Capítulo 5

Shibboleth: una aplicación de SAML

Contenido

- 5.1 Introducción
- 5.2 Internet2
- 5.3 Estándares y código abiertos
- 5.4 SAML y OpenSAML
- 5.5 El intercambio de atributos
- 5.6 Seguridad
- 5.7 Componentes de la arquitectura
- 5.8 Esquema de funcionamiento
- 5.9 Shibboleth en un entorno real

5.1 Introducción

El proyecto de software Shibboleth es una iniciativa que pretende desarrollar una solución abierta y basada en estándares que satisfaga las necesidades de las organizaciones que desean intercambiar información sobre sí mismas y sus usuarios de

una manera segura y que preserve la privacidad. La iniciativa está promovida por los miembros de **Internet2** (véase apartado siguiente) y sus socios. Las organizaciones que podrían hacer uso de un sistema como Shibboleth incluyen universidades, proveedores de contenidos de Internet, agencias gubernamentales, etc. La misión principal del programa consiste en determinar si una persona que está utilizando un navegador de Internet tiene permiso para acceder a un recurso electrónico, basando dicha decisión en información tal como ser miembro de una determinada institución, grupo de alumnos, etc. El sistema consigue preservar la privacidad de sus usuarios al poder éstos decidir qué información sobre sí mismos será conocida por sistemas externos al propio del que proceden.

La iniciativa Shibboleth surgió en Febrero de 2000. En un principio se trabajó sistemáticamente en posibles escenarios de aplicación para establecer los requisitos mínimos que debía cumplir la arquitectura final, la cual está siendo implementada actualmente.

Con respecto al nombre del proyecto, una de las definiciones que podemos encontrar para la palabra Shibboleth dice así: "eslógan o fórmula adoptada por un grupo o secta, con el que se puede distinguir a sus miembros o seguidores, o excluir a los que no lo son". La palabra Shibboleth es, por lo tanto, un término apropiado para un proceso de gestión de accesos.

5.2 Internet2

Internet2 es un consorcio sin ánimo de lucro que desarrolla aplicaciones y tecnologías de redes avanzadas, la mayoría de ellas para transferir información a alta velocidad. Es llevado por 207 universidades de Estados Unidos y otras compañías tecnológicas como Comcast, Intel, Sun Microsystems y Cisco Systems. Algunas de las tecnologías que han desarrollado han sido IPv6, IP multicasting y calidad del servicio. Internet2 está reactivando la cooperación entre el mundo académico, la industria y el gobierno que promovió la Internet actual en sus orígenes.

La enseñanza, el aprendizaje y la investigación en colaboración pueden requerir interconexión y altas conexiones de banda ancha en tiempo real. La infraestructura de Internet 2 soporta esas aplicaciones. También intenta investigar y desarrollar nuevas maneras de usar Internet y la infraestructura de Internet 2 para propósitos educacionales. Internet2 no es un reemplazo de Internet, los organizadores de Internet2 esperan compartir el desarrollo de las redes incluyendo la red de Internet2.

Los principales objetivos de Internet2 son:

- Crear servicios pioneros de red para la comunidad de investigación nacional de los EEUU.
- Posibilitar aplicaciones de Internet revolucionarias.
- Asegurar la transferencia rápida de nuevos servicios y aplicaciones de red a la comunidad de Internet en general.

Shibboleth surgió de la necesidad por parte de las instituciones de Internet2 de colaborar en proyectos "on-line". Las empresas miembros de la organización, tales como IBM y Sun, han aportado el capital tanto intelectual como económico, y arquitectos de software provenientes de instituciones como la Universidad de Washington, Carnegie Mellon, la Universidad del Estado de Ohio, el Instituto de Tecnología de Massachusetts (MIT) y la Universidad de California han contribuido con sus conocimientos y experiencia.

Shibboleth se enmarca dentro de la *Internet2 Middleware Initiative* (I2-MI), la cual intenta conseguir el desarrollo de los principales servicios middleware en las universidades miembros de Internet2. El middleware es una capa de software entre la red y las aplicaciones. Este software ofrece servicios tales como la identificación, la autentificación, la autorización, los directorios y la seguridad. Hoy en día, en general, las aplicaciones web tienen que ofrecer estos servicios por sí mismas, lo que conlleva estándares opuestos e incompatibles. La iniciativa de Internet2 en este ámbito promueve la normalización y la interoperabilidad.

5.3 Estándares y código abiertos

El hecho de que Shibboleth sea una solución "abierta" implica que lo son tanto el código fuente como los estándares utilizados.

Por una parte, las soluciones de código abierto o código libre tienden a ser más seguras al estar disponibles en su totalidad para su análisis por parte de todos los posibles colaboradores, que pueden contribuir a descubrir y reparar vulnerabilidades en el código fuente. La licencia que Shibboleth utiliza autoriza a cualquier persona a modificar y extender el código base. Dicho código se ha programado de forma modular, permitiendo su personalización para entornos existentes, mediante la conexión de nuevos módulos.

Por otra parte, basarse en estándares abiertos tiene la ventaja de que la información que se intercambia entre instituciones y organizaciones podría interoperar con aquella que provenga de otras soluciones. En concreto, Shibboleth está diseñado para utilizar los siguientes estándares, la mayoría de los cuales están ya ampliamente extendidos:

- Hypertext Transfer Protocol (HTTP)
- Extensible Markup Language (XML)
- XML Schema
- XML Signature
- SOAP
- Security Assertion Markup Language (SAML)
- Secure Sockets Layer (SSL)
- Lightweight Directory Access Protocol (LDAP)

El uso de estándares abiertos es particularmente importante en el desarrollo de aplicaciones middleware, al mejorar la interoperabilidad de los diferentes sistemas e incrementar la compatibilidad.

5.4 SAML y OpenSAML

La base de Shibboleth es el estándar SAML. La versión 1.1 de Shibboleth se basaba en la versión 1.1 de SAML, superándola y siendo más avanzada que la propia especificación en muchos aspectos. De hecho, en la versión 2.0 de SAML se incorporaron características que aparecían previamente en Shibboleth como, por ejemplo, las peticiones de autenticación ("authentication requests").

SAML establece normas generales para la estructura de la información y para el intercambio de mensajes en el contexto de un protocolo. Shibboleth proporciona la infraestructura y el "modelo de confianza" necesarios para convertir a SAML en una aplicación útil.

La última versión de Shibboleth que ha aparecido es la 1.3 y se espera que la versión 2 converja casi por completo con la correspondiente de SAML, la cual ya existe como estándar oficial. En EEUU existen ya pruebas de sistemas Shibboleth en fase alfa, las cuales se están llevando a cabo en algunas instituciones universitarias, proveedores de contenidos de Internet y en proyectos de la administración pública relacionados con bibliotecas y gestión de derechos digitales.

Un componente básico de Shibboleth es **OpenSAML**, un conjunto de librerías Java y C++ de código abierto que pueden ser utilizadas para construir, transportar y analizar mensajes SAML. OpenSAML es capaz de almacenar individualmente los campos de información que componen un mensaje SAML y construir correctamente su representación XML, así como de llevar a cabo el proceso contrario, descomponiendo un documento XML en sus elementos individuales para entregarlos a un destinatario.

Permite utilizar el **SOAP Binding** para el intercambio de peticiones y respuestas SAML (la versión C++ ofrece sólo soporte para peticiones). Proporciona, además, soporte adicional para la implementación de sistemas web de single sign-on que empleen perfiles SAML que impliquen el uso de un browser, redirecciones y mensajes POST. No ocurre lo mismo con casos que incluyan artifacts (véase capítulo 4), si bien se proporciona la "maquinaria" necesaria para poder implementarlos. La mayoría de los elementos del estándar SAML están soportados.

OpenSAML está diseñado para ser extensible y para poder integrar una amplia gama de "modelos de confianza" y requisitos de seguridad, aunque, por ahora, se orienta primordialmente a transacciones protegidas mediante PKI (Public Key Infrastructure) y TLS/SSL. Ha sido creado por miembros de la organización Internet2, como parte integrante del propio proyecto Shibboleth y, al igual que este último, es software de código abierto, libremente modificable y distribuíble, probado con éxito en Windows XP/2000, Red Hat Linux y Solaris.

5.5 El intercambio de atributos

Cuando un usuario perteneciente a una institución (servidor, sitio o nodo de origen) trata de acceder a un recurso situado en otro dominio de seguridad (servidor, sitio o nodo de destino), Shibboleth envía información sobre dicho usuario a dicho dominio remoto, en vez de forzar al usuario a someterse a un proceso de autentificación en el destino. El sistema donde radica el recurso deseado puede utilizar esta información, los atributos del usuario, para decidir si otorgar o no el acceso a dicho recurso.

Shibboleth permite al usuario elegir qué información sobre sí mismo se entregará al nodo de destino. Éste conocerá únicamente los atributos necesarios para llevar a cabo la decisión de control de acceso, protegiendo el anonimato del usuario en los casos en los que su identidad es menos importante que otros factores como, por ejemplo, formar parte de alguna institución o grupo de usuarios concreto. En muchos casos, lo realmente importante a la hora de conceder o no acceso a un recurso es conocer un conjunto de características del usuario, no su identidad. Como ejemplo en la vida "real", consideremos qué ocurre en un bar: un cliente no tiene porqué identificarse para beber una cerveza, tan sólo debe cumplir con el requisito de tener la edad mínima legal para poder consumir alcohol. Se antoja engorroso que el dueño del bar tuviera que pedir un nombre y una contraseña a cada cliente pero, sin embargo, una situación análoga la encontramos hoy en día en Internet.

5.6 Seguridad

Shibboleth trata con un tema especialmente sensible como es el acceso a recursos protegidos, de ahí que la seguridad sea una pieza fundamental a tener en cuenta. El código fuente ha sido cuidadosamente diseñado para hacerlo a prueba de ataques.. Asimismo, se usan técnicas para proteger a los atributos en tránsito, que son contempladas por el propio estándar SAML como, por ejemplo, el uso de un canal seguro utilizando SSL.

Todos los servidores que intervienen en las transacciones se autentifican usando certificados digitales. En la arquitectura Shibboleth no se especifica el uso de certificados por parte de los clientes pero se contempla la posibilidad de usarlos, pudiendo incluso simplificar de alguna manera el funcionamiento del sistema (véase más adelante, al final del apartado 5.8)

Es importante resaltar que Shibboleth no limita de ninguna forma lo que el sitio de origen envía como atributos ni las acciones que el servidor de destino pueda llevar a cabo basándose en los atributos recibidos. El sistema proporciona medios para distribuir las características de los usuarios, utilizando certificados digitales y la asociación de los últimos con determinados sitios origen y destino. Una vez que se han entregado los atributos de forma segura en el destino deseado, Shibboleth no garantiza que el uso que se haga de ellos sea el adecuado. De la misma forma, no puede afirmar la veracidad de los atributos entregados (si es verdad lo que un sitio dice sobre las características de un usuario), tan sólo que provienen de la autoridad apropiada y que dicha autoridad los remitió tal y como han aparecido en el destino.

5.7 Componentes de la arquitectura

Al utilizar la especificación SAML como base, algunos de los componentes de Shibboleth han aparecido anteriormente en la descripción del estándar. Los tres bloques funcionales fundamentales que nos encontramos son:

- Proveedor de Identidad
- Proveedor de Servicios
- Servidor “Where are you from?” (opcional)

A continuación describimos brevemente cada uno de estos bloques. Se recomienda acudir al apartado 5.8 para comprender en profundidad el funcionamiento global del sistema.

- **Proveedor de Identidad o Identity Provider**

  Emite aserciones que contienen afirmaciones de autentificación o de atributos (a diferencia de SAML, no se contempla la posibilidad de emitir declaraciones de autorización de acceso), a petición, principalmente, de un Proveedor de Servicios. Los componentes del IdP aparecen en la figura 17 y se describen a continuación.

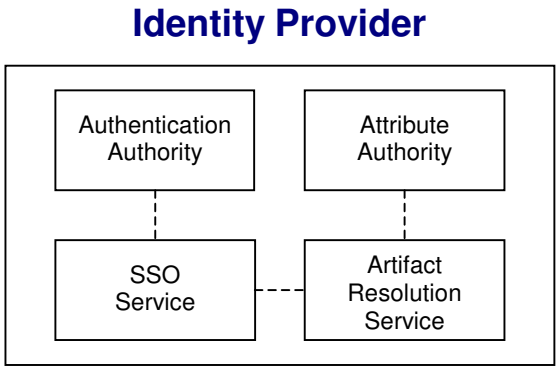

Figura 17: Proveedor de Identidad en Shibboleth

- **Authentication Authority**

  Se encarga de expedir afirmaciones de autentificación. Interacciona con el Servicio Single-Sign On.

- **Single Sign-On Service**

  Constituye el primer punto de contacto del IdP. Inicia el proceso de autentificación y, en último lugar, redirige al cliente al Proveedor de

Servicios. Este componente no se definía en la versión 1.1 de SAML, siendo uno de los aspectos en los que Shibboleth sobrepasaba al estándar.

- **Artifact Resolution Service**

  En determinados situaciones, el Proveedor de Identidad devuelve un SAML artifact al Proveedor de Servicios, en lugar de la aserción propiamente dicha. El SP envía entonces dicho artifact al Artifact Resolution Service, utilizando algún canal alternativo de comunicación. Como respuesta, el Proveedor de Identidad le devuelve la aserción de autentificación requerida.

- **Attribute Authority**

  Se encarga de procesar peticiones de atributos ("attribute requests") y emite aserciones de atributos.

- **Proveedor de Servicios o Service Provider**

  Gestiona recursos protegidos cuyo acceso se basa en la información que recibe del Proveedor de Identidad, en forma de aserciones. Hay que hacer notar la presencia del propio recurso protegido ("Target Resource") en el interior de la estructura del Service Provider. Los componentes del SP aparecen reflejados en la figura 18 y se especifican seguidamente.

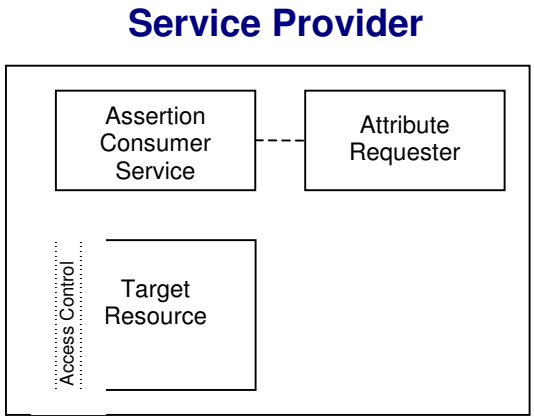

Figura 18: Proveedor de Servicios en Shibboleth

- **Assertion Consumer Service**

  Representa el interfaz utilizado para comunicarse con el Servicio Single Sign-On del Proveedor de Identidad. Procesa la aserción de autentificación devuelta por éste (o bien, por el Artifact Resolution Service, dependiendo del caso de uso), inicia una petición de atributos al IdP (opcional), establece un contexto de seguridad en el Proveedor de Servicios para el usuario actual y redirige al cliente al recurso deseado.

- **Attribute Requester**

  Una vez que ha sido establecido un contexto de seguridad en el Service Provider, el Attribute Requester puede llevar a cabo un intercambio de atributos comunicándose con la Attribute Authority del Proveedor de Identidad.

- **Access Control**

  El Proveedor de Servicios debe proveer algún medio para evitar el libre acceso a los recursos protegidos, permitiendo la intervención del Proveedor de Identidad para supervisar el control de acceso.

- **Servidor "Where are you from?" (WAYF)**

  Este servicio opcional opera independientemente del Proveedor de Servicios y del Proveedor de Identidad. Puede ser utilizado por el SP para determinar el IdP preferido por el usuario, ya sea con la intervención de éste o sin ella

5.8 Esquema de funcionamiento

La figura 19 muestra una transacción completa usando Shibboleth. Intervienen los tres elementos fundamentales mencionados anteriormente: Proveedor de Servicios, Proveedor de Identidad y Servidor WAYF, además del usuario. Existen muchas posibles variantes de este esquema que, en general, son más simples. Incluimos ésta por ser la opción más completa, siempre que no se use un artifact. En el caso de la figura 19, el usuario se presenta ante el Proveedor de Servicios sin poseer una sesión de seguridad en dicho sitio y sin ofrecer ninguna información previa sobre el Proveedor de Identidad donde se encuentran registrados sus datos identificativos.

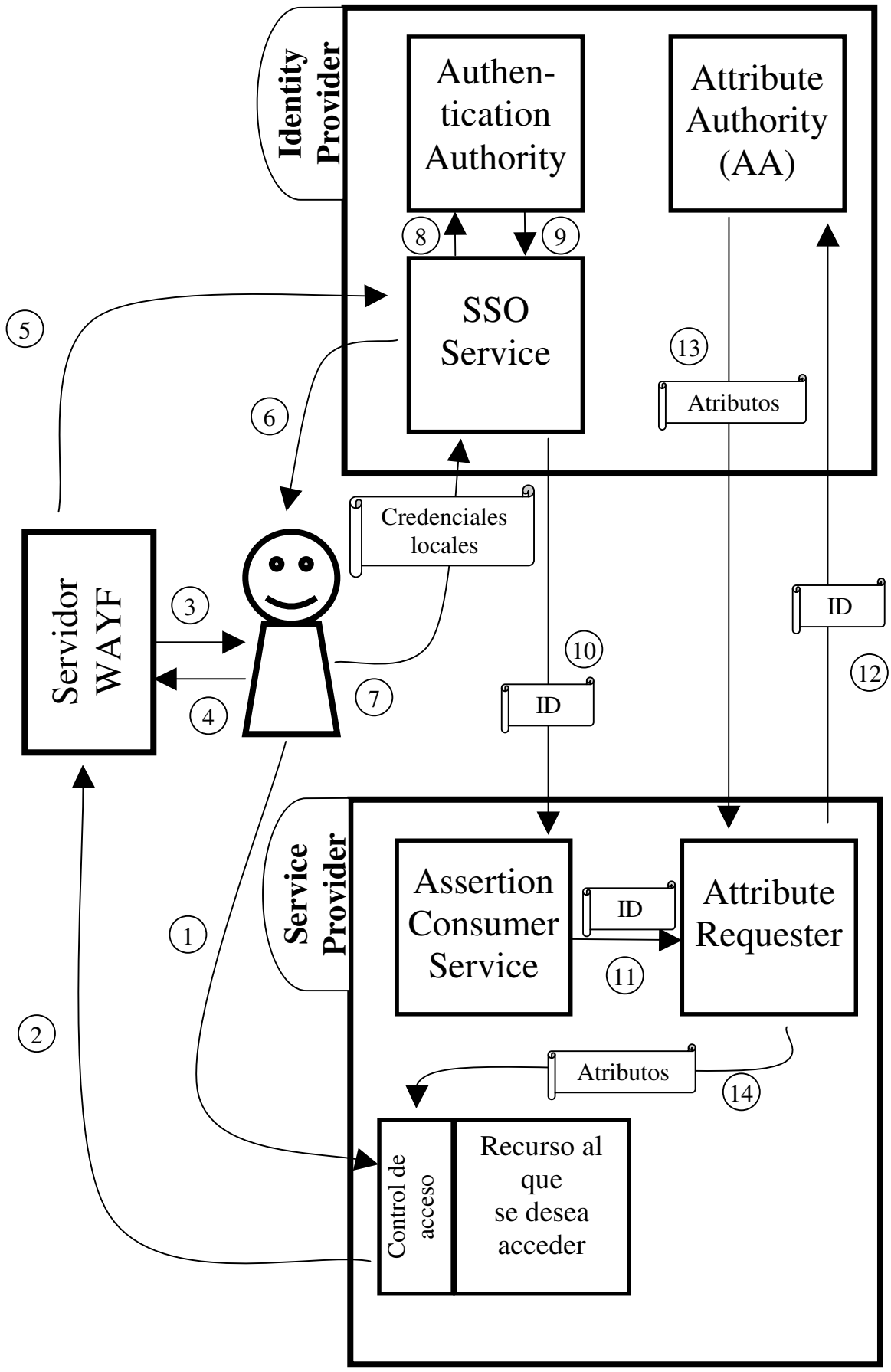

Figura 19:Shibboleth en acción

La interacción mostrada en el diagrama anterior sucede de la siguiente manera:

1. El usuario intenta acceder a un recurso protegido del Proveedor de Servicio, el cual se encuentra situado "detrás" de un control de acceso.

2. El control de acceso no conoce al usuario ni de qué sistema proviene, por lo que lo redirige al servidor WAYF.

3. El Servidor WAYF inicia un proceso para averiguar el Proveedor de Identidad que él usuario desea utilizar (aquél donde están sus datos personales o, en caso de existir más de uno, el preferido por el usuario). El proceso puede ser automático o, como en este caso, interactivo, presentando ante el usuario una lista de posibles Proveedores de Identidad, de entre los cuales el usuario elegirá una.

4. El usuario le indica al Servidor WAYF en qué Proveedor de Identidad desea autentificarse.

5. El Servidor WAYF redirige al usuario al Proveedor de Identidad adecuado.

6. El Servicio Single Sign-On, que forma parte del IdP elegido, pregunta al usuario por sus credenciales en dicho sistema, con el fin de autentificarlo.

7. El usuario responde enviando al Proveedor de Identidad sus credenciales locales (por ejemplo, el nombre de usuario que posee en dicho sistema y su contraseña).

8. El Servicio Single Sign-On comprueba que las credenciales del usuario son correctas y envía una petición de autentificación SAML a la Authentication Authority dentro del mismo Proveedor de Identidad.

9. La Authentication Authority devuelve una aserción SAML de autentificación como respuesta a la petición del Servicio Single Sign-On.

10. El Proveedor de Identidad genera un identificador único (ID) y redirige al usuario al Proveedor de Servicios, para que entregue la aserción de autentificación al Assertion Consumer Service.

11. El Assertion Consumer Service valida la aserción que acaba de recibir, crea una sesión de seguridad para el usuario y transfiere el control de ejecución al Attribute Requester.

12. El Attribute Requester utiliza el identificador que generó el Proveedor de Identidad en el paso 10 para solicitar los atributos del usuario. La solicitud va dirigida a la Attribute Authority, situada en el Proveedor de Identidad.

13. La Attribute Authority del IdP responde con una aserción SAML de atributos. Qué y cuántos atributos componen la respuesta, depende de la política de entrega de atributos que establezca el Proveedor de Identidad.

14. El Service Provider utiliza los atributos recibidos para decidir si permite al usuario acceder al recurso deseado, o bien rechaza dicho intento de acceso.

Como se ha comentado anteriormente, el uso de certificados digitales por parte de los **clientes** (navegadores web) no está incluido por ahora en la arquitectura Shibboleth, aunque se contempla la posibilidad de su uso. Para ello haría falta definir un proceso mediante el cual el cliente proporcionara un certificado en el que constara el nombre del Proveedor de Identidad del que procede y, opcionalmente, un pseudónimo identificativo del usuario. Esto podría obviar la necesidad de usar un Servidor WAYF y también, incluso, el hecho de que el usuario tenga que introducir su nombre y contraseña para autentificarse ante su Proveedor de Identidad.

5.9 Shibboleth en un entorno real

La versión 1.3 de Shibboleth, la más reciente, ha sido probada en Windows 2000/XP/2003, Solaris 2.8, Mac OS X 10.4, Fedora Core 3 y Red Hat Enterprise Linux AS 3 y 4. El Proveedor de Identidad se ha implementado completamente en Java, mientras que del Proveedor de Servicios existen una versión en C++ y otra en Java. La versión Java del Proveedor de Servicios es aún una versión beta, no testada ni optimizada por completo. No puede garantizarse, por tanto, la no existencia de errores en el código.

A la hora de desplegarlo en un entorno real de producción, Shibboleth requiere la existencia de una serie de elementos adicionales. En concreto, la versión Java de Shibboleth necesita lo siguiente:

- Soporte para Java: la versión 1.5 del JDK es la recomendada oficialmente.
- Contenedor de servlets: el sistema se diseña oficialmente para funcionar con Tomcat 5.5.
- Apache versión 1 ó 2: Shibboleth requiere de este servidor web para funcionar, aunque, para llevar a cabo tests, puede usarse un entorno donde únicamente esté presente Tomcat.
- Conector JK para enlazar Apache y Tomcat. Son válidos tanto mod_jk como mod_jk2.
- Soporte SSL: tanto Apache como Tomcat deben contar con él.
- Todos los Proveedores de Identidad y Proveedores de Servicios que intervengan deben contar con certificados digitales para identificarse. Para un entorno de producción se requieren certificados expedidos por una autoridad de certificación en la que confíen todos los Proveedores de Servicio y Proveedores de Identidad implicados. Para un entorno de prueba, basta con certificados generados localmente.
- Un mecanismo web para la autentificación de usuarios de Apache o Tomcat. El Servicio Single Sign-On de Shibboleth no proporciona este sistema, dejándolo a elección de los responsables del Proveedor de Identidad. Puede usarse una

solución software comercial que permita la introducción de un nombre de usuario y contraseña o bien algún mecanismo de los previstos por Apache o Tomcat, como la "Basic Authentication".

- Un repositorio ("Attribute Repository") dónde se almacenan los atributos que lee el Proveedor de Identidad, en concreto la Attribute Authority. Un ejemplo válido sería un directorio LDAP. Este depósito no es necesario en entornos de prueba.

Para terminar, y como curiosidad, merece la pena reseñar que, en uno de los primeros documentos que produjo el Grupo de Trabajo Shibboleth, denominado "Shibboleth Overview and Requirements", se mencionan algunos iniciativas previas en este mismo campo, agradeciendo su esfuerzo, entre otros, al proyecto **PAPI Authentication and Authorization Framework**, desarrollado por la entidad española Rediris.