# Infrastructure - Task #1616

Story # 1476 (Closed): Create Identity management system

## Implement mechanism for legacy accounts to be mapped to CILogon accounts

2011-05-31 17:49 - Matthew Jones

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 2011-11-01 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Ben Leinfelder | **% Done:** | 100% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **Milestone:** | None | **Story Points:** | |
| **Product Version:** | * | | |

**Description**

The current API allows for two accounts that can be authenticated via CILogon to be mapped as equivalent identities. However, there are many accounts in use in existing systems that can not be authenticated via CILogon (e.g., KNB LDAP accounts), and these are referenced in many access control rules for existing data. We need a mechanism to map these legacy accounts to new CILogon accounts without authenticating. Or, we need to determine how to allow users to authenticate against these accounts with CILogon. One mechanism might be to import the accounts into the IdM via an administrative action. Alternatively, we can allow CILogon to use these LDAP systems to authenticate.

Discuss with Matt, Ben, Dave, Randy, and Jim, and implement.

**Subtasks:**

| | |
|---|---|
| Task # 1967: Add CNIdentity.mapIdentity() to the API documentation | **Closed** |
| Task # 1968: Edit legacy KNB account mapping sequence diagram | **Closed** |

**Related issues:**

| | | |
|---|---|---|
| Blocked by Infrastructure - Story #1964: node registry needs to store Node.Se... | **Closed** | **2011-11-02** |

### History

**#1 - 2011-06-22 05:17 - Ben Leinfelder**

## My proposal:

1. Batch load all existing KNB accounts (including those that utilize referrals)
2. Direct users to CILogon
3. Users authenticate normally with CILogon using any compatible identity (institutional, Google, ProtectNetwork, etc...)
4. The CILogon certificate is downloaded locally
5. Either as an extension to the CILogon JWS app or as another JWS app we:
6. Prompt for the KNB identity and password
7. Authenticate using the current Metacat API (returns a sessionId and full name and email)
8. Use the CNIdentity service to map the CILogon identity to the KNB identity (we have the CILogon certificate for the CILogon identity subject)
9. Confirm the mapping request with CNIdentity service. We may need to generate an ad hoc certificate with the KNB identity as the subject or use an "admin certificate" that allows us to circumvent the normal CNIdentity authorization check for confirming an identity mapping.

I don't think we can accomplish this via a web browser - hence the JWS app - because we need to include the two different certificates for the CNIdentity service and having a user import them into their browser and then select the correct one would be very challenging (plus we just did a way with the .p12 certs).
If the CILogon JWS that downloads the initial certificate can be extended to include an extra wizard page that prompts for the KNB identity and credentials, then that makes our life all the easier, otherwise we'd be looking at launching our own JWS app.

#### #2 - 2011-06-23 00:25 - Ben Leinfelder

I've added to Figure 4 - the authorization sequence diagram - the same proposal I outlined here in text.
http://mule1.dataone.org/ArchitectureDocs-current/design/Authentication.html

#### #3 - 2011-06-27 19:29 - Ben Leinfelder

*- Status changed from New to In Progress*

It may be necessary to add a second Subject parameter to the identity mapping methods so that an admin account (passed in the Session param) can perform the mapping on two other identities that are not in the Session.Subject

#### #4 - 2011-11-01 16:44 - Ben Leinfelder

*- Milestone set to None*

Instead of using the JWS certificate generation app, we can use the portal to manage CILogon certificates. The KNB (Metacat) instance will be used to authenticate legacy LDAP accounts. When both of these accounts are concurrently logged in, the trusted KNB MN can invoke the CNIdentity.mapIdentity() method -- which takes 2 subject arguments as well as the session that represents the call is coming from our trusted MN. I've tested this using a local Metacat instance and am pretty happy with it.

#### #5 - 2011-11-01 17:13 - Ben Leinfelder

Need NodeList to store the ServiceMethodRestriciton list for the CNIdentity.mapIdentity() method

#### #6 - 2011-12-17 00:58 - Ben Leinfelder

*- Status changed from In Progress to Closed*

This is complete now that the method is restricted -- though it does require configuration.