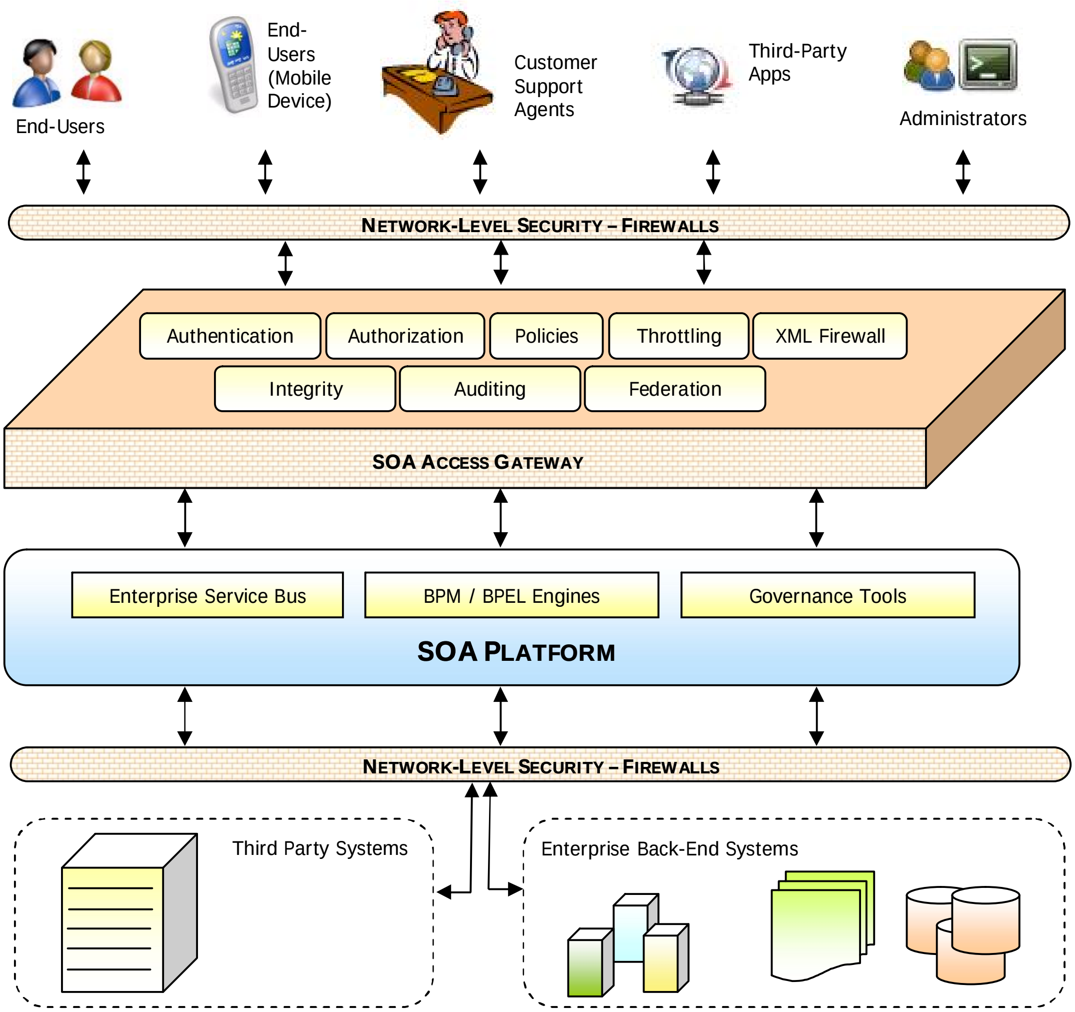
- Throttling – It is a concept to control the "bandwidth" offered by a service. Though not directly related to security, throttling is typically used to protect the service infrastructure so that service consumers do not "over-use" the services. In some cases, this can also be used to prevent "denial of service" attacks.

- Confidentiality / Network Level Security – The goal of network level security is to encrypt data packets transmitted to and from the SOA infrastructure. This is to prevent any packet-sniffing tools to intercept any passwords.

- Hack-proof – Even if a genuine service consumer successfully authenticates and has necessary role permissions on a service, it is very important to ensure that service boundaries are not crossed to prevent several web-service specific attacks such as XPath injection, XML structure manipulation, schema attacks, etc

SOA Security Implementation – A Logical View

The diagram below depicts the building blocks of a typical SOA security implementation:



The SOA Access Gateway is a critical component that enforces security to the SOA platform. In addition to the standard firewall based security, the SOA Access Gateway can specifically interpret Web Service requests. The firewall is used to allow "port" and "IP-Level" access. Like a Firewall, the SOA Access Gateway is a hardware box that directly fits into the IP network. However, the SOA Access Gateway is a step ahead and provides the following functions:

- Authentication – In the form of WS-Security tokens
- Authorization – Acts as a policy enforcement point (PEP) and policy definition point (PDP)
- Auditing – Captures usage statistics

- Throttling – Allows restricting bandwidth for a particular service. Example – Service A can be invoked only at 80 transactions per second, whereas Service B can be invoked at 100 transactions per second
- Encryption / Decryption and Integrity checks
- XML Firewall – Detects all types of XML related threats such as XPath injection, Schema attacks, etc
- Supports all security standards – WS-Security, SAML, WS-Federation, WS-Policy, WS-Metadata, etc

Industry Standards for Security

Industry standards help vendors and organizations follow a common approach such that solutions can be re-used reducing time, effort and investment and prevents re-inventing the wheel.

SOA Security Goal	Standards	Overview
Authentication	WS-Security	Originally drafted by IBM, Microsoft and VeriSign, WS-Security defines a standard way of specifying username and encrypted password in SOAP headers.
	WS-Trust	WS-Trust aims to enhance WS-Security by providing additional features such as a Security Token Service (STS). STS offers services such as Token Exchange, Token Issuance and Validation. This standard is approved by OASIS
	WS-Secure Conversation	WS-SecureConversation is another extension to WS-Security which defines the means to create a security context and allows a series of message exchanges (conversation) to be done when authentication
Authorization	XACML	XACML (eXtensible Access Control Markup Language) is an XML schema specification to define authorization and entitlement policies. XACML addresses the lack of fine-grained access control granularity in SAML
Federation	SAML	SAML (Security Assertion Markup Language) is primarily an XML-based standard authentication language to authenticate across different security domains, such as SSO – Single Sign-On.

Policy	WS-Policy	WS-Policy is a standard way for service providers to specify a wide range of service requirements (policies) such as maximum message size, service traffic handling capacity, etc. This standard is approved by OASIS
	WS-SecurityPolicy	WS-SecurityPolicy standard defines security related policies based on WS-Policy and WS-Secure Conversation standard. This standard is approved by OASIS
	WS-Metadata Exchange	WS-MetadataExchange specification defines a mechanism for service clients to retrieve service metadata information such as Schema, WSDL and WS-Policy.
Encryption / Confidentiality	XML-Encrypt	XML-Encrypt is a W3C recommendation to encrypt sensitive fields within XML documents and also to specify the encryption algorithm that is used
	XML-Signature (also known as XML-DSig)	XML-Signature is also a W3C recommendation for XML digital signature processing to allow clients to digitally sign an XML. This ensures message integrity, which allows service providers to detect content corruption, malicious content, etc. Advanced versions of XML-DSig already exists such as XAdES (XML Advanced Electronic Signatures)
	XKMS	XML Key Management System is a W3C recommendation which allows developers to secure communications using public key infrastructure (PKI). The specification describes protocols for distributing and registering public keys to be used in conjunction with XML-Encrypt and XML-Signature. XKMS consists of two parts – XKISS – XML Key Information Specification & XKRSS – XML Key Registration Service Specification
	SSL	Needless to say, Secure Sockets Layer is the basic foundation technology to ensure transport-level security.

SOA Security Product Vendors

There are several vendors in the industry who provide SOA security solutions that help organizations realize the security goals in the form of hardware and software. The following table lists the key vendors:

Sno	Product Name	Vendor
1	IBM Datapower (Access Gateway - Hardware)	IBM
2	Cisco ACE XML Gateway (Hardware)	Cisco
3	Intel XML Security Gateway (Hardware)	Intel
4	Web Services - Domain Boundary Controller (Hardware)	Xtradyne
5	Amberpoint SOA Management system (Software)	Amberpoint

4. Service Oriented Information Integration

This section aims to present an overview of Service Oriented Information Integration (SOII), technical considerations, SOII industry standards and a list of off-the-shelf products available in the market related to SOII.

Before giving an overview of SOII, it is important to understand the basic principles of Enterprise Information Integration (EII). EII is a process of providing a uniform interface for viewing all the data within an enterprise. By providing a single interface, it makes it possible for different departments in an enterprise to view data from different heterogeneous sources, eventually to achieve "integration" of information. Data exists in both structured and un-structured form in different formats such as excel files, RDBMS, XML and even in the form of text dumps such as comma-separated files. APIs such as ODBC, JDBC & tools like ETL (Extract Transform and Load) already exist and are predominantly used to integrate disparate types of data sources, which facilitate enterprises to implement EII. ETL deals with transferring batches of information from one system to another, EII aims to provide real-time views across multiple data sources.

To define SOII in simple terms, it is EII followed in a 'Service-Oriented' approach. The use of SOA – Service Oriented Architecture to solve EII problems is the fundamental principle of SOII. SOII enforces 'service' as the unified interface for access to all enterprise data.

Why SOII?

There are several business and technical benefits of using SOII. The top 3 benefits are listed below:

Aligned to SOA

SOII automatically inherits the benefits of Service Oriented Architecture. SOII helps organizations move away from point-to-point integration and makes information available as a repository of services on the network - in SOA terms, the Enterprise Service Bus. This enables organizations to reuse existing functionality for building new composite applications. Developers of these services publish information about them in SOA Service Registries and Repositories so that the service consumers can easily lookup and find them as and when required.

Standards Based

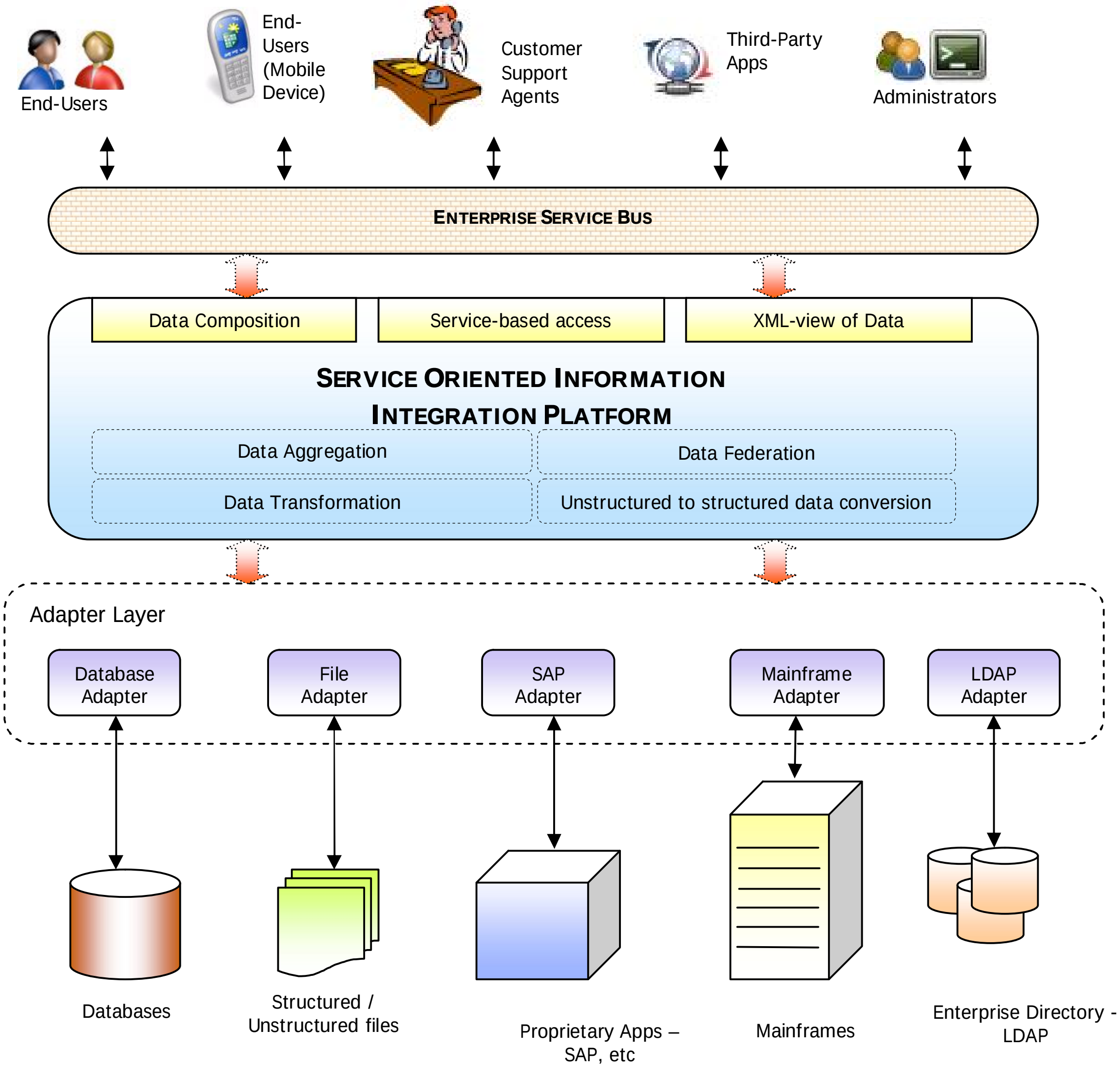
Several industry standards have evolved over Service Oriented Information Integration such as Service Data Objects, Service Connector Architecture, etc. Using proprietary APIs would mandate that users learn the details of a specific vendor's platform. Tools that are standards based help lower the cost of integration, increases familiarity within the developer community and prevents "vendor-lockin".

Tools Availability

The challenge of complex transformation and integration of structured and un-structured data requires specialized tools. The challenges in data transformation include handling the un-structured data such as text, PDF, un-structured excel files, etc. For SOII, Web Services are the ideal way to implement data-oriented services. Web Services use XML, when combined with technologies like XSLT and X-Query, makes it is an ideal choice to deal with representation and transformation of structured & unstructured data.

SOII – A Logical View

The following diagram shows a logical view of different functional blocks of SOII



Functions such as data aggregation, data federation and unstructured to structured transformation are functional features of any EII platform, however, the key difference between an EII platform and SOII platform is the ability to expose these data interfaces as web services & ability to provide composed services out of atomic services. Typically, any SOII product would also provide a wide range of adapters to establish connectivity to different data sources. Typically, each adapter is offered as a separate product that organizations can choose to buy based on their needs.

Industry Standards

Service Data Objects (SDO) specification allows applications to uniformly access and manipulate data from heterogeneous data sources, including relational databases, XML data sources, Web services and enterprise information systems. SDO was originally developed as a joint collaboration between BEA and IBM and is now being developed by BEA, IBM, Oracle, SAP, Siebel, Sybase and XCalia. SDO is based on the concept of disconnected data graphs, wherein, a client retrieves a data graph from a data source, transforms the data graph, and can then apply the data graph changes back to the data source. Technically, SDO can be used in conjunction with JDO (Java Data Objects) where JDO is a data source that SDO can access. SDO is part of the Java Community Process – JSR 235

SOII Products

The following table shows the top three products that specifically cater to the SOII space.

Sno	Product Name	Vendor
1	IBM DB2 Information Integrator	IBM
2	Aqualogic Data Services Platform (ALDSP)	BEA (now Oracle)
3	XA Suite	XAware Solutions

5. Conclusion

The intention of the paper is to outline a number of recommendations that could ensure successful integration of some of the security best practices into your SOA, WSOA initiatives. The following summarizes some of the key recommendations:

- Analyze and Capture security and access-control requirements (system and functional) and conduct a requirements mapping activity. Understand your security goals for SOA and accordingly devise a plan for implementation.

- It may be useful to work with a base implementation framework as indicated in the logical view diagram, as this is adopted from a real-world industrial scale implementation

- Assess the fitment of the access-control meta model for your WSOA needs

- Evaluation of the tools available in the market and following industry standards could greatly reduce the overall roll-out time while also providing a simpler approach to implementation

Torry Harris Business Solutions Inc, a US based services provider with a large base of technologists located in the UK, India and China has provided cost effective solutions at a design, development and support level to a variety of enterprise clients across the world since 1998. The company specializes in integration, distributed computing, and its focus on SOA is a result of nearly a decade of expertise gathered in the middleware space. The company has partnerships with almost all the leading SOA and integration product vendors. SOA, involving the creation of autonomous parts of a solution, lends itself admirably to the cost effective model of offshore service collaboration. A separate white paper entitled "SOA Implementation with an offshore partner" available for download, explores this model in a more detailed manner.

Further information about the company and a variety of white papers on SOA are available at www.thbs.com/soa.

For more information, write to us at soa@thbs.com.

Distributed Systems & Services Group, University of Leeds

The group unites two central research themes within the mainstream of Computer Science - architecture and systems, each linked through a common objective: to support the needs of the next generation of distributed/Internet computing. Grid Computing is one of such examples that enables advanced e-Science and e-Business applications, distinguished from conventional distributed computing by its focus on large-scale, dynamical interactions and resource sharing across different virtual organisations.

Further information about the group is available at www.comp.leeds.ac.uk/distsys

6. Reference

1. WS-Security: http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec
2. WS-Trust: http://docs.oasis-open.org/ws-sx/ws-trust/v1.3/ws-trust.html
3. WS-SecureConversation:http://docs.oasis-open.org/ws-sx/ws-secureconversation/v1.3/ws-secureconversation.pdf
4. XACML: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
5. SAML: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
6. WS-Policy: http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf
7. WS-SecurityPolicy: http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.2/ws-securitypolicy.html
8. WS-MetadataExchange: http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf
9. XML-Encrypt : http://www.w3.org/TR/xmlenc-core/
10. XML-Signature: http://www.w3.org/TR/xmldsig-core/
11. XKMS : http://www.w3.org/TR/xkms/
12. Erich Gamma, Richard Helm, Ralph Johnson, Jhon Vlissides: “Design Patterns”, Addison Wesley, 1998
13. SDO Specification JSR - http://jcp.org/en/jsr/detail?id=235
14. Patterns for Information Integration from IBM - http://www.redbooks.ibm.com/abstracts/SG247101.html
15. IBM’s SDO Page: http://www.ibm.com/developerworks/library/specification/ws-sdo/