Permissions API suggestion based on Grouper permissions

This is based on the Grouper API for central permission management, though genericized. Note that this suggestion is not a suggestion on what the API should be, it is what data needs to go in and out. If we use SAML or XACML or whatever that is fine.

Objects

Subject

Data for something that can be assigned privileges, returned from the privilege server

```
<subject>
  <id>12345</id>
  <source>pennperson</source> <!-- subjects could be read from multiple sources, e.g. an HR system, and a list
of external shib eppns -->
  <name>John Smith</name>
  <description>John Smith (12345, jsmith) Staff, Director of Human Resources</description>
  <attributes>
    <attribute>
      <name>pennkey</name>
      <values>
        <value>jsmith</value>
      </values>
    </attribute>
  </attributes>
</subject>
```

- id is a unique id for the source which does not change
- source is one of a few sources of subjects (e.g. in institution, and external)
- name is a name to display
- description is configurable for the site or source, it could be blank, the same as name, or something descriptive
- attributes are multivalued and could be anything

Subject lookup

Used to lookup a subject in the system

```
<subjectLookup>
  <id>12345</id>
  <source>pennperson</source>
  <identifier>jsmith</identifier>
</subjectLookup>
```

- pass in either an id or identifier. Identifier is configured to lookup the subject, e.g. by netId
- source is optional though if provided makes the query more efficient and more reliable if the same id if used across multiple sources

Subject abbrev

Used to point a subject object to save space

```
<subjectAbbrev>
  <id>12345</id>
  <source>pennperson</source>
</subjectAbbrev>
```

Role

Used to represent a role associated with the user and permissions. A role lives in a folder which an end user has control of. It has a system name which should not change (or should not change frequently), and a display name. An application will define roles for that application, or perhaps share them from other applications.

```
<role>
  <namespace>
    <folder>penn</folder>
    <folder>apps</folder>
    <folder>someApp</folder>
  </namespace>
  <name>users</name>
  <displayName>Users</name>
</role>
```

Role lookup

Used to lookup a role

```
<roleLookup>
  <namespace>
    <folder>penn</folder>
    <folder>apps</folder>
    <folder>someApp</folder>
  </namespace>
  <name>users</name>
</roleLookup>
```

Permission resource

Resource assigned to a subject in the context of a role and action. Each permission has a hierarchical namespace, a system name which should not change much, and a display name which can change. The application either re-uses common permission resources (e.g. an org tree), or creates its own resources to protect (e.g. login.jsp if it is screen centric)

```
<permission>
  <namespace>
    <folder>penn</folder>
    <folder>apps</folder>
    <folder>someApp</folder>
  </namespace>
  <name>org1</name>
  <displayName>Org 1</displayName>
</permission>
```

Permission resource lookup

Lookup a permission resource

```
<permissionLookup>
  <namespace>
    <folder>penn</folder>
    <folder>apps</folder>
    <folder>someApp</folder>
  </namespace>
  <name>org1</name>
</permissionLookup>
```

Application

Collection of roles, permissions, etc. paidTimeOff is one application among many at the institution. This concept make it natural for a query of: give me all the permissions that a user has for this application irrespecitive of which role it is associated etc. So roles and permission resources (e.g. a new parent resource which implies leaf nodes) could be added at run time.

```
<application>
  <namespace>
    <folder>penn</folder>
    <folder>apps</folder>
    <folder>someApp</folder>
  </namespace>
  <name>paidTimeOff</name>
  <displayName>Paid Time Off</displayName>
</application>
```

Application lookup

Lookup an application

```
<applicationLookup>
  <namespace>
    <folder>penn</folder>
    <folder>apps</folder>
    <folder>someApp</folder>
  </namespace>
  <name>paidTimeOff</name>
</applicationLookup>
```

Permission assignment

This object is returned from the permissions server. This is the assignment of a permission resource to a subject in the context of an action, and could have attributes associated with it.

```
<permissionAssignment>
  <permission>
    <namespace>
      <folder>penn</folder>
      <folder>apps</folder>
      <folder>someApp</folder>
    </namespace>
    <name>org1</name>
    <displayName>Org 1</displayName>
  </permission>
  <action>read</action>
  <role>
    <namespace>
      <folder>penn</folder>
      <folder>apps</folder>
      <folder>someApp</folder>
    </namespace>
    <name>users</name>
    <displayName>Users</name>
  </role>
  <subjectAbbrev>
    <id>12345</id>
    <source>pennperson</source>
  </subjectAbbrev>
  <active>T|F</active>
  <attributes>
    <attribute>
      <name>ipAddress</name>
      <values>
        <value>1.2.3.4</value>
      </values>
    </attribute>
  </attributes>
</permissionAssignment>
```

Basically things here are optional. You need the subjectAbbrev, and the permission. The action is optional, the role, the attributes.

API

From a high level, I think two things would be useful, a web service (request/response), and a real time message format (e.g. over XMPP).

Web service

The client should be able to ask:

1. Give me all the permissions for an application
2. Give me all the permissions for an application for a user (possibly in a certain role)
3. Give me the permission for an application for a user for a permission (and possibly an action). This is analagous to "hasPermission"

Anyways, these could be handled in one operation, basically if the input is blank, then all will be returned (e.g. dont sent any actions to query to get all).

Input:

```
<permissionRequest>
  <applicationLookup>
    <namespace>
      <folder>penn</folder>
      <folder>apps</folder>
      <folder>someApp</folder>
    </namespace>
    <name>paidTimeOff</name>
  </applicationLookup>
  <subjectLookups>
    <subjectLookup>
      <id>12345</id>
      <source>pennperson</source>
      <identifier>jsmith</identifier>
    </subjectLookup>
  </subjectLookups>
  <actions>
    <action>read</action>
  </actions>
  <active>T|F|A</action>
  <roleLookups>
    <roleLookup>
      <namespace>
        <folder>penn</folder>
        <folder>apps</folder>
        <folder>someApp</folder>
      </namespace>
      <name>users</name>
    </roleLookup>
  </roleLookups>
  <permissionLookups>
    <permissionLookup>
      <namespace>
        <folder>penn</folder>
        <folder>apps</folder>
        <folder>someApp</folder>
      </namespace>
      <name>org1</name>
    </permissionLookup>
  </permissionLookups>
  <attributes>
    <attribute>
      <name>ipAddress</name>
      <values>
        <value>1.2.3.4</value>
      </values>
    </attribute>
  </attributes>
</permissionRequest>
```

Output:

```
<permissionResponse>
  <subjects>
    <subject>
      <id>12345</id>
      <source>pennperson</source>
      <name>John Smith</name>
      <description>John Smith (12345, jsmith) Staff, Director of Human Resources</description>
      <attributes>
        <attribute>
          <name>pennkey</name>
          <values>
            <value>jsmith</value>
          </values>
        </attribute>
      </attributes>
    </subject>
  </subjects>
  <application>
    <namespace>
      <folder>penn</folder>
      <folder>apps</folder>
      <folder>someApp</folder>
    </namespace>
    <name>paidTimeOff</name>
    <displayName>Paid Time Off</displayName>
  </application>
  <permissionAssignments>
    <permissionAssignment>
      <permission>
        <namespace>
          <folder>penn</folder>
          <folder>apps</folder>
          <folder>someApp</folder>
        </namespace>
        <name>org1</name>
        <displayName>Org 1</displayName>
      </permission>
      <action>read</action>
      <role>
        <namespace>
          <folder>penn</folder>
          <folder>apps</folder>
          <folder>someApp</folder>
        </namespace>
        <name>users</name>
        <displayName>Users</name>
      </role>
      <subjectAbbrev>
        <id>12345</id>
        <source>pennperson</source>
      </subjectAbbrev>
      <active>T|F</active>
      <attributes>
        <attribute>
          <name>ipAddress</name>
          <values>
            <value>1.2.3.4</value>
          </values>
        </attribute>
      </attributes>
    </permissionAssignment>
  </permissionAssignments>
</permissionResponse>
```

Messaging

Message to send to an application for a real time update (note, I think XACML can be used for this... so this format is really just to express the data elements)

```
<permissionMessages>
  <subjects>
    <subject>
      <id>12345</id>
      <source>pennperson</source>
      <name>John Smith</name>
      <description>John Smith (12345, jsmith) Staff, Director of Human Resources</description>
      <attributes>
        <attribute>
          <name>pennkey</name>
          <values>
            <value>jsmith</value>
          </values>
        </attribute>
      </attributes>
    </subject>
  </subjects>
  <application>
    <namespace>
      <folder>penn</folder>
      <folder>apps</folder>
      <folder>someApp</folder>
    </namespace>
    <name>paidTimeOff</name>
    <displayName>Paid Time Off</displayName>
  </application>
  <permissionMessage>
    <operation>add|remove|change<operation>
    <permissionAssignment>
      <permission>
        <namespace>
          <folder>penn</folder>
          <folder>apps</folder>
          <folder>someApp</folder>
        </namespace>
        <name>org1</name>
        <displayName>Org 1</displayName>
      </permission>
      <action>read</action>
      <role>
        <namespace>
          <folder>penn</folder>
          <folder>apps</folder>
          <folder>someApp</folder>
        </namespace>
        <name>users</name>
        <displayName>Users</name>
      </role>
      <subjectAbbrev>
        <id>12345</id>
        <source>pennperson</source>
      </subjectAbbrev>
      <active>T|F</active>
      <attributes>
        <attribute>
          <name>ipAddress</name>
          <values>
            <value>1.2.3.4</value>
          </values>
        </attribute>
      </attributes>
    </permissionAssignment>
  </permissionMessage>
</permissionMessages>
```

sdfsadf