Managing Access to Service Providers in Federated Identity Environments: A Case Study in a Cloud Storage Service

DINIZ, Thomas, DE FELIPPE, Andre Castro, MEDEIROS, Taina, DA SILVA, Carlos <http://orcid.org/0000-0001-9608-439X> and ARAUJO, Roberto

Published version

DINIZ, Thomas, DE FELIPPE, Andre Castro, MEDEIROS, Taina, DA SILVA, Carlos and ARAUJO, Roberto (2015). Managing Access to Service Providers in Federated Identity Environments: A Case Study in a Cloud Storage Service. In: 2015 XXXIII Brazilian Symposium on Computer Networks and Distributed Systems. IEEE, 199-207.

Copyright and re-use policy

Managing Access to Service Providers in Federated Identity Environments: A Case Study in a Cloud Storage Service

Thomas Diniz*, Andre Castro de Felippe*, Taina Medeiros*, Carlos Eduardo da Silva* and Roberto Araujo†
*Metropole Digital Institute, Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil – 59098-970
Email: {thomasfdsdiniz,tainajmedeiros}@gmail.com, andre@anolis.com.br, kaduardo@imd.ufrn.br
†Faculdade de Computação – Universidade Federal do Pará (UFPA)
Rua Augusto Corrêa, 01, Belém, PA, Brazil – 66.075-110 Email: rsa@ufpa.br

Abstract

Currently the diversity of services, which are adhering to Identity Federation, has raised new challenges in the area. Increasingly, service providers need to control the access to their resources by users from the federation as, even though the user is authenticated by the federation, its access to resources cannot be taken for granted. Each Service Provider (SP) of a federation implements their own access control mechanism. Moreover, SPs might need to allow different access control granularity. For instance, all users from a particular Identity Provider (IdP) may access the resources due to some financial agreement. On the other hand, it might be the case that only specific users, or groups of users, have access to the resources. This paper proposes a solution to this problem through a hierarchical authorization system. Our approach, which can be customized to different SPs, allows the SP administrator to manage which IdPs, or users, have access to the provided resources. In order to demonstrate the feasibility of our approach, we present a case study in the context of a cloud storage solution.

Keywords

Federated Identity Environments; Cloud Computing; Access Control;

I. Introduction

Identity management consists of an integrated system of policies, technologies and business processes that allow organizations the treatment and handling of the identities (identity attributes) of its members [1]. From the perspective of a service provider (SP), which makes services and resources available through the Internet, identity management allows an SP to know who its users are (by means of authentication) and to manage what services they are entitle to use (by means of authorization) [2].

Different identity management models have been proposed for dealing with, for example, authentication related issues (e.g. [3], [4]). One example is the federated identity management (FIM) model, where different providers form an association, establishing a trust relationship between them [5]. In this model, *Identity Providers* (IdP) are responsible for authenticating users, sending messages containing authentication and authorization credentials of users to *Service Providers* (SP). In this way, a user can access resources offered by different SPs using a single set of credentials issued by an IdP of the federation.

FIM is a model that has been increasingly used in real world scenarios. One of these scenarios is the Academia. Different universities and research centres are increasingly providing some sort of service that can be consumed by users from different institutions. This scenario has called for an academic FIM environment, which is often characterized by the term "academic federation" [6]. In this way, each institution that takes part in the academic federation provides an IdP for authenticating and emitting credentials of its users, while SPs provide their services to users linked to these institutions. One such academic federation, denominated CAFe (*Comunidade Acadêmica Federada*), is currently maintained by the Brazilian NREN (RNP - *Rede Nacional de Ensino e Pesquisa*). Besides the CAFe federation, RNP also offers a series of services to the Brazilian academic community. Other well known federations are: Chimarrão and CAFe Expresso. Chimarrão is a federation responsible for approval IdPs and SPs candidates to be part of CAFe. Since the CAFe Expresso is a federation for experimental purposes, in a way that new technologies and scenarios are tested on it to be matured and after adhered by the Chimarrão and CAFe federations. SE-CNC has users coming from those three federation.

Although IdPs provide authentication, SP administrators very often intend to control whose users access their resources, and with different access control granularity. This is the case of SE-CNC (Experimental Service - Cloud Computing for Science), an experimental cloud service deployed by RNP. SE-CNC is a cloud storage service for the Brazilian academic community to be offered through the CAFe federation. Because SE-CNC is an experimental service, only users from a few institutions of the federation may have access to the service at the moment, characterizing a scenario where all users from a particular IdP may have access to the resources. On the other hand, we also have the case where only specific users, or groups of users, have access to the service. Moreover, due to its community deployment model, several institutions could collaborate to increase the cloud capacity, which could be reflected on different privileges granted to the users associated with those institutions (e.g., an increased storage quota).

These scenarios describe a problem that can be faced by any SP that joins an identity federation: how to delegate access control policy definition of a particular service to IdP administrators. Such delegation can be used, for example,

in a scenario where an institution has a total quota in the cloud storage service that can be distributed to its users according to its own rules, constrained by the total quota allocated to the institution.

This paper proposes a solution to this problem through a hierarchical authorization system, which can be customized to different SPs. Our approach trusts the authentication provided by IdPs of the federation, allowing an SP administrator to manage which IdPs, or users, have access to the provided resources, and supporting the delegation of the management of users of a particular IdP to its administrator. In order to demonstrate the feasibility of our approach, we present a case study in the context of the SE-CNC experimental cloud storage service.

The remain of this paper is organized as follows. Section II contextualizes the paper, presenting the SE-CNC cloud storage service, the different scenarios that motivated our approach, and a brief background on some concepts used throughout the paper. Section III describes our approach, while Section IV details its implementation. Section V discuss some related work found in the literature. Section VI concludes the paper and present some future directions.

II. CONTEXTUALIZATION

This section introduces the context in which our solution has been applied, describing the SE-CNC cloud storage service. In the sequence, we present a brief discussion on the user management scenarios that motivated our approach. Finally, we present the main concepts related with access control that have been considered by our work.

A. SE-CNC Cloud Storage Experimental Service

SE-CNC[1] is an experimental cloud storage service sponsored by the Brazilian NREN RNP (Rede Nacional de Ensino e Pesquisa). Its aim is to offer a cloud storage service for professors and researchers in Brazil. In order to accomplish this, it provides functionalities similar to known cloud services such as Google Drive, OneDrive, and Dropbox. For instance, it allows users to easily store and share data using computers or mobile devices.

As a differential from other solutions, SE-CNC intends to provide a more secure data storage service. This is achieved by employing cryptography to encrypt the stored data, allowing encryption at both the server side and the client side, and by locating all of its servers within the RNP infrastructure in Brazilian territory, being subject to Brazilian laws.

The SE-CNC cloud follows the IaaS and SaaS models defined by NIST [7]. That is, it has a physical infrastructure composed of servers that support the service as well as it disposes of a friendly environment to its users. In its experimental phase, the service is deployed as a community model. It is planned to be used by the community of RNP users that includes federal universities and research institutes, which contribute with resources to increase the cloud service capacity.

A general view of the SE-CNC cloud service is presented in Figure 1. The cloud service is composed of a frontend and a backend. The frontend is the part of the service that the users interact with by means of three different types of clients: a Web based client, a desktop synchronization client, and a mobile client. Differently, the backend stores users data. It includes all software necessary for storing data, managing and monitoring the cloud. Both the frontend and the backend are based on open source software.

Figure 1. General View of the SE-CNC Cloud Infrastructure.

The solution uses as its front end the ownCloud[2] community edition software. ownCloud acts as a middleware between users and the back end. It authenticates users by means of an identity federation, and control the access to the cloud resources. Through this tool, users can store their data in the cloud. In addition, it allows users to collaborate with others by sharing their data. The CNC cloud service runs in two Web servers geographically away from each other, accessed by a load balancer.

As back end, the cloud storage uses the OpenStack[3] software. In particular, the Swift object storage. The Swift is the tool responsible for storing users data. It ensures the consistency and the availability of the stored data. The Swift tool has two main components: the proxy and the storage services. The proxy service receives all requests to store or obtain data. The storage service stores data and replicates it.

[1]https://cnc.rnp.br/
[2]https://owncloud.org/
[3]https://openstack.org

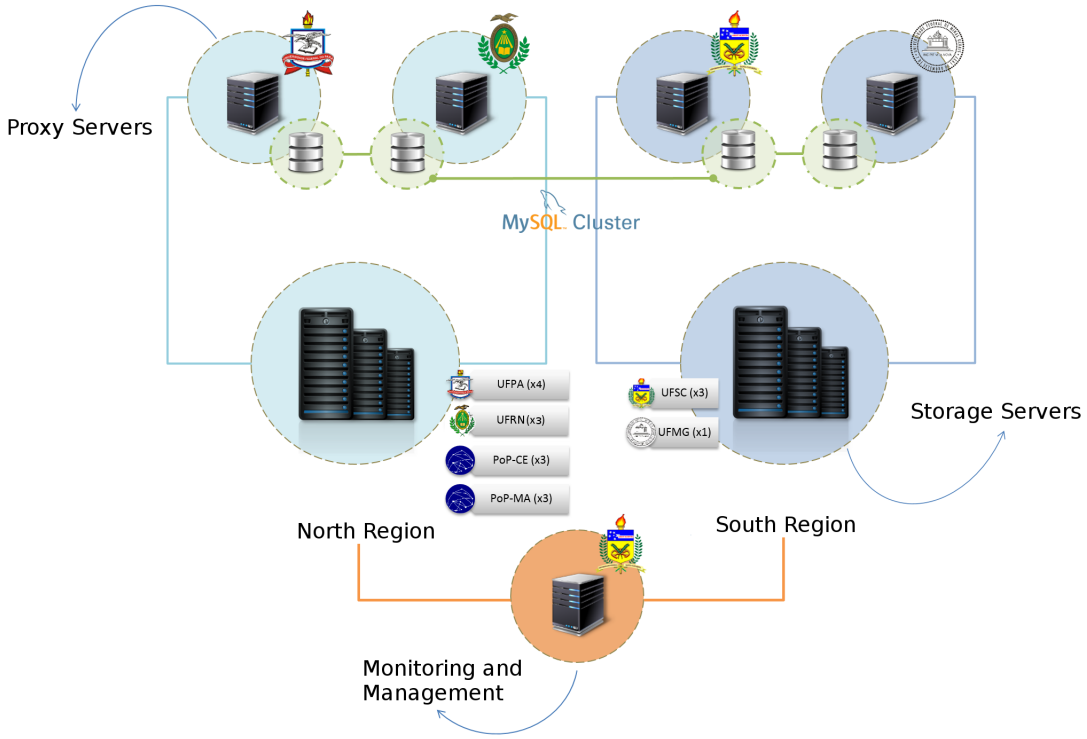

Figure 2. CNC cloud infrastructure.

Figure 2 illustrates the current infrastructure of the CNC cloud. Our infrastructure contains four proxy servers located in federal universities throughout Brazil: federal universities of Pará (UFPA), Rio Grande do Norte (UFRN), Minas Gerais (UFMG), and Santa Catarina (UFSC). Regarding storage servers, the CNC cloud infrastructure contains 15 storage servers distributed in the following way: four servers at UFPA; three servers at UFRN; three servers at UFSC; one server at UFMG; three servers at the RNP Point of Presence of Maranhão (*PoP-MA*) and Ceara (*PoP-CE*).

B. The User Management Scenarios

In its experimental phase, users from 10 institutions have accessed the SE-CNC cloud service. They have been using the service as they see fit, and asked to follow some specific test procedures defined by development team. The users then provide feedback on the tests conducted on a monthly base, and on any other problem that may rise during their daily use.

These interactions with the service clients, and with RNP, have raised two issues related to access control by the SP. The first issue is related to which users (and from which institutions), can access the service. The second issue concerns different access privileges for these users, which is usually captured by means of access control policies specific to the service, i.e., the quota each user has in the storage service.

Concerning the first issue, "the right to access the service", we have identified three different scenarios. Bear in mind that, in these three scenarios, we are still not considering the different privileges users may have within the service.

The first scenario is when we add an SP to the federation, and all users of that federation have access to the service through their respective IdPs. This is the simplest scenario, as the process of joining a federation is enough to allow its users to access the service.

The second scenario is when only users of some IdPs of the federation have access to the service being offered. This is the case when the use of a particular service requires some kind of contractual obligation between the institution offering the service and the institution that manages the IdP. In this case, once the IdP has been "authorized" by the SP, all IdP users will have access to the service. Notice that in this scenario, the trust relationship established by the academic federation is used to provide authentication through IdPs, while the authorization is still controlled by the SP.

The third scenario involves a finer grain access control, where only some users of the federation have access to the SP. This could be the case where the resources offered by the SP are restricted, and either the SP requires some sort of user agreement to be signed by each user, or only specific users of an IdP may have access to the service, as defined by, for example, the institution that manages the IdP.

The second and third scenarios capture the situation dealt with by the SE-CNC cloud storage service. As the service is currently following a community deployment model, it is expected that institutions contribute with the cloud infrastructure (i.e., with the provision of servers), while the users of those collaborating institutions will have access to the cloud service,

reflecting the situation presented in the second scenario. However, the cloud service is still in an experimental phase and thus, only a few users from selected client institutions (10 users for each one of the 10 institutions), has access to the service at the moment. As the choice of which user access the service is made by each participant client institution, there is a need to allow some sort of hierarchical access control mechanism, allowing each client institution to appoint an administrator to manage its user's institution access, delegated by the SE-CNC cloud service administrator. This not only would reduce the burden of the SP administrator, but it also allows more control in the distribution of quotas per users.

Besides granting access to users, a SP still needs to manage the different privileges users have within the service. These access control policies are usually SP specific in the sense that the SP knows what actions can be performed within the service it provides. We consider the situation where users may have different access levels to the service according to rules established by their home institution, i.e., the institution that manages the IdP. Considering the SE-CNC cloud as an example, one particular case that has been raised involves a particular quota allocated to an university (say 100TB), which can then be allocated to the members of the university. For example, all students from a particular department of an university may have a quota of 2 GB, while technical staff may have a quota of 5GB. Since the IdP is managed by an university administrator, we believe that the policies regarding quota should be defined by this administrator, as long as the quota allocated to the University (i.e., the total quota) is not violated.

C. A Background on ABAC/RBAC

According to [8] the central notion of the RBAC (Role-Based Access Control) model is that permissions are associated with roles, and users are assigned to appropriate roles. In this way, an user has access to resources based on the permissions granted to his roles. For example, a subject associated to role Developer has access to a different set of objects than someone associated to role Engineer. Once a subject requests access to an object, the RBAC engine computes which operations are associated to the given role attributed to this subject.

On the other hand, ABAC (Attribute-Based Access Control) is a generalization of the RBAC model. Instead of handling roles, the ABAC model deals with attributes, in a way that subject's requests on objects are granted or denied based on assigned attributes, environment conditions, and a set of policies that are specified in terms of those attributes [2].

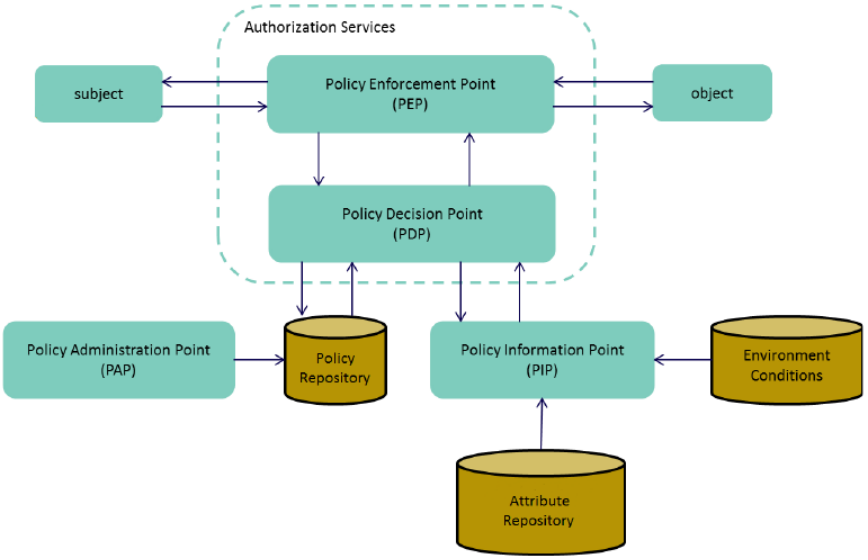

Figure 3. General view of ABAC/RBAC functional points [2].

The ABAC model defined a set of components for its access control mechanism. Figure 3 presents these components. When a subject wishes to access an object, its request goes through Policy Enforcement Point (PEP). The PEP works as a gateway, enforcing policy decisions in response to a subject request to the protected object. PEP communicates with the Policy Decision Point (PDP) in order to guarantee that the user can access the required resource. The PDP component analyses several information, including subject attributes, environment conditions and others to make a decision. For that, the PDP queries the Policy Information Point (PIP) for subject attributes (from an attribute repository) and environment conditions, and a Policy Repository for access control policies. Based on the information collected, the PDP makes a decision that is propagated to PEP, which allows or negates access to the object. A Policy Administration Point (PAP) manages policies.

III. Overview of the Approach

The two issues presented in Section II-B calls for an hierarchical solution that could be applied throughout the federation, easing the management of the different services provided by RNP. Based on this, we have envisioned a solution based on an hierarchy of administrators, which can be seen in Figure 4. In our approach, we consider the existence of a Federated Access Control System (FACS) maintained by RNP as the manager of the federation. The FACS is used by SP and IdP administrators to manage access to different services.

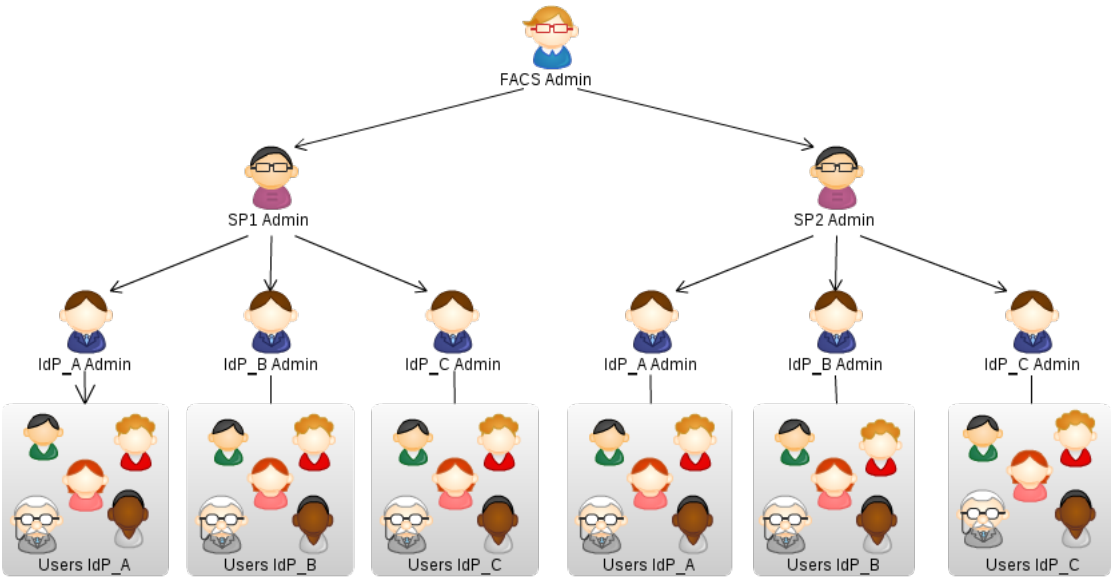

Figure 4. Hierarchical management view.

The FACS administrator is responsible for registering both SPs and IdPs, together with their respective administrators. Each SP administrator can grant IdPs access to its services considering the different access scenarios (i.e., second and third scenarios of Section II-B). The SP administrator can also delegate access decisions to IdP administrators, which manages its IdP users access to the service, with or without an associated restriction (e.g., number of users of the IdP that can access the service).

Besides dealing with which IdP users will have access to the service, this hierarchy is also explored for the definition of access control policies regarding different access levels.

These two issues have been considered as two levels of concerns for our solution. The first level, which we refer to as SP independent, consider the management of which IdP will have access to the SP. Its independence is related to the fact that access control policies for this situation does not depend on the specific actions an user may perform in the service. On the other hand, the SP dependent level deals with the different access control policies in place for the SP, whom which one must know details about the service in order to define a policy.

The FACS architecture, presented in Figure 5, can be seen according to this division. It is important to mention that we employ the NIST ABAC definitions [2], described in Section II-C, when presenting some components and their respective roles. FACS is composed by several modules that will be presented in the following. The *System Administration module* is the module used by the *FACS Admin*, which represents the system administrator. This module is used to register the IdPs and SPs of the federation, as well as their respective administrators. The *IdP Management Module* is the module used by the SP administrator (*SP Admin*) to manage (e.g., add, remove, update) information about his trusted IdPs. The *Access Management Module* is the module used by both, SP and IdP Administrators, for managing user access to SPs, and privileges within SPs. The *PIP* (Policy Information Point) represents the different IdPs of the federation.

FACS deals with SP independent and SP dependent policies. SP independent policies are stored internally in FACS, and used when making decisions about an user access to the SP. SP dependent policies are propagated to the SP by means of a *SP Adapter*. This component is specific to the SP, as it must be able to communicate with the SP's PAP (Policy Administration Point) in order to update its policies, understand the policy language employed by the SP, and their different concepts, which are used to define the policy rules.

On the side of the SP, the PEP (Policy Enforcement Point) module is responsible for enforcing requests related to the SP users (i.e., users of the federation). In our approach, this component needs to be customized to communicate with FACS PDP in order to query whether or not the user has access to the service. As it happens with other SPs, it is necessary to maintain a local base, which is then used for access control management and for helping the provision of the

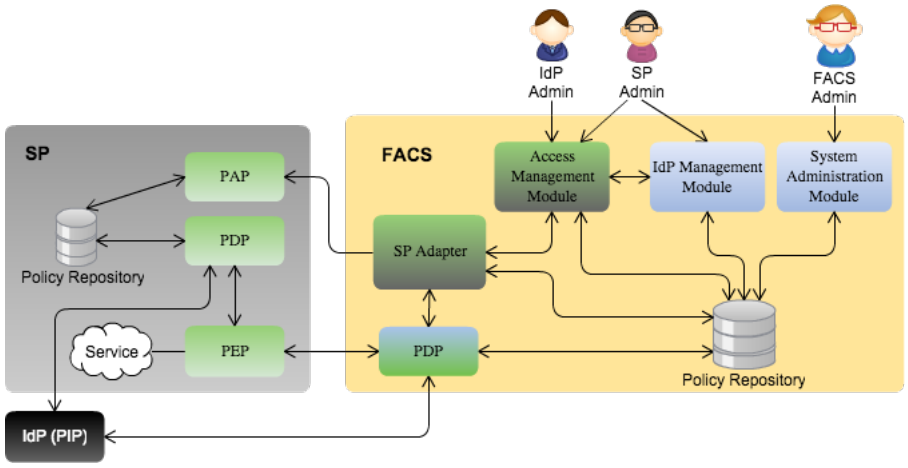

Figure 5. General architecture of FACS.

service. Among the information kept in this base, we have quota policies, files ownership, and file sharing information. This base is represented by the *Policy Repository* in the SP.

In the sequence, we present in Figure 6 the workflow of FACS. Once an user is authenticated by its IdP, the SP PEP enters in action (activity 1), capturing the attributes emitted by the user's IdP to enforce a decision. As we said before, this decision is related to whether the user has access to the service being offered. If the user has access to the service, which can be identified by querying the SP PDP, the PEP grants access (activity 2). In case the user has no access to the service, the SP PEP needs to communicate with the PDP in FACS. At this step two checks are made. First is whether the user's IdP is registered to access the SP (activity 3). If the answer is no, then the user will be denied access to the system (activity 4). In case of yes, a second check is made (activity 5). The second check concerns the different ways an user, coming from a particular IdP, may have its access allowed.

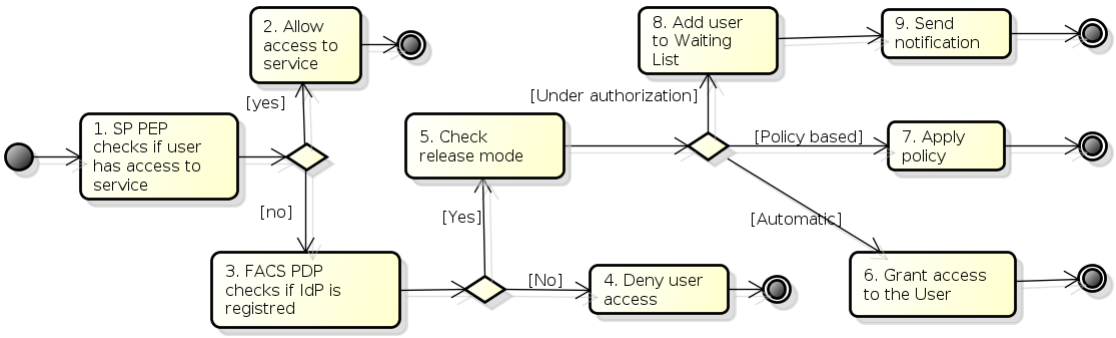

Figure 6. Workflow for access decision with FACS.

We envision an spectrum that ranges from automatic to fully manual access. In one end of this spectrum, the automatic mode, we deal with the situation where all users coming from one IdP have access to SP resources (activity 6). In an intermediary mode, access decisions could be made by means of policies, e.g., ABAC policies (activity 7). Although the decision in this mode can be made in an automatic manner, the policies still need to be defined by SP or IdP administrators. On the other hand, a fully manual mode would involve the addition of users into a waiting list (activity 8). This list would then be evaluated by SP or IdP admins, which decides on an user-by-user basis. Once a decision is made, FACS sends the user a notification (activity 9).

Anyhow, once an user has its access authorized, the SP Adapter is activated to update the user information in the SP Policy Repository. From this point on, access decisions related to the SP user, concerning both access to the service, as well as its privileges within the service, are made by the SP PDP.

IV. Case Study

This section presents the implementation details of FACS, as well as its application for managing access control to users of CNC cloud, followed by with a discussion about the results obtained.

A. Implementation Details

The SE-CNC cloud storage service is based on the ownCloud software, an open-source software for storing files in different back-end infrastructures, supporting functionalities like file sharing, and client synchronization. It supports several back-ends, such as, ftp, local files system, WebDav, and Openstack Swift, which is the case of SE-CNC cloud.

ownCloud is developed in php and java script languages, employs and a SQL database, and has 3 different clients: Web-based, desktop (Windows, Mac and Linux) and Mobile clients (IOS and Android). The desktop client provides synchronization capabilities. In fact, ownCloud is a complete development framework, with its internal architecture divided in several *apps*. There are several apps that can be attached to an ownCloud installation, such as apps for adding calendar and task management capabilities. It is also possible to develop an app in case a particular functionality is not provided, or even, to alter internal ownCloud workings.

Figure 7 presents a mapping of the components proposed by FACS (presented in Figure 5) and the modules of ownCloud. We consider ownCloud to be our SP domain, abstracting away from the underlying Openstack cloud infrastructure. Both ownCloud and FACS are part of the CAFe Expresso federation[4], an experimental federation provided by RNP for the development and testing of solutions.

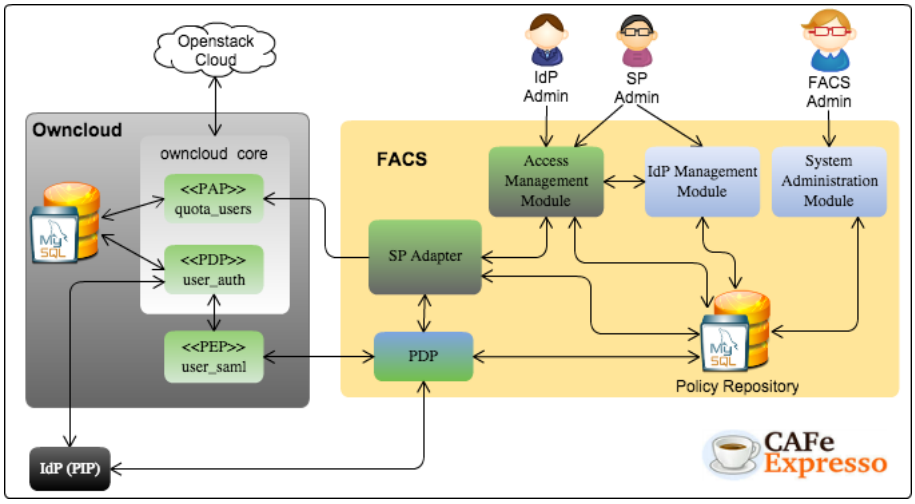

Figure 7. FACS applied to the SE-CNC cloud service.

Regarding the policy repository, ownCloud stores its data using a MySql database. The PEP module in ownCloud environment is implemented by a component called *user_saml*. This module aims to integrate our web client with federated environments, providing SAML authentication based on simpleSAMLphp software [5]. This software handle SAML assertions, the protocol employed by the CAFe Expresso federation. The PDP component is implemented by ownCloud module *user_auth*. This module is responsible for authentication and to approve user access to the service. The PAP module is implemented by ownCloud through *quota_users* module, which deals with all issues related to managing users, policies and quotas. User_auth and quota_users are part of the core owncloud module, as showed in Figure 7.

FACS is a web system implemented in php/Java script languages and its components communicates with a database, in our case, a MySql database. The three main modules of FACS have been implemented as different web interfaces, according with the role of the user accessing the system.

The *System Administration Module* can be seen as simple CRUD application, allowing the management of SPs and IdPs, and their respective administrators who will then be able to use FACS. The *IdP Management Module* supports the association of IdPs and IdP administrators to SPs, implementing the hierarchical delegation capabilities of FACS, as well as the different ways an user access is decided (automatic access, policy based or waiting list).

The *Access Management Module* is used by SPs and IdPs administrators to grant access and associate quotas to users, or groups of users. This module implementation is divided into SP independent and SP dependent. The SP independent part deals with access definition for IdP users, while its dependent part is based on the quota_users module of ownCloud. This module also supports the establishment of quotas to a particular institution, associating it to a particular IdP, which is then distributed by the IdP administrator among its users. SP Adapter is a module that knows how to interact with the SP. It also propagates the result of access control decisions.

[4]https://wiki.rnp.br/display/gidlab/
[5]https://simplesamlphp.org/

B. Discussion

Different versions of FACS have been used by SE-CNC cloud service administrators for about four months, and has now reached a maturity level where it can successfully manage access control of all cloud users (a total of approximately 150 users) belonging to 14 IdPs (including the 10 IdPs from the institutions involved in the tests conducted during the experimental phase, and others that have been added for different reasons.)

Our prototype has been extensively explored considering the third scenario, in which user access is released by both SP and IdP administrators through waiting list and automatic access to particular IdPs.

Although no formal satisfaction survey has been conducted, to this moment, FACS use has presented very satisfying results. It is important to mention that the results obtained have been considered satisfactory when compared with how the access control management was conducted before FACS, and how the problems that used to arise have been mitigated. This was an error-prone process conducted by CNC administrators. Each IdP administrator would obtain its users' attributes, in collaboration with its respective users. The IdP administrator then would inform these attributes to CNC administrators, which would in turn create a script to be run against the CNC system, allowing these users to access the system.

In this context, FACS has provided an automated environment, which helped to mitigate the human errors of the process, as well as reduced the time needed to give an user access to the service. FACS has considerably simplified the effort associated with user management where, by means of its delegation capabilities, each IdP administrator is now able to manage its users without intervention from SE-CNC administrators.

V. Related Work

There are some papers that also propose approaches to handle access control in federated environments. In [9], the goal is to simplify the integration of federated environments with distributed systems. For this, the authors have proposed an approach for access control based on scores, where the set of attributes emitted by an IdP is used to calculate a numerical value. Users are classified according with the score achieved by their attributes, which is then used to define policies to control the access to different SPs.

The SFERA project [10] considers a scenario with non-web SPs. Their approach consist of a tool for allowing federated remote access to console terminals. Similar to our approach, the IdP does not need to be modified. However, they adopt a overly simplified solution for access control, where each authenticated user is associated with a key that is deleted upon the ending of the session.

In [11], is proposed an access control as a service applied to Smart Grids that implements SAML or XACML. An application based on this model was integrated with the Google App Engine (GAE). Due to works on a PaaS cloud, the solution interacts with applications that implement the database supported by the GAE platform. There is a work in the same field that uses cryptographic techniques to control the access and to establish security access, it uses data re-encrypt [12]. It solution is applied to Public cloud storage. We noticed that both approaches is only applied to a such SP or category os SPs, in contrast with FACs that is independent of SP.

FACS has some differences when compared to the solutions cited above. In general, our solution aims to supports different flavours of SPs. Differently from [10], our approach deals with privilege management. Compared to [9] solution, FACS supports different levels of administration and give flexibility to the service to establish its own policies by means of delegation.

Some of the scenarios considered for the SE-CNC service are typical of virtual organizations (VO) [13], in the sense that only part of the users of a federation have access to the service. However, we understand that a VO presents a broader scope than the one currently being considered for FACS. Services offered by VOs can rely on authentication provided by the user's home institution, but they usually involves attributes obtained from multiple sources besides the user IdP. This multi attribute sources model requires some complex mechanism, such as support for attribute aggregation, currently no supported by FACS.

VI. Conclusions and Future Work

This paper has presented an approach for managing access control to service providers in federated environments. The FACS (Federated Access Control System) provides a hierarchical privilege management solution, where access control policies definitions are delegated to IdP administrators, which can then define whose users will have access to the service.

In order to demonstrate the feasibility of our approach, we have implemented a prototype in the context of an experimental service offered by RNP. The SE-CNC cloud is a cloud-based storage service deployed in the RNP infrastructure, which requires different levels of access control. In fact, the development of our solution has been motivated by issues identified during development of SE-CNC cloud during its experimental phase, particularly, regarding its provision as a service of an academic federation. The service has been developed using the CAFe Expresso experimental federation, and has now been migrated to the Chimarrão federation (the federation used by RNP for testing new services before they join the main CAFe federation).

Before FACS, all activities related to user access control were performed manually by the SE-CNC cloud administrators, which quickly became a burden given the crescent number of users the service was attracting. Since the deployment of FACS, the activities related to user management has been considerably simplified, mainly because of the delegation capabilities provided by FACS. An hierarchical based solution allowed us an increased flexibility, and has demonstrated to be an ideal solution for a federated environment. FACS has been able to deal with the three scenarios presented in this paper: (i) access to all users of the federation, (ii) access to particular IdPs, and (iii) access to specific users.

On the other hand, FACS has some limitations. Its use requires an implementation effort for each SP it is applied to. Although the SP dependent modules of FACS have been clearly identified and isolated, its customization process needs to be systematized and evaluated. In its current implementation, the policy based decision making for user access has been simplified to a simple attribute check. This decision was based on the scope and needs of the project at time, and now needs to be revisited. Another limitation is related to the removal of user access. Currently, FACS removes the user from its internal repository and does not propagate this change to the SP. In this way, user removal still needs to be done using the SP mechanisms. We intent to improve the SP adapter mechanism to propagate this, and eventually other, changes to the SP.

As future work, we intend to solve the limitations cited above, especially with respect to deletion of users and its propagation to the SP. In respect to policies, we envision the inclusion of a XACML policy engine, allowing an IdP administrator to specify access control policies regarding user access to services. Another goal is to apply FACS to other SPs, and to measure the effort needed to do so. We envision FACS as a template based system, which could greatly simplify the inclusion of SPs in an identity federation like the CAFE. Furthermore, we intend to extend FACS to be more generic, supporting scenarios such as Virtual Organizations (VOs), looking into new requirements such as mechanisms for attribute aggregation, currently not supported.

REFERENCES

[1] A. Jøsang and S. Pope, "User Centric Identity Management," in *AusCERT Asia Pacific Information Technology Security Conference*, 2005, p. 77. [Online]. Available: http://persons.unik.no/josang/papers/JP2005-AusCERT.pdf

[2] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, and K. Scarfone, "SP 800-162. Guide to Attribute Based Access Control (ABAC) Definitions and Considerations," National Institute of Standards and Technology, McLean and Clifton, VA, United States, Tech. Rep., 2014.

[3] A. Bhargav-Spantzel, J. Camenisch, T. Gross, and D. Sommer, "User Centricity: A Taxonomy and Open Issues," *J. Comput. Secur.*, vol. 15, no. 5, pp. 493–527, Oct. 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1370624.1370625

[4] A. Jøsang, J. Fabre, B. Hay, J. Dalziel, and S. Pope, "Trust Requirements in Identity Management," in *Proceedings of the 2005 Australasian Workshop on Grid Computing and e-Research - Volume 44*, ser. ACSW Frontiers '05. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2005, pp. 99–108. [Online]. Available: http://dl.acm.org/citation.cfm?id=1082290.1082305

[5] D. W. Chadwick, "Federated Identity Management," in *Foundations of Security Analysis and Design V*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5705, pp. 96–120.

[6] E. Q. a. Moreira, Éverton Didoné Foscarini, G. C. da Silva Junior, L. A. O. Alixandrina, L. P. V. Neto, and S. Rossetto, *Federação CAFe implantação do provedor de identidade*. RNP/ESR, 2014.

[7] P. M. Mell and T. Grance, "SP 800-145. The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Gaithersburg, MD, United States, Tech. Rep., 2011.

[8] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996. [Online]. Available: http://dx.doi.org/10.1109/2.485845

[9] E. F. Silva, D. Muchaluat-Saade, and N. C. Fernandes, "Controle de Acesso Baseado em Políticas e Atributos para Federações de Recusos," in *XIV Simposio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg 2014*, 2014. [Online]. Available: http://www.lbd.dcc.ufmg.br/colecoes/sbseg/2014/0073.pdf

[10] M. M. Galheigo and A. T. A. Gomes, "Um Estudo Sobre Autenticação Federada no Acesso a Recursos Computacionais por Terminal Remoto Seguro," in *XIV Simposio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg 2014*, 2014.

[11] G. Ryba, M. Jung, and W. Kastner, "Authorization as a service in smart grids: Evaluating the PaaS paradigm for XACML policy decision points," in *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*, Sept 2013, pp. 1–4.

[12] Y. Zhang and J.-L. Chen, "Access Control as a Service for Public Cloud Storage," in *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, June 2012, pp. 526–536.

[13] J. Cummings, T. Finholt, I. Foster, C. Kesselman, and K. Lawrence, "Beyond Being There: A Blueprint for Advancing the Design, Development, and Evaluation of Virtual Organizations. Final report from the NSF workshops on Building Effective Virtual Organizations." National Science Foundation, Tech. Rep., 2008.