June Update

Lots of work in progress to highlight this month, with one noteworthy concrete item:

We have formalized a set of Java distributions to support to varying degrees, and an expanded set that we are (or shortly will be) testing as part of our CI processes. The full, gory details are on the Java Distributions page, and it's also forward looking to the next version, but the boiled down information for V3 is on the updated SystemRequirements page, and spells out the Java options we are recommending going forward.

As an aid to transitioning people, we are still supporting Oracle Java 8 right now, but we don't see much of a future for that, though that could change if our members indicate they feel differently. The plan is to steer people toward other options of choice, and then officially end that support with V4; the rest of the options listed are the ones that would survive that transition.

The other thing to note is that we're formalizing the concept of "partial support", which is another differentiation for Consortium members by offering direct support for particular platforms with the understanding that we may have more limited ability to quickly reproduce those environments in the unusual case that it becomes necessary. This, we feel, best reflects our generally ambivalent attitude about platforms that "work" but that we don't have constant access to or familiarity with. Today this mostly revolves around Debian, but this gives us the ability to codify it. We also expect to expand this notion to the SP side of the world shortly.

On the development front, a number of projects continue, the bulk of which center on the design foundations for proxying support that I've discussed previously. As of this month we have completed a new service design for the attribute translation functions that were formerly controlled with the Attribute Resolver, and now also includes offloading the display metadata used during attribute release consent, so that will also be possible to supply with centralized, shareable defaults provided by federations or communities. We revamped much of the system's language handling to be more correct and honor browser settings more consistently.

The final form of the new configuration, which will ultimately be possible in a variety of ways, is not fully determined, but we're converging on it. We have no plans to deprecate the use of the Attribute Resolver's AttributeEncoder, DisplayName, etc. elements either, and they remain supported. There are situations where both old and new approaches will make sense, but the new design simplifies bulk proxying of attributes. We will probably start looking at opening the V4 documentation space so that we can start to document some of this new material and get feedback.

We have prototyped the attribute proxying features, including the use of the Attribute Filter engine in the "SP mode" of filtering data inbound as well as outbound, and this is functional when applied to simple examples like data passed in from the External login flow. Today that's a good catch all for proxying, but the next steps will be to start building more intelligent proxy login flows for the various protocols. I expect this will include CAS as a good starting example, building on the work Unicon has done for years in supporting CAS authentication to the IdP. In practice, features like filtering don't matter as much with CAS, but it's a good test case for the real thing later.

The focus is still more on delivering the foundation in V4, so it's possible much of this will remain a work in progress there, but should be more fully realized by the time V5 ships.

Other work in progress includes some prototyping of new anti-CSRF features, and work on adapting SAML SP metadata for use with OIDC relying parties, much as we have done for CAS. This extends a large range of additional IdP command and control to OIDC without having to reinvent the wheel. As the summer moves along we should have an idea soon about whether the OIDC OP support will be an updated add-on or a part of V4.