May Update

We have completed migration of most of the project infrastructure to AWS, which will be more cost effective and simpler to manage. All services have been moved and are fully functional except for Jenkins, which is being debugged to get all the jobs working again, after which we're hoping for some improvements to the frequent networking problems we have encountered in the past. Total time spent on this is over 75 hours and will hopefully not need to be repeated at quite this scale again. Core services are running on CentOS 8 now to extend the useful life of the deployment. We should be shutting down the old system within the next week. We have **not** deployed our services with region or availability zone redundancy, as that would greatly increase costs and not be an apples to apples comparison (we weren't redundant before).

The SP V3.1 release was completed, with no significant issues arising to this point. The next major hurdle is tracking the changes coming in OpenSSL 3.0, which is available in alpha form. OpenSSL has about the worst API documentation I know of (as in, there isn't any), so there's a significant risk of breakage to a lot of important code in the SP and its dependencies, so the work involved remains to be seen. I would expect this is going to be the sort of change that may result in a 4.0 update because of the impact.

I'm frankly expecting we may lose all ability to support use of certificate authentication in the SP's back channel support, but that's speculation at this stage, it's just something to watch. Moving to port 443 and signed messages is the best way to inoculate against that risk, for the few deployments still in need of SOAP support. We'll get a bug created to track this work.

We have made significant progress on drafting requirements and starting preliminary work on a new plugin design for the installer to enable more factoring of the code base. This is a big deal and it's not being undertaken lightly, but we don't see how to accomplish what we need with any off-the-shelf solution that would accomodate the freedom of deployment models we have now (i.e., we could use new tools and then write off anybody who doesn't use those tools, and we just don't believe in that approach for the core software). Spring, and particularly Web Flow, also more or less preclude some of the approaches we might otherwise consider because the architecture doesn't allow for the sort of class-loader isolation that something like OSGi would require.

Note that we do **not** plan to move to any sort of piecemeal "patch" approach to delivering bug fixes, as that's a totally chaotic and unmanageable deployment model that undermines the careful versioning policy we adhere to. But the model will accomodate detection of compatibility problems when upgrades are done, even retroactively (the compatibility rules themselves are expected to be remotely obtained). Once the basic framework exists, we probably will find new uses for it, but the focus is on delivering more manageable installation of complex add-ons.

We will be initially designing a number of deliverables to rely on this approach:

- Non-trivial storage plugins
- New (and old) scripting alternatives
- Official OIDC support, both inbound and outbound
- Duo integration via OIDC, eventually replacing the original integration

In the future, most new work that is expected to be of more limited adoption or that requires substantive new dependencies will probably take this form. We welcome interested parties to provide feedback on the dev list. Initial work on this will hopefully start showing up in V4.1.

I would also like to call attention to the fact that yet another IdP-of-last-resort for our services, UnitedID, has ceased operating. SUNET has kindly agreed to make their open IdP (eduID) available more widely. If you have existing bugs or pages you're tracking and need to rename a Jira or Confluence account name that was formerly issued by UnitedID just drop a note to our contact address and we can rename it for you.