AuthZ Thread Summary

#	Issue	Comment
1	Protocols	What protocols should be supported for interactions between the PEPs within the relying party application and the PDP? OAuth? XACML? What protocols should be supported for interactions between the PDP and the PIPs? What if the PIP is Grouper?
2	Integration	It seems too complex to insist on call-outs to a PDP from each point within an application that needs to enforce some form of access control; what might make app integration easier?
3	Groups	Aren't groups enough? That is, practically speaking, an app can base most of its access control decisions on a set of group memberships that are made available in the SAML assertion at the time the user shows up.
4	Entitlements	An alternative to groups (which tend to put access control logic at the application end) is entitlements (which tend to put access control logic at the IdP end)
5	Rules	Are there situations in which it is beneficial to go beyond groups and entitlements and manage access via policy rules that are evaluated at the time the user seeks to perform a particular action on a particular resource?
6	Prior art	There are a number of software packages and defined processes with which rule-based access control could be implemented today
7	BGP as analogue	We need something like the Border Gateway Protocol (BGP) before we can ramp up support for distributed/federated authZ
8	Policy Management	How can we give people the ability to create and manage policies in an intuitive and easy way?
9	User issues	How do we let users know what resources they have access to? what the RP/SPs policy is? In general how to give the user appropriate control over the process
10	Containers	Change the containers, not the apps, then integration challenges are easier to deal with (Django/VOOT as an example)
11	Flows	Is the desired flow AA --> RP --> PDP or AA --> PDP --> RP?
12	Identifiers	If the IdP is not the sole source of access control information (i.e., if an independent PIP carries relevant info), how do you correlate the identifier from the IdP with the identities known in the PIP repository?
13	Identifiers not needed	There are approaches in which it is not necessary to map identifiers
14	Privacy	In situations where correlation is needed to link IdP identifiers to PIP identities, it is common to rely on ePPN; This raises user privacy concerns. Sites that have this concern tend to favor evaluating the authorization rules at the IdP and passing only entitlements (or group memberships) to the SP.
15	AuthN vs Attributes	The IdP has a dual function: 1) Getting some component to authenticate the user (local WebSSO or social2SAML gateway or whatever) and 2) garnering a set of attributes to be passed to the SP in the SAML assertion. It confuses the picture when we call the authenticating component an IdP. In other words, Brendan's approach is not "chaining IdPs": There's only one IdP and a multiplicity of authenticating components depending on which credentials the user has available. The site IdP can configure its attribute resolver to decorate its SAML assertion with additional site-maintained authZ information. The attribute resolver embedded in the Shib IdP is an Attribute Authority (AA), it's just packaged together. More often when we have talked about AAs, it's in the context of an SP aggregating attributes obtained from the IdP with additional information from an independent AA.