Inc-AtlassianIdentityUseCases-Summary

> **Warning**
>
> Please do NOT set a watch on this page if your Preferences do NOT include an email address!
>
> Doing so will prevent any further editing of this page.
>
> Thanks!

Authentication and Authorization

Separating Authentication from Authorization. The currently available set of Confluence plugins combines both of these functions. For instance, a site can use Ldap for BOTH authn and for authz (group memberships). A site can't use Kerberos for authn and Ldap for authz. A site can't use its existing Web SSO product for authn and Ldap for authz. A site can use CROWD as an SSO and Ldap for authz... but why would a campus want to deploy yet another Web SSO framework? Many campuses report developing custom plugins to handle authN, authZ, and Groups management.

1. We want to authenticate Confluence users with our current intra-campus Web SSO system, but use ldap-based groups and user objects to manage permissions within Confluence. Currently, to do this, we would need to write a custom plugin.

2. Single Sign-On: We currently authenticate Confluence users against our campus AD. It would be nice to have single-sign-on (SSO) functionality using SPENGO / GSSAPI over HTTPS for authentication. Computers in an Active Directory environment could automatically authenticate the user when any wiki page is viewed or perhaps when the "login" button is clicked (to allow a space admin the ability to logout to check anonymous access to the space.)

3. (authn chaining) (UW-Madison). We have a group that is mostly campus people. There are also ad hoc external members. We need the campus people to authenticate via our Web Single Sign On solution. The ad hoc external members need to authenticate using internal (confluence) user accounts. Requirement - members of a group must be able to use different authentication methods (some may be internal application authentication, some campus SSO and some via Shibboleth).

4. (recipe for "best practice") When should user objects be created within Confluence? What is the "best practice"? What tools/interfaces are available?

Operating in a Hybrid environment. A growing number of campuses worldwide are reporting that they need to allow both local and remote users access to controlled spaces. They need to allow both local and Federated users to login to Confluence, and gain access to resources. This implies the need for some mechanism to persist privileges associated with remote users; perhaps the easiest approach is to dynamically create user objects within Confluence that are associated with the remote users, and associate privilege information with that user object.

5. A course based at Brown includes students registered at Brown, as well as 14 students registered at a university in France. The Brown-based instructor is able to compose an ldap-based group that includes the local students as well as the 14 french students (who access the brown-based services using their Federated credentials). The instructor is then able to grant this group write access to a space within Confluence. The local students use their brown credentials to access this space; the french students use their "local" credentials (at the French university), and the Shibboleth software, to access this space.

6. A collaborative development effort at Cornell includes staff members from Indiana University as well. The JIRA project housing the development issues resides at Cornell. Users from both campuses login with their local credentials, and use Shibboleth to login to the JIRA instance.

7. (Login error processing) When authentication fails, a recipe is need (and perhaps interfaces?) describing how an authN plugin should return information information to the browser user about the failure and "next steps".

Permissions -- Functionality and Management

3. **User/group/privilege management** in a large scale environment. Confluence currently contains a set of screens for managing Security and Permissions for a space and its pages. These screens support having group memberships stored locally within Confluence, or stored externally in an Ldap directory. However, these screens become unusable in environments with large numbers of groups (eg > 50K groups). In addition, there are a host of privacy and visibility issues related to group membership that are not addressed by the Confluence tools. Brown Univ has submitted a number of JIRA issues related to these problems.

8. The Permissions management pages/tools are difficult to use in environments with large numbers of groups (eg 100K).

9. I can use the Confluence permissions management pages to see and inspect groups that I'm not supopsed to know exist. There is curently no way to control the set of groups that the user sees. Confluence BINDs to the ldap directory with a superuser set of crendetials; every browser user sees every group.

10. (Get away from the "permissions only get narrower model"). Provide the ability for page level permissions to expand on the permissions granted at the space level, rather than the current model of only allowing further restrictions from what was granted at the space level. Allow the server administrator to globally control this feature.

11. A child page should optionally inherit access settings from the parent page, rather than the space.

12. Support for Nested LDAP Groups: Our Confluence installation queries the entire campus Active Directory for groups named with the prefix "wiki-". This allows department space administrators to create and manage Confluence AD groups within their own OUs. Oftentimes, however, the membership of a departmental-level wiki- group is the same as, or is a superset of, an existing group. If Atlassian supported nested LDAP groups, department group administration could be simplified by adding pre-existing groups to "wiki-" Confluence groups.

13. Allow the server administrator to optionally globally block the ability of space administrators to turn on anonymous edit. We don't allow that at our institution and don't want to let an instructor override our policy. (Brown - PL)

If anonymous permissions are allowed, Confluence should be able to report on permissions that have been enabled for anonymous users. Even better, Confluence has the option of sending email updates to the administrator or space admin on all anonymous edits. (Or perhaps not display anonymous comments until they are "approved" by a space admin.)

14. If a user is logged in, which means that at the least s/he is a member of
Confluence-Users (the default group), s/he CANNOT see content which is otherwise
anonymously viewable unless that Confluence-Users group is explicitly added to
the permissions to view. Otherwise said person sees nothingness.

15. For at least pages, add the ability to split the create and edit permission so that users and groups can be permitted to edit existing content but not to create new content.

Provisioning into Confluence

16. We provide instructors with a web-based tool to manage the applications that support their their course. From this tool, they are able to manage fifteen different applications and services that their course might use; "manage" includes clicking a button to indicate "I want my course to be able to use this application; do the appropriate initialization". In the case of Confluence, this tool wants to 1) create a new space within Confluence for the course, and 2) assign the appropriate permissions to this space.

This page describes the APIs that Confluence exports via SOAP and XML-RPC:

http://confluence.atlassian.com/display/DOC/Remote+API+Specification

It appears that all of the required functionality is available.

17. UW-Madison. We have a tool that manages groups across multiple collaboration tools. We also have a Course Management System that needs to push groups and group changes into Confluence. The initial request will also need to establish spaces and request pages in Confluence. We need to be able to push space and page creation commands, rights and groups from a variety of tools directly into Confluence. (This might exist via the API currently)

(stc -- I believe that the currently exported interfaces provide the required support.)

18. User Deprovisioning: Any user with a valid Active Directory account effectively has a Confluence account if they are granted use rights. Once they login they can setup a profile, create and edit content, set watches, etc. Once that AD account is no longer valid, however, there is no longer any way to clean up there access and settings. For example, if they had daily digests or watches enabled, Confluence will continue to try and send those notices. Permissions can be manually cleaned up by going to every single location that the user had individual access rights (very tedious). Issues such as the daily digests don't appear to be able to be dealt with in any manner other than directly modifying the backend database.

(existing API supports this)

Working in a Large-Scale Environment

19. UW-Madison - we need to be able to create a "domain-like space", and grant to a large group (like the Library) authority to manage that domain. We need to able to grant a user or set of users admin rights over that "domain-like space". They can they create new sub-spaces and manage groups and permissions under the domain-like space. They should not have other application administration privileges.

The master Confluence administrator creates a new collection container for the chemistry department and assigns a chemistry "collection administrator". The collection admin may create new spaces within the chemistry collection, and assign user/group collection browse/access rights. Spaces created within the Chemistry collection may inherit properties set at the collection level - user access, space styles, page templates, etc. Spaces may be made globally visible, or visible only to collection users. Users may select between global- or collection-specific Dashboard views.

Even with the various existing methods for tagging spaces, the sheer number of spaces produced by a large university installation can easily overwhelm the Confluence dashboard. The Team Label feature is useful, but lacks the kind of administration features we're imagining.

20. The dashboard on our server is getting almost unusable due to so many spaces being visible. It is worse after a user logs in as then they all show. What we really need is a way to organize these. I envision an AJAX organization system where the server admin can categorize each space into one or more of these categories which would show on the dashboard and do an AJAX show/hide as the user clicks on the category name. (Brown - PL)

21. (space level statistics) Created Feature Request CONF-9711 for allowing space stats, like a Google Analytics. The options that we've been given are to either increase logging through log4j, add Google Analytics in the Custom HTML area of the server or set it up for individual spaces in the Layout section under Space Admin. The problem with each is:

Logging:our log files are already huge. Adding the logging that would be needed for site analysis would require rotation at least hourly to keep them usable

Custom HTML: this sets one Google Analytics account for the entire server. Whoever has access to that one Google account sees data for every space

Layout: This is great in that it is the closest to what we want. We can set a Google Analytics account for one space, however it can only be set up by the global server admin. When you have hundreds of spaces, this isn't reasonable. Plus it is not upgrade proof. It would need to be re-done with each upgrade. Lastly, if you're running your server over SSL, you get a dialog box on every page because the Analytics code is non-SSL'd. (Brown - PL)

Operational Issues

22. Currently, we can not restore an exported space from a newer major version to an older major version. And a new major version was just released (2.6.0). So if I export some of my old spaces in my 2.5.3 server, then upgrade to 2.6.x or beyond, I can no longer import my old exported spaces. We need this backward compatibility. (Brown - PL)

23. Along those lines, for semester-bound spaces, an easy way to programatically archive a list of spaces at the end of the term would be useful.

24. Several satellite instances of Confluence have been installed at Cornell and a current exercise is underway to migrate content from the satellite instances in departments, to the centrally scaled enterprise instance. Currently, migration between instances is somewhat manual although there seems to be standards patterns of data mapping that need to get addressed in the migration, such that a helper plugin could ease the burden of this work.

25. It would be useful to be able to put the entire Confluence instance (all spaces, public and private) into a read-only state for occasions where we might want to redirect, for safety, to a backup server/db. For instance, when database maintenance / upgrade / migration is being done.

26. Managing a cluster

27. Plugin Certification: We would like to see Atlassian offer a plug-in certification program. Certified plugins would be evaluated for security, performance, and scalability, and would be known not to conflict with other certified plugins. If this were included in the basic confluence purchase price, great! But if not, I believe that many enterprise level Confluence users would be happy to pay a subscription fee to access a certified plugin library.

Additional Use Cases

Additional Use Cases which were submitted by single campuses can be viewed here