Introducing Federated WebDAV Access to Cloud Storage Providers

Sebastian Rieger
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Karlsruhe, Germany
sebastian.rieger@kit.edu

Harald Richter
Department of Informatics
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
hri@tu-clausthal.de

Yang Xiang
Rechenzentrum Garching
Max-Planck-Society
Garching, Germany
yang.xiang@rzg.mpg.de

***Abstract*—Affordable access to large online hard disks via the Internet has emerged by the continuous evolving of public and private storage clouds. However, difficulties arise as soon as users of such storages want to employ services from different cloud providers simultaneously, e.g., for collaboration among institutions that use different storage providers or for distribution of data backups. The reasons for this are dissimilar user accounts and incompatible access methods. This contribution describes a solution to that problem that does not need additional middleware to achieve the goal of unified authentication, authorization (AA) and access except WebDAV, which is an open standard. Our method is based upon a dynamic localization of the user by means of a world-wide unique user name. The solution is thus suitable for implementing federations of storage clouds in which multiple organizations can jointly provide a unified access to file systems that are distributed across the Internet.**

Keywords-Dynamic Federation; SAML; WebDAV; Storage; Cloud Computing

I. INTRODUCTION

A continuously increasing number of providers are offering online storage over the Internet. These providers are utilized by home users and professionals as well, for example to backup or to share files. The access to these files - although stored at different physical hard disks in the Internet - is transparent to the users because the services are cloud-based. Since the number of storage providers is already high, users also want to benefit from more than one provider at the same time, e.g., for exploiting the free storage space the providers offer them. We propose WebDAV [20] as a convenient way to access files in the cloud and over the Web. It is supported by various operating systems. However, WebDAV relies on the AA mechanisms of the underlying Web servers, which is why users have to maintain different credentials for each provider using an individual Web server. Furthermore, they have to login separately to every provider, thus creating multiple sessions simultaneously. A unified access across different public cloud storage providers is therefore not possible as of today. This also holds for private storage clouds that are established to offer access to distributed storage for users across different institutions. These problems are addressed in part by federated and user-centric identity management systems based on SAML [22] or OpenID [34] that offer Single Sign-On and unified AA across distributed Web applications.

In this paper, we introduce a solution developed to enhance WebDAV access to online storage with federated, SAML-based AA. An augmented WebDAV client was implemented by us to support HTTP redirects and sessions, as defined in the SAML profiles. The client is based on Shibboleth [17], which is a widespread SAML implementation in scientific communities.

Shibboleth is focusing on Web applications and requires the user to access his resources using a fully configured Web browser to handle HTTP sessions with storage providers and to manage the JavaScript- or HTTP redirects that enable the SAML-based Single Sign-On or the selection of the institution the user is affiliated to. To allow direct WebDAV-based file access e.g., in a file explorer without using a Web browser, we extended our client to support dynamic federation that allows an automatic discovery of the users' institution. The solution described in this paper allows consistent file access across different providers as in a single virtual file system. It enables federations that are spanning over multiple locations and companies to build-up a distributed, scalable and fault-tolerant file system across multiple cloud storage providers.

In Section 2, the state-of-the-art in WebDAV-based storage clouds is explained. Section 3 describes the mechanisms to enable federated AA for WebDAV-based file access in storage clouds using our novel combination of these techniques. Section 4 presents the implementation, together with the extensions to WebDAV and Shibboleth. Finally, Section 5 summarizes the results and gives an outlook to future research.

II. STATE-OF-THE ART IN WEBDAV-BASED STORAGE CLOUDS

There are several research groups such as [2] and [25] that are also working on Shibboleth-compatibility in WebDAV, however at the server side. An early version of this concept was for instance introduced by [17]. A solution similar to that but for grid environments was described in [10]. It is based on iRODS [10]. Other differences of these projects to our concept are that they do not employ WebDAV clients to support Shibboleth- and SAML-based

 ISBN: 978-1-61208-153-3

AA and focus on a Web browser instead. Additionally they treat storage providers as isolated items and therefore do not allow a unified, transparent access to different storage locations as offered in storage clouds.

A. *Isolated Storage Clouds*

State-of-the-art in storage cloud technology is that clouds are isolated as islands. Each cloud can be accessed by a user individually, and without any relation to other clouds as an online hard disk, either via a proprietary Web interface or via special software delivered by the cloud provider. Both methods allow to access and administer directories (sometimes called buckets) and files. Recent examples for such storage clouds are Amazon S3 [21], Google Storage [8] and Microsoft Azure Storage [1]. Based on these isolated clouds, additional services and providers have come into existence, which simplify access to and usage of online cloud storage for the end-user, such as DropBox [6], Mozy [16] or Ubuntu One [30]. However, a common de-facto or de-jure standard for an overall AA and access to multiple storage clouds does not exist yet. The industry consortium SNIA (Storage Networking Industry Association) [27] is working on such a standard called Cloud Data Management Interface [3] but it is unknown when it will be available. Furthermore, beside public clouds that already exist in IT infrastructures of scientific communities, private clouds are more and more emerging, e.g., as described in [11] and [28].

Private clouds are often based on open source implementations of online storage such as Eucalyptus Walrus [31]. They excel by providing a unified and thus simplified access method for a closed group of users from different institutions that is independent of the specific location the user wants to access his files from. The realization of such comfortable access normally needs proprietary applications and user interfaces to handle the AA. An example therefor is the AA infrastructure (AAI) of the Internet2 [4] or of the German research network DFN [5]. The underlying technology is usually SAML [22]. Shibboleth enables Single Sign-On and therefore unified AA across services, such as storage providers, that are joined in a cloud.

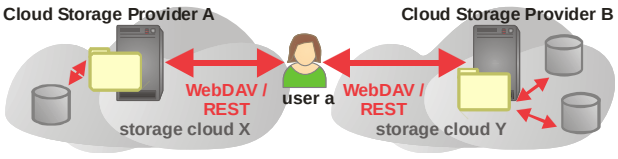

Figure 1. Access to WebDAV- or REST-based online file systems of different storage clouds.

This method is shown in Figure 1, together with a WebDAV- or REST-based access method. The REST application protocol is described in [18]. Providers for public storage clouds may also have a proprietary user interface. Examples therefore are the REST-based APIs in Amazon S3 or Google Storage which can be accessed only by special API function calls and by clients that are downloadable from these companies. The clients needed therefore typically map all file and directory accesses onto the HTTP methods PUT, GET, POST and DELETE which is similar to a REST-based approach. On top of these services, a few other providers offer a WebDAV access which enables users to create, read, write, move, rename and delete files and directories in an isolated storage cloud without proprietary applications or APIs. This was a paragon to us.

B. *User Credentials*

AA is typically performed with username and password as credentials. However, users must keep and maintain individual usernames and passwords for every provider e.g., because of company-dependent regulations with respect to password lengths and restrictions in the usage of numbers and special characters, because of the users' security concerns and because of the disjoint management of storage clouds used by different providers. As a consequence, no contemporary provider offers Single Sign-On across cloud borders yet. Additionally, the granting of read and write permissions is technically possible only within the runtime environments of the individual Web servers of a cloud storage provider that are in turn limited by their underlying file systems.

III. AA AND ACCESS IN FEDERATED STORAGE CLOUDS

The natural extension of isolated storage clouds lies in the coupling of them into federations. In [33], such a federation is described, which is based on REST and which uses uniform resource identifiers (URIs) instead of uniform resource locators (URLs). Beside HTML, it employs XML in its REST response packets. The disadvantage of that solution is that additional middleware is needed at the client side for accessing online file systems.

This paper describes a new approach for utilizing federations of storage clouds without extra middleware. To achieve this, the WebDAV protocol was employed that substantially augments the REST paradigm. Additionally, a Shibboleth-capable WebDAV-Client was developed by us as a replacement for Web browser-based user localization. It substitutes also static authentication and authorization by AA for a dynamic federation as it was described in [33]. By these measures, end-users can access federations that are evolving over time with respect to the number of users (which is mostly the case) and that are composed of various cloud providers (which is new) without installing middleware and without repetitive login. The latter feature results in Single Sign-On.

Using our approach, the WebDAV module mod_dav [12], which is native to the Apache Web-Server, can be engaged together with the mod_shib Shibboleth module [13] without any extensions on the server side and without using a Web browser to access the files on the client side, while maintaining Shibboleth compatibility at the same time. Our solution is also compatible with other Web servers such as Microsoft IIS. Only the WebDAV functionality at the client side had to be extended.

 ISBN: 978-1-61208-153-3

A. *Unique User IDs and Access Rights*

In our concept, user rights are not mapped to a username for the purposes of each individual provider but to a world-wide unique user ID, which is the email address [29]. From the email address, a second mapping is made to user rights that are valid for a service within a federation, i.e., also across cloud boundaries. Other examples for world-wide unique IDs are the Mícrosoft Security Identifier (SID) [26], the Globally Unique Identifier (GUID) e.g., for DCOM in Windows, or the Universal Unique Identifier (UUID) e.g., in the Interface Definition Language (IDL) of the Distributed Computing Environment (DCE). Usual Unix/Linux systems, however, have only the User Identifier (UID) [29] as defined in the POSIX standard, which is not globally unique. Thus, the simultaneous access to multiple storage clouds results in a problem when mapping the user name to a UID in Unix. Using our solution, this can be circumvented by outsourcing AA to an extra service, which is called identity provider, as described in the next section.

B. *Identity Provider*

In SAML-based federations [22], service providers (SPs) completely outsource the authentication to one or more identity providers (IdPs). A prominent example for such a federation in a scientific environment is the AA infrastructure of the Internet2 (= InCommon [4]) or the German research network DFN (= DFN-AAI [5]). We explain the functioning of DFN-AAI by means of an arbitrary employee at Max-Planck Society. This employee is easily authenticated and authorized by the IdP that is located at his home institute because the user is registered there. The IdP is responsible for all users of that specific institute and manages a well-defined set of access rights for its users via their usernames. However, the example employee can also profit from services that are offered at institutions outside of his home institute, even outside of Max-Planck Society itself, solely by his institute username, provided that those institutions also participate in DFN-AAI.

With this concept, Single Sign-On is possible across cloud borders because only the IdP that is responsible for its user performs AA. This holds if all cloud service providers (CSPs) of a (dynamic) federation make use of this IdP that is responsible for the example user. The described method is depicted in Figure 2.

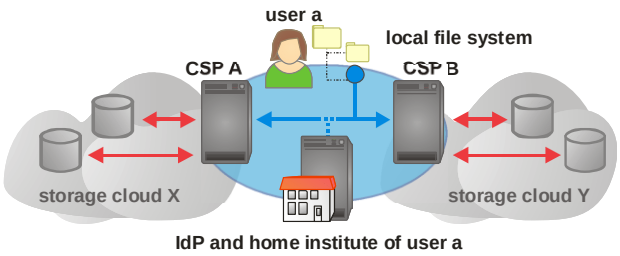

Figure 2. Single Sign-On in a (dynamic) federation of storage clouds by means of an Identity Provider (IdP).

Because of better scalability and the inclusion of many institutes, typically more than one IdP will exist in a federation of multiple cloud storage providers (CSPs). All IdPs must be registered at every provider. To simplify the operation, administration and management of this set of IdPs, the federation can make use of a function that is called „Identity as a Service“ (IDaaS).

In a scenario of CSPs, IdPs and IDaaS, the identity service is a central instance that is connected to the CSPs in a star topology and that acts for them as a proxy of the IdPs. The advantage for the CSPs resulting from the star topology and the proxy method is a significant simplification in AA because CSPs have to establish only a trust relationship to this central identity service and not to all IdPs. Subsequently, three indirections of trust come into existence according to the law of transitivity, starting with the trust relation from one CSP to the central IDaaS, and continuing with the trust relation to the individual IdP and finally to the user.

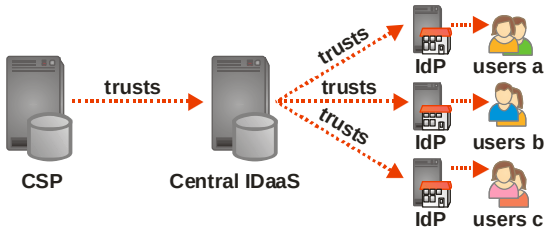

Figure 3. Multiple indirection of trust for AA in a federation according to [33].

This chain of indirection is depicted in Figure 3 and was first described in [33].

C. *Discovery Service*

In the following, it is assumed that the SAML-based Shibboleth system is used for AA. This is the groundwork for our solution. Shibboleth in turn, employs for user localization, i.e., for the determination of the user's home institute or organization a so-called discovery service (DS). The DS provides a Web page to all users under a static URL that is a priori known so that the CSPs can redirect the users to this page. If subsequent users request a service from a provider in the federation, the request is redirected first to the DS's Web site. On this Web page, a list of organizations that constitute the federation is presented to the user. Then, the user selects his home organization from the list, and thereby the IdP that is responsible for him. Afterwards, the browser window of the user is redirected a second time by an HTTP method to the responsible IdP, and the user must enter his username and password for AA.

D. *Confederations*

Several federations of storage clouds may even be coupled to an umbrella organization called confederation, for instance by using eduGAIN [7]. Then, the DSs of the constituting federations are cascaded, and the user must select first his home federation on the Web page of that DS

of the highest level, and proceed afterwards to the Web sites of the next lower levels. For this hierarchical procedure, the first DS presents a list of federations to the user that build the confederation, and in which the user selects his own storage cloud federation. After the selection, the Web page is redirected via HTTP to the DS of that federation where the user can select his home organization, and finally from there to his IdP where AA is accomplished as described.

E. User Comfort

Users wish for reasons of comfort a direct connection to their online file systems in the way of a virtual file system, as it was depicted in Figure 2, and they do not want to open an extra browser for this purpose, i.e., in order to select their federation or home organization. Nor do they want to switch continuously between a file explorer, such as Windows Explorer, and a Web browser for every list, move, rename and delete operation across cloud borders. A direct Web-based file access is instead accomplished by WebDAV. Furthermore, separate logins into all storage providers would result in a user-unfriendly approach. Finally, a Web page in the browser would fail if the number of IdPs in a storage cloud is rapidly changing over time or if the number of all clouds in a federation is time-dependent. Because of these reasons, we implemented a dynamic discovery service for user localization similar to that described in [32].

F. Dynamic User Localization

For dynamic user localization, the user first enters a static and a priori known URL of his CSP. Then he enters his email address instead of selecting his home organization, and our dynamic discovery service sends a DNS request to the responsible DNS server based on the domain name of his email address. The DNS response is a DNS NAPTR record in which his home organization has previously entered the IP address of the IdP that is responsible for him. If later a login of one of the users has to be performed then the user's NAPTR record is evaluated by the CSP, and with the data contained herein the proper IdP can be requested for AA ([33]). By this dynamic user localization, users can perform AA at their IdP from everywhere, and cloud borders are irrelevant. After successful AA, the user is automatically logged-in to all other CSPs of the federation (= Single Sign-On).

It is also possible to use the domain name of the user's email address as a shortcut to the domain of the IdP that is responsible for him. Another alternative to allow a domain-based DS using cascaded IdPs was described in [24].

IV. Software Architecture of Our Implementation

In this section, the software architecture of a Shibboleth-capable WebDAV client is described that can access federations of storage clouds. The client is based on an extension of the open source WebDAV client Sardine [23] and is therefore written as a Java Swing application. Our prototype supports down- and upload of directories and files from and to the online file system with a simple drag-and-drop command. Sardine in turn utilizes the known Apache HTTP Client [9] to access the WebDAV server. To allow for AA with Shibboleth, Sardine was extended to support HTTP sessions, redirects and SAML profiles.

This extension processes the HTTP redirects that are needed for the SAML-based Shibboleth system. Additionally, the extension provides the dynamic user localization for the CSP. Furthermore, it extracts the SAML response and the so-called relay state the user's IdP has sent after successful AA and transmits these data back to the user's CSP via a SAML HTTP POST profile. We have also extended Sardine with a simple session management to allow for stateful connections of the user while he is logged in. The user state is preserved in HTTP session cookies [19] that the CSP and the IdP have written by means of the WebDAV client during the user session. The WebDAV client can make use of this state information as needed. The session management enables Single Sign-On across different federated CSPs. Finally, the AA procedure of the IdP is performed such that the user's IdP and CSP exchange SAML attributes that define his access rights. Beside the CSP, also the WebDAV server can utilize the attributes (represented in HTTP headers), which grant or deny access to the underlying file system.

Inside of the Apache Web server used at the CSP, two modules named mod_dav and mod_davfs constitute the WebDAV server. While mod_dav implements the WebDAV protocol handling, the mod_davfs allows direct usage of the file systems that are available on the server side. The module mod_dav uses the module mod_auth from the Apache server and extensions of it for AA. Shibboleth is such an extension to mod_auth called mod_shib.

Using these modules it is possible to configure a so-called WebDAV location inside of Apache that uses the access control features of mod_shib. By requesting a resource from this location, the user is redirected to the DS and a subsequent AA is performed as described in the previous section. After successful AA (using the email address), the files in the location can be accessed and modified by our WebDAV client.

An example of an HTTP session offering Single Sign-On across different storage clouds and federations is shown in Figure 4.

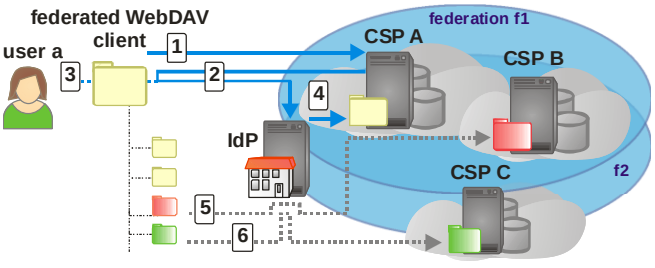

Figure 4. Federated WebDAV access to multiple cloud storage providers (CSP) and federations.

In step 1, our federated WebDAV client is used to access a cloud storage resource of CSP A, which is a member of federation f1. The corresponding request contains the e-mail address of user a. CSP A uses the e-mail address provided in step 1 to dynamically discover the user's home organization and IdP as described in the previous section. Step 2 depicts the redirection of the user to the corresponding IdP using the SAML redirection profile. The IdP requests the user to provide his e-mail address and password as login credentials using the WebDAV client in step 3. During the redirection process the WebDAV client stores the session cookies initialized by CSP A and the IdP. If the IdP is able to successfully authenticate the user by the credentials provided, then it transmits a SAML assertion to CSP A in step 4. Technically this is again performed using a redirection process.

As shown in Figure 4, the user has access to different folders via the federated WebDAV client. In step 5, the user requests a resource from CSP B that is also a member of federation f1. Using the e-mail address of the user sent by the federated WebDAV client in step 5, CSP B discovers the same IdP and redirects the client to ensure the AA. As the federated WebDAV client has already established an HTTP session with the IdP, we include the corresponding session cookie in the request on the client side. Hence the IdP does not require an additional authentication of the user, and the user gains seamless access to the resource.

One of the application areas of our solution is the federation of storage providers for private storage clouds. Such private clouds can be found, for example, in scientific communities in which bodies that are funded by the same organization and that are members of the same federation want to jointly aggregate their storage environments. Another scenario could be the aggregation of multiple public cloud storage providers e.g., in order to enhance flexibility and fault tolerance on the users' side.

In such scenarios, the need to access resources offered by CSPs that are not members of the same federation arises, especially with respect to distributed scientific communities that cooperate between multiple countries. Step 6 in Figure 4 illustrates this case. It is shown how resources of CSP C are accessed while C is a member of federation f2 and therefore outside the boundaries of federation f1. As we use a dynamic discovery of the user, our solution is able to handle such confederations.

A severe problem with respect to security arises if the federation is time-dependent in the number of their constituting CSPs, and thus in their number of users because the reliability and trustworthiness of new users may be unknown to the other users and to the CSPs as well. To overcome this problem, we have developed a so-called Trust Estimation Service (TES) [32][33]. This service can be incorporated into every IdP and CSP to increase security. The service is implemented as an extension to Shibboleth and it is thereby also an extension to the dynamic discovery service (DS). The extension utilizes the Internet Domain Name System DNS as described. The basic procedure of trust estimation is depicted in Figure 5.

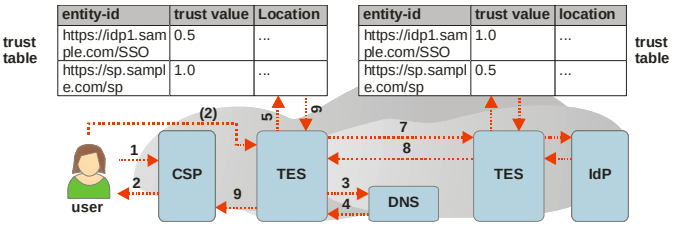

Figure 5. Extension of Shibboleth with a Trust Estimation Service according to [33].

In step 1, the user accesses a CSP and is requested to provide an e-mail address that is transmitted to the TES in step 2. The TES uses the domain name system (DNS) to obtain the Entity-ID of the user's home IdP (steps 3 and 4). Using this Entity-ID, the local TES selects the corresponding end-location in his trust table (steps 5 and 6). If the IdP is trusted, according to its previously calculated trust value (as described in [33]), the local TES of the CSP sends a request to the IdP's TES to retrieve the IdP's meta data in steps 7 and 8. Finally, the meta data is forwarded to the CSP in step 9.

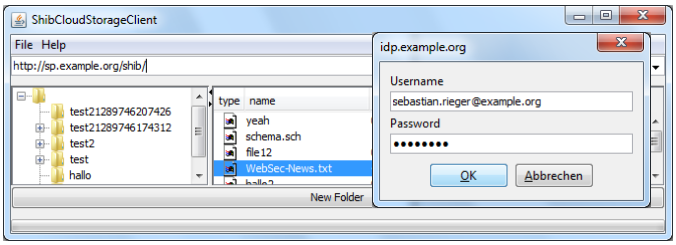

Figure 6. User interface to access federations of storage clouds via Shibboleth and WebDAV.

The resulting user interface of our WebDAV client is shown in Figure 6.

V. Conclusion and Future Work

The proposed solution allows unified authentication, authorization and data access in a federation of storage clouds. It can simultaneously present multiple online file systems from different cloud service providers to users as a virtual file system. It is based on Shibboleth and WebDAV that are extended by a dynamic user localization and a trust estimation service. This allows for Single Sign-On across cloud borders, for increased security in the federation, and for user-friendly access to the various storage clouds. Applications for the solution may be the federation between the scientific institutes of the Max-Planck-Society MPG-AAI [14] and the universities of the country of Lower Saxony Nds-AAI [15].

In the future, we will evaluate the performance of our approach and the influence of the WebDAV server configuration and protocol headers. The performance

 ISBN: 978-1-61208-153-3

evaluation will include an in-depth analysis of the TES and the established dynamic federation.

Additionally we'll focus on how object-oriented file systems e.g., based on NoSQL databases, can be integrated as a backend on the WebDAV server side. Furthermore we are planning to evaluate the inclusion of user-centric authentication (e.g., OpenID) and special authorization mechanisms such as OAuth to allow for the delegation of access rights across aggregated global file systems.

REFERENCES

[1] Windows Azure Storage, http://www.microsoft.com/windowsazure/ 9.5.2011.

[2] T. Bellembois and R. Bourges, "The open-source ESUP-Portail WebDAV storage solution", http://www.esup-portail.org/download/attachments/43515911/ESUPWebDAV.pdf 9.5.2011.

[3] Cloud data management interface, SNIA Web Site, April 2010, http://cdmi.sniacloud.com 9.5.2011.

[4] InCommon Identity and Access Management, http://www.incommonfederation.org/ 9.5.2011.

[5] DFN-AAI - Authentication and authorization infrastructure, https://www.aai.dfn.de 9.5.2011.

[6] Dropbox, http://www.dropbox.com/ and http://en.wikipedia.org/wiki/Dropbox_%28service%29 9.5.2011.

[7] eduGAIN, http://www.edugain.org/ 9.5.2011.

[8] Google Storage, http://code.google.com/intl/de-DE/apis/storage/ 9.5.2011.

[9] Apache HttpComponents, http://hc.apache.org/ 9.5.2011.

[10] S. Zhang, P. Coddington, A. Wendelborn, and A. Davis, "A generic interface for iRODS and SRB", 10th IEEE/ACM International Conference on Grid Computing, Banff, 2009.

[11] K. Keahey, R. Figueiredo, J. Fortes, T. Freeman, and M. Tsugawa, "Science Clouds: Early Experiences in Cloud Computing for Scientific Applications", Cloud Computing and Applications, 2008.

[12] Apache Module mod_dav, http://httpd.apache.org/docs/2.0/mod/mod_dav.html 9.5.2011.

[13] R. L. Morgan, S. Cantor, W. Hoehn, and N. Klingenstein, "Federated Security: The Shibboleth Approach", EDUCAUSE Quarterly, Vol. 27, 2004, S. 12-17.

[14] Max-Planck-Gesellschaft - MPG-AAI, https://aai.mpg.de 9.5.2011.

[15] Nds-AAI, Authentifizierungs- und Autorisierungs-Infrastruktur für Niedersachsen: http://www.daasi.de/projects/ndsaai.html 9.5.2011.

[16] Mozy, http://mozy.com and http://en.wikipedia.org/wiki/Mozy 9.5.2011.

[17] L. Ngo and A. Apon, "Using Shibboleth for Authorization and Authentication to the Subversion Version Control Repository System", Fourth International Conference on Information Technology - ITNG '07, Las Vegas, 2007.

[18] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures". PhD thesis, University of California, Irvine, 2000 and http://de.wikipedia.org/wiki/Representational_State_Transfer 9.5.2011.

[19] D. Kristol and L. Montulli, "HTTP State Management Mechanism", ftp://ftp.rfc-editor.org/in-notes/rfc2965.txt 9.5.2011.

[20] L. Dusseault, "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)" ftp://ftp.rfc-editor.org/in-notes/rfc4918.txt 9.5.2011.

[21] Amazon Simple Storage Service (Amazon S3), http://aws.amazon.com/de/s3/ 9.5.2011.

[22] OASIS: Security Services (SAML) TC, http://www.oasis-open.org/committees/security/ 9.5.2011.

[23] sardine - an easy to use webdav client for java, http://code.google.com/p/sardine/ 9.5.2011.

[24] S. Rieger, "User-Centric Identity Management in Heterogeneous Federations", Fourth International Conference on Internet and Web Applications and Services, 2009.

[25] DataFinder: A Python Application for Scientific Data Management, EuroPython 2008: The European Python Conference, Vilnius, 2008

[26] Security Identifier, http://en.wikipedia.org/wiki/Security_Identifier 9.5.2011.

[27] Storage Networking Industry Association, http://www.snia.org and http://en.wikipedia.org/wiki/Storage_Networking_Industry_Association 9.5.2011.

[28] J. Staten, S. Yates, J. Rymer, and I. Nelson, "Which Cloud Computing Platform is Right for You?", Forrester Research, 2009.

[29] User identifier, http://en.wikipedia.org/wiki/User_identifier 9.5.2011.

[30] Ubuntu one, https://one.ubuntu.com/ 9.5.2011.

[31] Interacting with Walrus (2.0) - Storage Service, http://open.eucalyptus.com/wiki/EucalyptusWalrusInteracting_v2.0 9.5.2011.

[32] Y. Xiang, J. A. Kennedy, H. Richter, and M. Egger, "Network and Trust Model for Dynamic Federation", The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences, Florence, 2010.

[33] Y. Xiang, S. Rieger, and H. Richter, "Introducing a Dynamic Federation Model for RESTful Cloud Storage", The First International Conference on Cloud Computing, GRIDs, and Virtualization, Lisbon, 2010.

[34] OpenID specification, OpenID Web site, http://openid.net/developers/specs/ 9.5.2011.

 ISBN: 978-1-61208-153-3