An Approach in the Design of Common Authentication Solution for a Multi-Platform Cloud Environment

Primož Cigoj[1,2], Borka Jerman Blažič[2] and Tomaž Klobučar[2]
[1]Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2]Jozef Stefan Institute, Laboratory for Open Systems and Networks, Jamova cesta 39, 1000 Ljubljana, Slovenia
{primoz, borka, tomaz}@e5.ijs.si

Keywords: Single Sign-on, Identity Management, Identity Federation, Cloud Computing, Security, Cloud Management, Cloud Provisioning, Infrastructure-as-a-Service, IaaS, Multi-platform Cloud, Access Control, Authentication, Authorization, Cloud Service Provider, Privacy, Software Platform, Centralized Systems, OpenStack, VMware.

Abstract: The security provision within multi-platform cloud computing environment is still considered not to be properly solved due to different problems with technical and human-based origin. This paper presents an attempt to provide an authentication and authorization solution based on the single sign-on (SSO) approach for cloud service users and administrators in a multi-platform environment. The problem of authentication in cloud services is briefly introduced and the approach implemented for cloud environment with two different proprietary (VMware) and open source (OpenStack) platforms is described.

1 INTRODUCTION

Cloud computing has revolutionized the provision of computing services, transferring them from local to locally unspecified remote environments, which are controlled by third party service providers. The main cloud computing challenge for vendors, providers and users of clouds remains protection of the cloud technology, services and users with adequate security measures. According to the survey conducted by Microsoft and the National Institute of Standards and Technology (NIST), security in the cloud computing model was the ICT executives' main concern (Jansen and Grance, 2011; Microsoft, 2010). Consequently, many business entities are still not very keen on adopting cloud computing.

In the cloud users alone have no control over their data and their cloud identity. Installation procedures are often complex and there is no adequate and complete security solution that ensures the safe deployment of the underlying infrastructure and safe use of the services. Especially in the multi-platform clouds and inter-clouds, where multiple cloud systems can be accessed, users and administrators are faced with different authentication and authorisation systems, different login dialogs, and each login dialog is matched with different credentials. In addition to being difficult implementing strong authentication at the user level (Tripathi and Mishra, 2011), it is also complex to manage and create authentication mechanisms for several services and several platforms (Fernandes, Soares, Gomes, Freire and Inácio, 2014). Identity federation and single sign-on (SSO) techniques address these issues by allowing exchange of authentication and authorization information between two parties, such as an Identity Provider (IdP) and a Service Provider (SP).

While the identity federation and single sign-on concepts have been well covered in the literature in the past years, in general as well as in the cloud context e.g. (Pérez-Méndez, Pereniguez-Garcia, Marin-Lopez, López-Millán and Howlett, 2014), (Cruz Zapata, Fernández-Alemán and Toval, 2014), there are still practical issues that prevent their straightforward and simple application in the multi-platform cloud environments. Our attempt described in this paper was to design and develop a solution that would provide trustworthy and secure authentication and authorization service in such environment. The proposed solution is based on the SSO principle and was implemented on two different platforms (OpenStack and VMware). It was tested and evaluated within a large National Competence Centre project on cloud-assisted services for different fields of application.

The paper discusses the problem and related work first. Then it introduces the selected SSO- based

solution, and describes the development approach and the implementation course. The presentation ends with discussion and plans for future work.

2 AUTHENTICATION AND IDENTITY MANAGEMENT SERVICES IN MULTI-PLATFORM CLOUD INFRASTRUCTURE

According to the Cloud Security Alliance (CSA) (Simmonds, Rezek and Reed, 2011), the leader in the field of cloud security, the largest identified cloud security problem is related to the shared technology issues. Infrastructure resource sharing can potentially allow one consumer to peek into another consumer's data if the system does not provide strong system for authorization and authentication. Here, the problems of account hijacking or user credential theft are also relevant. The traditional identity management (IdM) approach is more centralized compared to the current solution used in cloud computing, and usually is based on user personal data, such as real name, user name, e-mail address, identification number, access permissions, etc. The use of a separate IdM system within an organisation, and its connection with the cloud is quite complicated, and there is no simple way to extend its use to the cloud (Lonea, Tianfield and Popescu, 2013). In order this to become part of the cloud system the following actions and the corresponding implementations are required (Cantor, Kemp, Philpott and Maler, 2005):

- Registration of identities: Verification of a user account is needed before proceeding with the registration in accordance with the relevant security standards. Organizations that transfer their user accounts to the cloud must make sure their user account management system is up-to-date and safe.
- Authentication: Management and implementation of the user authentication must be performed in a trustworthy way. The IdM systems should allow configuration of the authentication systems. Another important property of the system should be the cloud-providers' identification disclosure to third party providers and the use of authentication by both parties.
- Federation of identities: Federation of identities allows users to use the same set of credentials to obtain access to different resources. User`s electronic identity and attributes are securely shared across multiple IdM systems. Federation of identities can be achieved in several ways; e.g., based on SAML (Cantor, Kemp, Philpott and Maler, 2005) or the OpenID solution (Ferg, Fitzpatrick and Howells, 2007). One of the important properties is the required compatibility of cloud providers' IdM systems.
- Authorization: Authorization specifies the rights of individual user accounts. It is important for the cloud the account management procedures to be set up and for the rights verification from the highest system authority. Furthermore, the granted right should be consistent with the policy.
- Access control: Access control requirements vary widely according to the type of the end-user (an individual or an organization). In order to implement an access control system, an access control policy must be set, and its implementation should allow the performed actions to be traceable.

Apart from these recommendations it becomes obvious that an optimal system for ensuring authentication and authorization in a multi-platform cloud system would also benefit from the single sign-on approach principles. The Open Group defines SSO as "a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where that user has access permission, without the need to enter multiple passwords" (Group TO, 2014). The most important security property that the SSO principle brings is the common secure infrastructure, which can be carefully managed and protected (Andronache and Nisipasiu, 2011). At the same time, when using the SSO approach, the cloud services do not have to manage each user account. Account management can be carried out by a central authentication system.

Although the concepts of identity federation and single sign-on are well known in the cloud research literature there are still some practical design and implementation issues worth discussing, especially in the multi-platform cloud environments, cloud federations, and inter clouds. General proposed solutions for a multi-platform environment often do not take into account the current implementation status, functionalities and openness of the available cloud platforms, as well as not take advantage of their existing services. The papers also do not give enough details of the proposed solutions and their integration into existing platforms.

Panarello et al., for example, analyse requirements for IaaS cloud federation (Panarello,

Celesti, Fazio, Villari and Puliafito, 2014). While they envisage SAML, OpenID and Shibboleth as an authentication solution, only a standard and general identity federation model is described without details and without taking into account the platform implementation specifics, such as current missing SAML support in OpenStack and Hyper-V. They also highlight only one type (OpenQRM) of cloud platforms.

Different initiatives, projects and libraries on regulation technologies for the Inter-Cloud environment are covered in (Grozev and Buyya, 2014), (Toosi, Calheiros and Buyya, 2014). Libraries, for example JCloud (java library), Apache LibCloud (python library), or Apache DeltaCloud (ruby library), and projects such as InterCloud, OPTIMIS, mOSAIC, STRATOS, Contrail, have been designed to abstract the programmers from the differences in the management APIs of clouds and to provide control over resource provisioning. The EU Contrail project, for example, provides support for SAML and oAuth in their latest version. However, this is still not a practical solution for various cloud platforms, as they are closed and don't have built-in support for SAML yet (e.g. Hyper-V).

In (Abdo, Demerjian, Chaouchi, Barbar and Pujolle, 2013) Abdo et al. discuss the cross-cloud federation manager and propose a centralised broker-based approach. A new entity named "broker" is similar to the centralised solution described in this paper, but has other goals (change of discovery and match making).

3 THE APPROACH TAKEN IN THE SOLUTION DESIGN

The multi-platform cloud infrastructure consisted of the OpenStack and VMware cloud platforms, two of the most known and widely used platforms for provision of a cloud infrastructure as a service. OpenStack is open source software for the construction of private and public clouds. It uses a role-based access control (RBAC) mechanism to manage accesses to its resources (Ferraiolo et al., 2001). The identity service in OpenStack is named Keystone. The service authenticates users and provides them with authorization tokens that can be used for accessing the OpenStack services. The current version of Keystone is centralized, and all its users need to be registered in the Keystone database.

The other platform, VMware is considered as one of the most feature-completed platforms in the field. VMware vCloud Director is a cloud platform software solution that enables enterprises to build secure, multi-tenant private clouds by pooling infrastructural resources into the virtual data centres, and exposing them to users through the web-based API and REST interfaces as fully automated, catalogue-based services. The vCloud Director also supports RBAC. The authentication method used in vCenter that helps automate VMware vCloud Director and other virtualization management system processes is called “Share a unique session”. vCenter uses a single set of credentials to enable connection to the vCloud. The latest release 5.5 of the vCenter application has introduced a new SSO approach capability, which allows users to log in just once, and obtain the valuated authentication for all vCloud’s components. The vCloud API uses basic HTTP authentication, which enables clients to obtain authentication tokens.

By considering these characteristic the solution for enabling a central SSO facility for both platforms was designed with an aim to provide flexible and secure authentication and authorization service for both platforms. The solution is briefly described in the next section. It was named Common Authentication Solution for multi-platform cloud or shortly CAS.

4 COMMON AUTHENTICATION SOLUTION FOR MULTI-PLATFORM CLOUD SERVICE

4.1 Functional Description

The main objective followed during the development of the CAS solution was to enable the cloud infrastructure administrators and the other users to access heterogeneous cloud’s infrastructure services using just a single credential. The solution takes into account cloud platforms’ specifics, as well as the services and APIs already offered by those platforms, and extends its usage to authorization and user access rights management. Other objectives that were followed during the development were to enable:

- better user experience; users should be able to move between services securely and uninterrupted,
- one dashboard for interacting with different cloud SPs,
- reduction of the processing costs, obtained by reducing the number of calls,

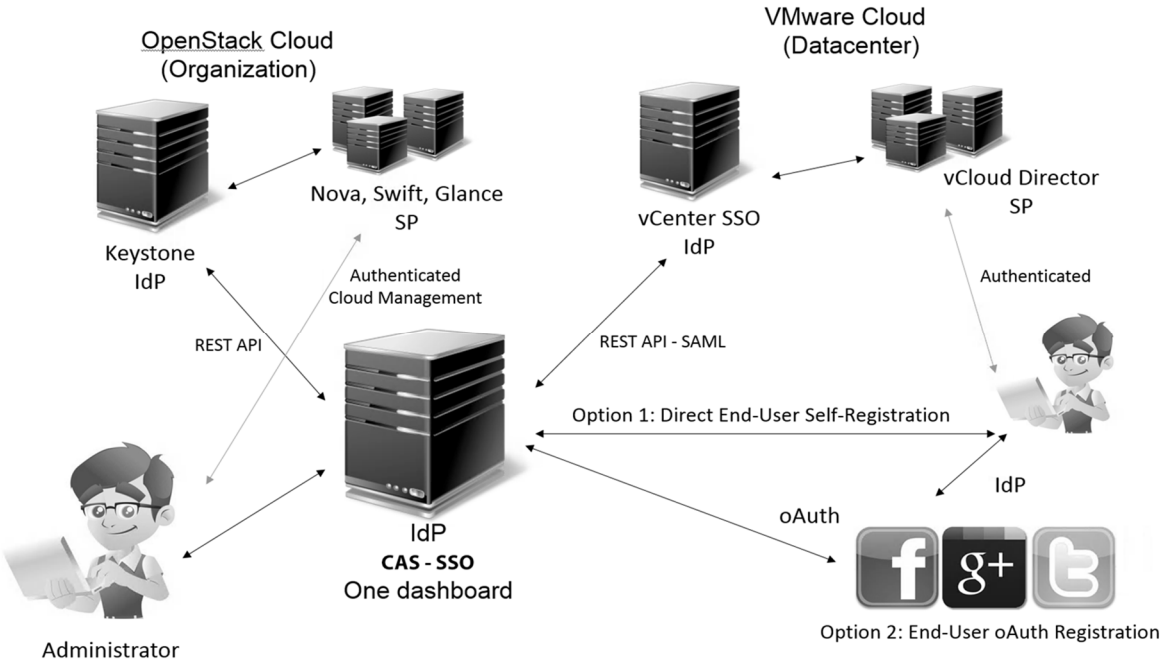

Figure 1: The CAS solution scheme.

- significantly shorter time required for management of multiple accounts,
- higher level of the information security,
- provision of an audit log of the user operations and actions.

By following these objectives CAS becomes a favourable solution that enables administrators, employees, consumers, and customers to access clouds and their services with a single credential. In order to illustrate how CAS works, we are using here the example of an organisation (e.g. corporation or university) with several different cloud infrastructures on site or at a dislocated facility. The infrastructure consists of different platforms for each of the cloud, e.g. OpenStack and VMware. CAS is located on the organisation site, and allows the site administrator to add multiple cloud platforms, either inside or outside the organisation. The CAS application contains a database with the end users of the organisation enabling the site administrator to manage the allocation of resources and, the end users applications in the different system platforms. The end users need only one login. A user logs into the CAS system and he finds there virtual machines assigned to different cloud platforms. Therefore, the federated approach and SSO in the CAS allows the successfully authenticated end users to access additional cloud platforms and services without re-authenticating, since the relationship of trust had already been established and was provided by the CAS system.

The basic operational principles implemented in the system are shown in Fig. 1. The system is split in two parts – front end and back end. The first part of the system is a web interface, built with the help of the programming languages PHP, HTML, and JavaScript. This interface enables end users to sign into a central system with a browser and gain access to all assigned cloud platforms. An administrator assigns permissions to end-users, and manages the access to different cloud platforms through the web browser. The back end of CAS is responsible for mapping, synchronizing and removing end users from the remote cloud platforms. It uses REST access calls (REST over HTTP protocol) for communication with a remote terminal. The format of the exchanged data depends on the end cloud platform accessed (JSON and XML).

CAS provides two ways of end-user registration; the first option is self-registration, and the second one is registration through a social network account (oAuth2). The oAuth2 solution, which has become a popular open standard for authorization supported by Google, Facebook, Twitter, and Yahoo was implemented to enable another simple access to the end users. It was designed for web applications with servers that store confidential information, maintain, state, and provide a secure way for an application like CAS without requiring usernames and passwords (Oracle, 2013). In addition to the oAuth and self-registration, CAS supports also LDAP or AD integration.

Before the CAS administrator can manage end-users, an instance of the remote cloud platform must be created in the central system. Figure 2 shows the CAS administrator dashboard, enabling the administrator to easily add any cloud platform that becomes part of the CAS. First, the administrator selects the type of the platform and triggers the generation of the extra required data fields for the selected platform type. These fields are needed for creation of the local instance of the remote platform. In the case of OpenStack, the fields that belong to the remote administrator, such as password, token and tenant, are required to be filled with relevant data. Once the cloud platform is added, the administrator can import users from remote cloud platforms or add them to the remote cloud platform on the CAS administrator dashboard.

The back end part of the system automatically adds a new end-user with the help of the REST calls. In case of OpenStack, the REST calls are used to insert end-users into the Keystone application. A more detailed explanation of the end-user mapping by the central system in the cloud platform can be found in (Cigoj, 2014).

An end user logs into CAS by providing his CAS login credentials. After a successful login, the user is presented with the dashboard of remote platforms that are accessible to him with a single click on the login button without a username or a password. The end user can then create, run and power off virtual instances.

4.2 Implementation

The implementation of the CAS was based on the available APIs of the cloud platforms, known applications, such as CURL, REST, HTTPS, and the Python programming languages.

For the OpenStack platform PHP scripts with the support of CURL (a command line tool for transferring data) were used to ensure the transfer of end users from CAS into the remote OpenStack and to automate common end user and administrative tasks in combination with the CAS. The OpenStack authentication service provided with the Keystone`s API interface which is a component that allows the administrators to manage and map end users into the remote Keystone database was used for enabling the end-user authentication process between the CAS and the remote OpenStack platform. The Keystone Identity Service enables clients to obtain tokens for access to the OpenStack cloud services. The Keystone API is using the RESTful web service interface where all authentication and operation requests directed to the Keystone API are performed with the SSL protocol over the HTTP (HTTPS). Since Keystone code can be executed by use of the HTTP sessions, the external authentication methods were

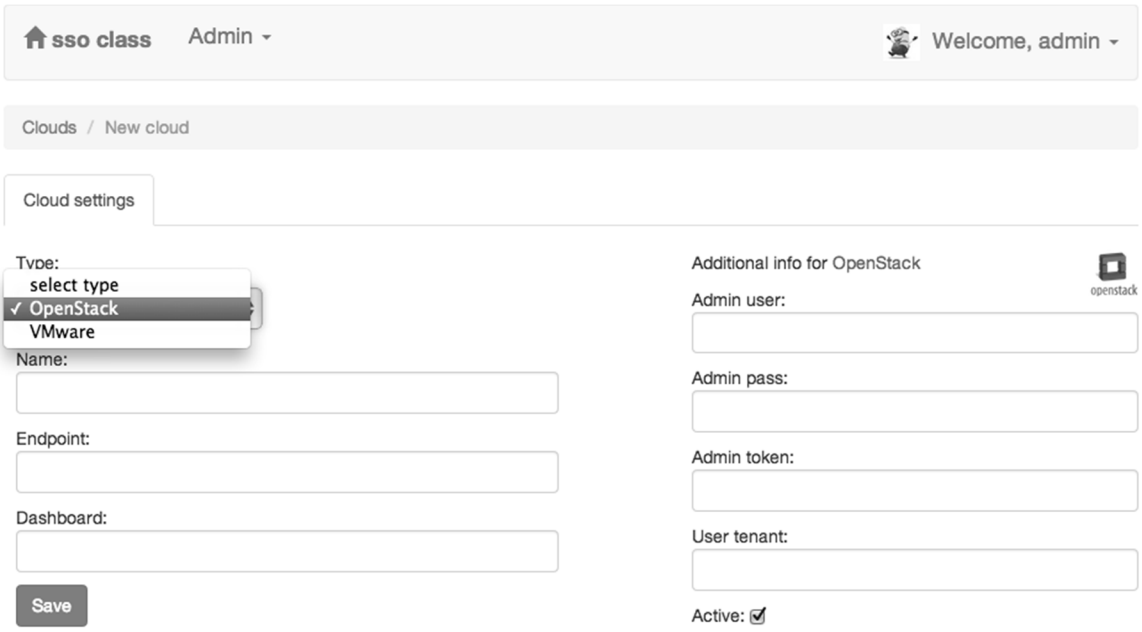

Figure 2: CAS administrative dashboard.

applied. The Keystone SQL identity back end facility was used together within the CAS for mapping the end users credentials only once and for login them into the OpenStack dashboard. Their credentials are supplied to CAS only once. The Keystone API supports both JSON and XML data serialization formats, the response format uses JSON by default. This feature was used in the CAS and the X-Auth-Token is accompanied with the URLs of the other services in the cloud.

The vCloud connection with the CAS is similar to the solution applied in the OpenStack platform. The vCloud APIs provide rich functionalities for the management part of the vCloud platform, the vCloud Director. Two methods were possible to be used for the interaction of the vCloud Director cells with the CAS: through a web browser UI or through the vCloud API. The vCloud APIs are RESTful- based, they are highly scalable, and use the HTTP or HTTPS protocol for communication. As the vCloud directory contains a set of APIs for vCloud's provisioning and management controls the programming of the necessary extensions was relatively easy. The returned objects from the API follow the XML scheme where the properties are represented as elements, and the object values as element attributes.

The prototype code that works across the OpenStack and vCloud platforms is offered as an open source code (https://github.com/primozc/sso).

4.3 User Migration and Federation

During the development of CAS a special attention was given to the migration of the users from the CAS to the cloud platform. An end user who is migrated to the cloud and authenticated at CAS is able to access the cloud platform assigned to him by an administrator. As all users are registered and authenticated in the CAS, the management of the end users in remote identity service is not necessary anymore.

API operations are performed to map the end users in the local CAS database to the OpenStack Keystone and the vCloud user database. These operations provided by CAS enable administrators to obtain and validate access tokens, manage users, tenants, roles, and service endpoints. The administrative API calls against services require authentication; the calls to discover services are the only ones without authentication.

An administration token (token for API calls) is used for various administrative operations, such as the integration of the OpenStack with the CAS or user mapping. The Keystone Identity API verifies the issued administration token and defines the administration role. Administration tokens are stored in the local CAS database system in the process of adding a new cloud platform. A set of identity attributes, such as email, username and tenant, are additionally inserted into the OpenStack Keystone API when the migration of the users into a cloud platform is performed

The connection between the vCloud component of vCloud platform and CAS is similar to the one described above. CAS connects a user to vCloud on the user's request by an API call where user credentials are passed as parameters. Since vCloud supports SAML, the open source SimpleSAMLphp library, which implements the SAML 2.0 standard, was used for exchange of user messages between CAS and the remote vCloud platform. In CAS an IdP was defined according to the vCloud Organization Federation Settings. End users, user groups' data and their roles in vCloud are required to be mapped from the organization's database and from the organization's SAML provider. This restriction required additional functionality in CAS to be developed for provision of the data mapping stored in the vCloud Director database. As SimpleSAMLphp application does not support dynamic generation of metadata in an SP's remote configuration file, an additional upgrading code was developed. The generated XML metadata file from the CAS IdP contains several certificates and information (e.g., SingleLogoutService or AssertionConsumerService) which are necessary by the vCloud Director to be able to communicate with the CAS IdP and to validate if it is sufficiently trustworthy. The generated XML metadata file from the CAS IdP needs to be uploaded into the vCloud federation metadata XML form (VMware, 2012). Since SAML users and groups cannot be found by use of a search function, user's data have to be mapped into vCloud with an automated API call. Adding this feature to the SSO the vCloud Director integration became complete.

5 DISCUSION AND CONCLUDING REMARKS

The security threats in cloud computing can be removed to a big extent with the system protection capable to resist the attacks. When business entity networks are migrated to the cloud, their data and systems are no longer isolated, they share resources with many other organizations, and this new status is becoming much more attractive for malicious

attackers. In the cloud computing core technologies identified vulnerabilities are mainly related to the virtual machine escape, poor password protection, poor authentication and authorization systems, session hijacking, and insecure cryptographic algorithms. The presented work in this paper can be considered as an attempt to remove some of these vulnerabilities that are related to the user authentication and authorization.

Authentication and authorization of the OpenStack and VMware platforms were carefully studied before a solution enabling a federated IdM approach to be applied in cloud computing environment emerged. The developed solution, the Common Authentication Solution connects both platforms and enables secure authentication service and friendly remote user management. CAS has been implemented for integrating the user authentication of the addressed platforms, but it is sufficiently general to be used for other environment as well. The Microsoft cloud solution already offers some options. A SOAP library (Ruby library), for example, can be used to use the functionality in Windows Remote Management (WinRM) to call native object in Windows. This includes, but is not limited to, running batch scripts, powershell scripts and fetching WMI variables. This way, we can communicate and map users between CAS and the Hyper-V cloud. Another popular open source cloud platform OpenNebula contains a patch in the authentication system, and two standard SimpleSAMLphp modules that can be used to establish connection between CAS and OpenNebula. Furthermore, Eucalyptus and CloudStack are still missing the SAML support in their authentication system, but their aim is to integrate the SSO SAML support. Despite the lack of SAML support there can be a patch developed to support this feature. It is necessary to reiterate at this point that our aim was to provide a unified interface for many other well-known cloud providers and provide simple integration of our platform with other IaaS platforms.

Acting as a kind of a broker, CAS introduces only a slight overhead (login to CAS) to the multi-platform cloud operation from the user point of view when only one platform is accessed. On the other hand CAS relieves the user from frustration of having to remember multiple passwords and enables him easier access to multiple cloud platforms. CAS functionality improves administration performance by providing one interface to manage multiple cloud platforms. The amount of time spent for logging on to different cloud platforms is reduced and it provides faster access to the resources.

Cloud computing still needs much more development and deployment for provision of secure and trustworthy services. The future development is planned to be oriented towards provision of a unified access point for many other well-known cloud providers such as Amazon, DigitalOcean, Slicehost, or Rackspace. For this reason, our future work is oriented to the extension of the functionality of the CAS system in order to support other features that are common to different cloud providers and platforms, such as management of a cloud network, virtual machine, image and storage.

REFERENCES

Abdo, J. B., Demerjian, J., Chaouchi, H., Barbar, K., & Pujolle, G. (2013). Broker-Based Cross-Cloud Federation Manager. In Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for (pp. 244-251). IEEE.

Andronache I., Nisipasiu C., 2011. Web single sign-on implementation using the simpleSAMLphp application. Journal of Mobile, Embedded and Distributed Systems. 3(1):21-9.

Cantor S., Kemp I.J., Philpott N.R., Maler E., 2005. Assertions and protocols for the oasis security assertion markup language. OASIS Standard.

Cigoj P., 2014. Cloud computing security and identity management in the OpenStack platform. Ljubljana: Jožef Stefan International Postgraduate School.

Cruz Zapata, B., Fernández-Alemán, J.L., & Toval, A. (2014). Security in Cloud Computing: a Mapping Study. Computer Science and Information Systems 12(1):161–184.

Ferg B., Fitzpatrick B., Howells C., Recordon D., Hardt D., Reed D., et al. 2007. OpenID authentication 2.0.

Fernandes, D.A.B., Soares, L.F.B, Gomes, J.V., Freire, M.M., & Inácio, P.R.M., 2014. Security issues in cloud environments: a survey. International Journal of Information Security, vol. 13, iss. 2, pp. 113-170.

Ferraiolo D.F., Sandhu R., Gavrila S., Kuhn D.R., Chandramouli R., 2001. Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC). 4(3):224-74.

Group TO, 2014. Single Sign On. Available from: http://www.opengroup.org/security/sso/.

Grozev, N., & Buyya, R. (2014). Inter□Cloud architectures and application brokering: taxonomy and survey. Software: Practice and Experience, 44(3), 369-390.

Jansen, W., Grance, T., 2011. Guidelines on security and privacy in public cloud computing. NIST special publication. 800:144.

Lonea A.M., Tianfield H., Popescu D.E., 2003. Identity management for cloud computing. New Concepts and Applications in Soft Computing: Springer. 175-99.

Microsoft. Microsoft Urges Government and Industry to Work Together to Build Confidence in the Cloud 2010. Available from: http://www.microsoft.com/en-us/news/press/2010/jan10/1-20brookingspr.aspx.

Oracle, 2013. Oracle Access Management OAuth Service 2013. Available from: http://www.oracle.com/technetwork/middleware/id-mgmt/overview/oauthservice white paper-2110557.pdf.

Panarello, A., Celesti, A., Fazio, M., Villari, M., & Puliafito, A. (2014). A Requirements Analysis for IaaS Cloud Federation. In 4th International Conference on Cloud Computing and Services Science, Barcelona, Spain.

Pérez-Méndez, A., Pereniguez-Garcia, F., Marin-Lopez, R., López-Millán, G., & Howlett, J. (2014). Identity Federations Beyond the Web: A survey. IEEE Communications Surveys & Tutorials, Vol. 16, No. 4.

Simmonds, P., Rezek, C., Reed, A., 2011. Security guidance for critical areas of focus in cloud computing v3.0. Cloud Security Alliance. 176 pages.

Tripathi, A., Mishra, A. (2011). Cloud computing security considerations. In: IEEE International Conference on Signal Processing, Communications and Computing, pp. 1–5.

Toosi, A. N., Calheiros, R. N., & Buyya, R. (2014). Interconnected cloud computing environments: Challenges, taxonomy, and survey. ACM Computing Surveys (CSUR), 47(1), 7.

VMware, 2012. vCloud director user's guide, 2012. Available from: http://pubs.vmware.com/vcd-51/topic/com.vmware.ICbase/PDF/vcd_51_users_guide.pdf.