COmanage Manifesto

This manifesto describes a broad view of COmanage, as well as directions in which to take the program, as imagined by its developers.

COmanage as a Gray Box

The COmanage product should lie somewhere between a completely closed off "black box" and a completely customizable "white box."

A black box, in which nothing could rely on existing infrastructure outside of the COmanage server, would likely be too limiting to be considered scalable. Moreover, such an approach would likely make the adoption a harder sell at sites that have some infrastructure in place (such as an existing LDAP infrastructure).

In contrast, a white box would be difficult to design since the number of potentially customizable features is limitless, making it complicated to offer reasonable upgrade and migration paths, reasonable stability and reliability, and reasonable support to users. The more users change configurations outside of the structured management interface, the more likely that future upgrades will break the COmanage instance and the greater the number of variables to consider when debugging or troubleshooting problems. Furthermore, institutions capable of making sophisticated changes may already employ persons who may find COmanage too limiting and thus less optimal for their needs. Or, such institutions may graciously lend time and energy into the COmanage project to add the features they seek.

As a gray box, COmanage should offer a reasonably simple and intuitive interface to allow quick deployments of a standard set of collaboration tools integrated with Internet2 middleware products. A streamlined set of features around the management of users, groups, permissions and the COmanage instance as a whole should be made available. Furthermore, where possible, practices for safe, manual extension and hand configuration should be documented so additional tweaking can be done if desired. However, as a developing product, it should be clearly noted that such efforts should focus on reasonable future proofing but not guaranteed future proofing.

Distribution Agnostic

At present, the core team of developers are working in a Debian distribution of Linux, but the COmanage components should be designed to be as distribution agnostic and easily portable as possible. Therefore, the packages should not rely on features specific to any one package management platform, such as Debian's .deb or Red Hat's .rpm. Instead, packages should confine operations to installing files, creating symlinks, etc., and should use only basic pre- and post- install scripts when necessary. Heavier configuration tasks should be relegated to setup scripts, configuration tools, and/or configuration webpages.

Repository of Domesticated Applications

Considering the earlier requirement for loose-coupling with the distribution, a central repository may sound difficult to create and maintain. However, a reliance upon native package formats and their associated repository technologies will offer the COmanage user community an easy way to install and update domesticated applications and the core software of COmanage. There are a number of challenges that the COmanage team must face to make this repository effective:

- providing and maintaining repositories for the most requested OS flavors (likely Debian apt repository, and Red Hat yum repository)
- converting domesticated applications into the native package formats such as .deb or .rpm
- committing to keep up-to-date the provided applications and QA updates before release

The benefits of such an approach include:

- Non-technical users, who may not otherwise know how to locate or install domesticated applications, will be able to easily use the central repository.
- The ability to notice when updates are available may make adopters less vulnerable to the exploits that are common among web apps. (This is difficult when not using native packages since it is hard to programmatically assess the version of a "hand-installed" application.)
- The ready availability of domesticated applications may increase the adoption of the COmanage platform.
- Developers will know exactly how an application is installed and configured, making troubleshooting and debugging easier.

Streamlined Feature Set

With regards to the management of users, groups, and permissions, the capabilities of the core infrastructure used in COmanage is virtually limitless. It would be difficult to create an interface that allows administrators to take advantage of all of these capabilities, and therefore, the following streamlined feature set will be provided:

1. The ability to add, delete, and modify users.
2. The ability to invite users and automatically assign them to a group.
3. The ability to add, delete, and modify groups.
4. The ability to install and uninstall domesticated applications.

Further streamlined features will depend on the specific applications running on the COmanage instance.

Extensible Collabmin Interface

The web-based "collabmin interface" should be easily extensible, allowing add-on packages to add tabs to the interface so a collabmin can configure items specific to that add-on. For instance, without such an interface it would be rather difficult to create a generic page capable of managing any and all wiki platforms. Such a page would either have to know which features are available to which flavor wikis, or it would have to limit the configurable items to the feature set common to all wikis. Moreover, since there may eventually be a large number of wiki platforms supported, giving the groups page the ability to manipulate the wiki (to add or remove group permissions, for instance) would be similarly difficult.

Therefore, the extensible collabmin interface should create a tab for each "mixed in" domesticated application. The design of these tabs should adhere to the general design and aesthetics of the collabmin interface and should offer a streamlined set of operations specific to the application. This interface must rely on the COmanage APIs to query information about the bits managed by the underlying COmanage instance (e.g. the list of users on the system, the list of groups, the location of the LDAP server, etc).

Extensible Hooking Mechanism

To accommodate semi-domesticated applications, an extensible hooking mechanism should be provided so installed applications can specify a script to be called anytime one of the above mentioned streamlined operations is executed.