

Policy-controlled storage

Title	Storage is defined by policy, not by configuration and code
Primary Actor	Repository managers, anyone concerned with archival soundness (auditors, archivists).
Scope	System-wide effects
Level	79F
Story	<p>Currently, storage for repository resources is defined by object code (in the sense in which programmers use that term) and configuration that is normally unconnected with the specifics of object (in the Fedora sense) contents. In order to vary the manner of storage based on the characteristics of a given object, significant amounts of time and expensive technical expertise have to be expended.</p> <p>For example, an archivist or researcher may need the repository to use different types of storage for objects with different expected amounts or types of use, or different legal restrictions on use. Or an auditor may need the repository to use storage with the ability to produce highly detailed reports about use (which must be expensive as compared with less full-featured storage), but only for some kinds of objects. Currently, fulfilling these kinds of needs requires work from skilled programmers familiar with the Spring framework, Akubra, and Fedora. This shouldn't be so hard. An example arises for APTTrust, wherein large content bitstreams must be persisted to cloud storage, but high-value metadata bitstreams must be persisted to local storage. Significant amounts of money and effort will be spent building workflow machinery to accomplish this end, because Fedora cannot elegantly handle asynchronous storage or content with varying needs for storage. This use case addresses the latter lack.</p> <p>For another kind of example, there is currently no way at all to propagate low-level persistence information (e.g. fixity or size) from a storage type that maintains it, into Fedora. An auditor that requires examining that information will have to find some way to get at it "around" Fedora. Of course, Fedora cannot and should not try to offer a complete contract for every type of storage and storage capability that could be integrated with it, but some types of low-level storage information <i>should</i> be visible at the level of repository policy. This use case arises today in the APTTrust context, in which access to low-level fixity information has required that administrative services integrate over two services (Fedora and the storage service) instead of simply managing the repository. A simple contract could be offered to allow information to flow as described, minimizing the need for expensive workflow integration in the future.</p> <p>To support both directions of example, just as access control is now defined by declarative policies (which can be stored in the repository) that can be associated to resources at a number of different levels and in a number of different ways, storage for resources should be allocated by declarative policies (which could be stored in the repository) that could be flexibly deployed. These policies should allow for the declaration of mappings from content to storage facilities (based on attributes of the content and attributes of the storage facilities) as well as mappings from storage capabilities (e.g. fixity) to content attributes (for use with other Fedora services).</p>