



Kidderminster College Repository Of Learning Objects – Final Report - 14/03/06

Project Name

KC-ROLO

Report Authors

Graham Mason
Ed Beddows

Contact Details

Graham Mason
01562 512067

Acknowledgements

The KC-ROLO project was funded by JISC under the Core Middleware programme.

We also gratefully acknowledge the following individuals contributions to the project:

Andy Morris (RSC West Midlands), Peter Kilcoyne (RSC West Midlands), all the IAMSECT project team for their Pubcookie guide, SDSS and the SIPS project team for their work and support with the PERMIS module.

Table of Contents

Executive Summary.....	3
Background.....	4
Aims and Objectives.....	4
Methodology.....	5
Implementation.....	6
Dissemination Events.....	13
Outputs and Results.....	14
Outcomes.....	16
Conclusions.....	17
Implications.....	17
References.....	17
Appendixes.....	18
Appendix A – KC-ROLO Federation Overview.....	19
Appendix B – Active Directory IdP Base DN issue.....	20
Appendix C – Apache directives.....	21
Appendix D - Shibboleth User guide.....	22

Executive Summary

The KC-ROLO project partnered with the University of Worcester and the RSC West Midlands to investigate and implement a secure way of sharing repositories and Virtual Learning Environments using the Shibboleth framework. The objectives of the project were:

- To share learning resources between Kidderminster College, RSC West Midlands and the University of Worcester via Shibboleth architecture and Moodle Open Source VLE.
- Develop a PERMIS/Shibboleth hybrid, with enhanced Role Based policies allowing for more powerful authorisation policies.
- Accompanying source code and documentation for PERMIS/Shibboleth hybrid.
- Create guides for implementing Shibboleth and all above components from scratch on different platforms.

It was later discovered that another project ran by the developers of PERMIS were themselves creating the Shibboleth/PERMIS module, this allowed us to concentrate on the other aims of our project. The previous aims were mostly attained in the first year of the project, this gave us time to look in more detail at the solutions available, and extend these capabilities where possible. Second year objectives were:

- Continue Shibbolizing of Moodle VLE.
- Develop Moodle to read course attributes.
- Use of PERMIS to improve Moodle authorisation issues.
- Use Windows Server 2003 for an IdP, looking at the possibility of automatic logons for internal users.

The projects main aim was to implement a Shibboleth framework between Kidderminster College and the RSC West Midlands before implementing a similar setup at the University of Worcester. However, most of the work throughout the project focused on our link to the RSC West Midlands because we both use Moodle as our VLE solution.

An unexpected added component of the project was the managing of the KC-ROLO federation, starting with just two servers, it has ended up serving 12 IdP's and 4 SP's, a lot more than originally expected.

The report will show how some of these objectives were changed in the middle of the project to allow us to focus on more important issues. It is concluded that the use of Shibboleth for sharing resources such as repositories, Virtual Learning Environments and other educational web resources is incredibly suitable, to the extent that Kidderminster College are now using it in a production state for their students and staff.

Background

Whilst many institutions are using Virtual Learning Environments it can be said that the information in these are closed, unable to be accessed by anyone but the users from the institution and sometimes just those on a specific course.

Repositories help with this problem by providing a central store for learning objects, however there still remains the issue of allowing access by other students or staff from another institution, for example where a collaboration exists between multiple sites, the simple solution to this is adding the remote users to local authentication system, this however scales badly and requires extra resources by IT staff.

At the start of the project we were in a similar situation to this, running an open source VLE called Moodle and thinking of implementing a repository, we also have students from the University of Worcester doing courses at our site, using our VLE. We achieved this previously by setting manual accounts in the VLE, this meant Worcester University students had yet another username and password to remember, and ultimately lead to more support calls and administration time by Kidderminster College IT staff.

The RSC West Midlands were involved largely due to the fact that they were championing the idea of a regional repository, which at the time was called WILMAR and now E-Source, working with us they could see if this was a feasible solution to roll out to the region, in addition to this the RSC West Midlands were also beginning to use the same Moodle VLE as us, which was hosted on a server where they wanted anyone in the region to be able to logon and test the software. Their implementation at the time consisted of manual accounts, this made it very hard to track user accounts as potentially any academic member in the region may desire access to it.

Shibboleth sounded like a perfect platform to resolve all of these issues; hence KC-ROLO was born.

Aims and Objectives

Our aims and objectives for the project consisted of the following.

1. To share learning resources between Kidderminster College, RSC West Midlands and the University of Worcester via Shibboleth architecture and Moodle Open Source VLE.
2. Develop a PERMIS/Shibboleth hybrid, with enhanced Role Based policies allowing for more powerful authorisation policies.
3. Accompanying source code and documentation for PERMIS/Shibboleth hybrid.
4. Create guides for implementing Shibboleth and all above components from scratch on different platforms.
5. Continue Shibbolizing of Moodle VLE.
6. Develop Moodle to read course attributes.
7. Use of PERMIS to improve Moodle authorisation issues.
8. Use Windows Server 2003 for an IdP, looking at the possibility of automatic logons for internal users.

Our first aim of sharing learning resources between Kidderminster College, RSC West Midlands and University of Worcester was met during the first year of the project, the only change on this objective was the University of Worcester IdP was not linked to the internal Authentication due to a lack of communication with their IT staff, this was due to the overwhelming pre new university semester work they had to achieve before the start of their new term.

We found out just before starting objective 2 and 3 that another project were already implementing a PERMIS module for Shibboleth, the expertise of the project members with PERMIS was far greater than ours as they actually designed and developed the original PERMIS software. Later PERMIS work packages consisted of testing the modules that were created by this projects work.

Objective 4 was changed slightly, instead of the guide applying to multiple platforms it was written specifically for Fedora Core, this was mainly because documenting the slight changes of each platform would require a lot of work, which perhaps was unnecessary due to the fact that a large bulk of the code was generic across all platforms of Linux. Windows systems administrators would however find little guidance from the guide apart from the file configuration process which is largely platform independent.

Objective 5 continued on with our work in objective 1, no changes were made and the objective was met.

Development in Moodle for reading course attributes in objective 6 was changed a little, instead of restricting access at the course level with Shibboleth we did this at the departmental level, this was easier to implement and allowed us to continue using Moodle’s internal enrolment functions.

Objective 7 required us to use PERMIS for authorisation, after testing PERMIS on blogging software and seeing the complex setup required in the background it was decided that this implementation was not suitable.

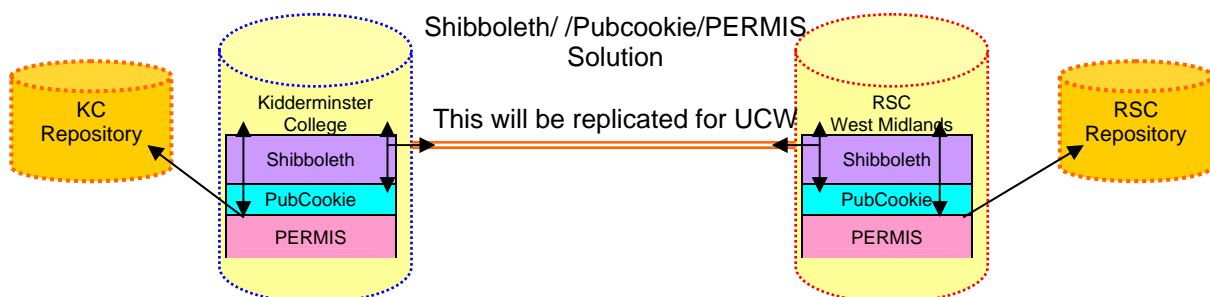
Our final objective of creating a windows IdP with automatic login was achieved, no changes were required.

Methodology

Our approach to getting this all setup was fairly simple, there are three institutions wanting to share at most a VLE and repository each, so it was decided to implement the Shibboleth framework first internally at Kidderminster College, this saved us time as it meant our first IdP and SP were both done on site. Once we had gained experience in the installation of Shibboleth in a “one on one” none federated environment we planned to introduce an SP at the RSC West Midlands to share Moodle, and an IdP at the University of Worcester. The introduction of multiple IdP’s meant we would also needed to implement a WAYF at this point.

PERMIS would later be integrated and tested once Shibboleth is up and running and any issues have been resolved.

Below is a simplified diagram done during the planning stage, it shows how we thought the implementation of all these components would look once the RSC WM had their SP in place, and our intentions to replicate this pattern for the University of Worcester later.



Throughout the project we were to maintain documentation to allow new users of Shibboleth to easily install and administer the relevant software. Our driving force for this is to get more FE institutions using

Shibboleth, perhaps the biggest barrier for this is the skill sets in such institutions, the guide will hopefully address this issue by lowering the skills required.

The second year of the project was to extend on the current work of the federation by looking further into the authentication and authorisation of the Moodle VLE's in place at both Kidderminster College and the RSC WM. Whilst we felt happy with our progress in this area we realised further research could be done to make the experience better for the end user.

Once we were happy with the state of the federation and Moodle components we would concentrate on implementing a Windows Server IdP using IIS and Tomcat, we see this as an important work package as many institutions only have experience in Windows, removing barriers such as OS requirements can only be a good thing. In addition to Windows being familiar to a lot of IT staff who may potentially use Shibboleth, we feel using Windows Server and IIS for an IdP would allow the institution to plug the Handle Service into the sites Active Directory, using "Windows Integrated Authentication" would provide users logged onto the domain with seamless access to any Shibboleth resources.

Project meetings were used at regular intervals to provide all parties with information on how the project is progressing, steering group meetings also took place to make sure the project work packages were staying on track.

Implementation

As described in the methodology section, we decided first to test Shibboleth just at Kidderminster College, this involved setting up two of the three servers we had purchased as part of the project, one was to be an IdP, the other an SP. Initially we ran into a number of issues regarding certificates, we used self signed certificates but did not reflect this fact in the Shibboleth trust file, this took a while to resolve, and looking at the Shibboleth mailing lists it was clear to see we were not the only ones hitting this problem. The IdP at this time linked to a simple Apache Basic Auth file, this was chosen simply because it was explained as an example in the official Shibboleth guide.

The SP was a little easier to setup, we managed to find a guide on this written by an American University, this helped with the compilation of Shibboleth and all its prerequisites, though we still had to take a while to resolve the issue with the trusts file.

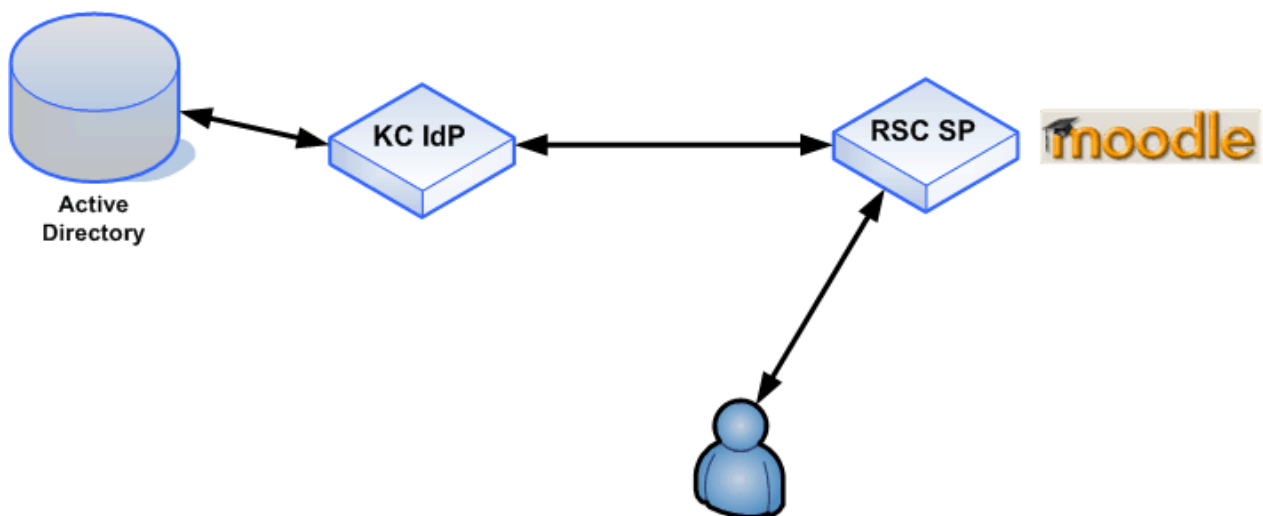
Following the official guides was not particularly useful at this time, it was more a booklet with information on how it works and what the configuration files do, which of course is very useful, however, it lacked clear information on the actual install process. This made us realise the importance of our install guide, and would allow ourselves to build Shibboleth much quicker throughout our project as well as help others install Shibboleth with greater ease.

So early in the project we had a working Shibboleth framework, even if the servers were both sitting next to each other. Our next step was to move the SP to the RSC West Midlands, where we would eventually install and Shibbolize Moodle. As an exercise to test and refine our install guide we decided to install and configure the SP from scratch, this was very useful as it ironed out a few problems with the guide. This method did however mean the first work package took a little longer than was planned, though putting this work in now probably saved time in later packages.

Once the RSC-WM SP was installed we first protected some simple web files to check Shibboleth was doing what it should, our tests were successful, so our next step was to install and secure Moodle. Moodle is built in a modular way (the M in Moodle stands for Modular), so after studying the code it was clear that we needed to create a Shibboleth authentication Module, this would just extend the authentication process so it could use the attributes in the header. Our Shibboleth Module required very little code, as the module assumes the user has already authenticated (they must have authenticated by Shibboleth else they wouldn't be able to get to the login script), the rest of the code to login/create the user was already built into the Moodle login process. Looking back, we did not approach the securing of Moodle in the best way, the way we achieved to protect Moodle was to use Shibboleth on the whole Moodle site, using HTTPS on every page slowed the site down a little and was unnecessary, later we were to realise that only the

Shibboleth authentication Module would be required to be protected by Shibboleth, another benefit of this method allowed manual logins, previously it was Shibboleth logins or nothing.

Clearly at this point anyone wanting to access our Shibbolized Moodle would have to login to the only IdP in the federation, this prompted us to hook our IdP into a Active Directory using the Apache auth_ldap module, this process was fairly simple as many people had already asked questions about it on the mailing lists. We decided early on in the project that we did not want to implement the Eduperson schema in our production Active Directory, for the simple reason that we didn't want any of the live system being effected by any errors that may arise from doing this. It appears to me that this could be an issue for many institutions, especially if this becomes a requirement to participate in a federation. For a while we just populated a number of test users in the Kidderminster College Active Directory store to allow access to the RSC-WM Moodle, their IdP was to be created at a later date.

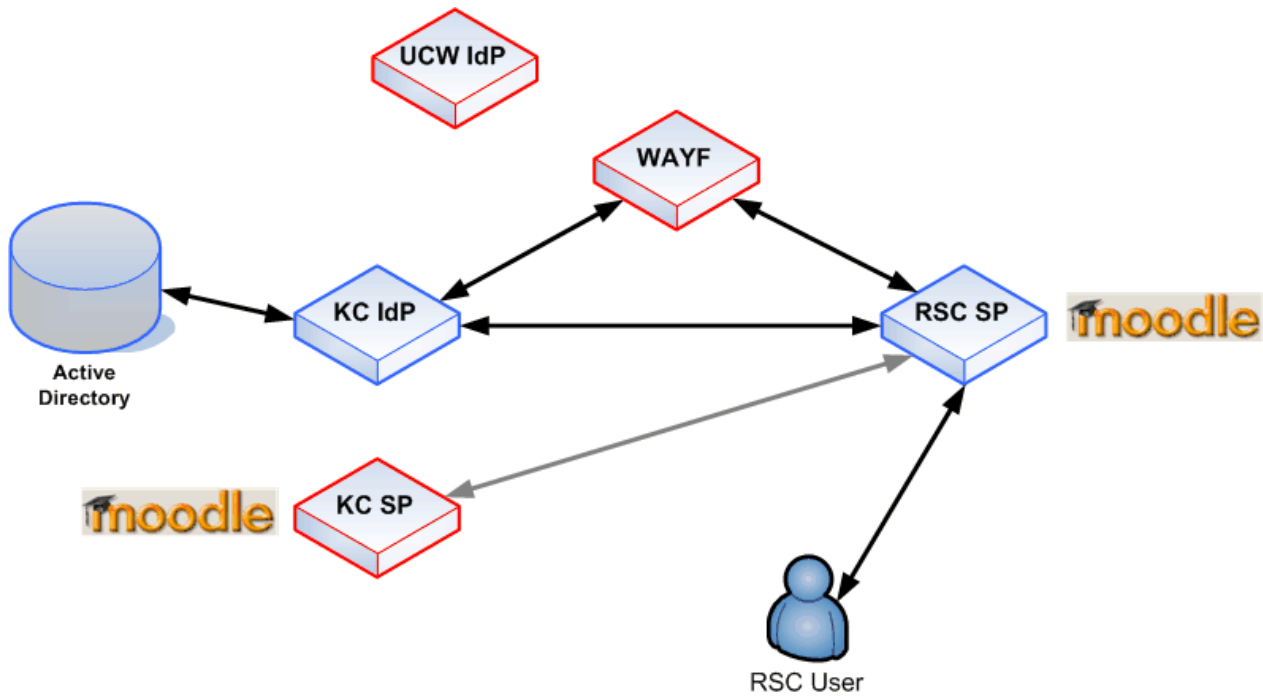


The early stages, one IdP, one SP and a Moodle.

Creating another IdP was important as the whole aim of this project was to provide a way for students and staff to use their own logins on another institutions resource. We decided to implement the IdP at the University of Worcester so we could test it with some of their students/staff, the RSC WM don't actually use their own attribute store, we would have to create a test LDAP server at a later date to allow them to login.

Installing another IdP was very easy as we could just follow our install guide, initially this IdP was to just use basic Apache auth, as the LDAP server details were unknown, however this was never updated as this information was never supplied despite persistent requests. The introduction of a second IdP inevitably meant we would have to setup a WAYF for users. This was easy to setup, requiring only a few extra commands, this was documented in the guide.

To test Moodle further we implemented an SP at Kidderminster College, this was to explore how linking courses and their resources across multiple Moodle's would work. We setup Moodle exactly the same as at RSC WM, once a user had logged on they could seamlessly move from one VLE to another, this showed how students could use the technology to share courses with partner institutions.



Added another IdP and SP, and introduced a WAYF.

We found out during the project that a Shibboleth/PERMIS implementation was already being developed by another Middleware project (the SIPS project), a decision was made to use this software in our project instead of developing our own, the SIPS team actually produced PERMIS so would achieve their goals much quicker than we could. This allowed us to spend more time on our other work packages. Research into how PERMIS works was still an important endeavour, this gave us a better understanding of how it will be integrated with Shibboleth later in the project when the SIPS team release the module.

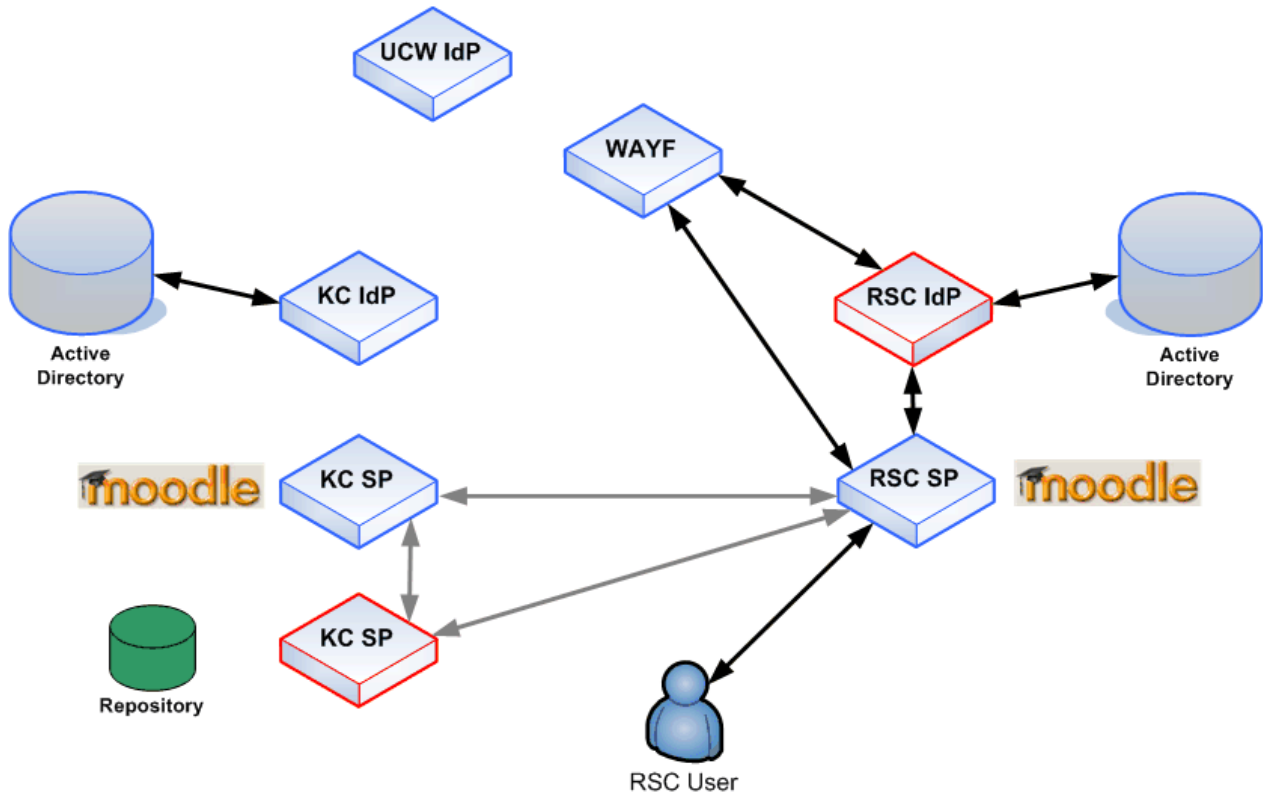
An important part of the project was to implement and securely share a repository, we chose a repository written in Perl by Rob Talbot of City College Coventry, it was agreed that we could test the repository for free. Another important part of the project was to not be tied down by one operating system, we wanted to see how Shibboleth would run on a windows system, up till now all our install were done in Linux. We decided to install the repository target on a Windows machine, using IIS instead of Apache was also decided as we thought a lot of Windows administrators may not have skills in Apache as they would more likely use the built in IIS web server. It was decided at his point that installation of a Windows IdP would be done later in the project, so as not to effect any current work packages, at this time nobody appeared to have installed an IdP on a Windows server without using Apache, this could have taken up more time than we had originally allotted.

Installation of the SP software on Windows is very easy, the package can be downloaded as an executable, meaning the user does not have the worries of compiling and configuring the code to get it running. The configuration files themselves were also easy to setup, the trusts and sites files were just copied to the machine after the new SP details had been added centrally (at Kidderminster College).

It was later discovered when trying to put some authorisation rules in that IIS and Shibboleth is missing some very important features, for example in Apache you can state which attributes a user must have to access a particular directory or virtual host, in IIS this cannot be achieved at all. This makes IIS as an SP very limiting indeed.

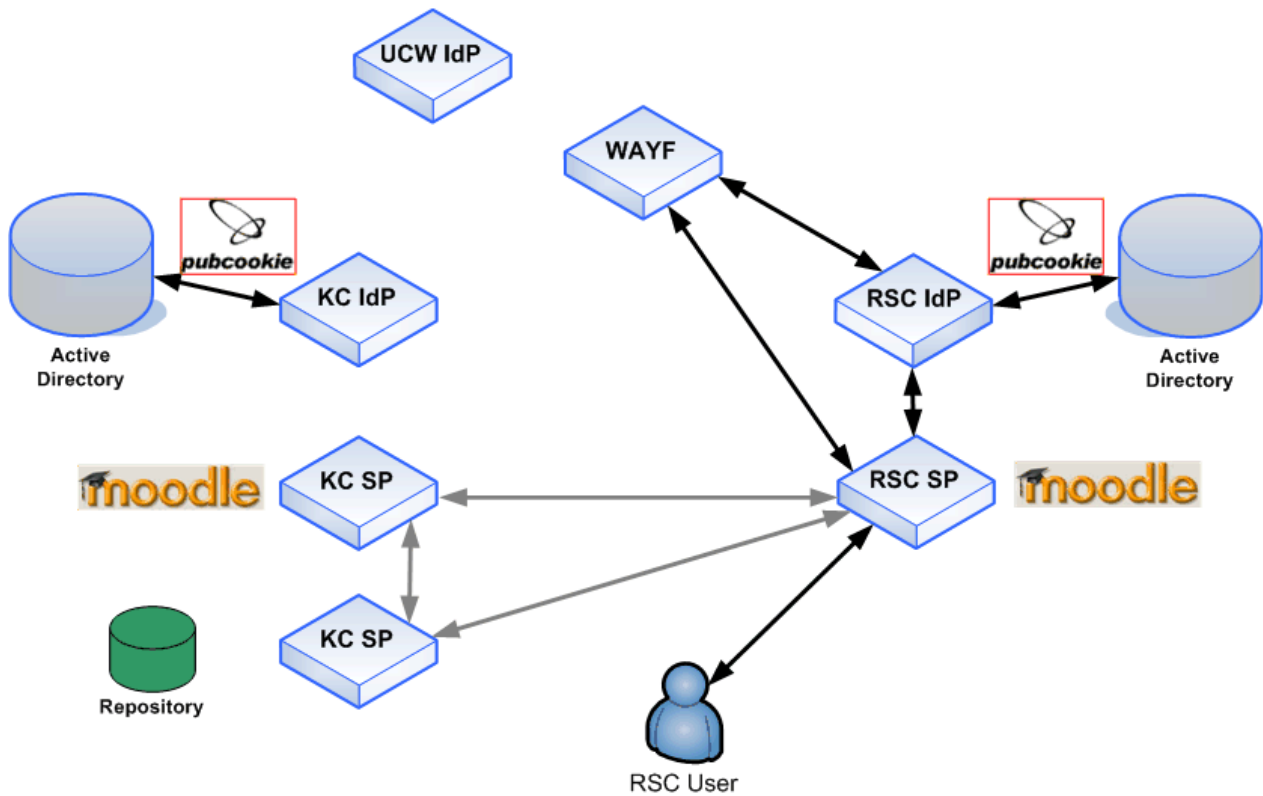
After implementing the repository into the federation we added an IdP at the RSC WM, we wanted to have at least to IdP's linking to a user attribute store, before this only Kidderminster College were linking to their user store, UCW was still linked to basic Apache authentication. A test Active Directory was used at the RSC WM end, settings were almost identical to that of Kidderminster College, however we did come across an issue with Base DN's in Windows Server 2003. The issue meant that a specific OU must be entered as the base DN and not the top of the domain tree (e.g. DC=mydomain,DC=local). After doing

some research it turned out to be a referral issue, though this still did not fully solve the problem, after some discussion on the Shibboleth mailing list we discovered it was also a DNS issue, see Appendix B for the solution to this issue, I fed the solution back to the list in the event that another administrator comes up against the same problem.



Repository, SP and RSC-WM IdP added.

The only step now required to get our test Shibboleth federation as we intended (excluding issues with University of Worcester) was to implement Pubcookie SSO on the IdP's. Whilst SSO is achieved through Shibboleth itself, by using Pubcookie you have the added advantage of providing SSO on none shibbed resources (but protected by Pubcookie authentication), also in addition to this the interface for logging on is more user friendly, now in the form of a web page, as opposed to the popup produced ordinarily. To install Pubcookie we used the excellent guides produced by the IAMSECT project.



Pubcookie added.

Once we heard news of the PERMIS/Shibboleth module being released we began implementing it. In our original plan we were to use the fine grained controls provided by PERMIS to enhance the authorisation side of Moodle, it was decided however that we should test the PERMIS/Shibboleth solution in a more simple setup first allowing us to evaluate the usefulness before spending lots of time on attempting something complex, only to find out later it is not a feasible solution.

To test PERMIS we used Pebble Blogger software, our simple example allows a staff member to create and read posts, whilst students can only read posts. Values in the AttributeCertificateAttribute certificate determined whether the user is a member of staff or a student.

During the install process we came across numerous issues with the apache module, but these were promptly fixed by the team at Kent University. Previous research allowed us to have a good understanding of the RBAC system, but the install and setup of PERMIS was still quite complicated.

The complex setup, configuration and maintenance of PERMIS seemed to offer little advantage to the kind of setup we would use here at Kidderminster College for e-learning, using a role based PMI would also require extra attributes to be setup in our LDAP server, and extra processes would be required to be carried out (signing certificates etc), it was agreed between us and our partners that this extra complexity would not benefit us for what we are currently doing.

Another issue raised was the fact that although the roles can be very granular, the actions are based on those imposed by HTTP, for example get and post, this meant we could only prevent users based on their roles for specific get or post functions on specific pages, more granular processing cannot occur unless specific code changes are put in place.

It was clear that the University of Worcester required more testing, up till this point their involvement was quite limited, it was planned that we would shibbolize a concurrent version of their repository on the same box as their production repository, this way we could interrogate the same store of objects making it a more real experience to the user than you would get connecting to a test repository with limited items in it,

whilst still maintaining their current service in its normal state. This however was not possible as we could not obtain root access to the server to get the Shibboleth target software installed. To get around this issue, we agreed to install a separate box at the University of Worcester site with the target software on already, the only down side to this is that we wouldn't get the real user usage that we wanted, in effect this is just another test box. The repository at Kidderminster College we realised would prove to be more useful, as it will be used as our production repository for all students and staff to create and view learning objects.

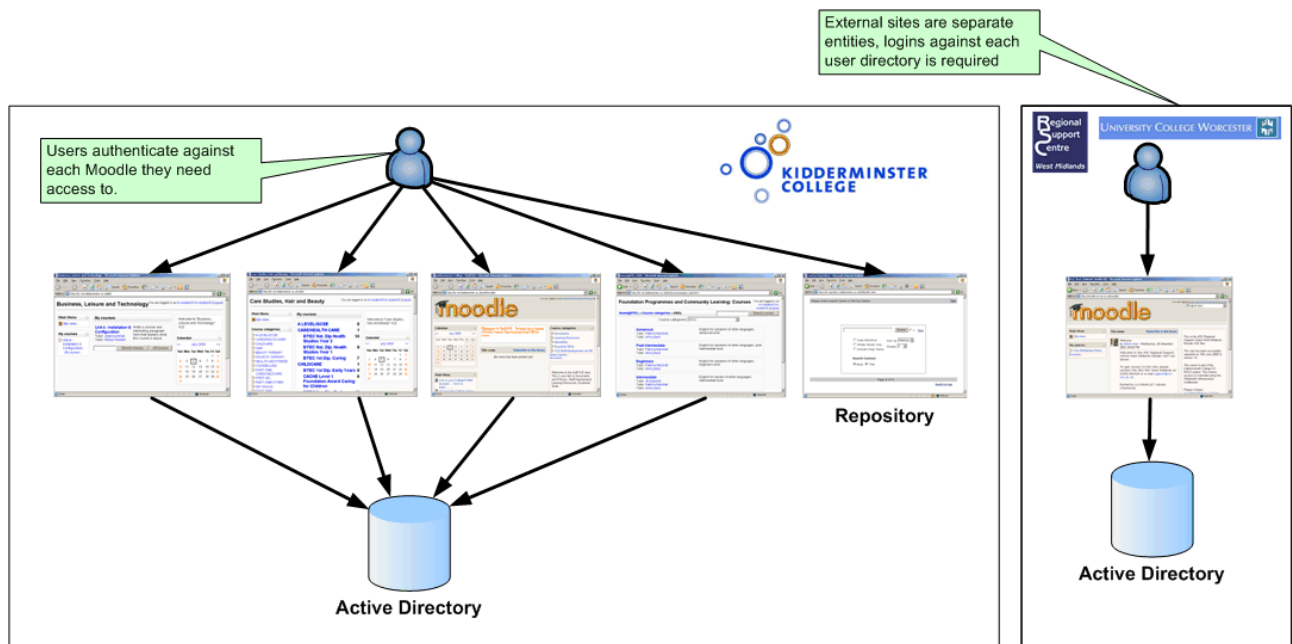
In the end UW decided that they would not find the Shibboleth access useful enough to warrant a change to their repository, this we believe was down to lack of understanding and trust in the technology.

The first part of the second year of the project was to upgrade the version of Moodle to 1.5, which provides Shibboleth support out of the box, this is a more comprehensive version of what we had implemented ourselves last year. We also wanted to upgrade the Shibboleth SP software itself, at the time 1.3 was not out, so we upgraded to 1.21a.

The new Shibboleth module in Moodle allowed us to map attributes to all of the properties of a Moodle user, it also allowed manual logins and shib-enabled users to share the site, the previous implementation only allowed one or the other, which limited its use somewhat.

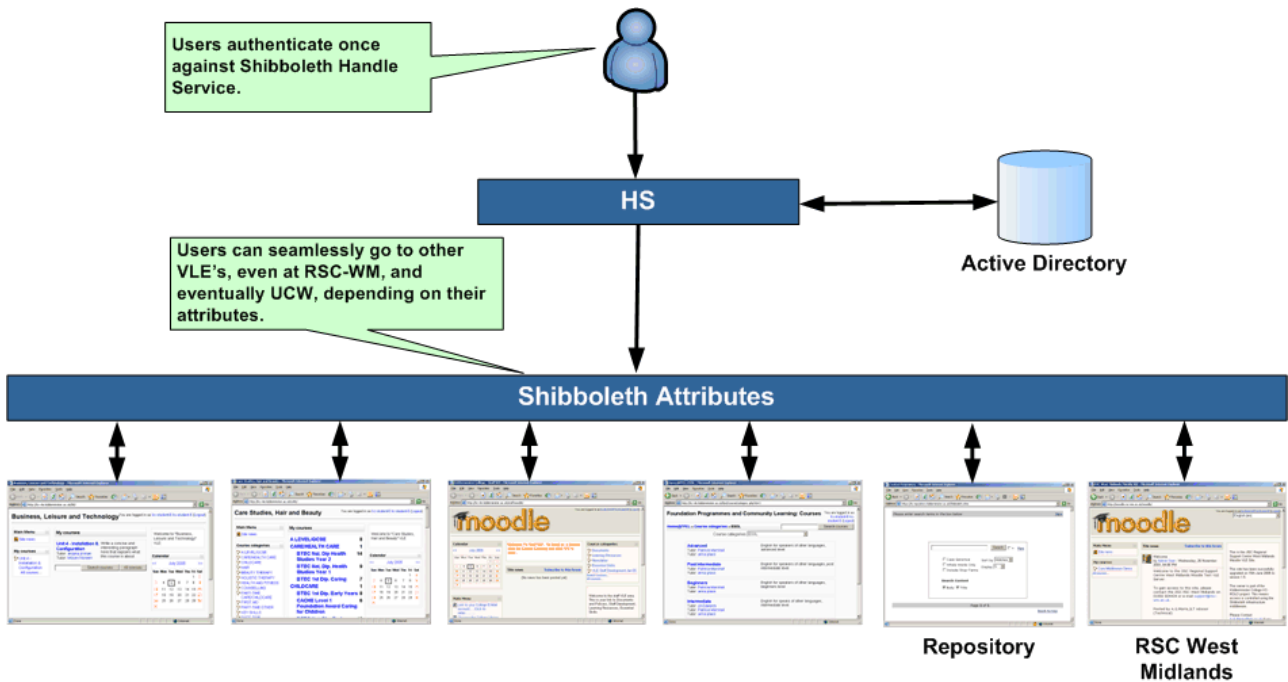
The first year of our project we setup a Moodle server which we used for testing, now we have experience using Moodle with Shibboleth we decided to setup an exact replica of our current production VLE.

At Kidderminster College we originally had Moodle setup in a way which gave each section its own instance of Moodle, previously this meant that when a student needed to access a course from another sections Moodle they would be requested to login again (see diagram below), naturally this was not a perfect solution.



Users previously had to log on to each Moodle separately

By Shibboleth enabling all these instances of Moodle, students are now able to jump across Moodle sites with only one login, they are also able to access our repository, the RSC-WM Moodle site, and a test UCW repository (see diagram below), all assuming they have access of course..



Now using the single sign on capabilities Shibboleth provides, not only can inter institutional sharing be improved, but also intra institutional.

Each instance was protected by the department attribute (remember we haven't implemented the eduperson schema), to see how we protected each site in Apache see Appendix C.

A portal was also developed, this was protected by Shibboleth, once logged in they can access all of the other Shibboleth resources without requiring a login. Initial tests have proved very successful.

As part of the development of the portal we decided it would be a good time to get some third party certificates, as a result the federation now supports certificates signed by Global Sign, the main reason for us doing this was to remove the popup which is shown to users as they visit the sites we have protected, this is particularly important now our Shibbolized Moodle server is going into production.

Our final aim of the project was to implement a Windows IdP, we did this in Windows Server 2003 using IIS 6 and Tomcat. We configured the HS to map directly to IIS which was protected by "Integrated Windows Authentication", this was then passed onto Tomcat, whilst the AA was bound to Tomcat. Users currently logged on to the network internally can now be automatically logged onto any Shibboleth resource once the institution has been chosen in the WAYF. A small code change was required to do this, the username IIS returns is preceded by the domain name and a slash, subsequent searches in LDAP by the AA would fail as the user "domain/username" will not exist, the code change simply removes the "domain/" prefix from the username used by Shibboleth.

The documentation has evolved over the course of the project, all notes from each work package are being integrated into the final packaged up guide. The guide has been written to not overcomplicate the install process, but at the same time, we have tried to avoid scripting any commands to allow the installer to understand what they are doing, we are very conscious of the fact that scripting can hide important issues from the system administrator. The guide was initially based on Fedora Core 1, and has been updated as each new version came out, we have recently updated it to Core 4. SELinux changes a little in each iteration of Fedora, so this was the area which changed the most in the guide on the last OS update.

Dissemination Events

Kidderminster College Moodle Moot

We held a regional Moodle event at Kidderminster College, we did a presentation on the work we have been doing with Shibboleth and Moodle, describing the advantages of a Shib enabled Moodle and the basics of how it will work, there was a lot of interest in the technology.

Dublin ALT Conference

In April 2005 we did a Shibboleth & Moodle hands-on workshop at the ALT Spring Conference and Research Seminar 2005, this started with an overview of Shibboleth, what it is, where it can be used, and how it works. In addition to this, we wanted everyone to take part in a live example, we created 20 shib enabled accounts back at Kidderminster, the users then tried to access the Moodle website at RSC-WM, they were then consequently forwarded to the WAYF, where they entered their assigned username and passwords, and forwarded again to the site itself. Users were then linking items from the Kidderminster repository into their course at RSC-WM without any additional authentication.

We felt showing an example of Shibboleth in action was worth while, as many presentations and papers have been written on the way shibboleth works, the very nature of Shibboleth being almost transparent to the end user can cause people to think more about the application they are looking at, as opposed to seeing the power and usefulness of the technology allowing them to view the pages they are on, this is definitely a good thing, but does make it harder to get the message across during workshops.

Birmingham

We did a presentation similar to previous ones at a JISC conference in Birmingham on April the 6th ..

Loughborough

In May we attended the Core Middleware meeting and Early adopters meeting the following day at Loughborough, we did the same presentation on each day, showing the attendees an overview of the KC-ROLO project, describing the federation we have running and the IdP's and SP's we have in place.

After this presentation we showed a live example of the linking between our repository and the Moodle's as we have done before, but in addition to this we showed an example of a PERMIS protected Web Blogger, though it was a simple example, the way it worked could be seen clearly.

Cambridge

In July we did a presentation and demonstration at the JISC joint meeting at Cambridge, at this event we focussed on how we are using Shibboleth at Kidderminster, in particular showing how Shibboleth is actually enabling us to improve our Intra institutional sharing as well as inter institutional by allowing all our internal VLE's to talk to each other as well as protected sites from other institutions.

London Working Mans College

In March 2006 we presented our Moodle/Shibboleth setup at a Moodle moot at the London Working Mans College.

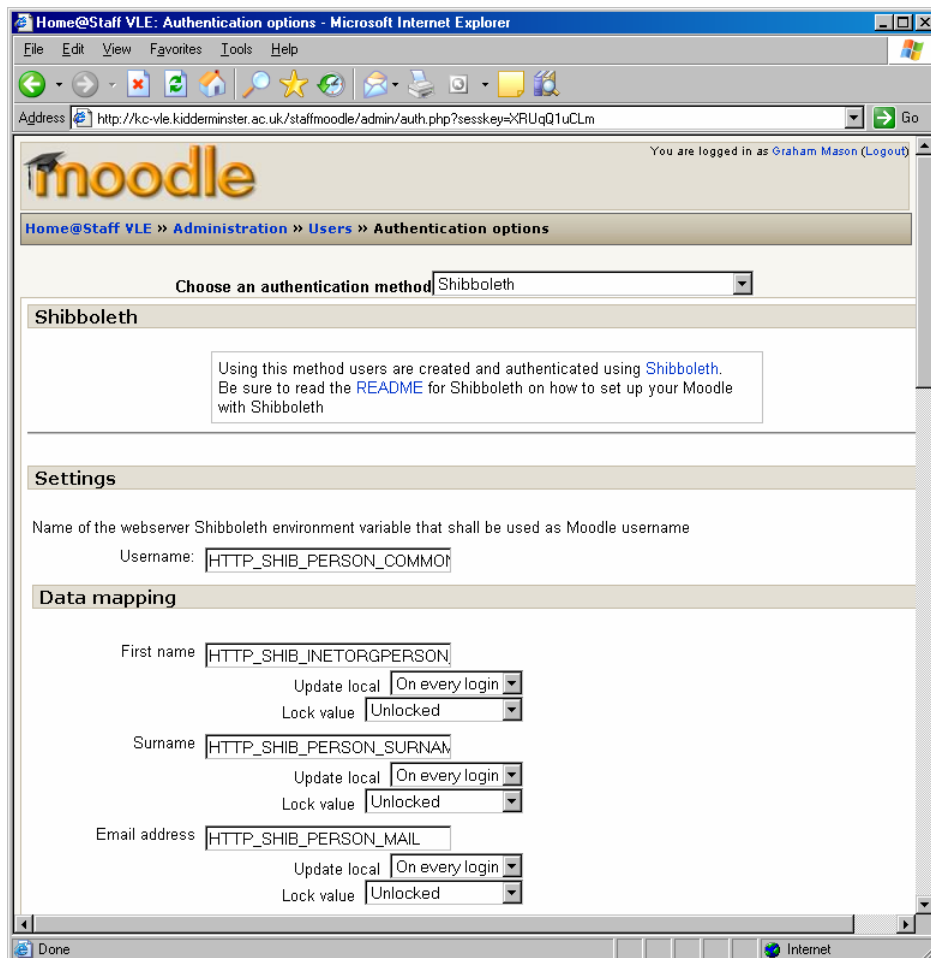
Outputs and Results

When compared to our aims and objectives the outputs of the project were comprehensive, though not all were completed exactly as originally planned.

Our first and main aim was to share learning resources between Kidderminster College, RSC West Midlands and University of Worcester via Shibboleth architecture and Moodle Open Source VLE, this framework was fully implemented between Kidderminster College and the RSC-WM, with the minimum requirements for this at each site, this being an IdP, an SP protecting a shibbolized Moodle and at Kidderminster College an additional shibbolized repository. The lack of involvement by the University of Worcester did not have an effect on the project in any detrimental way, we now know that the technology works and could implement fairly painlessly an IdP and SP at their site to allow them to join the sharing process.

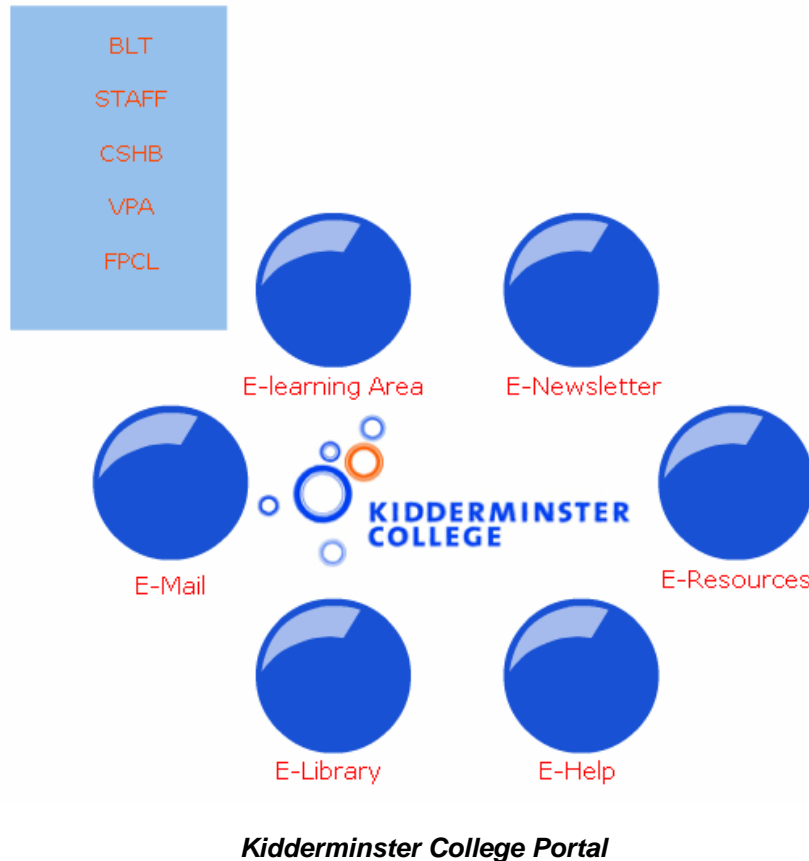
Many people have now logged into either the RSC-WM and Kidderminster College VLE's and repository with positive comments, it is fair to say that nobody wants to type in multiple passwords when they don't need to, we feel this will break down at least one of the barriers staff and students have to using online learning resources.

With hindsight our own Shibboleth authentication module was fairly primitive, though it did work well and was being used a long time before the official version which was bundled with Moodle from version 1.5 onwards. The main disadvantage of ours was that it lacked configurability within Moodle itself, the screenshot below shows the power of the new module and its ability to allow the administrator to easily choose what attributes it wants to map to which Moodle user attribute.



Latest Moodle Shibboleth authentication module.

To make our multiple VLE's and repository easier to move from one to the other we created a simple portal, this is essentially a list of links which go to each resource, however by protecting the whole portal with Shibboleth the staff or student can then access all of the links with no extra logins, coupled with our Windows IdP (described later in this section) users of this new system are very impressed.



Kidderminster College Portal

The PERMIS objectives of the first year, namely the development of the module were clearly not met by us, but rather the SIPS project team, this we feel was a good thing, had we had to do this ourselves the project would definitely have slipped, affecting all the other projects aims and objectives. We believe the complexity of the PERMIS engine and the development of the module for Shibboleth would have been above our ability as a team, just configuring PERMIS was hard enough in the initial stages.

The guides objective changed slightly throughout the project, it was clear to us that creating a guide that was all encompassing of common operating systems would have to be so generic that it would be useful to few readers, much like the original install guide produced by the Shibboleth team. Creating a separate guide for every operating system was also impractical, because of this we decided to stick to just one platform, Fedora Core, chosen because it is what we are familiar with already. The guides install sections are effective and make use of downloading large chunks of configuration files to reduce installation time (useful in the event where the text cannot be copied and pasted into the config file directly). The install section could easily be ported to other Linux distributions easily, we believe a Linux administrator familiar with another distribution could translate any areas that may differ to Fedora specific commands and directories.

One unexpected outcome of the project was the federation itself, whilst we knew we would end up with multiple sites in our federation it wasn't clear how important this would be to the overall project and Shibboleth framework, initially I think we just saw it as a set of IdP's, SP's and a WAYF, when clearly the driving force behind this all is the federation itself, defining the policies to which all other components must follow, affecting not only configuration of the components, but the user attribute stores as well, this can have far reaching consequences on an institutions implementation. You only had to go to a JISC programme meeting to see the importance of Federations, it has become a real hot topic and one we

expect to continue to grow.

Developing Moodle to read Shibboleth course attributes was another area we didn't pursue due to time restrictions and the fact that for us this may not even be the best solution. At the time we were using manual enrolments with pass keys, this allows the student to enrol on a course if they have been given the key by their tutor. We already have plans of enrolling students using external database methods, to start work on a Shibboleth enrolment module would effect this somewhat, and would require us to move our course attributes (which are not in our user attribute store) into Active Directory. In the future however such a module may become useful, and I have already seen a post on the Moodle forums suggesting they are to develop such a solution.

Our last deliverable was the implementation of a Windows Server 2003 IdP using IIS and "Windows Integrated Authenticated", this allows authentication to Active Directory using the credentials of the logged on user, the mechanism only works whilst the user is logged onto the internal domain, so would not provide automatic logon whilst at home, however the result is the user is still only logging on once (once to IdP login prompt whilst at home, and once at initial windows login internally).

Outcomes

The methodology of the project worked quite well, simply evolving the components of the federation, for example adding IdP's and SP's at specific times, only when previous components had been tested first.

Students and staff now have the ability to span internal resources without multiple logins, with the addition (unlike normal SSO's) of being able to access external resources in a completely secure manner by authenticating back to their own institutions IdP.

In April 2006 we will roll our Shibboleth solution out for staff and students, this being the central shib protected portal, departmental Moodle VLE's and staff extranet Moodle and our shibbed repository. This solution will provide all users with the ability to use these resources with only one login per session (including initial Windows login if used internally). We feel this will make access to the VLE's so easy that users will have no reason not to use them for every day learning activities.

University of Worcester students will not be able to login to our VLE currently because their IdP is not linked to their central user store for reasons mentioned previously. Whilst this is a shame as it means our only external resources are essentially for test purposes, it does give us the framework which we can easily build on should the time come when the University of Worcester or another partner institution would like to share resources with us.

The RSC-WM are very happy with their developmental Moodle, many regional members of the academic community have logged in via Shibboleth onto this Moodle. Even though the users are not there specifically to look at Shibboleth we think it is a good thing that users in most cases do not see it as important (from their perspective as a tutor for example), whilst this may sound strange the best solutions really are the ones you can't see or don't think about, that is to say they don't get in the way of the primary reason you are there, to learn.

Our install guide has evolved from a simple 3 page document to a guide nearly 30 pages in length. Readers should be able to get an understanding of how Shibboleth works in the initial pages, then follow the actual install documentation to get it working. Unfortunately all our work with Shibboleth is with version 1.2, this means some of this guide is not applicable to the latest version 1.3 release, perhaps in the future we will update this.

Conclusions

It should be remembered that our ultimate objective was to get staff and students using learning resources in a collaborative way, regardless of institutional boundaries and other technical issues such as multiple logins, Shibboleth was our chosen tool to achieve this goal, and we feel it has fulfilled this task admirably.

Using Shibboleth not only opens up possibilities with inter-institutional access to resources, it gives the user a single sign on system which can be used for any Shibboleth protected resource. At Kidderminster College we have used this to our advantage, Shibboleth is also being exploited for its use as an SSO system for resource access. Though you could say that this is overkill for such a role it does give us the ability for us to share with external users using the same implementation with no modifications, protecting these resources with CAS or Pubcookie for example would be short sighted we believe given our circumstances regarding sharing our VLE in the future with partner institutions.

It is important to see the effectiveness of Shibboleth in a system like this, for example a large University with many campuses may implement an internal university wide federation allowing them to link up all their disparate authentication systems into a common secure framework.

We as a college will continue our use and development work with Shibboleth, looking forward to the possibilities it can bring to our staff and students in the future.

Implications

Whilst the work we have done is by no means unique, we hope it shows that Shibboleth is a suitable tool for sharing resources securely, even for a relatively small FE institution. Following our example we hope other institutions will realise its potential and see it is not just for HE and large corporations, with benefits for all, big and small.

We see future work with Shibboleth being mainly in the federation area, the KC-ROLO federation evolved out of necessity, we currently don't have the policies written and laid out like some of the bigger federations already in use in Finland for example. This is a large area and one we wish to contribute to in the future.

References

Pubcookie install guides

<http://iamsect.ncl.ac.uk/deliverables/>

Appendixes

Contents

- A. KC-ROLO federation overview.
- B. Active Directory IdP Base DN issue.
- C. Apache directives.
- D. User install guide.

Appendix B – Active Directory IdP Base DN issue.

Initial request:

Hi,

We have an issue connecting our AA with Active Directory (on Server 2003), the issue arises when we try and do searches on the base of our domain (e.g. DC=kiddercoll,DC=local), it produces an error saying it cannot find the user.

After looking in the archives it was suggested to use `java.naming.referral` with the value `follow`. Whilst that appeared to work for that particular person, it has not resolved our problem.

I read another thread on this issue which pinpointed the error to DNS naming, this also does not resolve our problem. We are currently connecting to the global catalog as a work around, however this is only a short term fix as not all the attributes we may need are available from it.

Settings of our `resolver.ldap.xml` which fails (port 389):

```
<JNDIDirectoryDataConnector id="directory">
  <Search filter="sAMAccountName=%PRINCIPAL%">
    <Controls searchScope="SUBTREE_SCOPE" returningObjects="false" />
  </Search>
  <Property name="java.naming.factory.initial" value="com.sun.jndi.ldap.LdapCtxFactory" />
  <Property name="java.naming.provider.url" value="ldap://172.16.0.3:389/DC=kiddercoll,DC=local" />
    <Property name="java.naming.referral" value="follow" />
  <Property name="java.naming.security.principal" value="CN=ldapauth,CN=Users,DC=kiddercoll,DC=local" />
  <Property name="java.naming.security.credentials" value="xxxxxx" />
</JNDIDirectoryDataConnector>
```

Has anyone got a solution for this?

Thanks

Final resolution after research:

Thanks for the replies, it was a DNS issue in the end, as the server is querying against our external DNS server there are some dns entries missing for the referral part of the search (other partitions in AD), to find these out i did an ldap search, the output of which displays the search referral dns names being used. In our case:

search reference

```
ref: ldap://ForestDnsZones.kiddercoll.local/DC=ForestDnsZones,DC=kiddercoll,DC=local
```

search reference

```
ref: ldap://DomainDnsZones.kiddercoll.local/DC=DomainDnsZones,DC=kiddercoll,DC=local
```

search reference

```
ref: ldap://kiddercoll.local/CN=Configuration,DC=kiddercoll,DC=local
```

To fix the issue i simply put `kiddercoll.local`, `DomainDnsZones.kiddercoll.local` and `ForestDnsZones.kiddercoll.local` in the servers hosts file. So even if you are querying your DC using an IP, the referrals in the background will use names.

Thanks again for the info.

Ed.

Appendix C – Apache directives

```
<Location /blt/auth/shibboleth/>  
AuthType shibboleth  
ShibRequireSession On  
ShibRequireAll on  
require commonName ~ .@kidderminster.ac.uk  
require department ~ .000KC3. [sS]taff  
</Location>
```

```
<Location /cshb/auth/shibboleth/>  
AuthType shibboleth  
ShibRequireSession Off  
require shibboleth  
ShibRequireAll on  
require commonName ~ .@kidderminster.ac.uk  
require department ~ .000KC4. [sS]taff  
</Location>
```

```
<Location /vpa/auth/shibboleth/>  
AuthType shibboleth  
ShibRequireSession On  
ShibRequireAll on  
require commonName ~ .@kidderminster.ac.uk  
require department ~ .000KC2. [sS]taff  
</Location>
```

```
<Location /fpcl/auth/shibboleth/>  
AuthType shibboleth  
ShibRequireSession On  
ShibRequireAll on  
require commonName ~ .@kidderminster.ac.uk  
require department ~ .000KC1. [sS]taff  
</Location>
```

```
<Location /staffmoodle/auth/shibboleth/>  
AuthType shibboleth  
ShibRequireSession On  
ShibRequireAll on  
require department ~ [sS]taff  
require commonName ~ .@kidderminster.ac.uk  
</Location>
```

```
<Location /portal>  
AuthType shibboleth  
ShibRequireSession On  
require valid-user  
</Location>
```

Appendix D - Shibboleth User guide

See attached document, or visit <http://kidderminster.ac.uk/kc-rola> for the latest version of our install guide.