Brightening an Archive

Streamlining access to OA datasets

Eric Lopatin, Terry Brady – April 2021

UC3
UC Curation Center

University of California
CDL
California Digital Library

Merritt

CORE TRUST SEAL

Merritt is a CoreTrustSeal certified, open-source digital preservation system maintained by UC3 team at CDL.

- Three independent copies, three different cloud-based storage providers, across two geographic locations with differing disaster threats.

- Microservices for ingest, storage, inventory, audit, and replication.

Our users

Merritt serves librarians and researchers, as well as systems internal and external to the University of California.

- Ten UC campus libraries and affiliated organizations
- eScholarship integration: 85 open access journals
- Dryad OA datasets

Content management in Merritt

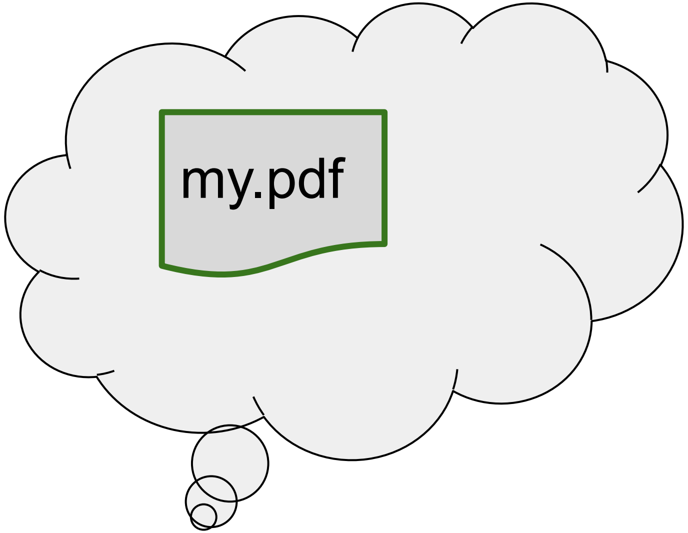
Merritt *individually* manages files and objects.

- A file being a single digital file
- An object can contain one or more digital files + metadata

“Files” in Merritt are stored in cloud storage

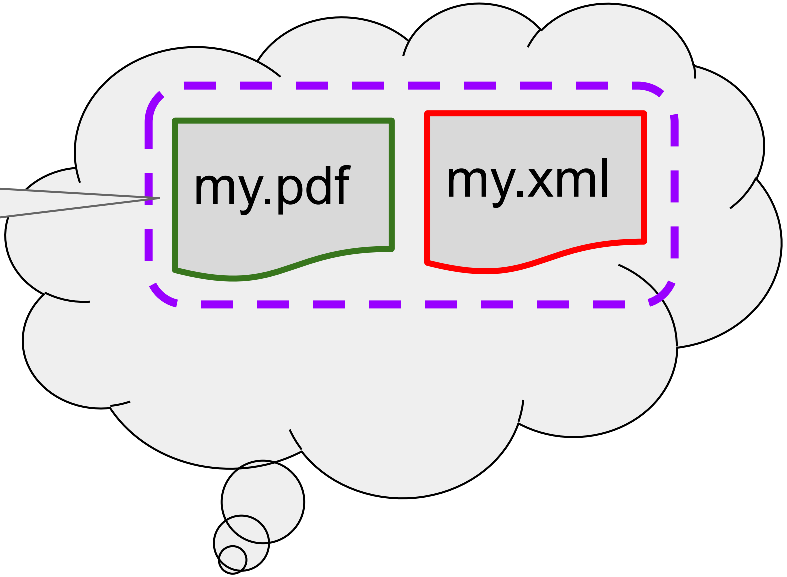
Management occurs over multiple cloud storage providers

- Amazon S3 & Glacier
- SDSC Qumulo
- Wasabi



UC3
UC Curation Center

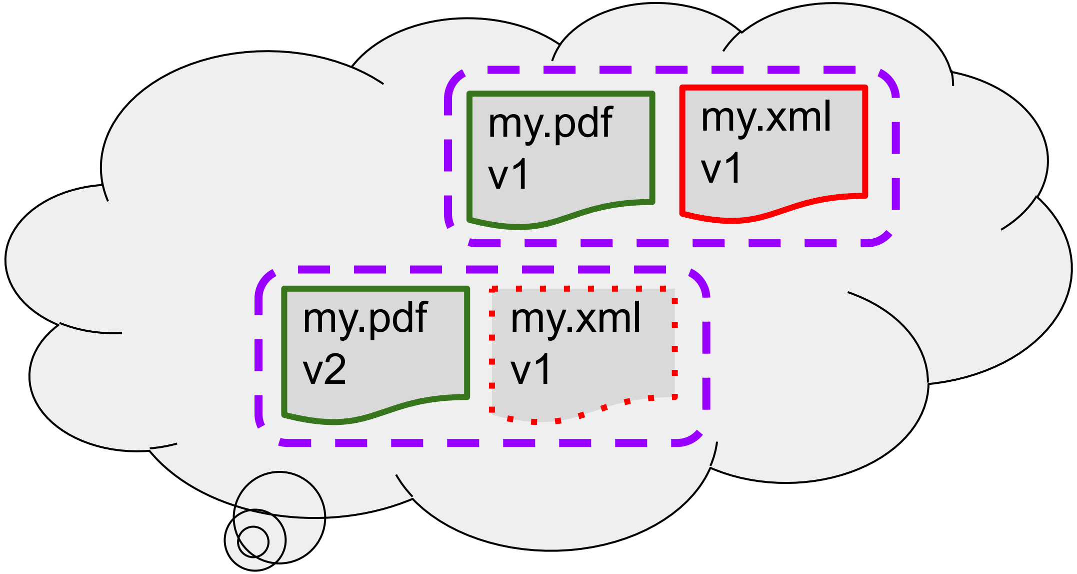
“Objects” in Merritt have digital files and metadata



Objects in Merritt are referenced with an ARK

UC3
UC Curation Center

Objects in Merritt can have “Versions”



UC3
UC Curation Center

Range of file sizes across Merritt

Max file size	Min file size	Average	Producer files
289.8 GB	0	8.5 GB	20,102,755 †

Under 1MB	1MB - 10MB	10MB - 100MB	100MB - 1.0GB	1GB - 10GB	10GB - 100GB	> 100GB
17,830,145	671,802	1,475,116	112,584	11,782	1,231	95

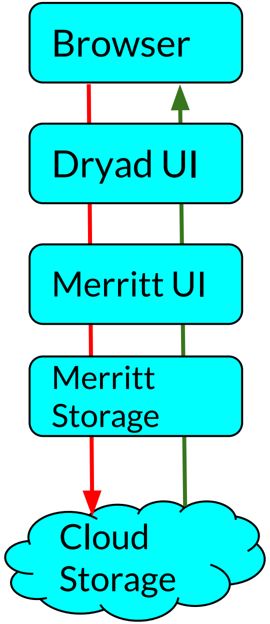
† Many more system files for restructuring the database in the event of a catastrophic failure.

Dryad integration

Dryad contains 34K CC0 datasets from approximately 2000 institutions, and 100K researchers.

- Every new dataset flows from Dryad into Merritt.
- Datasets are replicated to Zenodo
- Dryad generates the majority of access requests (for individual files and entire objects)

http://datadryad.org

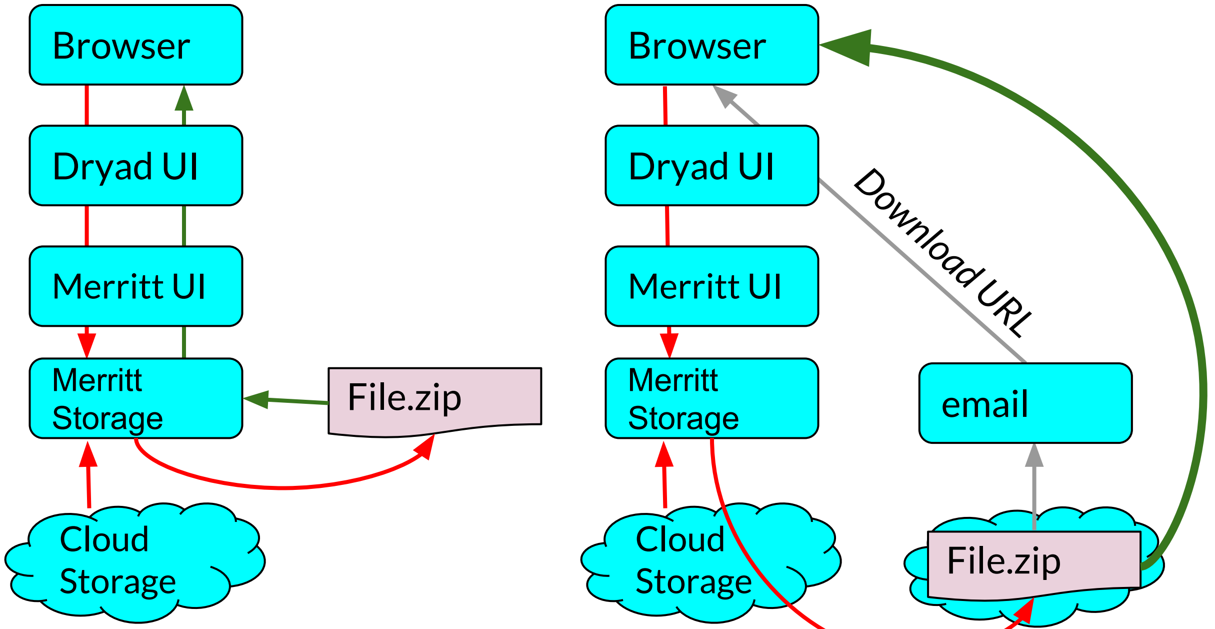


Connecting Merritt & Dryad

Initial integration of the systems

- Downloads of both files and objects were enabled during Merritt/Dryad integration.
- However the initial implementation provided an inefficient means for doing so.

Object Requests (legacy)



Normal Size

Large Object - Asynchronous

UC3
UC Curation Center

Connecting Merritt & Dryad

Major hurdles to address

- Long-running operations
  - Ingest
  - Download
- Multiple simultaneous downloads occurring through the Dryad frontend.
- Each byte of digital content was streamed through multiple applications.

Long-running operations

- Exceptionally large files: less than .05% to .5% of content
- Could be ingested or retrieved at any time
- Subject to timeout errors on upload and download
- AWS Load balancers timeout at ~1 hour
- We have configured Apache load balancers to permit sessions to last 24 hours
- Waiting for a session to terminate can complicate the timing of a software deployment

Connecting Merritt & Dryad

Additional hurdles to address

- Large dataset objects were delivered asynchronously, with an email notification to the user.
- Depending on the domain, emails could be blocked.

Re-engineering Access

Approach

- Needed a dependable way to stream content directly from the cloud.
  - S3 compatible method that would work across all of our service providers
- Settled on use of **presigned URLs.**

Presigned URL Example: Public

```
$ cat > hello.txt
Hello There

$ aws s3 cp hello.txt s3://terrywbrady-test-ucop-public/ --acl
public-read
upload: ./hello.txt to s3://terrywbrady-test-ucop-public/hello.txt

$ curl
https://terrywbrady-test-ucop-public.s3-us-west-2.amazonaws.
com/hello.txt
Hello There
```

Presigned URL Example – Non-public S3 Object

```
$ aws s3 cp hello.txt
s3://terrywbrady-test-ucop-public/nonpublic.txt
upload: ./hello.txt to
s3://terrywbrady-test-ucop-public/nonpublic.txt

$ curl
https://terrywbrady-test-ucop-public.s3-us-west-2.amazonaws.com/
nonpublic.txt
<Error><Code>AccessDenied</Code><Message>Access
Denied</Message>...
```

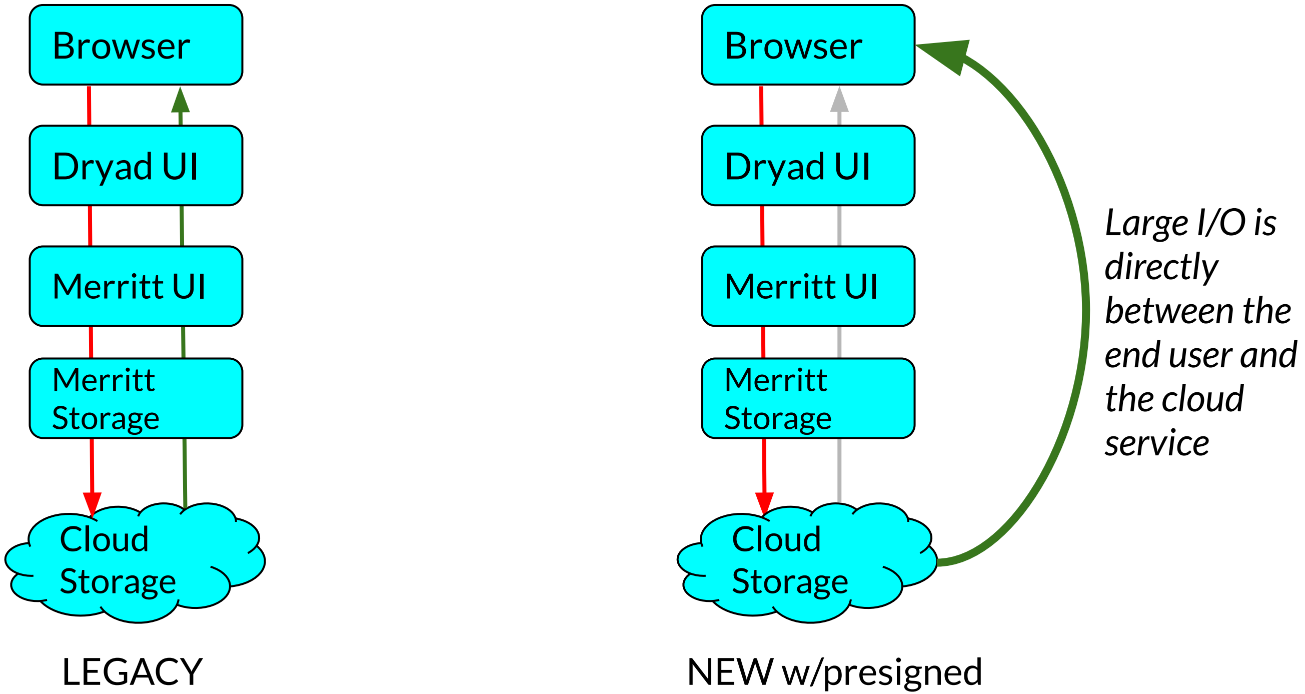
Presigned URL Example – Non-public S3 Object

```
$ aws s3 presign s3://terrywbrady-test-ucop-public/nonpublic.txt

https://terrywbrady-test-ucop-public.s3.amazonaws.com/nonpublic.txt?AWSAccessKeyId=AKIAYOGNF7UOV5VSVMJR&Expires=1591999565&Signature=zppZEk83FBk6UtlNTqmD5ZR3EOA%3D

$ curl "https://terrywbrady-test-ucop-public.s3.amazonaws.com/nonpublic.txt?AWSAccessKeyId=AKIAYOGNF7UOV5VSVMJR&Expires=1591999565&Signature=zppZEk83FBk6UtlNTqmD5ZR3EOA%3D"
    Hello There
```

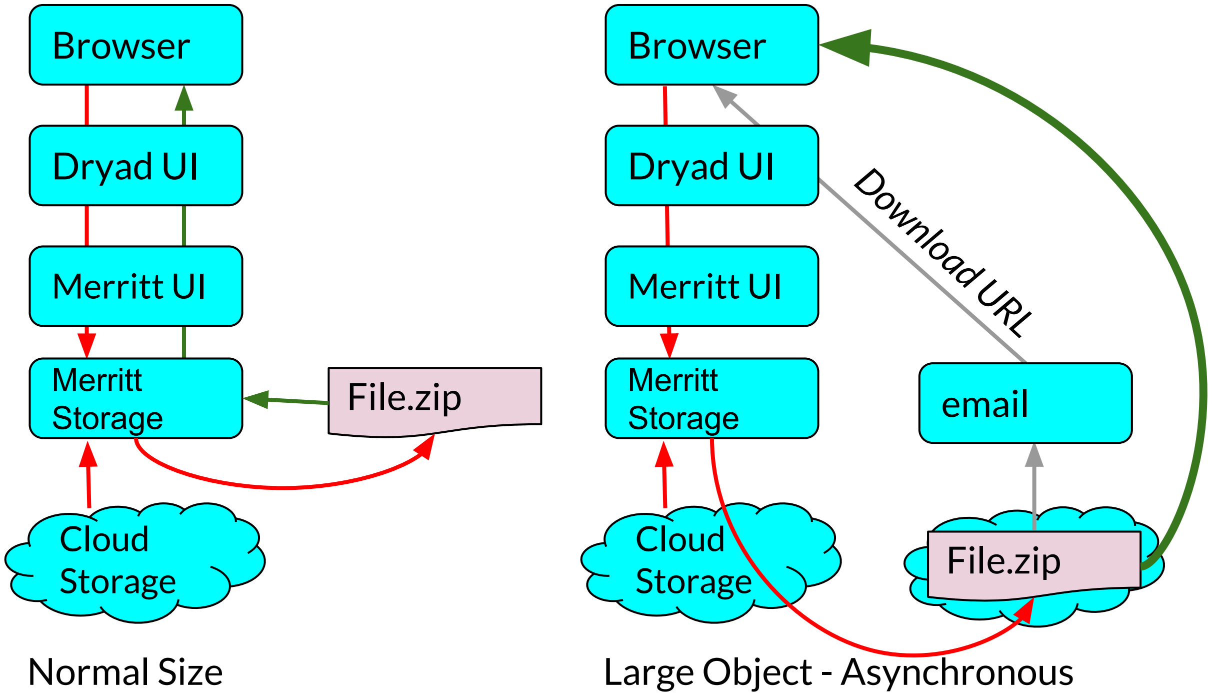
File Requests



Large I/O is directly between the end user and the cloud service

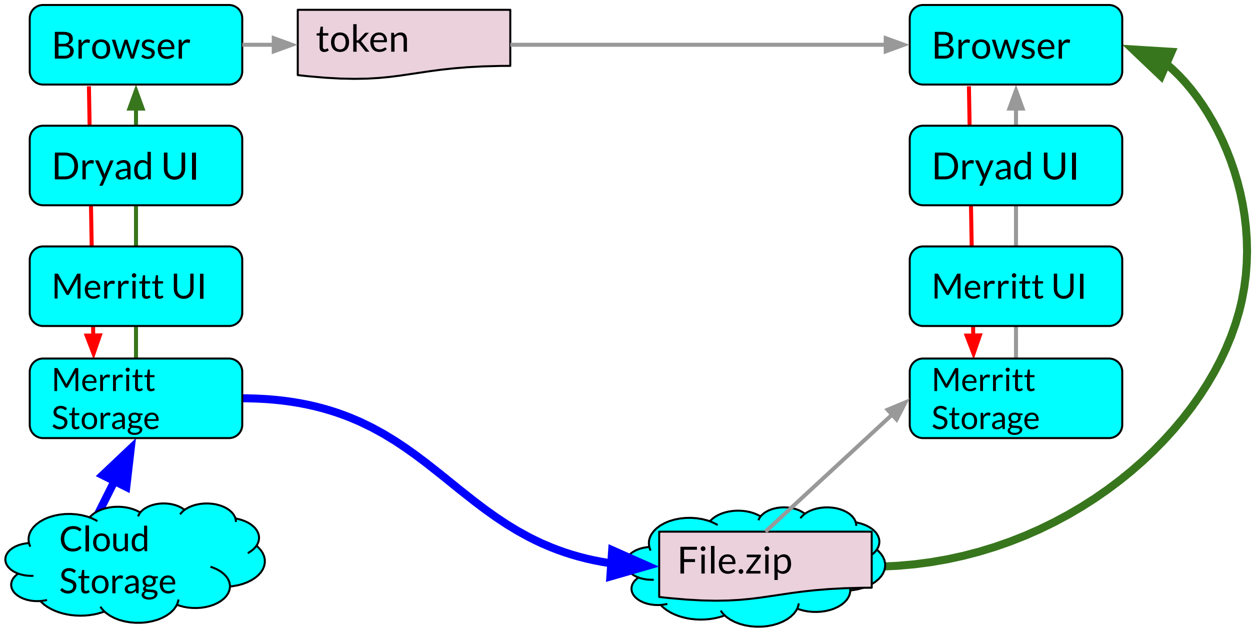
UC3
UC Curation Center

Object Requests (legacy)



UC3
UC Curation Center

Object Requests (presigned)



All object requests are asynchronous using presigned URLs

UC3
UC Curation Center

AWS usage change

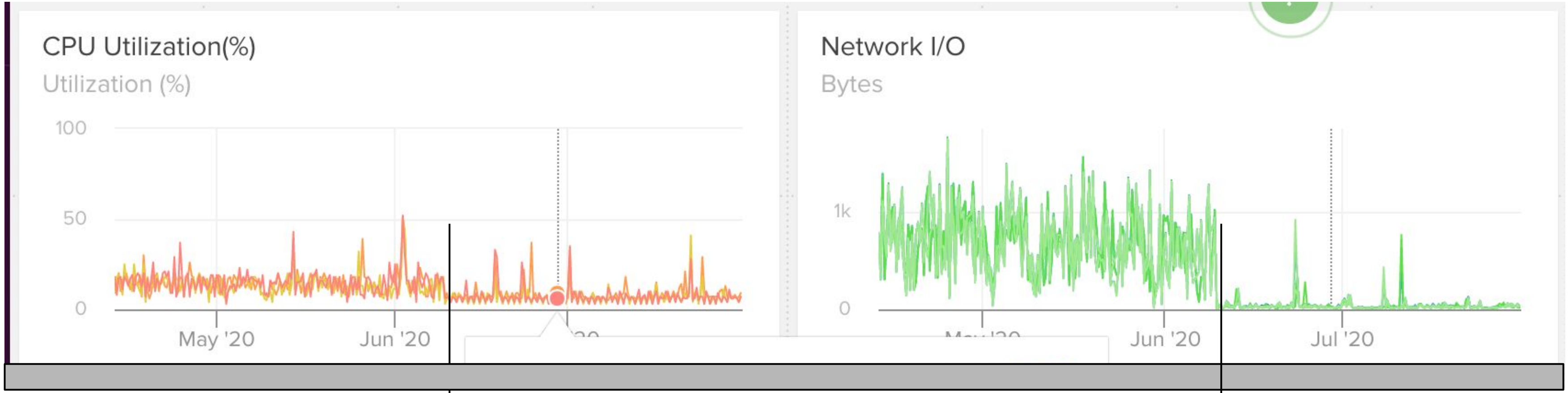
Before

- CPU utilization and Network I/O on Merritt UI hosts was consistently high due to the streaming content back to Dryad.

After

- Network I/O transactions were greatly reduced. CPU utilization also lessened.

Server metrics and the presigned effect



Before | After

Before | After

UC3
UC Curation Center

Going forward – What about user uploads?

Dryad Presigned Uploads – Went live in March

- AJAX call to authorize an S3 upload and obtain a secure, presigned URL
- Evaporate.js is used to chunk user files into parts
- Upon upload completion, all parts are reassembled via an AWS command
- A manifest is delivered to Merritt for ingest

Dryad deposits in Zenodo

Every dataset is subsequently deposited in Zenodo

- Presigned URLs are used to first download content from S3 – it is then staged for streaming up to Zenodo.
- If a dataset contains software, Dryad now supports sending software files directly to Zenodo.
- All Zenodo deposits are tracked via DOI.

Future plans for Merritt

- Learning from Dryad team’s experience with uploads.
- Presigned upload implementation in Ruby and Java
- Recording Zenodo DOIs with associated Merritt objects

Brightening an Archive

The goal of refined integration with Dryad drove us to make these improvements.

- Resulting API endpoints can be used to obtain *any* file or object in Merritt.
- Serve as building blocks for campuses and institutions, enabling the construction of access layers directly on top of a preservation repository.

Demo

- Let’s retrieve a Dryad dataset object.
- Review an example of a dataset with software.

Questions?

DRYAD

Merritt