Determining the Effective Authorization Using WebAC

This page describes in detail how the effective ACL for a resource is found, and how the request is authorized using that ACL.

Steps in determining the effective authorization

Finding the ACL

1. Get the ACL of the requested resource, if exists, else.
2. Get the ACL of the next ancestor recursively (using either ldp:contains or fedora:hasParent), if exists, else.
3. If no more ancestor exist (root node reached) the default root ACL **allows reads**.

Finding the effective authorization

1. Find union of authorizations that specify access for the requesting **user**. This includes:
    a. authorizations that specify *accessTo* to the requested **resource**.
    b. authorizations that specify *accessToClass* of the requested **resource type**.
    c. If authorizations exist for user, go to step 6, else go to next step.
2. Find union of authorizations that specify access for the requesting **user's group**. This includes:
    a. authorizations that specify *accessTo* to the requested **resource**.
    b. authorizations that specify *accessToClass* of the requested **resource type**.
    c. If authorizations exist for group, go to step 6, else go to next step.
3. Find union of authorizations that specify access for the requesting **user**. This includes:
    a. authorizations that specify *accessTo*to the requested **resource's ancestor**.
    b. authorizations that specify *accessToClass* of to the requested **resource's ancestor type**.
    c. If authorizations exist for user, go to step 6, else go to next step.
4. Find union of authorizations that specify access for the requesting **user's group**. This includes:
    a. authorizations that specify *accessTo* to the requested **resource's ancestor**.
    b. authorizations that specify *accessToClass* of to the requested **resource's ancestor type**.
    c. If authorizations exist for group, go to step 6, else go to next step.
5. If no authorization exists for user or group: **Allow Access.**
6. Use the least permissive from the set of authorizations found.
    a. if the authorizations permit requested access mode: **Grant access.**
    b. if the authorizations do not permit requested access mode: **Deny access.**

Example Request Authorization Flow