Proyecto de Grado

Presentado ante la ilustre Universidad de Los Andes como requisito final para obtener el Título de Ingeniero de Sistemas

Diseño e Implementación de una infraestructura de identidad digital para la Red de Datos de la Universidad de Los Andes

Por

TSU Daniela Carolina Gutiérrez González

Tutor: Rafael Rivas Estrada, PhD

Asesor Industrial: Ing. Alejandra Stolk

Tutor Industrial: Ing. Juan Luis Chaves

Febrero 2017

Diseño e Implementación de una infraestructura de identidad digital para la Red de Datos de la Universidad de Los Andes

TSU Daniela Carolina Gutiérrez González

Proyecto de Grado — Sistemas Computacionales, 108 páginas

Resumen: El acceso a la mayoría de los servicios de la Red de Datos de la Universidad de Los Andes (RedULA) está sujeto al proceso de identificación de los usuarios. Mantener la identidad de dichos usuarios puede resultar todo un reto en el área de seguridad en redes, el cual esta directamente vinculado a la confiabilidad de la información académica y de investigación que ReDULA resguarda para la Universidad de Los Andes. Actualmente en RedULA la autenticación y autorización a los recursos se hace mediante el servicio de directorio y diferentes clientes de este servicio, que son mediadores en el proceso de autenticación. El mecanismo actual es considerado obsoleto ya que presenta desventajas como: duplicidad en la información de los usuarios, vulnerabilidades en los diversos clientes que realizan la autenticación, entre otras. Para abordar este problema se propone el diseño y la implementación de una infraestructura de seguridad de identidad basada en arquitecturas de identidad federadas, que permita tanto la autenticación como autorización de los usuarios de forma segura y estandarizada entre los diversos servicios de la red.

Palabras clave: SAML, *Identidad Digital*

Este trabajo fue procesado en LaTeX.

A mis padres, mi esposo, mis hermanos y mis sobrinas

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

A Dios y la Virgen, por ser la fuente de fe en cada meta trazada.

A la Universidad de Los Andes, por ser el recinto donde consolidé mi formación intelectual y profesional.

Al Profesor Rafael Rivas por depositar su confianza en mí y en este proyecto.

A los Profesores Jurados, por su dedicación a la revisión de este trabajo de investigación y los consejos oportunos.

A los Tutores Industriales Juan Luis y Alejandra por su guía y apoyo durante el desarrollo de este trabajo.

A RedULA, mi segunda escuela y el lugar donde hice amistades invaluables.

Capítulo 1

Introducción

1.1. Antecedentes

El problema de autenticación y autorización de usuarios a diversos recursos en la red, ha sido objeto de estudio por diversas universidades y organizaciones. En este sentido se han desarrollado varias propuestas de estándares y protocolos que buscan resolver de forma eficiente éste problema, manteniendo niveles de seguridad que permita a los usuarios y a los sistemas funcionar de forma robusta. Algunas propuestas han abordado este problema diferenciando la autenticación de la autorización.

Entre los protocolos de autorización de acceso a recursos se encuentra *OAuth* que es un protocolo que permite el acceso de terceros a recursos limitados. Este acceso se logra por medio de una capa de autorización a través de la cual el cliente solicita acceso a los recursos protegidos alojados en un servidor. De esta forma diferentes aplicaciones pueden acceder a recursos de un usuario que están en otra aplicación. Sin embargo con el protocolo *OAuth* sólo se resuelve el problema de autorización de acceso a recursos, para el caso de autenticación de usuarios han sido planteados otros protocolos y arquitecturas.

Otro enfoque para tratar este problema viene dado por el estándar SAML (*Security Assertion Markup Language*) que se define como "un marco de referencia basado

en XML para comunicar la autenticación de usuario, permisos, y la información de atributos. Como su nombre indica, SAML permite a las entidades de negocio hacer afirmaciones sobre la identidad, atributos y derechos de un sujeto (una entidad que es a menudo un usuario humano) a otras entidades, como una empresa asociada u otra aplicación empresarial." (OASIS Security Services, 2015) SAML es ampliamente utilizado en arquitecturas federadas de identidad, así como en software para manejar autenticación web y SSO (*Single Sign On*).

En este sentido encontramos soluciones como las propuestas por las arquitecturas federadas de identidad. Este tipo de arquitectura busca resolver el problema de compartir la identidad digital de las entidades a través de múltiples proveedores de servicio. Como se explica en (WS-Federation, 2015) básicamente una arquitectura de este tipo tiene los siguientes elementos: entidad, identidad digital, dominio de aplicación, atributos y credenciales. Entre las diferentes propuestas de arquitecturas federadas de identidad tenemos *Shibboleth*.

Shibboleth nace de una iniciativa de las universidades miembros de Internet2, cuyo objetivo principal es permitir la colaboración y el acceso a recursos entre instituciones (Shibboleth Consortium, 2015). Shibboleth es una solución de identidad federada, implementada bajo un software de código abierto que basa su arquitectura en el estándar SAML para la gestión de la autenticación y funciona principalmente como la mayoría de las soluciones SSO, las cuales tienen como componentes esenciales: un navegador o cliente, un recurso restringido al cual se desea tener acceso, un proveedor de identidad (*IdP Identity Provider*) que es el servicio encargado de autenticar al usuario y proveedor de servicio (*SP Service Provider*) quien realiza las operaciones de SSO para permitir el acceso a los recursos restringidos.

Existen otras soluciones de uso más comercial para federación de identidad como lo es el proyecto OpenAM (OpenAM Reference, 2015). Éste proyecto desarrollado por la comunidad ForgeRock provee un software de código abierto para manejar autenticación, autorización, federación y SSO, lo que permite manejar usuarios y

permisos a través de dominios diferentes.

En la Universidad de Los Andes algunas investigaciones han desarrollado propuestas que involucran soluciones de éste tipo, una de éstas es el la tesis de pre-grado que plantea el desarrollo de un servicio web para la modeloteca del sistema nacional de simulación (Bruno M. Rengifo C., 2011). Parte de la solución descrita en este trabajo incluye la implementación de OpenID como mecanismo de autenticación para iniciar sesión en los diferentes sistemas (Dokeos y Gitorius) que componen la solución desarrollada. Otra investigación de interés desarrollada en la tesis de pre-grado que expone el desarrollo de un servicio web para el simulador de eventos discretos GALATEA (Gustavo J. Marcano V, 2015) implementa la solución del proyecto OpenAM para la gestión de usuarios.

1.2. Justificación

Existe la necesidad de actualizar el sistema que actualmente está en funcionamiento para la autenticación y autorización de los usuarios en los servicios de RedULA, por un sistema robusto y seguro que facilite la unificación de la identidad de los usuarios entre diversos dominios y dependencias, considerando para esta mejora una arquitectura de identidad federada y estándares para el intercambio seguro de información.

1.3. Planteamiento del problema

Actualmente el acceso a diversos recursos es gestionado mediante el servicio de directorio, que mantiene los datos de los usuarios requeridos para el acceso a otras plataformas de servicios tales como: territorio digital, repositorio institucional, servicios bibliotecarios, entre otros. Esto conlleva a un problema de seguridad debido a que no se proveen condiciones de autenticidad e integridad en el intercambio de información entre el servidor de directorio y los clientes instalados en el resto de los de servicios.

1.4. Objetivos

1.4.1. Objetivo General

Implementar una infraestructura de identidad digital para los servicios de RedULA, basada en una arquitectura de identidad federada que garantice la autenticidad e integridad en el intercambio de datos de autenticación y autorización.

1.4.2. Objetivos Específicos

1. Identificar los requerimientos de manejo de identidad de RedULA
2. Diseñar una infraestructura de identidad digital que se adapte a las necesidades de RedULA
3. Implementar el diseño de la infraestructura de identidad digital en un ambiente con servidores de pruebas que permitan realizar la validación y verificación de la infraestructura propuesta sin poner en riesgo la integridad de la red.
4. Validar la implementación de la infraestructura con pruebas de concepto de integración de los servicios: Territorio Digital en la facultad de ingeniería y Webmail de la Universidad de los Andes.
5. Documentar con manuales en línea la implementación de la infraestructura de identidad digital de la Universidad de Los Andes.

1.5. Metodología

El diseño e implementación de la infraestructura de seguridad digital será desarrollado de forma secuencial y dividiendo el trabajo en una serie de tareas, el producto final de cada etapa debe ser la entrada para continuar el trabajo en la etapa siguiente. Se utilizará como apoyo en la fase de desarrollo la metodología ágil Kanban, comúnmente utilizada para mejoras sobre sistemas existentes (Scott W. Ambler y

Matthew Holitza, 2012).

El término Kanban se refiere al origen de esta metodología en Japón que es utilizada para procesos industriales de manufactura un Kanban es una tarjeta con un ítem de trabajo (Anderson D.J, 2010). Kanban utiliza varios principios entre los cuales destacan: el principio de regulación de trabajo en progreso, mejorando la productividad del equipo de desarrollo y reduciendo los tiempos de ejecución de cada tarea; y el principio de visualización del flujo de trabajo, para lo que se utilizan tableros en donde están las tareas en todas sus fases. Utilizar esta metodología para organizar el trabajo permitirá descomponer en tareas muy específicas el proyecto y de esa forma ir cumpliendo los objetivos.

Las fases principales en este trabajo se dividen de la siguiente forma:

1. Análisis de requerimientos: En esta etapa se debe recopilar los requerimientos y las necesidades en el área de identidad digital existentes en los diferentes servicios de RedULA.

2. Diseño de la solución: En la etapa de diseño es necesario evaluar las diferentes soluciones existentes que pueden ser aplicadas para resolver el problema y determinar cual es la más adecuada. Al finalizar esta etapa debe estar definido el diseño de la infraestructura de seguridad.

3. Implementación: En la etapa de implementación se realizará la instalación de los diferentes servicios definidos en el diseño, así como la programación de los módulos de autenticación para el servicio de webmail y territorio digital de RedULA.

4. Verificación: En la etapa de verificación se procederá a realizar las diferentes pruebas que validen el diseño implementado.

1.6. Estructura del documento

Capítulo 1: Introducción. En este capítulo se definen las bases y el alcance del proyecto, como lo son los objetivos, planteamiento del problema y la justificación. El primer capítulo ofrece una revisión general de algunos antecedentes así como también una guía general sobre el proyecto.

Capítulo 2: Marco Teórico. En esta sección del documento se realiza una revisión de los aspectos básicos relacionados con identidad digital, protocolos y las características más relevantes de las arquitecturas de identidad federadas.

Capítulo 3: Diseño implementación y prueba del sistema. En este capítulo se encuentra la descripción del sistema implementado organizado por grupos de tareas destinados a cumplir cada uno de los objetivos. En esta parte se estudia las diferentes alternativas evaluadas y se justifican las decisiones tomadas en base al cumplimiento de objetivos y requerimientos. Contiene la descripción de los pasos ejecutados para llevar a cabo cada tarea, y explica la mayoría de las actividades de configuración realizadas.

Capítulo 4: Conclusiones y recomendaciones. Aquí se hace un análisis de los resultados obtenidos durante el desarrollo del proyecto y se añaden recomendaciones para futuros trabajos afines.

Capítulo 2

Marco Teórico

En este capítulo se presentan los aspectos básicos relacionados con las infraestructuras de identidad digital existentes. Adicionalmente se revisan las características más relevantes de una arquitectura de federación de identidad, generando de esta forma la base conceptual que sustenta el diseño de la infraestructura de identidad digital propuesta en este trabajo.

2.1. Identidad digital

2.1.1. Entidad

Se conoce como entidad a cualquier persona, grupo de personas, una institución ó incluso dispositivo que puede ser capaz de realizar una transacción (Fragoso-Rodriguez et al., 2006). Este concepto es fundamental para las siguientes definiciones relacionadas con la identidad digital.

2.1.2. Identidad

En (Real Academia Española, 2016) se define identidad como "el conjunto de rasgos propios de un individuo o de una colectividad que los caracterizan frente a los demás" Éstos rasgos y características propias pueden ser físicas (por ejemplo: color de cabello, altura, contextura, etc.) o también pueden ser ciertas habilidades que un sujeto posee

(saber conducir un automóvil, hablar un idioma, programar en cierto lenguaje). No existen dos identidades iguales, aunque dos sujetos posean características comunes esto no quiere decir que compartan la identidad, a cada identidad se asigna un conjunto único de características, las cuales pueden evolucionar el en transcurso del tiempo (Abelson et al., 1998).

Otra definición para el término establece que la identidad o la identidad parcial es un conjunto de atributos relacionados a una entidad. (Talamo et al., 2014) En ese orden de ideas y tomando como referencia las definiciones de la investigación realizada en (Mora et al., 2014) se puede articular un concepto un poco más claro de lo que es la identidad digital, y que no difiere mucho de la definición de la palabra identidad establecida en (Real Academia Española, 2016) : "es el conjunto de características (atributos) que son utilizados, desde un punto de vista técnico, como los datos que identifican a un sujeto o entidad y que pueden ser procesados por sistemas computacionales e internet".

2.1.3. Identificador

A menudo las definiciones de identidad y de identificador tienden a confundirse, dado que la identidad de un sujeto es reconocida a través de un identificador (Jøsang et al., 2007). Sin embargo (Talamo et al., 2014) lo define muy claramente como la información de identidad que puede distinguir inequívocamente una entidad de otra. Un ejemplo de identificador es el número de cédula de identidad venezolana, pues es un número único para cada portador de éste documento, otro ejemplo de identificador es el nombre de usuario dentro de un servicio de correo electrónico. Un identificador actúa como tal dentro de un dominio dado, por ejemplo el nombre de usuario alice@dominioX.com sólo identifica una entidad con atributo nombre de usuario = *alice* dentro del dominio de aplicabilidad *dominioX.com*.

Un identificador es entonces un atributo de identidad *especial* ya que dos entidades no pueden compartir el mismo identificador, mientras si pueden compartir otros atributos de identidad (Chadwick, 2009). En (Fragoso-Rodriguez et al., 2006) se habla de la identidad digital como un conjunto de identificadores que pueden ser asignados, seleccionados o implícitos para el usuario, y es por esto que se dice que los conceptos de identidad y de identificador están tan relacionados.

2.1.4. Atributos

Los atributos son datos que se pueden obtener sobre una identidad dada. Todas las identidades constan de una serie de atributos. El en caso particular de la identidad digital se puede decir que los atributos asociados a una entidad pueden ser: rol, directorio, nombres etc. Dichos atributos son intercambiados entre los entes involucrados (proveedores de identidad y proveedores de servicio) en transacciones de autenticación y/o autorización.

En la figura 2.1 se puede observar cómo la identidad de diferentes entidades consisten en atributos, los cuales pueden ser comunes para más de una entidad.

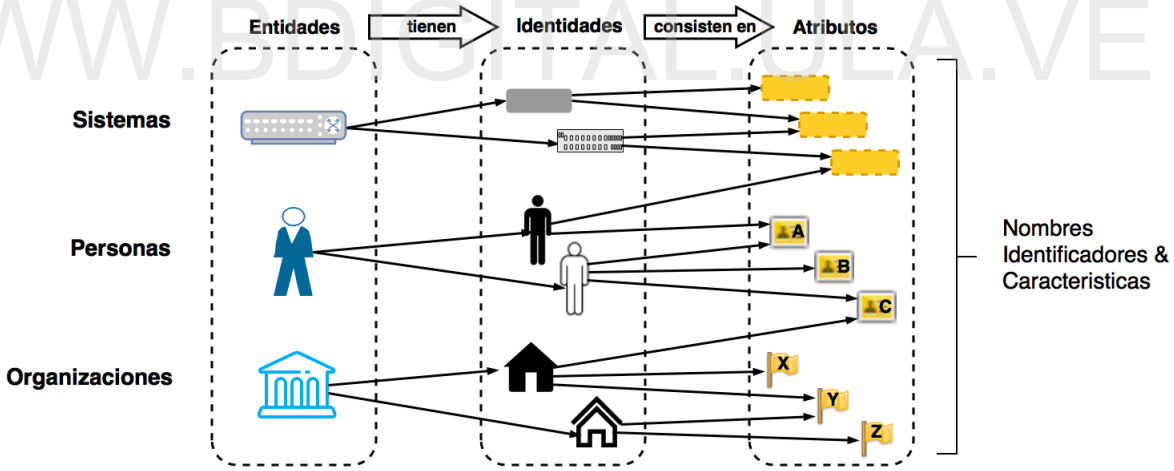

Figura 2.1: Relación entre entidad, identidad y atributos (Loutfi y Josang, 2015).

2.1.5. Credencial

Una credencial es "cualquier elemento que sirva para autenticar la identidad de un usuario o entidad con el propósito de validar sus identificadores". En este sentido una credencial puede ser una contraseña, una pregunta de desafío, un certificado digital,

entre otros. En otros ambientes también puede ser cualquier característica física de la entidad como su huella dactilar, voz, entre otros. (Fragoso-Rodriguez et al., 2006)

Existe una diferencia entre una credencial auténtica y una credencial válida que es necesario resaltar:

Credencial auténtica: es aquella que no ha sido manipulada de ninguna forma y que es recibida exactamente como ha sido emitida por la autoridad expedidora. Para probar su autenticidad se utiliza la firma digital de quien la emite.

Credencial válida: es aquella que es de confianza para su uso por el receptor, que en este contexto es llamado *relying party* (parte que confía). La diferencia fundamental entre estos dos tipos de credentiales es si el *relying party* confía o no en el emisor de la credencial para emitir la una credencial particular.

2.1.6. Autenticación

Cuando se habla de autenticación en el contexto de identidad digital es importante resaltar la diferencia existente entre un sistema como entidad (por ejemplo un servidor) y una entidad legal y cognitiva (una persona o una institución), esto deja entrever la posibilidad que existe de tener distintas entidades en cada extremo de un esquema de comunicación cliente-servidor. Típicamente en el acceso a servicios en línea las entidades que juegan un rol en la autenticación: son el usuario y el servidor. En (Loutfi y Josang, 2015) se denota este tipo de autenticación de la siguiente manera: [U $\rightarrow$ S] para autenticación de usuarios y [S $\rightarrow$ U] autenticación del servidor. Esa misma investigación define tres modalidades de autenticación:

Autenticación sintáctica de una entidad: Es la verificación de que el identificador único de la entidad remota en la interacción es correcto. Este tipo de autenticación sola no provee ninguna seguridad ante ciertos tipos de ataque.

Autenticación semántica de una entidad: Es la verificación de que el identificador único de la entidad remota es correcto, adicional a esto también verifica que las características semánticas de la entidad remota están en concordancia con una política de seguridad específica, un ejemplo de una política de este tipo es una lista de

entidades autorizadas, entre otras.

Autenticación cognitiva de una entidad: Este tipo de autenticación no sólo verifica la autenticidad del identificador y las políticas de seguridad pertinentes, sino que también realiza una verificación con atributos de identidad que permiten reconocer aspectos relevantes de la entidad remota y tomar decisiones basados en esta información, por lo que requiere que la entidad que confía tenga habilidades cognitivas y de razonamiento. Esta modalidad de autenticación bloquea eficazmente los ataques de suplantación de identidad ya que los usuarios pueden reconocer la identidad de un servidor y decidir si es el correcto.

2.1.7. Autorización

La palabra autorización denota el permiso que se le otorga a una persona para realizar alguna acción. En el caso de autorización en el entorno de identidad digital una vez el proceso de autenticación valida al usuario, comienza el proceso de autorización que se encarga de determinar en cuales sistemas o servicios se le permite acceder a dicho usuario. En los sistemas gestión de identidad federada claramente se separan el acceso a los recursos del establecimiento de la identidad y la autorización. De este modo las instituciones no tienen que crear y mantener una base de datos con un gran número de usuarios y credenciales, en su lugar manejan solo sus propios usuarios y aceptan credenciales de otros miembros de la federación. Los atributos de los usuarios son verificados por el proveedor de identidad de la institución a la que pertenecen , lo que permite tener información precisa y actualizada y se elimina la necesidad de propagar cambios a través de los sistemas de identidad de múltiples instituciones. (EDUCAUSE, 2009)

2.1.8. Confianza

La palabra confianza puede ser definida como la seguridad que existe en que un acuerdo entre dos partes será cumplido. Cuando se habla de confianza en sistemas de gestión de identidad, especialmente en infraestructuras federadas, la definición se basa

en el hecho de que tanto los proveedores de identidad (IdP) como los proveedores de servicio (SP) son confiables, por lo que si uno de los participantes se vuelve malicioso o corrupto, entonces el resto de participantes pueden afectarse. Por ejemplo un proveedor de servicio vulnerado puede utilizar los atributos de los usuarios obtenidos del IdP de manera incorrecta (Broeder et al., 2012). Desde la perspectiva del SP se confía en los atributos que son entregados por el IdP, desde la perspectiva del IdP se confía en que el SP hará un uso correcto de los atributos que le son entregados, y desde la perspectiva de la entidad autenticada (usuario) se confía en que el IdP y SP tratarán sus datos con la confidencialidad esperada.

2.2. Federación de Identidad

Una federación de identidad se ocupa de federar dominios de identidad que estén separados a lo largo de organizaciones y empresas, para que usuarios en un dominio puedan tener acceso a servicios en otro dominio, en este sentido la federación es la unión de estos servicios de identidad.

En seguridad web, el término Federación se ha vuelto popular en los últimos años y, como se hace mención en el párrafo anterior, se refiere al manejo de la identidad web de un usuario a través de diferentes dominios de seguridad. La premisa "piensa local, actúa global" describe muy bien lo que es un modelo de identidad federado, en donde los usuarios se autentican con su proveedor de identidad, que es con quien tienen una relación de confianza (piensa local) y luego pueden acceder a recursos en uno o varios proveedores de servicios (actúa global). Una de las principales razones para implementar federaciones en el entorno web es que en general el flujo de trabajo e información en los sistemas web requieren que los usuarios se autentiquen en varios dominios (Anggorojati et al., 2012).

Los modelos de identidad federada se basan en grupos de SPs que convienen en un acuerdo mutuo de seguridad y autenticación con el fin de permitirle a sus usuarios SSO en sus servicios. Dichos grupos son los llamados círculos de confianza. Siguiendo

este planteamiento, una federación de identidad puede entonces ser definida como el conjunto de acuerdos, estándares y tecnologías que permiten que diferentes SPs tengan acceso a atributos de la identidad de otros usuarios y permisos que puedan tener en otros dominios de SPs.

La reciente evolución en el área de identidad, impulsada principalmente por el interés técnico, social y económico han cambiado significativamente la forma de manejar la identidad de los usuarios. La identidad centrada en el usuario (*User-centric identity*) han llevado a la separación lógica del IdP y los SP, lo que permite que los usuarios seleccionen el IdP adecuado para proveer su identidad a un SP dado. Una federación de identidad se ocupa de vincular dominios intra e inter organizaciones de forma tal que la identidad en un sistema sirva para acceder a servicios en otro sistema (McLaughlin et al., 2010).

De acuerdo con la investigación realizada en (Scudder y Josang, 2010) la federación de identidad abre una puerta a una situación de ganar-ganar en la que la experiencia del usuario es mucho más simple ya que no se requiere que el usuario inicie sesión en cuentas individuales, y al mismo tiempo es mucho más segura pues sus credenciales y atributos están mas protegidos. Como un ejemplo de esto si un usuario solo necesita una contraseña donde antes necesitaba diez, entonces la nueva contraseña puede ser más larga, más segura y se puede cambiar con más regularidad sin sobrecargar al usuario.

Debido al alto grado de confianza que existe en una federación es importante el establecimiento de políticas y requerimientos organizacionales tomando en consideración los estándares para la implementación de la federación.

2.2.1. Arquitectura de identidad federada

Una arquitectura de identidad federada es un grupo de organizaciones entre las cuales existe un acuerdo de confianza mutua para el intercambio de información de la identidad digital de los usuarios, de forma segura y preservando la integridad y la

privacidad de la información personal de los usuarios. (Fragoso-Rodriguez et al., 2006) Aunque los elementos que componen una federación pueden variar dependiendo de la implementación particular, a continuación se describen a grandes rasgos los que son comunes para la mayoría.

2.2.1.1. Proveedor de Identidad (IdP)

En general un proveedor de identidad es el encargado de autenticar a sus usuarios. en (Broeder et al., 2012) se define como la autoridad de atributos combinado con el servicio de autenticación, puesto que dentro de una federación de identidad el IdP puede hacer la tarea de autenticar un usuario y luego emitir una afirmación con los atributos del usuario. Estas afirmaciones vienen firmados digitalmente para asegururar la autenticidad e integridad. Dentro de una arquitectura de identidad federada se puede tener un solo IdP (esquema centralizado) o varios IdPs (esquema distribuido).

2.2.1.2. Proveedor de Servicios (SP)

Dentro de una federación de identidad, el proveedor de servicios es quien necesita validar la identidad de un usuario, en este contexto el SP solicita al IdP la autenticación, y espera como respuesta un conjunto de atributos con los cuales realizará alguna acción. Por ejemplo: un SP para el servicio de correo electrónico no solo necesita verificar la identidad del usuario, sino también conocer atributos, como por ejemplo el directorio hogar, para saber donde está ubicado el buzón con los correos del usuario.

2.2.2. Funcionalidades de una arquitectura de identidad federada

Desde el punto de vista del usuario final, proveedor de servicios y proveedor de identidad, la arquitectura de identidad federada debe cumplir con ciertas funcionalidades, descritas en (Fragoso-Rodriguez et al., 2006) como:

- **Single Sign On:** Permite a los usuarios realizar un solo inicio de sesión y con éste poder acceder a diferentes aplicaciones y servicios. La arquitectura de identidad

federada debe garantizar que exista dicha funcionalidad, la cual se explica más adelante en este capítulo.

- **Intercambio de atributos:** Al autenticar un usuario con el IdP, el SP necesita atributos adicionales que permitan proveer un servicio personalizado.
- **Gestión del ciclo de vida de la identidad:** La creación, mantenimiento y eliminación de una identidad digital debe ser simple y no debe representar un alto costo de operación.
- **Arquitectura estandarizada:** La arquitectura debe ser desplegada de acuerdo con estándares que permitan el crecimiento de la misma, y faciliten la integración de nuevos IdPs y SPs.

En la figura 2.2 se observa como un usuario dentro de un dominio centralizado de identidad federada puede acceder a diferentes servicios (SP 1 y SP 2) siendo autenticado únicamente en el proveedor de identidad centralizado (UserIDP Centralizado 3), dicho proveedor es el encargado de afirmarle a los proveedores de servicio la autenticidad del usuario.

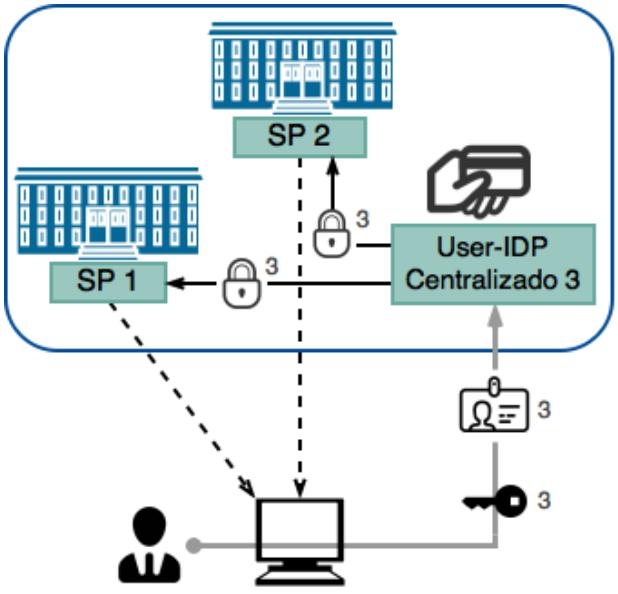

Figura 2.2: Dominio centralizado de identidad federada (Jøsang et al., 2007).

De forma análoga en la figura 2.3 se detalla en una arquitectura de múltiples dominios centralizados cómo un usuario puede acceder a servicios de diferentes dominios autenticándose con un sólo proveedor de identidad.

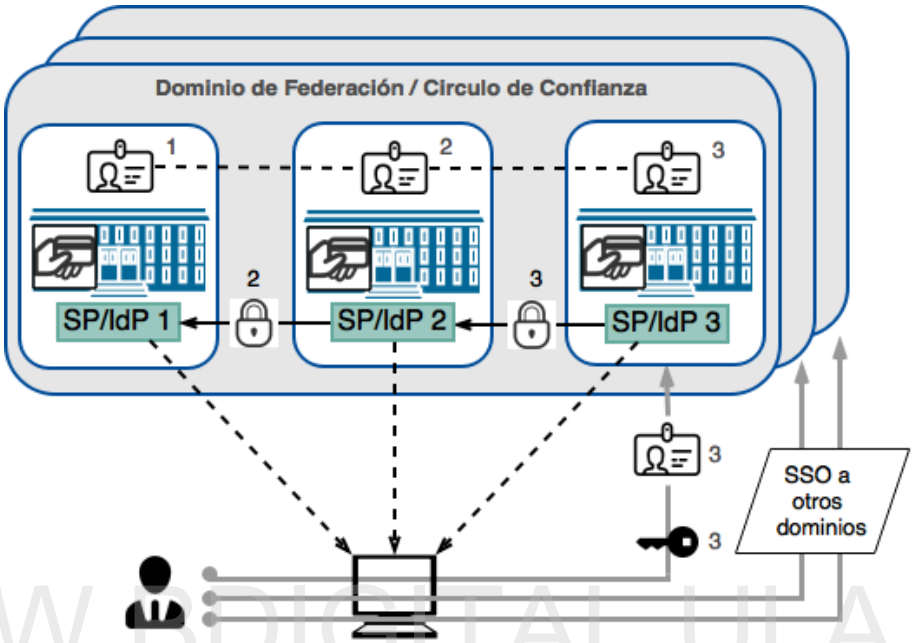

Figura 2.3: Múltiples dominios centralizados de identidad federada (Jøsang et al., 2007).

2.2.3. Single Sign On (SSO)

Se refiere a inicio de sesión unico, y es el mecanismo mediante el cual se le permite a un usuario autenticarse con un IdP y luego poder acceder a los servicios de diferentes SPs sin tener que autenticarse varias veces. Investigaciones como la realizada en (Baldoni, 2012) explican el SSO como el acceso transparente a los servicios federados dentro de una o varias organizaciones con un solo nombre de usuario y contraseña. Según (Radha y Reddy, 2012), existe diferentes clasificaciones de SSO, tomando como categorias: lugar de despliegue, arquitectura, tipo de credencial usada y por último el protocolo que utiliza. Esta clasificación se se muestra en la figura 2.4 y se explica a continuación

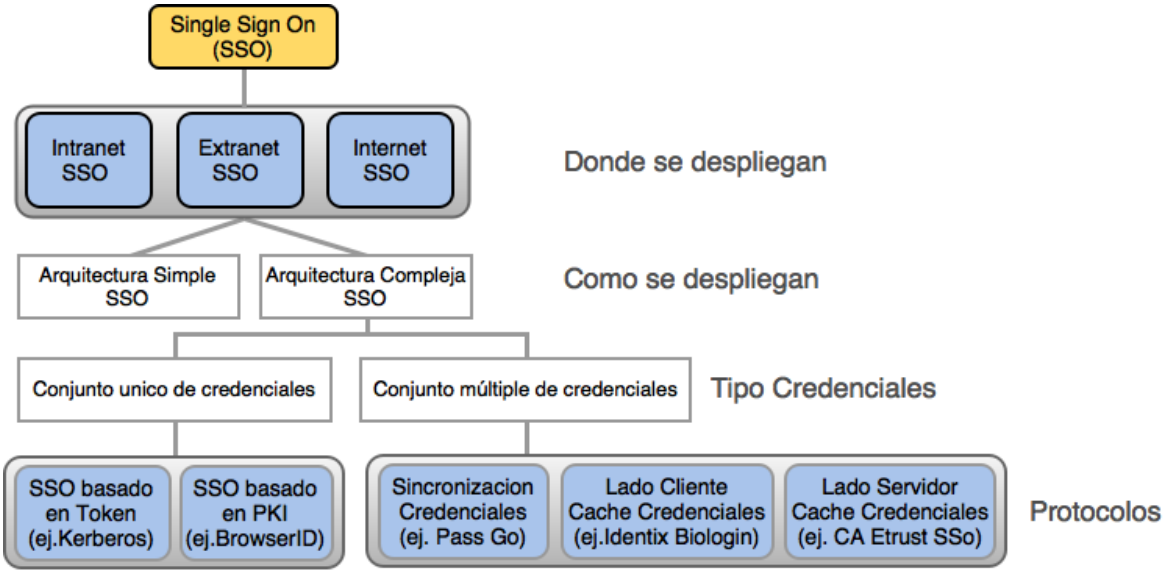

Figura 2.4: Clasificación de Single Sign On.

2.2.3.1. Lugar de despliegue:

- ***Intranet o SSO Empresarial (ESSO):*** Permite la conexión a múltiples sistemas dentro de una misma organización, y está pensado para minimizar el número de veces que un usuario debe escribir sus credenciales de acceso para acceder a diferentes aplicaciones.

- ***Extranet o SSO Multi-dominio:*** Permite la conexión a múltiples sistemas dentro de una misma organización y entre organizaciones con las que se tenga acuerdos de acceso a recursos y servicios. El usuario hace el inicio de sesión en su institución de origen pero puede acceder a recursos de la otra institución, por lo tanto no debe tener diferentes credenciales.

- ***Internet Web SSO:*** Es un mecanismo basado en el navegador o cliente, que provee de acceso con un solo inicio de sesión a diferentes aplicaciones desplegadas en servidores web.

2.2.3.2. Arquitectura de despliegue:

- ***Simple:*** Utiliza una autoridad de autenticación, y un conjunto único de credenciales para cada usuario.
- ***Compleja:*** Utiliza múltiples autoridades de autenticación con un conjunto único o múltiple de credenciales para cada usuario.

2.2.3.3. Tipo de credencial:

- ***Compleja con un conjunto único de credenciales:***
  - **Sistemas SSO basado en token:** Aquí un usuario envia sus credenciales a la autoridad de autenticación, y ésta una vez que verifica las credenciales envía de regreso un token. Cuando el usuario quiere acceder a un servicio con otra autoridad de autenticación diferente envía el mismo token recibido. El éxito en estos procesos reposa en la confianza que existe entre las diferentes autoridades de autenticación. En un ambiente web este tipo de SSO se puede implementar utilizando las *cookies* del navegador. Las *cookies* son un conjunto de información enviada al navegador por el servidor web y que se almacena en la máquina del usuario.
  - **Sistemas SSO basado en PKI:** En estos sistemas los servidores y los usiarios se autentican entre sí utilizando su respectivo par de llaves. La autoridad de certificación que emite estas llaves puede ser diferente para los usuarios y los servidores, y si ese es el caso debe también existir confianza entre estas autoridades.
- ***Compleja con múltiple conjunto de credenciales:***
  - **Sincronización de credenciales:** Los diferentes conjuntos de credenciales necesarias para el acceso a múltiples sistemas son de cierta forma enmascarados como un solo conjunto, dando al usuario la ilusion de que sólo necesitan uno.

- **Almacenamiento de credencial en cliente:** Permite a los usuarios almacenar su informacion de acceso y otra información sensible como contraseñas. Son almacenados en archivos especiales cuyo nombre y estructura puede variar de acuerdo al sistema operativo. Un ejemplo es *Windows credential manager*.
- **Almacenamiento de credencial en servidor**: Es similar al almacenamiento de credenciales en el cliente solo que en este caso las credenciales se alojan en un servidor. Utiliza un servidor central que lleva a cabo la tarea de administrar las diferentes contraseñas y datos de acceso y proveer la información necesaria directamente a la aplicaciñon que lo solicita. Un ejemplo de este mecanismo es CA Etrust.

2.2.3.4. Protocolo de despliegue:

- ***Kerberos:*** Kerberos constituye una implementacion clásica la autenticación basada en *token*. Todo el proceso de autenticacion se divide en tres partes entre sus cuatro entidades:
  Cliente: es quien desea acceder a los recursos.
  Servidor de autenticación: es quien autentica a los clientes y a los recursos.
  Servidor de Ticket: es quien reparte los tickets para el acceso a los recursos.
  Servidor de aplicación: es el recurso, servicio o aplicación al cual se desea acceder.

- ***Security Assertion Markup Language (SAML):*** Es un estándar abierto para el intercambio de información de autenticación y autorización entre dominios. Este estándar no especifica directamente cómo autenticar los usuarios, sino que define un mecanismo para el intercambio de la información de identidad. Más adelante en este capítulo se desarrolla un poco más el funcionamiento de este protocolo.

- ***OpenID:*** Es un protocolo de autenticación descentralizada y consiste en tres entidades principales: **Identificador**: cadena que identifica al usuario, **Relying Party (RP)**: aplicación o servidor que necesita autenticar al usuario, **Proveedor**

OpenID (OP): servidor central que emite y gestiona los identificadores de los usuarios.

2.3. SAML

Es es acrónimo de *Security Assertions Markup Language*, el concepto más completo de SAML tomado del resumen ejecutivo de su documentación oficial (Madsen et al., 2005) lo define como un marco de referencia basado en XML para comunicar información de autenticación, derechos y atributos de los usuarios. Como su nombre lo indica permite realizar afirmaciones (*assertions*) con respecto a la identidad de un sujeto (generalmente un usuario final). Es un protocolo diseñado para ser flexible y extensible y es ampliamente utilizado por otros proyectos. Actualmente se está en uso la versión 2.0, donde unifica las características y funcionalidades que hicieron exitosas sus versiones anteriores, especialmente las relacionadas con identidad federada (versión 1.1) con el aporte e iniciativas de proyectos para la educación superior como ***Shibboleth***. SAML 2.0 es un paso crítico hacia la plena convergencia de las normas de identidad federada.

2.3.1. Casos de uso

SAML es utilizado en una gran numero de diferentes aplicaciones, algunos de los casos de uso más comunes se encuentran detallados a continuación:

2.3.1.1. Web SSO

De acuerdo con el resumen técnico (Hughes y Maler, 2005) el SSO en múltiples dominios es el caso de uso más importante en el que se aplica SAML. Un ejemplo de lo anterior se puede observar el la figura 2.5: donde un usuario inicia sesión en el sitio *airline.example.com* y accede a los recursos protegidos en ese sitio. Luego en algun punto es redireccionado al sitio *cars.example.co.uk* y en este caso el IdP realiza una afirmación sobre la identidad del usuario al SP de *cars.example.co.uk*, como existe una relación de confianza entre estas organizaciones la identidad del usuario solo necesita establecerse una vez.

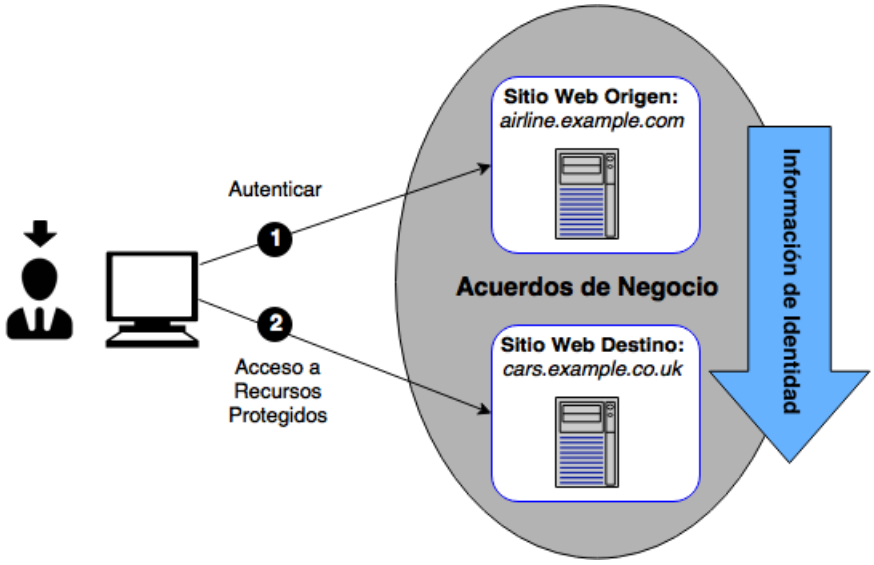

Figura 2.5: SAML: Caso de Uso Web SSO (Hughes y Maler, 2005).

2.3.1.2. Federación de identidad

Se dice que una identidad es federada cuando entre un varios proveedores existe un acuerdo de confianza sobre el conujunto de atributos de identidad que los diversos proveedores necesitan de los usuarios. En la figura 2.6 se observa un ejemplo del flujo e interacción en federación de identidad. Existen ciertas consideraciones que se deben tomar cuando se decide realizar acuerdos (federarse) con otras organizaciones, como por ejemplo:

- Verificar si el usuario tiene identidades locales en los servicios que desean federarse.
- Metodo de inicio y cierre de sesión dinámico o existirá algun IdP prestablecido.
- Definir si es necesario que el usuario apruebe el intercambio de sus atributos entre miembros de la federación.

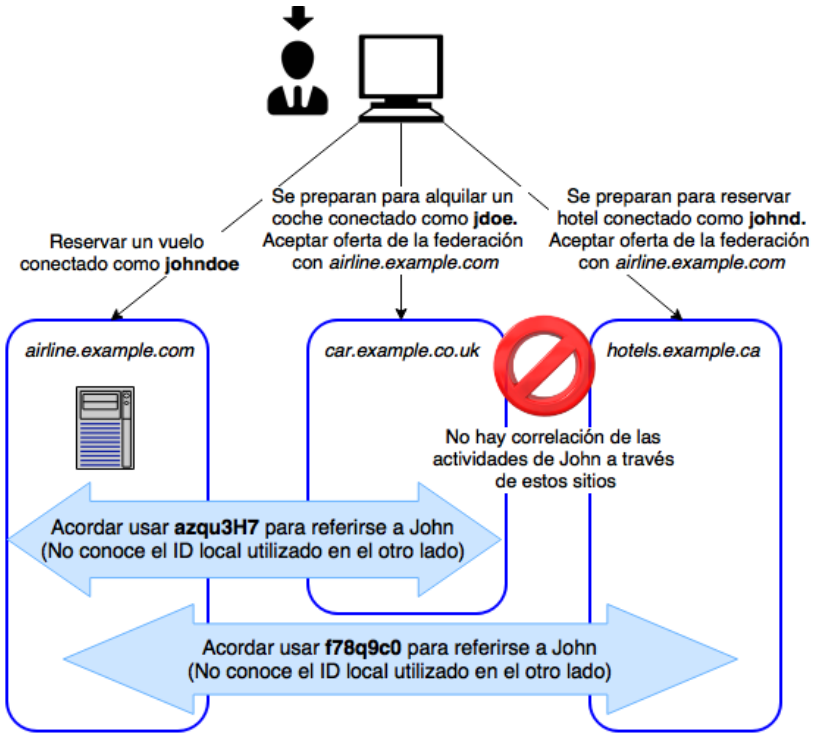

Figura 2.6: Caso de Uso Federación de identidad del protocolo SAML (Hughes y Maler, 2005).

2.3.2. Ventajas

- **Neutralidad de la plataforma:** SAML logra abstraer el marco de seguridad de las aplicaciones particulares de otros proveedores, logrando de esta forma que la seguridad sea independiente del resto de la lógica de la aplicación, esto tomando como principio la arquitectura orientada a servicios.
- **Acoplamiento flexible de directorios:** SAML no requiere que la información de los usuarios sea mantenida y sincronizada entre diferentes directorios.
- **Experiencia en línea mejorada para el usuario final:** permite que el usuario realice un inicio de sesión único y pueda acceder a diversos servicios y recursos sin necesidad de realizar ninguna autenticación adicional.
- **Reduce el costo administrativo para los SP:** los diferentes SP ya no tienen que mantener la información de acceso de los usuarios, reduciendo el costo de

mantenimiento de esa información.

- **Transferencia de riesgo:** la gestión adecuada de los datos de identidad es tarea del IdP, lo que es más compatible con su modelo de negocio que con el de un SP.

2.3.3. Componentes

2.3.3.1. Afirmaciones (Assertions)

Las afirmaciones, o afirmaciones de seguridad (*security assertions*) indican cuando los datos personales del usuario son intercambiados. Una afirmación lleva tanto el identificador como los atributos. Cada afirmación está dirigida a un conjunto particular de SP (Scudder y Josang, 2010). Hay tres tipos de afirmaciones que puede hacer SAML:

- ***Autenticación (Authentication):*** indica que un sujeto particular fue autenticado por una solicitud particular en una hora particular. Esta afirmación generalmente es realizada por un IdP.
- ***Atributo (Attribute):*** especifica los atributos para el sujeto en particular.
- ***Desición de autorización:*** indica que para el sujeto en particular se le ha concedido o denegado permisos para el acceso a un recurso en particular.

En la figura 2.7 se muestra un ejemplo de como puede ser transmitida una afirmación del protocolo SAML que conteniene una serie de declaraciones dentro de una respuesta SAML, que a su vez está empaquetada en un protocolo de transporte.

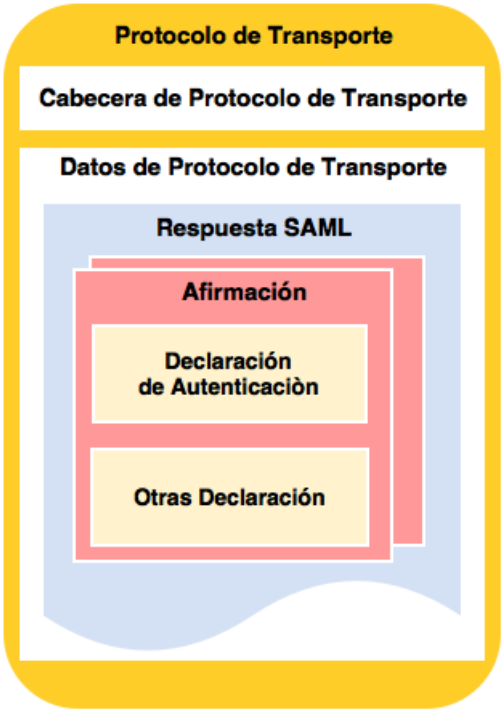

Figura 2.7: Interacción entre los componentes de SAML y el protocolo de transporte (Hughes y Maler, 2005).

2.3.3.2. Protocolos

SAML define un número de protocolos de solicitud/respuesta para permitirle a los SP realizar diferentes acciones, tales como:

- Solicitar de una autoridad alguna afirmación.
- Solicitar a un IdP que autentique una entidad y devuelva una afirmación como respuesta.
- Solicitar el registro o la terminación del uso de un identificador.
- Solicitar el cierre de sesión único.
- Solicitar el mappeo de un identificador.

2.3.3.3. Enlaces (Bindings)

Son definidos como asignaciones del protocolo SAML a otros protocolos de mensajería y comunicación estándar, para poder realizar la comunicación de forma interoperable entre los diferentes elementos. Un ejemplo de esto es el SOAP SAML Binding, que define la forma en la que los mensajes del protocolo SAML pueden ser comunicados en el formato de mensajes SOAP, como se muestra en la figura 2.8.

Figura 2.8: SAML: Interacción con otros protocolos (Hughes y Maler, 2005).

2.3.3.4. Perfiles (Profiles)

Los perfiles en SAML son los que definen las restricciones y las extensiones para el soporte del uso de SAML en diferentes aplicaciones con la finalidad de mejorar la interoperabilidad. Un ejemplo claro de esto es el perfil para Web SSO, que define como SAML comunica las afirmaciones entre IdP y SP para el SSO de un usuario en un navegador (Hughes y Maler, 2005).

2.4. Shibboleth

Es un paquete se software de fuente abierta diseñado especialmente para web SSO tanto dentro de una organización como a través de diferentes organizaciones. Con el uso de ***Shibboleth*** los sitios web pueden tomar decisiones de autorización para el acceso a sus recursos basandose en la información de seguridad que se tiene disponible con este software, preservando de este modo la seguridad de sus sitios. ***Shibboleth*** utiliza estándares para federaciones de identidad, específicamente SAML, para dar soporte al intercambio de atributos(Baldoni, 2012).

Shibboleth nace como iniciativa entre los miembros del consorcio Internet2, que en su grupo de trabajo de seguridad abordan el problema del manejo de identidad desarrollando *Shibboleth* como un proyecto, que más adelante se convierte en este producto. Fue concebido como una forma de facilitar la colaboración y el acceso a los recursos protegidos entre las instituciones sin utilizar las cuentas externas o temporales. Algunas aplicaciones que se ven beneficiadas con la adopción de esta solución son el acceso a la información de base de datos de la biblioteca, cursos a distancia, aplicaciones de colaboración para el desarrollo de proyectos, entre otros (Fragoso-Rodriguez et al., 2006).

En Shibboleth, la información sobre la identidad digital de los usuarios se administra en la institución a la que pertenecen. Cuando un usuario requiere acceso a los recursos que se encuentran en otra institución, los atributos de identidad son enviados junto con la solicitud, tomando en cuenta los acuerdos preexistente sobre los atributos que serán compartidos. Estos atributos se utilizan para finalmente tomar decisiones de aprobación o rechazo de la solicitud de acceso del usuario de acuerdo con la política de control de acceso local. *Shibboleth* permite crear una estructura segura que simplifica la gestion de identidades, le facilita al usuario el acceso recursos de proveedores de diferentes organizaciones que pertenezcan a una misma federación (Leandro et al., 2012).

Siguiendo el esquema de una arquitectura de identidad federada ***Shibboleth*** cuenta con los elementos descritos a continuación (Leandro et al., 2012).

- **Proveedor de identidad (IdP):** Como se dijo anteriormente, el IdP es el responsable de la autenticación de los usuarios, mantiene y controla las credenciales y atributos de éstos, y difunde ésta información cuando es requerida por organizaciones dentro de su circulo de confianza. En particular un IdP de ***Shibboleth*** tiene cuatro componentes:
  - **Handle Service (HS):** autentica a los usuarios y junto al mecanizmo de autenticación crea un *handle token*, que es una afirmación SAML que porta las credenciales. Este servicio le permite a las organizaciones seleccionar el mecanismo de autenticación.
  - **Attribute Authority (AA):** maneja las solicitudes de atributos que vienen de los SP, y aplica politicas para la liberación de estos atributos (con un mecanismo llamado *Attribute Release Policies* - *ARP*). Este componente permite que el usuario especifíque quien puede acceder a sus atributos, así como también le permite a la organización decidir que servicio de directorio es utilizado.
  - **Servicio de directorio:** aunque este servicio es externo a ***Shibboleth*** es un componente fundamental para el IdP, y especifica el almacenamiento local de los atributos de los usuarios.
  - **Mecanismo de autenticación:** este servicio también es externo, y es el que permite a los usuarios autenticarse con el servicio central sólo utilizando su nombre de usuario y contraseña.
- **Proveedor de servicios (SP):** el SP es donde los recursos protegidos, a los cuales el usuario desea acceder, se encuentran almacenados. El SP realiza acciones de control de acceso basado en la información de identidad enviada por el IdP. Un solo SP puede tener varias aplicaciones, sin embargo será tratado como una sola entidad por el IdP, es por esto que la responsabilidad del control de acceso dentro del SP es delegada a las aplicaciones. Un SP tiene tres componentes principales:

  - **Assertion Consumer Service (ACS):** éste es el responsable de recibir los mensajes (bajo el formato SAML) para establecer un entorno seguro para el SSO.
  - **Attribute Requester (AR):** es el responsable de obtener y enviar al gestor de recursos los atributos del usuario.
  - **Resourse Manager (RM):** intercepta las solicitudes de recursos y toma desiciones de control de acceso basado en los atributos del usuario.

- **Where Are You From (WAYF):** también conocido como *Discovery Service* (Servicio de Descubrimiento), es un componente opcional por lo que no siempre se encuentra en todos los sistemas que despliegan ***Shibboleth***, permite la asociación entre un usuario y una organización. La forma típica del funcionamiento es así: cuando un usuario trata de acceder a un recurso, es redireccionado a una interfaz que solicita que elija su organización de origen, luego de elegida se inicia el proceso de autenticación. Este servicio permite que los usuarios en una federación siempre se autentiquen con el proveedor de identidad de la organización a la que pertenecen, que es una de la principales funciones de una arquitectura federada. Cuando un SP desea ofrecer servicios a usuarios de diferentes IdP el servicio WAYF es de gran ayuda.

En la figura 2.9 se presenta el flujo de operación de ***Shibboleth*** y los pasos se describen a continuación:

- Paso 1: el usuario navega hacia el SP y trata de acceder a un recurso protegido.
- Pasos 2 y 3: *Shibboleth* redirecciona al usuario al servicio WAYF para que seleccione su IdP.
- Paso 4: el usuario accede al IdP.
- Paso 5: el HS en el IdP envía al usuario al sitio de autenticación.
- Pasos 6 y 7: el usuario ingresa sus credenciales de autenticación.
- Paso 8: el HS autentica al usuario, crea un handle token para identificar al usuario.

- Paso 9: se envía el token al AA quien revisa las politicas para liberar atributos, y al ACS quien realiza la revisión y transfiere al AR.
- Paso 10: en este punto la sesión se ha establecido satisfactoriamente.
- Paso 11: el AR solicita los atributos del usuario al IdP
- Paso 12: AA verifica cuales atributos puede el IdP enviar de respuesta a la solicitud.
- Paso 13: el AA envia una respuesta con los atributos y los valores.
- Paso 14: el SP recibe los atributos y estos son enviados al RM quien finalmente carga los recursos para ser mostrados al usuario en el último paso.
- Paso 15: Los recursos protegidos son presentados al usuario.

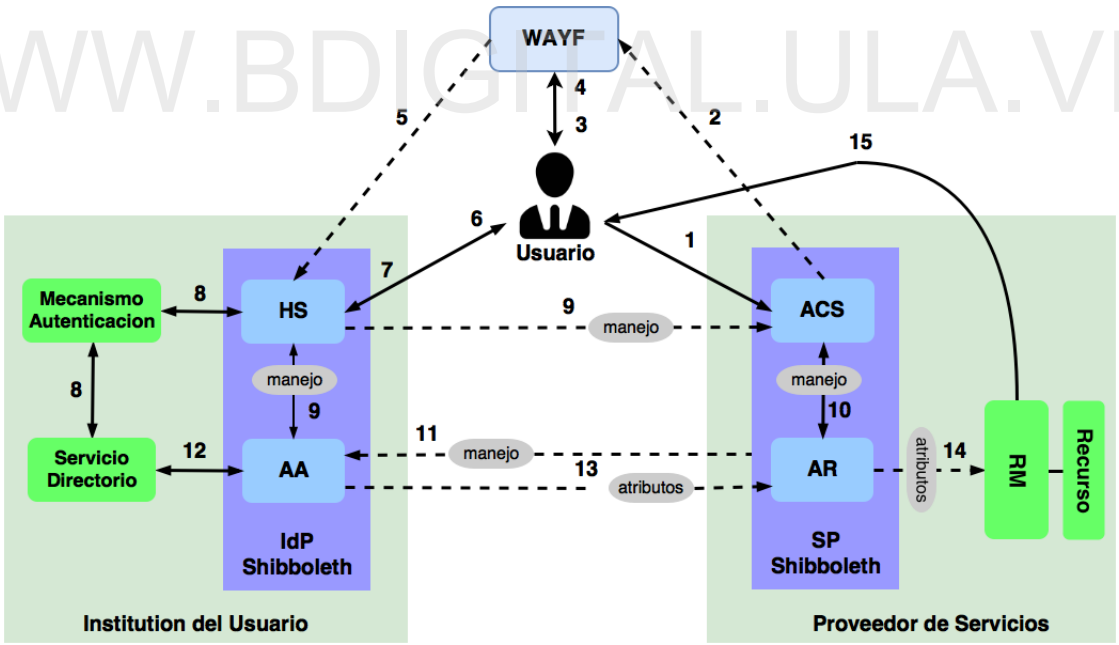

Figura 2.9: Flujo de Operación de Shibboleth (Hughes y Maler, 2005).

El uso de *Shibboleth* va más allá de la colaboración en entornos académicos, trabajos como (Leandro et al., 2012) proponen el uso de *Shibbolet* como herramienta de federación de identidad para para servicios y sistemas en la nube, siendo posible

utilizarlo en forma de "Identidad como servicio" (IDaaS por sus siglas en inglés) o en configuraciones locales donde el comprador de los servicios de la nube decide mantener el control sobre la identidad digital de sus usuarios.

2.5. OAuth

OAuth es un protocolo abierto concebido para transmitir las decisiones de autorización de acceso a recursos, es utilizado frecuentemente junto con otros protocolos para realizar autenticación. Éste protocolo define tres roles principales en el flujo del proceso de autorización:

- **Cliente:** es la aplicación que desea información
- **Servidor de Recursos (Resource Server RS):** es el servicio que desea verificar la identidad del usuario final
- **Servidor de Autorización (Authorization Server AS):** provee un token de acceso

OAuth no es propiamente un protocolo de autenticación en sí, es decir, no provee nunguna información sobre el usuario (Brail y Ramji, 2015), por lo tanto no se puede hacer ninguna afirmación de seguridad (*security assertion*) sobre la identidad del usuario.

2.6. OpenId

OpenId es un "Estándar abierto y un protocolo de autenticación descentralizado" (OpenID Connect - Wikipedia, 2015) Este estándar permite que un usuario pueda identificarse en diferentes sitios web utilizando un proveedor de identidad con soporte a dicho protocolo. La especificación de *OpenId* define tres roles principales:

- **Usuario final:** entidad que desea verificar su identidad
- **Parte confidente (Reliying Party RP):** es el servicio que desea verificar la identidad del usuario final
- **Proveedor de identidad (OpenId OP):** es la entidad que puede verificar la identidad del usuario

2.7. Portal Cautivo

Un portal cautivo es un sistema basado en una página web a través de la cual se controla el acceso de los usuarios a una infraestructura de servicios de red. Se usa generalmente en plataformas de redes inalámbricas públicas o privadas. El sistema del portal cautivo ofrece a los administradores de la red el control sobre sesiones, autenticación, interfaces, consumo, calidad de servicio, entre otras.
Los clientes de un portal cautivo pueden pasar por tres estados: (NETGEAR Support, 2016)

- **Desconocido:** El cliente aún no ha sido redirigido a la página web de autenticación.
- **No autenticado:** El cliente es redirigido a la página web de autenticación.
- **Autenticado:** Si la autenticación es exitosa, el cliente pasa al estado autenticado y tiene acceso a la red.

El mecanismo del portal cautivo puede variar entre los diferentes proveedores que implementan este servicio.

2.7.1. ChilliSpot

Es un proyecto open source para portal cautivo de redes inalámbricas. El mecanismo mediante el cual realiza las tareas de autenticación, autorización y control de cuentas es utilizando un servidor *Radius* (ChilliSpot, 2015).

2.7.2. CoovaChilli

Al igual que su antecesor ChilliSpot, es un software de código abierto para el control de acceso, fue desarrollado por uno de los desarrolladores de ChilliSpot y es quien aún lo mantiene activo (CoovaChilli, 2015). Éste software no maneja ningún estándar para SSO.

2.7.3. ZeroShell

Es una distribución de Linux que integra diferentes soluciones que proporcionan los principales servicios necesarios para el funcionamiento y gestión de una red LAN. ZeroShell no se basa en otra distribución, sino que todo el software utlizado se recopila y empaqueta usando las directrices de *Linux From Scratch*. El nombre Zeroshell indica que es un sistema Linux tradicionalmente administrable desde una consola (*shell*), sin embargo todas las operaciones de administración se pueden realizar a través de interfaz web a la que se puede acceder una vez configurada la interfaz administrativa y desde cualquier navegador web. (Zeroshell, 2015)

Entre las principales características para aplicaciones de red se encuentran:

- Balanceo de carga y conmutación.
- Conexión a internet por tecnología celular (UMTS/HSDPA).
- Servidor RADIUS para proporcionar una autenticación segura y la gestión automática de las claves de cifrado para el Wireless 802.11b, 802.11g y 802.11a, así como de redes que soportan el protocolo 802.1x en la forma EAP-TLS, EAP-TTLS y PEAP .
- Portal Cautivo para inicio de sesión web en redes cableadas e inalámbricas. Zeroshell actúa como la puerta de enlace de las redes en las que el portal cautivo está activo, éstas redes generalmente tienen dirección IP privada y se les asigna dinámicamente por el servidor DHCP. Para que un cliente tenga acceso a esta red privada se debe autenticar a través de un navegador web proporcionando un

usuario y contraseña para que el firewall de Zeroshell le permita el acceso a la LAN.

- Calidad de servicio (QoS) y de gestión de tráfico para controlar el tráfico en la red. Con esta herramienta es posible garantizar el ancho de banda mínimo, limitar el ancho de banda máximo y asignar una prioridad a una clase de tráfico (por ejemplo para aplicaciones que utilizan VoIP). También es posible realizar clasificación de tráfico e inspección de paquetes.
- Servidor proxy de HTTP capaz de bloquear las páginas web que contienen virus con la solución antivirus ClamAV y el servidor proxy HAVP. El proxy funciona de forma transparente al cliente puesto que las peticiones http son redirigidas automáticamente.
- Enrutador con rutas estáticas y dinámicas bajo el protocolo RIPv2.
- Filtrado de paquetes con cortafuegos en todas las interfaces.
- Servicio NAT en la red LAN.
- Servidor DNS miltizona.
- NTP (Network Time Protocol) tanto cliente como servidor.
- Servidor Syslog para la recepción y catalogación de los registros del sistema producido por los hosts remotos.
- Autorización con LDAP, NIS y RADIUS.

2.8. Virtualización

El término de virtualización de forma general se refiere a la simulación de un escenario o recurso que no existe en el mundo real, siendo este recurso indistinguible de un recurso real dentro de un escenario virtualizado. En computación es un concepto que se ha tenido siempre muy presente, un ejemplo de esto es la memoria virtual que conceptualmente permite utilizar más memoria de la fisicamente disponible. En

el el área de sistemas la virtualización puede ser aplicada a servidores, aplicaciones, almacenamiento y redes, con la ventaja de que reduce costos de operación y aumenta la escalabilidad de los sistemas. Dependiendo del tipo de recurso a virtualizar, se clasifica en (Kusnetzky, 2011):

- **Virtualización de servidores:** se trata de utilizar un software que crea máquinas virtuales (VM) que emulan un equipo físico con lo que se crea "un entorno de sistema operativo independiente que es, lógicamente, aislado del servidor anfitrión. Tener múltiples máquinas virtuales permite que varios sistemas operativos corran de forma simultánea en una única máquina física" (Margaret Rouse, 2016). Según algunos proveedores (VMware, 2016) un servidor no pasa del 15 % de su capacidad, por lo que la virtualización representa una mejora en la utilización de los recursos, aumentando el rendimiento de las aplicaciones y un incremento de la disponibilidad .

- **Virtualización de redes:** con la virtualización de redes se realiza una separación del ancho de banda en varios canales que son luego asignados a servicios o dispositivos independientes, teniendo cada uno de estos canales políticas de gestión y enrutamiento independientes (Michelle McNickle, 2016). En una red virtualizada se tienen las mismas funciones que en una red física y además se saca provecho de las ventajas que trae la virtualización.

- **Virtualización de almacenamiento** este tipo de virtualización es otro ejemplo del uso común que se hace de la virtualización en sistemas, se refiere al agrupación de múltiples discos o espacios de almacenamiento físico para formar un solo dispositivo lógico de almacenamiento con mayor capacidad y prestaciones como recuperación ante fallas, crecimiento dinámico del espacio, entre otras. Los avances en esta área apuntan a controlar estos disposivos mediante almacenamiento definido por software (Software - Defined Storage, SDS) para gestionar el almacenamiento de forma independientemente del hardware.

Otra clasificación para la virtualización depende del nivel de abstracción que se hace de los recursos físicos:

- **Virtualización completa (Full Virtualization):** en este tipo de virtualización cada huésped ejecuta su propio sistema operativo y no es necesario realizarle ningún cambio, por lo tanto éste no sabe que está corriendo en una máquina virtual. Para poder realizar este tipo de virtualización es necesario un hipervisor que funciona entre el sistema operativo huésped y el sistema operativo anfitrión, éste hipervisor se encarga de emular todo los componentes necesarios que necesita el huésped, y de gestionar adecuadamente cualquier instrucción, sea privilegiada o no, que el huésped necesite ejecutar. Es necesario tener soprte de virtualización a nivel de hardware para poder hacer virtualización completa.

- **Paravirtualización (Paravirtualization):** en este tipo de virtualización cada huésped ejecuta su propio sistema operativo, sin embargo es necesario realizar modificaciones para que esto sea posible. Existe un hipervisor más ligero que el utilizado en la virtualización completa. Las instrucciones privilegiadas son manejadas por el hipervisor y el resto son enviadas directamente al hardware, esto causa menor impacto en el rendimiento y permite una mejor utilización de los recursos.

- **Virtualización por contenedores (OS Virtualization):** este tipo de virtualización facilita la ejecución de ambientes aislados entre si pero que comparten el mismo núcleo. Se configura cada contenedor y se le asigna los recursos que puede utilizar en sus procesos, éstos recursos son compartidos con los demás contenedores.

2.8.1. Hipervisor XEN

Es una herramienta de fuente abierta, desarrollada en la Universidad de Cambridge con colaboración de varias compañías com AMD, Cisco, Dell, HP, IBM, Intel, entre otras. Xen puede ser utilizado para virtualización completa (tomando en cuenta que requiere soporte en hardware) y paravirtualización, y se encuentra disponible para varias arquitecturas de procesador. El hipervisor de Xen funciona de manera independiente al sistema base y reside entre el hardware y el sistema base (Xen Project, 2016).

2.8.1.1. Arquitectura de Xen:

- Permite que el controlador de dispositivo principal de un sistema pueda funcionar dentro de una máquina virtual, puesto que realiza un aislamiento del los controladores.
- En equipos sin soporte para virtualización en hardware se puede implementar paravirtualización.
- El hipervisor corre directamente sobre en hardware y maneja los recursos de memoria y procesamiento e interrupciones.
- Sobre el hipervisor se ejecutan las diferentes instancias de máquina virtual, son llamadas dominios o huésped. Existe una especial llamada dominio 0 el cual es es dominio de control, donde se pueden crear, eliminar y realizar diferentes tipo de operaciones con el resto de los dominios. A través de este dominio se controla todo el sistema

2.9. Mail User Agent

El envío de un mensaje de correo electrónico están involucrados diferentes componentes que realizan tareas muy específicas en determinados momentos. Uno de estos componentes es el *Mail User Agent* (MUA), que es donde generalmente un correo comienza y termina su ciclo de vida. EL MUA es el encargado de hacer la interfaz con el usuario, cuando un usuario utiliza una aplicación para leer su correo o para escribir y enviar uno, está haciendo uso de un MUA. Algunas de las aplicaciones más populares son: Microsoft Outlook, Icedove, Thunderbird. Cuando un usuario entra en a una dirección web y abre a su correo mediante un navegador web también está haciendo uso de un MUA, incluso cuando desde una consola se usa directamente un programa como sendmail para enviar un correo, se está usando sendmail como un MUA.En resumen el término MUA se usa como la descripción formal de los programas con los

cuales un usuario crea, envía y lee su correo electrónico.(Vicki Stanfield, Roderick W. Smith, 2006)

2.9.1. Squirrelmail

SquirrelMail es un paquete de correo web desarrollado en PHP, con este lenguaje tiene funciones de soporte para los protocolos IMAP y SMTP. Una de las mayores ventajas que ofrece es que tiene pocos requisitos y la instalación es relativamente sencilla. SquirrelMail cuenta con todas las funcionalidades básicas necesarias de un cliente de correo electrónico, algunas de ellas son: soporte MIME, libreta de direcciones, y la manipulación de carpetas.
El origen de Squirrelmail se remonta al año 1999 como un proyecto dirigido a muy pocos usuarios, aproximadamente el 5 % de los usuarios de correo en el mundo en ese entonces. Para lograr este propósito los desarrolladores establecieron unos princios: Squirrelmail sería muy fácil de instalar y administrar, tendría pocos requerimientos tanto del lado del cliente como del servidor, y no utilizaría extensiones del lenguaje PHP adicionales ni Javascript ni HTML dinámico, puesto que sus usuarios no necesitan nada de esas cosas llamativas en el webmail, simplemente algo que funcione y sea administrable. No obstante los principios básicos establecidos al comienzo, el proyecto creció a medida que se agregaron nuevas características como libreta de direcciones, soporte a pluggins, entre otras (SquirrelMail, 2015).

2.9.2. Internet Mail Access Protocol (IMAP)

IMAP es un protocolo de capa de aplicación utilizado como alternativa al protocolo POP (*Post Office Protocol*) y que permite la sincronización de los correos electrónicos. Esta sincronización consiste en acceder uno a uno a todos los correos ya sea desde clientes de correo o directamente en el servidor y mantener en ambos sistemas la data actualizada. IMAP hace uso del protocolo TCP para el envío confiable de la información.

2.9.2.1. Dovecot

Dovecot es una solución de código abierto para servicios IMAP y POP, desarrollado con la finalidad de ofrecer un software seguro y de implementación sencilla, cumpliendo con los estándares para este tipo de servicios. Dovecot provee alto rendimiento en comparación con otros software existentes para éste fin (Dovecot, 2016). Entre sus caracteristicas más relevantes se tienen:

- Índices auto-optimizables que contienen lo que generalmente los clientes necesitan.
- Auto-reparación de problemas, con lo que trata de corregir los problemas que el puede detectar como por ejemplo índices rotos.
- Sencillo de implementar desde el punto de vista del administrador del sistema.
- Permite que los buzones y sus índices sean manipulados por varios equipos al mismo tiempo sin afectar su rendimiento, por lo tanto puede ser implementado en sistemas de archivo distribuidos.
- Flexibilidad para la autenticación de los usuarios, tiene soporte para diversos mecanismos de autenticación y bases de datos.
- Cuenta con mecanismos para la fácil migración de otros sistemas IMAP y POP, permitiendo realizar cambios de forma transparente para los usuarios.

2.10. Pruebas de desempeño

2.10.0.2. Apache JMeter

JMeter es una herramienta de código abierto desarrollada en java y concebida para realizar pruebas de rendimiento en aplicaciones web. Permite realizar muestreo simultáneo y también simular diversos escenarios de los servicios, por ejemplo: carga en los servidores, número de usuarios, entre otros. Ofrece la capacidad de exportar los resultados de estas pruebas para ser analizados por diferentes sistemas que realicen correlación y análisis de los resultados. (JMeter, 2016)

Capítulo 3

Diseño e implementación

En este capítulo se detalla todo el proceso de diseño e implementación de la infraestructura de identidad federada, es importante destacar que fue necesario evaluar software preexistente, así como tomar en cuenta la interoperabilidad de la solución planteada con los diversos sistemas que actualmente operan en la plataforma de servicios de la red de datos de la universidad, por lo tanto para cumplir con dichos requerimientos fue necesario engranar diferentes soluciones y realizar algunas adaptaciones.

3.1. Descripción de la metodología utilizada

Dada la naturaleza del proyecto, se decidió seguir una metodología de desarrollo secuencial donde el resultado de cada fase es utilizado como entrada para continuar el desarrollo en la fase siguiente, utilizando como apoyo para la planificación y el seguimiento de las actividades una metodología ágil como kanban, que permitió dividir en pequeñas tareas cada fase, así como también llevar un amplio control de todas las etapas del desarrollo del proyecto.

3.2. Descripción de las tareas realizadas durante el proyecto

Las diferentes actividades realizadas en el diseño de la solución, se dividen en seis grupos con actividades de menor complejidad que permitieron ir implementando secciones particulares de la solución, dichos grupos son los siguientes:

- Grupo 1: Evaluación de requerimientos
- Grupo 2: Diseño de la arquitectura, selección del protocolo y software a implementar.
- Grupo 3: Instalación de la plataforma inicial para elproveedor de identidad y los diferentes proveedores de servicios.
- Grupo 4: Instalación y configuración del servicio de directorio de pruebas y el proveedor de identidad.
- Grupo 5: Instalación y configuración del proveedor de servicios para el webmail.
- Grupo 6: Instalación y configuración del proveedor de servicios para el servicio de portal cautivo para la zona wifi.

3.2.1. Grupo 1: Evaluación de requerimientos

El primer paso para el desarrollo del proyecto fue definir las bases por las que se rige el diseño y la implementación de la infraestructura de identidad digital, determinando las necesidades y requerimientos de RedULA en cuanto al manejo de la identidad digital de sus usuarios. Para obtener estos requerimientos fue necesario evaluar las necesidades de acuerdo con la visión de los diferentes interesados de la siguiente manera:

- **Requerimientos desde el punto de vista de la coordinación:**
  - La infraestructura a implementar debe utilizar estándares conocidos y se debe adaptar a la naturaleza académica y de investigación de RedULA.

- La infraestructura debe ser compatible con la plataforma actual de comunicación y servicios de ReDULA, tanto en software como en hardware.
- La infraestructura a implementar debe permitirle a la institución unirse a federaciones académicas y de investigación tanto dentro como fuera del país.
- La infraestructura debe asegurar el acceso de los usuarios autorizados así como impedir el acceso no autorizado a los sistemas de información de la Universidad de Los Andes en plena conformidad con los objetivos de control de acceso estipulados en la norma internacional ISO27001 y cualquier legislación que tenga su base legal en referencia al contenido de esta norma.
- Los procedimientos implementados en la infraestructura deben dar cumplimiento con el Proyecto de Ley de Protección de Datos Personales y Habeas Data en Venezuela.

- **Requerimientos desde el punto de vista del administrador de servicios:**
  - *La infraestructura se debe implementar en un entorno virtualizado:* La virtualización se ha adoptado como una de las políticas de administración de servicios tanto para la optimización de los recursos disponibles en la dependencia, como para la recuperación en caso de fallas, es por esto que se requiere tomar en consideración el uso de servidores en máquinas virtuales para desplegar la solución.
  - *Los paquetes de software necesarios para la infraestructura deben instalarse bajo el sistema operativo que la dependencia utiliza para sus servicios:* Una manera de mantener una plataforma estándar para los servicios de la red de datos ha sido la utilización de forma homogénea del sistema operativo base sobre el cual se despliegan los servicios.
  - *La infraestructura a implementar se debe ser interoperable con los servicios actuales de directorio y el servicio de webmail:* los cambios necesarios en los servicios que se encuentran en operación actualmente deben ser mínimos para adaptarse a la infraestructura propuesta.

- **Requerimientos desde el punto de vista del usuario de la red de datos:**

- *El usuario debe poder utilizar un único par de credenciales (nombre de usuario y contraseña) para el acceso a los diferentes servicios:* una característica muy importante que debe ofrecer la nueva infraestructura de identidad es permitir al usuario utilizar las mismas credenciales de acceso en diversos servicios haciendo un inicio de sesión único.
- *El cambio del mecanismo de autenticación de la zona wifi debe ser transparente para el usuario:* El acceso a la zona wifi se debe hacer por medio de la infraestructura de identidad digital utilizando un portal cautivo.

3.2.2. Grupo 2: Diseño de la arquitectura, selección del protocolo y software a implementar

El siguiente paso para el diseño y posterior implementación y desarrollo de una solución para la gestión de identidad digital, fue tomar en consideración la naturaleza de uso académico y de investigación de RedULA, es por esto que entre los estándares y lineamientos a seguir se tomó como marco de referencia las buenas prácticas para federaciones de identidad establecidas por organismos y proyectos académicos y de investigación como Geant y su servicio eduGain, (eduGAIN, 2015), así como otras federaciones de redes académicas, algunas de ellas muy conocidas como la federación de identidad que reune instituciones de educación e investigación en Brazil (Rede Nacional de Ensino e Pesquisa, 2015) y la red académica y de investigación Española RedIRIS (RedIRIS, 2016)

Un tema importante a considerar para la implementación de la solución es el hecho de poder escalar y más adelante formar parte de federación, es por esto que se considera que la infraestructura de identidad digital que se implemente debe ser tener una arquitectura federada.

Tomando lo anterior como punto de partida, el diseño que mejor se adecua tanto a los requerimientos como a los recursos disponibles es el de una arquitectura con dominio centralizado de identidad federada como se plantea en la figura 3.1. Este

diseño contempla un servidor para el IdP que estará conectado con el servicio de directorio y será el que maneje lo mecanismos de autenticación de la identidad digital de los usuarios, un servidor para el SP del servicio *Webmail*, y un servidor para el SP encargado del acceso a la zona wifi.

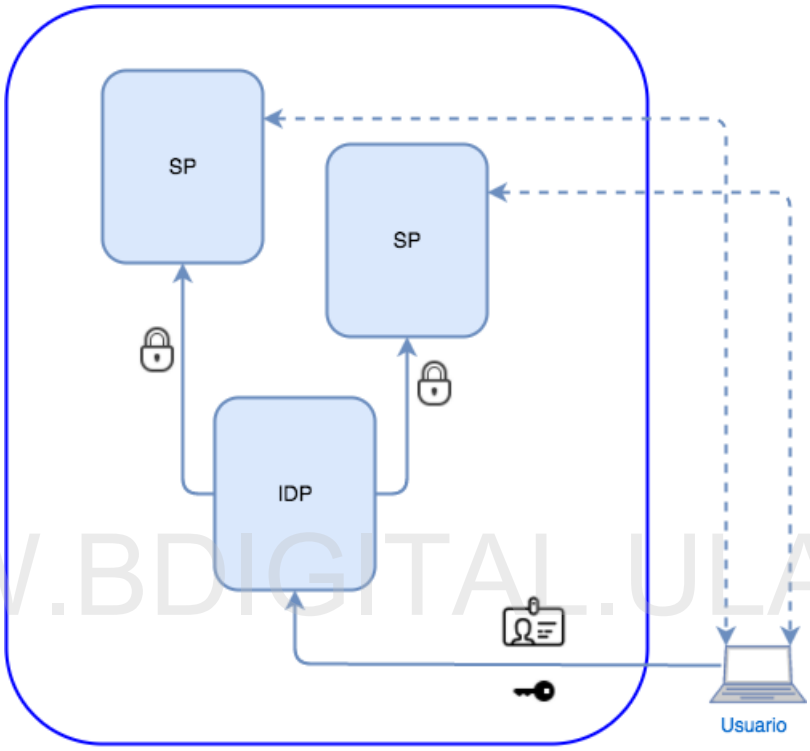

Figura 3.1: Diseño propuesto basado en arquitectura de dominio centralizado de identidad.

Las características de cada uno de estos elementos son descritas más adelante en este capítulo.

En este sentido la investigación sobre las tecnologías y el software a ser implementado se orientaron hacia los que son utilizados actualmente en diversas instituciones académicas que forman parte de federaciones de identidad. Entre las tecnologías estudiadas tenemos:

- **OAuth**
- **OpenId Connect**

- **SAML**

Oauth está mas orientado a gestionar la autorización de acceso a los recursos que la autenticación, por esta razón es un protocolo que necesita ser implementado junto con otras capas de protocolos para poder cumplir con el manejo de la identidad digital de los usuarios.

Por otro parte OpenId Connect posee características muy similares a las que se desean tener en la infraestructura , sin embargo no es un protocolo utilizado ampliamente en el área de las redes académicas.

Paralelamente entre las tecnologías utilizadas para federación de identidad encontramos *SAML*, que se ha conocido por tener un modelo de seguridad y privacidad robusto, y ha demostrado ser interoperable con múltiples plataformas. Aunado a esto tenemos que es la tecnología elegida para la mayoría de las aplicaciones de federación utilizadas en educación superior y gobierno electrónico (Identity, 2016) por lo cual es la tecnología que se utilizará para la infraestructura de identidad digital de RedULA.

Una vez seleccionado *SAML* como la tecnología a implementar, llegamos a *Shibboleth* quien se ha convertido en la implementación de facto de *SAML*, y una solución de identidad federada open source (McLaughlin et al., 2010), siendo el más utilizado en el entorno académico y de investigación (Broeder et al., 2012).

3.2.3. Grupo 3: Instalación de la plataforma inicial para el IdP y los diferentes SP

Para comenzar a implementar la solución fue necesario diseñar y planificar la estructura de los servidores que alojarán los diferentes elementos necesarios para la infraestructura de identidad federada. Siguiendo los estándares para una arquitectura de identidad federada y los elementos necesarios para instalar y operar con *Shibboleth* tenemos que los requerimientos mínimos para la plataforma son:

- Un servidor para el servicio de directorio de usuarios

- Un servidor para el proveedor de identidad
- Un servidor para el proveedor de servicio para el webmail
- Un servidor para el proveedor de servicio de la zona wifi (Territorio digital)

Es necesario señalar que todo el dimensionamiento para esta plataforma se realizó tomando como punto de partida la virtualización de dichos servicios, esto significa que cada uno de los servidores, exceptuando el proveedor de servicio de la zona wifi, fueron instalados como sistemas húesped en diversos equipos bajo el hipervisor XEN. La virtualización de éstos servicios ofrece varias ventajas como aprovechar mejor los recursos computacionales, replicar los servicios instalados en las máquinas virtuales de forma sencilla e implementar políticas de alta disponibilidad lo que permite tener una respuesta más rápida y eficaz ante posibles fallas.
El sistema operativo utilizado para el servicio de directorio, proveedor de identidad y el proveedor de servicio para el webmail es Debian 8. Para el proveedor de servicio de la zona wifi se utilizó ZeroShell 3.1

La figura 3.2 describe en detalle cómo se alojan los sistemas huésped en el equipo anfitrión. Las máquinas huésped se ejecutan sobre un sistema hipervisor en el equipo anfitrión, a su vez el software necesario para el IdP y para el SP del *Webmail* se ejecutarán en un servidor huésped cada uno.

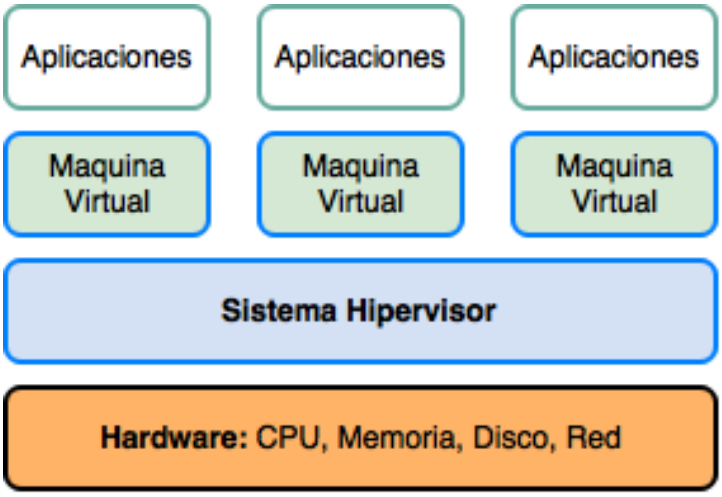

Figura 3.2: Esquema de varias máquinas virtuales en un equipo anfitrión.

Para el servicio de directorio fue necesaria la instalación de un servidor con *OpenLDAP* que mantuviera la configuración y los esquemas del servidor principal que actualmente está en producción y sobre el cual se pudiesen realizar pruebas y agregar nuevos esquemas. Como se mencionó en el capítulo dos, el proveedor de identidad hace afirmaciones de seguridad sobre la identidad de los usuarios y dichos datos para los usuarios de la red están almacenados en un servidor de directorio *OpenLDAP*.
En el momento de la implementación de la infraestructura de identidad el servicio de directorio de la Universidad se encuentra configurado de la siguiente manera: un servidor principal y cuatro réplicas. Por ser éste un servicio crítico para el funcionamiento del correo electrónico se realizó la instalación de un servidor de pruebas donde se realizaron los cambios necesarios en el servicio de directorio para la integración con la infraestructura, éstos cambios son explicados más adelante.

A continuación se detallan las especificaciones de los servidores que estaban a dispisición para ser utilizados en la infraestructura de identidad digital:

Servidor anfitrion que aloja el servicio de directorio:

- **Procesador:** Intel Quad core 2.66 GHz
- **Memoria RAM:** 4Gb
- **Sistema Operativo:** Debian 7 *Wheezy*

Servidor anfitrion que aloja el servicio de proveedor de identidad:

- **Procesador:** Intel Quad core 2.66 GHz
- **Memoria RAM:** 4Gb
- **Sistema Operativo:** Debian 7 *Wheezy*

Servidor anfitrion que aloja el proveedor de servicio para el webmail:

- **Procesador:** Intel Xeon e5440
- **Memoria RAM:** 16Gb
- **Sistema Operativo:** Debian 8 *Jessie*

Servidor huésped para el servicio de directorio:

- **Procesador:** Un núcleo del procesador del equipo anfitrión
- **Memoria RAM:** 512Mb
- **Sistema Operativo:** Debian 7 *Wheezy*

Servidor huésped para el webmail:

- **Procesador:** Un núcleo del procesador del equipo anfitrión
- **Memoria:** 1Gb
- **Sistema Operativo:** Debian 7 *Wheezy*

Servidor para el proveedor de servicio de la zona wifi (Territorio Digital):

- **Procesador:** Pentium(R) dual core 2.2GHz
- **Memoria:** 2Gb
- **Sistema Operativo:** ZeroShell-3.4.0

3.2.4. Grupo 4: Instalación y configuración del servicio de directorio y el proveedor de identidad (IdP)

Luego de tener definida la arquitectura, se comenzó con la instalación del software necesario para los servidores. El servidor de directorio fue el primero en ser instalado dado que éste es parte fundamental para el funcionamiento del proveedor de identidad. Aún cuando existen varias opciones de opensource para implementar un directorio bajo el protocolo LDAP, para el servicio de directorio de la Red de datos de la Universidad se continuó utilizando *OpenLdap* como software gestor de directorio.

3.2.4.1. OpenLDAP

OpenLDAP es el software más utilizado y ya viene preinstalado en una gran variedad de distribuciones de Linux, sin embargo la decisión de utilizarlo se debe a que es el sistema en el cual opera actualmente el servicio de directorio y, como fue solicitado en los requerimientos, la nueva infraestructura de identidad digital debe adaptarse a los servicios existentes. El servicio de directorio es una parte importante del servicio de correo electrónico, pues alli se almacena la información de todas las cuentas de correo de la Universidad.

La instalación se realizó sin ningún inconveniente, se tuvo en cuenta dos configuraciones particulares para adaptar el servicio de directorio a las necesidades tanto del servicio de correo electrónico como de la infraestructura de identidad. Estas configuraciones son:

- Agregar el esquema especial del correo de la Universidad *mailULA.schema*
- Agregar el esquema para federaciones educativas EduPerson.schema

El cuadro 3.1 muestra un fragmento del archivo mailULA.schema con la definición los atributos *expiryDate* y *status*. De la misma manera en cuadro 3.2 la definición de los atributos *eduPersonNickname* y *eduPersonOrgDN* dentro del esquema eduPerson.

```
attributetype ( Mail_ULA_Attr:6
 NAME 'expiryDate'
 DESC 'Account Expiry Date'
 EQUALITY generalizedTimeMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

attributetype ( Mail_ULA_Attr:7
 NAME 'status'
 DESC 'Account Status'
 EQUALITY caseIgnoreMatch
 SUBSTR caseIgnoreSubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.44 )
```

Cuadro 3.1: Atributos en esquema mailULA.

```
attributeType ( 1.3.6.1.4.1.5923.1.1.1.2
    NAME 'eduPersonNickname'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

attributeType ( 1.3.6.1.4.1.5923.1.1.1.3
    NAME 'eduPersonOrgDN'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY distinguishedNameMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.12'
    SINGLE-VALUE )
```

Cuadro 3.2: Atributos en esquema eduPerson.

El esquema mailULA es un esquema personalizado que agrega campos específicos para el uso del directorio con servicios de RedULA como correo electrónico y Web del profesor, entre otros. El esquema eduPerson es un esquema desarrollado por el consorcio Internet2 para proveer un conjunto de atributos y definiciones estándar utilizados en instituciones de educación superor.

Los detalles de estas configuraciones, así como de toda la instalación del servicio de directorio se encuentran en la documentación en línea en la siguiente dirección http://190.168.24.43/wiki/index.php/Servicio_de_Directorio

Archivo de configuración .

EduPerson.schema

3.2.4.2. Shibboleth

Una vez configurado el servicio de directorio, será el que mantenga los datos de los usuarios de la red, se continuó con la instalación del software para proveedor de identidad. *Shibboleth* es un software open source que permite hacer inicio de sesión

única (Single Sign On) en diferentes servicios dentro de una institución, así como a través de instituciones. Para cumplir con los requisitos de instalación y funcionemiento de shibboleth, se debió instalar primero los siguientes componentes:

- Openjdk-6
- Tomcat
- Apache2
- Módulo de apache libapache2-mod-jk

Aunque existen implementaciones de *Shibboleth* para otros servidores web como Nginx, las mismas son a nivel de proveedor de servicio, aunado a esto y para mantener la estandarización de la plataforma de servicios en RedULA se toma Apache2 como el servidor web implementado para el IdP y los diferentes SP. El software para el proveedor de identidad *Shibboleth* está disponible en la página web de Internet 2 como Shibboleth-IDP. Adicionalmente se debió descargar el archivo bcprov-jdk16-144.jar, una biblioteca que implementa protocolos criptográficos en Java y sustituye a las bibliotecas estándar de OpenJDK. La versión utilizada para el proveedor de identidad fue la estable para el momento de la instalación: 2.3.8.

El IdP de *Shibboleth* utiliza los siguientes archivos de configuración para manejar su operación:

- handler.xml: Contiene el tipo y los métodos de autenticación que serán utilizado por el IdP. El cuadro 3.3 muestra un fragmento de éste archivo dónde se define el tipo de autenticación como "*UsernamePassword*" para el IdP.

```
<!--  Username/password login handler -->
<ph:LoginHandler xsi:type="ph:UsernamePassword"       jaasConfigurationLocation="
    file:///opt/shibboleth-idp/conf/login.config">
<ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0
    :ac:classes:PasswordProtectedTransport</ph:AuthenticationMethod>
</ph:LoginHandler>
<ph:LoginHandler xsi:type="ph:PreviousSession">
   <ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0
       :ac:classes:PreviousSession</ph:AuthenticationMethod>
</ph:LoginHandler>
```

Cuadro 3.3: Fragmento del archivo handler.xml.

- attribute-resolver.xml: Contiene la configuración, codificación y transformación de los diferentes atributos de LDAP a *Shibboleth*. El cuadro 3.4 muestra un fragmento de la configuración en donde se define el atributo *mail* como un atributo a ser mapeado por el IdP.

```
<resolver:AttributeDefinition xsi:type="ad:Simple" id="email" sourceAttributeID=
    "mail">
  <resolver:Dependency ref="myLDAP" />
  <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="
      urn:mace:dir:attribute-def:mail" />
  <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0
      .9.2342.19200300.100.1.3" friendlyName="mail" />
</resolver:AttributeDefinition>
```

Cuadro 3.4: Fragmento del archivo attribute-resolver.xml.

- attribute-filter.xml: Contiene la configuración para la liberación de los atributos dependiendo del proveedor de servicio que los solicita. En el cuadro 3.5 muestra parte de la configuración de éste archivo, en la que se define el atributo *eduPersoAffiliation* como un atributo que puede ser entregado a los diferentes SP.


```
<afp:AttributeRule attributeID="eduPersonAffiliation">
   <afp:PermitValueRule xsi:type="basic:OR">
     <basic:Rule xsi:type="basic:AttributeValueString" value="faculty"
         ignoreCase="true" />
     <basic:Rule xsi:type="basic:AttributeValueString" value="student"
         ignoreCase="true" />
     <basic:Rule xsi:type="basic:AttributeValueString" value="staff" ignoreCase=
         "true" />
     <basic:Rule xsi:type="basic:AttributeValueString" value="alum" ignoreCase="
         true" />
     <basic:Rule xsi:type="basic:AttributeValueString" value="member" ignoreCase
         ="true" />
     <basic:Rule xsi:type="basic:AttributeValueString" value="affiliate"
         ignoreCase="true" />
     <basic:Rule xsi:type="basic:AttributeValueString" value="employee"
         ignoreCase="true" />
     <basic:Rule xsi:type="basic:AttributeValueString" value="library-walk-in"
         ignoreCase="true" />
   </afp:PermitValueRule>
</afp:AttributeRule>
```

Cuadro 3.5: Fragmento del archivo attribute-filter.xml.

- relying-party.xml: Contiene la configuración para el manejo de los mensajes que el IdP recibe de los SP. En éste archivo también se indica donde se almacena el certificado y la llave del proveedor de identidad. La tabla 3.6 muestra un

fragmento de la configuración en donde se define un SP de prueba junto con la ubicación de su archivo de metadata.

```
<metadata:MetadataProvider id="URLMD" xsi:type="
    metadata:FileBackedHTTPMetadataProvider"
  metadataURL="https://190.168.24.44/Shibboleth.sso/Metadata"
  backingFile="/opt/shibboleth-idp/metadata/spwifi-metadata.xml">
  <metadata:MetadataFilter xsi:type="metadata:ChainingFilter">
    <metadata:MetadataFilter xsi:type="metadata:EntityRoleWhiteList">
    <metadata:RetainedRole>samlmd:SPSSODescriptor</metadata:RetainedRole>
  </metadata:MetadataFilter>
 </metadata:MetadataFilter>
</metadata:MetadataProvider>
```

Cuadro 3.6: Fragmento del archivo relying-party.xml.

Los pasos de instalación del IdP y los diferentes archivos de configuración se encuentran en la documentación en línea en la siguiente dirección http://190.168.24.43/wiki/index.php/Proveedor_de_Identidad

3.2.5. Grupo 5: Instalación y configuración del SP para el *webmail*

La siguiente fase para la implementación de la infraestructura de identidad federada corresponde a los diferentes SP, éstos serán quienes tengan la facultad de consultar al IdP los datos de los usuarios. Para todos los SP hay una serie de configuraciones comunes relacionadas con la capa que se comunica con el IdP para la autenticación de los usuarios.

Como ha sido el caso del proveedor de identidad y siguiendo las pautas de los requerimientos, para los proveedores de servicio de RedULA se mantuvo Apache2 como servidor web, y en base a esto se continuó con la instalación del software necesario para la comunicación entre el IdP y los diferentes SP.

Lo primero que se realizó en el IdP fue la generación de un certificado ssl para firmar los mensajes del SP *Shibboleth*. Como se trabajó en un equipo de pruebas donde se instaló el servicio de *webmail* completo también debió generarse un certificado para Apache, el cual es utilizado en el navegador del cliente.

En el cuadro 3.7 se muestra la configuración que se utilizó para la generación del certificado ssl en el SP. En esta configuración se incluyen los datos de la organización responsable del servicio, así como también datos del servidor.

```
[ req ]
default_bits = 2048 # Size of keys
string_mask = nombstr # permitted characters
distinguished_name = req_distinguished_name
[ req_distinguished_name ]
0.organizationName = Universidad de Los Andes
organizationalUnitName = DTES
emailAddress = root@ula.ve
emailAddress_max = 40
localityName = Merida
stateOrProvinceName = MRD
countryName = VE
countryName_min = 2
countryName_max = 2
commonName = sp.ing.ula.ve
commonName_max = 64
organizationalUnitName_default = DTES
countryName_default = VE
commonName_default = sp
```

Cuadro 3.7: Fragmento del archivo openssl.cnf.

El software necesario para la instalación del SP *Shibboleth* se lista a continuación:

- apache2
- libapache2-mod-php5
- libapache2-mod-shib2

El siguiente paso consistió en la configuración del SP. Básicamente toda la configuración se realiza en el archivo Shibboleth2.xml. Hay varias partes del archivo cuya configuración es clave para el correcto funcionamiento del proveedor de servicios y son las siguientes:

- Definición de la entidad del proveedor de servicio: en esta etiqueta se define la URL por la cual se conecta al proveedor de servicio, así como los atributos que el puede recibir como identificador del usuario remoto. El cuadro 3.8 muestra parte de la configuración de esta sección para el proveedor de servicios del webmail, en donde se define la entidad del SP, así como los atributos utilizados para identificar al usuario.

```
<ApplicationDefaults id="default" policyId="default"
        entityID="https://150.185.138.166/shibboleth"
        REMOTE_USER="uid mail givenName eppn"
        signing="false" encryption="false">
```

Cuadro 3.8: Definición de la entidad y atributos de identificación en archivo shibboleth2.xml.

- Definición de la credencial: en esta etiqueta se define el nombre de la llave y el certificado ssl que fue generado anteriormente. El cuadro 3.9 muestra en detalle esta sección en la que se configura la llave y el certificado del SP para el servicio *webmail*.

```
<CredentialResolver type="File" key="/etc/ssl/private/spwebmail.key" certificate
    ="/etc/ssl/certs/spwebmail.crt"/>
```

Cuadro 3.9: Definición del cerfificado y llave en el archivo shibboleth2.xml.

- Definición de la metadata del proveedor de identidad: en esta etiqueta se define el nombre del archivo que contiene la metadata con la información del proveedor de identidad. Esta metadata está disponible una vez se ha configurado el proveedor de identidad y se debe almacenar en un archivo en el proveedor de servicio. En el cuadro 3.10 muestra en detalle como se definen los datos del IdP y la el nombre del archivo que contiene la metadata del mismo.

```
 <MetadataProvider type="XML" uri="https://190.168.63.55/idp/shibboleth"
             backingFilePath="idp-metadata.xml" reloadInterval="7200">
 </MetadataProvider>
```

Cuadro 3.10: Definición de la metadata del IdP en el archivo shibboleth2.xml.

Adicionalmente el arcivo attribute-map.xml contiene el mapeo de los atributos que serán recibidos del IdP, el cuadro 3.11 muestra como se deffine el mapeo para el atributo **eppn**.

```
    <Attribute name="urn:mace:dir:attribute-def:eduPersonPrincipalName" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
```

Cuadro 3.11: Atributos que el SP recibe del IdP definido en attribute-map.xml.

Los pasos de instalación del SP y los diferentes archivos de configuración se encuentran en la documentación en línea en la siguiente dirección http://190.168.24.43/wiki/index.php/Proveedor_de_Servicio

Una vez configurado todo lo relacionado con el proveedor de servicio de *Shibboleth* se continuó con la instalación del Mail User Agent (MUA) que es el conjunto de programas y/o servicios que permiten a los usuarios enviar y leer sus correos electrónicos. El siguiente software fue instalado y configurado para cumplir con ésta tarea:

- **Squirrelmail**
- **Dovecot**

Para adaptar el Squirrelmail a la autenticación del proveedor de servicios se hicieron cambios en el plugin de autenticación en el archivo config.php. El cuadro 3.12 muestra una sección del archivo configuración del plugin de autenticación en la que se configuran los siguientes parámetros:

- El comportamiento del *Webmail* ante una solicitud de inicio de sesión.
- La definición del atributo que se toma como identificador del usuario.
- La variable de ambiente donde el módulo de mod-shib almacena los atributos recibidos del IdP.
- La contraseña maestra para la integración con el servidor IMAP.

```
$normal_login_behavior='https://idp-ula/Shibboleth.sso/Login?return=%e';
$authenticated_username_location = 'uid';
$authenticated_password_location = 'Shib-Session-ID';
$external_auth_validation_type = 'trusted_saml';
$required_environment_variable = 'REMOTE_USER';
$required_environment_variable_value_type = 2;
//mastercredential
$trusted_saml_username = 'masterauth';
$trusted_saml_password = 'U-H6b!Qu';
$authenticated_saml_compress_assertion = 0;
```

Cuadro 3.12: Fragmento del archivo config.php.

El siguiente servicio del **MUA** que requiere autenticación por parte del usuario para su funcionamiento es el *courier*, quien se encarga de mostrar la lista de los correos

que están en el buzón del usuario, así como la información de cada correo que el usuario selecciona.

El courier que tiene el servicio correo en producción para el webmail es *IMAP*, éste fue instalado en un principio para realizar las pruebas de integración con la autenticación del proveedor de servicios sin éxito. Por consiguiente se instaló y configuró *Dovecot* como courier para la maqueta de pruebas. *Dovecot* es más abierto al tipo de autenticación que utiliza, en este caso el confía en los datos de autenticación que el SP ha recibido y mantiene en sesión , por lo que puede acceder al atributo *homeDirectory* y de esta manera tener disponible todo lo necesario para mostrar el buzón de correo al usuario.
Una vez autenticado el usuario, el plugin de autenticación de *Squirrelmail* envia al servicio *courier* (*Dovecot*) un par de credenciales "*master*" con las cuales le indica que se puede confiar en la identidad del usuario y puede proceder a recuperar el buzón del mismo para entonces ser desplegado por *Squirrelmail*. De esta forma se logra la integración del servicio Webmail con la infraestructura de identidad. La figura 3.3 muestra la interacción entre el módulo de *apache* quien es el que realiza la autenticación, el *Squirrelmail* que es la interfaz mediante la cual el usuario accede a su buzón, y finalmente *Dovecot* que es el servicio que permite la recuperación de los correos del usuario.

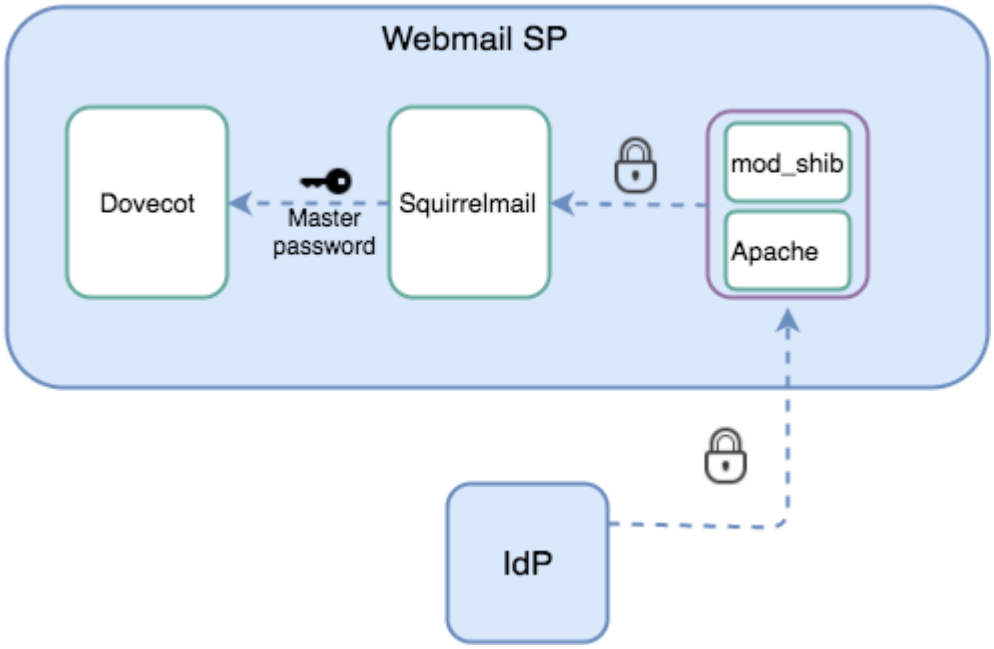

Figura 3.3: Integración del SP con Webmail y Courier.

En el cuadro 3.13 se detalla parte del archivo auth-master.conf en la cual se configura la ubicación del archivo que contiene la contraseña maestra que *Dovecot* espera recibir de *Squirrelmail* para proceder a recuperar en buzón de correo de un usuario.

```
 auth_master_user_separator = *
passdb {
  driver = passwd-file
  args = /etc/dovecot/passwd.masterusers
  master = yes
  pass = no
}
```

Cuadro 3.13: Fragmento del archivo auth-master.conf.

Tanto la instalación de *Squirrelmail* como la de *Dovecot* y las configuraciones se encuentran en la documentación en línea en la siguiente dirección http://190.168.24.43/wiki/index.php/Dovecot_y_Squirrelmail

En la figura 3.4 se muestra cómo está definida la infraestructura de identidad digital. Hasta este punto se ha implementado satisfactoriamente el IdP, se ha configurado con el servicio de directorio y se ha implementado satisfactoriamente el SP para el *Webmail*

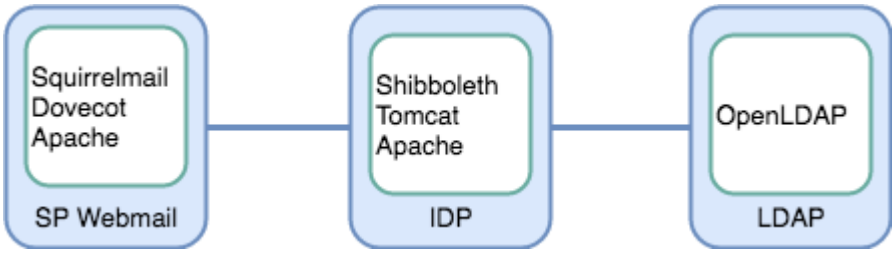

Figura 3.4: Infraestructura de identidad digital con un SP para webmail.

3.2.6. Grupo 6: Instalación y configuración del SP para el servicio de portal cautivo para la zona WiFi

Un servicio que, desde la coordinación de la red de datos, era fundamental integrar con la infraestructura de identidad digital es el servicio de conexión a la zona wifi (también conocido como Territorio Digital). Esta parte de la solución muestra un uso de identidad federada en una aplicación "no web" que es el uso que se ve con más frecuencia en este tipo de soluciones.

En esta parte la investigación se enfocó en buscar soluciones existentes y casos exitosos de uso de autenticación SSO en un portal cautivo para conexión inalámbrica. La investigación se redujo a los siguientes posibles sistemas:

- ChilliSpot
- CoovaChilli
- ZeroShell

Ni ***ChilliSpot*** ni su sucesor ***CoovaChilli*** cuentan con soporte nativo para *Shibboleth*, aunque en la red académica francesa Renater en el año 2011 realizaron las modificaciones necesarias a ***ChilliSpot*** para agregarle ésta autenticación, ambas opciones fueron descartadas en un principio por considerar que el mantenimiento y actualización a los cambios en un software de este tipo podrían representar un problema para los administradores de los servicios en RedULA.

Por su parte ***ZeroShell*** cuenta con soporte para la autenticación web mediante *Shibboleth* en el portal cautivo, lo que representa una ventaja en comparación con las

opciones anteriores.

Con la desición tomada respecto al sistema a implementar en esta fase, se comenzó con la instalación de la distribución ZeroShell en un equipo con las características descritas en el Grupo 2. Las figuras 3.5 y 3.6 muestran detalles del menú principal de la consola administrativa de ZeroShell, y detalles de las opciones de configuración disponibles en la interfaz web.

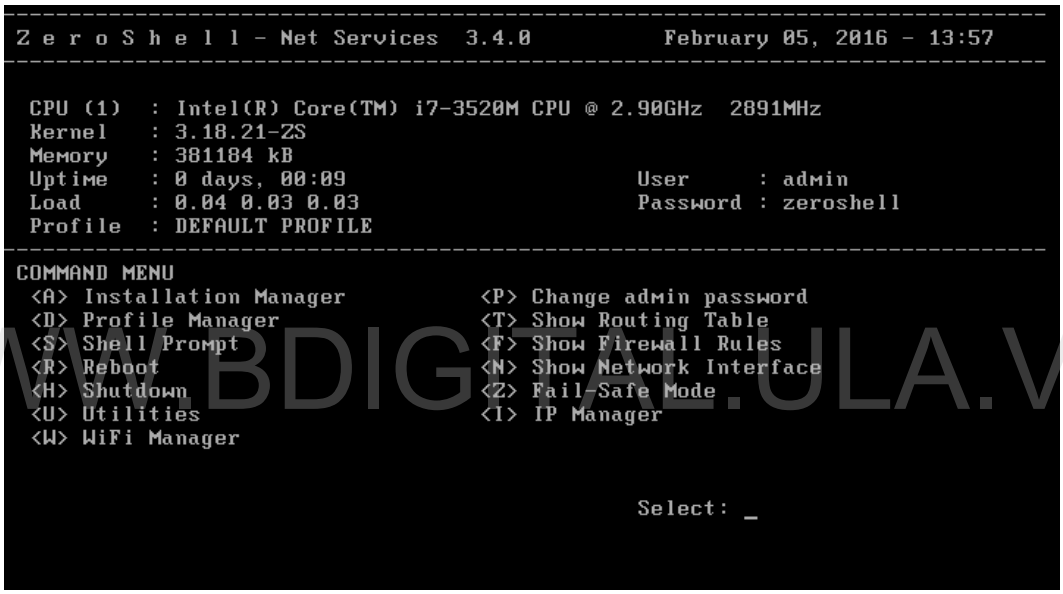

Figura 3.5: Menú principal de la consola administrativa de ZeroShell

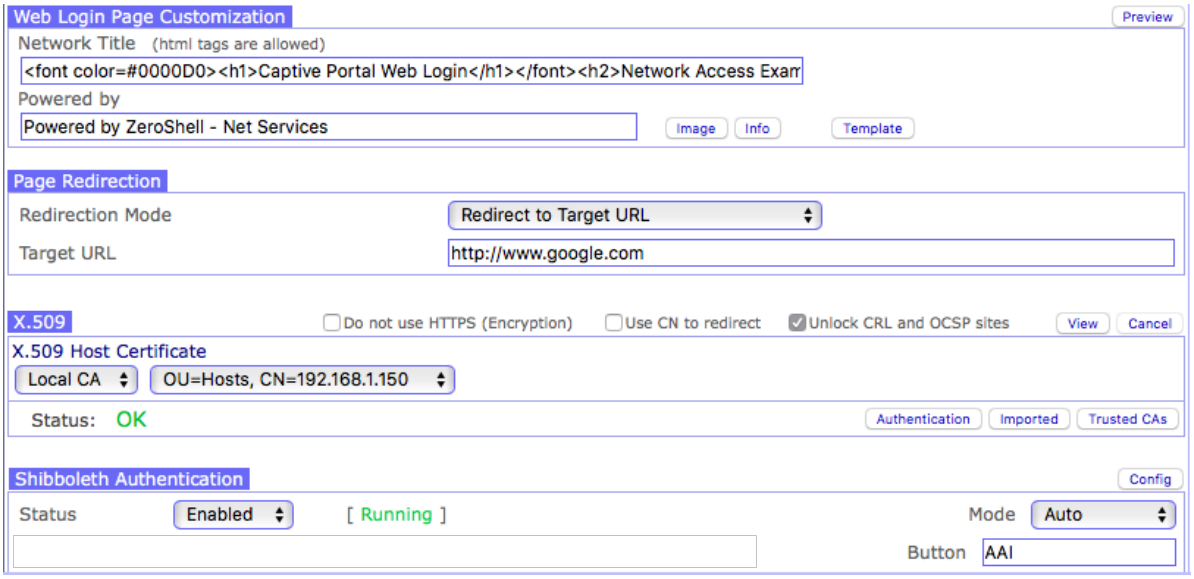

Figura 3.6: Interfaz web administrativa de ZeroShell

Al igual que el SP del webmail, la configuración de este proveedor de servicio debe hacerse en el archivo shibboleth.xml, tanto el archivo para éste SP como para el SP del webmail tienen la misma estructura y deben ser configurados de la misma forma con la salvedad que para éste SP la configuración se realiza mediante la interfaz administrativa web que permite manejar todas las funciones que ZeroShell ofrece.

El detalle de la instalación de *ZeroShell* y las configuración de éste SP se encuentran en la documentación en línea en la siguiente dirección `http://190.168.24.43/wiki/index.php/ZeroShell`

En la figura 3.7 se muestra la topología propuesta para la zona wifi integrada con la autenticación con *Shibboleth*. En esta topología se tiene una red local en la que están conectados los *wireless routers* funcionando en modo AP y una interfaz de red del SP con ZeroShell. El SP tiene otra interfaz por la cual se comunica con el IdP y de este modo puede realizar la autenticación de los clientes.

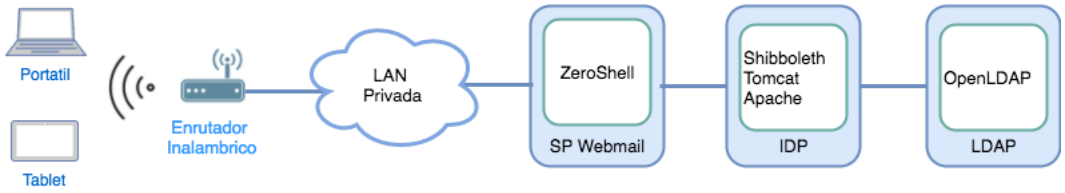

Figura 3.7: Topología para el SP de la zona wifi.

Para realizar la prueba de concepto de integración del SP de la zona WiFi con el proveedor de identidad se utilizaron equipos con las mismas características del resto de los equipos utilizados actualmente para dar soporte de conexión a las zonas WiFi en la Facultad de Ingeniería. Estos equipos tienen su *firmware* modificado ya que originalmente no es posible deshabilitar la opción de NAT ni cambiar el portal cautivo que estos dispositivos traen por omisión, el firmware con el que están funcionando es DD WRT V1.4.1.

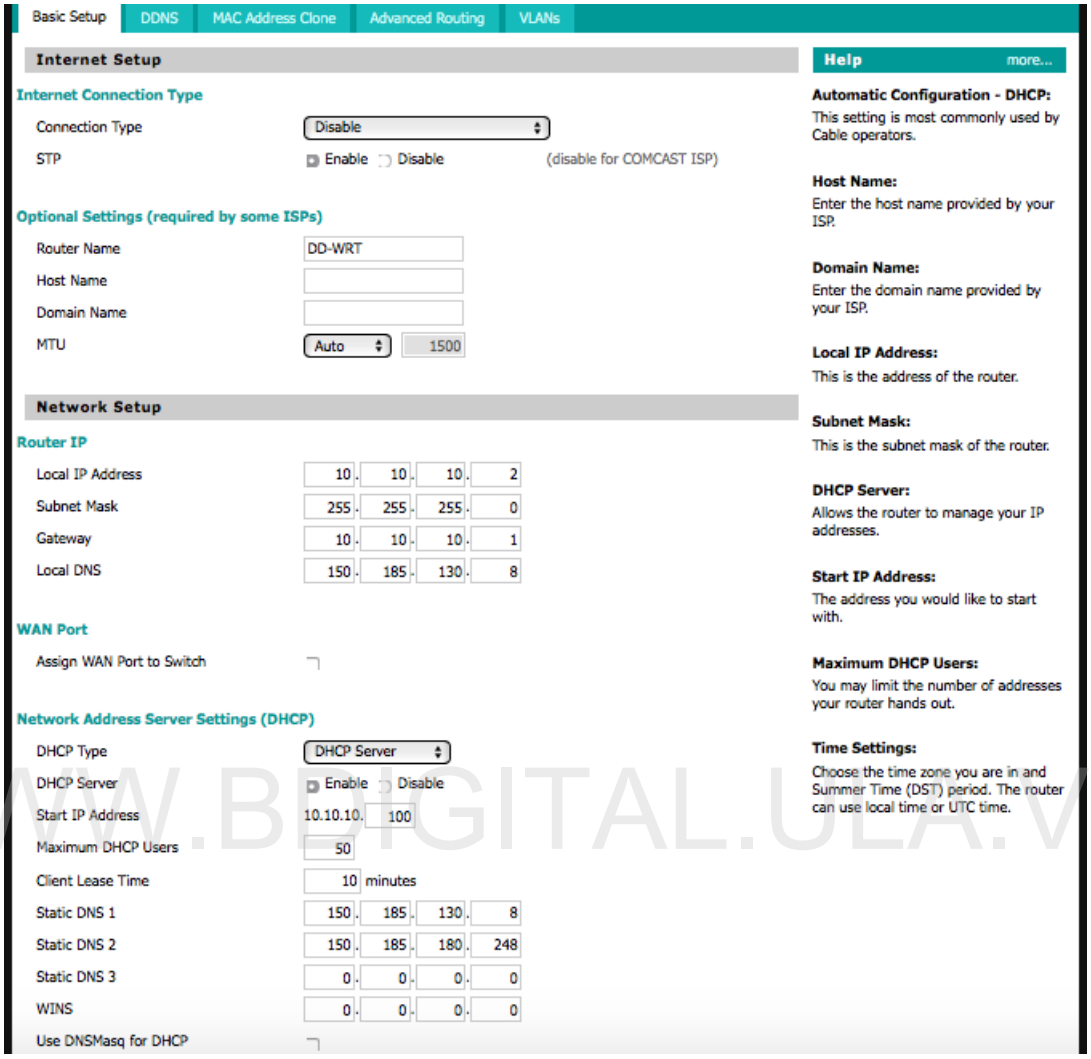

Figura 3.8: Configuración de red del equipo de acceso inalámbrico.

En la figura 3.8 se muestra detalles de la configuración de red en uno de los equipos con los que se realizaron las pruebas de integración del portal cautivo con el IdP. La dirección IP **10.10.10.1** corresponde a la interfaz de red del SP que esta en la misma red de los *wireless routers*. La dirección IP 10.10.10.2 corresponde al dispositivo. Adicionalmente se configuró como única ruta hacia todas las redes la misma ip del SP, como se observa en la figura 3.9. La figura 3.10 corresponde a una vista de las rutas configuradas en el dispositivo

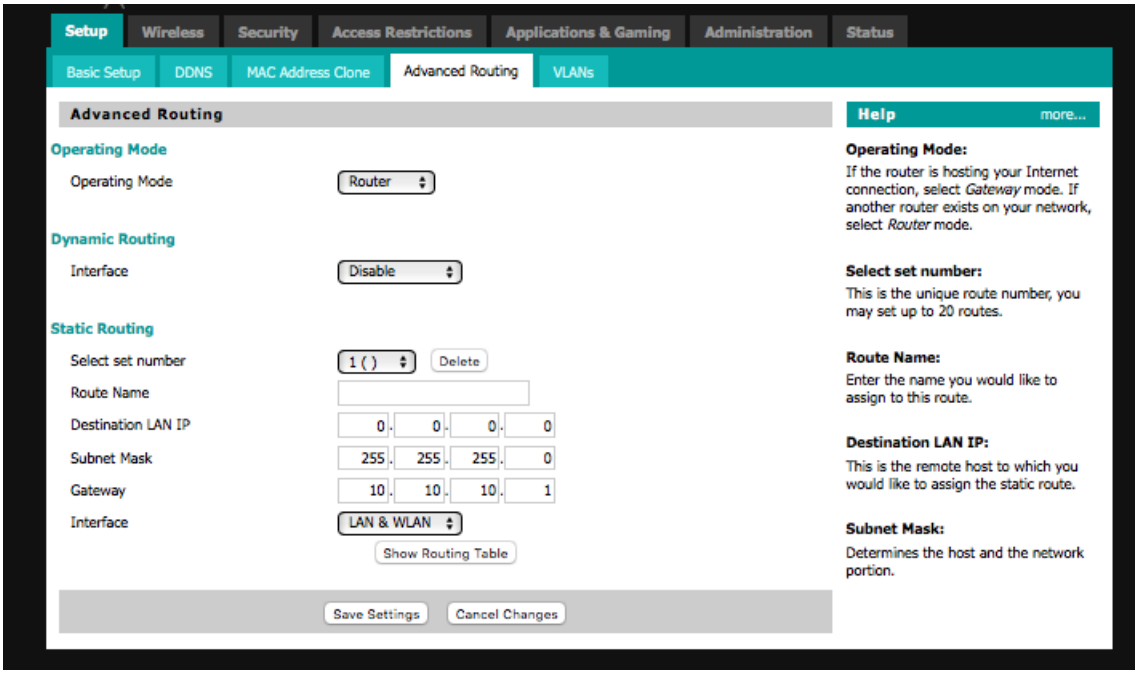

Figura 3.9: Configuración de las rutas en el *wireless router*

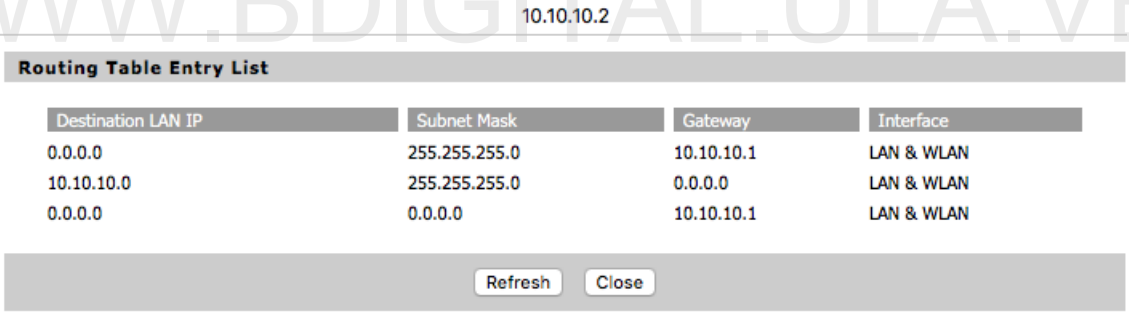

Figura 3.10: Vista con las rutas configuradas en el dispositivo

En este punto todos los elementos que forman parte de la Infraestructura de Identidad Digital se encuentran operativos y listos para pasar a la fase de pruebas. En la figura 3.11 se describe la arquitectura implementada en la infraestructura de identidad digital de RedULA.

Tanto el SP correspondientes al servicio de *Webmail* como el SP del servicio de portal cautivo de la zona WiFi están conectados al IdP a través de la red local de servicios, por esta misma red también se conecta el IdP con el servicio de directorio. Los *wireless routers* que proveen conectividad a la red de la zona WiFi están conectados al SP a

través de una red privada. El SP del portal cautivo además de autenticar a los usuarios hace el servicio de NAT para que los clientes conectados puedan salir a internet.

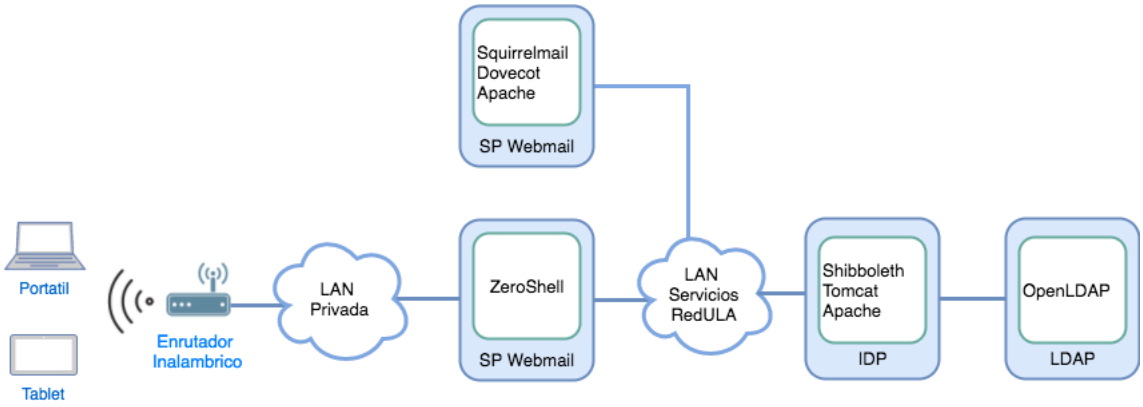

Figura 3.11: Arquitectura completa para el piloto de la infraestructura de identidad digital

Capítulo 4

Pruebas sobre la infraestructura

El desempeño de la plataforma bajo diferentes escenarios es relevante a la hora de realizar consideraciones de despliegue. En el caso de la infraestructura de identidad digital se utilizó la herramienta ***JMeter*** para realizar pruebas de rendimiento sobre el proveedor de identidad que deberá servir a un número considerable de usuarios en transacciones de autenticación. Se realizó un plan de pruebas específicas tomando como guía un plan de pruebas desarrollado en la Universidad de Chicago para su IdP con *Shibboleth* (Shibboleth Wiki, 2016) con las siguientes características:

- Las pruebas se realizaron en un equipo conectado a la red del SP y del IdP, esto con el fin de evitar que la latencia de la red modificara los resultados del tiempo de respuesta del IdP.

- Se generó una lista de usuarios en el directorio LDAP para realizar las pruebas, ésta lista se pasa en formato *csv* al archivo de configuración testidpredula.jmx como se observa en el cuadro 4.1. La cantidad de usuarios se aumenta hasta llegar a 800 usuarios atendidos.

- La primera parte de la prueba emite una solicitud de tipo HTTP GET a un punto de prueba que responde con un formulario de acceso que solicita usuario y contraseña para iniciar sesión. El siguiente paso consiste en enviar una solicitud de tipo POST a dicho formulario obteniendo de este modo una respuesta del SAML2 que es enviada al IdP para completar el proceso de autenticación. Esta

prueba obtiene como respuesta un SAML assertion.

- De forma similar, la segunda parte emite una solicitud de tipo HTTP GET a un punto disponible en el IdP llamado unsolicited SSO. Dicho endpoint responde con un formulario de inicio de sesión y la prueba envía una solicitud POST con credenciales de prueba para completar de esta forma la autenticación y obtener como respuesta un SAML assertion.

La prueba completa se repitió por intervalos de 10 minutos hasta que tanto el IdP como el SP no pudieron aceptar más solicitudes.

```
<CSVDataSet guiclass="TestBeanGUI" testclass="CSVDataSet" testname="./usuarios-
    ulatest.csv" enabled="true">
  <stringProp name="filename">/Users/danielagutierrez2/Documents/clases/tesis/
      users.csv</stringProp>
  <stringProp name="fileEncoding"></stringProp>
  <stringProp name="variableNames">User,Password</stringProp>
  <stringProp name="delimiter">,</stringProp>
  <boolProp name="quotedData">false</boolProp>
  <boolProp name="recycle">true</boolProp>
  <boolProp name="stopThread">false</boolProp>
  <stringProp name="shareMode">shareMode.all</stringProp>
</CSVDataSet>
```

Cuadro 4.1: Configuración de la lista de usuarios en el archivo testidpredula.jmx.

En el cuadro 4.1 se muestra como se realizaron algunas de las configuraciones para las pruebas con la herramienta ***JMeter***, como la configuración del IdP y del SP, así como el puerto para la conexión al IdP.

```
<elementProp name="ShibSP" elementType="Argument">
<stringProp name="Argument.name">ShibSP</stringProp>
<stringProp name="Argument.value">https://sp-ula</stringProp>
<stringProp name="Argument.metadata">=</stringProp>
</elementProp>
<elementProp name="ProviderId" elementType="Argument">
<stringProp name="Argument.name">ProviderId</stringProp>
<stringProp name="Argument.value">https://idp-ula/shibboleth</stringProp>
<stringProp name="Argument.metadata">=</stringProp>
</elementProp>
<elementProp name="IdPPort" elementType="Argument">
<stringProp name="Argument.name">IdPPort</stringProp>
<stringProp name="Argument.value">443</stringProp>
<stringProp name="Argument.metadata">=</stringProp>
</elementProp>
```

Cuadro 4.2: Configuración del IdP SP y puertos en el archivo testidpredula.jmx.

El archivo de configuración de la prueba se encuentra en la documentación en línea en la siguiente dirección http://190.168.24.43/wiki/index.php/Proveedor_de_

`Identidad` en la sección *Disponibles para descargar*.

La figura 4.1 muestra gráficamente los resultados obtenidos:

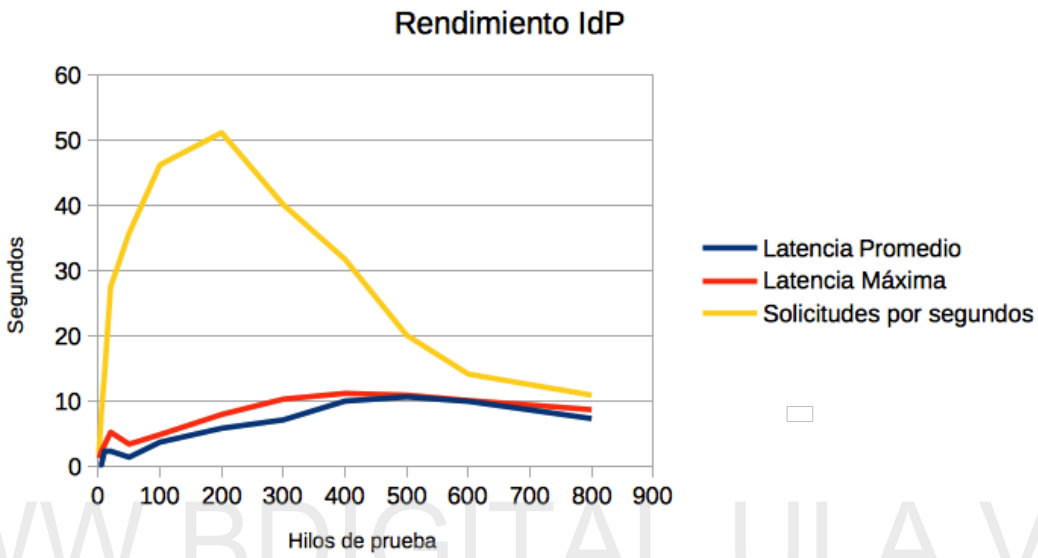

Figura 4.1: Test de rendimiento

En los resultados se puede observar que el uso de recursos es mínimo durante los primeros 10 usuarios, y la latencia promedio se mantuvo en 8 segundos hasta 200 hilos de prueba (300 usuarios), menos de 10 segundos hasta 500 hilos de prueba, y se mantuvo cercano a los 10 hasta 800 hilos de prueba. En cuanto a la latencia máxima, esta permaneció relativamente baja hasta aproximadamente 55 usuarios, donde comenzó a subir considerablemente. Los tiempos de respuesta máximos no ideales permanecieron constantes hasta que el IdP no pudo manejar el más tráfico satisfactoriamente. En ningún caso fue necesario reiniciar los servicios dado que al finalizar las pruebas los equipos recuperaron su capacidad de respuesta habitual.

Capítulo 5

Conclusiones y Recomendaciones

La investigación sobre los diversos protocolos existentes para el manejo de identidad, así como de las tecnologías utilizadas en el ámbito de las redes académicas para abordar este tema, permitió llegar al diseño de una infraestructura que cumple con los requerimientos inicialmente planteados. En términos generales las pruebas realizadas sobre la infraestructura de identidad digital implementada permitieron validar dicho diseño propuesto logrando de esta forma cumplir con el objetivo principal planteado al inicio de este proyecto de grado.

Dada la naturaleza de los servicios que se desean integrar, la infraestructura aún se mantiene en fase de pruebas, para el despliegue y pase a producción es necesario realizar algunos cambios en los servicios actuales, como lo son: agregar en el servicio de directorio el nuevo esquema LDAP para *eduPerson*, siendo este esquema el que permitirá que los usuarios de la Universidad en un futuro puedan acceder a servicios de otras redes y federaciones. Cambiar en el servicio de *webmail* el *Courier* actual (Courier-IMAP) por Dovecot para poder integrarlo satisfactoriamente a la infraestructura.

En cuanto a la integración de la infraestructura con el servicio de conexión en zonas WiFi, las pruebas de concepto realizadas fueron satisfactorias probando que se puede realizar autenticación de usuarios con el IdP, esto ofrece la posibilidad de tener SSO entre los servicios de webmail y la conexión a la red inalámbrica. Sin embargo se recomienda estudiar otras opciones para la autenticación de éste servicio como el proyecto ***eduroam*** (EduRoam, 2016) que provee un nivel de seguridad mayor, y es

una plataforma que esta desplegada en instituciones de educación superior alrededor de 70 paises.

Otro de los objetivos planteados como lo es la documentación en línea fue cumplido sin tener mayor inconveniente, esta documentación está alojada en servidores administrados por RedULA a la que se puede acceder en http://190.168.24.43/wiki/.

El diseño que se implementó no contemplaba caracteristicas de alta disponibilidad como lo son balanceo de carga y *fail over*, una de las recomendaciones para trabajos futuros es utilizar la solución ofrecida por ***Shibboleth*** para abordar el tema de alta disponibilidad. Estos últimos puntos estan fuera del alcance del proveedor de identidad como tal, los mecanismos recomendados para tal fin incluyen los comunes para *clustering* como lo son DNS Round Robin y Clustering basado en hardware. Adicionalmente *Shibboleth* también recomienda seguir la guia de *clustering* para el proveedor de identidad que ellos han desarrollado. Este mecanismo incluye otro paquete de software llamado ***Terracotta***, que permite replicar el estado de la memoria del proveedor de identidad entre los nodos.

El ámbito académico de federaciones de identidad en Latinoamérica y el mundo se mantiene en constante actualización y mejoras, por lo que también se sugiere a la organización mantenerse activo y realizar las gestiones necesarias para formar parte de una federación educativa, así como difundir los aspectos técnicos que permitieron llevar a cabo el diseño y la implementación de la infraestructura.

Capítulo 6

Anexos

6.1. eduperson.schema

```
#
# eduperson.schema (OpenLDAP)
# other schema files: see https://spaces.internet2.edu/display/macedir/LDIFs
#
# eduPerson Objectclass (200806)
#
# See http://middleware.internet2.edu/eduperson/ for background and usage
#
# eduPerson is an effort of Internet2 and EDUCAUSE
#
#
#
# When modifying objectclass eduperson --
#                  we first must delete the objectclass
# and then re-add -- make sure all replicas are functioning.  Try to do this
# during an inactive period of services (if possible).
#
# Modifying schema may only affect the instance being modified --
#                     it may NOT replicate!
#
# check your server documentation to verify this.
#
# 1.3.6.1.4.1.5923 is the toplevel OID for this work
#             .1 = MACE related work
#             .1.1 = eduPerson
#             .1.1.1 = attributes
#             .1.1.2 = objectclass
#             .1.1.3 = syntax (probably never used)
#             .1.2 = eduOrg
#             .1.2.1 = attributes
#             .1.2.2 = objectclass
#             .1.2.3 = syntax (probably never used)
#
#
# "eduPerson" attributes
#
##############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.1
    NAME 'eduPersonAffiliation'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

#
##############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.2
    NAME 'eduPersonNickname'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

#
##############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.3
    NAME 'eduPersonOrgDN'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY distinguishedNameMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.12'
    SINGLE-VALUE )

#
##############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.4
    NAME 'eduPersonOrgUnitDN'
```

```
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY distinguishedNameMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.12' )

#
#############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.5
    NAME 'eduPersonPrimaryAffiliation'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    SINGLE-VALUE )

#
#############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.6
    NAME 'eduPersonPrincipalName'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    SINGLE-VALUE )

#
#############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.7
    NAME 'eduPersonEntitlement'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseExactMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

#
#############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.8
    NAME 'eduPersonPrimaryOrgUnitDN'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY distinguishedNameMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.12'
    SINGLE-VALUE )

#
#############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.9
    NAME 'eduPersonScopedAffiliation'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

#
#############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.10
    NAME 'eduPersonTargetedID'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

#
#############################################################################
#
attributeType ( 1.3.6.1.4.1.5923.1.1.1.11
    NAME 'eduPersonAssurance'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
```

```
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

#
#############################################################################
#
objectClass ( 1.3.6.1.4.1.5923.1.1.2
    NAME 'eduPerson'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    AUXILIARY
    MAY ( eduPersonAffiliation $ eduPersonNickname $ eduPersonOrgDN $
          eduPersonOrgUnitDN $ eduPersonPrimaryAffiliation $
          eduPersonPrincipalName $ eduPersonEntitlement $
          eduPersonPrimaryOrgUnitDN $ eduPersonScopedAffiliation $
          eduPersonTargetedID $ eduPersonAssurance ) )


#############################################################################
```

6.2. handler.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ph:ProfileHandlerGroup xmlns:ph="urn:mace:shibboleth:2.0:idp:profile-handler" xmlns:xsi="http://
    www.w3.org/2001/XMLSchema-instance"
                        xsi:schemaLocation="urn:mace:shibboleth:2.0:idp:profile-handler classpath:/
                            schema/shibboleth-2.0-idp-profile-handler.xsd">

    <!-- Error Handler -->
    <ph:ErrorHandler xsi:type="ph:JSPErrorHandler" jspPagePath="/error.jsp"/>

    <!-- Profile Handlers -->
    <!--
        All profile handlers defined below are accessed via the Servlet path "/profile" so if your
            profile
        handler's request path is "/Status" then the full path is "<servletContextName>/profile/Status"
     -->
    <ph:ProfileHandler xsi:type="ph:Status">
        <ph:RequestPath>/Status</ph:RequestPath>
    </ph:ProfileHandler>

    <ph:ProfileHandler xsi:type="ph:SAMLMetadata" metadataFile="/opt/shibboleth-idp/metadata/idp-
        metadata.xml">
        <ph:RequestPath>/Metadata/SAML</ph:RequestPath>
    </ph:ProfileHandler>

    <ph:ProfileHandler xsi:type="ph:ShibbolethSSO" inboundBinding="urn:mace:shibboleth:
        1.0:profiles:AuthnRequest"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:1.0:profiles:browser-post
                                                   urn:oasis:names:tc:SAML:1.0:profiles:artifact-01">
        <ph:RequestPath>/Shibboleth/SSO</ph:RequestPath>
    </ph:ProfileHandler>

    <ph:ProfileHandler xsi:type="ph:SAML1AttributeQuery" inboundBinding="urn:oasis:names:tc:SAML:
        1.0:bindings:SOAP-binding"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding">
        <ph:RequestPath>/SAML1/SOAP/AttributeQuery</ph:RequestPath>
    </ph:ProfileHandler>

    <ph:ProfileHandler xsi:type="ph:SAML1ArtifactResolution" inboundBinding="urn:oasis:names:tc:SAML:
        1.0:bindings:SOAP-binding"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding">
        <ph:RequestPath>/SAML1/SOAP/ArtifactResolution</ph:RequestPath>
    </ph:ProfileHandler>

    <ph:ProfileHandler xsi:type="ph:SAML2SSO" inboundBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
        POST"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
                           SimpleSign
                                                   urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
                                                   urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact">
        <ph:RequestPath>/SAML2/POST/SSO</ph:RequestPath>
    </ph:ProfileHandler>

    <ph:ProfileHandler xsi:type="ph:SAML2SSO" inboundBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
        POST-SimpleSign"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
                           SimpleSign
                                                   urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
                                                   urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact">
        <ph:RequestPath>/SAML2/POST-SimpleSign/SSO</ph:RequestPath>
    </ph:ProfileHandler>

    <ph:ProfileHandler xsi:type="ph:SAML2SSO" inboundBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
        Redirect"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
                           SimpleSign
                                                   urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
                                                   urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact">
        <ph:RequestPath>/SAML2/Redirect/SSO</ph:RequestPath>
    </ph:ProfileHandler>
```

```
58
59      <ph:ProfileHandler xsi:type="ph:SAML2SSO" inboundBinding="urn:mace:shibboleth:
            2.0:profiles:AuthnRequest"
60                         outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
                               SimpleSign
61                                                     urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
62                                                     urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact">
63          <ph:RequestPath>/SAML2/Unsolicited/SSO</ph:RequestPath>
64      </ph:ProfileHandler>
65
66      <ph:ProfileHandler xsi:type="ph:SAML2ECP" inboundBinding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
67                         outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:SOAP">
68          <ph:RequestPath>/SAML2/SOAP/ECP</ph:RequestPath>
69      </ph:ProfileHandler>
70
71      <ph:ProfileHandler xsi:type="ph:SAML2AttributeQuery" inboundBinding="urn:oasis:names:tc:SAML:
            2.0:bindings:SOAP"
72                         outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:SOAP">
73          <ph:RequestPath>/SAML2/SOAP/AttributeQuery</ph:RequestPath>
74      </ph:ProfileHandler>
75
76      <ph:ProfileHandler xsi:type="ph:SAML2ArtifactResolution" inboundBinding="urn:oasis:names:tc:SAML:
            2.0:bindings:SOAP"
77                         outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:SOAP">
78          <ph:RequestPath>/SAML2/SOAP/ArtifactResolution</ph:RequestPath>
79      </ph:ProfileHandler>
80
81      <!-- Login Handlers
82      <ph:LoginHandler xsi:type="ph:RemoteUser">
83          <ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</
                ph:AuthenticationMethod>
84      </ph:LoginHandler>
85      -->
86
87      <!-- Login handler that delegates the act of authentication to an external system. -->
88      <!-- This login handler and the RemoteUser login handler will be merged in the next major release. --
            >
89      <!--
90      <ph:LoginHandler xsi:type="ph:ExternalAuthn">
91          <ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</
                ph:AuthenticationMethod>
92          <ph:QueryParam name="foo" value="bar" />
93      </ph:LoginHandler>
94      -->
95
96      <!--  Username/password login handler -->
97
98      <ph:LoginHandler xsi:type="ph:UsernamePassword"
99                       jaasConfigurationLocation="file:///opt/shibboleth-idp/conf/login.config">
100         <ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</
                ph:AuthenticationMethod>
101     </ph:LoginHandler>
102
103
104     <!--
105         Removal of this login handler will disable SSO support, that is it will require the user to
                authenticate
106         on every request.
107     -->
108     <ph:LoginHandler xsi:type="ph:PreviousSession">
109         <ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession</
                ph:AuthenticationMethod>
110     </ph:LoginHandler>
111
112 </ph:ProfileHandlerGroup>
113
```

6.3. attribute-resolver.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    This file is an EXAMPLE configuration file.  While the configuration presented in this
    example file is functional, it isn't very interesting.  However, there are lots of example
    attributes, encoders, and a couple example data connectors.

    Not all attribute definitions, data connectors, or principal connectors are demonstrated.
    Deployers should refer to the Shibboleth 2 documentation for a complete list of components
    and their options.
-->
<resolver:AttributeResolver xmlns:resolver="urn:mace:shibboleth:2.0:resolver" xmlns:xsi="http://
    www.w3.org/2001/XMLSchema-instance"
                            xmlns:pc="urn:mace:shibboleth:2.0:resolver:pc" xmlns:ad="urn:mace:shibboleth:
                                2.0:resolver:ad"
                            xmlns:dc="urn:mace:shibboleth:2.0:resolver:dc"
                                xmlns:enc="urn:mace:shibboleth:2.0:attribute:encoder"
                            xmlns:sec="urn:mace:shibboleth:2.0:security"
                            xsi:schemaLocation="urn:mace:shibboleth:2.0:resolver classpath:/schema/
                                shibboleth-2.0-attribute-resolver.xsd
                                                urn:mace:shibboleth:2.0:resolver:pc classpath:/schema/
                                                    shibboleth-2.0-attribute-resolver-pc.xsd
                                                urn:mace:shibboleth:2.0:resolver:ad classpath:/schema/
                                                    shibboleth-2.0-attribute-resolver-ad.xsd
                                                urn:mace:shibboleth:2.0:resolver:dc classpath:/schema/
                                                    shibboleth-2.0-attribute-resolver-dc.xsd
                                                urn:mace:shibboleth:2.0:attribute:encoder classpath:/
                                                    schema/shibboleth-2.0-attribute-encoder.xsd
                                                urn:mace:shibboleth:2.0:security classpath:/schema/
                                                    shibboleth-2.0-security.xsd">

    <!-- ========================================== -->
    <!--      Attribute Definitions                 -->
    <!-- ========================================== -->

    <!-- Schema: Core schema attributes-->
    <resolver:AttributeDefinition xsi:type="ad:Simple" id="uid" sourceAttributeID="uid">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:uid" />
        <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.1"
            friendlyName="uid" />
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition xsi:type="ad:Simple" id="email" sourceAttributeID="mail">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:mail" />
        <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.3"
            friendlyName="mail" />
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition xsi:type="ad:Simple" id="homePhone" sourceAttributeID="homePhone">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:homePhone"
            />
        <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.20"
            friendlyName="homePhone" />
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition xsi:type="ad:Simple" id="homePostalAddress"
        sourceAttributeID="homePostalAddress">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
            def:homePostalAddress" />
        <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.39"
            friendlyName="homePostalAddress" />
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition xsi:type="ad:Simple" id="mobileNumber" sourceAttributeID="mobile">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:mobile" />
```

```
54          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.41"
                friendlyName="mobile" />
55      </resolver:AttributeDefinition>
56
57      <resolver:AttributeDefinition xsi:type="ad:Simple" id="pagerNumber" sourceAttributeID="pager">
58          <resolver:Dependency ref="myLDAP" />
59          <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:pager" />
60          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.42"
                friendlyName="pager" />
61      </resolver:AttributeDefinition>
62
63      <resolver:AttributeDefinition xsi:type="ad:Simple" id="commonName" sourceAttributeID="cn">
64          <resolver:Dependency ref="myLDAP" />
65          <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:cn" />
66          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.3" friendlyName="cn" />
67      </resolver:AttributeDefinition>
68
69      <resolver:AttributeDefinition xsi:type="ad:Simple" id="surname" sourceAttributeID="sn">
70          <resolver:Dependency ref="myLDAP" />
71          <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:sn" />
72          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.4" friendlyName="sn" />
73      </resolver:AttributeDefinition>
74
75      <resolver:AttributeDefinition xsi:type="ad:Simple" id="locality" sourceAttributeID="l">
76          <resolver:Dependency ref="myLDAP" />
77          <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:l" />
78          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.7" friendlyName="l" />
79      </resolver:AttributeDefinition>
80
81      <resolver:AttributeDefinition xsi:type="ad:Simple" id="stateProvince" sourceAttributeID="st">
82          <resolver:Dependency ref="myLDAP" />
83          <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:st" />
84          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.8" friendlyName="st" />
85      </resolver:AttributeDefinition>
86
87      <resolver:AttributeDefinition xsi:type="ad:Simple" id="street" sourceAttributeID="street">
88          <resolver:Dependency ref="myLDAP" />
89          <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:street" />
90          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.9"
                friendlyName="street" />
91      </resolver:AttributeDefinition>
92
93      <resolver:AttributeDefinition xsi:type="ad:Simple" id="organizationName" sourceAttributeID="o">
94          <resolver:Dependency ref="myLDAP" />
95          <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:o" />
96          <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.10" friendlyName="o" />
97      </resolver:AttributeDefinition>
98
99      <resolver:AttributeDefinition xsi:type="ad:Simple" id="organizationalUnit" sourceAttributeID="ou">
100         <resolver:Dependency ref="myLDAP" />
101         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:ou" />
102         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.11" friendlyName="ou" /
                >
103     </resolver:AttributeDefinition>
104
105     <resolver:AttributeDefinition xsi:type="ad:Simple" id="title" sourceAttributeID="title">
106         <resolver:Dependency ref="myLDAP" />
107         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:title" />
108         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.12"
                friendlyName="title" />
109     </resolver:AttributeDefinition>
110
111     <resolver:AttributeDefinition xsi:type="ad:Simple" id="postalAddress"
            sourceAttributeID="postalAddress">
112         <resolver:Dependency ref="myLDAP" />
113         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:postalAddress" />
114         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.16"
                friendlyName="postalAddress" />
115     </resolver:AttributeDefinition>
```

```
116
117     <resolver:AttributeDefinition xsi:type="ad:Simple" id="postalCode" sourceAttributeID="postalCode">
118         <resolver:Dependency ref="myLDAP" />
119         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:postalCode" />
120         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.17"
                friendlyName="postalCode" />
121     </resolver:AttributeDefinition>
122
123     <resolver:AttributeDefinition xsi:type="ad:Simple" id="postOfficeBox"
            sourceAttributeID="postOfficeBox">
124         <resolver:Dependency ref="myLDAP" />
125         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:postOfficeBox" />
126         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.18"
                friendlyName="postOfficeBox" />
127     </resolver:AttributeDefinition>
128
129     <resolver:AttributeDefinition xsi:type="ad:Simple" id="telephoneNumber"
            sourceAttributeID="telephoneNumber">
130         <resolver:Dependency ref="myLDAP" />
131         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:telephoneNumber" />
132         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.20"
                friendlyName="telephoneNumber" />
133     </resolver:AttributeDefinition>
134
135     <resolver:AttributeDefinition xsi:type="ad:Simple" id="givenName" sourceAttributeID="givenName">
136         <resolver:Dependency ref="myLDAP" />
137         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:givenName"
                />
138         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.42"
                friendlyName="givenName" />
139     </resolver:AttributeDefinition>
140
141     <resolver:AttributeDefinition xsi:type="ad:Simple" id="initials" sourceAttributeID="initials">
142         <resolver:Dependency ref="myLDAP" />
143         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:initials" />
144         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.43"
                friendlyName="initials" />
145     </resolver:AttributeDefinition>
146
147     <!-- Schema: inetOrgPerson attributes-->
148     <resolver:AttributeDefinition xsi:type="ad:Simple" id="departmentNumber"
            sourceAttributeID="departmentNumber">
149         <resolver:Dependency ref="myLDAP" />
150         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:departmentNumber" />
151         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.16.840.1.113730.3.1.2"
                friendlyName="departmentNumber" />
152     </resolver:AttributeDefinition>
153
154     <resolver:AttributeDefinition xsi:type="ad:Simple" id="displayName" sourceAttributeID="displayName">
155         <resolver:Dependency ref="myLDAP" />
156         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:displayName" />
157         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.16.840.1.113730.3.1.241"
                friendlyName="displayName" />
158     </resolver:AttributeDefinition>
159
160     <resolver:AttributeDefinition xsi:type="ad:Simple" id="employeeNumber"
            sourceAttributeID="employeeNumber">
161         <resolver:Dependency ref="myLDAP" />
162         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:employeeNumber" />
163         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.16.840.1.113730.3.1.3"
                friendlyName="employeeNumber" />
164     </resolver:AttributeDefinition>
165
```

```
166     <resolver:AttributeDefinition xsi:type="ad:Simple" id="employeeType"
            sourceAttributeID="employeeType">
167         <resolver:Dependency ref="myLDAP" />
168         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:employeeType" />
169         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.16.840.1.113730.3.1.4"
                friendlyName="employeeType" />
170     </resolver:AttributeDefinition>
171
172     <resolver:AttributeDefinition xsi:type="ad:Simple" id="jpegPhoto" sourceAttributeID="jpegPhoto">
173         <resolver:Dependency ref="myLDAP" />
174         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:jpegPhoto"
                />
175         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.60"
                friendlyName="jpegPhoto" />
176     </resolver:AttributeDefinition>
177
178     <resolver:AttributeDefinition xsi:type="ad:Simple" id="preferredLanguage"
            sourceAttributeID="preferredLanguage">
179         <resolver:Dependency ref="myLDAP" />
180         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:preferredLanguage" />
181         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.16.840.1.113730.3.1.39"
                friendlyName="preferredLanguage" />
182     </resolver:AttributeDefinition>
183
184     <!-- Schema: eduPerson attributes -->
185     <!--
186     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonAffiliation"
            sourceAttributeID="eduPersonAffiliation">
187         <resolver:Dependency ref="myLDAP" />
188         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonAffiliation" />
189         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
                friendlyName="eduPersonAffiliation" />
190     </resolver:AttributeDefinition>
191
192     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonEntitlement"
            sourceAttributeID="eduPersonEntitlement">
193         <resolver:Dependency ref="myLDAP" />
194         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonEntitlement" />
195         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
                friendlyName="eduPersonEntitlement" />
196     </resolver:AttributeDefinition>
197
198     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonNickname"
            sourceAttributeID="eduPersonNickname">
199         <resolver:Dependency ref="myLDAP" />
200         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonNickname" />
201         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.2"
                friendlyName="eduPersonNickname" />
202     </resolver:AttributeDefinition>
203
204     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonOrgDN"
            sourceAttributeID="eduPersonOrgDN">
205         <resolver:Dependency ref="myLDAP" />
206         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonOrgDN" />
207         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.3"
                friendlyName="eduPersonOrgDN" />
208     </resolver:AttributeDefinition>
209
210     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonOrgUnitDN"
            sourceAttributeID="eduPersonOrgUnitDN">
211         <resolver:Dependency ref="myLDAP" />
212         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonOrgUnitDN" />
```

```
213         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.4"
                friendlyName="eduPersonOrgUnitDN" />
214     </resolver:AttributeDefinition>
215
216     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonPrimaryAffiliation"
            sourceAttributeID="eduPersonPrimaryAffiliation">
217         <resolver:Dependency ref="myLDAP" />
218         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonPrimaryAffiliation" />
219         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.5"
                friendlyName="eduPersonPrimaryAffiliation" />
220     </resolver:AttributeDefinition>
221
222     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonPrimaryOrgUnitDN"
            sourceAttributeID="eduPersonPrimaryOrgUnitDN">
223         <resolver:Dependency ref="myLDAP" />
224         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonPrimaryOrgUnitDN" />
225         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.8"
                friendlyName="eduPersonPrimaryOrgUnitDN" />
226     </resolver:AttributeDefinition>
227
228     <resolver:AttributeDefinition xsi:type="ad:Scoped" id="eduPersonPrincipalName" scope="63.55"
            sourceAttributeID="uid">
229         <resolver:Dependency ref="myLDAP" />
230         <resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString" name="urn:mace:dir:attribute-
                def:eduPersonPrincipalName" />
231         <resolver:AttributeEncoder xsi:type="enc:SAML2ScopedString" name="urn:oid:
                1.3.6.1.4.1.5923.1.1.1.6" friendlyName="eduPersonPrincipalName" />
232     </resolver:AttributeDefinition>
233
234     <resolver:AttributeDefinition xsi:type="ad:Scoped" id="eduPersonScopedAffiliation" scope="63.55"
            sourceAttributeID="eduPersonAffiliation">
235         <resolver:Dependency ref="myLDAP" />
236         <resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString" name="urn:mace:dir:attribute-
                def:eduPersonScopedAffiliation" />
237         <resolver:AttributeEncoder xsi:type="enc:SAML2ScopedString" name="urn:oid:
                1.3.6.1.4.1.5923.1.1.1.9" friendlyName="eduPersonScopedAffiliation" />
238     </resolver:AttributeDefinition>
239
240     <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonAssurance"
            sourceAttributeID="eduPersonAssurance">
241         <resolver:Dependency ref="myLDAP" />
242         <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
                def:eduPersonAssurance" />
243         <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.11"
                friendlyName="eduPersonAssurance" />
244     </resolver:AttributeDefinition>
245     -->
246
247     <!--
248     <resolver:AttributeDefinition xsi:type="ad:SAML2NameID" id="eduPersonTargetedID"
249                                   nameIdFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
                                          sourceAttributeID="computedID">
250         <resolver:Dependency ref="computedID" />
251         <resolver:AttributeEncoder xsi:type="enc:SAML1XMLObject" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
                />
252         <resolver:AttributeEncoder xsi:type="enc:SAML2XMLObject" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
                friendlyName="eduPersonTargetedID" />
253     </resolver:AttributeDefinition>
254     -->
255
256     <!-- Do NOT use the version of eduPersonTargetedID defined below unless you understand
257          why it was deprecated and know that this reason does not apply to you. -->
258     <!--
259     <resolver:AttributeDefinition xsi:type="ad:Scoped" id="eduPersonTargetedID.old" scope="63.55"
            sourceAttributeID="computedID">
260         <resolver:Dependency ref="computedID" />
261         <resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString" name="urn:mace:dir:attribute-
                def:eduPersonTargetedID" />
```

```
262     </resolver:AttributeDefinition>
263     -->
264
265     <!-- Name Identifier related attributes -->
266     <resolver:AttributeDefinition id="transientId" xsi:type="ad:TransientId">
267         <resolver:AttributeEncoder xsi:type="enc:SAML1StringNameIdentifier"
                nameFormat="urn:mace:shibboleth:1.0:nameIdentifier"/>
268         <resolver:AttributeEncoder xsi:type="enc:SAML2StringNameID" nameFormat="urn:oasis:names:tc:SAML:
                2.0:nameid-format:transient"/>
269     </resolver:AttributeDefinition>
270
271     <!-- ========================================== -->
272     <!--      Data Connectors                       -->
273     <!-- ========================================== -->
274
275     <!-- Example Static Connector -->
276     <!--
277     <resolver:DataConnector id="staticAttributes" xsi:type="dc:Static">
278         <dc:Attribute id="eduPersonAffiliation">
279             <dc:Value>member</dc:Value>
280         </dc:Attribute>
281         <dc:Attribute id="eduPersonEntitlement">
282             <dc:Value>urn:example.org:entitlement:entitlement1</dc:Value>
283             <dc:Value>urn:mace:dir:entitlement:common-lib-terms</dc:Value>
284         </dc:Attribute>
285     </resolver:DataConnector>
286     -->
287
288     <!-- Example Relational Database Connector -->
289     <!--
290     <resolver:DataConnector id="mySIS" xsi:type="dc:RelationalDatabase">
291         <dc:ApplicationManagedConnection jdbcDriver="oracle.jdbc.driver.OracleDriver"
292                                          jdbcURL="jdbc:oracle:thin:@db.example.org:1521:SomeDB"
293                                          jdbcUserName="myid"
294                                          jdbcPassword="mypassword" />
295         <dc:QueryTemplate>
296             <![CDATA[
297                 SELECT * FROM student WHERE gzbtpid = '$requestContext.principalName'
298             ]]>
299         </dc:QueryTemplate>
300
301         <dc:Column columnName="gzbtpid" attributeID="uid" />
302         <dc:Column columnName="fqlft" attributeID="gpa" type="Float" />
303     </resolver:DataConnector>
304      -->
305
306     <!-- Example LDAP Connector -->
307     <!--
308     <resolver:DataConnector id="myLDAP" xsi:type="dc:LDAPDirectory"
309         ldapURL="ldap://ldap.example.org"
310         baseDN="ou=people,dc=example,dc=org"
311         principal="uid=myservice,ou=system"
312         principalCredential="myServicePassword">
313         <dc:FilterTemplate>
314             <![CDATA[
315                 (uid=$requestContext.principalName)
316             ]]>
317         </dc:FilterTemplate>
318     </resolver:DataConnector>
319     -->
320
321     <resolver:DataConnector xsi:type="dc:LDAPDirectory"
322     id="myLDAP"
323         ldapURL="ldap://mi-ldap-test-ula"
324         baseDN="ou=xxx,dc=ula,dc=ve"
325         principal="cn=xxx,dc=ula,dc=ve"
326         principalCredential="password">
327         <dc:FilterTemplate>
328             <![CDATA[
329                 (uid=$requestContext.principalName)
```

```
            ]]>
        </dc:FilterTemplate>

    </resolver:DataConnector>

    <!-- Computed targeted ID connector -->
    <!--
    <dc:ReturnAttributes>mail</dc:ReturnAttributes>
    <dc:ReturnAttributes>uid</dc:ReturnAttributes>
    <dc:ReturnAttributes>ou</dc:ReturnAttributes>

    <resolver:DataConnector xsi:type="dc:ComputedId"
                            id="computedID"
                            generatedAttributeID="computedID"
                            sourceAttributeID="uid"
                            salt="your random string here">
        <resolver:Dependency ref="myLDAP" />
    </resolver:DataConnector>
    -->

    <!-- ========================================== -->
    <!--      Principal Connectors                  -->
    <!-- ========================================== -->
    <resolver:PrincipalConnector xsi:type="pc:Transient" id="shibTransient"
        nameIDFormat="urn:mace:shibboleth:1.0:nameIdentifier"/>
    <resolver:PrincipalConnector xsi:type="pc:Transient" id="saml1Unspec"
        nameIDFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"/>
    <resolver:PrincipalConnector xsi:type="pc:Transient" id="saml2Transient"
        nameIDFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>

</resolver:AttributeResolver>
```

6.4. attribute-filter.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    This file is an EXAMPLE policy file.  While the policy presented in this
    example file is functional, it isn't very interesting.

    Deployers should refer to the Shibboleth 2 documentation for a complete list of components
    and their options.
-->
<afp:AttributeFilterPolicyGroup id="ShibbolethFilterPolicy"
                                xmlns:afp="urn:mace:shibboleth:2.0:afp" xmlns:basic="urn:mace:shibboleth:
                                    2.0:afp:mf:basic"
                                xmlns:saml="urn:mace:shibboleth:2.0:afp:mf:saml" xmlns:xsi="http://
                                    www.w3.org/2001/XMLSchema-instance"
                                xsi:schemaLocation="urn:mace:shibboleth:2.0:afp classpath:/schema/
                                    shibboleth-2.0-afp.xsd
                                                    urn:mace:shibboleth:2.0:afp:mf:basic classpath:/
                                                        schema/shibboleth-2.0-afp-mf-basic.xsd
                                                    urn:mace:shibboleth:2.0:afp:mf:saml classpath:/
                                                        schema/shibboleth-2.0-afp-mf-saml.xsd">

    <!--  Release the transient ID to anyone -->
    <afp:AttributeFilterPolicy id="releaseTransientIdToAnyone">
        <afp:PolicyRequirementRule xsi:type="basic:ANY"/>

        <afp:AttributeRule attributeID="transientId">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
        </afp:AttributeRule>

    </afp:AttributeFilterPolicy>

    <afp:AttributeFilterPolicy id="releaseUIDToAnyone">
        <afp:PolicyRequirementRule xsi:type="basic:ANY"/>
        <afp:AttributeRule attributeID="uid">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
        </afp:AttributeRule>

    </afp:AttributeFilterPolicy>

    <afp:AttributeFilterPolicy id="releaseMailToAnyone">
        <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://
            172.16.171.82/shibboleth" />
        <afp:AttributeRule attributeID="mail">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
        </afp:AttributeRule>
    </afp:AttributeFilterPolicy>

    <!--
        Release eduPersonEntitlement and the permissible values of eduPersonAffiliation
        to three specific SPs
    -->
    <afp:AttributeFilterPolicy>
        <afp:PolicyRequirementRule xsi:type="basic:OR">
        <basic:Rule xsi:type="basic:AttributeRequesterString" value="https://sp-webmail/shibboleth" />
        <basic:Rule xsi:type="basic:AttributeRequesterString" value="https://sp-wifi/shibboleth" />
            <basic:Rule xsi:type="basic:AttributeRequesterString" value="urn:example.org:sp:Portal" />
            <basic:Rule xsi:type="basic:AttributeRequesterString" value="urn:example.org:sp:SIS" />
            <basic:Rule xsi:type="basic:AttributeRequesterString" value="urn:example.org:sp:LMS" />
        </afp:PolicyRequirementRule>

        <afp:AttributeRule attributeID="uid">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
        </afp:AttributeRule>

        <afp:AttributeRule attributeID="mail">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
        </afp:AttributeRule>

    </afp:AttributeFilterPolicy>
    <!--
    <afp:AttributeFilterPolicy>
```

```
        <afp:PolicyRequirementRule xsi:type="basic:OR">
            <basic:Rule xsi:type="basic:AttributeRequesterString" value="urn:example.org:sp:Portal" />
            <basic:Rule xsi:type="basic:AttributeRequesterString" value="urn:example.org:sp:SIS" />
            <basic:Rule xsi:type="basic:AttributeRequesterString" value="urn:example.org:sp:LMS" />
        </afp:PolicyRequirementRule>

        <afp:AttributeRule attributeID="eduPersonAffiliation">
            <afp:PermitValueRule xsi:type="basic:OR">
                <basic:Rule xsi:type="basic:AttributeValueString" value="faculty" ignoreCase="true" />
                <basic:Rule xsi:type="basic:AttributeValueString" value="student" ignoreCase="true" />
                <basic:Rule xsi:type="basic:AttributeValueString" value="staff" ignoreCase="true" />
                <basic:Rule xsi:type="basic:AttributeValueString" value="alum" ignoreCase="true" />
                <basic:Rule xsi:type="basic:AttributeValueString" value="member" ignoreCase="true" />
                <basic:Rule xsi:type="basic:AttributeValueString" value="affiliate" ignoreCase="true" />
                <basic:Rule xsi:type="basic:AttributeValueString" value="employee" ignoreCase="true" />
                <basic:Rule xsi:type="basic:AttributeValueString" value="library-walk-in"
                    ignoreCase="true" />
            </afp:PermitValueRule>
        </afp:AttributeRule>

    </afp:AttributeFilterPolicy>
    -->

        <!-- Release the given name of the user to our portal service provider -->
<!--
    <afp:AttributeFilterPolicy>
        <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://sp-webmail/
            shibboleth" />
        <afp:AttributeRule attributeID="mail">
            <afp:PermitValueRule xsi:type="basic:ANY" />
        </afp:AttributeRule>
        <afp:AttributeRule attributeID="givenName">
            <afp:PermitValueRule xsi:type="basic:ANY" />
        </afp:AttributeRule>
        <afp:AttributeRule attributeID="ou">
            <afp:PermitValueRule xsi:type="basic:ANY" />
        </afp:AttributeRule>
        <afp:AttributeRule attributeID="uid">
            <afp:PermitValueRule xsi:type="basic:ANY" />
        </afp:AttributeRule>
    </afp:AttributeFilterPolicy>

    <afp:AttributeFilterPolicy>
        <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://sp-webmail/
            shibboleth" />
        <afp:AttributeRule attributeID="mail">
            <afp:PermitValueRule xsi:type="basic:ANY" />
        </afp:AttributeRule>
    </afp:AttributeFilterPolicy>
-->

    <afp:AttributeFilterPolicy>
        <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://
            172.16.171.82/shibboleth" />

        <afp:AttributeRule attributeID="givenName">
            <afp:PermitValueRule xsi:type="basic:ANY" />
        </afp:AttributeRule>

    </afp:AttributeFilterPolicy>

 <afp:AttributeFilterPolicy>
    <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://sp-webmail/
        shibboleth"/>

        <afp:AttributeRule attributeID="email">
            <afp:PermitValueRule xsi:type="basic:ANY" />
        </afp:AttributeRule>

    </afp:AttributeFilterPolicy>
```

```
130
131     <afp:AttributeFilterPolicy>
132         <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://sp-wifi:
                12081/shibboleth" />
133
134         <afp:AttributeRule attributeID="givenName">
135             <afp:PermitValueRule xsi:type="basic:ANY" />
136         </afp:AttributeRule>
137
138     </afp:AttributeFilterPolicy>
139
140  <afp:AttributeFilterPolicy>
141     <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://sp-wifi:12081/
            shibboleth"/>
142
143         <afp:AttributeRule attributeID="email">
144             <afp:PermitValueRule xsi:type="basic:ANY" />
145         </afp:AttributeRule>
146
147     </afp:AttributeFilterPolicy>
148
149  <afp:AttributeFilterPolicy>
150     <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://sp-test12081/
            shibboleth"/>
151
152         <afp:AttributeRule attributeID="email">
153             <afp:PermitValueRule xsi:type="basic:ANY" />
154         </afp:AttributeRule>
155
156     </afp:AttributeFilterPolicy>
157
158  <afp:AttributeFilterPolicy>
159     <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://sp-test2:12081/
            shibboleth"/>
160
161         <afp:AttributeRule attributeID="givenName">
162             <afp:PermitValueRule xsi:type="basic:ANY" />
163         </afp:AttributeRule>
164
165     </afp:AttributeFilterPolicy>
166
167
168 </afp:AttributeFilterPolicyGroup>
169
170
171
```

6.5. relying-party.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    This file is an EXAMPLE configuration file.

    This file specifies relying party dependent configurations for the IdP, for example, whether SAML assertions to a
    particular relying party should be signed.  It also includes metadata provider and credential definitions used
    when answering requests to a relying party.
-->
<rp:RelyingPartyGroup xmlns:rp="urn:mace:shibboleth:2.0:relying-party" xmlns:saml="urn:mace:shibboleth:2.0:relying-party:saml"
                      xmlns:metadata="urn:mace:shibboleth:2.0:metadata" xmlns:resource="urn:mace:shibboleth:2.0:resource"
                      xmlns:security="urn:mace:shibboleth:2.0:security" xmlns:samlsec="urn:mace:shibboleth:2.0:security:saml"
                      xmlns:samlmd="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:schemaLocation="urn:mace:shibboleth:2.0:relying-party classpath:/schema/shibboleth-2.0-relying-party.xsd
                                          urn:mace:shibboleth:2.0:relying-party:saml classpath:/schema/shibboleth-2.0-relying-party-saml.xsd
                                          urn:mace:shibboleth:2.0:metadata classpath:/schema/shibboleth-2.0-metadata.xsd
                                          urn:mace:shibboleth:2.0:resource classpath:/schema/shibboleth-2.0-resource.xsd
                                          urn:mace:shibboleth:2.0:security classpath:/schema/shibboleth-2.0-security.xsd
                                          urn:mace:shibboleth:2.0:security:saml classpath:/schema/shibboleth-2.0-security-policy-saml.xsd
                                          urn:oasis:names:tc:SAML:2.0:metadata classpath:/schema/saml-schema-metadata-2.0.xsd">

    <!-- ========================================== -->
    <!--      Relying Party Configurations          -->
    <!-- ========================================== -->
    <rp:AnonymousRelyingParty provider="https://idp-ula/idp/shibboleth" defaultSigningCredentialRef="IdPCredential"/>

    <rp:DefaultRelyingParty provider="https://idp-ula/idp/shibboleth" defaultSigningCredentialRef="IdPCredential">
        <!--
            Each attribute in these profiles configuration is set to its default value,
            that is, the values that would be in effect if those attributes were not present.
            We list them here so that people are aware of them (since they seem reluctant to
            read the documentation).
        -->
        <rp:ProfileConfiguration xsi:type="saml:ShibbolethSSOProfile" includeAttributeStatement="false"
                                 assertionLifetime="PT5M" signResponses="conditional" signAssertions="never"/>

        <rp:ProfileConfiguration xsi:type="saml:SAML1AttributeQueryProfile" assertionLifetime="PT5M"
                                 signResponses="conditional" signAssertions="never"/>

        <rp:ProfileConfiguration xsi:type="saml:SAML1ArtifactResolutionProfile" signResponses="conditional"
                                 signAssertions="never"/>

        <rp:ProfileConfiguration xsi:type="saml:SAML2SSOProfile" includeAttributeStatement="true"
                                 assertionLifetime="PT5M" assertionProxyCount="0"
                                 signResponses="never" signAssertions="always"
                                 encryptAssertions="conditional" encryptNameIds="never"/>

        <rp:ProfileConfiguration xsi:type="saml:SAML2ECPProfile" includeAttributeStatement="true"
                                 assertionLifetime="PT5M" assertionProxyCount="0"
                                 signResponses="never" signAssertions="always"
                                 encryptAssertions="conditional" encryptNameIds="never"/>

        <rp:ProfileConfiguration xsi:type="saml:SAML2AttributeQueryProfile"
                                 assertionLifetime="PT5M" assertionProxyCount="0"
```

```
                                signResponses="conditional" signAssertions="never"
                                encryptAssertions="conditional" encryptNameIds="never"/>

        <rp:ProfileConfiguration xsi:type="saml:SAML2ArtifactResolutionProfile"
                                signResponses="never" signAssertions="always"
                                encryptAssertions="conditional" encryptNameIds="never"/>

    </rp:DefaultRelyingParty>


    <!-- ========================================== -->
    <!--      Metadata Configuration                -->
    <!-- ========================================== -->
    <!-- MetadataProvider the combining other MetadataProviders -->
    <metadata:MetadataProvider id="ShibbolethMetadata" xsi:type="metadata:ChainingMetadataProvider">

        <!-- Load the IdP's own metadata.  This is necessary for artifact support. -->
        <metadata:MetadataProvider id="IdPMD" xsi:type="metadata:FilesystemMetadataProvider"
                                   metadataFile="/opt/shibboleth-idp/metadata/idp-metadata.xml"
                                   maxRefreshDelay="P1D" />

        <!-- Example metadata provider. -->
        <!-- Reads metadata from a URL and store a backup copy on the file system. -->
        <!-- Validates the signature of the metadata and filters out all by SP entities in order to save
            memory -->
        <!-- To use: fill in 'metadataURL' and 'backingFile' properties on MetadataResource element -->
        <!--
        <metadata:MetadataProvider id="URLMD" xsi:type="metadata:FileBackedHTTPMetadataProvider"
                          metadataURL="http://example.org/metadata.xml"
                          backingFile="/opt/shibboleth-idp/metadata/some-metadata.xml">
            <metadata:MetadataFilter xsi:type="metadata:ChainingFilter">
                <metadata:MetadataFilter xsi:type="metadata:RequiredValidUntil"
                                 maxValidityInterval="P7D" />
                <metadata:MetadataFilter xsi:type="metadata:SignatureValidation"
                                 trustEngineRef="shibboleth.MetadataTrustEngine"
                                 requireSignedMetadata="true" />
                <metadata:MetadataFilter xsi:type="metadata:EntityRoleWhiteList">
                    <metadata:RetainedRole>samlmd:SPSSODescriptor</metadata:RetainedRole>
                </metadata:MetadataFilter>
            </metadata:MetadataFilter>
        </metadata:MetadataProvider>
        -->

        <metadata:MetadataProvider id="SPWIFI" xsi:type="metadata:FileBackedHTTPMetadataProvider"
                          metadataURL="https://sp-wifi:12081/Shibboleth.sso/Metadata"
                          backingFile="/opt/shibboleth-idp/metadata/spwifi-metadata.xml">
            <metadata:MetadataFilter xsi:type="metadata:ChainingFilter">
                <metadata:MetadataFilter xsi:type="metadata:EntityRoleWhiteList">
                    <metadata:RetainedRole>samlmd:SPSSODescriptor</metadata:RetainedRole>
                </metadata:MetadataFilter>
            </metadata:MetadataFilter>
        </metadata:MetadataProvider>

        <metadata:MetadataProvider id="SPA" xsi:type="metadata:FileBackedHTTPMetadataProvider"
                          metadataURL="https://sp-webmail/Shibboleth.sso/Metadata"
                          backingFile="/opt/shibboleth-idp/metadata/spwebmail-metadata.xml">
            <metadata:MetadataFilter xsi:type="metadata:ChainingFilter">
                <metadata:MetadataFilter xsi:type="metadata:EntityRoleWhiteList">
                    <metadata:RetainedRole>samlmd:SPSSODescriptor</metadata:RetainedRole>
                </metadata:MetadataFilter>
            </metadata:MetadataFilter>
        </metadata:MetadataProvider>

    </metadata:MetadataProvider>


    <!-- ========================================== -->
    <!--      Security Configurations               -->
    <!-- ========================================== -->
    <security:Credential id="IdPCredential" xsi:type="security:X509Filesystem">
```

```
123         <security:PrivateKey>/opt/shibboleth-idp/credentials/idp.key</security:PrivateKey>
124         <security:Certificate>/opt/shibboleth-idp/credentials/idp.crt</security:Certificate>
125     </security:Credential>
126
127     <!-- Trust engine used to evaluate the signature on loaded metadata. -->
128     <!--
129     <security:TrustEngine id="shibboleth.MetadataTrustEngine"
            xsi:type="security:StaticExplicitKeySignature">
130         <security:Credential id="MyFederation1Credentials" xsi:type="security:X509Filesystem">
131             <security:Certificate>/opt/shibboleth-idp/credentials/federation1.crt</security:Certificate>
132         </security:Credential>
133     </security:TrustEngine>
134      -->
135
136     <!-- DO NOT EDIT BELOW THIS POINT -->
137     <!--
138         The following trust engines and rules control every aspect of security related to incoming
                messages.
139         Trust engines evaluate various tokens (like digital signatures) for trust worthiness while the
140         security policies establish a set of checks that an incoming message must pass in order to be
                considered
141         secure.  Naturally some of these checks require the validation of the tokens evaluated by the
                trust
142         engines and so you'll see some rules that reference the declared trust engines.
143     -->
144     <security:TrustEngine id="shibboleth.SignatureTrustEngine" xsi:type="security:SignatureChaining">
145         <security:TrustEngine id="shibboleth.SignatureMetadataExplicitKeyTrustEngine"
                xsi:type="security:MetadataExplicitKeySignature" metadataProviderRef="ShibbolethMetadata"/>
146         <security:TrustEngine id="shibboleth.SignatureMetadataPKIXTrustEngine"
                xsi:type="security:MetadataPKIXSignature" metadataProviderRef="ShibbolethMetadata"/>
147     </security:TrustEngine>
148
149     <security:TrustEngine id="shibboleth.CredentialTrustEngine" xsi:type="security:Chaining">
150         <security:TrustEngine id="shibboleth.CredentialMetadataExplictKeyTrustEngine"
                xsi:type="security:MetadataExplicitKey" metadataProviderRef="ShibbolethMetadata"/>
151         <security:TrustEngine id="shibboleth.CredentialMetadataPKIXTrustEngine"
                xsi:type="security:MetadataPKIXX509Credential" metadataProviderRef="ShibbolethMetadata"/>
152     </security:TrustEngine>
153
154     <security:SecurityPolicy id="shibboleth.ShibbolethSSOSecurityPolicy"
            xsi:type="security:SecurityPolicyType">
155         <security:Rule xsi:type="samlsec:Replay" required="false"/>
156         <security:Rule xsi:type="samlsec:IssueInstant" required="false"/>
157         <security:Rule xsi:type="samlsec:MandatoryIssuer"/>
158     </security:SecurityPolicy>
159
160     <security:SecurityPolicy id="shibboleth.SAML1AttributeQuerySecurityPolicy"
            xsi:type="security:SecurityPolicyType">
161         <security:Rule xsi:type="samlsec:Replay"/>
162         <security:Rule xsi:type="samlsec:IssueInstant"/>
163         <security:Rule xsi:type="samlsec:ProtocolWithXMLSignature"
                trustEngineRef="shibboleth.SignatureTrustEngine"/>
164         <security:Rule xsi:type="security:ClientCertAuth"
                trustEngineRef="shibboleth.CredentialTrustEngine"/>
165         <security:Rule xsi:type="samlsec:MandatoryIssuer"/>
166         <security:Rule xsi:type="security:MandatoryMessageAuthentication"/>
167     </security:SecurityPolicy>
168
169     <security:SecurityPolicy id="shibboleth.SAML1ArtifactResolutionSecurityPolicy"
            xsi:type="security:SecurityPolicyType">
170         <security:Rule xsi:type="samlsec:Replay"/>
171         <security:Rule xsi:type="samlsec:IssueInstant"/>
172         <security:Rule xsi:type="samlsec:ProtocolWithXMLSignature"
                trustEngineRef="shibboleth.SignatureTrustEngine"/>
173         <security:Rule xsi:type="security:ClientCertAuth"
                trustEngineRef="shibboleth.CredentialTrustEngine"/>
174         <security:Rule xsi:type="samlsec:MandatoryIssuer"/>
175         <security:Rule xsi:type="security:MandatoryMessageAuthentication"/>
176     </security:SecurityPolicy>
177
```

6.6. shibboleth2.xml

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    logger="syslog.logger" clockSkew="180">

    <!-- The OutOfProcess section contains properties affecting the shibd daemon. -->
    <OutOfProcess logger="shibd.logger">
        <!--
        <Extensions>
            <Library path="odbc-store.so" fatal="true"/>
        </Extensions>
        -->
    </OutOfProcess>

    <!--
    The InProcess section contains settings affecting web server modules.
    Required for IIS, but can be removed when using other web servers.
    -->
    <InProcess logger="native.logger">
        <ISAPI normalizeRequest="true" safeHeaderNames="true">
            <!--
            Maps IIS Instance ID values to the host scheme/name/port. The name is
            required so that the proper <Host> in the request map above is found without
            having to cover every possible DNS/IP combination the user might enter.
            -->
            <Site id="1" name="sp.example.org"/>
            <!--
            When the port and scheme are omitted, the HTTP request's port and scheme are used.
            If these are wrong because of virtualization, they can be explicitly set here to
            ensure proper redirect generation.
            -->
            <!--
            <Site id="42" name="virtual.example.org" scheme="https" port="443"/>
            -->
        </ISAPI>
    </InProcess>

    <!-- Only one listener can be defined, to connect in-process modules to shibd. -->
    <UnixListener address="shibd.sock"/>
    <!-- <TCPListener address="127.0.0.1" port="1600" acl="127.0.0.1"/> -->

    <!-- This set of components stores sessions and other persistent data in daemon memory. -->
    <StorageService type="Memory" id="mem" cleanupInterval="900"/>
    <SessionCache type="StorageService" StorageService="mem" cacheAssertions="false"
                  cacheAllowance="900" inprocTimeout="900" cleanupInterval="900"/>
    <ReplayCache StorageService="mem"/>
    <ArtifactMap artifactTTL="180"/>

    <!-- This set of components stores sessions and other persistent data in an ODBC database. -->
    <!--
    <StorageService type="ODBC" id="db" cleanupInterval="900">
        <ConnectionString>
        DRIVER=drivername;SERVER=dbserver;UID=shibboleth;PWD=password;DATABASE=shibboleth;APP=Shibboleth
        </ConnectionString>
    </StorageService>
    <SessionCache type="StorageService" StorageService="db" cacheAssertions="false"
                  cacheTimeout="3600" inprocTimeout="900" cleanupInterval="900"/>
    <ReplayCache StorageService="db"/>
    <ArtifactMap StorageService="db" artifactTTL="180"/>
    -->

    <!--
    To customize behavior for specific resources on Apache, and to link vhosts or
    resources to ApplicationOverride settings below, use web server options/commands.
    See https://spaces.internet2.edu/display/SHIB2/NativeSPConfigurationElements for help.

    For examples with the RequestMap XML syntax instead, see the example-shibboleth2.xml
    file, and the https://spaces.internet2.edu/display/SHIB2/NativeSPRequestMapHowTo topic.
```

```
71      -->
72      <RequestMapper type="Native">
73          <RequestMap>
74              <!--
75              The example requires a session for documents in /secure on the containing host with http and
76              https on the default ports. Note that the name and port in the <Host> elements MUST match
77              Apache's ServerName and Port directives or the IIS Site name in the <ISAPI> element above.
78              -->
79              <Host name="sp.example.org">
80                  <Path name="secure" authType="shibboleth" requireSession="true"/>
81              </Host>
82              <!-- Example of a second vhost mapped to a different applicationId. -->
83              <!--
84              <Host name="admin.example.org" applicationId="admin" authType="shibboleth"
                    requireSession="true"/>
85              -->
86          </RequestMap>
87      </RequestMapper>
88
89      <!--
90      The ApplicationDefaults element is where most of Shibboleth's SAML bits are defined.
91      Resource requests are mapped by the RequestMapper to an applicationId that
92      points into to this section (or to the defaults here).
93      <ApplicationDefaults entityID="https://sp.example.org/shibboleth"
94                           REMOTE_USER="eppn persistent-id targeted-id"
95                           signing="false" encryption="false">
96      -->
97      <ApplicationDefaults id="default" policyId="default"
98          entityID="https://spwifi/shibboleth"
99          REMOTE_USER="mail givenName uid eppn"
100         signing="false" encryption="false">
101
102             <!--
103             Controls session lifetimes, address checks, cookie handling, and the protocol handlers.
104             You MUST supply an effectively unique handlerURL value for each of your applications.
105             The value defaults to /Shibboleth.sso, and should be a relative path, with the SP computing
106             a relative value based on the virtual host. Using handlerSSL="true", the default, will force
107             the protocol to be https. You should also add a cookieProps setting of "; path=/; secure"
108             in that case. Note that while we default checkAddress to "false", this has a negative
109             impact on the security of the SP. Stealing cookies/sessions is much easier with this disabled.
110             -->
111             <Sessions lifetime="28800" timeout="3600" checkAddress="false"
112                 handlerURL="/Shibboleth.sso" handlerSSL="false" relayState="ss:mem"
113                 exportLocation="http://localhost/Shibboleth.sso/GetAssertion" exportACL="127.0.0.1"
114                 idpHistory="false" idpHistoryDays="7">
115
116                 <!--
117                 The "stripped down" files use the shorthand syntax for configuring handlers.
118                 This uses the old "every handler specified directly" syntax. You can replace
119                 or supplement the new syntax following these examples.
120                 -->
121
122                 <!--
123                 SessionInitiators handle session requests and relay them to a Discovery page,
124                 or to an IdP if possible. Automatic session setup will use the default or first
125                 element (or requireSessionWith can specify a specific id to use).
126                 -->
127
128                 <!-- Default directs to a specific IdP (favoring SAML 2 over Shib 1). -->
129                     <!--
130                 <SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="Login"
131                                   entityID="https://idp.example.org/shibboleth">
132 ▸               ▸
133                     <SessionInitiator type="SAML2" template="bindingTemplate.html"/>
134                     <SessionInitiator type="Shib1"/>
135
136                     To allow for >1 IdP, remove entityID property from Chaining element and add
137                     *either* of the SAMLDS or WAYF handlers below:
138
139                     <SessionInitiator type="SAMLDS" URL="https://ds.example.org/DS/WAYF"/>
```

```
            <SessionInitiator type="WAYF" URL="https://wayf.example.org/WAYF"/>
        </SessionInitiator>
            -->
    <SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="Intranet"
                relayState="cookie" entityID="https://172.16.171.83/idp/shibboleth">
            <SessionInitiator type="SAML2" acsIndex="1" template="bindingTemplate.html"/>
            <SessionInitiator type="Shib1" acsIndex="5"/>
        </SessionInitiator>


        <!--
        md:AssertionConsumerService locations handle specific SSO protocol bindings,
        such as SAML 2.0 POST or SAML 1.1 Artifact. The isDefault and index attributes
        are used when sessions are initiated to determine how to tell the IdP where and
        how to return the response.
        -->
        <md:AssertionConsumerService Location="/SAML2/POST" index="1"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
        <md:AssertionConsumerService Location="/SAML2/POST-SimpleSign" index="2"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"/>
        <md:AssertionConsumerService Location="/SAML2/Artifact" index="3"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
        <md:AssertionConsumerService Location="/SAML2/ECP" index="4"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"/>
        <md:AssertionConsumerService Location="/SAML/POST" index="5"
            Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
        <md:AssertionConsumerService Location="/SAML/Artifact" index="6"
            Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>

        <!-- LogoutInitiators enable SP-initiated local or global/single logout of sessions. -->
        <LogoutInitiator type="Chaining" Location="/Logout">
            <LogoutInitiator type="SAML2" template="bindingTemplate.html"/>
            <LogoutInitiator type="Local"/>
        </LogoutInitiator>

        <!-- md:SingleLogoutService locations handle single logout (SLO) protocol messages. -->
        <md:SingleLogoutService Location="/SLO/SOAP"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
        <md:SingleLogoutService Location="/SLO/Redirect" conf:template="bindingTemplate.html"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"/>
        <md:SingleLogoutService Location="/SLO/POST" conf:template="bindingTemplate.html"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
        <md:SingleLogoutService Location="/SLO/Artifact" conf:template="bindingTemplate.html"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>

        <!-- md:ManageNameIDService locations handle NameID management (NIM) protocol messages. -->
        <md:ManageNameIDService Location="/NIM/SOAP"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
        <md:ManageNameIDService Location="/NIM/Redirect" conf:template="bindingTemplate.html"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"/>
        <md:ManageNameIDService Location="/NIM/POST" conf:template="bindingTemplate.html"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
        <md:ManageNameIDService Location="/NIM/Artifact" conf:template="bindingTemplate.html"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>

        <!--
        md:ArtifactResolutionService locations resolve artifacts issued when using the
        SAML 2.0 HTTP-Artifact binding on outgoing messages, generally uses SOAP.
        -->
        <md:ArtifactResolutionService Location="/Artifact/SOAP" index="1"
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>

        <!-- Extension service that generates "approximate" metadata based on SP configuration. -->
        <Handler type="MetadataGenerator" Location="/Metadata" signing="false"/>

        <!-- Status reporting service. -->
        <Handler type="Status" Location="/Status" acl="127.0.0.1"/>

        <!-- Session diagnostic service. -->
        <Handler type="Session" Location="/Session" showAttributeValues="false"/>
```

```
210
211             <!-- JSON feed of discovery information. -->
212             <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>
213         </Sessions>
214
215         <!--
216         Allows overriding of error template information/filenames. You can
217         also add attributes with values that can be plugged into the templates.
218         -->
219         <Errors supportContact="root@localhost"
220             logoLocation="/shibboleth-sp/logo.jpg"
221             styleSheet="/shibboleth-sp/main.css"/>
222
223         <!--
224         Uncomment and modify to tweak settings for specific IdPs or groups. Settings here
225         generally match those allowed by the <ApplicationDefaults> element.
226         -->
227         <!--
228         <RelyingParty Name="SpecialFederation" keyName="SpecialKey"/>
229         -->
230
231         <!-- Example of remotely supplied batch of signed metadata. -->
232         <!--
233         <MetadataProvider type="XML" uri="http://federation.org/federation-metadata.xml"
234               backingFilePath="federation-metadata.xml" reloadInterval="7200">
235             <MetadataFilter type="RequireValidUntil" maxValidityInterval="2419200"/>
236             <MetadataFilter type="Signature" certificate="fedsigner.pem"/>
237         </MetadataProvider>
238         -->
239
240         <MetadataProvider type="XML" uri="https://idp-ula/idp/shibboleth"
241               backingFilePath="idp-metadata.xml" reloadInterval="7200">
242         </MetadataProvider>
243
244         <!-- Example of locally maintained metadata. -->
245         <!--
246         <MetadataProvider type="XML" file="partner-metadata.xml"/>
247         -->
248
249         <!-- TrustEngines run in order to evaluate peer keys and certificates. -->
250         <TrustEngine type="ExplicitKey"/>
251         <TrustEngine type="PKIX"/>
252
253         <!-- Map to extract attributes from SAML assertions. -->
254         <AttributeExtractor type="XML" validate="true" path="attribute-map.xml"/>
255
256         <!-- Use a SAML query if no attributes are supplied during SSO. -->
257         <AttributeResolver type="Query" subjectMatch="true"/>
258
259         <!-- Default filtering policy for recognized attributes, lets other data pass. -->
260         <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>
261
262         <!-- Simple file-based resolver for using a single keypair. -->
263         <CredentialResolver type="File" key="/etc/ssl/private/spa.key" certificate="/etc/ssl/certs/
                spa.crt"/>
264
265         <!--
266         The default settings can be overridden by creating ApplicationOverride elements (see
267         the https://spaces.internet2.edu/display/SHIB2/NativeSPApplicationOverride topic).
268         Resource requests are mapped by web server commands, or the RequestMapper, to an
269         applicationId setting.
270
271         Example of a second application (for a second vhost) that has a different entityID.
272         Resources on the vhost would map to an applicationId of "admin":
273         -->
274         <!--
275         <ApplicationOverride id="admin" entityID="https://admin.example.org/shibboleth"/>
276         -->
277     </ApplicationDefaults>
278
```

```
279     <!-- Policies that determine how to process and authenticate runtime messages. -->
280     <SecurityPolicyProvider type="XML" validate="true" path="security-policy.xml"/>
281
282     <!-- Low-level configuration about protocols and bindings available for use. -->
283     <ProtocolProvider type="XML" validate="true" reloadChanges="false" path="protocols.xml"/>
284
285 </SPConfig>
286
```

6.7. attribute-map.xml

```xml
<Attributes xmlns="urn:mace:shibboleth:2.0:attribute-map" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <!-- First some useful eduPerson attributes that many sites might use. -->

    <Attribute name="urn:mace:dir:attribute-def:eduPersonPrincipalName" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>

    <Attribute name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation" id="affiliation">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9" id="affiliation">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
    </Attribute>

    <Attribute name="urn:mace:dir:attribute-def:eduPersonAffiliation" id="unscoped-affiliation">
        <AttributeDecoder xsi:type="StringAttributeDecoder" caseSensitive="false"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" id="unscoped-affiliation">
        <AttributeDecoder xsi:type="StringAttributeDecoder" caseSensitive="false"/>
    </Attribute>

    <Attribute name="urn:mace:dir:attribute-def:eduPersonEntitlement" id="entitlement"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" id="entitlement"/>

    <!-- A persistent id attribute that supports personalized anonymous access. -->

    <!-- First, the deprecated/incorrect version, decoded as a scoped string: -->
    <Attribute name="urn:mace:dir:attribute-def:eduPersonTargetedID" id="targeted-id">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
        <!-- <AttributeDecoder xsi:type="NameIDFromScopedAttributeDecoder" formatter="$NameQualifier!$SPNameQualifier!$Name"
            defaultQualifiers="true"/> -->
    </Attribute>

    <!-- Second, an alternate decoder that will decode the incorrect form into the newer form. -->
    <!--
    <Attribute name="urn:mace:dir:attribute-def:eduPersonTargetedID" id="persistent-id">
        <AttributeDecoder xsi:type="NameIDFromScopedAttributeDecoder" formatter="$NameQualifier!$SPNameQualifier!$Name"
            defaultQualifiers="true"/>
    </Attribute>
    -->

    <!-- Third, the new version (note the OID-style name): -->
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" id="persistent-id">
        <AttributeDecoder xsi:type="NameIDAttributeDecoder" formatter="$NameQualifier!$SPNameQualifier!$Name" defaultQualifiers="true"/>
    </Attribute>

    <!-- Fourth, the SAML 2.0 NameID Format: -->
    <Attribute name="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" id="persistent-id">
        <AttributeDecoder xsi:type="NameIDAttributeDecoder" formatter="$NameQualifier!$SPNameQualifier!$Name" defaultQualifiers="true"/>
    </Attribute>

    <!-- Some more eduPerson attributes, uncomment these to use them... -->
    <!--
    <Attribute name="urn:mace:dir:attribute-def:eduPersonPrimaryAffiliation" id="primary-affiliation">
        <AttributeDecoder xsi:type="StringAttributeDecoder" caseSensitive="false"/>
    </Attribute>
    <Attribute name="urn:mace:dir:attribute-def:eduPersonNickname" id="nickname"/>
    <Attribute name="urn:mace:dir:attribute-def:eduPersonPrimaryOrgUnitDN" id="primary-orgunit-dn"/>
    <Attribute name="urn:mace:dir:attribute-def:eduPersonOrgUnitDN" id="orgunit-dn"/>
    <Attribute name="urn:mace:dir:attribute-def:eduPersonOrgDN" id="org-dn"/>

    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.5" id="primary-affiliation">
        <AttributeDecoder xsi:type="StringAttributeDecoder" caseSensitive="false"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.2" id="nickname"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.8" id="primary-orgunit-dn"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.4" id="orgunit-dn"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.3" id="org-dn"/>

    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.11" id="assurance"/>

    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.5.1.1" id="member"/>

    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.6.1.1" id="eduCourseOffering"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.6.1.2" id="eduCourseMember"/>
    -->

    <!--Examples of LDAP-based attributes, uncomment to use these... -->
    <Attribute name="urn:mace:dir:attribute-def:cn" id="cn"/>
    <Attribute name="urn:mace:dir:attribute-def:sn" id="sn"/>
    <Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>
    <Attribute name="urn:mace:dir:attribute-def:mail" id="mail"/>
    <Attribute name="urn:mace:dir:attribute-def:telephoneNumber" id="telephoneNumber"/>
    <Attribute name="urn:mace:dir:attribute-def:ou" id="ou"/>
    <Attribute name="urn:mace:dir:attribute-def:title" id="title"/>
    <Attribute name="urn:mace:dir:attribute-def:initials" id="initials"/>
    <Attribute name="urn:mace:dir:attribute-def:description" id="description"/>
    <Attribute name="urn:mace:dir:attribute-def:carLicense" id="carLicense"/>
    <Attribute name="urn:mace:dir:attribute-def:departmentNumber" id="departmentNumber"/>
    <Attribute name="urn:mace:dir:attribute-def:displayName" id="displayName"/>
    <Attribute name="urn:mace:dir:attribute-def:employeeNumber" id="employeeNumber"/>
    <Attribute name="urn:mace:dir:attribute-def:employeeType" id="employeeType"/>
    <Attribute name="urn:mace:dir:attribute-def:preferredLanguage" id="preferredLanguage"/>
    <Attribute name="urn:mace:dir:attribute-def:manager" id="manager"/>
    <Attribute name="urn:mace:dir:attribute-def:seeAlso" id="seeAlso"/>
    <Attribute name="urn:mace:dir:attribute-def:facsimileTelephoneNumber" id="facsimileTelephoneNumber"/>
```

```
    <Attribute name="urn:mace:dir:attribute-def:street" id="street"/>
    <Attribute name="urn:mace:dir:attribute-def:postOfficeBox" id="postOfficeBox"/>
    <Attribute name="urn:mace:dir:attribute-def:postalCode" id="postalCode"/>
    <Attribute name="urn:mace:dir:attribute-def:st" id="st"/>
    <Attribute name="urn:mace:dir:attribute-def:l" id="l"/>
    <Attribute name="urn:mace:dir:attribute-def:o" id="o"/>
    <Attribute name="urn:mace:dir:attribute-def:ou" id="ou"/>
    <Attribute name="urn:mace:dir:attribute-def:businessCategory" id="businessCategory"/>
    <Attribute name="urn:mace:dir:attribute-def:physicalDeliveryOfficeName" id="physicalDeliveryOfficeName"/>

    <Attribute name="urn:oid:2.5.4.3" id="cn"/>
    <Attribute name="urn:oid:2.5.4.4" id="sn"/>
    <Attribute name="urn:oid:2.5.4.42" id="givenName"/>
    <Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
    <Attribute name="urn:oid:2.5.4.20" id="telephoneNumber"/>
    <Attribute name="urn:oid:2.5.4.12" id="title"/>
    <Attribute name="urn:oid:2.5.4.43" id="initials"/>
    <Attribute name="urn:oid:2.5.4.13" id="description"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.1" id="carLicense"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.2" id="departmentNumber"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.3" id="employeeNumber"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.4" id="employeeType"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.39" id="preferredLanguage"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/>
    <Attribute name="urn:oid:0.9.2342.19200300.100.1.10" id="manager"/>
    <Attribute name="urn:oid:2.5.4.34" id="seeAlso"/>
    <Attribute name="urn:oid:2.5.4.23" id="facsimileTelephoneNumber"/>
    <Attribute name="urn:oid:2.5.4.9" id="street"/>
    <Attribute name="urn:oid:2.5.4.18" id="postOfficeBox"/>
    <Attribute name="urn:oid:2.5.4.17" id="postalCode"/>
    <Attribute name="urn:oid:2.5.4.8" id="st"/>
    <Attribute name="urn:oid:2.5.4.7" id="l"/>
    <Attribute name="urn:oid:2.5.4.10" id="o"/>
    <Attribute name="urn:oid:2.5.4.11" id="ou"/>
    <Attribute name="urn:oid:2.5.4.15" id="businessCategory"/>
    <Attribute name="urn:oid:2.5.4.19" id="physicalDeliveryOfficeName"/>

</Attributes>

```

6.8. config default.php

```php
<?php

/**
 * SquirrelMail Login Authentication Plugin
 *
 * Copyright (c) 2004-2012 Paul Lesniewski <paul@squirrelmail.org>,
 * Copyright (c) 2001 Tyler Akins
 *
 * Licensed under the GNU GPL. For full terms see the file COPYING.
 *
 * @package plugins
 * @subpackage login_auth
 *
 */

global $normal_login_behavior, $authenticated_username_location,
       $authenticated_password_location, $external_auth_validation_type,
       $use_alternate_signout_page, $trusted_saml_username, $trusted_saml_password,
       $required_environment_variable, $required_environment_variable_value_type,
       $required_environment_variable_value, $external_session_expiration_relogin_link,
       $authenticated_saml_compress_assertion;


// Should logins using the normal login page be allowed if the user
// has not authenticated already via other means?
//
//    1  =  Yes, allow the user to log in using the standard SquirrelMail login page
//    0  =  No, don't allow webmail-only logins - display error page
//    <address of external login page> = Redirect unauthenticated users
//                                       to external login page; %s in this
//                                       address will be replaced with the
//                                       address of the SquirrelMail login
//                                       page and %e will be replaced with
//                                       with the URL-encoded version
//                                       of the SquirrelMail login address
//
// Normally, it isn't helpful to your users if you set this to 0 since
// it's not clear for them where they need to go to log in.  Examples:
//
// $normal_login_behavior = 'https://example.com/sso/login?return=%e';
// $normal_login_behavior = 0;
//
//$normal_login_behavior = 0;
$normal_login_behavior = 'https://spwebmail/Shibboleth.sso/Login?return=%e';
//$normal_login_behavior = 1;


// This plugin provides an alternate landing page for the "Signout" link
// that explains to users that they need to close their browser in order
// to actually sign out.  If this is the case for you (typical for HTTP
// authentication and the like), then you should enable the use of this
// special page.
//
// If you don't use this, it may be prudent if you set the SquirrelMail
// $signout_page configuration variable (or see config/conf.pl -->
// 1. Organization Preferences --> 5. Signout Page) to point to a page
// that actually ends the user's login session (perhaps by logging them
// out of your single sign-on system).
//
// Another option is that you can set this value to point to an external
// logout page (with SquirrelMail's logout address possibly included in
// it - use either the "%s" or "%e" replacement to specify where it
// should be included).
//
//    0  =  Do not use special signout page
//    1  =  Use special signout page
//    <address of external logout page> = Must be a web address of an
//                                        external logout page, %s will
//                                        be replaced in the address
//                                        with the SquirrelMail logout
//                                        address and %e will be replaced
//                                        with the URL-encoded version
//                                        of the SquirrelMail logout address
//
// Here's an example of an external logout page
//
// $use_alternate_signout_page = 'https://example.com/sso/logout?return=%e';
//
$use_alternate_signout_page = 1;


// What are the names of the server environment variables that hold
// the user credentials?  If you'd like to search a list of possibile
// variable names, you may specify these as a list (array), in which
// case the first one in your list that exists in the server environment
// is used.  For example:
//
// $authenticated_username_location = array('REMOTE_USER', 'PHP_AUTH_USER');
//
// NOTE that not all types of external authentication provide both of
// these; in such a case, these are ignored.
//
// An example when using the "authenticated_saml" module:
//
// $authenticated_username_location = 'REMOTE_USER';
// $authenticated_password_location = "MELLON_SAML_RESPONSE";
//
// An example when using the "trusted_saml" module:
```

```
//
// $authenticated_username_location = 'mail';
// $authenticated_password_location = 'Shib-Session-ID';
//
// These defaults are typical for HTTP authentication:
//$authenticated_username_location = 'PHP_AUTH_USER';
//$authenticated_password_location = 'PHP_AUTH_PW';
//$authenticated_username_location = 'PHP_AUTH_USER';
//$authenticated_password_location = 'Shib-Session-ID';
//$authenticated_username_location = 'mail';
$authenticated_username_location = 'uid';
$authenticated_password_location = 'Shib-Session-ID';


// How should the external authentication be identified and validated?
// What IMAP credentials should be used to access the user's email?
//
// The answers to this question will depend on your external authentication
// system.  The most simple variety is standard HTTP authentication, but
// other possibilities are one of several single sign-on systems, etc.
//
// This plugin provides functionality that knows how to see and use HTTP
// authentication credentials as well as a couple modules that can detect
// and process SAML (single sign-on) assertions.
//
// You can also define your own function tailored to your particular
// needs.  An example of such a function is included herein.
//
//
//
// Set this to "http_auth" if you use HTTP authentication.  In this case,
// $authenticated_username_location and $authenticated_password_location
// should usually be set to "PHP_AUTH_USER" and "PHP_AUTH_PW" respectively.
//
//
//
// Set this to to "authenticated_saml" to take advantage of the SAML
// single sign-on module that will ask the IMAP server to authenticate
// the SAML assertion in place of password authentication.
//
// This module knows how to find a SAML assertion provided by a
// SAML-compliant authentication module in the web server (such as
// mod_auth_mellon - see https://code.google.com/p/modmellon/ ) and
// pass it to the IMAP server instead of a password.  The IMAP server
// will need to use an authentication mechanism that knows how to
// validate the assertion - one package that provides such mechanisms
// is crudesaml (see http://ftp.espci.fr/pub/crudesaml/README )
// (Note that you must set all this up before using this plugin).
//
// Note that you need to set $authenticated_username_location
// and $authenticated_password_location to the values that correspond
// to the username and the SAML assertion in the server environment.
// For example, if you're using the mod_auth_mellon implementation,
// you'll usually want to set $authenticated_username_location to
// "REMOTE_USER" and $authenticated_password_location to "MELLON_SAML_RESPONSE".
//
//
//
// Set this to "trusted_saml" to take advantage of the SAML single
// sign-on module that treats the presence of a SAML token in the web
// server environment as authoritative (it trusts that the web server
// has properly authenticated the user and that the token would not be
// available if the user were not authenticated).
//
// This module knows how to find a SAML assertion provided by a
// SAML-compliant authentication module in the web server (such as
// Shibboleth - see http://shibboleth.internet2.edu/about.html or
// mod_auth_tkt - see http://www.openfusion.com.au/labs/mod_auth_tkt ),
// after which it will use a shared secret to log the user in to
// the IMAP server.  This requires that the IMAP server support
// the SASL PLAIN login mechanism with the ability to accept separate
// authorization and authentication identities (for Dovecot, refer
// to the "master user" feature:
// http://wiki.dovecot.org/Authentication/MasterUsers and for Cyrus,
// refer to
// http://cyrusimap.web.cmu.edu/docs/cyrus-imapd/2.3.16/install-auth.php
// and http://cyrusimap.web.cmu.edu/docs/cyrus-sasl/2.1.23/sysadmin.php ).
// This also requires that SquirrelMail version 1.4.23 or above and that
// you set SquirrelMail's IMAP authentication mechanism to "plain".  You
// can do that in the main SquirrelMail configuration file or use the
// configuration tool to do so: config/conf.pl ==> 2. Server Settings ==>
// A. Update IMAP Settings ==> 6. Authentication type
//
// Note that you need to set $authenticated_username_location
// and $authenticated_password_location to the values that correspond
// to the username and the SAML assertion in the server environment.
// For example, if you're using the Shibboleth implementation, you'll
// usually want to set $authenticated_username_location to "uid" and
// $authenticated_password_location to "Shib-Session-ID".  In some more
// complex environments, the $authenticated_username_location might
// need to be set to a list, such as (remember, the first one in
// this list that is non-empty is used):
//
// $authenticated_username_location = array('mail', 'eduPersonPrincipalName', 'irisMailMainAddress', 'mail', 'uid');
//
// Some single sign-on implementations may not populate the server
// environment with any SAML data, only providing the authenticated
// user name (mod_auth_tkt?).  In this case, you'll typically want
// to set BOTH $authenticated_username_location AND
// $authenticated_password_location to "REMOTE_USER".  While this may
```

6.9. auth-master.conf

```
# Authentication for master users. Included from 10-auth.conf.

# By adding master=yes setting inside a passdb you make the passdb a list
# of "master users", who can log in as anyone else.
# <doc/wiki/Authentication.MasterUsers.txt>

# Example master user passdb using passwd-file. You can use any passdb though.
#passdb {
#  driver = passwd-file
#  master = yes
#  args = /etc/dovecot/master-users
#
#  # Unless you're using PAM, you probably still want the destination user to
#  # be looked up from passdb that it really exists. pass=yes does that.
#  pass = yes
#}
auth_master_user_separator = *
passdb {
  driver = passwd-file
  args = /etc/dovecot/passwd.masterusers
  master = yes
  pass = no
}
passdb {
  driver = pam
}
userdb {
  driver = passwd
}

```

6.10. testidpredula.jmx

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="2.9" jmeter="3.0 r1743807">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Pruebas SSO Shibboleth RedULA" enabled="true">
      <stringProp name="TestPlan.comments"></stringProp>
      <boolProp name="TestPlan.functional_mode">false</boolProp>
      <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
      <elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel" testclass="Arguments"
          testname="User Defined Variables" enabled="true">
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
      <stringProp name="TestPlan.user_define_classpath"></stringProp>
    </TestPlan>
    <hashTree>
      <Arguments guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">
        <collectionProp name="Arguments.arguments">
          <elementProp name="ThreadCount" elementType="Argument">
            <stringProp name="Argument.name">ThreadCount</stringProp>
            <stringProp name="Argument.value">800</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
            <stringProp name="Argument.desc">Number of Threads</stringProp>
          </elementProp>
          <elementProp name="Duration" elementType="Argument">
            <stringProp name="Argument.name">Duration</stringProp>
            <stringProp name="Argument.value">100</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
            <stringProp name="Argument.desc">in seconds</stringProp>
          </elementProp>
          <elementProp name="RampUpPeriod" elementType="Argument">
            <stringProp name="Argument.name">RampUpPeriod</stringProp>
            <stringProp name="Argument.value">420</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
            <stringProp name="Argument.desc">in seconds</stringProp>
          </elementProp>
          <elementProp name="IdPHost" elementType="Argument">
            <stringProp name="Argument.name">IdPHost</stringProp>
            <stringProp name="Argument.value">https://idp-ula</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
          </elementProp>
          <elementProp name="IdPPort" elementType="Argument">
            <stringProp name="Argument.name">IdPPort</stringProp>
            <stringProp name="Argument.value">443</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
          </elementProp>
          <elementProp name="IdPContext" elementType="Argument">
            <stringProp name="Argument.name">IdPContext</stringProp>
            <stringProp name="Argument.value">idp</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
          </elementProp>
          <elementProp name="ProviderId" elementType="Argument">
            <stringProp name="Argument.name">ProviderId</stringProp>
            <stringProp name="Argument.value">https://idp-ula/shibboleth</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
          </elementProp>
          <elementProp name="startupDelay" elementType="Argument">
            <stringProp name="Argument.name">startupDelay</stringProp>
            <stringProp name="Argument.value">10</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
            <stringProp name="Argument.desc">in seconds</stringProp>
          </elementProp>
          <elementProp name="ShibSP" elementType="Argument">
            <stringProp name="Argument.name">ShibSP</stringProp>
            <stringProp name="Argument.value">https://sp-ula</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
          </elementProp>
        </collectionProp>
      </Arguments>
      <hashTree/>
      <CSVDataSet guiclass="TestBeanGUI" testclass="CSVDataSet" testname="./usuarios-ulatest.csv" enabled="true">
        <stringProp name="filename">/Users/danielagutierrez2/Documents/clases/tesis/users.csv</stringProp>
        <stringProp name="fileEncoding"></stringProp>
        <stringProp name="variableNames">User,Password</stringProp>
        <stringProp name="delimiter">,</stringProp>
        <boolProp name="quotedData">false</boolProp>
        <boolProp name="recycle">true</boolProp>
        <boolProp name="stopThread">false</boolProp>
        <stringProp name="shareMode">shareMode.all</stringProp>
      </CSVDataSet>
      <hashTree/>
      <CookieManager guiclass="CookiePanel" testclass="CookieManager" testname="HTTP Cookie Manager" enabled="true">
        <collectionProp name="CookieManager.cookies"/>
        <boolProp name="CookieManager.clearEachIteration">true</boolProp>
        <stringProp name="CookieManager.implementation">org.apache.jmeter.protocol.http.control.HC4CookieHandler</stringProp>
      </CookieManager>
      <hashTree/>
      <ConfigTestElement guiclass="HttpDefaultsGui" testclass="ConfigTestElement" testname="HTTP Request Defaults" enabled="true">
        <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User
            Defined Variables" enabled="true">
          <collectionProp name="Arguments.arguments"/>
        </elementProp>
        <stringProp name="HTTPSampler.domain"></stringProp>
        <stringProp name="HTTPSampler.port"></stringProp>
        <stringProp name="HTTPSampler.connect_timeout">120000</stringProp>
        <stringProp name="HTTPSampler.response_timeout">120000</stringProp>
        <stringProp name="HTTPSampler.protocol">https</stringProp>
        <stringProp name="HTTPSampler.contentEncoding"></stringProp>
        <stringProp name="HTTPSampler.path"></stringProp>
        <stringProp name="HTTPSampler.implementation">HttpClient4</stringProp>
        <stringProp name="HTTPSampler.concurrentPool">4</stringProp>
      </ConfigTestElement>
```

```
    <hashTree/>
    <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Shibboleth NoSilver SSO Test" enabled="true">
      <stringProp name="ThreadGroup.on_sample_error">startnextloop</stringProp>
      <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel" testclass="LoopController"
          testname="Loop Controller" enabled="true">
        <boolProp name="LoopController.continue_forever">false</boolProp>
        <intProp name="LoopController.loops">-1</intProp>
      </elementProp>
      <stringProp name="ThreadGroup.num_threads">${ThreadCount}</stringProp>
      <stringProp name="ThreadGroup.ramp_time">${RampUpPeriod}</stringProp>
      <longProp name="ThreadGroup.start_time">1435179289000</longProp>
      <longProp name="ThreadGroup.end_time">1435179289000</longProp>
      <boolProp name="ThreadGroup.scheduler">false</boolProp>
      <stringProp name="ThreadGroup.duration">${Duration}</stringProp>
      <stringProp name="ThreadGroup.delay">${StartupDelay}</stringProp>
    </ThreadGroup>
    <hashTree>
      <ModuleController guiclass="ModuleControllerGui" testclass="ModuleController" testname="Get Login Page" enabled="true">
        <collectionProp name="ModuleController.node_path">
          <stringProp name="-1227702913">WorkBench</stringProp>
          <stringProp name="-1408792136">Shibboleth SSO Tests</stringProp>
          <stringProp name="342767694">Get NoSilver Login Page</stringProp>
        </collectionProp>
      </ModuleController>
      <hashTree/>
      <ModuleController guiclass="ModuleControllerGui" testclass="ModuleController" testname="POST SAMLResponse Credentials"
          enabled="true">
        <collectionProp name="ModuleController.node_path">
          <stringProp name="-1227702913">WorkBench</stringProp>
          <stringProp name="-1408792136">Shibboleth SSO Tests</stringProp>
          <stringProp name="1293039389">POST SAMLReponse Credentials</stringProp>
        </collectionProp>
      </ModuleController>
      <hashTree/>
      <ModuleController guiclass="ModuleControllerGui" testclass="ModuleController" testname="Send SAMLResponse to IDP" enabled="true">
        <collectionProp name="ModuleController.node_path">
          <stringProp name="-1227702913">WorkBench</stringProp>
          <stringProp name="-1408792136">Shibboleth SSO Tests</stringProp>
          <stringProp name="1723882378">Send SAMLResponse to IDP</stringProp>
        </collectionProp>
      </ModuleController>
      <hashTree/>
    </hashTree>
    <ResultCollector guiclass="ViewResultsFullVisualizer" testclass="ResultCollector" testname="View Results Tree" enabled="true">
      <boolProp name="ResultCollector.error_logging">false</boolProp>
      <objProp>
        <name>saveConfig</name>
        <value class="SampleSaveConfiguration">
          <time>true</time>
          <latency>true</latency>
          <timestamp>true</timestamp>
          <success>true</success>
          <label>true</label>
          <code>true</code>
          <message>true</message>
          <threadName>true</threadName>
          <dataType>true</dataType>
          <encoding>false</encoding>
          <assertions>true</assertions>
          <subresults>true</subresults>
          <responseData>false</responseData>
          <samplerData>false</samplerData>
          <xml>false</xml>
          <fieldNames>false</fieldNames>
          <responseHeaders>false</responseHeaders>
          <requestHeaders>false</requestHeaders>
          <responseDataOnError>false</responseDataOnError>
          <saveAssertionResultsFailureMessage>false</saveAssertionResultsFailureMessage>
          <assertionsResultsToSave>0</assertionsResultsToSave>
          <bytes>true</bytes>
          <threadCounts>true</threadCounts>
        </value>
      </objProp>
      <stringProp name="filename"></stringProp>
    </ResultCollector>
    <hashTree/>
    <ResultCollector guiclass="StatVisualizer" testclass="ResultCollector" testname="Aggregate Report" enabled="true">
      <boolProp name="ResultCollector.error_logging">false</boolProp>
      <objProp>
        <name>saveConfig</name>
        <value class="SampleSaveConfiguration">
          <time>true</time>
          <latency>true</latency>
          <timestamp>true</timestamp>
          <success>true</success>
          <label>true</label>
          <code>true</code>
          <message>false</message>
          <threadName>false</threadName>
          <dataType>false</dataType>
          <encoding>false</encoding>
          <assertions>false</assertions>
          <subresults>false</subresults>
          <responseData>false</responseData>
          <samplerData>false</samplerData>
          <xml>false</xml>
          <fieldNames>false</fieldNames>
          <responseHeaders>false</responseHeaders>
          <requestHeaders>false</requestHeaders>
          <responseDataOnError>false</responseDataOnError>
```

```
            <saveAssertionResultsFailureMessage>false</saveAssertionResultsFailureMessage>
            <assertionsResultsToSave>0</assertionsResultsToSave>
            <threadCounts>true</threadCounts>
          </value>
        </objProp>
        <stringProp name="filename"></stringProp>
        <boolProp name="useGroupName">true</boolProp>
      </ResultCollector>
      <hashTree/>
      <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Shibboleth SSO No solicitado RedULA" enabled="true">
        <stringProp name="ThreadGroup.on_sample_error">startnextloop</stringProp>
        <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel" testclass="LoopController"
            testname="Loop Controller" enabled="true">
          <boolProp name="LoopController.continue_forever">false</boolProp>
          <intProp name="LoopController.loops">-1</intProp>
        </elementProp>
        <stringProp name="ThreadGroup.num_threads">${ThreadCount}</stringProp>
        <stringProp name="ThreadGroup.ramp_time">${RampUpPeriod}</stringProp>
        <longProp name="ThreadGroup.start_time">1411047603000</longProp>
        <longProp name="ThreadGroup.end_time">1411047603000</longProp>
        <boolProp name="ThreadGroup.scheduler">true</boolProp>
        <stringProp name="ThreadGroup.duration">${Duration}</stringProp>
        <stringProp name="ThreadGroup.delay">${StartupDelay}</stringProp>
      </ThreadGroup>
      <hashTree>
        <ModuleController guiclass="ModuleControllerGui" testclass="ModuleController" testname="GET Unsolicited SSO" enabled="true">
          <collectionProp name="ModuleController.node_path">
            <stringProp name="-1227702913">WorkBench</stringProp>
            <stringProp name="-1408792136">Shibboleth SSO Tests</stringProp>
            <stringProp name="-1405117056">GET Unsolicited SSO</stringProp>
          </collectionProp>
        </ModuleController>
        <hashTree/>
        <ModuleController guiclass="ModuleControllerGui" testclass="ModuleController" testname="POST Login Credentials" enabled="true">
          <collectionProp name="ModuleController.node_path">
            <stringProp name="-1227702913">WorkBench</stringProp>
            <stringProp name="-1408792136">Shibboleth SSO Tests</stringProp>
            <stringProp name="-2018788091">POST Login Credentials</stringProp>
          </collectionProp>
        </ModuleController>
        <hashTree/>
      </hashTree>
      <TestFragmentController guiclass="TestFragmentControllerGui" testclass="TestFragmentController" testname="Get NoSilver Login Page"
          enabled="true"/>
      <hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="GET NotSilver Login Page" enabled="true">
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments"
              testname="User Defined Variables" enabled="true">
            <collectionProp name="Arguments.arguments"/>
          </elementProp>
          <stringProp name="HTTPSampler.domain"></stringProp>
          <stringProp name="HTTPSampler.port"></stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol"></stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">${ShibSP}/test</stringProp>
          <stringProp name="HTTPSampler.method">GET</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSamplerProxy>
        <hashTree/>
      </hashTree>
      <TestFragmentController guiclass="TestFragmentControllerGui" testclass="TestFragmentController" testname="POST SAMLReponse Credentials"
          enabled="true"/>
      <hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="POST Login Credentials for SAMLResponse"
            enabled="true">
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments"
              testname="User Defined Variables" enabled="true">
            <collectionProp name="Arguments.arguments">
              <elementProp name="j_username" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">${User}</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">j_username</stringProp>
                <stringProp name="Argument.desc">false</stringProp>
              </elementProp>
              <elementProp name="j_password" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">${Password}</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">j_password</stringProp>
                <stringProp name="Argument.desc">false</stringProp>
              </elementProp>
              <elementProp name="_eventId_proceed" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value"></stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">_eventId_proceed</stringProp>
                <stringProp name="Argument.desc">false</stringProp>
              </elementProp>
            </collectionProp>
```

```
          </elementProp>
          <stringProp name="HTTPSampler.domain"></stringProp>
          <stringProp name="HTTPSampler.port"></stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol"></stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">${IdPHost}/idp/Authn/UserPassword</stringProp>
          <stringProp name="HTTPSampler.method">POST</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSamplerProxy>
        <hashTree>
          <RegexExtractor guiclass="RegexExtractorGui" testclass="RegexExtractor" testname="Extract RelayState Variable" enabled="true">
            <stringProp name="RegexExtractor.useHeaders">unescaped</stringProp>
            <stringProp name="RegexExtractor.refname">RelayState</stringProp>
            <stringProp name="RegexExtractor.regex">&lt;input type=&quot;hidden&quot; name=&quot;RelayState&quot; value=&quot;(.+?)&quot;</
                stringProp>
            <stringProp name="RegexExtractor.template">$1$</stringProp>
            <stringProp name="RegexExtractor.default"></stringProp>
            <stringProp name="RegexExtractor.match_number">1</stringProp>
            <stringProp name="Sample.scope">all</stringProp>
          </RegexExtractor>
          <hashTree/>
          <RegexExtractor guiclass="RegexExtractorGui" testclass="RegexExtractor" testname="Extract SAMLResponse Variable" enabled="true">
            <stringProp name="RegexExtractor.useHeaders">unescaped</stringProp>
            <stringProp name="RegexExtractor.refname">SAMLResponse</stringProp>
            <stringProp name="RegexExtractor.regex">&lt;input type=&quot;hidden&quot; name=&quot;SAMLResponse&quot; value=&quot;(.+?)&quot;</
                stringProp>
            <stringProp name="RegexExtractor.template">$1$</stringProp>
            <stringProp name="RegexExtractor.default"></stringProp>
            <stringProp name="RegexExtractor.match_number">1</stringProp>
            <stringProp name="Sample.scope">all</stringProp>
          </RegexExtractor>
          <hashTree/>
        </hashTree>
      </hashTree>
      <TestFragmentController guiclass="TestFragmentControllerGui" testclass="TestFragmentController" testname="Send SAMLResponse to IDP"
          enabled="true"/>
      <hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="POST Login to Shib IDP" enabled="true">
          <boolProp name="HTTPSampler.postBodyRaw">true</boolProp>
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments">
            <collectionProp name="Arguments.arguments">
              <elementProp name="" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">RelayState=${__urlencode(${RelayState})}&amp;SAMLResponse=${__urlencode(${SAMLResponse})}</
                    stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
              </elementProp>
            </collectionProp>
          </elementProp>
          <stringProp name="HTTPSampler.domain"></stringProp>
          <stringProp name="HTTPSampler.port"></stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol"></stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">${ShibSP}/Shibboleth.sso/SAML2/POST</stringProp>
          <stringProp name="HTTPSampler.method">POST</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSamplerProxy>
        <hashTree>
          <ResponseAssertion guiclass="AssertionGui" testclass="ResponseAssertion" testname="Response Assertion" enabled="true">
            <collectionProp name="Asserion.test_strings">
              <stringProp name="-410284789">You have authenticated using Context</stringProp>
            </collectionProp>
            <stringProp name="Assertion.test_field">Assertion.response_data</stringProp>
            <boolProp name="Assertion.assume_success">false</boolProp>
            <intProp name="Assertion.test_type">2</intProp>
          </ResponseAssertion>
          <hashTree/>
        </hashTree>
      </hashTree>
      <TestFragmentController guiclass="TestFragmentControllerGui" testclass="TestFragmentController" testname="GET Unsolicited SSO"
          enabled="true"/>
      <hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="GET Unsolicited SSO URL" enabled="true">
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments"
              testname="User Defined Variables" enabled="true">
            <collectionProp name="Arguments.arguments"/>
          </elementProp>
          <stringProp name="HTTPSampler.domain"></stringProp>
          <stringProp name="HTTPSampler.port"></stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol"></stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">${IdPHost}/idp/Authn/UserPassword</stringProp>
          <stringProp name="HTTPSampler.method">GET</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
```

```
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSamplerProxy>
        <hashTree>
          <ResponseAssertion guiclass="AssertionGui" testclass="ResponseAssertion" testname="Assert No Application Error" enabled="true">
            <collectionProp name="Asserion.test_strings">
              <stringProp name="2116026104">Application Error</stringProp>
            </collectionProp>
            <stringProp name="Assertion.test_field">Assertion.response_data</stringProp>
            <boolProp name="Assertion.assume_success">false</boolProp>
            <intProp name="Assertion.test_type">6</intProp>
          </ResponseAssertion>
          <hashTree/>
        </hashTree>
      </hashTree>
      <TestFragmentController guiclass="TestFragmentControllerGui" testclass="TestFragmentController" testname="POST Login Credentials"
          enabled="true"/>
      <hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="POST Login Credentials" enabled="true">
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments"
              testname="User Defined Variables" enabled="true">
            <collectionProp name="Arguments.arguments">
              <elementProp name="j_username" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">${User}</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">j_username</stringProp>
              </elementProp>
              <elementProp name="j_password" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">${Password}</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">j_password</stringProp>
              </elementProp>
              <elementProp name="_eventId_proceed" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value"></stringProp>
                <stringProp name="Argument.metadata"></stringProp>
                <boolProp name="HTTPArgument.use_equals">false</boolProp>
                <stringProp name="Argument.name">_eventId_proceed</stringProp>
              </elementProp>
            </collectionProp>
          </elementProp>
          <stringProp name="HTTPSampler.domain"></stringProp>
          <stringProp name="HTTPSampler.port"></stringProp>
          <stringProp name="HTTPSampler.connect_timeout"></stringProp>
          <stringProp name="HTTPSampler.response_timeout"></stringProp>
          <stringProp name="HTTPSampler.protocol"></stringProp>
          <stringProp name="HTTPSampler.contentEncoding"></stringProp>
          <stringProp name="HTTPSampler.path">${IdPHost}/idp/Authn/UserPassword</stringProp>
          <stringProp name="HTTPSampler.method">POST</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.monitor">false</boolProp>
          <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
        </HTTPSamplerProxy>
        <hashTree>
          <ResponseAssertion guiclass="AssertionGui" testclass="ResponseAssertion" testname="Assert Has IdP Session" enabled="true">
            <collectionProp name="Asserion.test_strings">
              <stringProp name="-141605550">Set-Cookie: shib_idp_session=[0-9a-f]+;Path=/idp;HttpOnly</stringProp>
            </collectionProp>
            <stringProp name="Assertion.test_field">Assertion.response_headers</stringProp>
            <boolProp name="Assertion.assume_success">false</boolProp>
            <intProp name="Assertion.test_type">2</intProp>
          </ResponseAssertion>
          <hashTree/>
        </hashTree>
      </hashTree>
    </hashTree>
  </hashTree>
</jmeterTestPlan>
```

Bibliografía

Abelson, H., Lessig, L., Covell, P., et al. 1998, White Paper Submitted for 6.805/Law of Cyberspace: Social Protocols

Anderson D.J. 2010, Kanban, english edición, Kanban, english edición (Hole Press)

Anggorojati, B., Mahalle, P. N., Prasad, N. R., y Prasad, R. 2012, Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on, 604

Baldoni, R. 2012, Electronic Government, an International Journal, 9, 64

Brail, G., y Ramji, S. 2015, OAuth - The Big Picture, https://apigee.com/about/blog/technology/oauth-big-picture-free-ebook, [En línea; Recuperado el 09 de Diciembre 2015]

Broeder, D., Jones, B., Kelsey, D., et al. 2012, https://cds.cern.ch/record/1442597, [En línea; Recuperado el 12 de Diciembre 2015]

Bruno M. Rengifo C. 2011, Desarrollo de un servicio Web para la modeloteca del sistema nacional de simulación, Proyecto de Grado, Universidad de Los Andes, Escuela de Ingeniería de Sistemas

Chadwick, D. W. 2009, en Foundations of security analysis and design V, en Foundations of security analysis and design V (Springer), 96–120

ChilliSpot. 2015, ChilliSpot - Open Source Captive Portal, http://www.chillispot.org, [En línea; Recuperado el 22 de Septiembre 2015]

CoovaChilli. 2015, CoovaChilli, an open source captive portal access controller, http://coova.github.io/CoovaChilli/, [En línea; Recuperado el 22 de Septiembre 2015]

Dovecot. 2016, Dovecot Secure IMAP server, http://www.dovecot.org, [En línea; Recuperado el 28 de Enero 2016]

EDUCAUSE. 2009, Seven Things You Should Know About Federated Identity Management, doi:10.1007/s12394-009-0036-0

eduGAIN. 2015, About eduGAIN, http://services.geant.net/edugain/About_eduGAIN/Pages/Home.aspx, [En línea; Recuperado el 20 de Mayo 2015]

EduRoam. 2016, EduRoam – World Wide Education Roaming for Research and Education, https://www.eduroam.org/, [En línea; Recuperado el 30 de Julio 2016]

Fragoso-Rodriguez, U., Laurent-Maknavicius, M., y Incera-Dieguez, J. 2006, Proceedings of the 1st Mexican Conference on Informatics Security 2006 (MCIS'2006), 1

Gustavo J. Marcano V. 2015, Desarrollo de un servicio Web para el Simulador de Eventos Discretos GALATEA, Proyecto de Grado, Universidad de Los Andes, Escuela de Ingeniería de Sistemas

Hughes, J., y Maler, E. 2005, OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08, 29

Identity, P. 2016, Security Assertion Markup Language (SAML) Whitepaper, https://www.pingidentity.com/en/resources/white-papers/saml-101.html, [En línea; Recuperado el 12 de Julio 2016]

JMeter, A. 2016, Apache JMeter, http://jmeter.apache.org, [En línea; Recuperado el 12 de Julio 2016]

Jøsang, A., Zomai, M. A., y Suriadi, S. 2007, Conferences in Research and Practice in Information Technology Series, 68, 143

Kusnetzky, D. 2011, Virtualization: A Manager's Guide, Virtualization: A Manager's Guide, 74

Leandro, M. a. P., Nascimento, T. J., dos Santos, D. R., y Westphall, C. B. C. M. 2012, The Eleventh International Conference on Networks, 88, http://www.thinkmind.org/index.php?view=article&articleid=icn_2012_5_10_10065

Loutfi, I., y Josang, A. 2015, IFIP Advances in Information and Communication Technology, 454, 165

Madsen, P., Maler, E., Microsystems, S., et al. 2005, SAML V2.0 Executive Overview, Reporte Técnico April

Margaret Rouse. 2016, ¿Qué es Virtualización?, http://searchdatacenter.techtarget.com/es/definicion/Virtualizacion, [En línea; Recuperado el 20 de Mayo 2015]

McLaughlin, M., Briscoe, G., y Malone, P. 2010, Digital Identity in The Absence of Authorities: A New Socio-Technical Approach

Michelle McNickle. 2016, Diez definiciones esenciales de virtualización de redes, http://searchdatacenter.techtarget.com/es/consejo/Diez-definiciones-esenciales-de-virtualizacion-de-redes, [En línea; Recuperado el 20 de Mayo 2015]

Mora, E., Araujo, A., Bravo, V., et al. 2014, SEGURIDAD INFORMATICA LA IDENTIDAD DIGITAL Fundamentos y Aportes, 1er edición, SEGURIDAD INFORMATICA LA IDENTIDAD DIGITAL Fundamentos y Aportes, 1er edición (Fundación CENDITEL), 189

NETGEAR Support. 2016, What is the captive portal and how does it work with my managed switch?| Answer | NETGEAR Support, http://kb.netgear.com/22006/What-is-the-captive-portal-and-how-does-it-work-with-my-managed-switch?cid=wmt_netgear_organic, [En línea; Recuperado el 14 de Noviembre 2015]

OASIS Security Services. 2015, OASIS Security Services (SAML) TC, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, [En línea; Recuperado el 22 de Abril 2015]

OpenAM Reference. 2015, OpenAM Reference, https://forgerock.org/openam/doc/bootstrap/reference/index.html, [En línea; Recuperado Mayo-2015]

OpenID Connect - Wikipedia. 2015, Wikipedia, la enciclopedia libre (2015, Diciembre). OpenID Connect., https://en.wikipedia.org/wiki/OpenID_Connect, [En línea; Recuperado el 12 de Diciembre 2015]

Radha, V., y Reddy, D. H. 2012, Procedia Technology, 4, 134, http://dx.doi.org/10.1016/j.protcy.2012.05.019

Real Academia Española. 2016, Diccionario de la lengua española - Edición del Tricentenario, http://dle.rae.es/?id=KtmKMfe, [En línea; Recuperado el 10 de Mayo de 2016]

Rede Nacional de Ensino e Pesquisa. 2015, The Federated Academic Community (CAFe), https://memoria.rnp.br/en/services/cafe.html, [En línea; Recuperado el 18 de Diciembre 2015]

RedIRIS. 2016, RedIRIS-Identidad Digital, http://www.rediris.es/servicios/identidad/, [En línea; Recuperado el 26 de Mayo 2016]

Scott W. Ambler, y Matthew Holitza. 2012, Agile for dummies, ibm limite edición, Agile for dummies, ibm limite edición (John Wiley & Sons,), 74

Scudder, J., y Josang, A. 2010, IFIP Advances in Information and Communication Technology, 343 AICT, 85

Shibboleth Consortium. 2015, What's Shibboleth?, http://shibboleth.net/about/, [En línea; Recuperado el 20 de Mayo 2015]

Shibboleth Wiki. 2016, Confluence Mobile - Shibboleth Wiki, https://wiki.shibboleth.net/confluence/plugins/servlet/mobile#content/view/17072886, [En línea; Recuperado el 09 de Septiembre 2016]

SquirrelMail. 2015, SquirrelMail - Webmail for Nuts, https://squirrelmail.org/, [En línea; Recuperado el 13 de Septiembre 2015]

Talamo, M., Barchiesi, M. L., Merella, D., y Schunck, C. H. 2014, en ITU Kaleidoscope Academic Conference: Living in a converged world-Impossible without standards?, Proceedings of the 2014, en ITU Kaleidoscope Academic Conference: Living in a converged world-Impossible without standards?, Proceedings of the 2014, IEEE, 15–21

Vicki Stanfield, Roderick W. Smith. 2006, Linux System Administration: Craig Hunt Linux Library, english edition edición, Linux System Administration: Craig Hunt Linux Library, english edition edición (Sybex Inc)

VMware. 2016, Virtualización de VMware, http://www.vmware.com/latam/solutions/virtualization.html, [En línea; Recuperado el 20 de Mayo 2015]

WS-Federation. 2015, Web Services Federation Language (WS-Federation) Version 1.2., http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html, [En línea; Recuperado el 20 de Mayo 2015]

Xen Project. 2016, Xen Project Software Overview, https://www.xenproject.org/, [En línea; Recuperado el 20 de Mayo 2015]

Zeroshell. 2015, Router/Bridge Firewall Linux, http://www.zeroshell.net/es/, [En línea; Recuperado el 29 de Octubre 2015]