AttributeStorage

Where should an attribute be stored?

Deployers of Shibboleth quickly encounter a fundamental problem in identity management. There are a lot of attributes used by applications. Some of them represent privileges, some personally identifying information, and others serve specific functions within each application.

Central storage of attributes has a lot of proven benefits. The data don't have to be stored in a lot of locations, keeping data fresh, limiting confusion and complexity, and giving easy control over identity spaghetti. It's more secure, because only one server needs to be really hardened, and applications do less work.

There are reasons to keep attributes at or in the application too. Applications can quickly make local decisions based on information stored deep in their systems that no other application would be particularly interested in. That saves the trouble of needless transport and storage, clutter from too much information, and limits the need for involvement of central staff. Many applications also simply can't be modified to use the central infrastructure.

This is an old question, and the time-tested answer is simply that information used by a lot of applications should be stored centrally. It wouldn't make any sense to keep access control information for a CMS centrally, nor would you want the CMS to maintain a primary username. Deciding where each attribute should live is more of an art than a science, and study of which LDAP attributes have proven the most useful is a good start.

Centralization of data was the main reason for the creation of common enterprise directories and databases. It also laid very important groundwork for the introduction of single sign-on systems.

Federated identity doesn't eliminate the problem, and it doesn't change the answer much. It does make the solution a little more complicated because there are many sources of identity and many authorities. The organization controlling the CMS might be different than the one controlling the identities. The question of which entity is truly authoritative for the data becomes important, and you may not have the ability to move information inward.

Applications benefit the most if they can eliminate all need to store attributes, because it offers dramatic simplification. If this isn't possible, use of a separate database or account at the SP that is keyed by a persistent identifier sent from the IdP is the preferred solution today. This allows for a combination of attributes from the IdP and the SP to be used to give the user the best service possible.