Single Sign-On Implementation for the Applications Running on Cloud Platform using Open Source Technologies

[*1]Serdar Arslan and [2]Engin Dağdeviren
[*1]KoçSistem Information Communication Services Inc.
[2]Kavi Information and Electronics Inc.

Abstract

This paper aims to explain recently designed architecture for a Single Sign-On (SSO) implementation for the applications running on cloud environment. Single Sign-On is a feature of access control of both related and unrelated applications that have trust relationship. The main purpose of SSO is to allow users to sign-on, and gain access without being further prompted for credentials each application. Proposed architecture provide solution for SSO and IDM matters using open source technologies along with the real life implementation of defined architecture on EU FP7 cloud project [1].

Key words: Single Sign-On, OpenID, OAuth, Keycloak, Security on Cloud

1. Introduction

Until recently, most of the internet applications are designed to run on single server model and used password authentication to allow access to applications meaning that users need to enter credentials to gain access to the specified system. This also means that users need to remember their credentials for each web application that they want to have an access. Recent developments starting allowing us central management for users and their credentials. This architecture is designed for any project that targets to provide a platform for various applications running on cloud environments. Although these applications run independently, for such platforms it is important to provide a trust relationship between applications and the platform, allowing users to sign-on and gain access without being further prompted for credentials for each application. Proposed architecture is implemented and currently is used on EU FP7 cloud project [1] and platform currently provides solution for Single Sign-On and central management of users. Mentioned cloud environment is created using on eight servers, one being controller for OpenStack, which is an open source cloud computing software. OpenStack is used as a base of Cloud-hosting infrastructure. This platform provides control over pools of processing, storage, and networking resources throughout a web-based data center. Servers and resources will be easily managed for increasing computation power for running necessary algorithms and for executing server side communication protocols [2]. Core modules of the platform and the applications are located on virtual machines (VM) created on this environment.

2. Materials and Method

Architecture consist of open source technologies currently used by numerous platforms. Proposed

*Corresponding author: Address: KoçSistem Information Communication Services Inc. Unalan Mah. Ayazma Cad. Camlica Is Merkezi B3 Blok Uskudar Istanbul TURKEY. E-mail address: serdar.arslan1@kocsistem.com.tr, Phone: +905327001012 Fax: +902162171910

architecture aims to provide solution for Single Sign-On and Single Log Out [3] for browser applications using OpenID Connect as well as authorization, Social Login, user registration, OAuth bearer token and grant request functionalities in addition to JavaScript Web Token (JWT) using OAuth 2.0. Current implementation of the architecture is succeeded using these listed technologies and methods. Although over solution targeted single domain, this solution can be applied to multiple domains [4].
Keycloak [5] is an SSO solution supporting multiple realms (domains) along with support of OAuth 2.0 Clients. Keycloak's powerful graphical user interface provides user management, OAuth clients, roles, social logins, sessions, and general settings of each realm defined by administrating user. Single Sign-On is a feature of access control of both related and unrelated applications that have trust relationship and main purpose of SSO is to allow users to sign-on and gain access without being further prompted for credentials over and over again. Having this trust relationship also allow easy management of the users as well as to provide easy access to the users' granted applications [6]. SSO functionality completed using OpenID Connect [7] protocol supported by Keycloak. OpenID Connect is an identity layer that allows clients to validate the identity of the users based on the response of Authorization Server [8]. Basic profile information about users can be fetched using the protocol in REST-like method. Once authentication of the user is done, then authorization take place using OAuth 2.0 –an open standard for authorization. Authenticated user gets authorized and an access token is received from the AuthServer. This token contains user and authorization (user roles etc.) information. OAuth protocol allows resource owner to grant access to clients allowing accessing resources to be provided by the application [9].

Mentioned architecture is currently used by EU FP7 platform and our implementation is actualized successfully. In our work, we installed Keycloak 1.1.0 on a virtual machine running on Ubuntu 14.04 LTS. For this specific platform only one realm created that controls platform roles and users running on this platform. Built-in Java adapters provided by Keycloak encouraged us to complete the development of one of the core application's authentication and authorization using Java EE and other two are secured using JWT.

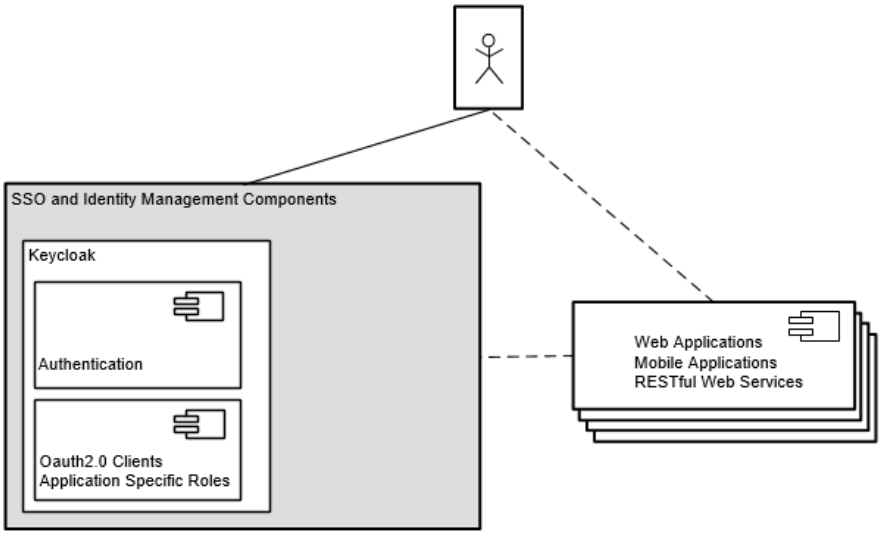

Figure 1: General Structure of the components

Although creation of multiple realm is possible; since only one realm is required for this platform, a single realm is created and users registered on the platform is inserted into this realm. Upon creation of the realm, platform specific roles defined and this role information is used to grant proper access to users.

Single Sign-On implementation is required for all the applications running on the platform as well as the platform itself. This implementation is done by using Open ID Connect. Once users get authenticated from the entry point application and gets the OAuth 2.0 access token, then user can access all the other applications without the need of providing credentials to the applications running on the specified realm until the user session expires. OAuth 2.0 access token contains all the user information including but not limited to username, first and last name, email address and assigned roles. On the platform, this entire flow is done by using bearer access token [10]. In Figure 2, a detailed sequence can be found.

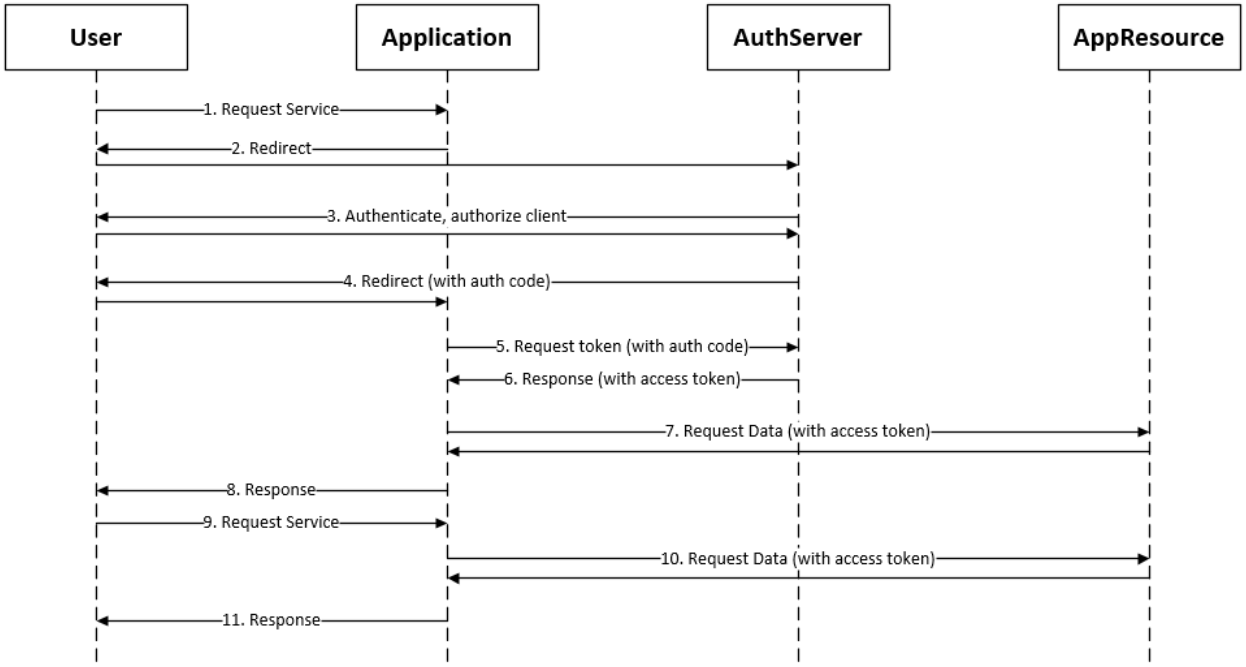

Figure 2: OAuth 2.0 Bearer Access Token Sequence

For a better user experience, value added functionalities also added to the platform including social login, user self-registration, forgotten password, reset password and centrally managed user and role mapping.

Mentioned tools in this proposed architecture can provide extra functionalities including LDAP and Active Directory support, one time password support using Google Authenticator, SAML Support, CORS Support. However in our current platform we are not using these extras.

3. Results

Our experience in this work helped us to understand some of the important advantages of SSO along with some considerations on the topic. From the user experience perspective: This solution helps users not to memorize many different password and allows them easy password management. It improves users' experience through automatic login, and it significantly reduces the risk of account lockouts since users don't try to enter different passwords for different applications running on the same platform.

Even though this technology has many advantages, it requires strong security policies to protect the users' account such as complex passwords, 2-factor authentication [11], verifying users' birthday and possibly using smart cards. If this system is to be setup in an environment, where the shared computers are used then session timeout periods must be short and activity confirmation can be requested from the users. In addition, the final and most important issue might be experienced is the server downtime of the authentication server. Single point of entry might turn into a single point of failure if authentication server is down since all the applications

are using this very same authentication server to authenticate the users.

Multiple open source technologies were tested for a successful authentication and authorization solution. Although each product managed to support the visualized structure to a certain point, all the tested products are failed to provide successful SSO functionality except Keycloak. Above mentioned architecture is implemented, tested and successfully providing all the functionalities required for the platform. Also support for OAuth 2.0 and native Java adapters allow developers to virtually provide SSO integration for applications using any development language that support OAuth 2.0. JBoss regularly updates the Keycloak application and potentially it will be the leading open source project for SSO and Identity Management.

Conclusions

Open source technologies used in this work are sufficient to provide authentication and authorization from small to medium scale platforms and applications. There are some limitations to this solution when it comes to user group management and hierarchical roles meaning that this architecture cannot provide a complete role based access control.

Acknowledgements

We would like to thank Seyhun Mehmet Futacı for his support on virtualization of the servers using OpenStack and for his feedbacks.

References

[1] FIspace is a business-to-business (B2B) collaboration platform. http://fispace.eu/ 10.04.2015
[2] Openstack Foundation. Chapter 1: Example Architecture. OpenStack Operations Guide. April 2013.
[3] Sanna Suoranta, Kamran Manzoor, Asko Tontti, Joonas Ruuskanen, Tuomas Aura. Logout in single sign-on systems: Problems and solutions. Journal of Information Security and Applications. February 2014; Volume 19, Issue 1
[4] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence. Generic AAA Architecture. http://www.hjp.at/doc/rfc/rfc2903.html 10.04.2015
[5] Keycloak Reference Guide - SSO for Web Apps and REST Services. http://docs.jboss.org/keycloak/docs/1.1.0.Final/userguide/pdf/keycloak-reference-guide-en-US.pdf 10.04.2015
[6] Chin-Chen Chang, Chia-Yin Lee. A Secure Single Sign-On Mechanism for Distributed Computer Networks. March 2011. Industrial Electronics, IEEE Transactions on; Volume: 59, Issue: 1.
[7] Ryan Boyd. Chapter 7: OpenID Connect Authentication. Getting Started with OAuth 2.0. February 2012.

[8] OpenID Connect – An identity layer. http://openid.net/connect/ 10.04.2015
[9] D. Hardt, Ed. The OAuth2.0 Authorization Framework. http://tools.ietf.org/html/rfc6749.html 10.04.2015
[10] M. Jones, D. Hardt. The OAuth 2.0 Authorization Framework: Bearer Token Usage. http://www.hjp.at/doc/rfc/rfc6750.html 10.04.2015
[11] Aloul, F. Zahidi, S. El-Hajj, W. Two factor authentication using mobile phones. May 2009. Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on.