Managing Authorization info in Angular UI

In Nov/Dec 2019, Additional discussion on this topic has occurred from the REST API standpoint, see REST Authorization.

Also see the new authorization endpoints in REST Contract:

- "authorizations" endpoint: https://github.com/DSpace/Rest7Contract/blob/master/authorizations.md
- "features" endpoint: https://github.com/DSpace/Rest7Contract/blob/master/features.md
- (Also related: "resourcepolicies" endpoint: https://github.com/DSpace/Rest7Contract/blob/master/resourcepolicies.md)

This page is meant for brainstorming & idea gathering. Currently a final solution does not exist. Once it does, this page can be replaced or removed.

Problem

1. Ideally, we would not like to hardcode "roles" or permission levels into the Angular UI itself.
    a. Some areas of the UI currently hardcode roles, e.g. https://github.com/DSpace/dspace-angular/issues/393
2. Instead, we'd prefer all authorization information (for currently logged in user) be passed to the UI layer via the REST API response (or cached as needed based on previous responses, e.g. a user's group membership may need to be cached on login)
3. There is no clear single solution for all scenarios/needs in the UI layer. We may need to use different solutions for different scenarios (uncertain)
    a. More background info here: https://github.com/DSpace/dspace-angular/issues/242

Possible Solutions / Partial Solutions

1. For some areas of the UI, it may be relatively easy to describe authorization rights using predefined HAL `_links` (i.e. if a link is present you can perform a specific action) in the latest REST response.
    a. For example, on an Collection page, the UI could display an "Edit" link *only if* a HAL link named "edit" is provided in the Collection's REST response. This may be an easy way for the backend to return the currently logged in user's access *on a single object* by just specifying which links are available to that user.
2. For some areas of the UI, it may be necessary to check a user's Group membership to determine authorization rights
    a. For example, Administrative options (at the Site level) should only be displayed if the user is a member of the "Administrator" group.
    b. This group membership information may need to be cached in the UI layer, as it won't be available in every response.