# APTrust and Hydra

A good fit for a lightweight application

# Preserving the Scholarly and Cultural Record

What We Do

Latest News MO

CHIP GERMAN NAMED APTRUST PROGRAM DIRECTOR

**APTRUST.ORG** 

Search this site



#### **▼ Home**

- ▶ APTrust Roadmap
- Content Advisory Group
   Getting Started for Testers
- ▶ Tech Advisory Group

#### Files

October Partner Meeting Agenda

**Project Definition** 

- Technical Documentation
  - Architecture
     Coding Best Practices
     Deletions
  - DPN First Node
     Git Branching and Tagging
     Preservation & Storage
  - Reporting & Auditing Restoration
  - Submission & Ingest

#### Sitemap

Recent site activity

#### APTrust Resources Content Advisory Group

Tech Advisory Group APTrust Public Website Presentation List PivotalTracker GitHub Oranization APTrust Operations Wiki APTrust YouTube

#### Recent site activity

October Partner Meeting Agenda edited by Bradley Daigle

Content Advisory Minutes

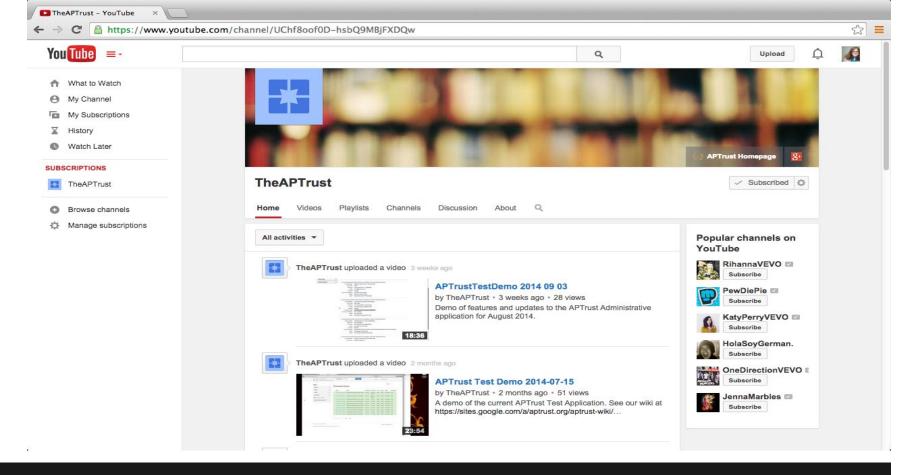
#### Home

This wiki contains details of the Services, Specifications and Implementation of the APTrust Aggregate Preservation Repository. See the appropriate section for more detail documentation or use the search feature as needed. As with any wiki this documentation may change over time so check back for additions or changes as they occur. For general information about APTrust, please visit aptrust.org.

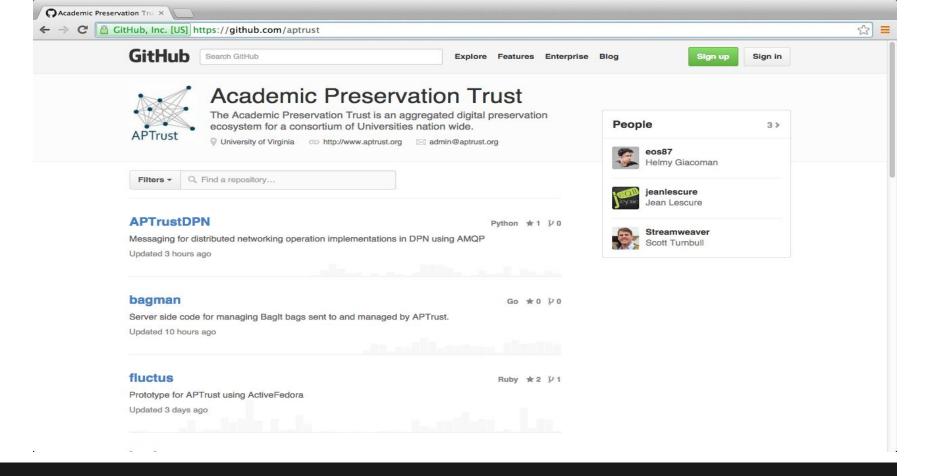
Questions or Comments? Please send email to info@aptrust.org

APTrust Current Development Plan												
Content Processing & Ingestion	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Se
Implement a durapable work queue for content processing							S	X				
Service to watch for incoming bags and initiate ingest							SX					
Validate Bags and Read Submission Metadata							SX					
Generate Checksums and Gather Technical Metadata							S	X				
Move Files to Preservation Storage									S	X		
Register bag, file metadata and audit data in Fedora					94 - E				S	х		
Register outcome with processing results list									S	X		
Mark review processed items/Purge failed items.									S	X		
Improve Ingest Failure Handling for Large Items										S	X	
Ongoing Fixity & Audit	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Se
Impelment service to queue oldest objects first for fixity												*
Implement object copy and checksum												*
Register outcome of fixity check with Fedora												*
Restoration	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Se
Queue requests for restoration											SX	
Implement scripts to create and validate bags											SX	
Move hads to restoration buckets and notify requestor					7						SY	

# **AP Trust wiki**



# **APTrust on YouTube**



# **APTrust on Github - github.com/aptrust**

### Based on notes from Scott Turnbull, APTrust Technical lead:

- With Fedora as a requirement, Hydra was very useful for rapidly building a front end with one Jr Engineer as the primary developer.
- Additionally the myriad of examples from other projects and community out on the Hydra lists were good resources.
- APTrust started developing after the availability of RDF datastreams in Hydra which was
  particularly useful. APTrust needs metadata to be portable between systems like DPN and it
  seems better to rely on semantic assertions rather than structured XML to package content for
  distribution between repositories with the intention of storing them on disk.
- APTrust's use of Hydra is a critical part of a strategy for upgrading to Fedora 4. APTrust is purposefully avoiding any unneeded complexity or uniqueness in the deployment of Hydra and will use the momentum of the Hydra community for the transition to Fedora 4
- The ability to leverage the rails framework and mix Hydra features with rails features. As APTrust is processing large batches of material the system is necessarily asynchronous and has latency depending on the load on the system and size of content being processed. This allows APTrust to manage workflows for system processing using simple ActiveRecord objects in rails. It also allows us to leverage known rails gems for things like user authentication (Devise) or user authorization (CanCan/Pundit).

# **Benefits of Hydra for APTrust**

#### Some Differences that may be of interest:

- Binary files are stored and managed externally from Fedora in S3. Externally Referenced Content (Pointer) datastreams point to binary files, which are actually stored in cloud storage.
- APTrust began developing its Hydra head using CanCan for authorization and user groups, which
  has been deprecated since last year. VirginiaTech partnered with APTrust and refactored CanCan
  authorization to use Pundit instead, with a much more maintainable set of code and permissions.
- In APTrust's particular use case Fedora is not the primary content manager but is primarily used for metadata storage and management, with Hydra for the reporting interface. APTrust currently uses S3 and Glacier for primary preservation storage and for receiving and restoration services and content transfer. Content processing, including ongoing checkum and fixity is managed by server side processes built using the Go language. The different components of the system ultimately talk to Fedora through REST calls implemented in the Hydra and rails interface, allowing a high degree of flexibility in running running multiple servers and services if need be.

## Some differences

#### Some downsides:

- Hydra rights being separate from the Fedora rights and authorization is unpalatable particularly since the Hydra app needs administrative access to Fedora by default. This has been lamented by the community at large at times.
- The difficulties of keeping gems updated, particularly when some go in and out of favor in the Rails community can be disruptive. Additionally the differences in interfaces between some minor updates in either ruby, rails or individual gems can be unpredictably disruptive.
- Performance has been a problem even with only a few 10s of thousands of items. This seems
  due to a combination of Fedora itself, how related objects are loaded through Fedora and in
  Hydra, and how Indexing is done in Blacklight. APTrust has done some work in improving
  performance through better implementations but further improvements are not apparent and
  the overall bottleneck isn't in transfer, or processing, but in the response time of Hydra and
  Fedora.

# Some downsides of Hydra

Scott Turnbull
APTrust Technical Lead
University of Virginia
scott.turnbull@aptrust.org
www.aptrust.org
678-379-9488

Lightning talk at Hydra Connect #2
presented by
Linda Newman
Head, Digital Collections &
Repositories
University of Cincinnati Libraries
linda.newman@uc.edu
https://scholar.uc.edu