University of Wisconsin - Madison - External Authentication and Authorization

Use Case 1: External Authentication and Authorization

Title (goal)	External Authentication and Authorization
Primary Actor	developer, consuming applications
Scope	Organizational, black box
Level	
Story	1. User on a browser clicks on a link to see information about a digital object in Fedora. The request will pass through a few layers of front-end applications before it reaches Fedora. ○ No user information is submitted with the request; it's an anonymous request from an unauthenticated user. 2. Fedora receives the anonymous request for the resource (object, datastream, datastream metadata, etc.). It asks the external PDP if this resource is accessible; no role attributes are delivered to the PDP (an anonymous, public request). ○ Is the source IP for the request is passed along, for access control? ○ PDP needs access to the object; policy may be evaluated based on a properties of the object. a. Resource is available: PDP responds with "yes". Fedora sends back the requested resource, with a HTTP 200 response code. Work is done. b. Resource has restricted access. PDP responds with "no". Fedora sends back a HTTP 401: Unauthorized. At this point, the front-end application decides what to do with that 401: in our case, it will redirect the user to an authenticating web service, protected by Shibboleth; the authentication web service will do the Shibboleth dance, then redirect the user back to the front end web application, with user attributes included. 3. The front-end web web application will re-request the resource from Fedora, this time with user attribute information. 4. Fedora receives the authenticated request for the resource. It asks the PDP again if the resource is accessible, this time passing along user attributes. a. Resource is available to the user: PDP responds with "yes". Fedora sends back the requested resource, with a HTTP 200 response code. Work is done. b. Resource is not available to the user: PDP responds with "no". Fedora sends back a HTTP 403: Forbidden (final, request should not be re-submitted). At this point, the front-end application decides what to do with the 403: show an error page, mask with a 404, etc.