Condiciones necesarias para la Auditoría en tecnologías derivadas de Web Services e Identidad Federada

Lic. Francisco Javier Díaz- Mg. Lía Molinari
Calle 50 y 115 – 1er Piso – Edificio Bosque Oeste
L.I.N.T.I.. - Facultad de Informática – U.N.L.P.

Palabras claves
Sistemas distribuidos - Web Services – Seguridad – Auditoría - SOAP- SAML – XML Encription – Identidad Federada – Single Sign on – Sarbanes-Oxley

Resumen
Ante la necesidad de adaptarse a estándares y respetar el stack de protocolos que recomiendan organizaciones como W3C, OASIS o WS-I, surgen un conjunto de alternativas abiertas para la implementación de transacciones en Web services (de ahora en más, WS) que deban superar varios "saltos", es decir, nodos intermedios, resguardando la confidencialidad y permitiendo la auditoría.
La solución de seguridad debe garantizar la interoperabilidad, apto para semánticas de web services e identidad federada (Federated Identity, de ahora en más, FI) que permita la trazabilidad de los procesos que incluyen varios "saltos".
Las nuevas tecnologías en WS involucran más que SSL y firewalls. Si bien hay un importante avance en la implementación de seguridad más allá de la capa de red o transporte, **vale analizar si: a) estas soluciones garantizan la interoperabilidad, b) proveen un esquema de seguridad para permitir WS Composition, c) Permiten implementar auditoría para WS y FI**.
Este último punto toma trascendental importancia considerando la exigencia regulatoria que imponen leyes actuales, entre ellas, Sarbanes –Oxley, en cuanto a la obligación de contar con pistas de auditoría para el control antifraude de sus procesos.

Workshop: **VI WORKSHOP DE PROCESAMIENTO DISTRIBUIDO Y PARALELO (WPDP)**

Introducción. Web services.

Web Services (WS) es una tecnología ya instalada en el mundo IT[1] [13].
Según W3C, *"Web Service es un sistema de software identificado por una URI, cuyas interfases y enlaces (bindings) son definidos y descriptos usando XML. Este sistema puede ser "descubierto" y usado por otros sistemas de software. Esa interacción debe darse según la manera descripta en la definición, usando mensajes de acuerdo a protocolos de Internet"*.

En esta definición no se nombra SOAP [11] ni WSDL [2]. Otros mecanismos pueden usarse para empaquetar XML y otra forma de descripción, como DAML-S, en vez de WSDL.
No obstante, WSA (Web Service Architecture, de W3C) acuerda lo conveniente de basarse en una base común para messaging y descripción (lo define como "una necesidad práctica").

[1] Se recomienda también consultar http://www.w3c.org, http://www.ws-i.org, http://www.webservices.org, http://www.oasis-open.org.

Los beneficios de utilizar WS se manifiestan en:

- Usa estándares abiertos, basados en texto. Se pueden comunicar componentes escritos en diferentes lenguajes y distintas plataformas.
- Fácil de implementar. No es costoso: se usa una infraestructura existente.
- La mayoría de las aplicaciones pueden re-empaquetarse como Web Service.

Berners-Lee en [3] detalla la arquitectura de software para la implementación de web services seguros:

<table>
<tr><td colspan="6">Management</td></tr>
<tr><td colspan="3">Seguridad</td><td colspan="3">Semánticas de aplicación</td></tr>
<tr><td rowspan="2">Firma</td><td rowspan="2">Manejo de claves</td><td rowspan="2">Encriptación</td><td>Transacciones</td><td>Coreografía</td><td>Scripting</td></tr>
<tr><td colspan="2">WSDL</td><td>XPath</td></tr>
<tr><td colspan="4">SOAP 1.2</td><td>XML Schema</td></tr>
<tr><td colspan="2">HTTP 1.1</td><td colspan="2">XML</td><td>Namespaces</td></tr>
<tr><td colspan="3">URI</td><td colspan="3">Unicode</td></tr>
</table>

Haciendo un análisis de este esquema, la seguridad como concepto global está a la altura de las semánticas de aplicación e incluye firma, manejo de claves y encriptación.
Qué queremos asegurar? Obviamente, se quiere proteger la información que viaja en los mensajes que se generan en una transacción WS, a través de autenticación, autorización, integridad, disponibilidad...
La alternativa de seguridad que elijamos debe cumplir con estas características, garantizando la interoperabilidad.
La seguridad en WS ha requerido una minuciosa discusión que se ha definido en cierto punto en el documento Basic Security Profile versión 1.0 [1].
Pero además de cumplir con las exigencias de autenticación-autorización, la tendencia cada vez mayor de obtener pistas de auditorías nos hace analizar un modelo que las provea a través del registro de las transacciones/sesiones que se realicen y los objetos involucrados.
Autenticación, Autorización y Auditoría es lo que se ha dado en llamar **las 3 A**.
En Estados Unidos hay distintas leyes que obligan poner mayor atención a la privacidad de los clientes, a la confiabilidad de la información y a la seguridad.
La Ley Sarbanes-Oxley, la Ley Gramm-Leach-Bliley, el Acuerdo de Basilea II, la Ley de Responsabilidad y Transferencia del Seguro Médico (HIPAA), exigen a las empresas desde la certificación antifraude de los estados financieros a la protección de información personal de pacientes.
La Directiva de Protección de Datos de la Unión Europea requiere que todas las naciones miembro aprueben la legislación que exige controles de confidencialidad e integridad de las redes, los sistemas y los datos que contengan información personal (tanto de empleados como clientes).
Tomemos por ejemplo, la Ley Sarbannes-Oxley que se centra en todo lo relacionado con la creación, documentación, control y comunicación de procedimientos. Obviamente no se puede esperar que la tecnología escogida para seguridad cumpla con todos estos requisitos. Pero sí debe considerar que se registre toda la información referente a las transacciones que se realicen, para que luego, a través de un modelizador de procesos de negocio, puedan obtenerse los productos resultantes para el cumplimiento de la Ley.

Cuando un WS invoca a otro WS se suceden un conjunto de relaciones entre nodos de la Red que involucran varios saltos entre los extremos a comunicarse. Y la tecnología que se utilice debe garantizar que las empresas puedan cumplir con las exigencias de auditoría planteadas.

Tecnologías derivadas de WS

Este documento se inicia con la definición de W3C sobre WS. Las tecnologías derivadas de WS (cual es el caso de WS Composition[2] o Web Semantics [8]) han ido haciendo cada vez más complejos los escenarios.
Cuando las aplicaciones son preparadas para invocar WS, el stack recomendado y una lógica adecuada son aptas para lograr interoperabilidad entre componentes. Pero ante la necesidad de componer WS o implementar web semantics, se debe analizar si estos protocolos o sus nuevas versiones ofrecen la funcionalidad que se desea implementar.
Cuando se plantean políticas de seguridad, muchas veces se analizan alternativas que son las adecuadas para la capa de transporte y de red.
Cuando el objetivo es la implementación es WS las políticas a adoptar no deben descuidar asegurar la interoperabilidad incluyendo autenticación, autorización y auditoría (lo que en la bibliografía llaman las 3 A) para controlar la composición de servicios de servicios que se hace necesaria.
Consideramos que en tecnologías derivadas que involucran varios "saltos" entre el end-to-end sumando la exigencia de las 3 A, el esquema de seguridad elegido a)debe ir más allá de la capa de transporte y de red, b)debe ser tolerante a fallas, c) debe adaptarse ágilmente a las nuevas tecnologías de comunicación, d) debe mantener la trazabilidad de los procesos a lo largo de transacciones o sesiones.

Identidad Federada

Para evitar, o por lo menos reducir, los inconvenientes de seguridad en los ambientes distribuidos, el tráfico debería tener alguna forma de identidad asociada con la fuente de emisión.
El concepto de identidad federada se basa en que todos los usuarios de una red deben estar autentificados. Y lo que no esté asociado con una comunidad miembro, es rechazado.
Se podría incluso etiquetar tráfico como anónimo, pero porque el usuario ha sido autentificado para utilizar el servicio anónimo.
La identidad es un conjunto de atributos que describen el perfil de un usuario, una organización, o una entidad de software.
La existencia de distintas identidades y "logins" para acceso, y las aplicaciones basadas en políticas no escalables (o la ausencia de políticas), restringen seriamente la interacción entre aplicaciones y servicios.
Para la implementación de arquitecturas que manejen identidad hay una alternativa centralizada y otra federada.
En la centralizada, un operador permite autenticación y autorización manteniendo la información de identidad, En la federada, la información de identidad es distribuida entre comunidades.

[2] Se define ws Composition cuando un conjunto de WS relacionados son expuestos como un gran WS, lo que exige que haya una cadena de invocaciones a WS que pueden estar en diferentes servidores.

La opción centralizada adolece de los problemas comunes de este modelo: hay un punto único de falla o ataque que inhabilita el esquema, y el tráfico hacia el nodo central. Pero a ello se le suma el peligro que representa que un sólo lugar tenga toda la información (desde el punto de vista de la comunidad de negocios, bancos por ejemplo, es difícil de aceptar...).
El objetivo es que cada organización pueda mantener información propia mientras comparte datos de identidad con sus "socios", de acuerdo a sus objetivos.
Actualmente hay dos plataformas importantes de Identidad Federada: la Shibboleth[3] y el Proyecto Liberty[4]. Ambos usan SAML [7] y SOAP. Uno representa el mundo de la educación y el otro, el de los negocios. Pero... pueden interoperar? [12]. El uso de estos protocolos por sí mismo veremos que no garantizan la interoperabilidad.
En la tabla que se muestra a continuación se comparan algunos ítems en ambas plataformas.

Shibboleth	**Liberty Alliance**
Uso educacional	Uso comercial
Aplicaciones basadas en browser	Aplicaciones basadas en browser
Single sign on	Single sign on
Es una iniciativa Internet2 MACE	Es una iniciativa de un consorcio comercial
Usa SAML 1.1	Usa SAML 1.1
Usa SOAP	Usa SOAP
Disponibilidad de código y especificaciones	Disponibilidad de especificaciones
El proveedor de identidad es una universidad	El proveedor de identidad es una entidad comercial
Pocas federaciones	Muchas federaciones

Shibboleth usa el concepto de club, donde dos partes se ponen de acuerdo para intercambiar, analizar y habilitar según las assertions de cada una. Cada club debe establecer un conjunto de políticas lo que limita la cantidad de miembros, dificultando la escalabilidad.
Shibboleth y Liberty han elegido diferentes schemas. No obstante, al ser públicos, los desarrolladores pueden adaptar las soluciones.
En otro punto que hay una diferencia importante es en la metodología de liberación de información privada. En Shibboleth cuando un usuario se quiere comunicar con un sitio se solicitan a la Autoridad Atributiva un conjunto de atributos del cliente. Se analiza con el proveedor de identidad si el cliente se comunicó antes con el vendedor y si no es así, se le pasa un mensaje al cliente para validar la solicitud y de esta forma, el cliente analiza si quiere compartir con el nuevo sitio parte o todos sus datos de identidad. Por ejemplo, para usar la biblioteca, el único dato que necesita el sitio para otorgar el acceso es si el usuario es miembro de la comunidad, pero puede no ser necesaria la identidad.
En Liberty se definen líneas guías entre las partes que son documentos legales. EL cliente puede definir qué cuentas quiere tener enlazadas, pero no combinar elementos como puede hacerse en Shibboleth.
Por último, si bien ambas usan SOAP y SAML, hay esperanzas en algunos miembros que los problemas de interoperabilidad entre ambas arquitecturas puedan solucionarse en SAML 2.0.

[3] http://shibboleth.internet2.edu

[4] http://www.projectliberty.org

Con respecto a auditoría, el esquema federado garantiza que usuarios y recursos están autenticados y autorizado, lo cual es imprescindible para cumplir con la transparencia de procedimientos deseada.

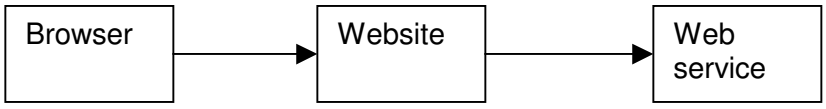
Análisis de los esquemas de seguridad tradicionales. WS y SSL

SSL es una solución ampliamente utilizada entre dos entidades, pero se deben analizar otras soluciones cuando hay una sucesión de entidades, donde cada uno puede ver y modificar el mensaje. Desencriptar y volver a encriptar en cada nodo habilita durante un lapso la visualización y posible modificación del mensaje. Tampoco puede realizarse una autenticación en cada nodo pues la identidad del que envía queda oculta para los intermediarios.

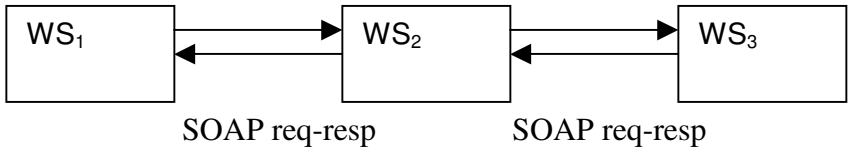
SSL provee confidencialidad a través de la encriptación. Obviamente se quiere garantizar que los datos no pueden ser leídos o modificados en tránsito. Pero resguardar la confidencialidad cuando los mensajes pasan por nodos intermediarios, sin perder información tal como cuáles son los usuarios end-to-end exige analizar otras tecnologías.

IPsec o SSL son alternativas para implementar confidencialidad end-to-end, por ejemplo, en el caso de un mensaje SOAP request-response. O cuando existe una VPN entre la máquina emisora del mensaje y el WS.

Pero distinta es la situación cuando tenemos otros escenarios.
Analicemos un escenario donde intervienen un browser, un website y un WS. Un usuario llenó un formulario a través del browser, y ese dato es el que debe llegar a través del website al WS que está en otro servidor.



Hay que analizar dos conexiones: entre el browser y el website, y entre el website y el web service.
En ambas podría utilizarse IPsec o SSL. Incluso podría haber una VPN entre el website y el WS.
Pero si el WS usa seguridad a nivel de transporte para autenticar y autorizar el mensaje SOAP que llega, conocerá la identidad de la máquina que generó el último requerimiento, no el usuario inicial.
Si usara SSL para autenticar ocurriría lo mismo.
Otro escenario posible es cuando se realiza composición de WS o casos semejantes, donde la confidencialidad debe superar múltiples saltos (nodos intermedios).



Si el mensaje desde WS_1 de pasar a WS_2 para llegar a WS_3 manteniendo la confidencialidad resguardando el paso por WS_2, la seguridad a nivel de transporte o red no será suficiente.
En WS_2 puede ser necesario procesar una parte del requerimiento SOAP ya que de allí se supone que se toma la decisión de invocar a WS_3, sin perder la referencia de WS_1.
En el siguiente cuadro se detalla lo que aporta cada tecnología de seguridad.
Las nuevas tecnologías orientadas a dar solución a este problema involucran XML Encryption, XML signatura y SAML.
Comparando surge SAML como alternativa que ofrece autenticación y autorización, lo que la hace coherente con el análisis objetivo de este documento

Tecnología	Permite
HTTP básico	Autenticación
SSL	Autenticación, Confidencialidad, Integridad
XML Signatura	Autenticación, Integridad
XML Encryption	Integridad, Confidencialidad
XKMS	Autenticación, Confidencialidad, Integridad
SAML	Autenticación, **Autorización**, No Repudio, Integridad

La última A es la que define

Al inicio de este documento se comentó la tendencia actual a cumplir con las 3 A.
Hasta ahora hemos hecho hincapié en la autorización y la autenticación. Qué ocurre con la auditoría?
Las tecnologías que se enumeran en la tabla de arriba no la proveen de por sí.
Si bien SSO (single sign on)[5] podría ser una alternativa para lograr la auditoría, su esquema centralizado y el uso de APIs muchas veces propietarias para lograr que las aplicaciones sean *SSO enabled* , sumado a que no hay acuerdos de interoperabilidad entre los distintos vendedores de SSO complican lograr un modelo como el que exige la auditoría.

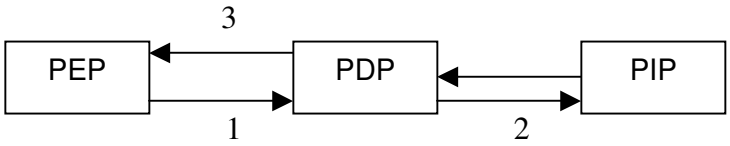
Qué pasa con respecto a SAML? SAML es un estándar de OASIS que guía en la implementación de autentificación, autorización e información de permisos sobre la Red. Es una forma de PMI (Permissions Management Infraestructure), es decir, que usa políticas para manejar el control de acceso y autorización de los sistemas.
Antes, los PMI eran fuertemente propietarios. De allí que la opción de usar SAML haya sido bienvenida: es un estándar abierto que la hace accesible a muchas empresas.
Provee un método para autorización y autenticación por single sign on.

Como está diseñado para ser usado en aplicaciones que interoperan, se puede intercambiar información de forma segura entre grupos de socios sin ningún agregado ni modificación a la configuración de seguridad.

SAML se basa en *Security assertions*, información compuesta por credenciales para autenticación y otros datos necesarios para la autorización.
El *assertion subject* puede ser un usuario o un programa del entorno de ese usuario.

[5] Single sign on permite que los usuarios ingresen información necesaria para autenticación que, una vez aceptada, sirve para acceder a distintos dominios. El usuario no debe recordar distintos usernames y passwords

Al generar un assertion la autoridad de autenticación crea un token SAML que habilita el acceso del subject a aquellas aplicaciones que lo acepten.
El token es un identificador único con datos de autenticación y autorización. Cuando un usuario quiere acceder a un sitio que requiere autenticación, el token es enviado a una aplicación que ejerce la autorización. A esta aplicación se le llama PEP (policy enforcement point).
PEP es el responsable de requerir y reforzar las decisiones de autorización.
El PDP (Policy Decision Point) es el responsable de tomar decisiones en base a las políticas de seguridad existentes.
PIP (Policy Information Point) es donde se almacenan las políticas que usa el PDP.



1. PEP envía un requerimiento de autorización
2. PDP lo analiza contra una política PIP
3. PDP toma la decisión y lo envía al PEP

SAML permite a los emisores indicar 3 tipos de assertions: de autenticación, de autorización y de atributos.
Si bien el esquema puede a primera vista asemejarse mucho a Kerberos, la diferencia es, entre otras, la implementación de arquitectura descentralizada.

Lo que retorna el PDP luego de hacer el análisis contra el PIP es un *authorization decision assertion* que se adjunta al token. Recién allí si la aplicación y recursos a los que quiere acceder son los habilitados por la política, está autorizado el acceso
Como se indica en cualquier texto básico de seguridad, las políticas deben establecerse antes que los mecanismos de implementación. Esas políticas deben comunicarse para el conocimiento de las partes intervinientes.
OASIS desarrolló XACML (Extensible Access Control Markup Language). Por XACML las organizaciones pueden definir y comunicar políticas para el acceso de información on line, por ejemplo, qué clientes pueden acceder a qué información, cuando y cómo.
Cómo trabaja SAML con XACML? El PEP envía un requerimiento de seguridad al PDP, que consulta al PIP donde están almacenadas las políticas en XACML. PDP analiza esas políticas contra el requerimiento.

Una política XACML contendrá campos tipo:
Rules, esquemas que describen los recursos protegidos, acciones y condiciones
Efecct, Deny o Permit
Description, Descripción de la política (a través de esquemas)
Target, Define recursos, sujetos y acciones.

El conjunto de assertions, tickets, junto a los análisis que se llevan a cabo en los PDPs pueden combinarse para lograr el modelo de seguridad que incluya la auditoría que se quiera lograr.

Conclusiones

Como en toda nueva tecnología las vulnerabilidades se descubren a través de su uso. Este duro "aprendizaje" inicia el camino de la optimización del modelo, ganando estabilidad y madurez.
Y en seguridad de web services aún los estándares están inmaduros. De allí que las organizaciones se inicien trabajando en redes internas, con restricción del acceso externo.
Hasta la madurez de los estándares es prudente implementar una combinación de métodos de seguridad tradicionales más los específicos de los WS.
Podemos implementar WS seguros combinando las distintas tecnologías. Por ejemplo, SAML assertions pueden firmarse digitalmente usando XML Digital Signature. Las mismas assertions pueden encriptarse usando XML encryption para asegurar confidencialidad, lo que podría ser validad y registrado por XKMS.
Sobre este punto consideramos que si bien logramos el esquema de seguridad deseado, esta alternativa exige **que las partes que intervengan tengan el mismo stack de productos**, lo que agrega a cada nodo participante capas de software (más carga de procesamiento y aumento del tráfico de mensajes). Es un costo que se debe evaluar.
En Identidad Federada se debe lograr autenticación, proteger los mensajes, implementar service discovery. PKI es una tecnología válida y probada. Pero se hace necesario un administrador robusto de identidad que maneje emisión y revocación, pero en una arquitectura descentralizada.
Los esquemas de seguridad tradicionales pueden ofrecer estos servicios combinando tecnologías para ofrecer un soporte a un framework de este tipo. Pero el manejo del concepto de identidad sumado a la determinación de categorías según políticas y la exigencia de contar con pistas de auditoría, orienta la elección hacia SAML.

Cuál es el costado sombrío? Para implementar SAML, todas las aplicaciones involucradas en la transacción deben ser SAML-enabled, es decir, ser capaz de producir o consumir datos SAML. Vale entonces, lo indicado párrafos arriba, en cuanto a recursos que deben invertir los participantes. La carga de tráfico en la red es un factor muy importante a considerar dado el alto y frecuente grado de intercambio necesario para implementar las 3 A.
Cómo afecta el modelo propuesto, otras características deseables en las aplicaciones distribuidas, tales como la escalabilidad, la extensibilidad? El esquema descentralizado si bien previene la concentración de tráfico en la red, genera un alto grado de intercambio para la comunicación de las políticas, y la autorización-autenticación. Cada nuevo usuario o aplicación debe superar un tiempo inicial de negociación que si bien no atentan contra la escalabilidad, la demoran.
En cuanto a la extensibilidad, este modelo la habilita, pero ante a incorporación de nuevos servicios se deben generar las políticas relacionadas y difundirlas. Lo cual no hace de éste un modelo muy dinámico, ni de reacción inmediata a los cambios.
Para obtener un círculo de confianza entre las partes intervenientes y velar por el mantenimiento sostenido del esquema, creemos que la alternativa de Shibboleth orientada a código abierto, es la opción que garantiza el crecimiento. Pero Shiboleth es un modelo que se adoptó en el ámbito de la educación, donde las pautas de competencia y el secreto inherente a ella hace dudar sobre la adopción en el ámbito comercial (cómo es el caso de Liberty Alliance). Si bien sabemos que esto se soluciona determinando adecuadamente parte visibles o públicas y privadas, hacen al modelo más complejo.
No obstante el mundo real integra ambos universos: el de la educación y el comercial. Esta integración que aun no se da entre los dos modelos de identidad federada que incluimos aquí, puede comenzar a

lograrse a través de SAML 2.0, de reciente publicación, que provee un mayor nivel de seguridad y funcionalidad con respecto a SAML 1.1. Entre otras características, algunos elementos que se consideraban obligatorios han pasado a ser opcionales y hay mayor flexibilidad en el manejo de los contextos, añade prestaciones como enlace de cuentas, logs globales, intercambio de atributos. Pero, SAML 2.0 sigue sin ser compatible con los desarrollos que Microsoft e IBM sobre federación, el protocolo WS-Federation.

Además, consideramos necesario subrayar que cuando el objetivo primordial es la interoperabilidad, características tales como la performance, son secundarias. Los modelos vistos no resistirían la comparación en cuanto a performance con modelos propietarios que garantizan una interoperabilidad entre ambientes homogéneos.

Por último, el esquema de identidad federada, si bien tiene una dinámica inherente que sacrifica la rapidez de disponibilidad de nuevos procesos y usuarios en pos de la seguridad, da un marco de intercambio donde el registro de la información necesaria para la generación de pistas de auditoría es posible.

Bibliografía

1. Basic Security Profile Version 1.0. Working Group Draft. Date: 2005/01/20
2. Basic Understanding of WSDL documents and how to Use a WSDL File to Access a SOAP Service. Lance Robinson, 2002
3. Berners-Lee T. 1999
4. Canales Valenzuela, Glade, Madsen, Rouault. Liberty ID-WSF: A Web Services Framework.
5. Deitel Developer Series – Web Services. A Technical Introduction. ISBN 0-13-046135-0. Pearson Education.
6. Marc Hadley – What's new in SOAP 1.2
7. Microsoft Corporation. Federated Identity Management Interoperability. WS-Federation Passive Requestor Profile Interoperability Workshop. May 2004
8. Moyano, Buccella, Cechich y Estevez – Characterizing semantic web services.. CACIC 2004.
9. SAML Token Profile Version 1.0. Working Group Draft. Date: 2005/01/19
10. Sang Shin – Secure Web Services. The upcoming Web services schemes should help drive web services forward.
11. Simple Object Access Protocol (SOAP) 1.1. W3C Note 08. http://www.w3.org/TR/2000/NOTE-SOAP-20000508. Don Box (DevelopMentor), David Ehnebuske (IBM), Gopal Kakivaya,(Microsoft), Andrew Layman (Microsoft), Noah Mendelsohn (Lotus Development Corp.), Henrik Frystyk Nielsen (Microsoft), Satish Thatte (Microsoft), Dave Winer (UserLand Software). May 2000
12. Steinemann Marc-Alain. Comparison of Liberty Alliance and Shibboleth
13. Web Services Architecture Requirements. W3C Working Draft 14 November 2002. Daniel Austin (, W. W. Grainger), Abbie Barbir (Nortel Networks), Christopher Ferris (IBM), Sharad Garg (Intel Corporation).

Lecturas de interés

- Identidad Federativa en la Educación: Shibboleth y Proyecto Libertad. http://www.sun.com/products-n-solutions/edu/newsletter/educonnection/es/jun04/insidetech01.html
- Framework for Security and Trust Standards. http://www.ninebynine.org/SWAD-E/Security-formats.html

Sitios de interés

- http://www.vordel.com
- http://www.sarbanes-oxley.com
- http://www.w3c.org
- http://www.ws-i.org
- http://www.webservices.org,
- http://www.oasis-open.org
- http://shibboleth.internet2.edu
- http://www.projectliberty.org