RESTful API ideas

RESTful API Ideas from Jim Fox, University of Washington, October 2012

I promised some ideas on a RESTful permissions API. Here they are:

(Bear in mind that our present access control service at UW is SOAP
RPC based. And its organization is different than I describe here.
So I'm looking to move us to something like this as well.)

Imprecise definitions:

A permission grants access to do something.
A permission may be assigned to one or more roles.
A role is a thing to which permissions are granted.
A role contains none or more subjects who have the role's permissions.
A limit is a qualifier on a permission.

Resources and APIs:

Subject. A person, application or thing. A subject has zero or more roles.

resources: id, etc. (like group members)
Subjects are referenced by, but not managed by, the permission system.

e.g. GET (root)/subject/(subject_id)

Role. Essentually a thing with members (subjects who have the role).

resources: id, name, description, member_subjects

e.g. GET/PUT/DELETE (root)/role/(role_id)
GET/PUT/DELETE (root)/role/(role_id)/member/
GET/PUT/DELETE (root)/role/(role_id)/member/(subject_id)
GET (root)/role/search? permission_id=(id) & member=(id) & limit=(id)

Permission. Authorization for subjects in a role to do something. Roles are assigned to a permission.

resources: id, name, description, member_roles

e.g. GET/PUT/DELETE (root)/permission/(permission_id)
GET/PUT/DELETE (root)/permission/(permission_id)/role/
GET/PUT/DELETE (root)/permission/(permission_id)/role/(role_id)
GET (root)/permission/search? role_id=(id) & member=(id) & limit = (id)

Limit. Qualification on a permission. Limits the authorization.

The limits are a complication. They may have particular meaning
to only some applications. As such the authorization service may
not be able to interpret the limits, and is confined to allowing
users to edit the names and values, and to pass the limit to
requesting applications.

e.g.:

role = budget approver,
limit = budget nos xxx, yyy, zzz.
limit = $10,000
etc.

A limit is assigned to the couple (permission,role).

resources: id, name, description, name, permission_id, role_id, type, value

e.g. GET/PUT/DELETE (root)/limit/(limit_id)
GET (root)/limit/search? permission_id=(id) & role_id=(id)

Does a user have permission do do something?

GET (root)/permission/(permission_id)/has_permission/(subject_id)

returns

200 if the subject, through its roles, has the permission
in addition returns a list of limits

404 if subject does not have the permission

possibly query string arguments could cause the service to do some
filtering on the more well defined limits.

Possible implementation.

A role is a group, with subjects as members.
A permission is a group, with roles as members.
(efficiency might put subjects directly in the permission group)
"Does a user have a permission" is an effective member search on the permission group.

Limits I think are outside the group model.