SP Metadata Management

Approaches to SP Metadata Management

The *Software as a Service* (SaaS) model of software distribution often gives rise to the notion of client or customer. For example, sponsored partners in the InCommon Federation usually cater to customers (which are often campuses), but there are non-commercial web apps, like HathiTrust, that also fall into the SaaS category.

A frequently asked question from new SaaS providers is: Should I publish one comprehensive entity descriptor for all my customers or should I publish a separate entity descriptor for each of them? Either is fully supported by InCommon and selection of one or the other is more an art than a science. For your convenience, here we provide some information that will help you make an informed decision.

First, there are neither technical nor policy constraints with respect to endpoint locations in SP metadata. That is, a single SP entity descriptor may have any number of endpoints. There are SPs in InCommon metadata with hundreds of endpoints in a single entity descriptor.

Similarly, there are no technical limits on the number of SP entity descriptors a single organization can publish, but there are limits set by policy. Each organization is allowed up to 50 SP entity descriptors in metadata. Another 50 entity descriptors may be purchased for $1K per year. Here again we have organizations with scores of entity descriptors in metadata. In fact, one Federation participant has more than 300 SP entity descriptors in the InCommon metadata aggregate.

So it goes both ways. Which approach is best?

> The *entity descriptor* is the basic unit of policy and interoperability. If entity descriptors are handled in a common way for all your customers, then consolidation into a single entity often makes sense. A question to begin with is, “does my web app look like one common service for all customers, or does it look more like an isolated instance of the service deployed specifically for a given customer?”

If the web app does not look like one shared service environment for all customers, then the ability for policy and interoperability to vary by customer across multiple entity descriptors is valuable. Such an approach offers the greatest flexibility. On the other hand this approach can complicate documentation and requires more deployment effort for each organization and for the service provider itself.

For their part, campuses vary with respect to what they require for vendor integration. Some want to own the metadata (for business continuity reasons) but are happy to delegate the administration of that metadata to the vendor (and our Federation Manager supports that model). Others prefer a more hands-off approach and would rather leave the management of metadata entirely to the vendor. What your customers want (combined with what you, the SaaS provider, are willing and/or able to provide) will influence your approach to metadata management.

The Structure of SP Metadata

Consider the following elements in SP metadata:

- entityID
- X.509 Certificate
- User Interface Elements
- Requested Attributes
- SAML Protocol Endpoints
- Contacts

In each case, ask yourself the question: Is one such element sufficient for all my customers or does each customer require a unique such element?

Take certificates in metadata, for instance. In SP metadata, such a certificate is primarily an *encryption certificate*. Is the intention to provide each customer (which in this case corresponds to an Identity Provider) with their own encryption certificate? This depends in part on the technical characteristics of your particular deployment, and your deployment’s privacy model, but note that in the event of a compromise, it is much easier to replace a certificate for one customer rather than 50 customers. This should be a major consideration in your decision-making process.

> *Avoid sharing certificates across entity descriptors*. Not only is this viewed as a poor security practice, but some SAML implementations (such as AD FS 2.0) will simply **not** consume two entity descriptors that contain the same certificate.

As another example, consider user interface elements in metadata. Identity Providers (IdPs) use these elements to enhance the login and consent pages presented to their users. If all your customers (i.e., IdPs) share the same entity descriptor, there is no opportunity to provide personalized user interface elements, which will be viewed by some IdPs as a deployment limitation.

Single logout (SLO) is another area of concern. An SP whose endpoints are based on multiple vhosts within a single entity descriptor should probably avoid SLO. A user who explicitly logs out of one vhost will necessarily be logged out of **all** vhosts, which may or may not be intended. Consequently, SLO works best for SPs with simple entity descriptors based on a single vhost.

Similar questions may be asked for each type of element in SP metadata.

Finally, please note that delegated administration of metadata operates at the level of the entity descriptor. So if you want to spread out metadata management across multiple administrators, one entity descriptor per customer works best.