Single Sign on and Single Sign Off in a non homogeneous Portal front ended environment.

Alan Berg, Bas Toeter

Central Computing Services, Universiteit van Amsterdam, The Netherlands.
a.m.berg@uva.nl, b.toeter@uva.nl

Abstract

The Central Computing services at the Universiteit van Amsterdam (UvA) [17] have deployed an instance of the JASIG Uportal [18] system for use by a student audience of around 22,000. To achieve the requirement of a homogenous experience for the end user required a transition to a more consistent and unified Single Sign-On (SSO) and Single Sign-Off interaction with the underlying diverse infrastructures. This paper describes the experiences of the local integration developers and places their lessons learned into the wider context of campus wide infrastructure building.

Keywords: Portal, Single Sign-On, Blackboard

1 Overview

Authentication and directory services at UvA are currently dominated by LDAP. A secondary methodology is that of the proprietary Microsoft ActiveDirectory, that has password synchronization with LDAP and partial record synchronization with the student administration system ISIS. This usable combination delivers a single campus wide id (named uvanetid) and password, but not the Single Sign On and Sign Off experience that is expected by our primary customers, the full population of the University, especially portal users.

The main functionality of any portal is to accumulate information from diverse sources in one place. For UvA, channels to help students view notifications of new email or course announcements, is a must have requirement. Ergonomics dictate that once your have such information in a portal then the end user would expect to be able to travel freely between deep linked systems that are therein mentioned. If this ability is not supplied then the expectation management will fail and the portal project will be perceived as a failure.

The predominant Electronic Learning environment at UvA is Blackboard [5] and with over 3700 active courses and around 15,000 unique active users that have logged on in the last three months it is now considered a vital system. The same can be stated for student Webmail which has an even denser usage pattern than Blackboard. The Webmail system is based on the same software line as the LDAP services. Both use SUN ONE [14] products and mail routing information sits also in LDAP records.

With the advent of a student portal project "mijnUvA", ending last year, the failure to deal with Single Sign-On, read basic session management, came to the fore. Using an unaltered infrastructure was not supportable. For one thing there would be significant resistance to adoption of the personalized portal. Imagine clicking links within the Portal, but with the end resource outside the portals grasp that the students would have to have log in to mismatched web based applications with the same uvanetid and password. over and over and did I say over again. The developers clearly not wanting to disappoint choose the Yale Central Authentication Service (CAS) [7] open source server to remove this session management obstacle.

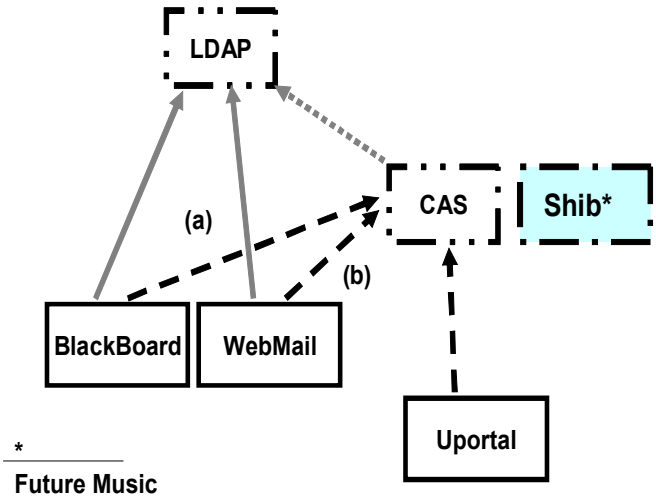

Figure 1: defines the strategy for inserting an SSO solution into the legacy infrastructure. Please note: Shibboleth or another product with similar functionality may be added later in combination with CAS without propagating changes to CASified applications.

Figure one describes the strategy chosen, adding a second authentication method that is activated for one specific login URL per application. Previously both BlackBoard and the Webmail system pointed at LDAP for authentication. Uportal[1] did not have such a relationship.

To enhance the surfing experience of the student both Webmail and Blackboard were given in addition to LDAP a second authentication possibility. The easy to highlight advantage of this is that the student has the legacy way to login, but within the portal a new mechanism exists that does this for you generating a CAS session. The uPortal acts as a proxy, hence for the end user no retraining of surfing habits. The cut and splice approach is transparent.

Enabling was done per application:

(a) Blackboard used a custom security plugin developed at Bristol University [6] and then slightly modified at UvA. The modification was minimal and just meant that fallback after CAS went to the LDAP provider and then the database rather than directly to the database of Blackboard. The end result of this effort is that if a user browses a specifically defined URL within Blackboard before logging in, CAS authentication takes place and the student is then logged in automatically. In addition a parameter can also be sent describing where blackboard should redirect the user after login. With a small amount of imagination this redirection can place you exactly where you want from a portal channel link.

(b) For Webmail and other products SUN has its own identity management suite, but this was not considered as viable as a more open solution. However SUN has what can now be considered a legacy method named trusted circle [15]. We adopted that method for short term gain and wrapped CAS with a filter[2] [11] to stimulate session generation within Webmail. This process chain will be fully described in detail in the next section. If a user first logs into CAS an extra cookie is returned to the users browser with information that can be verified under the water by Webmail. When a user browses the login page of Webmail the cookie is then verified against a UvA specific session manager and thus a new session is generated.

2 Details

The Uportal System is a rapidly evolving personalized content system that is freely available and based on the Java programming language and specific frameworks built there over. The developer is free to create authentication providers or use one or a number from an ever increasing list of included providers. UvA pragmatically choose to authenticate students via CAS as designed and pioneered by Yale University. The seven most significant advantages of this choice were:

(1) CAS works well with the standard version of uPortal [8]. In fact no programmatic changes are required.

(2) It is an open sourced market leader with a solid number of deployments.

(3) CAS is a secure and a well known and documented product. In fact the number of known security flaws that have been patched are nicely limited.

(4) The web application is extremely easy to deploy. Basically you drop the code into a servlet container such as Tomcat [16] and with a few trivial changes in configuration you have a workable prototype.

(5) CAS is based on Java and thus all the coding in the project was based on the same object orientated, team orientated programming language, with all the inherent advantages of a uniform IDE tool base.

(6) The product is specifically designed to be easily customizable. Changing of look and feel was a question of modifying a few Java Server Pages (JSP)

(7) CAS is nicely future proofed. It will in the very near future work well with Shibboleth [9,13] which will allow a privacy enabled form of SSO across campus boundaries and allow a federation of Universities to share virtual organizations and resources.

The one definable weakness of CAS is that it is a single point of failure. The servlet cannot at present be distributed across a number of servers with the same session manager and thus is open to denial of service attacks and random hardware

---

[1] **uPortal**

uPortal picks up user information the first time the user logs in, but not the password from LDAP. The user information for example the department name can be used for personalization of channels within the portal. Some of the information is then stored permanently in the portal database.

[2] **Filters**
Filters work in a chain, as a request is sent by the browser to a server. First the filter sees the request and can modify it, then the next filter in the chain and so on. When the filter chain has finished the web application processes the modified request. The same is true in the other direction. Once a web application has sent a response the filter chain is free to modify it before it hits the browser.

The major advantage of the filter is that you can add functionality to the application without changing the code.

failures. We have tried our collective best to lower the risks by good old hardware redundancy and solid monitoring. Further we connected the CAS security provider to multiple copies of the LDAP directory structure. This provider is responsible for checking the directory server for the correct password and for fail over in case a particular LDAP[3] instance is down or that an intermediate piece of the network has issues.

For the Blackboard environment It was possible at the time of integration to choose between two methods of joining with CAS. The first was a custom channel for Uportal that also included an extra authentication provider for Blackboard. The provider added a user to the Blackboard session table directly, that is under the condition that the channel asked for information and the user was not logged in. Early in the project the developers felt that this process was too raw and should have been more abstracted by a data integration API that was provided by Blackboard. API's tend to be more stable than the underlying data model.

The second approach and the one that was applied was that of configuring a specific CAS provider for Blackboard. Blackboard has a stackable security plugin framework, similar to the Pluggable Authentication Module (PAM) framework [3], but with a pure Java perspective where by implementing a specific interface and reconfiguring the servers properties it was possible to enable CAS. Bristol University had already done the ground work and their excellent recipe was thus deployed with very minor tweaking. The tweaking involved allowing a fallback authentication to LDAP and then the Blackboard database rather than just the database.

The positive implication for the end user was that they could choose between the normal authentication method or CAS. This made deep linking[4] within Uportal channels viable and transparent.

A significant issue for the project was the integration of Sun specific products. UvA's student Webmail system is a specific configuration of the Sun One Product. One of the native Single Sign On solutions was trusted circle. Each server is configured with a list of other servers that are trusted. On creation of a session for one server a cookie is placed in the users browser with a session id. The cookie name allows another server in the circle to know which server has set the cookie. On browsing another trusted member in the circle the cookie is seen and then the target server checks against the origin server for session correctness. In other words to enable this type of authentication through the CAS server requires that the CAS server becomes also a member of the trusted circle. This was first achieved via integration in Uportal and then later factored out to the more logically placed CAS server.

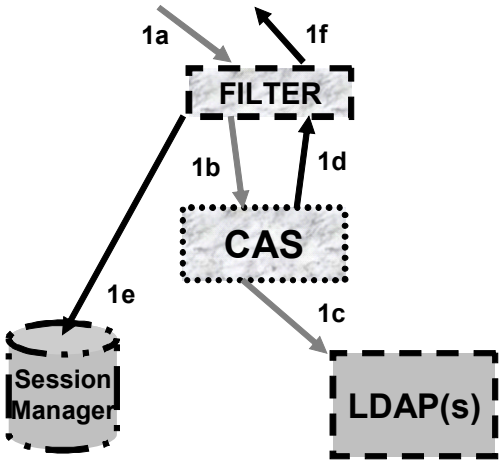

Figure 2: The enhanced SSO filter and its relationship to logging onto CAS.

The exact details of this mechanism are given in figure 2 and 3. Figure 2 describes how a user logs into CAS and has a session generated for trusted circle. The collaboration is as follows:

(1a) Login page CAS. User enters username and password. The information, in the form of a request passes through the filter without hindrance.

(1b, 1c) CAS checks via a custom security provider the information against LDAP

(1d) the servlet sends a response, including a CAS cookie [2].

(1e) the filter seeing login generates a trusted circle session. This is achieved via talking to the session manager of the local instance of the trusted circle that sits within the same tomcat server, but as a different web application.

---

[3] **TIP:** The consortium ESUP-portail [10] version of CAS has a number of extra authenticators including one for LDAP. So check there work before custom coding.

[4] **Deep linking**
Placing a link in a portal that points to a resource outside its own authority and deep within another's.

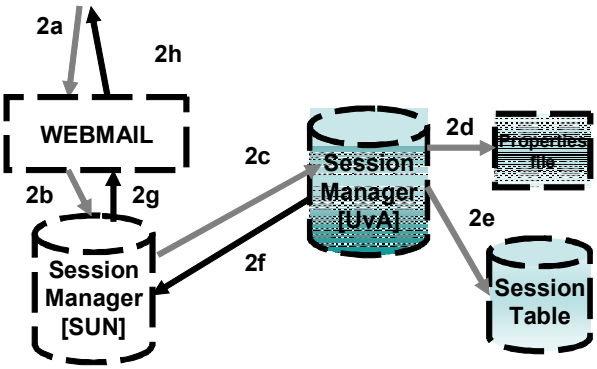

Figure 3: Logging onto Webmail after logging into CAS

(2a) later user browses Webmail login page

(2b, 2c) Local instance of trusted circle talks with local CAS instance of trusted circle.

(2d, 2e, 2f) The UvA manager checks that the service exists in its personal allowed list and then checks that the session mentioned is in the local session table. Finally sending affirmation.

(2g, 2h) The Sun manager updates its own session table and sends session information back in the form of a parameter and an extra cookie.

Just as the developer thinks that their work is finished other issues come to the fore. Yes Single Sign On is achieved, but Single Sign Out is also expected. Life is full of compromises. Building future proofed Single Sign On infrastructure with a backdrop of legacy systems requires a considerable amount of compromise and attention to detail. During this integration project it became very clear that there is a difference between Sign-On and Sign-Off. Each mechanism involving a session manager potentially requires a different approach to signing out. The worst case situation is that a student is sitting in an Internet Café and has signed off but has not closed their browser. Five minutes later someone else sits by the same machine and by going through the history replays the previous URL's. If cookies are left or information in the form of parameters in the URL itself then unwanted hijacks may occur. Sessions have to be deleted from multiple systems to avoid such replay attacks. For Blackboard and Uportal all that was required was that the server deleted the correct cookie. For trusted circle an explicit logout was necessary and this was achieved via redirects within a hidden frame, a dirty but necessary technical compromise.

3 Experiences

Introduction of the MijnUvA portal to the campus infrastructure was incremental and restful. The system performed as expected. A few minor issues have been found and resolved. At a low level occasional random sign on problems emerged with CAS and Blackboard. This issue is difficult to track and appears not to be an issue with CAS, but rather timing of session creation in Blackboard.

Figure 4 is generated by an agent based tracking system for Blackboard [1]. Figure 4b shows unique user logons per every hour. The lowest plotted line is for the login failures. Around one in four logon fails. The ratio is quite normal and within our expectations. The one in four baseline was not increased noticeably by the introduction of the portal system. Further figure 4a shows the usage profile of Blackboard over approximately a month. The plot is the number of unique users that have logged on against time in segments of three hours. Introducing a portal system should push up average usage of Blackboard. Further it should push up usage in the weekends as the portal system motivates more people to surf during off duty hours. This should push up the peaks in the weekend relative to the peaks in the week. At present we see no such flattening of distribution. However as more channels (read services) are introduced into the portal we are in a position to track the changes and observe if the influence is as predicted.

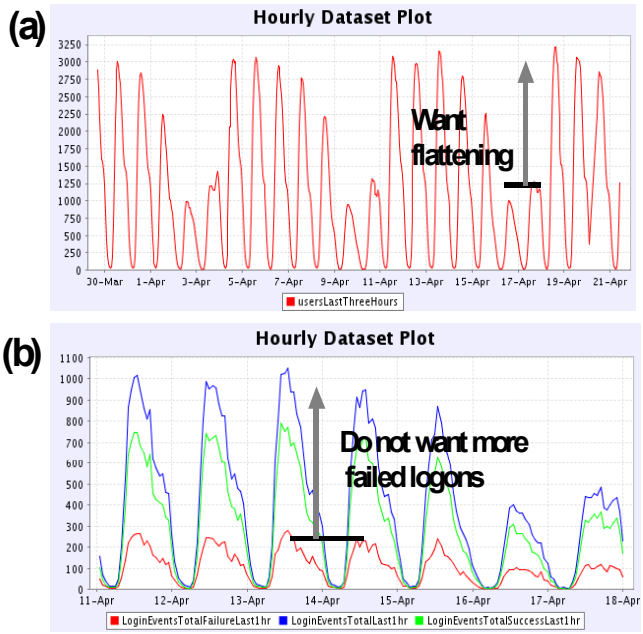

Figure 4: Usage patterns in Blackboard after the installation of CAS authentication.

4 Onwards to the future

The Dutch have a have positivistic and resonant sounding phrase "Toekomstmuziek", future music, looking to the

future. Something that is not relevant in the daily strive to improve campus wide issues, but may well be very soon is the accumulation of federations of educational institutes. That is building of relationships via sharing of resources and building of virtual organizations or grids [12] There is an oncoming tidal wave of integration of electronic resources used within the educational market place. The authors personally see a future for CAS and Shibboleth in synergy [9]. Shibboleth enables login from users from other institutes without revealing unnecessary information that may have privacy implications [4]. For example which department a user works in may be relevant to login, but not always their specific id or email address. These present but not always fully recognized issues may require UvA to change infrastructural course. However, as shown in figure 1 the CAS Shibboleth hybrid may in the near future plug easily into the current CAS position.

At the time of writing this paper at least one of the authors is actively working with Ex Libris towards CAS authentication for the Metalib library system, a significant player in the market. As soon as this effort is finished it can be expected that contact with Metalib will be added as significant channel functionality to the student portal or perhaps directly with Blackboard.

5 Summary and conclusions

Implementation of Single Sign On and Sign Off requires much detailed configuration and tweaking of the campus wide infrastructure. The details of which can be at times more than complex. However this was achieved at UvA for the student portal via the reuse of freely available open sourced code and recipes. The authors see open source and community effort as the most productive way to develop, build and deploy for Universities on such a scale and to an acceptable price. The programming language Java has provided us with a robust object orientated, team orientated language that is a common factor spanning all our current projects. It has not let us down. Yale CAS works now and will work later with more federated SSO solutions; it looks very much like a good bet.

Acknowledgements

The authors would like to acknowledge the quality feedback during the compilation of this paper from Marc van den Berg a fellow hard worker at the University of Amsterdam and Erik van der Velde the chief architect and also chief pressure behind the project.

References

[1] A. Berg, V. Maijer, Frank Benneker. “Blackboard 6 usage patterns and implications for the Universiteit van Amsterdam”, Eunis 2004.

[2] D. Kristol, L. Montulli “RFC 2965 - HTTP State Management Mechanism”.

[3] V. Samar, C. Lai “Making Login Services Independent of Authentication Technologies”. http://www.sun.com/software/solaris/pam/pam.external.pdf

[4] Von Welch, T. Barton, K. Keahey, F. Siebenlist “Attributes, Anonymity, and Access: Shibboleth and Globus.Integration to Facilitate Grid Collaboration” In *4th Annual PKI R&D Workshop (To appear)*, 2005.

[5] Blackboard. http://www.blackboard.com

[6] Bristol Recipe for Blackboard authentication against CAS. http://www.bris.ac.uk/is/projects/portal/software/blackboard_cas

[7] CAS from Yale. http://tp.its.yale.edu/tiki/tiki-index.php?page=CentralAuthenticationService

[8] CAS in relation to Uportal. http://jasigch.princeton.edu:9000/display/CAS/uPortal+Client

[9] CAS in relation to Shibboleth. http://tp.its.yale.edu/shib/tiki-index.php?page=CasShib

[10] ESUP CAS variant. http://esup-casgeneric.sourceforge.net/

[11] Filters for servlets. http://www.javaworld.com/javaworld/jw-06-2001/jw-0622-filters.html

[12] Grid Shibboleth. http://grid.ncsa.uiuc.edu/papers/gridshib-pki05-final.pdf

[13] Shibboleth. http://shibboleth.internet2.edu/

[14] SUN Identity management. http://www.sun.com/identity_mgmt

[15] SUN ONE Calendar Server Programmer's Manual. http://docs.sun.com/source/816-6416-10/pr8ssn.html

[16] Tomcat servlet container. http://jakarta.apache.org/tomcat/

[17] Universiteit van Amsterdam. http://www.uva.nl

[18] Uportal. http://www.uportal.org