



Original software publication

Collaboration between SAML federations and OpenStack clouds

Mihály Héder^{a,*}, Szabolcs Tenczer^a, Andrea Biancini^b
^a MTA SZTAKI Institute for Computer Science and Control, Kende u. 13-17, Budapest, 1111, Hungary

^b RETI Institute, Via Dante 6, 21052 Busto Arsizio (VA), Italy


ARTICLE INFO

Article history:

Received 23 January 2017

Received in revised form 9 November 2018

Accepted 13 December 2018

Keywords:

SAML

OpenStack

Research loud

eduGAIN

ABSTRACT

In this paper, we present a novel OpenStack module called **regsite** for enabling easy access for researchers to OpenStack research clouds. Many researchers have an account in an Academic AAI federation, such as national research and education federations or the eduGAIN SAML meta-federation. The software solution presented here makes it possible to use these institutional accounts together with so-called virtual organization managers for authenticating and authorizing in OpenStack instances in a clean and secure way. An analysis of earlier generations of OpenStack-related developments trying to tackle the same problem is given. Many aspects of this software integration can be generalized to serve as a template for federative research cloud access.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

Current code version
 Permanent link to code/repository used of this code version
 Legal Code License
 Code versioning system used
 Software code languages, tools, and services used
 Compilation requirements, operating environments & dependencies
 If available Link to developer documentation/manual
 Support email for questions

V1.0
<https://github.com/ElsevierSoftwareX/SOFTX-D-17-00009>
 Apache 2.0
 Git
 Python, Django, html, OpenStack Keystone service
 The software does not require compilation. It can be operated on OpenStack KILO or newer, with Keystone API v3.0 or newer.
<https://github.com/burgosz/openstack-horizon-shibboleth/blob/master/README.md>
Mihaly.heder@sztaki.mta.hu

1. Motivation and significance

The SAML protocol, which is implemented by national research and education federations and the eduGAIN [1] meta-federation is the de-facto standard for authentication and authorization in the global research community. The majority of nations of the Americas, the majority of EU countries and many nations in Asia are all part of eduGAIN. In these federations user accounts are maintained by higher education and research institutes, meaning that a higher level of confidence is achieved than with self-registered authentication sources like social media. Expensive High Performance Computing, Grid and Cloud resources as well as Journal Subscriptions

are being provisioned to users based on this technology. Therefore, it is important all tools that are used by researchers can be integrated with SAML. This article discusses a novel, clean solution of SAML integration for OpenStack Cloud, which is increasingly becoming the most used cloud technology in the research scene. The result of our work is an OpenStack module called **regsite** and an overall OpenStack deployment layout we suggest in this article.

Regsite was developed to achieve a certain set of desirable engineering properties lacked by earlier solutions. One of these was encapsulation [2] of new functionality within a self-contained module. This was especially important, because the alternative – implementing the functionality via source code patches to OpenStack – requires an update of the patches every time the host code changes. In the case of OpenStack's half-year release cycle, the necessary changes would have been frequent. Also, the merging of the functionality to one of the OpenStack mainline components would

* Corresponding author.

E-mail addresses: mihaly.heder@sztaki.mta.hu (M. Héder), tenczer.szabolcs@sztaki.mta.hu (S. Tenczer), andrea.biancini@reti.it (A. Biancini).

have undermined its modularity, and the OpenStack developers we contacted also advised against it.

Reuse of mature components [3] was also targeted for development. This mostly involved the reuse of SAML-related software components. The handling of SAML protocol and metadata requires complex logic, and since these components implement authentication, the correct design and implementation is critical to security. As a result, the development of any new SAML-related code was avoided.

Full compatibility with SAML federations was set as a goal. This resulted in federated login, logout, the latest metadata defining the IdPs, metadata refresh, and the use of Virtual Organization Management Systems supporting SAML Attribute Authority interface, even multiple ones, in the same session. Compliance with legal requirements (such as the need of informed consent of attribute release) was also essential.

Delegation of administration [4], in this case, user provisioning and authorization by external systems, was also achieved. SAML federations have a number of solutions for Virtual Organization and Virtual Group management [5] that can be relied on. All of the above properties helped contribute to our overarching goal of easy, long-term operation and maintenance.

1.1. A short description of SAML and OpenStack

SAML [6] identity and attribute federations are common in the research and education environment. They allow users to use their home institution credentials to access resources at partner institutions. This is achieved through the exchange of digitally signed XML [7] assertions. There are three major roles in mature SAML federations: service providers (SPs) [8], identity providers (IdPs) [9], and attribute authorities (AAs) [10]. SPs provide resources for users. In this case study, an OpenStack [11] cloud is such a resource, with OpenStack as the SP. IdPs are sources of user identity information that the SPs and AAs trust. Attribute authorities (AAs) are sources of user attributes that SPs trust. While user data in IdPs are managed by the HR department of institutes and by the administrative departments of universities and colleges, groups and virtual organizations are managed by inter-organization collaborations. Trust between these entities is pre-established through the exchange of signed metadata [12] that contains signing keys, trusted network endpoints and administrative information.

OpenStack is an open source, Infrastructure-as-a-Service (IAAS) [13] cloud system that is designed to be modular. The focus of this paper is on the authentication and authorization functionalities of the OpenStack system, which are designed to be highly configurable and extensible. Initially, OpenStack did not support SAML federations. There have been a number of previous integration efforts (detailed in Related work) trying to resolve issues in this area. The two modules involved in the authentication and authorization process are Keystone [14], OpenStack's authentication component, and Horizon [15], the system's web interface.

1.2. Related work

(A) In 2012, the University of Kent initiated a project to SAML-enable Keystone and OpenStack [16,17]. One drawback of this pioneering solution was that it did not include the reuse of existing middleware to handle the SAML protocol, but instead, it relied on SAML programming libraries to implement its own SAML functionality. As a result, it did not achieve full SAML compatibility: it did not handle external attribute authorities, and it did not consume SAML metadata. Instead, OpenStack maintained a list of trusted IdPs in its own database format and consequently the updates

to the metadata did not automatically come to effect in these deployments.

(B) In 2014, as a part of the HEXAA [18] project a new solution was created [19] based on Shibboleth [20] as SAML middleware, and on OpenStack's ability to rely on external authentication modules. The results were presented at OpenStack CEE Day 2015 [20]. The main issue with this solution was that its source code was not encapsulated, but acted as a patch for the main OpenStack codebase, and therefore, would require future significant maintenance.

(C) A completely new solution was included in the 2015–1 Kilo [21] release of OpenStack. The primary assignee was Red Hat, with additional contributors from CERN and IBM [22]. The University of Kent contributed significantly to the Kilo release by replacing the SAML dependent code with protocol independent code. Also the matching rule code was primarily Kent's, taken from their original implementation. The approach was detailed at the OpenStack Cloud Identity Summit (slides 23–38) [23]. The solution is called WebSSO, a protocol-agnostic federation module that works with OpenID [24], SAML, and other protocols. As WebSSO does not include SAML-related code, it makes it possible to encapsulate SAML functionality in a mature SAML middleware component. Shibboleth, mod_shib [25], and the resulting Apache environment is used for authentication. Using the WebSSO solution with Shibboleth achieves full compatibility, reuse of mature components and encapsulation. However, it is not able to create users, tenants and projects within OpenStack. Therefore, either each user must first be created in OpenStack before s/he can login via WebSSO or they need to be ephemeral users. As a result, the WebSSO solution does not fully achieve delegation of administration in permanent user mode, while in ephemeral mode (where no user creation is necessary) CLI is unavailable. The module described in this paper is a converged solution to overcome this problem, based on WebSSO.

(D) There is also Keystone-to-Keystone SAML flow, supported by OpenStack. In this, Keystone acts as an IdP, and another Keystone instance as SP (see the OpenStack Cloud Identity Summit presentation (slide 40) [26]. This solution is not compatible with SAML federations, as it does not consume SAML metadata, and also does not reuse mature components. However, it might be a viable solution for a completely different use case, in which an OpenStack user database is the identity source to be trusted by a federation. This approach allows cloud bursting or outsourcing from a private cloud to a public cloud by enabling OpenStack-to-OpenStack interoperation.

2. Software description

2.1. Software architecture

Our solution required a number of software design decisions to be made based on the design criteria outlined in the introduction. The most important was that OpenStack should be used in combination with mature, in-production (proven on Technology Readiness Level 9 [27]) software, so that proper handling of SAML-level actions were not demanded from OpenStack itself. This includes: (a) metadata handling, as per eduGAIN or other federation requirements, with signature verification, (b) handling of stand-alone AAs, (c) collaboration with discovery services, and (d) SAML single logout. As previously discussed, using OpenStack with mature SAML middleware achieved our goal of reusing mature components.

Moreover, it was important that the new software should not only be a patch to the OpenStack Horizon or Keystone components. Because of the encapsulation of new functionality in its own module, no regular patching of any other OpenStack components will be necessary. Python and Django [28] were selected for consistency with other OpenStack components.

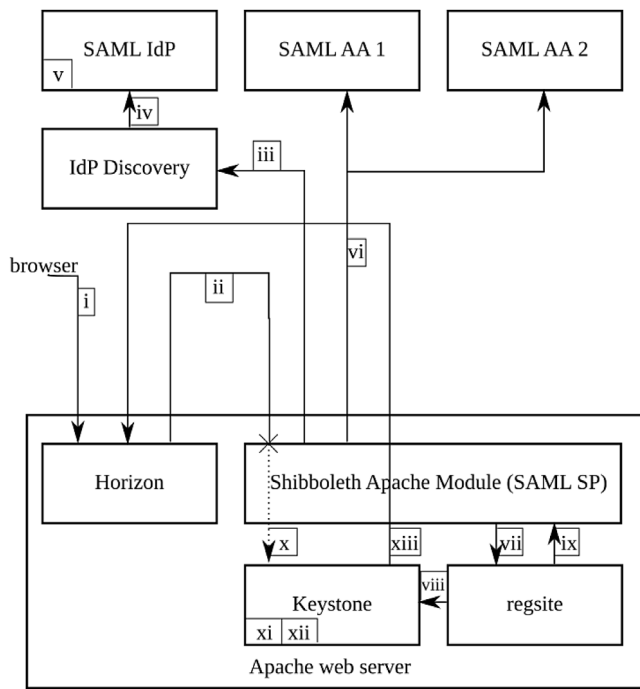


Fig. 1. The architecture and the main workflow of the collaboration between SAML federations and OpenStack. The software solution presented here is an OpenStack module called “regsite”.

It was also key to align the software with the OpenStack Keystone project's vision for the future. At the OpenStack CEE Day 2015 event, a converged solution for the future was agreed upon. From the solution B (the letters refer to the paragraphs in section “Related work” above), the user/project creation part would be separated into a standalone ‘signup page’ web application, using Keystone API calls, and the user/project/tenant created at signup, if necessary. The mainline code would then perform the rest of the authentication and map the SAML session to Keystone users. By 2018 this was achieved via the Shadow mapping [29] capability of OpenStack that regsite leverages. The rest of the B solution would be discontinued, and Horizon would not be patched in the future. It was indicated that the OpenStack Keystone project is open to accept improvement patches to C as long as they do not involve user/tenant creation (or de-provisioning).

The resulting software, **regsite**, in collaboration with WebSSO (see Related work), implements the desirable engineering properties.

2.2. Software functionalities

User provisioning: The solution needed to ensure that the user is always properly provisioned into Keystone before it makes contact with OpenStack. Otherwise, the user would successfully login into Shibboleth federation middleware, but would be denied access and greeted with an error message from Keystone. According to our design criteria, this should not be implemented by adding a patch to an OpenStack module. As a result, we relied on the sessionHook [30] ability of Shibboleth SP instead. The user identifier received from the SAML IdP will be used without modification by OpenStack as username.

Consent: The software also needed to include the means to request user consent since personal data is transmitted from the SAML

federation and stored in OpenStack. This is achieved by a website presented to the user, if necessary.

Role and project provisioning: In the entitlement string sent by the SAML AA for authorization, projects and roles are derived in the following way: “<entitlement_prefix>:project:role”. However, the project and role could be non-existent at the first access. Regsite creates these resources as needed at first access. Additionally, regsite makes it possible to be assigned to multiple projects at the same time.

Password setup for CLI access: Since there is no widespread implementation of non-web SAML access (Moonshot [31] is a solution but not supported by many IdPs and federations), a password is still necessary for using the command line interface of OpenStack. This password cannot be used on the web interface; it is solely intended to be used by scripts.

User authentication and authorization: When all the preliminary steps are done regsite with WebSSO maps the user account information to a local OpenStack user and assigns the user to the roles and projects as described by the entitlement attribute. If the contents of this attribute change the assignments will be re-aligned.

User deprovisioning: When a user's federation memberships are terminated the associated resources – virtual machines, storage, etc. – are not deleted automatically. Regsite provides hooks that may be triggered by the SAML AA or IdP when a user's entitlements are removed or the user itself is deleted. This makes it possible to implement automatic clean-up procedures. With regsite, it is possible to de-enroll from certain project only and this happens at the next login attempt.

Fig. 2 shows the place of these functions (except deprovisioning and CLI password setup).

3. Illustrative examples

The numbered arrows on Fig. 1 show the default workflow of regsite. The inner logic of regsite is depicted in Fig. 2. The following workflow demonstrates how a federated user gets provisioned, authenticated and authorized for at first time access of a regsite-enabled OpenStack instance. The numbering refers to Fig. 1.

The workflow steps are as follows:

(i) The user tries to access the OpenStack Horizon web interface with a web browser.

(ii) Horizon redirects the user to the Keystone component's web endpoint.

(iii) The Keystone component is hosted by an Apache web server and is guarded by a Shibboleth SP. The user does not have a Shibboleth session yet, therefore a SAML login sequence is initiated. The user forwarded to a SAML IdP discovery service, where s/he can select an identity provider.

(iv) The discovery service forwards the user to the IdP.

(v) The user logs in at the identity provider using his/her home institutional credentials.

(vi) Additional profile attributes, and authoritative information is gathered from external attribute authorities, as defined by the SP's configuration. The number of AAs contacted can range from 0 to many, however, the SP sequentially queries the AAs, which aggregates the round-trip times of the single queries. Meanwhile, the user is blocked, which suggests that querying more than five AAs is not practical.

(vii) Shibboleth SP merges and filters the received attributes, then executes its configured sessionHook. It forwards the user to a location hosted on the same server as the SP, which also relays all the attributes gathered during the login process. In sessionHook, Shibboleth SP passes over the identity, profile and authoritative

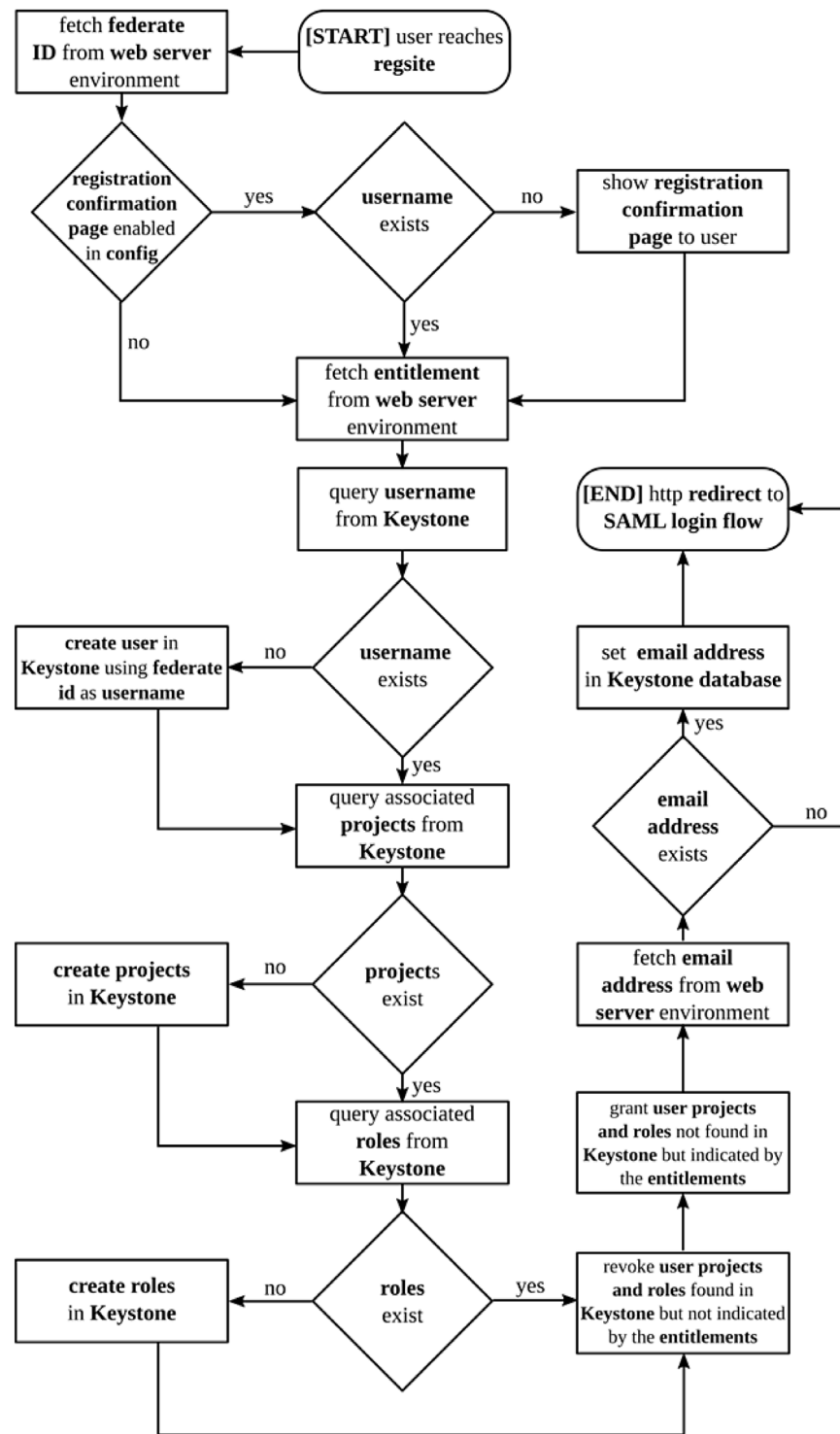


Fig. 2. The flowchart demonstrating some functionalities of regsite.

information to regsite. Steps (iii) to (vii) can all be completed by a standard Shibboleth SP.

(viii) regsite creates the user and the tenant, if necessary, using Keystone API calls.

(ix) regsite directs the user back into the Shibboleth login sequence.

(x) The Shibboleth login sequence finishes, and the user finally reaches Keystone. The same set of information is passed in Apache Environment variables to Keystone, as in Step (vii), to regsite.

(xi) Step (viii) ensures that the user is already existent in Keystone, as well as the tenants they are assigned to, therefore, Keystone successfully authenticates the user.

(xii) Keystone creates a token for the user.

(xiii) Keystone redirects the user to the Horizon web interface, accompanied by the newly created token. Horizon authenticates the user using this token and access is granted.

4. Impact

As the result of our work, an OpenStack deployment installed from the mainline source code or by a deployment tool can now be configured to use eduGAIN or other SAML federation for user authentication and any SAML Attribute Authority for user authorization. No patching or other changes that hinder maintenance are necessary.

Researchers and students can use their home accounts to access the cloud. Authorization can be managed by the given research community that by a Virtual Organization management software that has SAML AA interface (which is the majority now). This means that users and their corresponding roles and resources can be provisioned in an OpenStack installation without the intervention of either the OpenStack administrators or the administrators of the SAML IdP. This removes the two most significant bottlenecks from research cloud access while providing Single Sign-On for the users and the ability to use their home credentials and not having to remember new credentials.

Regsite is free and open source software available for everyone to use. Because of this, the authors cannot know exactly how many deployments use it. However, the two major research and education cloud deployments in Hungary – The “MTA Cloud” [32] of the Hungarian Academy of Sciences and the “C4E” [33] cloud for education project of the KIFU institute – both use it as the exclusive means of cloud access. In MTA Cloud several dozen of national and international physics, chemistry and biology projects use it, while the usage of C4E is increasing by the Higher Education community.

The requests reaching the authors indicate that our software is being tried out by several research-oriented OpenStack deployments worldwide.

5. Conclusions

This paper presented a new method of collaboration between OpenStack cloud systems and mature SAML federations. Such collaboration was made possible by the modular design of OpenStack, which supports customized authentication and authorization, and by the generic nature of Shibboleth SAML middleware.

The regsite module exhibits key engineering properties essential for long-term operation of deployed systems and maintenance of regsite code. Encapsulation of the new functionality necessary for the integration of OpenStack cloud systems and mature SAML federations is key for source code maintenance, while the reuse of mature components helps to minimize the size of that source code. Full compatibility with SAML ensures that the services provided by the federation – i.e. metadata distribution, discovery services, single login, and single logout – can be used. The fact that there is a very loose coupling between the systems, e.g. Shibboleth hides all SAML-related actions from OpenStack, and regsite relies on a small portion of Keystone API, ensures that the inevitable evolution of SAML federations and OpenStack will not endanger the easy maintenance of the collaboration.

Acknowledgments

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation program under Grant Agreement No. 691567 (GN4-1). The authors are deeply indebted to the whole GN4-1 Joint Research Activity 3 (JRA3) Team, especially: Kristóf Bajnok, Maarten Kremers, Alejandro Perez Mendez, Remco Poortinga – van Wijnen and Michal Procházka. Many thanks also to Morgan Fainberg, for his valuable guidance in OpenStack Identity Issues.

References

- [1] Howlett Josh, Nordh V, Singer W. “Deliverable DS3. 3.1: eduGAIN service definition and policy Initial Draft.” Project Deliverable, May 2010.
- [2] Meyer B. *Object-oriented software construction*. 2nd ed. New Jersey: Prentice Hall; 1997, p. 53.
- [3] Meyer B. *Object-oriented software construction*. 2nd ed. New Jersey: Prentice Hall; 1997, p. 67.
- [4] Bajnok K, Héder M, Magyar Z, Tétényi I. “HEXAA: Higher education external attribute authority” *Connect Magazine* 18 14–15, http://issuu.com/danteprm/docs/connect_issue_18_web/17?e=6131560/11460567.
- [5] Tétényi I, Héder M, Magyar Z, Bajnok K. Open call deliverable OCK-DS1.1 final report (HEXAA), GÉANT, 2013, p. 5, http://www.geant.net/Resources/Open_Call_deliverables/Documents/HEXAA_final_report.pdf.
- [6] OASIS security services technical committee, security assertion markup language. 2005, <http://docs.oasis-open.org/security/saml/v20/saml-20-os.zip>.
- [7] Bartel M, Boyer J, Fox B, LaMacchia B, Simon E. XML signature syntax and processing. 2nd ed. 2008, <http://www.w3.org/TR/xmlsig-core/>.
- [8] OASIS Security Services Technical Committee, Glossary for the OASIS Security Assertion Markup Language (SAML) v 2.0. 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf> (line 361).
- [9] OASIS Security Services Technical Committee, Glossary for the OASIS Security Assertion Markup Language (SAML) v 2.0. 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf> (line 201).
- [10] OASIS Security Services Technical Committee, Glossary for the OASIS Security Assertion Markup Language (SAML) v 2.0. 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf> (line 146).
- [11] The OpenStack Foundation. Open source software for creating private and public clouds. 2015, <http://docs.openstack.org/>.
- [12] OASIS Security Services Technical Committee, Metadata for the OASIS Security Assertion Markup Language (SAML) v 2.0. 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf> (line 1158).
- [13] Ali M, Khan S, Vasilakos A. Security in cloud computing: Opportunities and challenges. *Inform Sci* 2015;305:357–83, p. 360, <http://dx.doi.org/10.1016/j.ins.2015.01.025>.
- [14] The OpenStack Foundation. OpenStack Identity. 2018, <https://docs.openstack.org/keystone/latest/admin/identity-concepts.html>.
- [15] The OpenStack Foundation. OpenStack Dashboard. 2018, <https://docs.openstack.org/horizon/latest/>.
- [16] Germonville D, Fouillat Y, Chadwick D. Adding federated access to OpenStack. 2012, <https://dl.dropboxusercontent.com/u/44986510/Adding%20federated%20access%20to%20OpenStack%201.pdf>.
- [17] Chadwick D. Adding federated identity management to OpenStack, The OpenStack summit. 2012, <https://www.openstack.org/summit/san-diego-2012/openstack-summit-sessions/presentation/adding-federated-identity-management-to-openstack>.
- [18] Hungarian Academy of Sciences Institute for Computer Science and Control and National Information Infrastructure Development Institute, HEXAA project site. 2018, <https://hexaa.eu>.
- [19] Tenczer S. Shibboleth authentication backend for Horizon. 2018, <https://github.com/burgosz/openstack-horizon-shibboleth>.
- [20] Tenczer S, Héder M. OpenStack SAML Integration with HEXAA, OpenStack CEE Day. 2015, <http://openstack.hexaa.eu/>.
- [21] The OpenStack Foundation. OpenStack 2015.1.0 (Kilo) Release Notes. 2015, <https://wiki.openstack.org/wiki/ReleaseNotes/Kilo>.
- [22] Young A, Martinelli S, Denis M, Leon JC, Tran T. Web single sign on portal. 2018, <https://docs.openstack.org/keystone/latest/advanced-topics/federation/websso.html>.
- [23] Martinelli S. Building iam for openstack, openstack cloud identity summit. 2015, <http://www.slideshare.net/SteveMartinelli1/building-iam-for-openstack>.
- [24] David R, Reed D, Openl D. 2.0: a platform for user-centric identity management. In: *Proceedings of the Second ACM Workshop on Digital Identity Management*. ACM; 2006.
- [25] Cantor S, Carmody S, Erdos M, Hazelton K, Hoehn W, Morgan RLB, Scavo T, Wasley D. Shibboleth architecture. 2005, <https://wiki.shibboleth.net/confluence/download/attachments/2162702/internet2-mace-shibboleth-arch-protocols-200509.pdf>.
- [26] The OpenStack Federation. Federated Keystone. 2018, <https://docs.openstack.org/security-guide/identity/federated-keystone.html>.
- [27] Héder M. From NASA to EU: the evolution of the TRL scale in public sector innovation. *Innov J* 2017;22(2):1–23.
- [28] Django Software Foundation. The web framework for perfectionists with deadlines. 2015, <https://www.djangoproject.com/>.
- [29] The OpenStack Federation. Shadow mapping. 2018, <https://specs.openstack.org/openstack/keystone-specs/specs/keystone/ocata/shadow-mapping.html>.
- [30] Native shibboleth SP application. 2005, <https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApplication>.
- [31] Project Moonshot. <https://www.jisc.ac.uk/rd/projects/moonshot>.
- [32] MTA Cloud. <https://cloud.mta.hu>.
- [33] C4E Cloud. <https://c4e.niif.hu>.