Using Role Based Access Control in the Sakai Collaborative Framework

Rob Allan and Xiaobo Yang
STFC, e-Science Centre

e-Mail: r.j.allan@dl.ac.uk

March 7, 2008

Abstract

We document issues of role-based access control showing requirements and examples of how it can be implemented usin Sakai as a gateway to in an e-Research Infrastructure. We show what works and how to use it based on a number of project we are currently undertaking.

Nearly all of what is described in this document is also relevant to using Sakai in a Learning and Teaching, or indeed any other environment.

1 Introduction

For a definition and background to Role-Based Access Control, see Wikipedia `http://en.wikipedia.org/wiki/RBAC`.

1.1 e-Research Projects and Requirements

We describe how we are using the Sakai Collaborative Framework as a Science Gateway for several research communities, including UK e-Social Science [9], the North West England regional Grid [10], and the European Psi-k Network of Excellence [11] – the latter having up to 1,900 users. The Gateways are, at the time of writing, based on Sakai 2.4.1 with additional tools to support access to relevant data and Grid resources. They are being designed to meet the requirements of all the projects involved using an e-Infrastructure for research purposes. A number of consultancy excercises were completed to obtained the detailed set of requirements [2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 15], some of these were based on our previous JISC-funded work on Sakai [16].

Research resources comprise of software, Grid-enabled datasets and the infrastructure on which these will reside. They also comprise of people, i.e. the researchers, students and consumers of the research outputs. We stress the importance of enabling people to access the software, data and computer systems and to be able to collaborate and interact in a rich variety of ways.

The user environment being developed to interface this e-Infrastructure is comprised of a portal, a service registry and workflow tools. The portal offers collections of tools and services appropriate to the needs of different researchers working in virtual organisations (VOs). A VO represents a particular grouping of users with a particular set of requirements, in this case working together on a specific research question. Users will typically belong to more than one VO. Each VO has its own "worksite" on the portal that provides easy access to a collection of applications, data and information resources. Thus the worksite provides the security "context" or "realm" which we will discuss below.

A number of tools are generic to all VOs, such as wiki's, blogs and video-conference applications, but the content managed by the tool is specific to the VO in question. Many of the required tools are built into and released with Sakai, others were developed in previous JISC-funded projects [16, 18] and continue to be developed to meet specific requirements.

Clearly we need to manage large numbers of groups of users in a very flexible way. Figure 1 shows the proces of accessing content in a Sakai worksite, which is now described.

1.2 Role-based Security in Sakai

The Sakai portal framework has implemented a sophisticated internal security structure with role-based access control mechanisms at a fairly fine grained functional level. We here summarise how this is implemented and the procedures adopted in applying the rules. Sakai not only provides a portal client interface rendered as HTML, but also has kernel components which expose a Web services interface or WebDav. It also has a WSRP producer integrated in the kernel.

Sakai was designed from the outset as an "enterprise" server, so has the additional ability to link into external

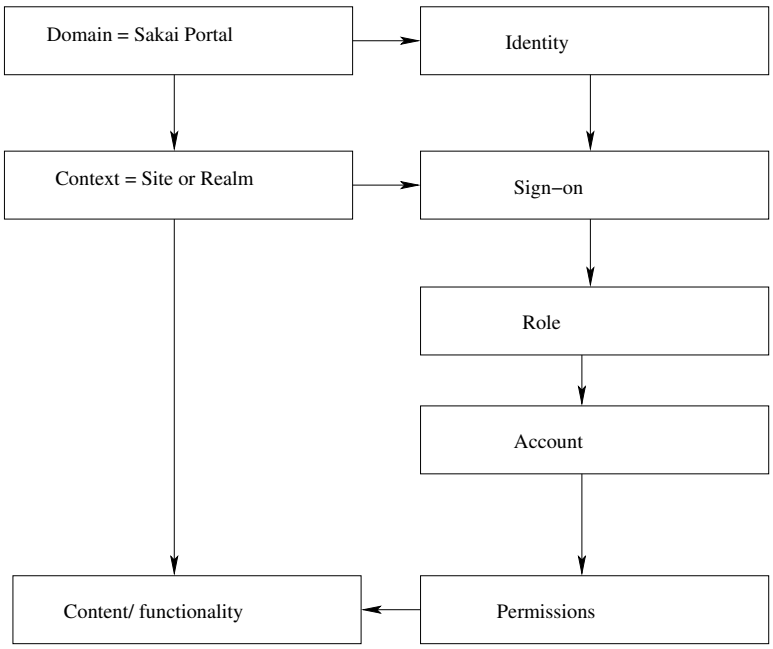

Figure 1: Context, Role and Account Relationships in Sakai

information providers.

Users: Each user of the portal framework has an account. Information stored includes [userid | first name | last name | e-mail | passwd | type]. We often use e-mail for userid, but it could also be the FedId. Validation of the userid/ passwd pair is the primary authentication mechanism. Clearly any other authentication can be used provided the user can then be mapped onto a Sakai account. The account is identified in the DB with an InternalId, either the userid or a unique string, depending on how it was created.

Types: A User can have a Type which is defined when the user is created. This has system-wide scope. The user's account is pre-populated using the template `!user.template.type`. Examples include "guest", "maintain".

Sites: A site, or "worksite" is a specific DB area for related content. It can be "public", which means it can be joined by any authenticated user, or it is "private" which means it is moderated by one or more users with "maintain" or equivalent role for adding participants to the site. Participants can be added with any of the roles which have been defined for that site; "access" and "maintain" are the default ones. A site is identified in the DB with a unique SiteId string. A site can also have a type, e.g. "project" or "course" are defaults.

Pages: A page within a site has an associated left hand menu button and is a view onto a set of tools and corresponding data. A page has a PageId which is a unique string.

Tools: Each page on a site contains one or more tools arranged in one- or two-column layout A specific tool has a ToolId which is a unique string. For instance "sakai.resources" is a resource folder and content management tool which can be configured for each site. Tools can be developed to apply permissions based on users' roles, see comments on functions and API below.

Role: A role is identified by a RoleId which is a name. Users' base roles are ".anon" or ".auth" depending if they have not or have logged into Sakai. Other roles can be defined (see below) and have scope within a site.

Realms: Realms are used to associate roles with users for a particular activity. A site realm is identified as `/site/SideId`. The definitions of the roles applies within a Realm. A Realm may contain many different roles; "access" and "maintain" are the default ones. Other types of realm are `/content/user`, `/content/group`. In addition to these specific realms, there are defaults such as `!group.template`, `!site.template`, `!user.template`, `!site.user`, `!pubview`, etc.

Function: Functions (or permissions) within a set for each role in a realm can be switched on or off. For instance a user with role ".anon", i.e. the public, will probably only have functions like "content.read" allowed.

Alias: In the system, an alias can be used to provide a user-friendly name for an Id. This is mainly for presentation purposes, as aliases can be ambiguous and referencing content by alias will not work. For instance `r.j.allan@dl.ac.uk` which is my UserId can be aliased to "Rob". Sites and other objects can be given an alias.

Group: Users can be put into named groups, currently only a few tools such as Announcement, Resources and Schedule plus some teaching tools respond to this. Note that `!group.template.type` would apply to a site of type "type".

Section: Course material can be put into sections.

To illustrate some of the complexity of the system we refer to the entity model diagram developed for the Rwiki tool at Cambridge, see Figure 2. Similar models could be developed for each tool and used to ensure consistency and appropriate behaviour within the framework. Note also that the upper part of this figure illustrates the relationships between realm, worksite, tool, user and role as already discussed. The concepts of group, owner, permission, page, content, lock are also shown.

Table 1 shows a subset of the functions associated with a few Sakai worksite roles. Another list (incomplete) of functions is given on Sakaipedia `http://bugs.sakaiproject.org/confluence/display/ENC/Permissions+list`.

In an implementation for a teaching establishement the roles could be extended to be: Affiliate, Assistant, Candidate, Instructor, Member, Observer, Student in addition to access and maintain (example from Chuck Severance). This is based on a knowledge of the stakeholders in such an establishment.

Site templates can be defined; the defaults are "course" and "project". Each template, such as `!site.template` has an associated realm containing a number of pre-defined roles, e.g. "maintain" and "access". `!site.template.course` has "Instructor", "Student" and "Teaching Assistant" pre defined. `!site.template.portfolio` has "CIG Coordinator", "CIG Participant", "Evaluator" and "Reviewer" pre defined. When a site is first created an appropriate template is chosen for it and these roles with all their permissions are inherited for the tools it will contain. Very similar templates apply to groups.

User templates can also be defined according to type, as shown above. With each template comes a realm and associated roles. There are two which are slightly different to site roles, namely ".auth" and ".anon". The ".auth" role, for an authenticated user, contains the possible "site.add" function which controls whether the user can create new worksites. This is included by default only in the `!user.template.maintain` realm.

Table 1: Example Functionality for Tools based on Sakai Worksite Roles.

Function/ Role	maintain	member *	access	.anon
calendar.all.groups	yes			
calendar.delete.any	yes			
calendar.delete.own	yes	yes		
calendar.import				
calendar.new	yes	yes		
calendar.read	yes	yes	yes	yes
calendar.revise.any	yes			
calendar.revise.own	yes	yes		
content.all.groups	yes			
content.delete.any	yes			
content.delete.own	yes	yes		
content.hidden				
content.new	yes	yes		
content.read	yes	yes	yes	yes
content.revise.any	yes			
content.revise.own	yes	yes		
disc.delete.any	yes			
disc.delete.own	yes			
disc.new	yes	yes	yes	
disc.new.topic	yes			
disc.read	yes	yes	yes	yes
disc.revise.any	yes			
disc.revise.own	yes	yes	yes	
rwiki.admin	yes			
rwiki.create	yes	yes	yes	
rwiki.read	yes	yes	yes	yes
rwiki.superadmin				
rwiki.update	yes	yes	yes	

* Note: “member” is not a default role, but has been added to illustrate how it can be used.

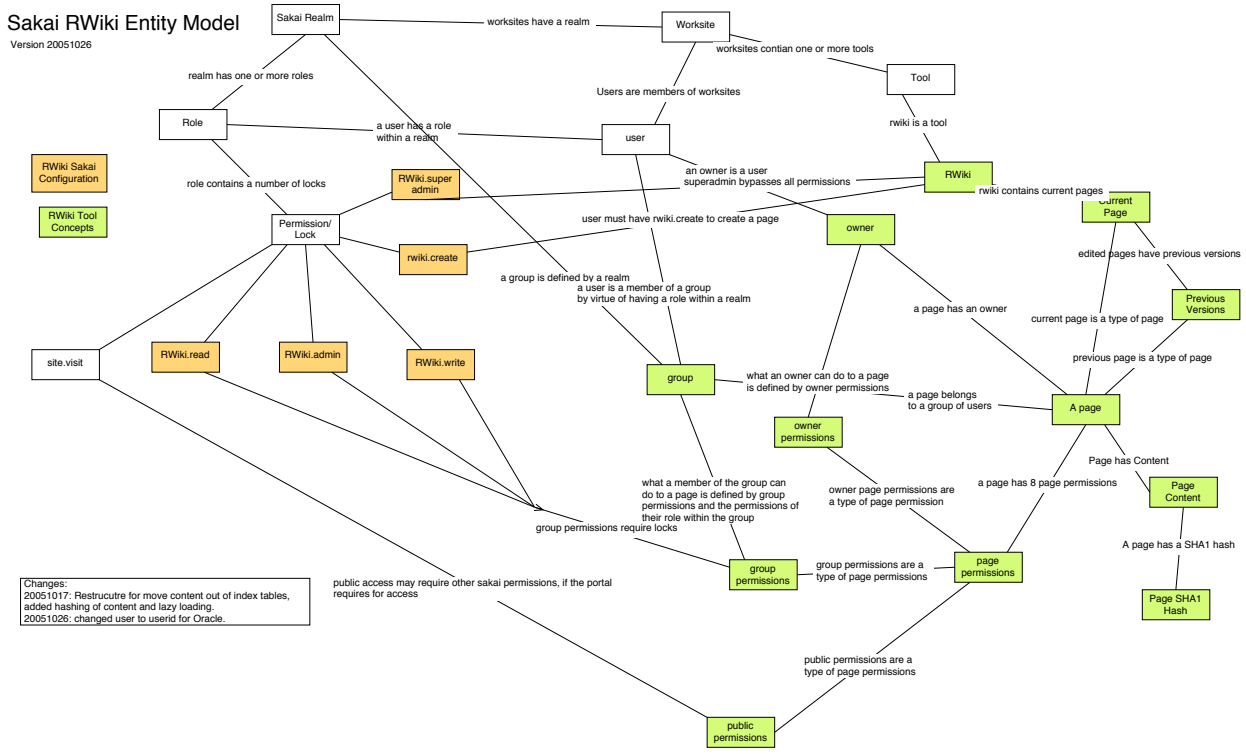

Figure 2: Rwiki Entity Model for Sakai

How are the rules applied? Hierarchy?

TBA

Comment: information about `!site.helper` roles, maybe too complex for this doc so is currently not included.

2 Sakai meeting our Requirements

2.1 Tricks with Roles and Permissions

Publically viewable content and the Gateway site

The Gateway site is what a Sakai user sees when they access the portal server before they log in. We will use specific examples in the rest of this document to illustrate the points being made. For NCeSS: the UK's National Centre for e-Social Science, the gateway site is at `http://portal.ncess.ac.uk` which in our case is a DNS alias to `http://weaver.dl.ac.uk:8080/portal`.

Currently, only a small number of tools have the option to make the content they control publically viewable on the Gateway site. These include Site Info, Announcement, Resources, Syllabus. Links to viewable content are automatically included in the output of the Search tool which can be configured into the public-facing Gateway site. Other tools can be added to this site and pages are configurable as normal. For instance an iFrame can be

used to display a Web page to users not yet logged in using the Web Content tool.

Another possibility using Sakai roles is to enable a normal worksite to be publically viewable. This is done by adding the `.anon` role to that site, with suitable permissions as shown above. Members of the site can log in and edit the content in the normal way. Anyone can then read the site's content just be accessing its URL. For instance, we have added a site for general information about NCeSS that we want an anonymous user to be able to read. Its SiteId is `ncess`, so we can publish the link as `http://weaver.dl.ac.uk:8080/portal/site/ncess`. If there is a Resources tool on that site, we can even view the folders and content using `http://weaver.dl.ac.uk:8080/access/content/group/ncess`. In this way it is possible to link the resources into a normal Web site, for instance using an iFrame.

Defaults, setting up a new site as admin or as a power user

The default roles for users added to the system and included as members in new sites are "admin" and "maintain". Default realms already contain these roles and the functions they are permitted to invoke in the default tools. An individual user cannot have a default site role, it is set when joining a site; one can be set for all users joining a site with the Membership tool or for multiple participant additions with the Site Info tool. Once a participant is added to a site, their role can be changed by the site owner or admin to any of the roles defined in the corresponding realm for that site, e.g. "member" as above.

To create a new site of a different type it is important (as admin) to use the Sites tool, which gives the possibility to give the site a simpler SiteId and to specify its Type. It is also possible to use the Worksite Setup tool, once the `sakai.config.xml` file has been edited to include the full list of all site types. This allows priveleged users to create sites of all available types.

Importing Content from other Sites

It is possible to select another site from which to import content, e.g. when configuring the Resources tool. The use of this facility, and its behaviour under the security rules desribed here, requires further investigation. TBA

Sakai API for using Functions and Roles

API needs a better description for TPP developers. TBA

Authorisation in Sakai is controlled by "providers". There are User Provider, Role (Realm) Provider and Site Provider which access data stored in the underlying DB and elsewhere. The Provider architecture is designed to take in a wide range of enterprise information sources.

The User (UserDirectory) Provider can fully populate new user objects and responds to API queries about UserId or passwd, e.g. at log-in time. JLDAP, OpenLDAP, Kerberos, and IMS Enterpries can be plugged into this provider. Future work is to support the existence of proxy credentials inside Sakai.

The Site Provider is invoked when a new site is created. The type is chosen and the new site populated with information, such as start date, end date, title. APIs are available to check this information.

The Realm Provider is also checked at log-in time. It defines what sites the user can see and roles within each site the user will have. This information is cached in Sakai internal tables. The Realm Provider can take in roles from external enterprise sources, but will apply rules to ensure there are no security breaches possible.

2.2 Using Sakai Tools

In this section we go through the steps of setting up a new site type with a new role, creating a site and adding members. If the new role is `.anon` the site will be accessible to users who have not logged in. The `.anon` role mus be given appropriate permissions, e.g. read-only for all content.

First log in as admin and go into the Adminstrative Workspace.

Realms

The following steps illustrate how to create a new site template for a site of type "ncess" with an additional role "member".

1. Select the Realms tool;
2. To create a new type of site select `!site.template`;
3. Click "save as" and type in the new template name, e.g. `!site.template.ncess` to create a new template for a site type of ncess. This effectively duplicates a remplate;
4. Now click on `!site.template.ncess`, scroll down to the list of roles, and click on "access";
5. Click on "Copy Role", and enter a new RoleId such as "member";
6. Now go back to `!site.template.ncess` and select the "member" role from the list. You can then edit all the permissions for that role, e.g. as shown in Table 1.

Further roles for this realm can be added as above. This completes the creation of a new site template.

Sites

The following steps illustrate how to create a new worksite of type "ncess". This will have the roles "access", "maintain", and "member".

1. Select the Sites tool and click on "New Site";
2. Enter a SiteId such as "ncess", a Title such as "NCeSS" and a type of "ncess" with appropriate descriptions.
3. Select as follows:
   - Published
   - Joinable if it can be joined by anyone logged in using the Membership tool. A site which is not joinable is "moderated" by the site mainteiners.
   - Public View if the site should be listed by the Search tool on the Gateway site.
4. Click on "Save" at the bottom of the page.

This completes the creation of a new site called “NCeSS of type “ncess”.

Worksite Setup

This tool can be used to add default pages to the new sites. Pages can also be added and customised using the Sites tool, but the procedures if more complex.

1. Select the Worksite Setup tool;
2. Scroll down and select the NCeSS site, then click Edit on the toolbar;
3. Click on Edit Tools;
4. Select the tools required from the default set;
5. Click on Continue and then Finish;
6. Click on Add Participants;
7. Add any participants who will maintain the site, we use e-mail addresses as unique UserIds – cut and paste from a list into the lower text box, then click Continue;
8. Give these users the “maintain” role.

This completes the initial selection of tools and maintainers for a new worksite. It is also possible to use this tool to duplicate a site or to import content.

This is all the work the admin normally has to do.

Site Info

The Site Info tool can be used by someone who is a member of a site and has “maintain” role. It can be used to change the list of default tool pages, add additional Web Content and News tools and add, remove and change the roles of members. We here illustrate the latter.

1. Select the Site Info tool and click on Add Participants;
2. add new site members as above, e.g. those with “member” or “access” roles;

The role or status of site members can be changed at any time by any one of the members with “maintain” permission. Be careful not to remove your own permission!

2.3 Enabling Users to create their own Sites

For a user to be able to create their own sites, they have to be given “maintain” type by admin. Log on as admin, go into Administrative Workspace and then proceed as follows.

Users

In the Users tool, type the name of a user and search.

1. Select the Users tool;
2. Select the UserId of the person who's identity you want to edit;
3. Ensure that they have "maintain" in their Type field;
4. clock Update Details.

Next time that user logs in they should be able to use the Worksite Setup tool to create new sites following procedures similar to those outlined above. In order to add sites of all the available types, these must be added to the list in the `sakai.properties.xml` configuration file and Sakai re-started by admin.

Worksite Setup

As a user, select the Worksite Setup tool.

1. Select the Worksite Setup tool and click on New;
2. Select the required type of site, e.g. "ncess";
3. Add a title and description;
4. Follow the rest of the procedure to create the new site.

You are automatically included as the site maintainer. Once a site is created, fucther tools and participants can be added using the Worksite Setup tool again. A user-created worksite is no different from any other type of worksite.

3 Acknowledgements

We acknowledge funding from ESRC, JISC and the NWDA for some of the work described in this report.

The North West Grid (NW-GRID), is a collaboration of Daresbury Laboratory and the Univerties of Lancaster, Liverpool and Manchester with funding from the North West Development Agency.

References

[1] Sakai Project Web site `http://www.sakaiproject.org` and Sakai Confluence site (Sakaipedia) `http://confluence.sakaiproject.org/confluence/dashboard.action`

[2] R.J. Allan, R. Crouchley and C. Ingram *Scenarios, Use Cases and Reference Models* (CCLRC, June 2006)

[3] R.J. Allan, R. Crouchley and C. Ingram *Comparison of Surveys* (CSI Consultancy, June 2006)

[4] R.J. Allan, R. Crouchley and C. Ingram *Web-based Library and Information Services* (CCLRC, June 2006)

[5] R.J. Allan, R. Crouchley and C. Ingram *The Information Environment and e-Research Portals* (CCLRC, June 2006)

[6] R.J. Allan, R. Crouchley and C. Ingram *e-Research, Portals and Digital Repositories Workshop* [6] Notes from the workshop held at University of Lancaster 6-7/9/06 (CSI Consultancy, September 2006)

[7] R.J. Allan, R. Crouchley and C. Ingram *Final Report* (CCLRC, June 2006)

[8] M.J. Smith *Use Case Compendium of Derived Geospatial Data* (GRADE Project, December 2005) http://www.edina.ac.uk/projects/grade/usecasecompendium.pdf

[9] ESRC e-Infrastructure Project Web site: http://www.ncess.ac.uk/services/research Portal: http://portal.ncess.ac.uk

[10] NW-GRID: The North West Grid Web site: http://www.nw-grid.ac.uk Portal: http://rhine.dl.ac.uk:8080/portal

[11] Psi-k Network of Excellence Web site: URL Portal: http://cselnx9.dl.ac.uk:8080/portal

[12] K. Miller *Primary Selection of Datasets* Deliverable D1.1.1 of the ESRC e-Infrastructure Project (August 2007) http://www.ncess.ac.uk/services/research

[13] D.M. Sergeant, S. Andrews and A. Farquhar *Embedding a VRE in an Institutional Environment (EVIE). Workpackage 2: User Requirements Analysis* User Requirements Analysis Report (University of Leeds, 2006)

JISC Virtual Research Environments Programme http://www.jisc.ac.uk/index.cfm?name=programme_vre

[14] M. Mascord, M. Jirotka and C. Sieunarine *Integrative Biology VRE, Work Package 2: Initial Analysis Report* http://www.vre.ox.ac.uk/ibvre/IBVREInitialAnalysisReport.pdf University of Oxford (November 2005)

[15] G. Klyne *SakaiVRE User Requirements* http://wiki.oss-watch.ac.uk/SakaiVre/UserRequirements

[16] R.J. Allan, R. Crouchley, M. Baker and M. Fraser *The Sakai VRE Demonstrator* http://www.grids.ac.uk/Sakai

[17] National Grid Service http://www.ngs.ac.uk

[18] A. Fish and M. Gonzalez *Agora video-confrencing tool.* http://agora.lancs.ac.uk