

InC-AtlassianSummary

Integrating with an Enterprise Environment

Here is a summary of the issues that have been presented so far:

1. **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.

2. **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.

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.

Many campuses now deploy Confluence as part of a suite of applications supporting collaboration and instruction. In this environment, it seems best to provide instructors and project leaders with a single tool, external to all the applications, which would be used for managing which applications which will support the course/project. Clearly, this tool would be outside of all of these applications, and each application would have to export functionality required by this tool.

4. **Page-watch email bug.** (this is mentioned, since it seems to occur mostly frequently in Federated environments)

- Any user can set a watch on a space s/he is able to view (to be notified by e-mail of changes), even if s/he doesn't have a valid e-mail address in his/her profile. If a user without an e-mail address has a watch set on a space, any attempts to edit a page in that space results in the resulting attempt to send the watch mail failing and this in turn causes the edit attempts to fail with a very unhelpful/misleading generic error message. Thus a careless action by any one user renders an entire space uneditable.
- There is apparently no way to review which users have watches set on a space. This makes managing the platform much more difficult (due to the bug noted above) and eliminates a very valuable piece of information about usage patterns.
- For many reasons it would be much better to at least have the admin option of requiring a valid e-mail address in a profile before a user is able to complete the registration step, and to allow for an error page to be displayed to the browser user - no matter what mechanism is used for authN access - explaining the why of the failure and how to fix it.

Operational Issues

While many of these may seem minor, many of them cause immense headaches on a day-to-day basis:

1. **Scalability and robustness.** Atlassian has provided recommendations on how to [run Confluence in a clustered environment](#), in order to meet scalability and robustness requirements. We don't currently know of any campus that is actually configured in this manner. Note that operating in this mode currently requires that all attachments be stored in the backend database, and sites are expressing concerns about that requirement. Note that the recommended clustering approach must be compatible with the approaches to the enterprise issues listed above.

2. **Backup and restore.** We love the simplicity and ease-of-use when it comes to Confluence's backup and restore procedure. Unfortunately, though, it only allows a site to restore the whole Confluence instance (wiping out everything that was there before the restoration procedure began). This is great for disaster recovery, but not very practical for fixing user accidents ("Oops! I deleted my space!" or "Oops! I deleted her space!"). It would be very good to have partial restore (space-by-space and page-by-page) capabilities.

3. **Migration and consolidation.** We've found that other units who have been running their own wikis are, in many cases, quite eager to move all of their content into a central Confluence service so that they can retire their own wiki and make use of Confluence's considerable advantages. Atlassian has provided some really good tools, which we have utilized, for merging the content of other wiki tools (SnipSnap, etc) into a Confluence instance, but merging all or part of a different Confluence instance is not as smooth.