Registry of Domesticated Applications

Updates are being done in cooperation with SURFNet's COIN project and can be found online at: <https://wiki.surfnetlabs.nl/display/domestication>

Application	Purpose (email, science, bug tracking, etc)	License	Language (Perl, C, Java)	Level of Domestication (see below) : Groups	Level of Domestication (see below) : Authentication	Level of Domestication (see below) : Authorization	Provisioning/ Deprovisioning	Schema and Protocol Compliance	Overall certification of Domestication (Full, Partial, Not)	Comments
Confluence	Wiki	Commercial	Java	L4/LDAP L1	L3	L4/LDAP L1	Possible, not standardized	None	Partial	
Grouper	Group Management	Apache 2	Java (mostly)	L1	UI: L4/CAS, remote_user L2 WS: L2	L1	web services API; Grouper loader	SPML2	Partial	
OpenSSH	Remote Login	BSD	C	L4	L4	L4	Depends on underlying support	SSH	Full	This is really about federated provisioning

Definition of Application Domestication

The ability to externalize group management, authentication, and/or authorization from within the application, allowing for the use of federated identity by that application.

Note that the targeted audience Application Integrators What does federated groups look like?

Level of Domestication

Level 0 - Not externalizable, not provisionable via automated tools (black box)

- Not considered domesticated.

Level 1 - Not externalizable, but provisionable via automated tools

- Not considered domesticated. eg for authn, this would imply passwords stored locally.

Level 2 - Externalizable via hooks

- Considered domesticatable.

Level 3 - Externalizable via third-party plugin

- Considered domesticated for available plugins.

Level 4 - Externalizable via vendor supported functionality

- Considered domesticated for supported functionality.

Level may vary according to protocol/technology. eg an app that supports OpenID natively but SAML via hooks would be Level 4 for the former and Level 2 for the latter. We probably need to capture that somehow.

App that has hooks built in for SAML (or OpenID, or something like that) is very good, app that has SAML (or something) support built in is good, versus something that as in-house customizable thing that can be used; versus something that only integrates with LDAP (not nec. Useful in a federated environment); versus internal only

Have to write glue code, versus some kind of third-party support, versus vendor support - all means some level of domestication, but imply different amount of work and long-term assurances

What to include - just ones we know about, some "FAQ" type ones, more
This is like calculating LOA. So overall, have a summary level that says, at the end of the day, the app is silver/gold/platinum (tin/pewter/aluminum); model the certification on the InCommon list

Compliance for data interchangeability