Implementación de Autenticación de Usuarios con Múltiples Credenciales

Cristian Rubén Pacheco

Universidad Nacional de la Patagonia San Juan Bosco, Facultad de Ingeniería,
Departamento de Informática, de Puerto Madryn.
cpacheco@gmx.com

Resumen. Hoy día los usuarios de Internet hacen uso de una multitud de servicios web, algo que los obliga a tener una cantidad a veces inmanejable de credenciales. Para reducir esta complejidad, este trabajo explicará como se implementó una solución denominada *User's Alias Authentication Workflow* (UAAW) que permite a los usuarios de la web identificarse inequívocamente en los sitios que visitan independientemente de la credencial utilizada y sin necesidad de crear una nueva. UAAW utiliza las últimas tendencias en autenticación, facilitando el proceso y otorgando gran flexibilidad al usuario al momento de identificarse. UAAW reutiliza las credenciales que poseen los usuarios a la manera de aliases delegando la autenticación de los mismos a los servidores responsables de credenciales. En el caso de un usuario autenticado el acceso al sitio se ve simplificado.

Palabras clave: Alias, Autenticación, Credenciales, Usuario.

1 Introducción

Uno de los temas pocas veces mencionados, pero de suma importancia para los usuarios de la web, es la gestión de su identidad. No es un tema trivial y requiere un cuidadoso manejo para evitar violaciones a la privacidad. Por ello, no es conveniente que los usuarios tengan sus datos dispersos por la red y sobre todo en sitios de dudosa reputación. Para resolver esta situación, se han desarrollado tecnologías que permiten que los datos del usuario se encuentren centralizados en sitios confiables que puedan proteger su identidad.

Aunque estas tecnologías tienen el mismo propósito, utilizan distintas formas para lograrlo, por lo que existe una extensa variedad de sitios seguros que están construidos sobre ellas. El presente trabajo presenta una solución para que los sitios puedan permitir a sus usuarios acceder independientemente de la tecnología de autenticación que se utilice.

2 *Single Sign On* e Identidades Federadas

No hay acuerdo en una definición de *Single Sign-On* (*SSO*). Una definición extensamente aceptada es que se trata de un mecanismo mediante el cual un usuario debe iniciar sesión una única vez para tener acceso a múltiples recursos. Así, por ejemplo, puede haber muchas aplicaciones en una red empresarial y, sin embargo, cuando el usuario se conecta a una aplicación y luego se dirige a la siguiente, el mismo no tiene que autenticarse de nuevo. Sus credenciales son entregadas a todas las aplicaciones que participan en un sistema *SSO* y todo esto sucede sin intervención del usuario.

El *SSO* resuelve una serie de problemas asociados a nombres de usuario y contraseñas, principalmente la necesidad de múltiples nombres de usuario y contraseñas, que son difíciles de recordar.

Con *SSO* se pueden poner en práctica mejores controles para obligar a que los usuarios deban utilizar contraseñas difíciles de adivinar. Sin embargo, *SSO* tiene un punto débil: si el usuario pierde su nombre de usuario y contraseña, alguien puede tener acceso a todos las aplicaciones y sistemas a los que tiene acceso. Es más prudente utilizar *SSO* con autenticación de múltiples factores para que se reduzca el riesgo asociado.

Normalmente *SSO* es utilizado dentro de una empresa y se vuelve inútil cuando es necesaria la autenticación fuera de sus fronteras. Esto se debe a que un usuario puede disponer de un nombre de usuario y contraseña para las cuentas de una empresa y otros para una empresa distinta. *Identidad Federada* [1] es otro mecanismo que permite la autenticación a través de empresas o dominios, que requiere una fuerte infraestructura, así como un factor de confianza. Por ejemplo, a menos que una empresa confíe en la administración de identificación de otra, sería arriesgado aceptar las credenciales proporcionadas por esta otra. [2]

2.4 Introducción a los Protocolos de Autenticación

Veamos una breve descripción de algunas tecnologías existentes.

OpenID: es un protocolo abierto que permite a una persona a utilizar una URL como una identidad en múltiples sitios web. Las aplicaciones web pueden utilizar la dirección URL como un identificador para la autenticación, autorización, y otros fines. Es un concepto relativamente nuevo que pone el control de la identidad en las manos de su propietario, el usuario final. El propietario de la identidad puede decidir y controlar qué información debe ser presentada a una aplicación o sitio web para fines de autenticación. [3] [4] [5]

OAuth Core: Permite compartir recursos privados de un usuario (fotos, videos, lista de contactos, cuentas bancarias) almacenados en un sitio con otro sin tener que entregar su nombre de usuario y contraseña.

Aunque *OAuth* no es un protocolo de autenticación, sino uno de autorización que requiere que el usuario ya se encuentre autenticado para dar acceso a los recursos, muchos proveedores de autenticación lo utilizan para autenticar a los usuarios. Tal es el caso de Facebook Connect, Twitter OAuth, etc. [6] [7] [8] [9]

3 UAAW: Autenticación Federada para Aplicaciones Web

Hoy en día los usuarios de Internet se encuentran registrados en múltiples sitios web, lo que provoca que les resulte muy difícil manejar diversos nombres de usuarios y contraseñas. Algunos utilizan el mismo nombre de usuario y contraseña en todos los sitios, registrando sus credenciales en un cuaderno o programa, obviamente estas prácticas provocan graves fallas de seguridad.

Solo es necesario que se descubra un combinación de usuario y contraseña o alguien tenga acceso al registro de las credenciales, para que todas las cuentas de usuario se vean comprometidas.

Muchas veces los usuarios no confían lo suficiente en un sitio web como para entregar sus datos o no tienen intenciones de registrarse en un servicio que solo utilizarán una vez. Sin embargo, es necesario dejar por sentado que, debido a los servicios o recursos que el sitio o servicio ofrece, se hace necesario brindar dicha información. Para resolver estos problemas se crearon las tecnologías de autenticación previamente descriptas, que permiten que los datos personales del usuario se encuentren centralizados en un sitio de plena confianza y se acceda a los otros sitios o servicios indicando su aprobación al proveedor de identidad para que el sitio acceda a sus datos. [10]

Actualmente la mayoría de quienes navegan por Internet cuentan con la capacidad de aprovechar estas tecnologías ya que empresas como Google, Microsoft, AOL y Yahoo, entre otras, actúan como proveedores de identidad. Si bien existen relativamente pocos sitios que permiten a sus usuarios aprovechar estas tecnologías y sólo ofrecen un soporte parcial que, únicamente les permiten autenticarse con un proveedor específico.

Este panorama hizo propicio el desarrollo de una innovadora solución, muy esperada por todos.

Así nace UAAW, una nueva tecnología que permite:

- *Autenticar a un usuario mediante el proveedor de identidad que desee*. No se limita a un solo proveedor ni tampoco, en consecuencia, a un solo protocolo de autenticación. El usuario puede escoger, también, un "proveedor genérico" que le permita autenticarse aún si no encuentra su proveedor de identidad entre los disponibles (posibles proveedores). Considerando los inconvenientes descriptos previamente, una gran cantidad de usuarios se ve claramente beneficiado como

así también los sitios visitados que aumentarán notablemente su tasa de usuarios identificados.

- *Identificar a un mismo usuario aún cuando utilice otras credenciales*. Los usuarios pueden estar registrados en distintos proveedores de identidad, por ejemplo pueden tener una cuenta de correo electrónico en Gmail y otra para MySpace, con un identificador distinto para cada uno de ellas. Por esa razón, UAAW brinda un servicio donde con un sólo registro se puede utilizar cada identificador que el usuario posea como un *alias* distinto.

3.1 Definiciones

Es importante comprender la terminología utilizada en los procesos de autenticación que suceden en la web para comprender la totalidad de esta tecnología. Existen términos ligeramente diferentes usados para el mismo concepto en la documentación de los distintos protocolos existentes.

- *Usuario*. Es el usuario o persona real que desea acceder a diferentes sitios web utilizando sus credenciales que se encuentran en el proveedor de autenticación.
- *Consumer o Relying Party (RP)*. Es el sitio web al que el usuario desea acceder. Es llamado *Consumer*, ya que consume las credenciales otorgadas por el usuario.
- *Identificador*. Es la URL, dirección de email, nombre de usuario, etc. que identifica las identidades digitales de los usuarios en la web.
- *Proveedor de identidad*. Es el *host* donde se almacenan las credenciales de un usuario. Durante el proceso de autenticación, el *Consumer* validará un identificador mediante el intercambio de algunos mensajes con el proveedor de identidad.
- *Agente de usuario*. Generalmente, es el navegador o *browser*. El usuario interactúa con estos agentes de usuario directamente.

Si bien existen otros términos relacionados sólo se dará una noción de su significado en caso de ser necesario.

Las definiciones mencionadas aquí son suficientes para iniciar la descripción del flujo de comunicación.

3.2 Arquitectura

Soporte de protocolos de autenticación. El soporte de los protocolos de autenticación se encuentra en el más bajo nivel en la jerarquía de funcionalidad provista por UAAW. Esta capa se puede subdividir en dos subcapas diferenciadas:

- *Soporte de bajo nivel*. Definición del protocolo. En esta subcapa, se utilizan y definen todas las bibliotecas y funciones necesarias para que los protocolos puedan ser utilizados. Es aquí donde se realizan todos los pasos necesarios para

permitir que el usuario pueda autenticarse con su proveedor, utilizando el protocolo que este entiende.

- *Soporte lógico*. Declaración del protocolo. En esta subcapa se especifica que el protocolo está disponible para su utilización. Si el protocolo está definido pero no es declarado en esta subcapa UAAW no sabrá de su existencia. Para declarar el protocolo solo se requiere asignarle un nombre e indicar la función de la subcapa de bajo nivel que inicia la ejecución del protocolo.

Soporte de proveedores de identidad. La capa de proveedores ofrece los siguientes niveles de abstracción de acuerdo a los distintos roles que lo accedan:

- *Entidad.* El proveedor de identidad es considerado como una entidad por los usuarios. Para los mismos el proveedor solo es un nombre con un icono que lo representa.
- *Instancia de protocolo*. El proveedor de identidad es considerado, por el webmaster, como un conjunto de parámetros que adecuan, en el momento de la invocación, al protocolo de autenticación subyacente.

Autenticación de un usuario. La autenticación de un usuario es la razón de ser de UAAW. Se encuentra en esta capa debido a que no se autentica al usuario directamente, se lo hace indirectamente a través de su alias. El proceso consta de dos grandes partes:

1. *Autenticación del identificador*. Este paso es vital para indicar al *consumer* que el identificador otorgado por el usuario es válido y lo identifica inequívocamente a él.
2. *Autenticación del usuario.* Si el identificador autenticado es un alias de un usuario, la autenticación del mismo es automática. Caso contrario se requiere que el usuario se autentique de la forma tradicional o utilizando otro identificador ya asociado al mismo.

Secuencia de acciones. Para ver cómo interactuán todos los elementos explicados hasta ahora, se detalla el proceso de autenticación:

1. El usuario selecciona el proveedor de autenticación a utilizar, entre los proveedores habilitados. Estos se muestran al usuario ordenados por popularidad. Lo que le permite, a la mayoría de los usuarios, ubicar más fácilmente su proveedor de preferencia.
2. El usuario ingresa los datos requeridos para iniciar el proceso. Aquí los datos solicitados variarán de acuerdo al proveedor seleccionado.
3. El usuario es redirigido por el *agente de usuario* al sitio del proveedor para que el mismo sea autenticado.
    - Si no se encontraba autenticado, el proveedor solicita al usuario sus datos de autenticación (generalmente la contraseña). Si la autenticación es exitosa el proveedor puede (no siempre) preguntarle al usuario si confía en el *consumer*.

4. El usuario es redirigido por el agente de usuario de regreso al *consumer*.
    - Si el proveedor informa que el usuario fue autenticado exitosamente.
        - Si su identidad es reconocida en el sistema (y el alias no se encuentra deshabilitado), el acceso es inmediato.
        - Si no se puede determinar que usuario es, en base a lo informado, se le solicita que indique su situación (si es usuario nuevo o si ya se encuentra registrado en el sistema).
    - Si el proveedor informa que ha fallado la autenticación. Se cancela todo y se permite al usuario que vuelva a iniciar todo el proceso nuevamente.

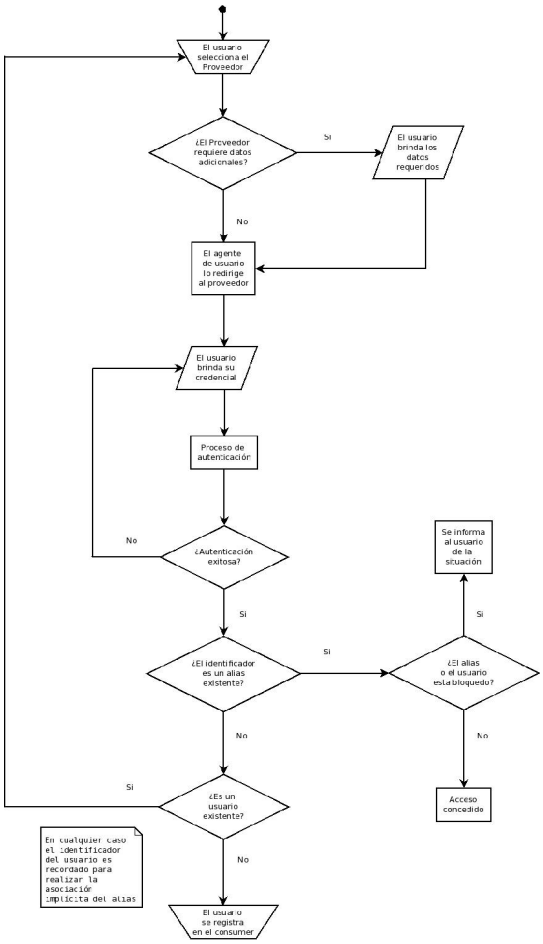

Fig. 1. Flujo funcional que indica la secuencia de acciones y eventos que suceden para lograr que un usuario se identifique en el sistema.

La mayoría de los usuarios no están conscientes de que están utilizando identificadores ni mucho menos que estos serán utilizados como alias cuando se registren. Por lo que a ellos les concierne podrían creer que los alias son como UAAW denomina a los proveedores con los que se han autenticado al menos una vez. Esta aproximación es errónea pero no los obliga a conocer detalles que no quieren saber. Esta abstracción de los identificadores se mantiene en todos los aspectos de UAAW.

Por lo tanto, para abstraer al usuario de los identificadores en el proceso de autenticación se utiliza el conocimiento que se tiene del proveedor de identidad para evitar o, al menos, reducir el identificador del usuario a un dato más ameno al mismo.

Los datos requeridos para iniciar el proceso de autenticación pueden ser algunos de los siguientes o ninguno, de acuerdo al proveedor elegido:

- Nombre de usuario con el que se encuentra registrado en el proveedor. Por ejemplo mrjones.
- Identificador otorgado por el proveedor que identifica al usuario en cualquier sitio web. Por ejemplo mrjones.myopenid.com (en el caso de OpenID).
- Contraseña. Si el proveedor de autenticación es, a la vez, el *consumer*.
- Nombre de perfil. Si el proveedor es un sitio de redes sociales.

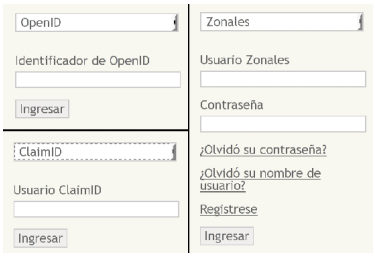

Fig. 2. Distintas pantallas de acuerdo a los datos requeridos por el proveedor de identidad.

Asociación de alias de usuario. Cuando el proveedor autentica exitosamente a un usuario entrega al *consumer* el identificador del usuario. Si el identificador está asociado a un usuario registrado en el *consumer*, entonces se lo denomina *alias*.

Asociación implícita de alias. Se le denomina implícita porque es invocada cuando el usuario ha solicitado iniciar el proceso de ingreso al sistema y no se ha encontrado su alias registrado.

Veamos cómo se logra la abstracción durante todo el proceso de autenticación:

En el paso 4 de la secuencia de acciones de la sección anterior, las operaciones se resuelven de la siguiente forma:

- *"Si su identidad es reconocida en el sistema..."* Significa que si el alias ha sido encontrado entonces el identificador obtenido se encuentra asociado a un usuario.
- *"Si no se puede determinar que usuario es, en base a lo informado, se le solicita que indique su situación..."* Significa que el alias no fue encontrado, por lo tanto el *consumer* no sabe si esto se debe a:
- *El identificador no se encuentra asociado al usuario*. El usuario informa esta situación entonces se le solicitará que se autentique con otro proveedor que haya utilizado antes, es decir, para el cual ya tenga un alias.
- *El usuario no se encuentra registrado.* Obviamente no tiene alias. Se le solicitará que se registre en el *consumer*.
  En cualquiera de las dos situaciones anteriores, realizada exitosamente, el identificador entregado por el proveedor será utilizado para registrar el nuevo alias.
  Aquí hemos descripto la asociación implícita de alias.

Asociación explicita de alias. El usuario dispone de una sección que le permite administrar sus alias. Una de las funciones que posee es la de permitir la asociación explícita de un nuevo alias. Esta funcionalidad ejecuta de manera similar a la secuencia de acciones explicada en la sección de *Autenticación de un usuario*, exceptuando el paso 4, primer subitem, (usuario identificado). La autenticación exitosa asocia el nuevo identificador al usuario vinculándolo con el nuevo alias.

La asociación es explícita porque el usuario indica explícitamente que desea registrar un nuevo alias.

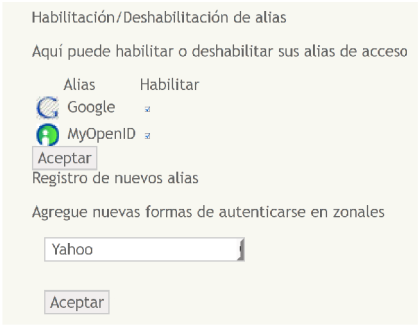

Fig. 3. Pantalla de administración de aliases de usuarios. En la cual el usuario puede habilitar o deshabilitar sus aliases y puede vincular nuevos aliases.

Administración de alias de usuario. La administración de alias es de suma importancia para el usuario ya que le permite controlar los distintos aspectos de sus alias.

Las operaciones soportadas en la administración de alias son las siguientes:

- *Habilitación/Deshabilitación de alias*. Si el alias esta habilitado significa que puede utilizarlo para autenticarse en el *consumer* todas las veces que lo desee.

Por lo tanto mientras el alias este deshabilitado no podrá usarlo. Esto permite un control fino sobre qué alias el usuario autoriza al *consumer* a aceptar. Esta característica resulta muy útil cuando el usuario se ha dado de baja en su proveedor de identidad o si considera que su cuenta en el proveedor de identidad se ha visto comprometida.

- *Asociación explicita de nuevos alias*. El usuario puede iniciar de forma explícita el proceso de autenticación de un identificador para convertirlo en un alias nuevo si la autenticación resulta exitosa.

Registro de un nuevo usuario. Si el usuario desea registrarse en el sitio que visita puede hacerlo de la forma tradicional. De esta manera el sitio le provee una nueva credencial y le genera un identificador interno al usuario. Cuando el usuario asocie un nuevo alias lo estará vinculando con el identificador interno del mismo.

Implementación

Todo lo descripto hasta ahora es logrado gracias a la división de UAAW en componentes que interactúan entre sí.

Componentes

- *Bibliotecas de autenticación*. Bibliotecas propias o de terceros que implementan el protocolo de comunicación entre el *consumer* y el proveedor de identidad. Usualmente los proveedores de identidad ofrecen bibliotecas propias para permitir a los *consumer* el correcto intercambio de mensajes. UAAW utiliza bibliotecas para los protocolos OpenID, FacebookConnect y Twitter OAuth.
- *Base de datos*. Soporte lógico de UAAW, vital para su correcto funcionamiento. UAAW utiliza las siguientes tablas:
  - *#__protocol_types*. Representa la subcapa de *soporte lógico* de la capa de protocolos. En ella se especifica el nombre del protocolo y el nombre de la función que inicia el proceso de autenticación utilizando dicho protocolo.
  - *#__providers*. Representa la capa de *proveedores*. En ella se registran los distintos proveedores de identidad soportados y el protocolo que utilizan.
  - *#__alias*. Representa la capa de *alias*. En ella se registran todos los identificadores asociados a los usuarios.
  - *#__users*. Tabla del sistema que UAAW aprovecha para consultar los usuarios, en particular su identificador interno.
- *Plg_externallogin*. Componente que tiene las siguientes responsabilidades:
  - *Preparación previa a la comunicación*. Utilizando la información del proveedor elegido se escoge la función que soporta el protocolo usado por el proveedor de identidad y se preparan los datos brindados por el usuario para ser utilizados por la función a invocar.

- *Ejecutar la función de autenticación*. Ejecuta la función que indica la subcapa de soporte lógico. La función realiza las siguientes operaciones excluyentes:
    - *Iniciar el proceso de autenticación*. La función prepara los mensajes a enviar (indicando que inicia el proceso de autenticación y el envío de los datos que le solicita al proveedor de identidad) e invoca las funciones de las bibliotecas para realizar el intercambio de mensajes con el proveedor. Si la negociación inicial con el proveedor de identidad es exitosa entonces el agente de usuario redirige al usuario al sitio del proveedor de identidad para que se autentique.
    - *Finalizar el proceso de autenticación*. Cuando el proveedor ha finalizado la autenticación del usuario, la función recibe el estado de la autenticación y el identificador del usuario.
- *Verificación de existencia de alias*. Si la autenticación fue exitosa entonces chequea que el identificador obtenido por el proveedor de identidad este asociado a un usuario existente. Si la autenticación no fue exitosa o el usuario o el alias se encuentra bloqueado entonces indica *autenticación fallida*.

- *Com_user*. El componente de usuarios tiene las siguientes responsabilidades:
    - *Registro de usuarios*. Tiene los siguientes comportamientos:
        - *Registro estándar*. Si el usuario indico que deseaba registrarse en el sistema de forma explicita entonces se le crea una nueva credencial de usuario solicitando usuario y contraseña.
        - *Registro con asociación implícita de alias*. Si el registro se realiza porque el usuario indico que no tenia cuenta en el sistema luego de no haberse encontrado el identificador del usuario, asociado al mismo, entonces se realiza un registro estándar salvo por el hecho de que la contraseña se genera aleatoriamente y se asocia el identificador inexistente al usuario.

          En cualquier caso el usuario recién registrado queda bloqueado hasta que confirme su cuenta de correo electrónico declarada.
    - *Registro de alias*. Vincula el identificador especificado con el usuario correspondiente. Independientemente si el registro de alias es consecuencia de la asociación explicita o implícita de alias.
    - *Login*. Invoca a plg_externallogin para autenticar al usuario. Si el mismo da su aprobación se continua con el proceso para permitir al usuario acceder al sistema. Si recibe junto con los demás datos el identificador obtenido del proveedor de identidad también realiza la asociación implícita de alias.
    - *Logout*. Permite al usuario abandonar el sitio de forma segura.
    - *Desbloquear a un usuario*. Permite que un usuario pueda usar su cuenta si su dirección de correo electrónico es validada.
- *Mod_zlogin*. Permite al usuario elegir el proveedor de identidad a utilizar y especificar los datos auxiliares necesarios para iniciar el proceso de

autenticación. Si además de los datos suministrados por el usuario se tiene un identificador válido otorgado por el proveedor de identidad entonces mod_zlogin envía el identificador a *login* de com_user para que realice la asociación implícita de alias.

- *Plg_authproxy*. Componente que se encarga de recibir los datos provenientes del proveedor de identidad, cuando el agente de usuario redirige al usuario de vuelta al *consumer*. Una vez recibidos los datos son entregados a la funcionalidad de *login* de com_user para que se finalice el proceso de autenticación.
- *Com_alias*. Componente que permite realizar la administración de los alias del usuario. Sus funcionalidades principales son la *asociación explicita de alias,* en el cual utiliza a mod_zlogin, y la *habilitación/desabilitación de alias*.

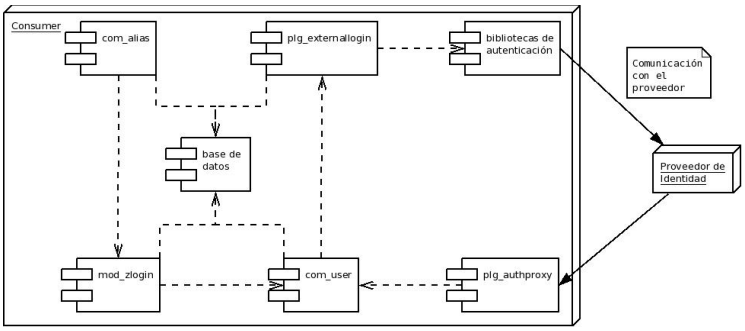

Fig. 4. Se muestran el sitio visitado (*consumer)* y en donde se encuentran las credenciales (el *proveedor de identidad)*. En el *consumer* se pueden ver los distintos componentes que forman UAAW. Las flechas punteadas indican la dependencia funcional entre los mismos.

Conclusiones

El desarrollo de UAAW ha permitido tener un conocimiento más profundo sobre el funcionamiento de las distintas tecnologías de autenticación mencionadas como así también un gran avance en el manejo de identidades digitales.

UAAW demuestra que la delegación del proceso de autenticación provee flexibilidad al *consumer* debido a que no necesita conocer ni soportar distintas técnicas de autenticación. Además brinda seguridad al usuario debido a que el sitio que lo está autenticando es de su confianza y no tiene que entregar su contraseña a un sitio desconocido.

Un problema muy común que se da en la web: sitios que consideran que credenciales distintas equivalen a distintas identidades de usuario. Cuando, en realidad, un mismo usuario puede tener múltiples credenciales. UAAW resuelve este problema de forma simple y elegante como ya quedó asentado.

Sin embargo, para UAAW quedan aún pendiente los siguientes ítems:

- *Falta de documentación.* Aún no se ha confeccionado un manual de usuario completo y de fácil comprensión ni tampoco uno estrictamente técnico.

- *Ausencia de una interfaz amigable de administración*. Actualmente, la administración de los protocolos y proveedores debe hacerse modificando los campos en la base de datos.
- *Pocos protocolos soportados.* Si bien existen muchos protocolos de autenticación, UAAW solo soporta unos pocos de ellos, dejando afuera a miles de usuarios.

Referencias

1. Wikipedia, http://en.wikipedia.org/wiki/Federated_identity
2. Wikipedia, http://es.wikipedia.org/wiki/Single_Sign-On
3. Rafeeq Ur Rehman: The OpenID Book. Conformix Technologies Inc.. , 247. , (2008)
4. No Design, http://security.nocdesigns.com/openid_white_paper.htm
5. OpenID Fundation, http://openid.net/specs/openid-authentication-2_0.html
6. Hueniverse, http://hueniverse.com/oauth/
7. OAuth, http://oauth.net/core/1.0a/
8. Messaging News, http://www.messagingnews.com/story/oauth-giving-access-castle-without-losing-control
9. O'Reilly, http://radar.oreilly.com/2010/01/whats-going-on-with-oauth.html
10. Ping Identity Corporation, Secure Internet SingleSign-On 101 (2010)