SmartScribo Blog Processor for Multi-Blogging in Smart Spaces

Rustam Kadirov, Dmitry Korzun
Petrozavodsk State University
Petrozavodsk, Russia
{kadirov, dkorzun}@cs.karelia.ru

Abstract

SmartScribo is a Smart-M3 application that provides advanced access to the blogosphere via blog-related knowledge representation in smart spaces [1]. SmartScribo users use their mobile devices and interact with blogs hosted at multiple blog services. The architecture defines three types of knowledge processors (KPs): blog clients, blog processors (BP), and blog mediators. Clients access in the smart space those blogs that the users are interested in. When some data are needed from a blog-service or the user has data for a blog-service, then the client publishes notifications in the space. Each BP tracks notifications and implements the interface to the particular blog-service. Mediators extend basic blogging with smart features, e.g., blog recommendations.

There are many popular blog services like LiveJournal, Twitter, Blogger, and WordPress. The current version of SmartScribo includes a Python-based BP-LiveJournal only. It uses a low-level Smart-M3 KP Interface (KPI) for accessing the smart space from Python code. This abstract introduces two additional SmartScribo BPs. 1) BP-Twitter uses Twitter API [2] and accesses the popular service of reduced blogs. 2) BP-RSS allows accessing various blog services with RSS support.

Twitter refers to the class of reduced-blog services. A corresponding BP implements a limited number functions compared with BPs for such a blog service as LiveJournal. In particular, Twitter has the small blog message size, comments to posts or to other comments are not allowed.

Our BP-Twitter implementation uses Twitter API. Service authorization follows the protocol Oauth [3]. Figure 1 shows the basic authorization process, which is not bind to a particular type of service (Twitter in our case) and can be used for many blog-services. Thus we can use the same scheme later to implement BPs for other services.

To implement the Oauth protocol on the client side we utilize liboauth library [4]. It is a collection of POSIX-compliant functions. Liboauth provides functions to escape and encode parameters according to the OAuth specification and offers high-level functionality to sign requests or verify OAuth signatures as well as to perform HTTP requests.

BP-Twitter is written in C/C++ using SmartSlog ADK [5], i.e., its logic fully follows the high-level ontological representation of smart space content. The basic function is reading, publishing, and refreshing posts (status in the Twitter terminology). The scheme is shown in Figure 2 (top).

RSS is an XML-based format and can be used to read blogs from various blog-services such as Twitter, LiveJournal, WordPress, and many others. Our BP-RSS is written in Python and uses the same Python-based low-level Smart-M3 KPI as the original SmartScribo BP-LiveJournal. The basic function is reading posts and comments from blogs hosted at services with RSS support. It uses the Universal Feed Parser to get access to a blog service. The scheme is shown in Figure 2 (bottom).

Index Terms: Smart Spaces, Smart-M3, SmartSlog, Blogging, SmartScribo.

REFERENCES

[1] D. Zaiceva, I. Galov, and D. Korzun, "A blogging application for smart spaces," in *Proc. 9th Conf. of Open Innovations Framework Program FRUCT and 1st Regional MeeGo Summit Russia–Finland*, Apr. 2011, pp. 154–163.
[2] "Documentation: Twitter developers," Sep. 2011. [Online]. Available: http://dev.twitter.com/docs
[3] E. Hammer-Lahav, "The OAuth 1.0 Protocol," RFC 5849 (Informational), Internet Engineering Task Force, Apr. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5849.txt
[4] R. Gareus, "Oauhth library functions: liboauth 0.9.4," Sep. 2011. [Online]. Available: http://liboauth.sourceforge.net/

[5] A. Lomov, P. Vanag, and D. Korzun, "Multilingual ontology library generator for Smart-M3 application development," in *Proc. 9th Conf. of Open Innovations Framework Program FRUCT and 1st Regional MeeGo Summit Russia–Finland*, Apr. 2011, pp. 82–91.

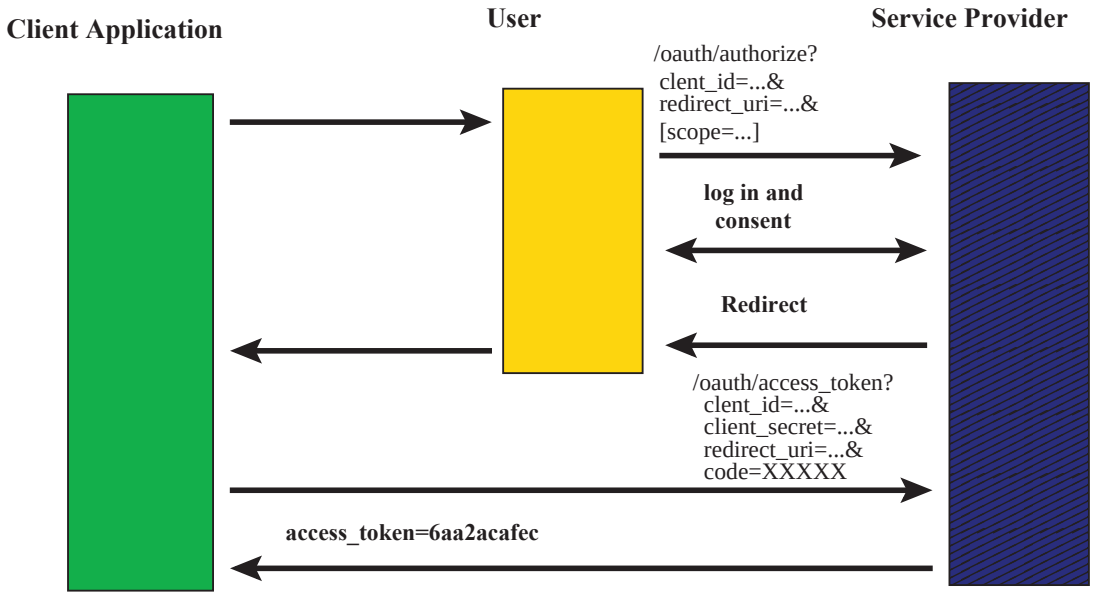

Fig. 1. Protocol Oauth [3]: authorization steps

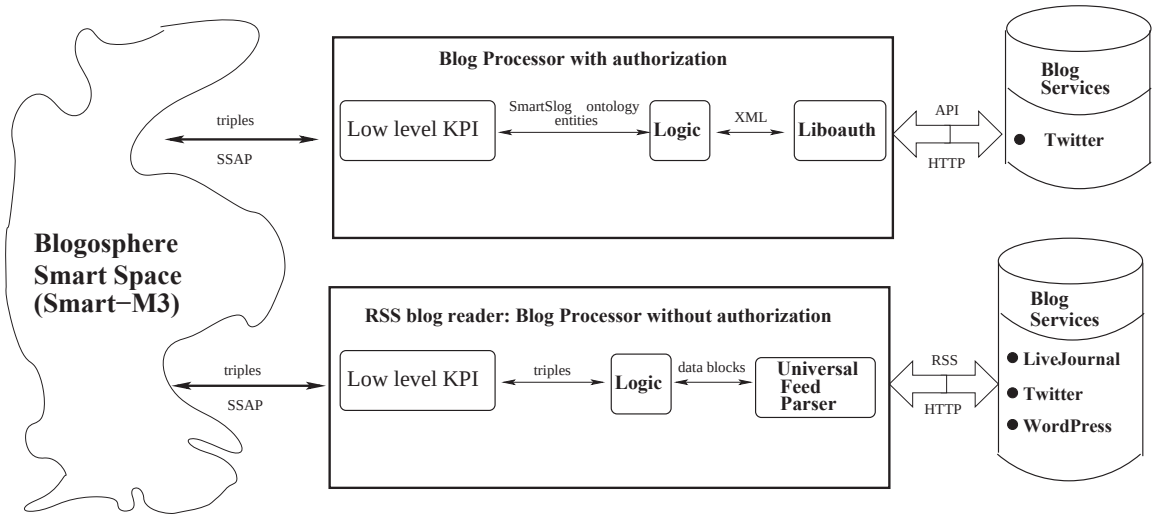

Fig. 2. Architecture of blog processors and environment.