

Data Integrity Management with Globus & BDBags



Rick Wagner

rick@globus.org

rpwagner@uchicago.edu

rwagner@anl.gov





Today's Topics

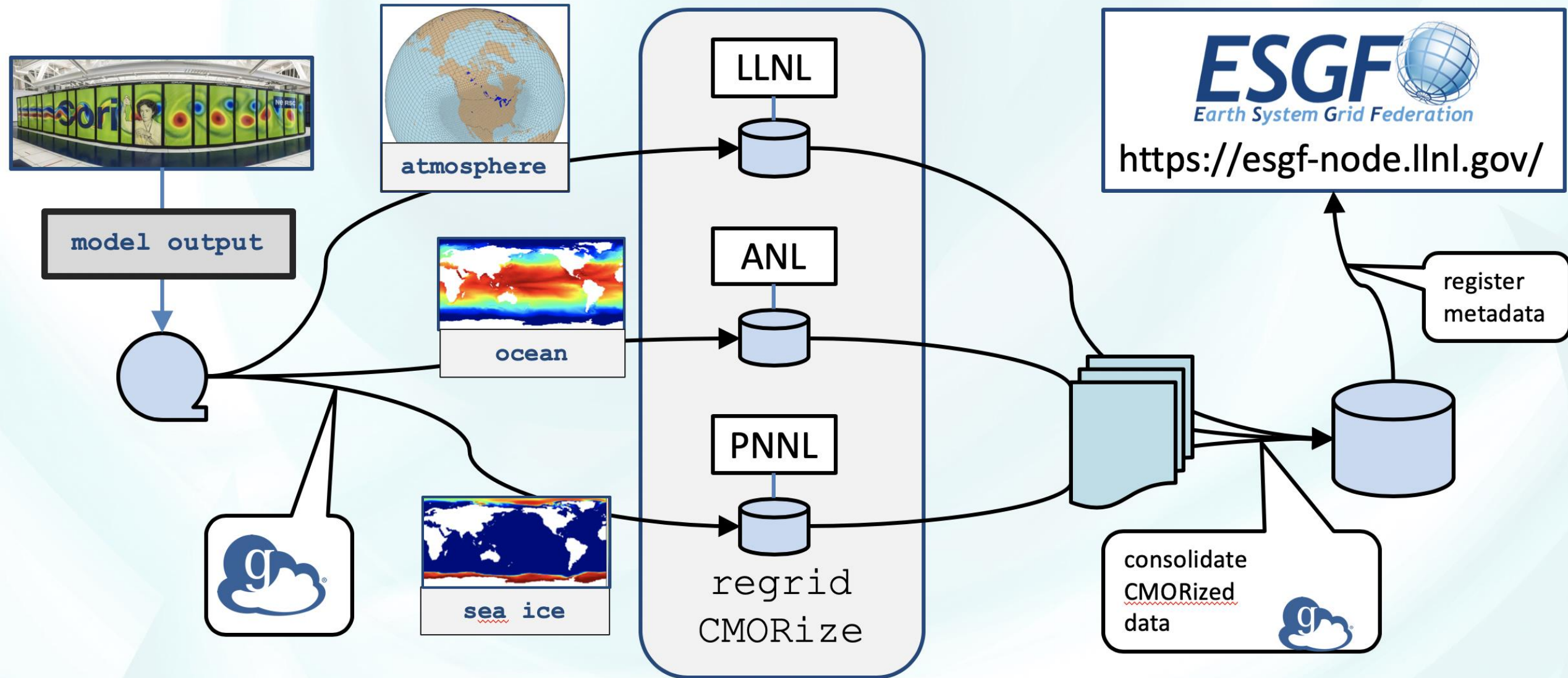
- **Globus**
 - Getting the data between systems reliably
- **BDBag**
 - Ensuring data integrity as it moves around



Example: E3SM to CMIP6 Process

- Large Earth system models created on DOE leadership systems
- Archived on tape at NERSC
- Pulled from tape, distributed for re-gridding, convert units...
- Aggregated for publication
- We have been bitten by **silent data corruption**
- Yes, file systems eat data
- So do networks
- And all of our file systems are built on networks

E3SM to CMIP6 pipeline





Globus Value Proposition

Fast and reliable **data transfer**,
sharing, and platform services...

...directly from your own **storage
systems**...

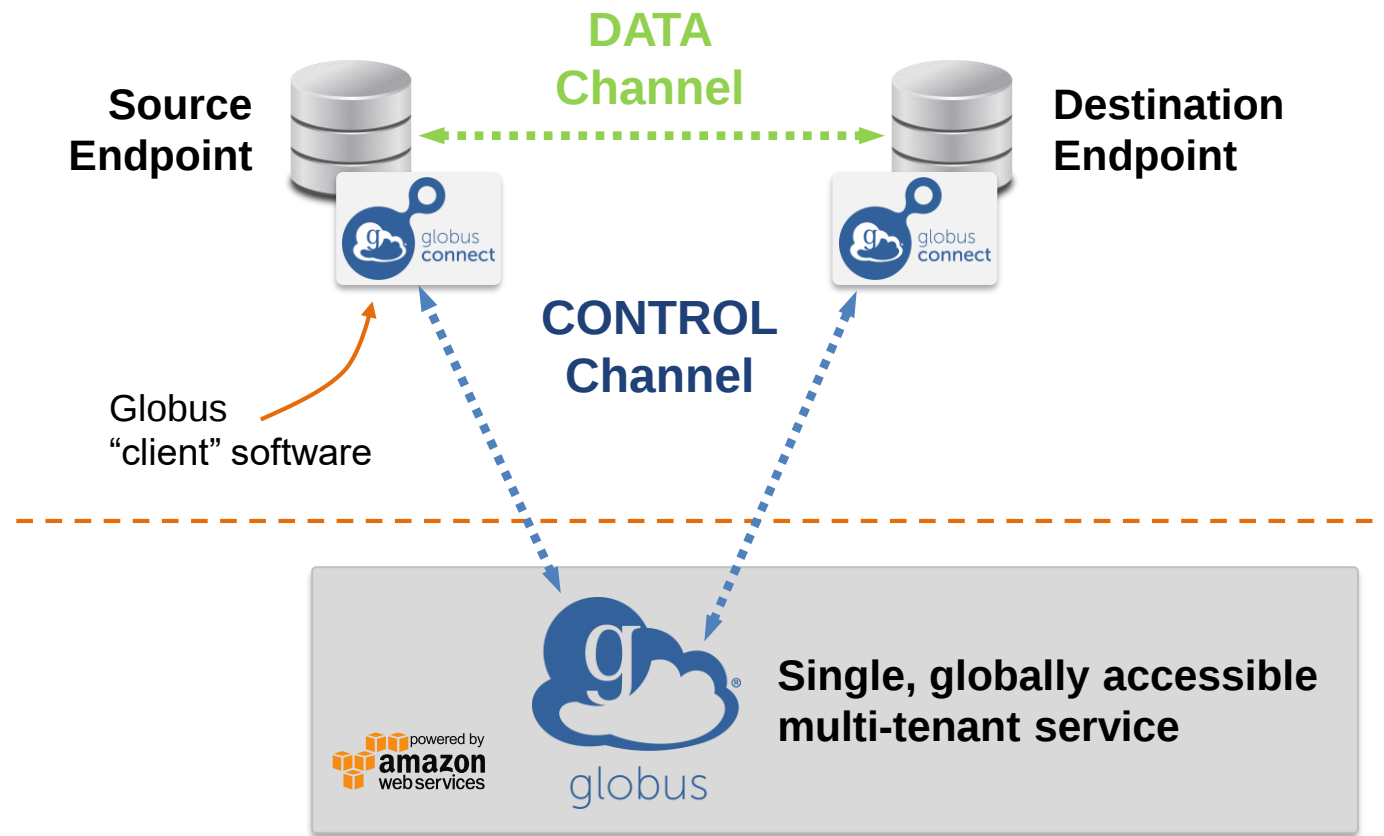
...via software-as-a-service using
existing identities with the overarching
goal of...



Our Bread & Butter: Reliably Copying Files

Reliability:

- Get the same data from source to destination
- Deal with storage & network failures
- Retry until copied
- Verify the data after





verify file integrity after transfer ⓘ

After transfer, verify that the source and destination file checksums match. If they don't, retransfer the entire file and keep trying until it succeeds.

conda.sif02/25/2019 05:05

e3sm.def02/26/2020 06:57

e3sm.sif02/26/2020 06:57

Start ▶

Transfer & Sync Options ⌵

◀ Start

Label This Transfer

Transfer Settings

- ☐ sync - only transfer new or changed files ⓘ
- ☐ delete files on destination that do not exist on source ⓘ
- ☐ preserve source file modification times ⓘ
- ☒ verify file integrity after transfer ⓘ

After transfer, verify that the source and destination file checksums match. If they don't, retransfer the entire file and keep trying until it succeeds.

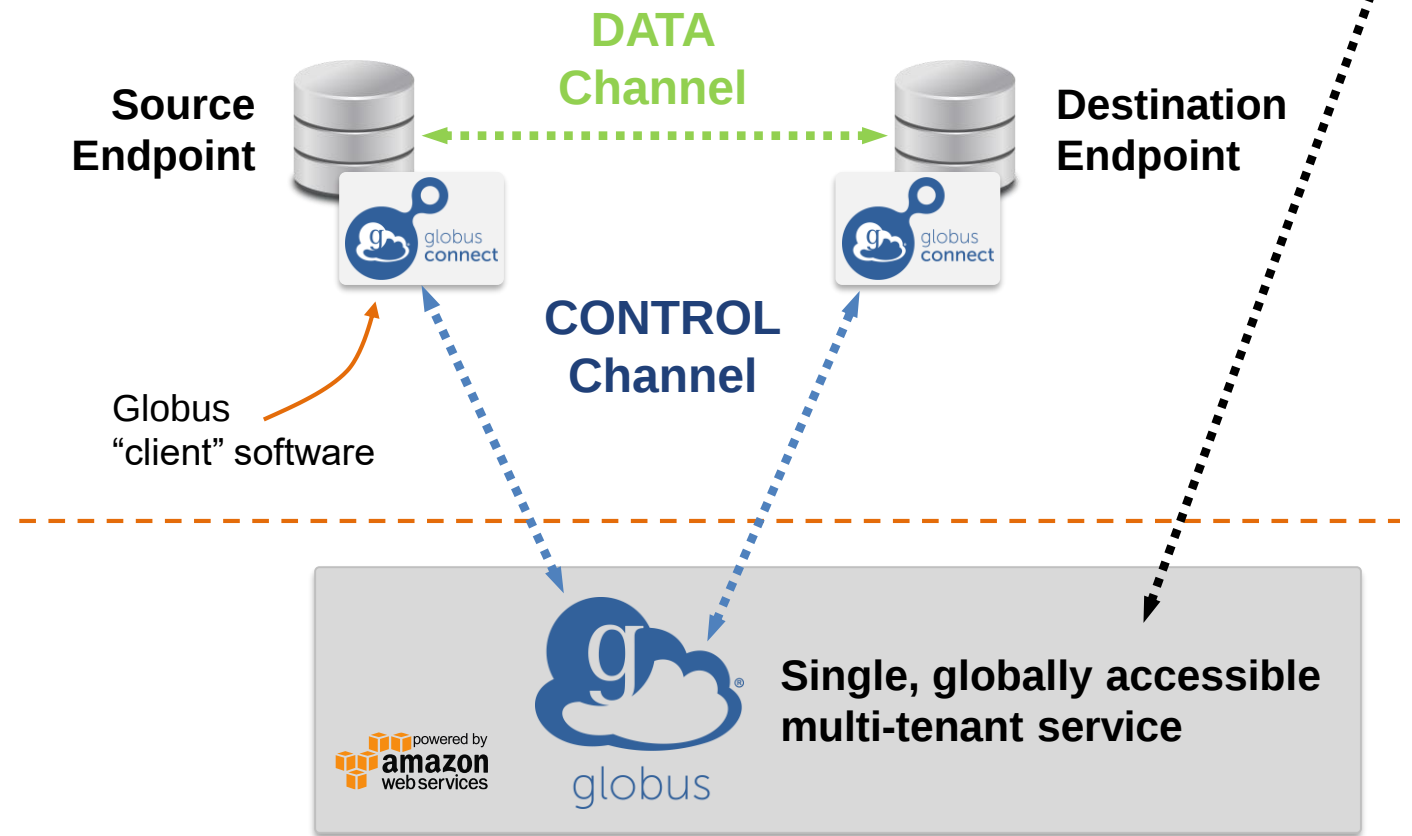


BYO Checksum

External Checksum:

- User provided hash
- Source no longer must be point of truth
- Currently in API
- CLI next

```
...  
"destination_path": "/Users/rpwagner/tmp/e3sm.sif",  
"external_checksum": "542943b76193e8aae68b23db8c5969b2",  
"checksum_algorithm": "MD5"  
}
```





Example: E3SM Data Stager

e3sm_to_cmip

The `data_stager.py` script stages files from a zstash archive that match a list of components or file patterns, and transfers the staged in files to a Globus endpoint for further processing.

The script depends on [zstash](#), [Globus SDK](#), and [FAIR Native Login](#) Python packages. For example, to set up an environment on NERSC/Cori, you can run:

```
cori01:~> module load python/2.7-anaconda-4.4
cori01:~> conda create -n data_stager_env -c e3sm -c conda-forge zstash=0.3.0 fair-research-login=0.1.5
cori01:~> conda activate data_stager_env
```

All messages are written to a logger. By default, the logging level is set to `WARNING`, and can be changed by setting the `LOGLEVEL` environment variable.

```
usage: data_stager.py [-h] [-l] [-d DESTINATION] [-s SOURCE] [-b] [-t T]
                    [-m M] [-z ZSTASH] [-c COMPONENT] [-f PATTERN_FILE] [-e]
                    [-w WORKERS]
                    [files [files ...]]
```

https://github.com/E3SM-Project/e3sm_to_cmip/blob/master/scripts/data_stager.py



Example: E3SM Data Stager

- Use E3SM archive tool (`zstash`)
- Pull files & checksums from HPSS
- Generate file manifest with checksum
- Submit transfer of data files & manifest
- Globus verifies files after transfer
- Manifest can be used to verify files later



Speaking of Manifests: **BDBag**



A packaging format for encapsulating
(The BagIt File Packaging Format (V1.0) draft-kunze-bagit-16)

```
Bio_data_bag/  
|-- data  
|  \-- genomic  
|    \-- 2a673.fastq  
|  -- manifest-md5.txt  
|    afbfa231324812378123bfa data/genomic/2a673.fasta  
|  -- bagit.txt  
|    Contact-Name: John Smith
```



- A specification based on **BagIt**
- Manifest lists all content and checksums
- Available content contained in **data** directory
- Content may be “missing”
 - Missing content must be listed in **fetch.txt**
 - Fetch entries list local name in data directory, and URL or digital identifier of where to fetch data
- Have created tool for creating, validating and materializing bags



Look in a BDBag

```
$ unzip my-bdbag.zip
$ cd my-bdbag/
$ ls -l
bag-info.txt
bagit.txt
fetch.txt
manifest-md5.txt
manifest-sha256.txt
tagmanifest-md5.txt
tagmanifest-sha256.txt
data/
```

```
$ head -5 manifest-md5.txt
8e83c43203830fda100c6c740957527a data/training/cosmo-b10206348_dim512_full.npy
b6e462938e6e8d85ee9f7ab8cbe7e913 data/training/cosmo-b10667031_dim512_full.npy
83cba9deadd18c9a272926457264c294 data/training/cosmo-b10900642_dim512_full.npy
06141caaad1035e9717a37ac5a134b9e data/training/cosmo-b11156005_dim512_full.npy
0522fb12a5e8ba6d98f8d242b338783d data/training/cosmo-b11616045_dim512_full.npy
```

```
$ head -5 fetch.txt
https://example.org/18625370-33/cosmo-b10206348_dim512_full.npy 1073741952 data/training/cosmo-b10206348_dim512_full.npy
https://example.org/18625370-329/cosmo-b10667031_dim512_full.npy 1073741952 data/training/cosmo-b10667031_dim512_full.npy
https://example.org/18625370-170/cosmo-b10900642_dim512_full.npy 1073741952 data/training/cosmo-b10900642_dim512_full.npy
https://example.org/18625370-422/cosmo-b11156005_dim512_full.npy 1073741952 data/training/cosmo-b11156005_dim512_full.npy
https://example.org/18625370-448/cosmo-b11616045_dim512_full.npy 1073741952 data/training/cosmo-b11616045_dim512_full.npy
```



BDBag CLI

bdbag: Command-line Interface (CLI)

Summary

The *bdbag* command-line program is designed to operate on either bag directories or single file archives of bags in a supported format, such as ZIP, TAR, or TGZ. Different functions are available based on the context of the input bag path. Therefore, when invoking *bdbag*, some command-line arguments may either be incompatible with other specified arguments, or may be invalid in the context of the bag path specified. In such cases, the program will notify the user with an error message indicating the incompatibility.

Usage

The only mandatory argument is a valid path to a local bag directory, a local bag archive file, or a URL/URI resolving to a remote bag archive file. All other arguments are optional.

Basic arguments:

```
usage: bdbag
[--version]
[--update]
[--revert]
[--archiver {zip,tar,tgz}]
[--checksum {md5,sha1,sha256,sha512,all}]
[--skip-manifests]
[--prune-manifests]
[--materialize]
[--resolve-fetch {all,missing}]
[--fetch-filter <column><operator><value>]
[--validate {fast,full,structure,completeness}]
[--validate-profile]
[--config-file <file>]
[--keychain-file <file>]
[--metadata-file <file>]
[--ro-metadata-file <file>]
[--ro-manifest-generate {overwrite, update}]
[--remote-file-manifest <file>]
[--quiet]
[--debug]
[--help]
<path>
```



Validate & Materialize

`--validate {fast,full,structure}`

Validate a bag directory or bag archive.

- If `fast` is specified, the `Payload-0xum` metadata field (if present) will be used to check that the payload files are present and accounted for.
- If `full` is specified, all checksums will be regenerated and compared to the corresponding entries in the manifest.
- If `structure` is specified, the bag will be checked for structural validity only.
- If `completeness` is specified, the bag will be checked for both structural validity and completeness (presence) of files listed in all manifests.

`--materialize`

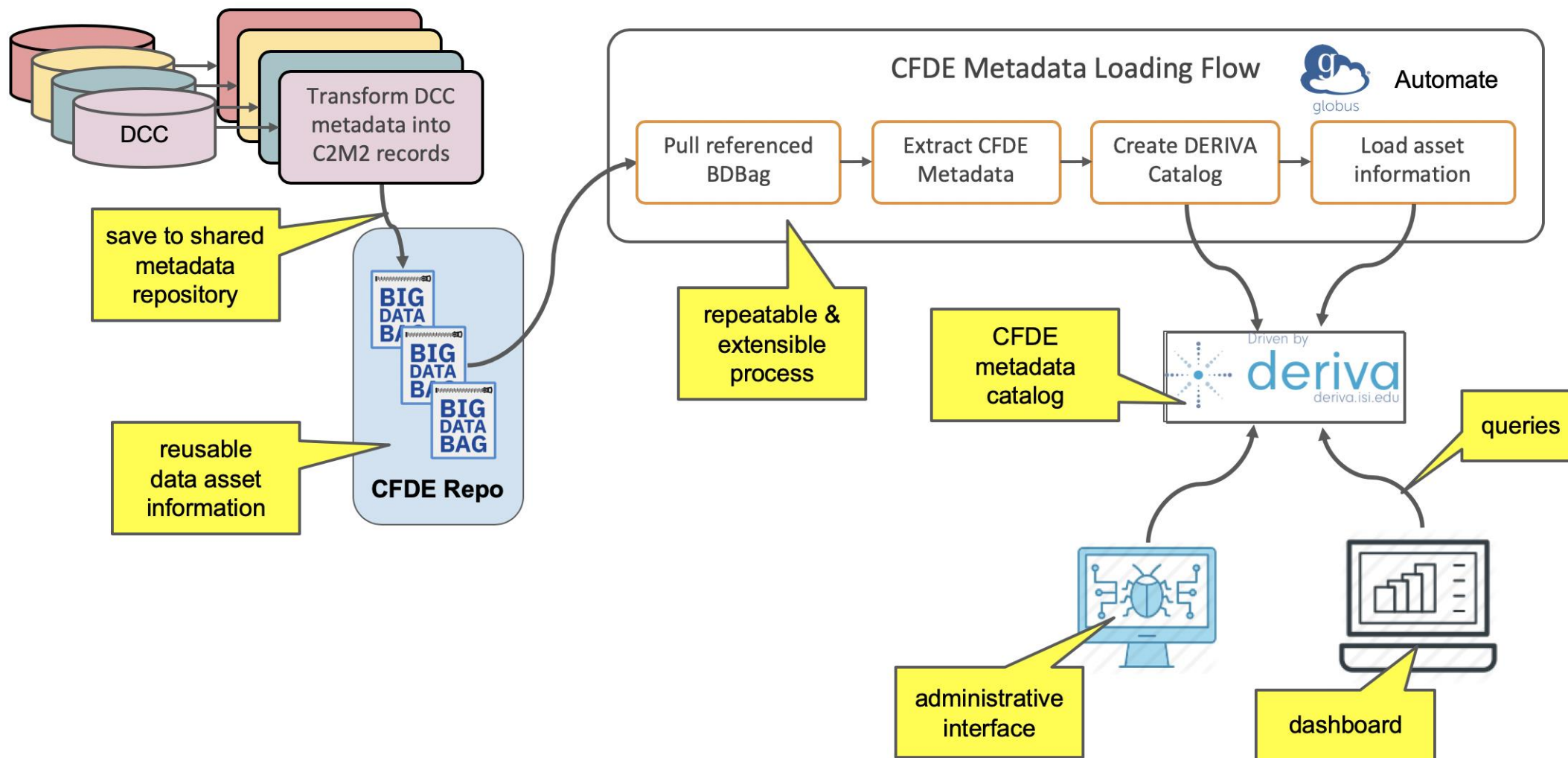
The `materialize` function is a bag bootstrapper. When invoked, it will attempt to fully *reconstitute* a bag by performing multiple possible actions depending on the context of the `<path>` argument.

1. If `<path>` is a URL or a URI of a resolvable identifier scheme, the file referenced by this value will first be downloaded to the current directory.
2. If the `<path>` (or previously downloaded file) represents a local path to a supported archive format, the archive will be extracted to the current directory.
3. If the `<path>` (or previously extracted file) represents a valid bag directory, any remote file references contained within the bag's `fetch.txt` file will attempt to be resolved.
4. Full validation will be run on the materialized bag. If any one of these steps fail, an error is raised.



Example: NIH Metadata Ingest

Scalable approach to integrating DCC asset information





References

- **Globus Transfer Docs**

- <https://docs.globus.org/api/transfer/>

- **E3SM Data Stager**

- https://github.com/E3SM-Project/e3sm_to_cmip/blob/master/scripts/data_stager.py
 - https://github.com/E3SM-Project/e3sm_to_cmip/blob/master/scripts/README.md

- **BagIt & BDBag**

- <https://tools.ietf.org/html/draft-kunze-bagit>
 - <https://en.wikipedia.org/wiki/BagIt>
 - <https://github.com/fair-research/bdbag>

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development
(NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674,
Fax: 202-459-9673, Email: nco@nitrd.gov, Website: <https://www.nitrd.gov>

