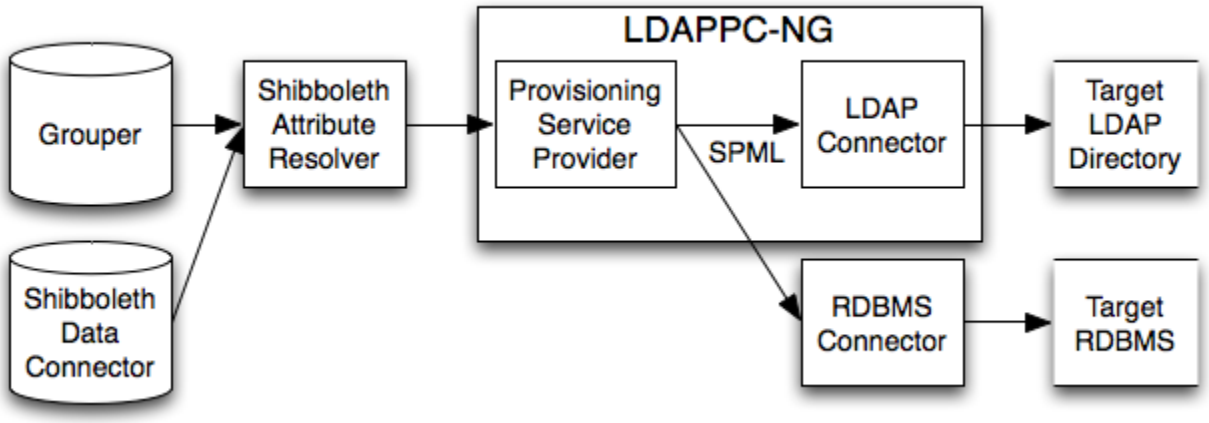
LDAPPCNG Documentation



Overview

LDAPPCNG serves as an SPMLv2 "provider" and LDAP "target" based on the Shibboleth Attribute Resolver as well as the VT Ldap and OpenSPML libraries.

LDAPPCNG provisions objects to targets. Objects consist of identifiers, attributes, and references to other objects. Group memberships are considered to be references.

Provisioned objects are calculated from source data processed by the Shibboleth Attribute Resolver. The Shibboleth Attribute Resolver accepts many DataConnectors (sources), including LDAP, RDBMS, and Grouper. Custom DataConnectors may be provided.

LDAPPCNG provisions targets using a standard provisioning language, SPML. An SPML to LDAP connector is provided. Provisioning non-LDAP targets requires a target specific connector, for example, SPML to RDBMS.

Currently, LDAPPCNG supports SPML requests represented as Java objects as provided by the Oasis implementation. The SPML requestor is Grouper's CLI, gsh. A future iteration should be able to send and receive SPML via web services.

Requests and Responses

The Provisioning Service Provider implementation class, PSP.java, accepts and returns SPML requests, which are similar to LDAP directory operations : add, delete, modify, lookup, and search.

Most SPML requests consist of an object identifier, target identifier, and "return data". The return data is either "identifier", "data" (identifier+attributes), or "everything" (identifier+attributes+references).

LDAPPC defines custom requests : calc, diff, sync, bulkcalc, bulkdiff, and bulksync.

Most Responses consist of a status code ("success" or "failure") and representations of provisioned objects.

calc

Upon receipt of a CalcRequest, the PSP will calculate how an object (or objects) should be provisioned via the Attribute Resolver, and will return the provisioning represented as a CalcResponse.

diff

Upon receipt of a DiffRequest, the PSP first performs a CalcRequest in order to determine how objects *should* be provisioned. Then, the PSP queries each target to determine how objects *are* provisioned. The PSP returns a DiffResponse representing the changes necessary to synchronize the provisioned objects. The changes consist of AddRequests, DeleteRequests, or ModifyRequests.

sync

Upon receipt of a SyncRequest, the PSP first performs a DiffRequest to determine provisioning changes. Then, the PSP requests targets to perform the changes, and returns the results as a SyncResponse.

bulk*

Bulk requests operate on all groups, and are similar to the pre-NG (LDAPPC) style of operation.

Custom Extensions

LDAPPC's configuration is modeled upon Shibboleth's use of Spring, which will allow deployers to customize targets as well as the identifiers, attributes, and references returned from the Attribute Resolver.

To provision a target other than LDAP, an implementation class must be able to handle add, delete, modify, lookup, and search requests.

The supplied LDAP target essentially maps SPML to LDAP operations.

Configuration Example

The following is an example of how a group might be provisioned.

The `ref` attribute of `<identifier>`, `<attribute>`, and `<reference>` elements refers to the `id` of an attribute as defined in the Attribute Resolver configuration. If the `ref` attribute is not specified, it defaults to the value of the `name` attribute.

```
<object id="group" ... >
  <identifier ref="groupDn" baseId="ou=groups,dc=grouper,dc=edu" />
  <attribute name="objectClass" />
  <attribute name="cn" />
  <references name="member" >
    <reference ref="members" toObject="group" />
  </references>
</object
```

The corresponding Attribute Resolver configuration will contain :

```
<resolver:AttributeDefinition id="cn" xsi:type="ad:Simple" sourceAttributeID="name">
  <resolver:Dependency ref="GroupDataConnector" />
</resolver:AttributeDefinition>

<resolver:DataConnector id="GroupDataConnector" xsi:type="grouper:GroupDataConnector" />
```