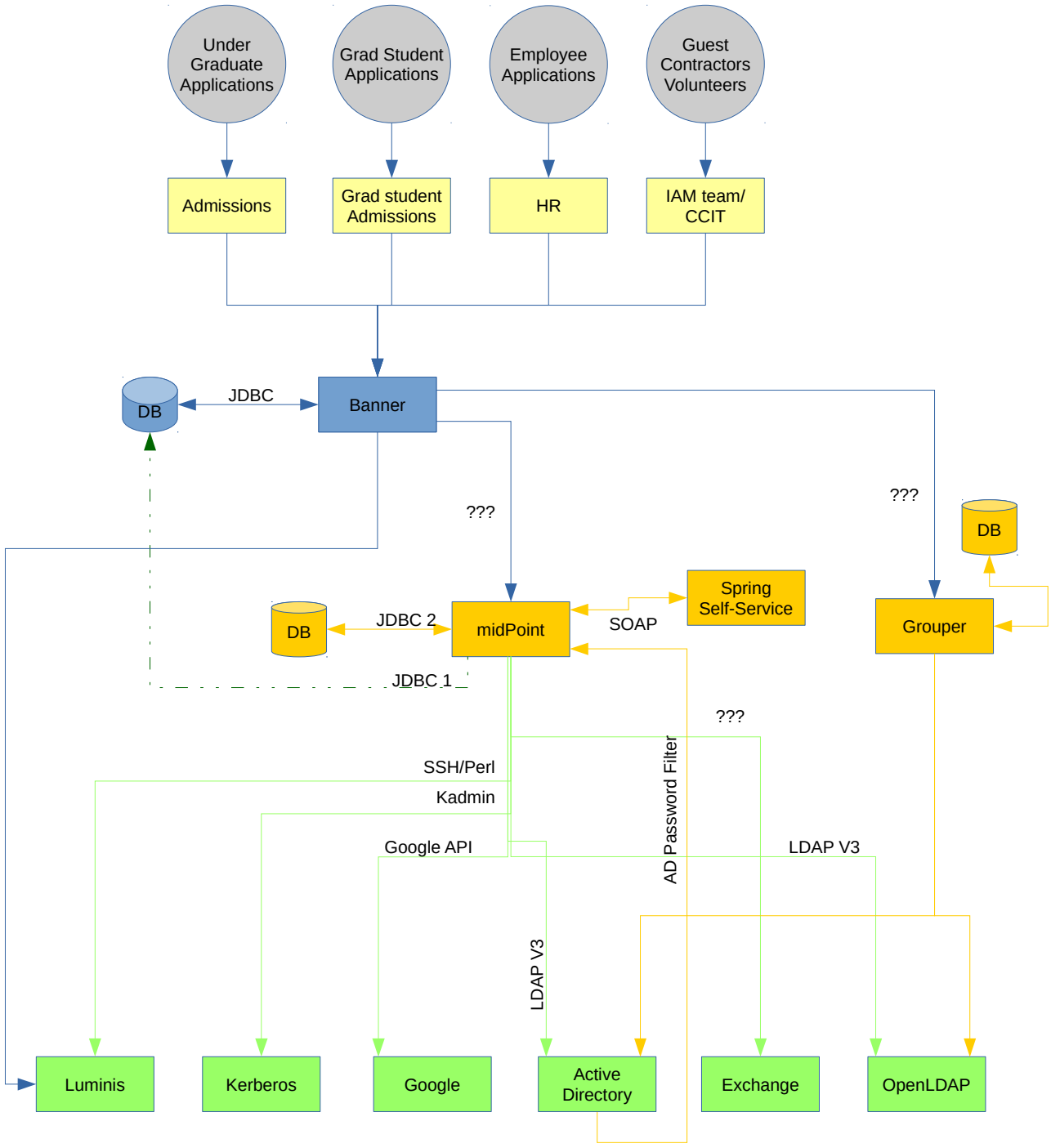
midPoint/Grouper Functional Model



See the file “Current IAM functional model.pdf” for a description of the IDM that Mines is replacing.

The midPoint/Grouper functional model is the first attempt at a design to replace an IDM system provided by a vendor in 2015. For the most part, midPoint has many features similar to the vendor IDM, delegated administration, role based access control, and password management. Mines had been planning to implement Grouper alongside the vendor IDM, so the addition of Grouper alongside midPoint fits our needs.

There are some missing components and some that are there, but implemented differently. The missing components include:

- self-service account claim workflow for new users
- ability for a new user to select a user name
- active directory password filter
- event driven connector to send identity data from Banner to midPoint

As with all projects, there are a lot of trade-offs, the paragraphs below will try to describe a few of them as well as compare and contrast features in the vendor solution and midPoint.

For various reasons, the name of the vendor and the software has been redacted from this document. In the descriptions below, the IDM that Mines intends to replace with midPoint and Grouper will be referred to as the vendor solution or the vendor IDM.

Account Claim/user name Selection

The self-service account claim workflow allows a new user to enter their full name, birth-date, and CWID to create an account at Mines. See the Current IAM functional model for more details on the account claim process.

The account claim self-service application has the ability to let the user pick a user name. The IAM staff thinks this is a nice feature. The previous IDM system picked user names for the user and there were always a few that would claim the account and then turn around and ask us to change the user name. Arguably, letting the user pick a name may reduce the number of user name changes.

With the vendor IDM, user name changes are quite difficult. A user name change requires a team of 8 working for an hour. It looks like midPoint has a feature that will allow a user name change from within midPoint and then the resources will change the user name in the connected systems. If user name changes in midPoint are better than the vendor IDM, we will be more willing to drop the feature that allows a user to select a user name.

MidPoint does not have an account claim feature like the Vendor IDM. Mines has several choices here:

- Implement a new self-service account claim for midPoint. With a strong understanding of the midPoint APIs, this could be done in a few weeks. We are new to midPoint, so probably longer.

- Write code to generate a user name for the new users and mail them. This is what we did before the vendor IDM. The letter also included a key, the key and the user name were used by the user to activate the account and set a password.
- Enable a feature in Banner to generate the user name. Configure midPoint to provision accounts and wait for the user to show up and set the first password. A simpler account claim that authenticated the user with CWID, name and birth-date would still be necessary.

The IDM system that Mines used for the vendor IDM required that user names and keys were sent via postal mail. This was a tedious process to prepare the letters, stamps, generate user names, and the key. Then the user had to wait several days for it to be delivered. When Mines deployed the vendor IDM in 2015, that process changed. When they were ready, HR, Admissions, or who ever is responsible for the new user would tell the user to go to the self-service web site and click on the account claim link. The data wanders over from Banner via SPML to the vendor IDM, the vendor IDM waits for the user to show up, configures the various accounts and the user logs in and does what ever they are here to do.

There is one glitch that we would like to fix in the current account claim. New applicants that want to see the financial aid shopping sheet need to log into Luminis. Due to the way messages are processed between Luminis, Banner, and the vendor IDM, new users are told to wait 10 minutes for the account to be created. Hopefully, we can find a way to shorten that time with midPoint. It is rather silly to tell a potential customer that is considering spending $100,000 over the next 4 years to wait 10 minutes – they might get bored and go somewhere else! Luminis is the only system with a delay. Google, OpenLDAP, Active Directory, and Kerberos are ready for use in a few seconds. The user can claim the account and then use their shiny new user name and password and log into Google.

Password Filter

The active directory password filter is a small program that runs on the windows Active Directory servers. It waits for a user to change the password on a windows workstation, collects the cleartext password before it is hashed by Active Directory and sends the cleartext password to the vendor IDM. The vendor IDM can then send the password to other systems.

The AD password filter is not that complex, we could re-write it and maintain it ourselves. The other option is to take away the windows password change tools and tell the users to go to the self-service system to change a password. The IAM staff expects that taking away the windows password change tools would irritate some of the users that find it convenient.

FYI, there is a partial solution to this problem, Evolveum has a password filter that will collect passwords form Active Directory and dump them to a file. We could modify it to feed passwords to the midPoint API. See:

https://github.com/Evolveum/midpoint-password-agent-ad

https://docs.microsoft.com/en-us/aspnet/web-api/overview/older-versions/build-restful-apis-with-aspnet-web-api

Event Driven Feed from Banner

The event driven feed from Banner has many interesting uses. A user can go to the HR department, change their last name and it shows up in active directory in 30 seconds. There are no batch jobs chewing up resources every day. The only records that the vendor IDM receives are for identity records that are new or have been modified.

MidPoint has a reconciliation feature that will require access to all of the identity data in Banner. The reconciliation feature will allow midpoint to take records in every system and match them to policy and then correct discrepancies. An event driven feed does not allow access to the necessary data for the reconciliation process. One of the problems with the SPML event driven model is that there is no reconciliation feature. IAM staff have had a number of long days looking at attributes in OpenLDAP, Google, Active Directory, Banner, the vendor IDM and log files from each system to track down discrepancies.

Do I trade the event driven model for a batch model, gain convenience for the IAM staff, simplify implementation using the JDBC interface that both Banner and midPoint support, but take away a feature that is good for users? How much effort will it take to implement an event driven interface for midPoint?

Implemented Differently

The vendor IDM self-service offers features like account claim, password change, forgotten password recovery and allows the user to update answers to questions for the password recovery feature.

MidPoint has a similar list of self-service features (except for the account claim), but they are implemented in the midPoint application. The vendor IDMs self-service is a separate application. Currently, the vendor IDM admin console is blocked behind firewalls, etc. Only the self-service application is exposed to the internet. No data is kept longer than the time it takes to process it by the self-service application. All access between vendor IDM and self-service is handled by a series API calls. MidPoints admin application supports both administrative and end user operations. If the vendor IDMs self-service is successfully attacked by a criminal, the criminal will only have access to the vendor IDM, and must figure out a way to use that access through the API. In the midPoint model, the application will need to be open to the internet so that end-users can use the self-service features. If midPoint is hacked, the criminal will be much closer to Active Directory, Google, Kerberos, OpenLDAP, with administrative keys to access them. There is some security value in implementing a vendor IDM style self-service for midPoint.

MidPoint has a number of advanced features that the vendor IDM does not have. To take advantage of these features will require that the admin console be exposed to users that need access. The features include workflows that can postpone actions to get approvals for role or group assignment, assignment of entitlements, and management of organizational structure. Currently, the vendor IDM is used to manage user name, password, and a number of useful attributes. Given that we are looking at Grouper,

the new features in midPoint will probably be handled in Grouper and midPoint can be locked up behind the firewall.

Other Items

Over the past 2.5 years, the vendor IDM has collected password recovery information for many users. We need to develop an export tool and then an import tool to move that data from the vendor IDM to midPoint. Passwords and answers to recovery questions are encrypted, they will need to be decrypted before they can be moved.

It looks like we will be able to seed users in midPoint by letting it pull the list of users from LDAP. Unfortunately, this does not include data from Banner that is not in LDAP, such as account claim data. We will need to write a tool that will either import records directly from Banner or convert the existing data in the vendor IDM.