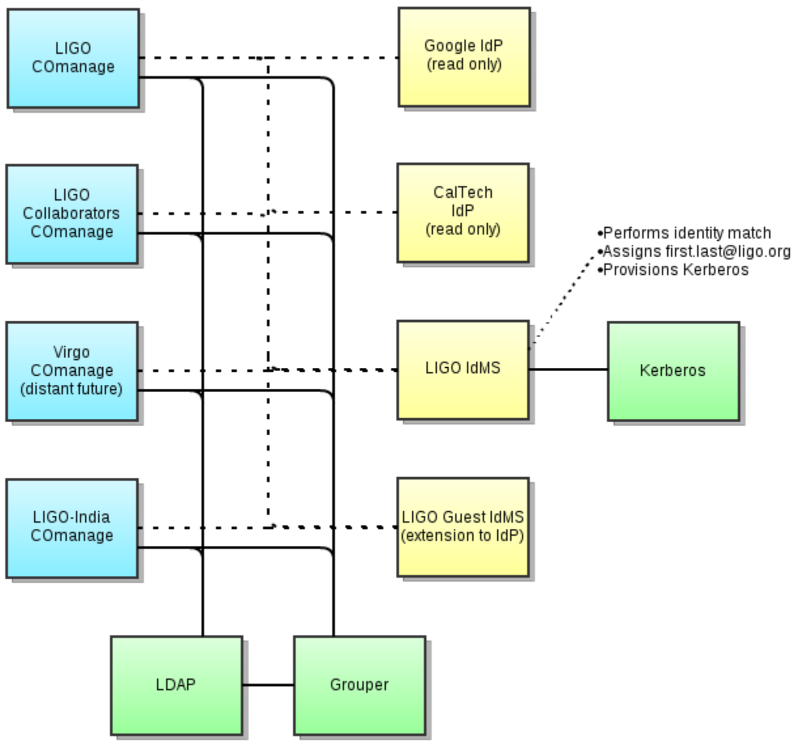
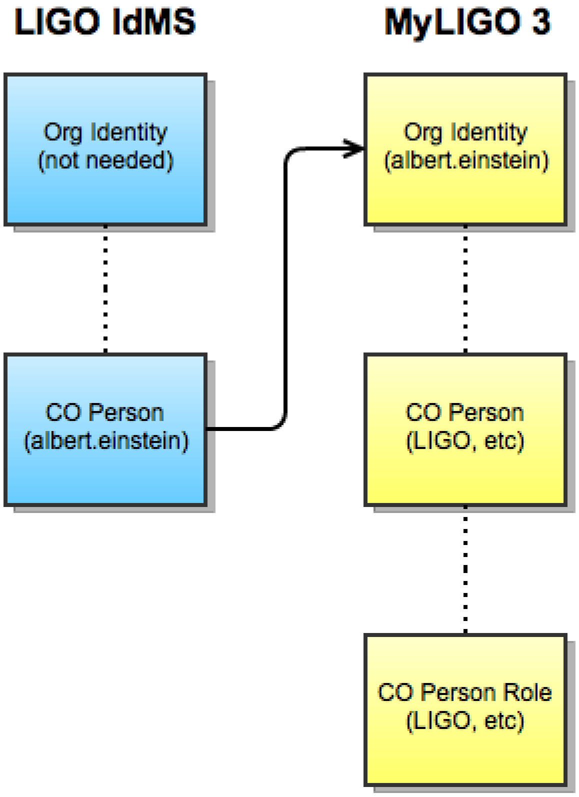
LIGO COmanage Deployment Architecture

LIGO COmanage Deployment Architecture


LIGO COmanage
LIGO Collaborators COmanage
Virgo COmanage (distant future)
LIGO-India COmanage
Google IdP (read only)
CalTech IdP (read only)
LIGO IdMS
LIGO Guest IdMS (extension to IdP)
Kerberos
LDAP
Grouper
•Performs identity match
•Assigns first.last@ligo.org
•Provisions Kerberos


LIGO IdMS
MyLIGO 3
Org Identity (not needed)
CO Person (albert.einstein)
Org Identity (albert.einstein)
CO Person (LIGO, etc)
CO Person Role (LIGO, etc)

A MyLIGO/COmanage enrollment flow begins much as any other enrollment flow would. The enrollment flows require authentication so COmanage will redirect the user to the protected authentication page. With a standard deployment the user then sees a discovery page. This discovery page is customized so that it allows a user to pick from an existing identity and IdP or choose something like "create a new LIGO identity". If the user chooses to create a new LIGO identity then control is passed to the LIGO IdMS.

The LIGO IdMS receives the SAML AuthnRequest but unlike with a normal AuthnRequest the process is really about creating a federated identity. The user fills out a form to collect name, home organization, and email and in the process there is identity matching and handling of name conditions. At the end of the process a LIGO identity (albert.einstein@LIGO.ORG) has been provisioned but is not active (no real authentication can happen using the normal LIGO IdP). The LIGO IdMS returns a SAML request with an assertion that contains the NameID and attributes and uses some special indicator (rather than the standard transport protected password) of the "authentication" event. Then COmanage can consume information from the assertion as normal.

When the Kerberos principal is provisioned by the IdMS the password is set but the principal is marked as expired so that it cannot be used for authentication. A later step at some part of the flow allows for that to change, possibly some of provisioning plugin.

LIGO IdMS Requirements

1. New user triggers MyLIGO Collaboration Enrollment
2. User is prompted for "Login using your federated identity" (FUTURE) or "Create a ligo.org identity" (required for all for now)
3. If creating a ligo.org identity
    a. Control transfers to the LIGO IdMS with a SAML AuthnRequest. The control is transferred to some plugin endpoint that "acts like an IdP" by consuming the AuthnRequest. It must store in the session the SP that sent the request. The plugin then sends the browser to execute an IdMS enrollment flow.
    b. IdMS Enrollment flow executes
        i. Enrollee is prompted for name/home organization/email
        ii. Identity matching checks for existing similar enrolled identities
        iii. On submit, a petition is created and an invitation is sent
        iv. The enrollee clicks to confirm the email address. Both the Org Person and CO Person record go active because approval is not required. At that point provisioners fire and the LIGO Kerberos provisioner can provision a Kerberos principal with a random password and probably not active. The IdMS is configured to send the browser to a plugin URL in the IdMS.
        v. The plugin must create the SAML artifact which includes the information that the MyLIGO COmanage enrollment form needs. It is created on the file system and then it sends the browser to the right Shibboleth URL so that the file system artifact is consumed.
        vi. SP authentication information provided (setting NameID and attributes)
        vii. Control is returned to MyLIGO Collaboration Enrollment which pulls the eppn and other attributes (via the SP), and populates the enrollment form
        viii. The approval process proceeds.
        ix. After approval, the IdMS needs to be notified to activate the Kerberos principal (MyLIGO Collaboration provisioner plugin calls out to IdMS) and allow the user to set a password.

High level

- Implements an identity matching service conforming to the CIFER Id Match API
- As part of the CIFER Id Match API implementation creates new LIGO identities when requested.
- As part of a LIGO specific implementation provisions Kerberos principals for those identities.
    - The IdMS does NOT provision details about identity into LDAP since the main COmanage deployment in the architecture is responsible for doing that.

Some things to consider:

- Can the IdMS just be another COmanage instance that has one flow for one "CO" and a special plugin or two? Note that it would not be the main MyLIGO/COmanage deployment.
    - Alternately, could the IdMS be implemented as an ID Match Service, with IdP of Last Resort plugins on the blue COmanage instances for credential management?
- Can we leverage the Shibboleth Native SP backdoor functionality to do the handoff of the SAML assertion back to the MyLIGO/COmanage, so that it can consume the new organizational identity information that way? If we use the artifact back door then the IdMS would have to run on the same server box, but if we use the external authentication handler back door then it does not, and that is likely to be less confusing.