SSO implementation experiences

Matthew J. Graham (Caltech, NVO)

NVO SSO infrastructure

- Setup instructions at http://wiki.ncsa.uiuc.edu/wiki/NVO_SSO
- Registration and credentials through Purse + MyProxy server at NCSA
- Client-side:
  - Pubcookie Apache module to log user in and manage granting cookie
  - mod_myproxy Apache module to retrieve proxy certificate using pubcookie granting cookie as passphrase
  - mod_jk Apache module to integrate Tomcat

Caltech implementation

- NESSSI portal (http://nvo.caltech.edu/apps/nesssi): uses XForms and AJAX (built on Orbeon)
- pubcookie relatively straightforward
- mod_myproxy uses myproxy_logon - requires Globus!
- Wrote a simple Java class - MyProxyLogon.java - to replace myproxy_logon: just requires Java CoG
- Lots of redirection over HTTPS: need to configure Tomcat to act as HTTPS client
- Problems with 3rd-level proxies (see next slide)

3rd-level proxies

- Purse creates **encrypted** credential in MyProxy server store
- User logs in - username and password go to MyProxy server via pubcookie in exchange for a 1st-level proxy credential
- Pubcookie verifier stores 2nd-level proxy (unencrypted) back in MyProxy server
- Tomcat gets pubcookie granting cookie from Pubcookie login server at NCSA which it submits to MyProxy server in exchange for 3rd-level proxy
- Some HTTPS clients do not regard multiple level proxies as valid

How to avoid multiple level proxies

- Don’t encrypt original credential: let MyProxy use a PAM module to do password checking
- Don’t store long-term credentials at all: use MyProxy online CA instead.
- Proposed scheme for federating login servers will still introduce 2 proxy levels to credentials when logging in to a login server that is not the one originally registered with.