# University of Pennsylvania - Expose content through Fedora, offloading serving of binary content to a dedicated service

| | |
|---|---|
| Primary Actor | Storage location/server, Fedora with filesystem federation |
| Scope | System |
| Level | Summary |
| Story | 1. Binary content is served from a separate storage location optimized for serving digital assets (e.g., Apache, S3).<br>2. Fedora, using filesystem metadata provided through filesystem federation, redirects the client to retrieve this content directly from storage, bypassing Fedora for the purpose of serving binary content. |

more information about our specific use case at UPenn, and some general thoughts on the usefulness of filesystem federation **without** Fedora serving binary content, below:

## Filesystem federation deprecation

~~Five~~ Four factors influencing decision to move toward deprecation:

1. Fedora as a RESTful specification does not include idiosyncratic features of the current reference implementation
2. Lack of use cases not addressed by points 2-3 (below)
3. Plausible alternative for known use cases: "message/external-body" URL-accessible content
4. Reason to prefer serving binary content from a server more specifically suited to the task of serving large static binary content (as opposed to having Fedora directly mediate the transfer of static binary content)
5. ~~Modeshape 5 not yet known to be capable of reflecting changes to the filesystem in a running application~~

### Developing use case

At Penn, we are 100% on-board with #4 above. Paradoxically, we have actually been planning to leverage filesystem federation to achieve a nearly identical goal: to redirect/offload the serving of static binary content to a dedicated server.

As is the case with any `FileSystemConnector`, our current implementation uses the filesystem connector to expose filesystem resources on demand. However, our implementation differs from the stock `[Fedora]FileSystemConnector` in that rather than directly serving binary content from the filesystem, we use filesystem metadata to (on demand) resolve/redirect to a URL that can be used to fetch the binary content.

In our use case, we happen to be resolving git-annex-flavored "broken" symlink targets (effectively a content address) into signed/authorized S3 requests against a content-addressable CEPH S3 gateway backend. A similar (but even simpler) implementation could likely prove useful as a drop-in replacement for existing "filesystem federation" capability that offloads the serving of actual content (e.g., to an Apache server over the same filesystem tree exposed through Fedora). This approach would avoid the need to manually maintain/sync corresponding "message/external-body" binary resource nodes (which would otherwise have to be handled explicitly).

From our perspective (and perhaps also that of anyone hoping to leverage Fedora's authn/z for content delivery), one drawback of the static-URL approach currently employed for "message/external-body" binary resources is that request-signing of the download URL is not currently supported.

Regarding reason 5 for deprecation: it seems that the filesystem caching behavior observed in Modeshape 5 may not be a bug, but rather a configuration change to agnosticize references to caching functionality implemented by newly-deprecated Infinispan. See MODE-2529; is it possible that simply removing the old `repository.json` config parameter `cacheTtlSeconds` and setting the new boolean config parameter `cacheable` to `true` would restore the desired functionality in Modeshape 5?

### Higher-level considerations

In a sense, it might be helpful to consider the `[Fedora]FileSystemConnector` (and the Modeshape `Connectors` more generally) not chiefly as tools for the transparent inclusion of content to be mediated by Fedora/Modeshape. Perhaps we could instead view the `Connectors` as a compatibility layer that allows Fedora/Modshape to dynamically include references to nodes as defined and administered in other system ... whether Fedora/Modeshape serve the content represented by those nodes, or offload that to another service could be considered as a separate question.

A step toward supporting this more general-purpose interpretation of a `FileSystemConnector` is slated for release with Modeshape 5.1.