MyVocs

Introduction to myVocs

myVocs is a virtual organization collaboration system (VOCS) developed at the University of Alabama at Birmingham funded by NSF ANI-0330543 "NMI Enabled Open Source Collaboration Tools for Virtual Organizations". This page gives an overview of myVocs. See the attachment at the bottom of this page for a recent presentation.

The myVocs and GridShib project teams are exploring ways to integrate their respective software products. See the topic MyVocsGridShibIntegration for more information about the proposed integration.

Basically, myVocs is a SAMLIdPProxy, a bridge between a federation of Shibboleth IdPs and a federation of Shibboleth SPs:

Using myVocs, the SPs (called VO SPs) may be aggregated into virtual organizations (VOs). We think of VOs as **people**, and the aggregated SPs as a federated set of distributed applications. It is an important feature of myVocs that a single VO SP may serve multiple VOs.

Like the !IdPs, the VO SPs may reside in arbitrary administrative domains. Using off-the-shelf, open source software components (such as Shibboleth, ! MySQL, and Sympa), myVocs provides the "glue" that authorizes access to a VO SP based on membership in some specific VO.

In myVocs, a VO includes of a set of tools or applications protected by VO SPs mutually trusted by a VO !IdP. The following diagram illustrates the relationship among the various myVocs components:

Here is an outline of a typical *myVocs flow for webapps*:

1. A browser client requests a VO web resource protected by a VO SP. If a security context for the principal already exists at the VO SP, skip to step 18.
2. The client is redirected to the VO !IdP (which is protected by a federation SP).
3. The client makes a Shibboleth AuthnRequest to the VO IdP. If a security context for the principal already exists at the VO !IdP, skip to step 12.
4. The client is redirected to the federation !IdP (ignoring a possible interaction with the federation WAYF).
5. The client makes a second Shibboleth AuthnRequest to the SSO service at the federation IdP. If a security context for the principal does not exist at the federation IdP, the IdP identifies the principal (details omitted).
6. The IdP updates security context for this principal, issues an authentication assertion, and returns an authentication response to the client.
7. The client submits the authentication response to the assertion consumer service at the federation SP. The assertion consumer service validates the authentication assertion in the response and passes control to the attribute requester.
8. The attribute requester queries the attribute authority at the federation !IdP.
9. The attribute authority returns an attribute response to the attribute requester.
10. The federation SP updates its security context for this principal and redirects the client to the VO !IdP.
11. The client makes a Shibboleth AuthnRequest to the VO !IdP, the same AuthnRequest made at step 3.
12. The VO IdP filters the attributes from the header of the request (by virtue of the attribute exchange in steps 8 and 9), persists these attributes to the VO database (if necessary), and returns an authentication response to the client.
13. The client submits the authentication response to the assertion consumer service at the VO SP. The assertion consumer service validates the authentication assertion in the response and passes control to the attribute requester.
14. The attribute requester queries the attribute authority at the VO !IdP.
15. The attribute authority returns an attribute response to the attribute requester. Both federation attributes (persisted at step 12) and VO attributes are included in the response.
16. The VO SP updates its security context for this principal and redirects the client to the VO resource.
17. The client requests the VO resource, the same request issued at step 1.
18. The resource filters the attributes from the header of the request (by virtue of the attribute exchange in steps 14 and 15), makes an access control decision, and returns the resource to the client.

Any number of webapps may be protected in this way. The topic OnBecomingVOSP describes the process of becoming a VO SP.

What attributes are captured and persisted at step 12? Today, myVocs requires the federation !IdP to release attribute `eduPersonPrincipalName` , a globally unique identifier for the principal. This global identifier is permanently bound to a local identifier in the VO database. It is this binding that permits myVocs to determine the VO attributes associated with the user.

The local identifier is determined as a result of a one-time registration step. At the time of registration, the user's global identifier is bound to a local identifier in the VO database. A more flexible registration process is being implemented.

In the architecture diagram above, the myVocs SP relies on an ordinary WAYF for !IdP discovery. In fact, myVocs can provide an enhanced IdP discovery experience for the end user.