Stateless Embargo Authorization Support

See also the associated JIRA Issue: DS-908 Embargo Overhaul: Utilize ResourcePolicy Start and Stop datestamps for enforcing embargo in DSpace

I would like to readdress how the current embargo is implemented in DSpace, this Issue comes up because we have started working on an Embargo solution that actually uses the start/end dates of resource policies to enforce the embargo rather than a cron tab script.

This approach is currently under deployment/test in IDEALS and based on existing embargo work that was completed there.
The new proposed approach would allow for Embargo to be applied at either the Item level or individual Bitstream levels as a series of ResourcePolicies that use start/stop datestamps currently on the ResourcePolicy object and does not require executing a cronjob to adjust the state of the Item Thus the record and enforcement of embargo is stateless. Being stateless means that the policies do not change over time, only the resulting outcome of the temporally sensitive evaluation that enforces of those policies changes.

The AuthorizationManager already supports the enforcement of timeframes in ResourcePolicies. I would like to propose that we expand ResourcePolicy in the following manner:

1.) To be a better DSpace Domain Model citizen and that would include having name and description fields available to define the reason for a resource policy being set.
2.) Establishment of a RESTRICT Action that would be enforced by the AuthorizationManager to allow for "Explicit Definition" of the Embargo Policy on Annonymous Users.

For example:

Bitstream A --> ResurcePolicy(Action=RESTRICT, Group=Annonymous, start=20110101, end=20120101, name="Embargo", description="Embargoed as required by publisher.")

Bitstream A --> ResurcePolicy(Action=READ, Group=UniversityAffiliates, start=null, end=null, name="Local University Affiliates", description="Local University Affiliates are Exempt from Embargo Restriction.")

The previous example would enforce Embargo and Access rights "Explicitly" and "Clearly" in the Policies attached to the Bitstream and/or Item. The AuthorizationManager may need minor enhancement to address "inheritance" of ResourcePolicies assigned on parent Items. It may be advisable to use such inheritance to enforce "DEFAULT_XXX" policies rather than copying them into place on each and every Bitstream/Bundle and Item created, this will reduce the "bloat" of ResourcePolicies currently in effect in the existing system.

And important benefit of these changes to ResourcePolicies and the underlying AuthorizationManager framework are that they can then be used to encode the explicit technical or administrative metadata sections into the AIP or METS manifests concerning the Policies that are in effect on the Item and its contents. Adjustments to the DSpace SIP Profile to capture enforcement of embargo details by consumers of those tools would be more clearly expressed and machine automatable than dumping it intot he metadata. Achieving Machine actionability means that Ingest Packagers and services that rely on them can define a more concrete business logic to be maintained.

As we evolve the Metadata capabilities to support system/tech/admin/descriptive metadata sections for all parts of the item, we can consider that the ResourcePolices will inform the production of metadata about the embargo state of the Item being exposed in OAI / SWORD /METS packagers and so-on. But for now, we really need to set a standard that actual Resource Policies be the mechanism that enforces the access rules/policies within the system and not some metadata field set in the item metadata description.

Somewhat a concern is how other areas of DSpace treat ResourcePolicies rather bluntly. Recommend that ResourcePolicies should be managed in central manner (such as ResourcePolicyService: or "ResorcePolicyManager") such that the manner in which policies are enforced or allowed to be edited does NOT cause emergent conflicting behavior across different parts of the system such as those described within DS-906 and DS-525.

According the DS-525, the issue of embargoed items is documented as a warning in our Documentation: https://wiki.duraspace.org/display/DSDOC/System+Administration#SystemAdministration-Movingitems

I consider this documentation insufficient as a solution to the problem of embargo permissions getting overridden in the mapping. A more appropriate solution would show to the user the exact changes that would happen to the item and allow them to decide which policies should be enforce/changed on the item.