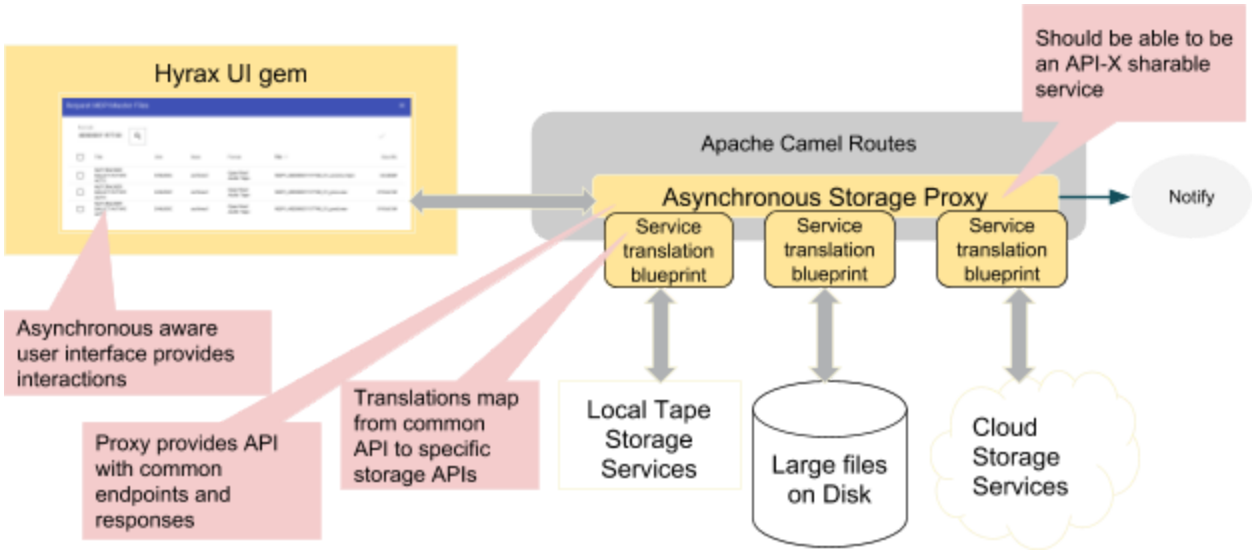
The API Extension Architecture (API-X) provides a framework for extending the native functionality of a Fedora 4 repository. The external file storage proxy is based on API-X so that it will easily extend Fedora 4 functionality to allow files to be stored on an external, asynchronous storage systerm, such as tape storage or other cloud based storage, such as S3 cloud service.

As part of the grant for PHYDO (formerly Samvera (Hydra)DAM2), WGBH-Boston and Indiana University held a three day Developers' Congress on the External File Storage Proxy involving invited members of the Samvera and Fedora communities. WGBH and IU had already developed an architecture to support asynchronous storage with IU's tape based system, the Scholarly Data Archive (SDA) and WGBH's storage of files on LTO tape. But we wanted to extend this design to encompass the needs of the Fedora and Samvera communities and use cases involving cloud storage. This architecture uses the implementation of Camel in Fedora 4 to provide a route between the tape storage, Fedora 4, and the Samvera application, PHYDO. In addition, these objects are stored in Fedora 4 as non-RDF resources with a redirect URL so that the object determines which Camel route to use to download the file associated with the object. For this developers' congress, we wanted to extend this architecture to include other external storage such as S3, Glacier or other cloud services. For the congress, we focused on the S3 cloud service.

In the weeks leading up to the congress and over the course of the three days of the meeting, we focused on two major areas. One was defining and developing an API between the Hyrax user interface and the external storage proxy. By the end of the three days, the user interface was able to send a JSON file as a request to the external storage proxy and receive a JSON file in return with status information for the file requested. The second area we focused on was defining and implementing the exact structure of the external storage proxy. In our original design, we had not focused enough on tracking the status of a job submitted to an external storage service. The new design used a database to operate more or less as a jobs queue to keep track of requests for files from the external storage proxy. That way, we would not need to process an additional request for a file if one was already pending.

The design for the user interface API is complete and integrated with PHYDO. The design and implementation in Camel of the File Storage Proxy is complete as well and integrated into PHYDO. Users of PHYDO will be able asynchronously download files from either tape or cloud storage. In fact, if you wanted to have all files be submitted through the external file proxy, in order to have a standard interface for downloading all files, that would be possible as well.



Evaluation section

Challenges encountered. In terms of weaknesses and strengths. Evaluation of outcomes compared to goals; link to accomplishments. Evaluation of the process and the project.