Privacy-preserving Identity Federations in the Cloud - A Proof of Concept

Daniel Ricardo dos Santos*, Tiago Jaime Nascimento, Carla Merkle Westphall, Marcos Aurélio Pedroso Leandro and Carlos Becker Westphall

Networks and Management Laboratory,
Department of Informatics and Statistics,
Federal University of Santa Catarina
Florianópolis, SC, Brazil
E-mail: {danielrs,tiagojn,carlamw,marcosleandro,westphal}@inf.ufsc.br
*Corresponding author

Abstract: Because of the growth in the use of cloud computing and the migration of services to this paradigm, it becomes necessary to investigate security issues that might compromise its use. Identity and Access Management is among these issues and is related to the management of users and access to their data. Federated Identity Management is widely adopted in the cloud to provide useful features to identity management systems, but maintaining user privacy in those systems is still a challenge. This paper describes the implementation of a privacy-preserving identity federation in the cloud. Our motivation was to develop a proof of concept, in order to elucidate the identity federation setup of Shibboleth and the handling of private attributes performed by uApprove in a cloud computing environment. The paper shows a description of the deployment of the identity and service providers, their integration and a detailed analysis of the scenario.

Keywords: Cloud computing; Identity; Federation; Privacy; Shibboleth.

Reference to this paper should be made as follows: dos Santos, D.R., Nascimento, T.J., Westphall, C.M., Leando, M.A.P., Westphall, C.B. (xxxx) 'Privacy-preserving Identity Federations in the Cloud - A Proof of Concept', *Int. J. Security and Networks*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Daniel Ricardo dos Santos holds a B.Sc. and a M.Sc. in Computer Science from the Federal University of Santa Catarina and is currently pursuing his PhD. He is a member of the Networks and Management Laboratory and has experience in software development and computer security.

Tiago Jaime Nascimento is an undergraduate Information Systems student at the Federal University of Santa Catarina. He is currently working at the university's IT department and has experience in development and security.

Carla Merkle Westphall is a professor in the Department of Informatics and Statistics at the Federal University of Santa Catarina. She is working in security since 1996. Her research interests include information security, distributed security, identity management and cloud security. Westphall received her PhD in electrical engineering (information systems security) from the Federal University of Santa Catarina. She is a member of the Networks and Management Laboratory which has many master and doctoral students developing security research.

Marcos Aurélio Pedroso Leandro holds a BSc. in Informatics from Unicentro and an MSc. in Computer Science from the Federal University of Santa Catarina. He is a member of the Networks and Management Laboratory and has experience in identity management.

Carlos Becker Westphall is a full professor in the Department of Informatics and Statistics at the Federal University of Santa Catarina, where he is the leader of the Networks and Management Laboratory. His research interests include network management, security and cloud computing. Westphall received his PhD in computer science from the Paul Sabatier University, France. In 2011, he received the Awarded IARIA Fellow. He is member of IARIA Liaison Board Chair and IFIP TC6 Working Group 6.6 (Management of Networks and Distributed Systems). He is also an editorial board member of Computer Networks Journal (Elsevier).

1 Introduction

Cloud computing is the delivery of shared computational resources, such as storage, processing power or even software, to users over the Internet. Security is fundamental to ensure the success of the cloud (Grobauer et al., 2012; Lee et al., 2009; Takabi et al., 2010) and privacy protection is especially important, since sensitive data may now be in the custody of a third party (Pearson, 2009).

Identity Management (IdM) is increasingly important because of the growth in the number of services that need to employ user authentication and access control (Bertino and Takahashi, 2011), such as many services that run in the cloud and need to establish the identity of their users, as well as protecting the privacy of these users.

The goal of this work is to show an initial deployment of a cloud-based identity federation. The deployed federation has been used to explore research challenges regarding authentication, authorization and security as a whole in cloud computing.

In our proof of concept, identities are managed by Shibboleth (Internet2, 2013), with the aid of the uApprove privacy plugin (uApprove, 2013). This structure composes an identity provider that runs in a Virtual Machine (VM) on Amazon EC2 (Amazon EC2, 2013). A service provider is also deployed in a VM and a circle of trust involving both entities is established, creating an identity federation in the cloud.

This work does not present novel components with respect to identity management or privacy in the cloud, but the proof of concept helps elucidate a cloud-based federation implementation and perceive its benefits and drawbacks.

The remaining of the paper is organized as follows: section 2 presents the related works; section 3 discusses identity, cloud and privacy; in section 4 the proposal is presented; section 5 shows the development; the results are presented in section 6 and section 7 has some conclusions and the description of future works.

2 Related Work

Privacy is investigated in several works (Chadwick and Fatema, 2012; Orawiwattanakul et al., 2010; Ranchal et al., 2010; Sánchez et al. 2012; Takabi et al. 2010; Watanabe et al., 2012), among them Sánchez et al. (2012) propose a dynamic, privacy-enhanced Federated Identity Management (FIM) architecture for cooperation, on-demand resources provisioning and delegation in cloud computing scenarios.

An enhanced privacy module is defined through a SAMLv2 extension and a privacy engine is incorporated in every client to carry out an appropriate management of users' identifiers according to their preferences and context, as well as to monitor how users' data are being accessed by identity or service providers without compromising identities.

The main tasks of the privacy engine are related to auditing and monitoring. Clients have "Enhanced Client Profile" components to control privacy aspects: clients can configure their preferences about the handling of personal data and specify options for the use and release of sensitive information; they can check the accuracy of personal information and visualize how data are being used; and they can also choose between multiple identities when interacting with cloud services. Although privacy aspects of this proposal are well discussed, the privacy engine is not described in detail.

Chadwick and Fatema (2012) describe an authorization infrastructure that a cloud provider can run as a service for its users, allowing them to set their own privacy policies. The infrastructure ensures that privacy policies are stuck to the users' data, so that access will always be controlled by the users, even if the data are transferred between cloud providers. The authorization infrastructure is implemented in Java, and is being used as a part of the TAS3 Project. Authorization is based on web services, offering an interface to cloud application developers, allowing them to call the service, passing the user's privacy policy and obtaining authorization decisions. This authorization service accepts policy preferences from users, which are converted into policy rules and stuck to the personal data. Receiving services will only store the data if they support a Policy Decision Point (PDP) that can evaluate the sticky policy language accompanying the data.

Watanabe et al. (2012) extended Opengate, a cloud-based mobile LAN user authentication system. Shibboleth functions were implemented into Opengate and this implementation was called "shibbolized Opengate". The advantage of this style of service is that no organization needs to operate single sign-on Opengate by itself, reducing the cost for installation and operation.

Orawiwattanakul et al. (2010) describe an uApprove extension called uApprove.jp, which allows individual users of a Shibboleth federation to choose which attributes will be released by identity providers to service providers.

Ranchal et al. (2010) propose an approach for identity management which is independent of a Trusted Third Party (TTP) and has the ability to use identity data on untrusted hosts. The approach is based on the use of predicates over encrypted data and multi-party computing for negotiating the use of a cloud service. It uses an active bundle, which is a middleware agent that includes Personally Identifiable Information (PII) data, privacy policies, a VM that enforces the policies and a set of protection mechanisms. An active bundle interacts on behalf of a user to authenticate to the cloud service using the user's privacy policies. A prototype using Java agents was developed.

In Table 1 there is an overview of some important characteristics addressed by the related works, as well as a comparison with our proposal. The Security Assertion

Markup Language (SAML) is a language used to manage identity federation environments. The papers listed in Table 1 have some implementation descriptions. In Watanabe et al. (2012) and in our proposal Shibboleth was used in a cloud environment, but our proposal considers privacy aspects with uApprove.

Sánchez et al. (2012) do not use Shibboleth federations, but a modified Authentic (Lasso) IdP, the ZXID service provider and a proprietary implementation of a client ECP. Chadwick and Fatema (2012) use a privacy policy stuck to the client data that is useful only if the service provider supports a correct PDP that can evaluate it and they do not specify how users could choose and enter their privacy policies.

3 Identity, Cloud and Privacy

3.1 Identity Management

Digital identities are collections of data that represent attributes, preferences and traits of an entity (Windley, 2005). Attributes are characteristics associated with the entity, such as access histories; preferences represent the wishes of the entity; and traits are permanent characteristics, such as birth date.

A digital identity may represent a person or any organization and is related to an identifier such as a username, an ID number or an IP address. Digital identities are used in authentication and authorization in applications such as e-commerce, social networks, e-health or online banking.

An IdM service can be defined as the "process that creates, manages and uses user IDs, and the infrastructure that supports this process" (Lee et al., 2009). The following roles exist in an IdM system (Bertino and Takahashi, 2011):

User - Owns an identity and uses the services of both the identity provider and the service provider.

Identity Provider (IdP) - Provides the identity management services necessary for a user to use the service provider.

Service Provider (SP) - Provides the services that a user effectively wishes to use. The SP can delegate the authentication and authorization of its users to an IdP.

In Federated Identity Management (FIM) (Chadwick, 2009; Olden, 2011), a federation is the association, trust and commitment between IdPs and SPs to allow the sharing of users' authentication and authorization information among themselves. The identity of a user is said to be "federated" among a set of providers when they have an agreement of a set of identifiers and attributes used to refer to the user. Federated identities allow entities to use credentials among different organizations, securing access to resources in different domains or SPs in the same federation, without the need for authenticating several times. This is known as Single Sign-on (SSO) (Harris, 2008).

Some examples of FIM technologies are the SAML standard and the Shibboleth system (Bertino and Takahashi, 2011; Chadwick, 2009).

3.2 Cloud Computing Security

Marston et al. (2011) define cloud computing as "an information technology service model where computing services (both hardware and software) are delivered on-demand to customers over a network in a self-service fashion, independent of device and location."

Cloud Service Providers (CSP) use technologies such as web applications, web services, virtualization and cryptography to enable their services. Implementations of these technologies present a number of known vulnerabilities and the use of cloud computing makes vulnerabilities more significant, also adding new ones (Buecker et al., 2009; Grobauer et al., 2011). It might be difficult, however, to distinguish specific vulnerabilities caused by cloud computing (Maggi and Zanero, 2011).

Some potential vulnerabilities of the technologies used in cloud computing are: session hijacking, in which an attacker steals a valid user session to get unauthorized access; escape from virtualization, when the attacker is able to leave the virtualized environment and access a physical machine; and insecure or obsolete cryptography.

Some vulnerabilities can be considered linked to the characteristics of cloud computing. Due to the characteristics of self-provisioning, cloud management is usually done through a web interface, making it easier for an attacker to get unauthorized access than if the management interface were only available to administrators. Because of resource relocation, it is possible that an attacker obtains access to storage and memory resources previously used by other users, which may still contain sensitive data. Malicious code injection vulnerabilities, such as SQL injection and Cross-Site Scripting, are also predominant in cloud services (Grobauer et al., 2011; Maggi and Zanero, 2011).

The security responsibilities of providers and clients vary according to the cloud service models. In the Software as a Service (SaaS) model, security controls are negotiated in Service Level Agreements (SLAs), involving issues such information privacy and compliance to regulations. In the Infrastructure as a Service (IaaS) model, the CSPs are responsible for protecting the basic structure and the abstraction layers. Platform as a Service (PaaS) balances the responsibilities, since protecting the platform is a provider task, but protecting the applications is the responsibility of the clients.

3.3 Cloud Identity Management

There must be mechanisms to securely manage the identities and accesses of clients to SPs in the cloud.

Table 1 Comparison between our proposal and the related works

Work	*SAML*	*Cloud-based*	*Implementation*	*Identity Fed.*	*Privacy*
Sánchez et al. (2012)	Yes, with extensions	Yes	Yes	Yes	Yes
Chadwick and Fatema (2012)	Yes	Yes	Yes	No	Yes
Watanabe et al. (2012)	Yes (Shibboleth)	Yes	Yes	Yes	No
Orawiwattanakul et al. (2010)	Yes (Shibboleth)	No	Yes	Yes	Yes
Ranchal et al. (2010)	No	Yes	Yes	No	Yes
This paper	Yes (Shibboleth)	Yes	Yes	Yes	Yes

Identity and Access Management (IAM) plays an important role in the control and charge for access of users to shared resources in the cloud. These resources are managed by different entities and often geographically distributed (Bertino and Takahashi, 2011). The main IAM functions in cloud computing are: creation and deletion of identities, authentication, authorization and federation establishment (Grobauer et al., 2011). IAM must still evolve to better attend to the needs of cloud computing (Olden, 2011).

There are three possible configurations for an organization to implement identity management in the cloud (Bertino and Takahashi, 2011): in the organization itself, as an outsourced service or in the CSP. When implemented in the organization, it is the organization that issues and manages the identity of its users.

The outsourced service that delivers identity management is called Identity as a Service (IDaaS), with identities issued and managed by IdPs. An IDaaS service can keep the full data set of the staff of an organization or only pseudonyms, which the organization uses to map to real identities.

Another possible configuration is for the service provider to implement identity management functions independently, which requires client organizations to have different identity sets for each provider.

Regardless of the configuration, there are challenges in authentication related to the management of credentials, strong and delegated authentication and trust management among all cloud services. SaaS and PaaS offerings typically provide the option of an authentication service embedded in their own applications and platforms, or they delegate authentication to the clients themselves. In this process there are also standard protocols such as SAML, which can be used to allow the delegation of authentication to the organization that consumes the services.

Access control policies vary, since users may act individually or as members of an organization (Grobauer et al., 2011). The selection and revision of the adequacy of access control solutions involves aspects such as the definition of the appropriate access control model to the given kind of service, the evaluation of support to necessary privacy policies, the selection of a format for the specification of user information and policies and the way that auditing information is registered.

Cloud partners that use FIM depend on each other to authenticate their respective users and to attest the identities and access privileges of these users (Lynch, 2011). The use of federations allows organizations to use whatever IdP they choose and there must be mechanisms to manage the identity life cycle and authentication methods that protect confidentiality and integrity and provide non-repudiation.

Currently, organizations may choose among SAML, Liberty Alliance or WS-Federation to federate their users. Federation is achieved through the use of these protocols, which standardize the communications among applications in different domains, and also through the agreement on which identifiers and user attributes are required. Thus, partners are not forced to adopt the same technologies for directory services and internal security.

However, interoperability is still an obstacle, because there are not only different protocols, but also different versions of the same protocol, which may be incompatible. To solve those differences, organizations use a federation gateway which provides FIM protocol translation services.

Organizations must ensure that the chosen CSPs support prominent standards and that federation gateways can be used to complete their federation implementations, translating tokens of different technologies and ensuring interoperability.

Another issue concerning identity management in the cloud is identity governance, specifically audit and monitoring. Those two processes must be facilitated by the identity management system and integrated into cloud security monitoring.

3.4 Privacy

Privacy is the ability of individuals to protect information about themselves (Mather et al., 2009) and a privacy policy expresses how an entity collects, uses, manages and discloses information about its users.

The Fair Information Practice Principles (FIPP) is a set of rules that regulates the use of private information in the United States and inspired similar rules in other countries (Federal Trade Commission, 1998).

The FIPP defines five basic principles: awareness means that users must be warned and understand how their information will be disclosed; choice means that users must choose how their information will be used; participation allows users to access and change their information; integrity is useful to ensure that users' data are correct and enforcement ensures that the above principles are respected.

Privacy is a critical aspect of security in cloud computing (Bertino and Takahashi, 2011; Goth, 2011; Mather et al., 2009; Pearson, 2009; Sánchez et al., 2012) and according to Mather et al. (2009) and Takabi et al. (2010) there are some aspects that may be questioned when researching privacy in cloud computing.

Users must have the right to know which of their data are being kept in the cloud and be able to request the removal of these data; they must also have guarantees that their data are being stored and transferred securely.

Service providers, in their turn, must follow laws, rules and regulations when dealing with private information. They must: know where and how private data are stored and how they can be transmitted; keep policies about data retention in the cloud; ensure that no copies of stored data remain after their destruction; ensure that they are following privacy requirements; keep data access logs; and, in the case of a privacy breach or information leakage, know who is responsible and how to control the case.

Identity management systems could help in providing privacy. Considering the use of IdM systems in the cloud, privacy policies are necessary in both IdPs and SPs. The control of the correct development and use of these policies is a big challenge in cloud computing as shown from previous experiences (Chadwick and Fatema, 2012; Sánchez et al., 2012; Jensen, 2012).

In Chadwick and Fatema (2012), a policy language has to be used in the communication between client and server. Sánchez et al. (2012) are concerned with the establishment of trust between the users and servers in order to interact considering privacy options chosen by the user maintaining the focus on auditing and monitoring. Also in cloud environments, users in some cases have very limited control over their credentials and do not have ways to control their identities (Jensen, 2012).

4 Federation Implementation

This section describes the deployment of the cloud-based privacy-preserving identity federation. The federation is deployed in a public cloud and is composed of an identity provider that ensures the privacy of the attributes of authenticated users and a service provider that consumes this information (Figure 1).

In this scenario, users initially access a SP and, when they wish to be authenticated in this provider, they are redirected to their IdP. Trust between identity and service providers is achieved by the establishment of the federation and requires a previous agreement between the parties. Both providers are deployed in the cloud, but this is transparent to the users. The identity provider asks for user authentication and accesses user attributes in its database. When the users are authenticated and before being redirected to the service provider again, their data go through a privacy plugin, when they become aware of and must consent to the disclosure of their attributes.

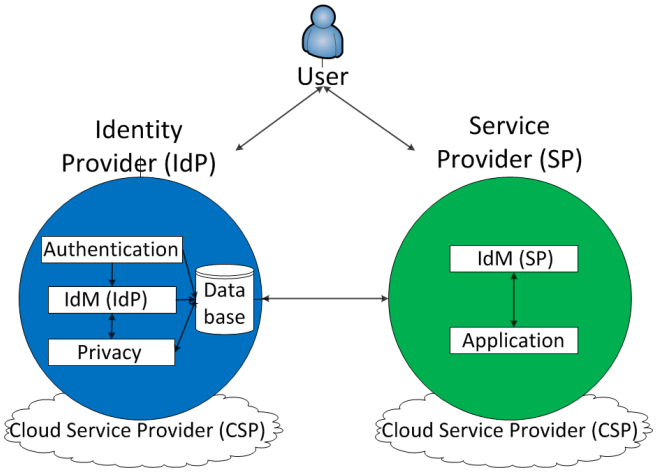

Figure 1 Overview of the proposal

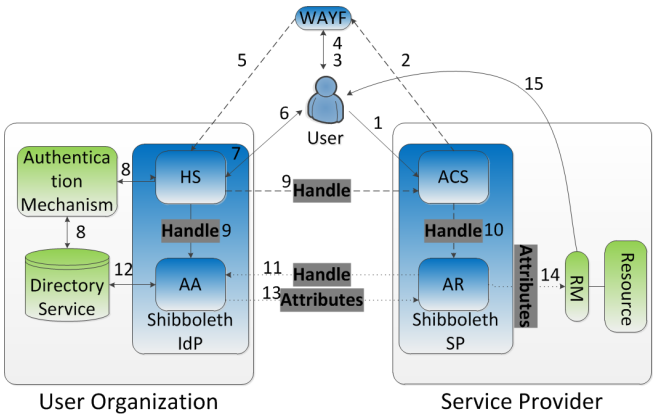

Figure 2 Shibboleth's execution flow

4.1 Amazon EC2

EC2 was the CSP used in the deployment. EC2 provides an IaaS in which it is possible to deploy VMs from predefined system images and to configure machine characteristics such as processing power, memory and storage capacity.

In EC2, the user can attach static IP addresses to the instantiated machines and configure the opening of access ports. Data persistence is achieved through the use of Elastic Block Storage (EBS) volumes that act as the hard disks for the machines.

4.2 Shibboleth

Among the several IdM systems available, Shibboleth was chosen because of its popularity, besides being free and open source software. Shibboleth is comprised of two main components: the IdP and the SP, which are separated but communicate to provide access to services. Shibboleth's execution flow is represented in Figure 2.

In step 1, users navigate to the service provider to access a protected resource. In steps 2 and 3 Shibboleth

redirects users to the Where are you from? (WAYF) service, where they must inform which is their identity provider. In step 4 users inform their IdP and in step 5 they are redirected to the site, which is the Handle Service (HS) component in their IdP. In steps 6 and 7 users inform their data and in step 8 the HS component verifies the validity of their attributes. The HS creates a handle to identify users and registers them in the Attribute Authority (AA). In step 9 this handle confirms the user authentication. The handle is verified by the Assertion Consumer Service (ACS) and transferred to the Attribute Requester (AR) and in step 10 a session is created. In step 11 the AR uses the handle to request user attributes to the IdP. In step 12 the IdP verifies if it can release the attributes and in step 13 the AA answers with the values of the attributes. In step 14 the SP receives the attributes and forwards them to the Resource Manager (RM), which in step 15 loads the resource (Bonetti et al., 2011).

4.3 uApprove

The most important privacy principles in the FIPP are user awareness of data collection and storage, and the possibility of choice about the disclosure of these data. uApprove (uApprove, 2013) is a Shibboleth privacy plugin that implements both of these principles. It is divided in three main components: the IdP plugin is a Shibboleth filter that tests if the tool must obtain user consent for the release of their attributes; the Viewer presents to the user a web page with the terms of use that must be accepted when using the IdP; and the Reset Approvals module allows the user to reset previous consents. Figure 3 shows the execution flow of the IdP plugin to decide if the Viewer should be invoked.

First, the plugin verifies if the LoginContext, a Java object created upon the request of an authentication, is correct. If the LoginContext is correct, it is verified if the Shibboleth Authentication Request (AuthN), a message sent by the SP to the IdP to initiate a session, was provided. If any of these verifications is negative the execution is canceled and the authentication process is terminated. If the first two verifications are positive, the plugin verifies if it is being executed in observation mode, in which case it only registers the attributes that will be disclosed, without user interaction. If the plugin is in this mode, the flow follows to the Shibboleth IdP. Otherwise, the plugin continues its flow, verifying if the SP is in the black list, a list of SPs in which uApprove automatically assumes user consent. If the SP is in the list, the flow follows to the Shibboleth IdP, otherwise the plugin verifies if the Principal (the unique identifier of the user) is known (which means the plugin was already used). If the principal is unknown or is known but has reset his consents, the Viewer will be invoked, otherwise the plugin follows. In the sequence, if the terms of use were altered since the last access the flow follows to the Viewer, otherwise the plugin continues. After that, the user's attribute disclosure global consent is verified. If the global consent was given, the flow goes to the IdP, if not it goes to the next verification. It is then verified if the user is accessing the SP for the first time, in which case the Viewer is invoked, otherwise the last verification is done. If the attributes requested by the SP were changed the Viewer is invoked, otherwise the flow goes to the IdP.

In every case where the flow goes to Shibboleth IdP, the execution of the plugin is transparent to the user. In every case where the Viewer is invoked, users must interact with it and give their consent.

4.4 DokuWiki

DokuWiki is an open-source software for the creation and management of wikis (DokuWiki, 2013) and was chosen to demonstrate authentication and authorization with Shibboleth in a real application. It is developed in PHP and was created to be simple to install and maintain, but still offering many resources for the editing of wiki pages.

The main reason why this software was chosen to be used in the demonstration is because it is ready to authenticate users with Shibboleth, through a backend developed by Novakov (2013). This extension triggers the Shibboleth daemon during authentication and maps the attributes received by the IdP to user account data that can be used by the application to authorize its users.

An official list of applications ready to work with Shibboleth, as well as information about adapting applications to work this way are available (Chamberlin, 2013; Bradley, 2013; Klingenstein, 2013).

5 Case study

The development of the case study was divided in three parts: the installation, configuration and tests of the identity provider; the installation, configuration and tests of the service provider and its application and, finally, the configuration and tests of the federation itself. The biggest challenges during the research were because of missing documentation of some of the software components, which makes their integration harder.

5.1 Identity Provider

Using Amazon EC2, a Windows Server 2008 VM was instantiated and a static IP address and an EBS volume were attached to it. The ports that were opened in the firewall were: 3306 for MySQL access, 3389 for remote access, 8009 for Shibboleth and 8080 for the Tomcat application server.

The Apache web server was installed and configured to accept non-SSL connections on port 80 and SSL connections on ports 443 and 8443. Then the Apache Tomcat application server was installed to serve the authentication, identity management and privacy plugin

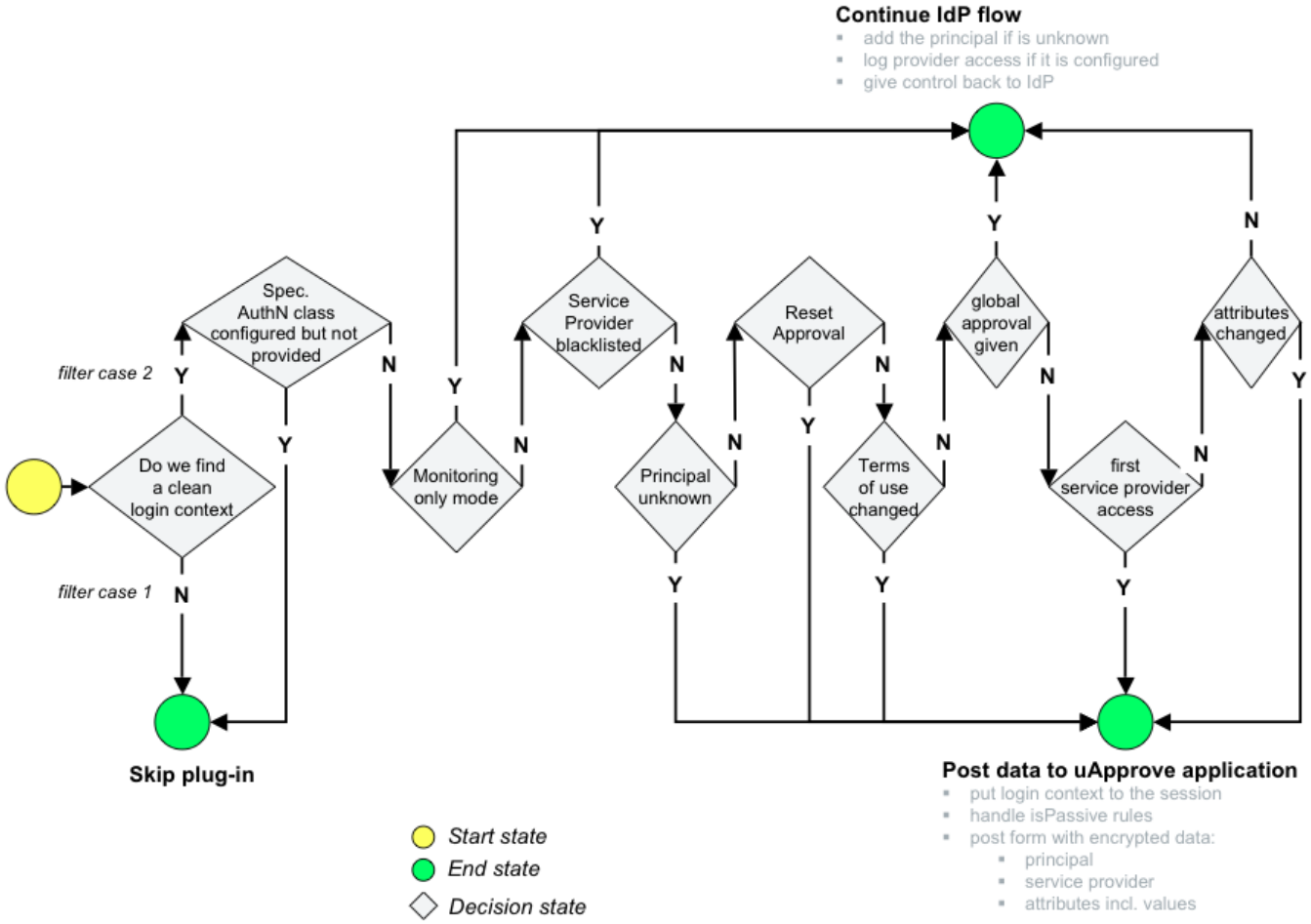

Figure 3 uApprove's execution flow (From uApprove (2013))

applications and an Apache proxy was configured to forward the requests for these applications to Tomcat.

The JASIG CAS (2013) authentication mechanism was installed. It authenticates users by their username and password and then forwards the authenticated users to Shibboleth. CAS was configured to search for users in a LDAP directory.

The TestShib federation (TestShib, 2013) was chosen to test the installation and configuration of the Shibboleth identity provider. In order to use it we had to register our IdP, informing its URL and the generated certificate, configuring also Shibboleth to use the federation metadata.

To configure the release of user attributes we used the brEduPerson schema, which is an extension of eduPerson for Brazilian academic federations (Robertson, 2009).

The next step was the installation of uApprove. The plugin uses MySQL to store information about the consent of users and the release of their attributes.

A file containing an example of Terms of Use was then generated and, with the whole configuration ready, a filter to activate the use of IdP plugin with Shibboleth was activated. The creation, editing and deletion of identities were not our main concern, so those functions were performed directly in the LDAP directory.

5.2 Service Provider

To install the Shibboleth SP, another Ubuntu Server VM was instantiated and the "etc/hostnames" and "/etc/hosts" files were configured with the host address and the IP of the instance obtained by EC2. The generation of the certificates for the Apache web server and the Shibboleth SP was done using openssl.

The Apache web server and its PHP modules along with Shibboleth were installed by the packages libapache2-mod-php5 and libapache2-mod-shib2. The ports opened for Apache connections were 80 and 443.

The standard Apache Virtual Host was defined, for connections in port 80, to allow initial unauthenticated access to the root address of the application, and the protected Virtual Host, using port 443, which ensures that Apache protects the authentication address with Shibboleth, accessed by the user when he chooses to log in.

After having Apache and Shibboleth installed and the main modules activated, some specific configurations were defined by editing Shibboleth files. The file attribute-map.xml maps the attributes to be used - the service provider only considers attributes that are correctly declared in this file, ignoring any other it may receive from the IdP. The file metadata-sp.xml defines

configuration details used by the SP that any other entity within the same federation must know to establish communication.

Lastly, the file shibboleth2.xml is responsible for much of the general configurations of Shibboleth, such as the definition of the scope of system protection, the type of session used, which IdP will be used and the locations defined by URIs that the mod_shib process monitors to receive SAML assertions.

The Shibboleth module was then configured to require the authentication of a specific URL address, regardless of what was provided through it, and to associate possible released attributes to an environment variable accessible by Apache. The next issue was how to install an application and associate it with this protected address.

5.3 *DokuWiki installation and its protection through Shibboleth*

DokuWiki installation is summarized by the downloading and execution of its package available in the Ubuntu repository. To provide external access through Apache, the definition file of the protected Virtual Host was modified to point to the location where the software was installed. Inside the configuration file shibboleth2.xml, the application protection and the type of session (lazy session) were declared.

Since the chosen application is already prepared to accept Shibboleth authentication and authorization, the specific configurations necessary to define this type of access were made by modifying php configuration files. In the same way, the identity attributes requested by the SP and their roles were established. Simple authorization rules were configured by defining which usernames are admins.

Once again for test purposes, TestShib was used with the SP, following a similar procedure to the one of the IdP, described previously.

5.4 *Federation and expansion*

With the IdP and the SP installed and configured, the certificates and metadata generated by them were made available and each one was set to trust the other party. The metadata is in fact what represents the federation and, in this case, the exchange of metadata represents the creation of a federation containing an identity provider and a service provider.

The configurations had to be performed separately in each provider. First the IdP set its MetadataProvider in Shibboleth's relying-party.xml file as the URL of the SP's metadata and then new filters were created in the attribute-filter.xml file, so that the users' attributes could be released to the desired SP. In the SP's side the MetadataProvider was set to our IdP in the shibboleth2.xml file. Figure 4 shows an excerpt of the relying-party.xml file that contains the specification of the metadata provided by the SP.

```
<metadata:MetadataProvider id="URLMD"
    xsi:type=
"metadata:FileBackedHTTPMetadataProvider"
    metadataURL=
"http://ec2-50-19-108-64.compute-1.amazonaws.com
/metadata.xml"
    backingFile=
"C:\shibboleth-idp/metadata/sp.xml">
</metadata:MetadataProvider>
```

Figure 4 Excerpt of the relying-party.xml file

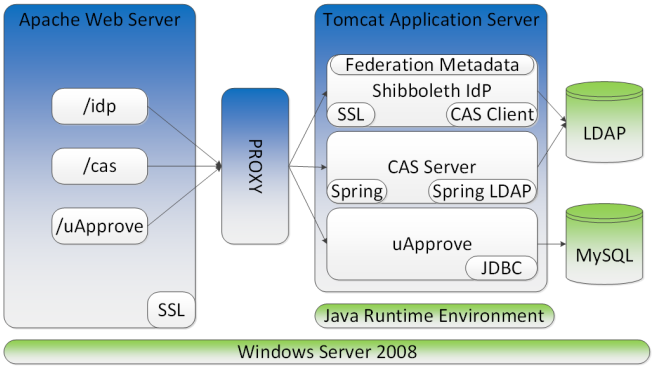

Figure 5 Detailed view of the IdP

After all the configurations, a detailed view of the identity provider, highlighting the technologies applied, can be seen in Figure 5.

We detailed the deployment of one IdP and one SP and the establishment of a circle of trust between them. To enable the expansion of this federation, it would be necessary to deploy as many providers as necessary and then expand the circle of trust. It can be done by creating federation specific metadata, loading it into the IdPs and SPs and then configuring the WAYF service (Klingenstein, 2013).

It is also possible to integrate the deployed IdPs and SPs into already established federations. In order to achieve that, it is necessary to set the desired federation as the metadata provider for both the IdPs and the SPs.

6 Results

To show the results of the deployment, we present an use case of the application. First of all, users access a service provider, running the DokuWiki service (Figure 5).

If the users wish to be authenticated in the application, they are redirected to their identity provider. In this case, the federation was established with only one IdP, so users are automatically redirected to it. In the IdP, users are presented to the authentication screen, provided by CAS (Figure 6).

The users must then enter their credentials so that the IdP can authenticate them. These data are retrieved in the LDAP directory that acts as a user database. After authentication, Shibboleth seeks in the directory which attributes must be released. In this moment the

Figure 6 Authentication by CAS Server

Figure 7 Attributes to be released to the SP

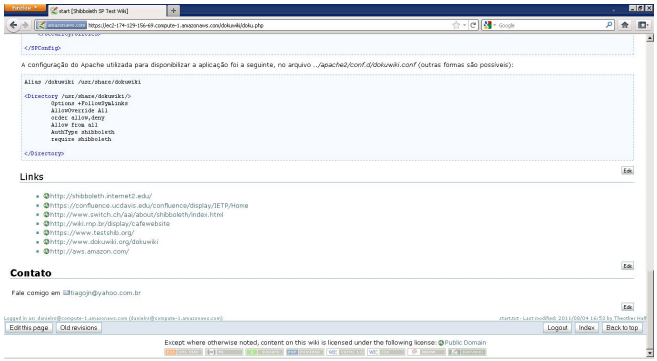

Figure 8 User authenticated in the SP

uApprove filter comes into action and shows a page containing the IdP terms of use.

If the requesting user accepts the terms of use, the plugin redirects him to a new page that shows the attributes about to be released (Figure 7). The authenticated user is again requested to accept the release of these attributes and, if he agrees, is taken to the restricted access page of the service provider (Figure 8).

The service provider is in charge of the authorization, which is based on the attributes requested to the IdP. In this case, for instance, the user name is used to define the role of this user in the system, granting him some specific permissions.

6.1 Analysis

As stated before, in order to achieve IAM in the cloud, there are three possible basic approaches (Bertino and Takahashi, 2011): placing the IdPs internally in an organization and the SPs in the cloud; placing the SPs internally in an organization and the IdPs in the cloud; or placing both the SPs and the IdPs in the cloud.

Since authentication is handled by the IdP and authorization by the SP, each model presents advantages and disadvantages. Placing the IdPs in the cloud means storing sensitive identity information in the cloud, while placing the SPs in the cloud means giving away authorization policies and rules.

We opted to place the IdP and the SP in the cloud because we intended to study the impact of the cloud in identity management. The main benefits of this approach are availability, scalability, accessibility and reduced costs, all of which are known to cloud users.

There are, nevertheless, drawbacks with this approach. Trust plays a big role in identity management. Users have to trust their identity and service providers and in the cloud, all of the entities have to rely on their CSPs to keep their data safe, which means not disclosing or tampering with their personal data and having contingency plans, among other precautions.

The choice of identity and service providers to move to the cloud must be guided by a thorough risk assessment and the amount of trust or lack thereof in their chosen cloud service provider.

Besides the chosen cloud deployment model, we can also analyse our proposed model, and the main advantages we can list are: it authenticates users respecting their privacy policies; and it provides minimal information to the SP.

The most important limitations we can list in our deployment are the fact that authorization is static, identity management is not user-centric and the attributes of each user come from a single provider, but these limitations are present because of Shibboleth.

In the cloud, Shibboleth should have other characteristics such as the dynamic federations, fine-grained authorization and privacy policies both in the IdP and the SP sides. Besides uApprove, Shibboleth could have a user-centric interface providing users a better control over their attributes release.

7 Conclusions and Future Work

This paper has shown the implementation of a complete identity federation in the cloud, respecting the privacy of its users. Identities are managed by Shibboleth and the uApprove privacy plugin. This structure composes an identity federation comprised by IdPs and SPs that runs in Amazon VMs.

Two specific privacy problems could be treated: the lack of user awareness about the release of their attributes to service providers and the lack of concern of

identity providers about the presentation of their terms of use.

The solution proposal combined the use of different applications, such as Shibboleth, uApprove, DokuWiki and the EC2 cloud to show the possibility of implementation of this federation.

The main advantages of having an identity federation in the cloud are the easy replication and instantiation of new identity and service providers, providing scalability for the federation and reducing operational costs.

We could identify as positive aspects of our development the possibility of easy expansion of the federation, or even its integration with already established federations, because of the cloud infrastructure and the use of popular mechanisms such as Shibboleth.

We also have identified some Shibboleth limitations in section 6.1. In Watanabe et al. (2012) and in our proposal, Shibboleth was used in the cloud, but our proposal considers privacy aspects with uApprove. Sánchez et al. (2012) do not use Shibboleth, but a modified Authentic (Lasso) IdP, the ZXID SP and a proprietary client privacy implementation. Chadwick and Fatema (2012) use a privacy policy stuck to the client data that is useful only if the SP supports a correct PDP that can evaluate it and they do not specify how users could choose and enter their privacy policies.

We can conclude that FIM systems should deal with privacy policies in a more refined way. If a good and established system like Shibboleth could be improved in order to deal correctly with privacy policies, it would be more used in cloud environments.

This work is an initial step in the construction of a federated scenario that can be used in the realization of future works and one of the first results of its use, dealing specifically with authorization in a multi-tenant cloud, can be found at Leandro et al. (2012).

As shown in the comparison table presented in the related works, this work follows in the steps of others and tries to clarify the implementation and deployment details, so that others can build and integrate different scenarios and establish good basis for research in federated IAM in cloud computing, which we believe still presents research challenges.

There are lots of possibilities for future development in our scenario. A work that is already in progress is the study and implementation of new access control models using different IAM systems and the eXtensible Access Control Markup Language (XACML) in the cloud.

Exploring the area of privacy, an interesting work would be the automation of compatibility verification between the privacy policies of the service providers and the users.

References

Amazon EC2. [online] `http://aws.amazon.com/ec2/` (Accessed 31 July 2013)

Bertino, E. and Takahashi, K. (2011) *Identity Management: Concepts, Technologies, and Systems*, Artech House

Bradley, A. (2013) 'Shibboleth Enabled Application and Services'. *Shibboleth* [online] `https://wiki.shibboleth.net/confluence/display/SHIB2/ShibEnabled.` (Accessed 31 July 2013).

Bonetti, T. M., Westphall, C. M., de Cordova, A. S., Westphall, C. B. (2011) 'Shib-drm: Anonymous usage licenses', *IEEE Latin America Transactions*, Vol. 9, pp.415-422.

Buecker, A., Lodewijkx, K., Moss, H., Skapinetz, K. and Waidner, M. (2009) *Cloud security guidance: IBM Recommendations for the implementation of cloud security* [online]. `http://www.redbooks.ibm.com/redpapers/pdfs/redp4614.pdf`. (Accessed 31 July 2013).

Chadwick, D. (2009) 'Federated identity management', *Foundations of Security Analysis and Design V*. Vol. 5705, pp.96-120

Chadwick, D.W. and Fatema, K. (2012) 'A privacy preserving authorisation system for the cloud', *Journal of Computer and System Sciences*, Vol. 78, pp.1359-1373.

Chamberlin, D. (2013) 'How to "shibbolize" your application'. CalTech Identity and Access Management [online] `https://wikihub.berkeley.edu/pages/viewpage.action?pageId=15669055`. (Accessed 31 July 2013).

DokuWiki. [online] `http://www.dokuwiki.org`. (Accessed July 31 2013).

Federal Trade Commission. (1998) *Privacy Online: A report to congress* [online] `http://www.ftc.gov/reports/privacy3/priv-23a.pdf`. (Accessed 31 July 2013)

Goth, G. (2011) 'Privacy gets a new round of prominence', *IEEE Internet Computing*, Vol. 15, pp.13-15.

Grobauer, B., Walloschek, T. and Stocker, E. (2011) 'Understanding cloud computing vulnerabilities', *IEEE Security and Privacy*, Vol. 9, pp.50-57.

Harris, S. (2008) CISSP All-in-One Exam Guide, 4th ed., Osborne/McGraw-Hill.

Internet2. *About shibboleth.* [online] `http://shibboleth.net/about/index.html`. (Accessed 31 July 2013).

JASIG CAS. [online] `http://www.jasig.org/cas`. (Accessed 31 July 2013).

Jensen, J. (2012) 'Federated Identity Management Challenges' in *ARES2012: Seventh International Conference on Availability, Reliability and Security*, pp.230-235

Klingestein, N. (2013) 'Build a Federation'. *Shibboleth* [online] `https://wiki.shibboleth.net/confluence/display/SHIB2/BuildAFederation`. (Accessed 31 July 2013).

Klingestein, N. (2013) 'NativeSPEnableApplication'. *Shibboleth* [online] `https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPEnableApplication`. (Accessed 31 July 2013).

Leandro, M.A.P., Nascimento, T.J., dos Santos, D.R., Westphall, C.M. and Westphall, C.B. (2012) 'Multi-tenancy authorization system with federated identity for cloud-based environments using shibboleth' in *ICN 2012, The Eleventh International Conference on Networks*, pp.88-93.

Lee, H., Jeun, I. and Jung, H. (2009) 'Criteria for evaluating the privacy protection level of identity management services'. in *The International Conference on Emerging Security Information, Systems, and Technologies*, pp.155-160.

Lynch, L. (2011) 'Inside the identity management game', *IEEE Internet Computing*, Vol. 15 pp.78-82.

Maggi, F. and Zanero, S. (2011) 'Is the future web more insecure? distractions and solutions of new-old security issues and measures'. in *Second Worldwide Cybersecurity Summit (WCS), 2011*, pp. 1-9

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., Ghalsasi, A. (2011) 'Cloud computing - the business perspective', *Decision Support Systems*, Vol. 51, pp.176-189.

Mather, T., Kumaraswamy, S. and Latif, S. (2009) *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, O'Reilly Media, Inc.

Novakov, I. (2013) Dokuwiki Shibboleth Authentication Backend. [online] `http://wiki.debug.cz/dokuwiki/auth/shib`. (Accessed 31 July 2013).

Olden, E. (2011) 'Architecting a cloud-scale identity fabric', *IEEE Computer*, Vol. 44, pp.52-59.

Orawiwattanakul, T., Yamaji, K., Nakamura, M., Kataoka, T. and Sonehara, N. (2010) 'User-controlled privacy protection with attribute-filter mechanism for a federated sso environment using shibboleth'. in *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 243-249.

Pearson, S. (2009) 'Taking account of privacy when designing cloud computing services'. in *Proceedings of the 2009 ICSE Workshop*, pp.44-52.

Ranchal, R., Bhargava, B., Othmane, L.B., Lilien, L., Kim, A., Kang, M. and Linderman, M. (2010) 'Protection of identity information in cloud computing without trusted third party'. in *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, pp.368-372.

Robertson, A. G. (2009) *brEduPerson*. [online] `http://portal.rnp.br/c/document_library/get_file?uuid=be00d22d-7791-4d12-ac45-637c24afbdae&groupId=797279`. (Accessed 31 July 2013).

Sánchez, R., Almenares, F., Arias, P., Díaz-Sánchez, D. and Marín, A. (2012) 'Enhancing privacy and dynamic federation in idm for consumer cloud computing', *IEEE Transactions on Consumer Electronics*, Vol. 58, pp.95-103.

Takabi, H., Joshi, J.B. and Ahn, G.J. (2010) 'Security and privacy challenges in cloud computing environments', *IEEE Security and Privacy*, Vol. 8, pp.24-31.

TestShib. [online] `https://www.testshib.org/testshib-two/index.jsp`. (Accessed 31 July 2013).

uApprove. [online] `http://www.switch.ch/aai/support/tools/uApprove.html`. (Accessed 31 July 2013).

Watanabe, K., Otani, M., Tadaki, S. and Watanabe, Y. (2012) 'Opengate on cloud'. in *Proceedings - 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012*, pp.1027-1030.

Windley, P. (2005) *Digital Identity*. O'Reilly Media, Inc.