SAMLLibertyDiffs

Differences Between SAML V2.0 and Liberty ID-FF 1.2

SAML V2.0 represents a significant feature upgrade to SAML V1.1. The enhancements include features derived from the Liberty Alliance Identity Federation Framework (ID-FF) V1.2 specifications that were contributed to the SSTC in 2003, capabilities present in the Internet2's Shibboleth architecture, and enhancement requests resulting from experience with numerous deployments of SAML V1.x in the industry.

The on-the-wire representations of SAML V2.0 assertions and protocol messages are incompatible with Liberty ID-FF processors. As is explained in the SAML Assertions and Protocols specification, only new major versions of SAML (of which this is one) typically cause this sort of incompatibility. In this release, much of the incompatibility is syntactic in nature; this was done for consistency and better component symmetry.

The following sections analyze the differences between SAML V2.0 and ID-FF V1.2 and provide guidance and other commentary, in order to aid developers and deployers who are undergoing an upgrade or need to support multiple versions at once. The analysis results are stated in the following (somewhat subjective) terms, which are not mutually exclusive:

- Same: SAML's approach is largely identical to Liberty's approach, including close similarity in specification text and even syntax to a large degree (though it cannot be assumed to be identical; at the very least, the markup resides in a different namespace).
- Equivalent: SAML's approach is functionally equivalent, even if achieved in a different manner structurally.
- More functional: SAML has generalized the Liberty functionality to account for more options or use cases.
- Different: SAML has significant structural differences from Liberty due to the refactoring activity done as part of the design and convergence effort for SAML V2.0.

Representation of Principals

<Subject> and <NameID>

Different, More Functional

The <NameID> element has been substantially enhanced to combine the information carried in SAML V1.1, Shibboleth, and the extensions added to <saml:Subject> in ID-FF. As in ID-FF and Shibboleth, specific Format values are used to connote identifiers designed with privacy-preserving properties in mind. Both persistent and so-called transient identifiers can be used, corresponding to the ID-FF federated and onetime identifiers, respectively.

ID-FF overloaded all non-transient identifiers into a single Format value, regardless of their privacy characteristics. For example, an employee ID number might be used instead of a pseudonym. SAML reserves the use of the persistent Format URN for pseudonyms having the pair-wise characteristics of ID-FF federated identifiers. Another URN MUST be used when violating the privacy or pair-wise semantics. Deployers can choose specific kinds of identifiers and enable and disable their use as needed.

When using any form of identifier (whether privacy-preserving or not), SAML V2.0 also incorporates the ability from ID-FF to qualify the identifier both in terms of the asserting party and the relying party by adding an SPNameQualifier to the original NameQualifer attribute. Also, any identifier can carry a second string identifier established by the relying party as an alias, termed the SPProvidedID. This eliminates the two-part subject structure created in ID-FF.

Commentary:

- Implementation or deployment guidance should specify use of the privacy-preserving formats when privacy is an issue.
- As with ID-FF V1.2, attention will be needed to address interoperation of SAML V2.0 with ID-FF V1.1 and V1.2 in terms of representing a single principal in all three message types. However, a direct mapping from ID-FF V1.2 to SAML V2.0 should be possible in most cases.

Encrypted Identifiers

Different, Equivalent

ID-FF defined a mechanism to hide encrypted identifiers inside standard ID-FF identifiers by encoding the XML Encryption content. SAML V2.0 permits direct use of XML Encryption in various places, including an <EncryptedID> element that can replace the usual <NameID> element. ID-FF's confusing rules for using NameQualifier in different ways in the encrypted case are gone.

Single Sign-On Profiles

<AuthnRequest>

Different, More Functional

The <AuthnRequest> protocol message in SAML is somewhat revised, but is a superset of the ID-FF message. The main difference relevant to ID-FF use cases is a revised <NameIDPolicy> element that addresses the ability to request specific principal representations. The other enhancements are intended for advanced use cases in the future in which assertions need to be tailored by the relying party for their intended use by including additional subject confirmations or conditions.

Commentary:

- Some of the advanced capabilities supported by the <AuthnRequest> message are permitted in the SAML SSO profiles, such as the ability to specify arbitrary <Conditions> in the assertion, that were not permitted in ID-FF.

Browser SSO

Different, Equivalent

All of the existing SAML V1.1 and ID-FF profiles for browser SSO have been merged into a single basic profile with support for different bindings. The general characteristics of the profile align well with ID-FF assumptions. For example, when using the new POST binding, the assertion is signed, rather than the response (a change to SAML, but not to ID-FF). All of the device accommodations in the ID-FF profiles are captured in the defined Redirect, POST, and Artifact bindings.

Commentary:

- You should check on the level of support for your required bindings among your identity federation partners.

Enhanced Client SSO

Different, Equivalent

The LECP use case in ID-FF is rended in a redesigned profile called ECP that uses SOAP and PAOS. It is functionally the same, but uses SOAP and SOAP header blocks to carry the information the ID-FF profile places inside custom XML envelopes. The most significant difference is that the interaction with the SP is via PAOS and not POST. This is a change, but an ID-FF SP could not support LECP before without explicit changes anyway.

Proxying

Same

Much of the specification language about proxying SSO is very similar to ID-FF's text. Some additional policy controls on proxying are supported in SAML, but the overall approach is about the same. Note that the reliance on the Authentication Context specification to carry the list of providers is removed.

Single Logout Profiles

Same

Logout is largely unchanged from ID-FF. All of the Liberty-generated errata around logout request expiration, `SessionIndex`, proxy failure, and so on have been incorporated into SAML. The distinct profiles have been combined into a single binding-independent profile, but the overall functionality is basically the same.

Name Identifier Registration and Federation Termination Profiles

Different, More Functional

SAML V2.0 combines the two protocols in ID-FF V1.2 into a single protocol for updating or terminating use of identifiers. Any kind of identifier (as opposed to just the federated variety) can be updated or terminated. The distinct profiles have been combined into a single binding-independent profile, but the overall functionality is basically the same.

Name Identifier Mapping

Different, More Functional

SAML V2.0 generalizes the mapping protocol by using the `<NameIDPolicy>` element to describe the properties of the identifier to be returned. This allows for arbitrary mappings between any two formats, even allowing "create" or "don't create" semantics that match the behavior during SSO. It's therefore possible to programmatically "federate" a principal explicitly using this protocol.

URL Encoding of Messages

Different, Equivalent

The per-message piecemeal encoding of XML into URL parameters was replaced in SAML V2.0 with a scheme based on compressing the actual XML with the DEFLATE algorithm (used by gzip). Signing is still permitted in a similar fashion, with somewhat simpler processing rules.

Metadata

Different, More Functional

SAML V2.0 metadata is substantially different in format from ID-FF, but is a functional superset in most respects. Support for SAML profiles unsupported by ID-FF are captured, such as SAML attribute exchange and authorization decisions. Problems with schema consistency, encryption metadata, and multi-endpoint support for a single protocol have been corrected. In the common case, a single SOAP endpoint might be duplicated in many metadata elements if it is used for many profiles, so a "space for clarity" trade-off was chosen to support cases in which the endpoint is not shared.

Commentary:

- All language about use of DNS, zone signatures, etc. is tightly bound to the DNS-based exchange profile, and it is not called out as a preferred or primary means of exchange. Check to see if it is supported in your chosen implementation.
- Since Liberty ID-WSF carries some of its bootstrapping information in SAML attributes, use of SAML attribute-oriented metadata might be useful when integrating SAML deployments with ID-WSF.

Authentication Context

Different, More Functional

Though mostly intact from ID-FF, some key changes include the removal of SAML `AuthenticationMethod` in favor of the `<AuthnContext>` element; support for including only context classes in an assertion, and omitting specific context declarations (formerly statements); and the ability to determine class conformance by a declaration instance using schema validation, enabling machine-processable conformance checking.