

Multiple Storage Backend Design

The goal here is to support multiple simultaneous storage backends.

There are several ways to accomplish this, along with varying levels of flexibility with each option.

- [Option A: Federating multiple instances of Clowder](#)
- [Option B: Configure all possible storage, let user choose when uploading](#)
- [UI Mockups](#)
 - [1. Single Dropdown](#)
 - [2. Multiple Dropdowns](#)
 - [3. Vertical Drag + Drop Groups](#)

Option A: Federating multiple instances of Clowder

In this case, for every unique storage backend, we require a small Clowder instance sitting in front of it to access the file bytes.

Each Clowder instance talks to a single storage backend, which houses the files uploaded to it.

From any other instance in the federation, we could offer the user an option to upload to any other instance that is also part of the federation.

Perhaps one Clowder instance works as the "master" for this federation, or maybe all Clowder instances join up with each other as peers.

This case likely has the most unanswered questions, as it is likely that a discussion regarding "federating multiple instances of Clowder" would mean something different to every project.

Open Questions:

- Should we display files from federated instances across instances bounds? Are there concerns with sharing at this level?
- Can we guarantee that configuration between instances is synchronized? What if one instance is private and another public - is federating prohibited in this case?
- Can we guarantee that software version between instances are synchronized? What if one instance is running 1.5.2 and 1.6.1 - is federating prohibited in this case?
- What do we do if an instance in the federation becomes unavailable or unstable? Can we mitigate instability between instances?
- Are we only federating files? Wouldn't we also need to share datasets/spaces/collections/folders to preserve the hierarchies between files?
- Could we add a sort of "virtual collection" in Clowder that works as a symbolic link to another Clowder instance?
 - Reasoning: this would allow us to easily list the Datasets/Files from another Clowder instance with minor modifications to the UI

Option B: Configure all possible storage, let user choose when uploading

We could choose to limit the scope of this task enough to make it achievable very quickly while we work to answer the larger questions about federation.

This would be a completely arbitrary decision, but choosing to ignore MongoDB / Disk options, and focus solely on expanding the S3ByteStorageService to allow the user to configure multiple buckets.

NOTE: We would need to separately add (if we decide its worth adding) multi-storage support for MongoDB / Disk later in this case.

Another option might be to configure all possible storage locations (MongoDB + Disk + S3) in Clowder and allow the user to choose where their files will be stored upon upload.

UI could default to whichever default option was chosen by the administrator - this could easily be determined by the order of storage backends in the config, or by adding another (more explicit) config option.

This seems to be the most flexible, since we could configure (for example) the buckets, disk storage, and a mongo collection.

Open Questions:

- Can Clowder/Play configuration support an array of objects? I know they support maps, but have yet to see an array of objects
- Is there any concern with only uploading particular files to particular places?
- Can we somehow limit which users can upload where using permissions?

UI Mockups

Open Questions:

- Would allowing the user to choose a default location per upload-set suffice? Do we really need to allow overriding this destination at the individual level?
- Should we allow users to override the admin's default selection for the storage backend at the dataset/space level?

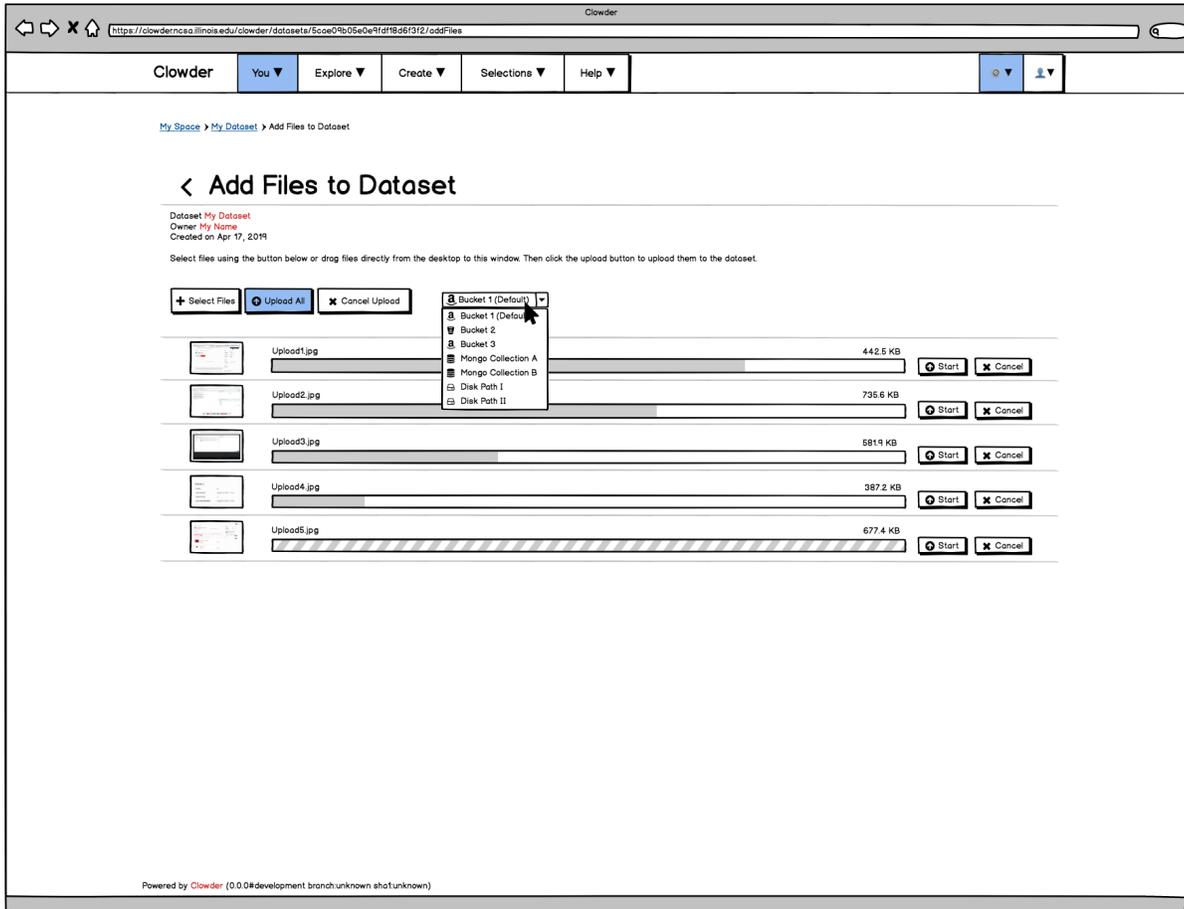
Possibilities:

1. Add a single dropdown at the top of the page to choose destination for all uploads?

2. Add a dropdown for each item? This seems tedious and not super user-friendly if (for example) you want to upload 50 files to somewhere other than the default.
3. Show a vertical group for each possible storage backend, and allow dragging between groups?

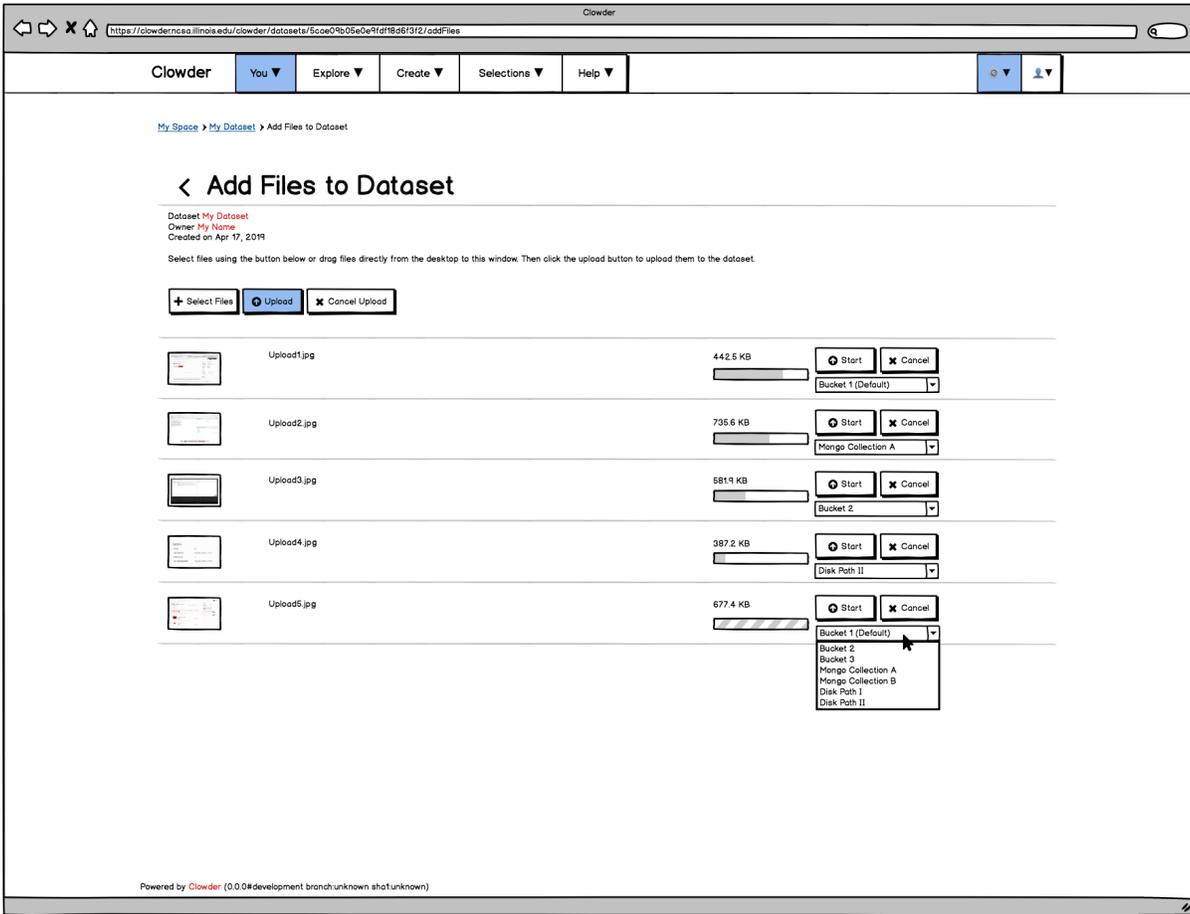
1. Single Dropdown

The easiest solution would be to offer the user a single dropdown at the top of the page to select the destination for all of their uploads:

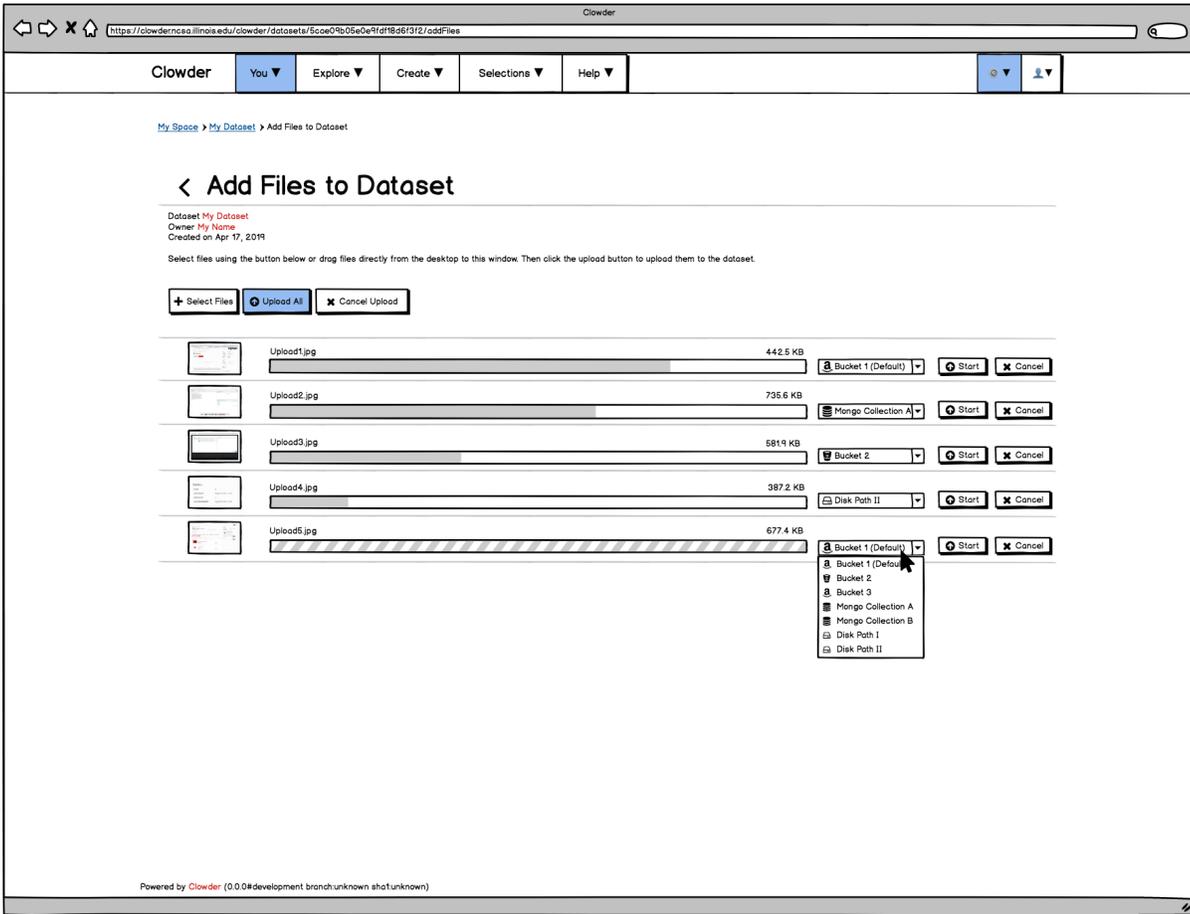


2. Multiple Dropdowns

Making only minor adjustments to the page, we could also add a dropdown beside each pending upload:



Alternatively, we could restyle a bit more of the page to produce something like this:



3. Vertical Drag + Drop Groups

By adjusting the upload view to show a vertical group for each possible destination, the user could drag + drop their staged uploads to the destination of their choice before clicking "Upload":

Clowder

My Space > My Dataset > Add Files to Dataset

< Add Files to Dataset

Dataset: My Dataset
 Owner: My Name
 Created on Apr 17, 2019

Select files using the button below or drag files directly from the desktop to this window. Then click the upload button to upload them to the dataset.

+ Select Files Upload x Cancel Upload

Bucket 1 (Default)

Upload1.jpg 442.5 KB Start Cancel

Upload5.jpg 677.4 KB Start Cancel

Bucket 2

Upload3.jpg 581.9 KB Start Cancel

Mongo Collection A

Upload2.jpg 735.6 KB Start Cancel

Disk Path II

Upload4.jpg 387.2 KB Start Cancel

Bucket 3 Drag files here to change upload destination

Mongo Collection B Drag files here to change upload destination

Disk Path I Drag files here to change upload destination

Powered by Clowder (0.0.0#development branch:unknown sha1:unknown)