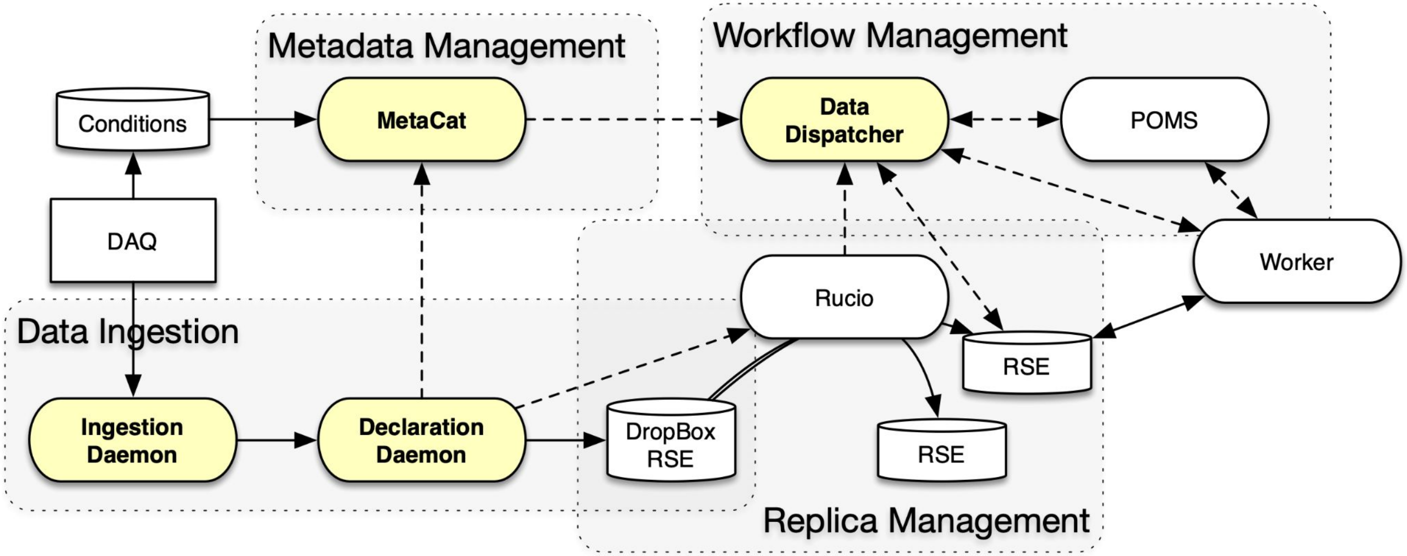
Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

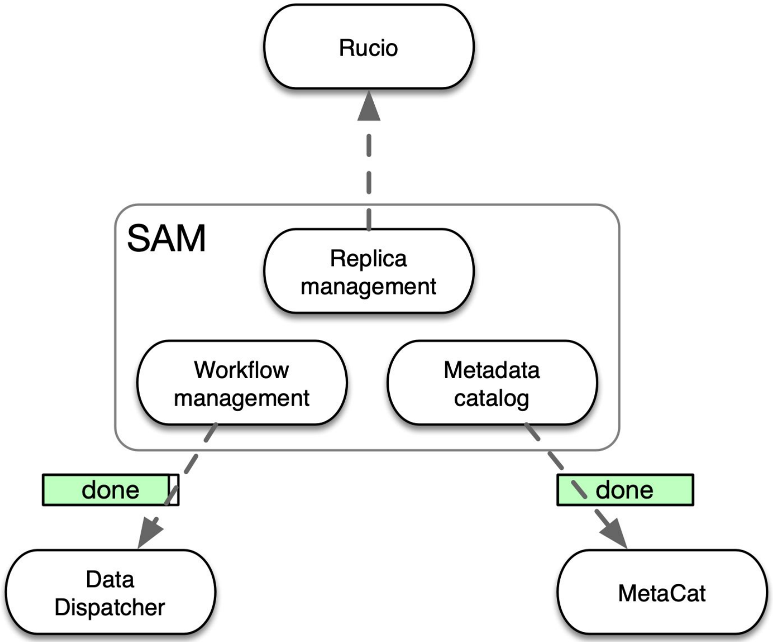
Data Management Software Update

Igor Mandrichenko, FNAL
DUNE, Summer 2022 Consortium Meeting

Big Picture


Metadata Management
Workflow Management
Conditions
MetaCat
Data Dispatcher
POMS
DAQ
Worker
Rucio
Data Ingestion
RSE
Ingestion Daemon
Declaration Daemon
DropBox RSE
RSE
Replica Management

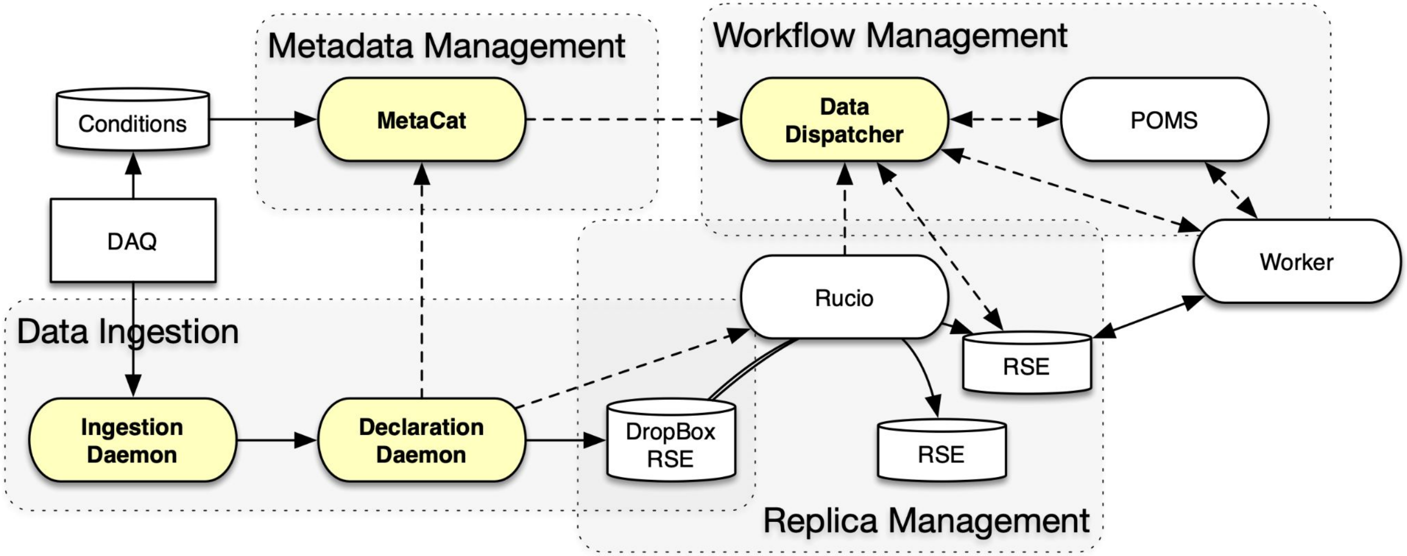
SAM Replacement Project



SAM functions:

- Replica Management -> Rucio (driver)
- Metadata Management -> MetaCat
- Workflow Management (SAM Station) -> Data Dispatcher

Data Ingestion



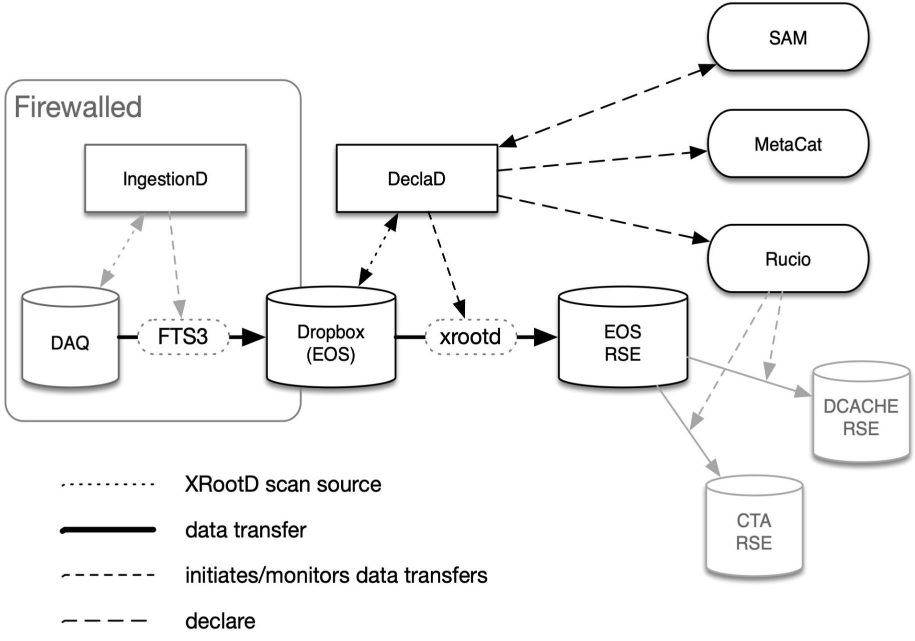
Ingestion Daemon

Declaration Daemon

How they both work

- Use XRootD client to scan the source
- Once data and metadata files are there,
- Copy data and metadata
  - Use FTS3 (IngestionD)
  - Use xrdcp --tpc (DeclaD, EOS to EOS)
- DeclaD: declare the file to:
  - Rucio
    - Create dataset with replication rules
  - MetaCat
  - SAM (just in case)
- Remove sources (rename, move, delete)
- Multiple threads, one file (data and metadata) per thread
- Local persistent DB with historic data, GUI for monitoring

Data Ingestion



Data challenge data flow

In terms of SAM replacement:

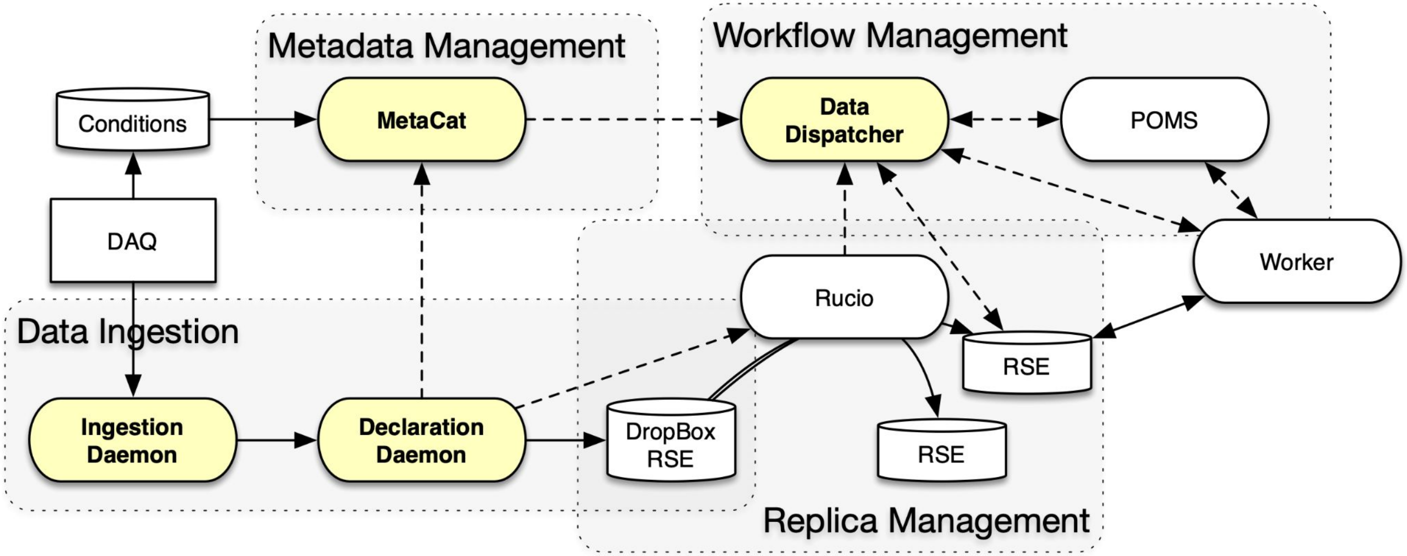
- FTS -> Declaration Daemon
- FTSLight -> Ingestion Daemon

- Scan the source
- Copy to the destination
- Declare (optionally)
- Remove/hide the source
- Repeat

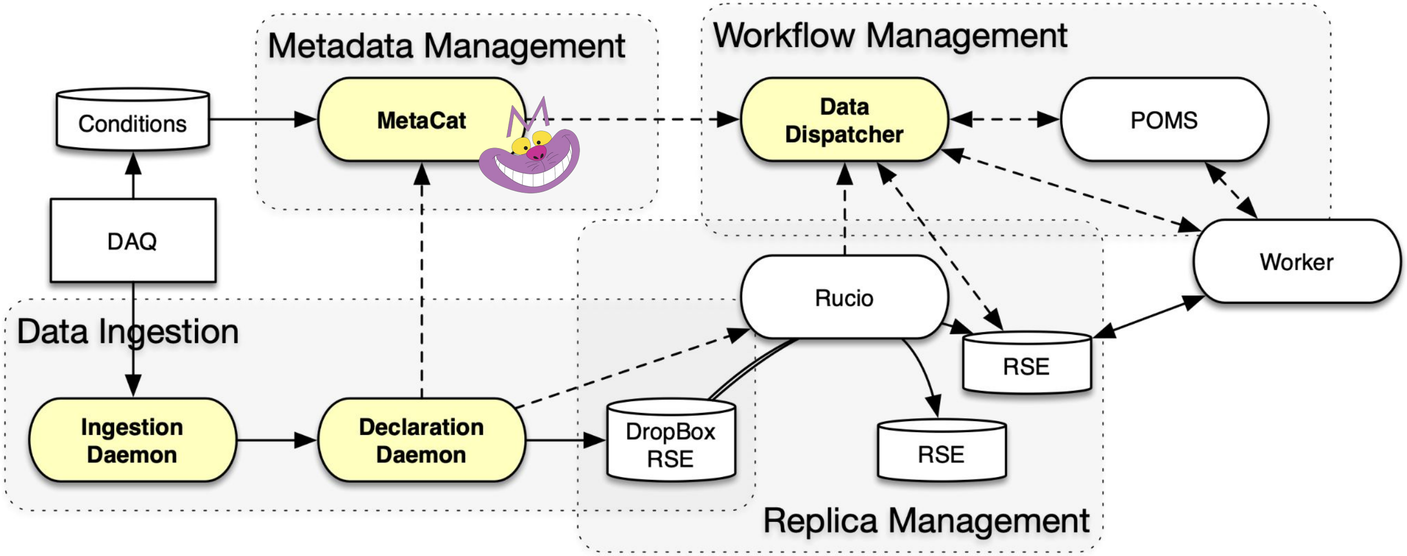
Data Ingestion in Data Challenge

- Much needed real life testing
  - Debugging, features
- Tested at the production file and data transfer rates
  - ~0.5 Hz in average, ~1 Hz in peak
  - 2GB/s data transfer rate average
- Lesson learned:
  - Combine Ingestion and Declaration daemons into single configurable product, with options:
    - Transport mechanism - FTS3 or “local” copy
    - Whether to declare files to MetaCat/Rucio/SAM

MetaCat - metadata catalog



MetaCat - metadata catalog

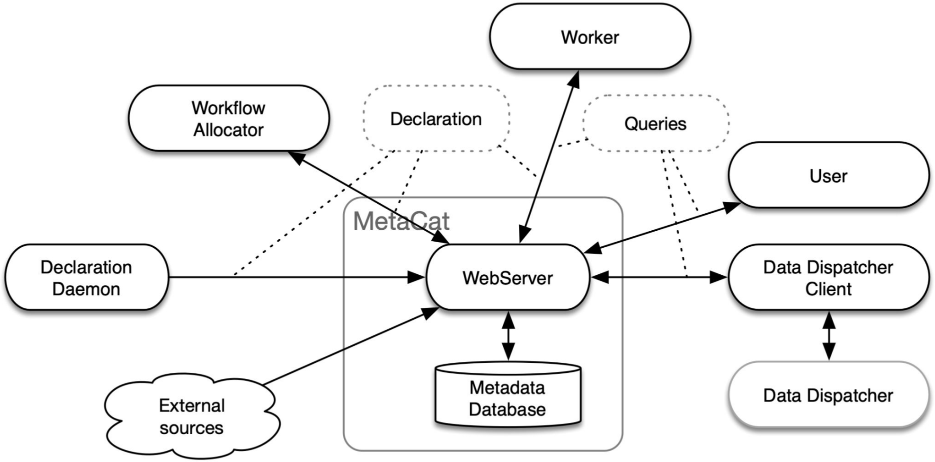


- Store metadata
- Efficient metadata queries
- External metadata access without copying

MetaCat Updates

- Project is pretty stable
- Demonstrated access to external data sources
  - Filter file set by data stored elsewhere
  - Make metadata from the external source available for querying/output
  - Runs conditions database
    - Access to another relational DB
  - Rucio replica locations
    - API access to another system
- Documentation: https://metacat.readthedocs.io

Integration with other components



- Data Dispatcher
  - Query when creating a project
- Declaration Daemon
  - Declaration of new files
  - POMS via the DD client (?)
- Workflow Allocator
  - Declaration of new files
- External data sources
  - Runs conditions DB
  - Rucio (replica locations)

MetaCat during the Data Challenge

Not much needed to be done

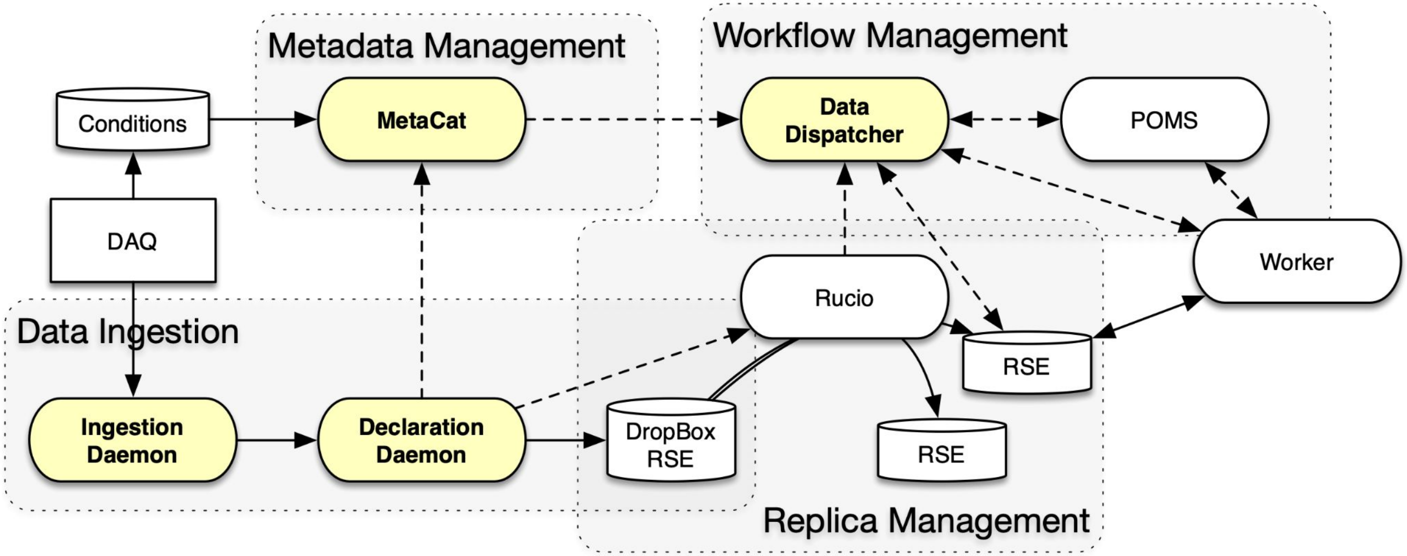
- Handled the query/declaration load well: ~0.5 Hz average (higher in peaks, 1-2 Hz ?)
- Still using development instance of the DB
  - Need to migrate to the one supported by DBAs
- Fixed minor things in API, documentation
- `declare_meta.py` - declare a file using JSON file in the format produced by DAQ
  - Convert to MetaCat API format
- Issues with handling delegated X.509 proxy certs while authenticating the client

MetaCat: to do

MetaCat is a mature product tested well by the data challenge

- Still need more user testing - may need more features
- Support for features SAM users need/like
  - File retiring ?
- Create parameter category structure for ProtoDUNE/DUNE
  - Test the functionality of parameter namespace enforcement
- WLCG tokens for authentication
- Anonymized user identity

Data Dispatcher

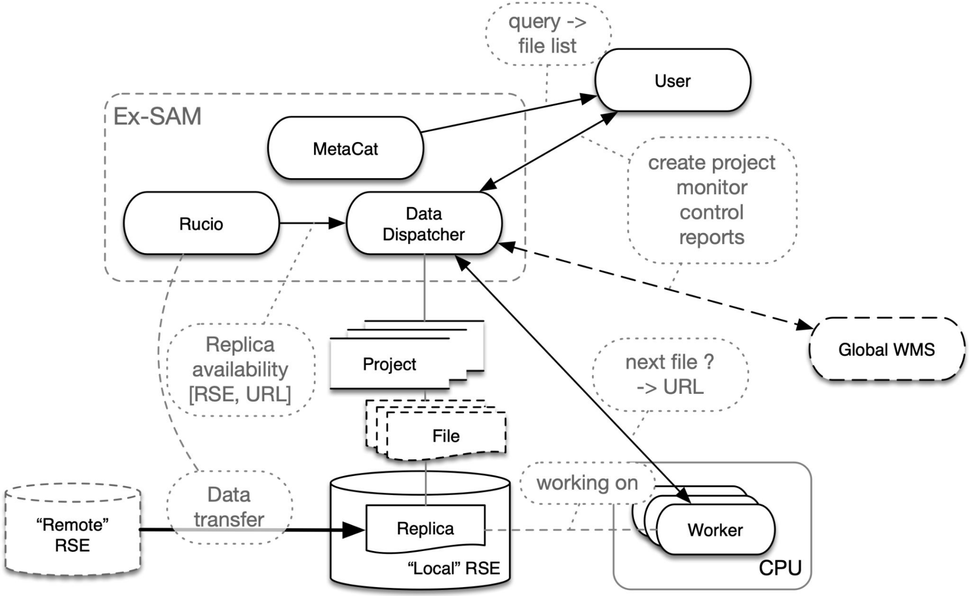


Project level workflow management

Data Dispatcher

Scope: replacement for SAM Station

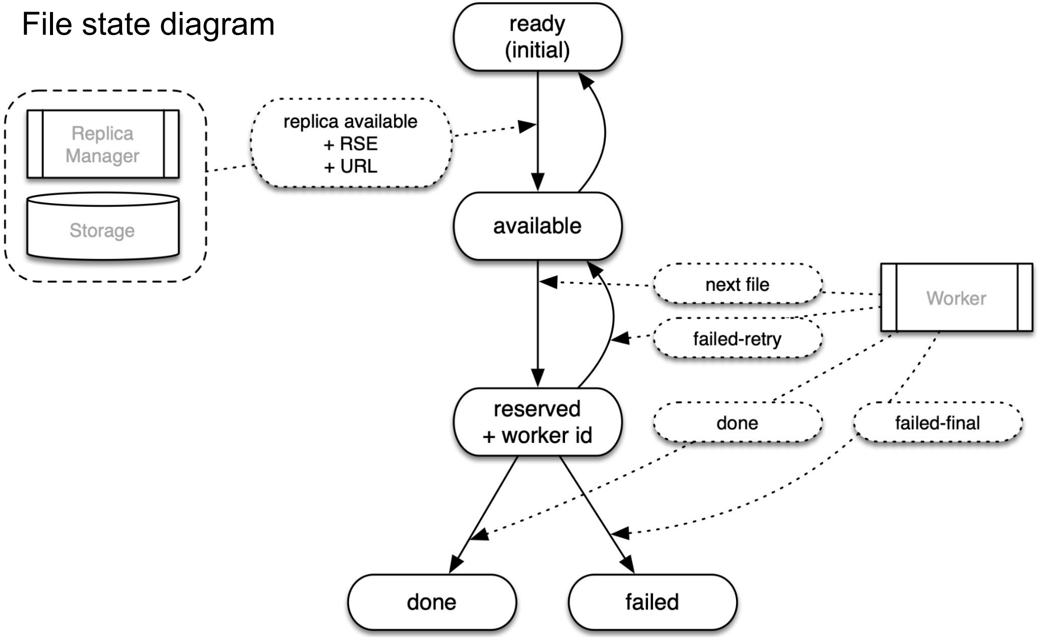
- “Local” workflow management
- Interface for Global WMS - POMS or similar



- Project is a set of files to process
- Worker interface “get next file”
- Project management, monitoring, history

Project-level Workflow Management

File state diagram

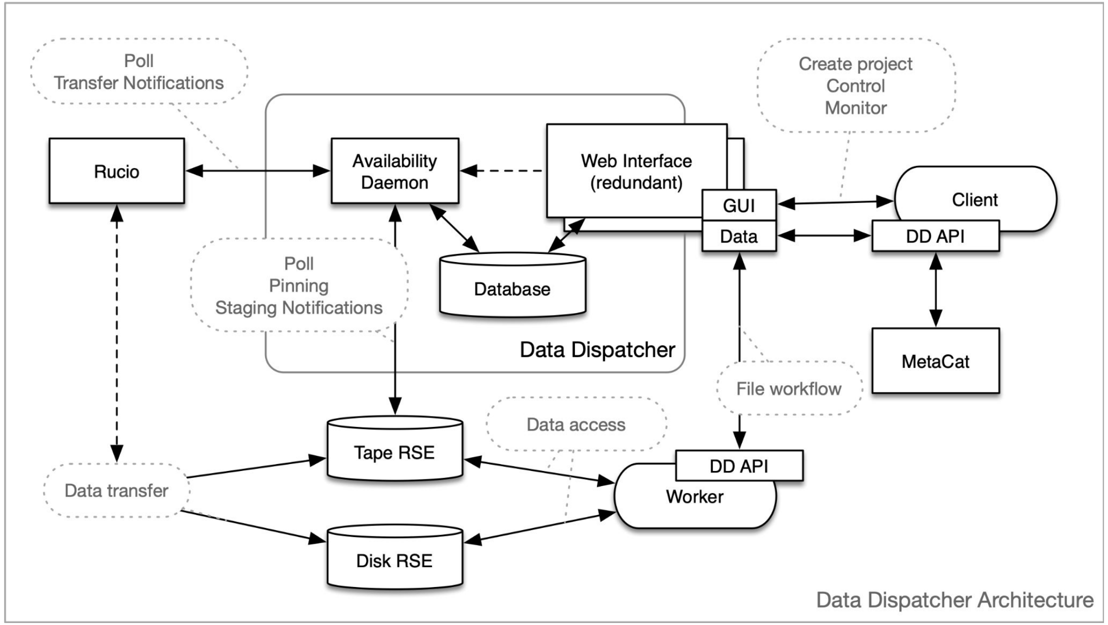


Worker:

- Get next file (CPU site, project, worker id)
  - -> DID, metadata, URL
  - Block until a file becomes available or project ends
- File done(project id, DID)
- File failed(project id, DID)
  - To be retried ?

To do: what if the worker dies without releasing the file ?

Interfaces

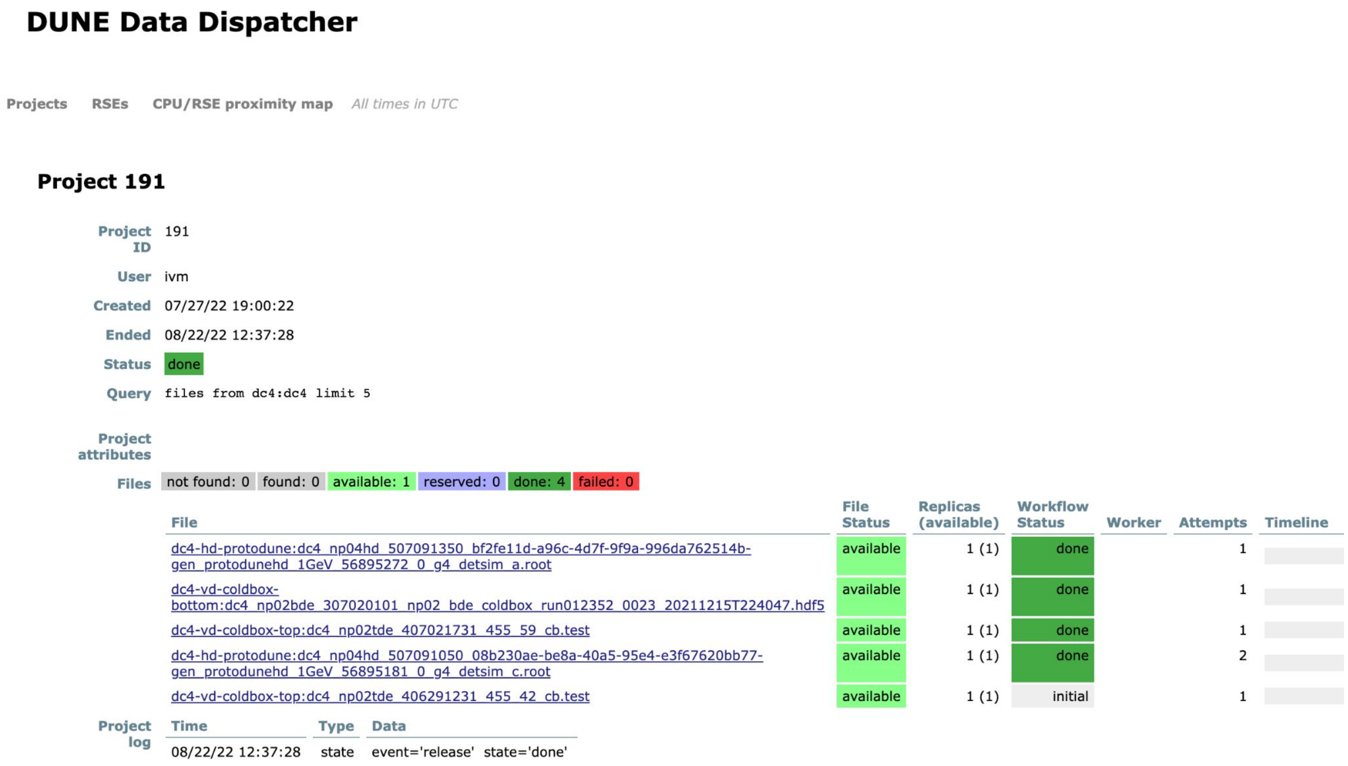


- Command line (CLI)
  - Create project
  - Monitor project status
  - Replica availability
  - Cancel/restart project
- Worker interface (CLI)
  - Next file
  - File done/failed
- API (Python), via REST
  - Same functions as CLI
- GUI
  - Project monitoring, history
  - Replica availability, history

Documentation: https://data-dispatcher.readthedocs.io

- Installation, CLI, API

GUI



DUNE Data Dispatcher

Projects **RSEs** **CPU/RSE proximity map** *All times in UTC*

Project 191

Project ID	191
User	ivm
Created	07/27/22 19:00:22
Ended	08/22/22 12:37:28
Status	done
Query	`files from dc4:dc4 limit 5`
Project attributes	
Files	not found: 0 found: 0 available: 1 reserved: 0 done: 4 failed: 0

File	File Status	Replicas (available)	Workflow Status	Worker	Attempts	Timeline
dc4-hd-protodune:dc4_np04hd_507091350_bf2fe11d-a96c-4d7f-9f9a-996da762514b-gen_protodunehd_1GeV_56895272_0_g4_detsim_a.root	available	1 (1)	done		1	
dc4-vd-coldbox-bottom:dc4_np02bde_307020101_np02_bde_coldbox_run012352_0023_20211215T224047.hdf5	available	1 (1)	done		1	
dc4-vd-coldbox-top:dc4_np02tde_407021731_455_59_cb.test	available	1 (1)	done		1	
dc4-hd-protodune:dc4_np04hd_507091050_08b230ae-be8a-40a5-95e4-e3f67620bb77-gen_protodunehd_1GeV_56895181_0_g4_detsim_c.root	available	1 (1)	done		2	
dc4-vd-coldbox-top:dc4_np02tde_406291231_455_42_cb.test	available	1 (1)	initial		1	

Project log

Time	Type	Data
08/22/22 12:37:28	state	event='release' state='done'

Project and file metadata

When creating a project in Data Dispatcher, optionally:

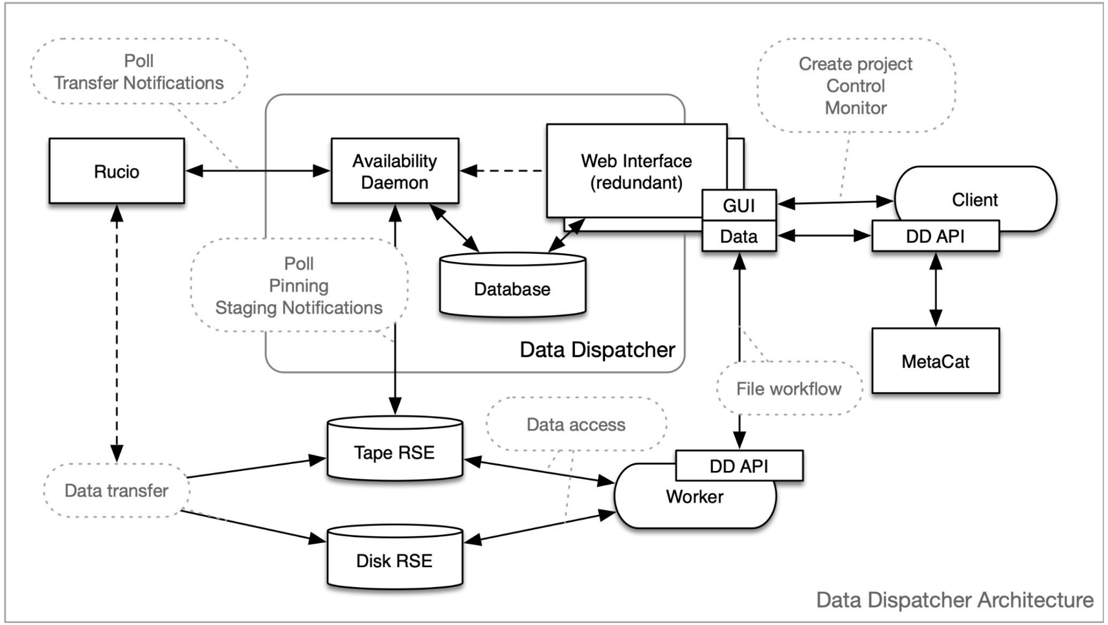
- Attach arbitrary metadata to the project and/or files
- Copy metadata from MetaCat

Worker (with “get next file”) receives:

- The project metadata
- File metadata

Can be used to parametrize the project

Replica Availability

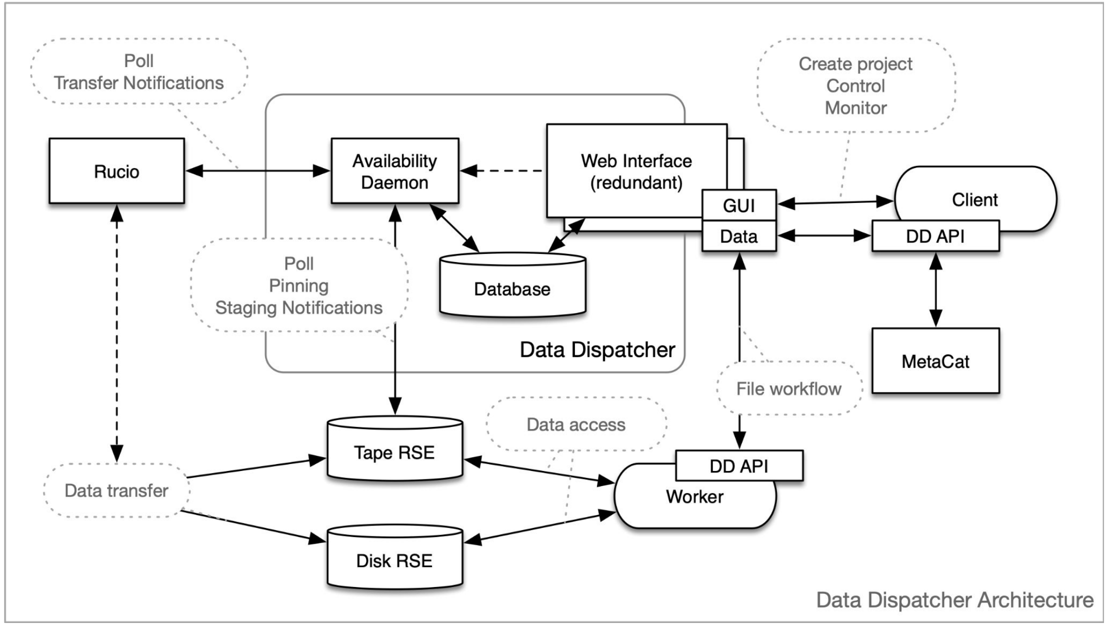


Data Dispatcher Architecture

Replica availability information

- Bulk query Rucio when DD starts or new project is created
  - Scale: 10K files
- Notifications from Rucio when replica availability changes
  - STOMP message queue
  - Message Broker

Replica Availability



Data Dispatcher Architecture

Replica staging for tape-based systems

- For now, dCache only
- Create bulk stage/pin request
- Poll for request status
  - Poll for individual replica availability if the request is in progress

Data Dispatcher: To DO

- Staging interface to CTA, other tape systems
- WLCG tokens for authentication
- Finish integration with POMS
- GUI/reporting improvements - implement SAM GUI features
- Anonymized user identity
- Proposal to integrate Workflow Allocator with Data Dispatcher published, can be discussed