Relating OIDC to a Multilateral federated SAML (aka SAML*) world

Basic Model:

The basic model for the SAML* perspective is User – Web Browser - IdP – SP.

OIDC came after SAML* and into a mobile world. As a result the basic model for the OIDC perspective is User – Client Device or App – IdP – SP. In OIDC terminology, User (=Resource Owner) – Client – OP – RP

- As a result, the orientation of SAML* is towards transactions and OIDC to persistence.
- As a result, OIDC tokens are designed small and in JSON instead of XML.
- As a result, OIDC permits client-side SSO (e.g. among various Google apps) and a persistent login

Deployment Challenges:

For both, setting up an IdP/OP can be difficult. For SAML*, setting up an SP can be difficult, while for OIDC there are lightweight deployment options that are expanding.

Flows and Consent:

For SAML*, the orientation is for temporary transaction permissions, with the choice of implementing and/or suppressing consent.

For OIDC the perspective is for more persistent delegated permissions, even when the user is offline, with the user having the choice of revoking that consent at some point out of band.

Consent is "mandatory" (i.e. a required API parameter) for OIDC. In SAML* it is optional but seldom deployed.

In both, the nature and implementation of consent is out of scope.

Trust:

For SAML*, trust determination of participants is a big deal and that includes which attributes should be released to an SP

OIDC envisions dynamic client registration and is silent on how those trust decisions are made on whom to trust and what information to release to them.

Attributes:

For SAML*, the set of attributes that are exchanged are defined by the federated community and may include some normative schema.

For OIDC, there are a few normative profiles of attributes in the spec itself and a mechanism for defining other bundles of attributes to be exchanged.

For SAML* the attributes tend to meet specific vertical needs (e.g. R&E, Law Enforcement); for OIDC the attributes tend to meet to meet general and social needs.

Metadata:

SAML* makes heavy use of shared metadata to describe characteristics of the organizations and applications participating in transactions.

OIDC has less notion of broadly interacting participants and shared metadata. Key characteristics are either defined in specifications or conveyed within individual exchanges.

Discovery:

Discovery refers to the need for a relying party to find out the who the user's federated identity provider. It is often solved with a list of options for the user to select from, either as a pull down list, icon to click on, field for a user to fill out, etc.

In SAML*, discovery is usually built on top of out of band use of federated metadata of possible IdP's.

In OIDC, discovery can be done either as an out of band process or dynamically as part of OIDC. In the latter case it must use a webfinger protocol to find the identity provider (OP), which raises issues of access control.

Integrating OIDC into a SAML* world

What does integration mean: Many possibilities:

Integrated UI – user has a consistent experience regardless of underlying technologies

Integrated OIDC token management - ability of a SAML* IdP to issue and manage OIDC tokens

Integrated Trust Management - add dynamic client registration to a multilateral trust fabric

Integrated Attributes – create OIDC attributes and scopes (i.e. bundles) to mirror our own?

Others?