available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

# Reverse OAuth: A solution to achieve delegated authorizations in single sign-on e-learning systems

*Jorge Fontenla González**, *Manuel Caeiro Rodríguez, Martín Llamas Nistal, Luis Anido Rifón*

*Escuela Técnica Superior de Ingenieros de Telecomunicación, University of Vigo, Campus Universitario, 36310 Vigo, Spain*

ARTICLE INFO

*Article history:*
Received 2 April 2009
Received in revised form
8 June 2009
Accepted 8 June 2009

*Keywords:*
Access granting
Delegated authorization
E-learning
Password stress
Reverse OAuth
Service provider
Single sign-on
Web service

ABSTRACT

Current scientific and technological progress has led to the proliferation of e-learning systems known as *Learning Management Systems*. These systems consist of a central application for managing the sequencing of students' tasks, and also on several other educational applications that allow its users (teachers and learners) to communicate, carry out experiments, etc. However, despite the widespread use of these systems they show a usability problem when both kinds of applications require spare authentication processes. Indeed, users have to introduce several kinds of credentials, preventing them from focusing their efforts on their studies and increasing the so-called "password stress". Several initiatives such as OAuth or Delegation Permits have dealt with the problem of delegated authorizations, but their requirements are different from those that arise from an e-learning environment. In this paper we introduce Reverse OAuth – a protocol to enable the granting of authorizations to access protected resources in educational environments.

## 1. Introduction

Given the rapid scientific and technological progress, organizations such as universities and enterprises have envisaged the need to provide continuous formation to their students and employees (namely, life-long learning). This fact, together with the widespread use of broadband Internet connections, has led many institutions to grant access to educational resources through e-learning systems known as LMSs (*Learning Management Systems*). These systems deal with the administration, provision and control of educational resources and functionalities. Therefore, LMSs have achieved a breakthrough in education, as they allow covering educational needs avoiding spatial and temporal barriers.

Current LMSs can be considered as complex Web applications. Some of the best-known examples are Moodle (Web site of the Moodle project, 2009), Blackboard (Web site of the Blackboard project, 2009), LRN (Web site of the dotLRN project, 2009) and Sakai (Web site of the Sakai project, 2009). These systems typically provide a centralized environment to supply data (pdf documents, multimedia files, etc.) along with applications or tools to manipulate them. Nonetheless, the growing complexity of LMSs is leading to a design approximation in which the tools are split from the LMS itself (Fontenla et al., 2008; IMS Tools Interoperability Specification, 2009; Vogten et al., 2006). Tools become standalone Web applications which are *not part of* but are *used* by the LMSs (IMS Tools Interoperability Specification, 2009).

* *Corresponding author*. Fax: +34 986 814 073.
E-mail addresses: jfontenla@det.uvigo.es (J.F. González), manuel.caeiro@det.uvigo.es (M.C. Rodríguez), martin.llamas@det.uvigo.es (M.L. Nistal), luis.anido@det.uvigo.es (L.A. Rifón).

0167-4048/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2009.06.002

However, this approach also has some difficulties. Some of them inherently arise from the invocation of remote services: discovery of remote interfaces, transmission of real-time data, privacity and integrity of the messages, etc. Nevertheless these problems, although critical, are to a great extent solved by existing technologies.

Other kinds of difficulties are related to the usability of these systems and, due to the innovative approach of splitting an LMS and its tools, they have not been tackled yet. It would be desirable if the students did not have to authenticate themselves after the tools, provided that they have already been authenticated at the LMS. This can be seen as the counterpart of what happens in ''old-fashioned'' LMSs containing tools as well as the LMS itself: a student has to authenticate only once at the beginning of the session, but after that he can use the tools freely without having to authenticate again. This authentication principle in which a user can access many systems with a single authentication instance is frequently known as *single sign-on* (Introduction to single sign-on, 2009).

The study of a way to accomplish single sign-on within the LMS and the different Web tools is the main purpose of this article. We take as a starting point four single sign-on technologies, analyze their suitability to our e-learning environment and present our own proposal that solves the problems identified.

This paper is structured as follows. In Section 2 we describe a typical e-learning scenario where a new single sign-on technology is required, and list the main requirements that must be satisfied. Section 3 depicts an overview of OAuth, Delegation Permits, Shibboleth and OpenSSO, and analyzes them against the requirements of Section 2. In Section 4 we come up with a proposal of solution which we called Reverse OAuth, that solves the lacks identified in Section 3, and in Section 5 we describe its implementation process. We end up the article with Section 6, where we extract some conclusions.

## 2. Problem description

In the previous section we mentioned some general-purpose authorization technologies. Nevertheless, in order to analyze their suitability for our purposes we have to formalize the problem we are dealing with. This section offers a clear picture of what we want. Section 2.1 goes into some detail on the architecture we based our work on, while Section 2.2 provides several use cases concerning the use of delegated authorizations that could take place over this architecture. From these use cases, Section 2.3 extracts the main requirements of a solution.

### 2.1. *Architecture description*

Fig. 1 depicts the architecture under study. We can see three entities – the LMS, a Tool and a user:

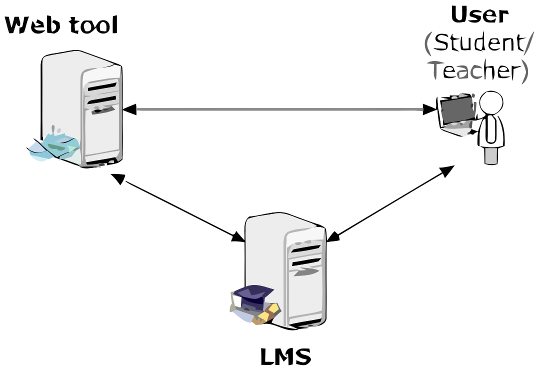

**Fig. 1 – Architecture description.**

- On the one hand, the LMS provides the core functionality of the educational system (e.g. authentication modules, databases to store personal data of the students, the logic to manage sequencing of tasks).
- On the other hand, the Web tool is a standalone application providing a specific functionality that help users to carry out their tasks. Examples of Web tools can be a forum, an assessment application, a hydrodynamics simulator, etc.
- Finally the user accesses the LMS to carry out his/her tasks with the aid of the Web tool.

This decoupling of LMS, users and Web tools allows LMSs to extend their functionalities in an easy way, as the only requirement to add new features to the LMS is establishing an Internet connection with the appropriate Web tool. Thus an LMS can use many Web tools, but also a Web tool can be used by many LMSs. This solution implies that the development of both LMSs and Web tools can follow separate paths.

To that end, the developers of the LMS have to adopt certain specifications to support the interaction with the Web tool. The developers of the Web tool must also embrace these interaction specifications in order to allow communication with the LMS. This mutual interaction is represented in Fig. 1 with an arrow that links the LMS with the tool. Users take advantage of this mutual interaction to carry out the tasks indicated by the LMS by using the Web tool, which is represented in Fig. 1 by the arrows coming from the user. Notice that there may be more that one Web tool available, and that the user accesses the one that fits better with his/her needs.

The Web tool may (and should) have implemented some access control in order to prevent unauthorized users to access it. This brings up the possibility that only the students from a university are allowed to use the tool, but not those from another university. In this document we consider two ways to perform this access control:

1. Each user of the LMS has a working account at the tool.
2. The LMS has a working account at the tool, and its users are granted access as invited users. This mechanism is known as *delegated authorization*.

The first solution is neither scalable nor practical, as users have to remember many passwords (one passwords to access the LMS, plus one password per tool). Moreover, they are asked to authenticate themselves every time they want to access a tool. The second solution solves the drawbacks of the first one, as it is based on a delegated authentication mechanism. Hence, we have chosen it to base our single sign-on solution.