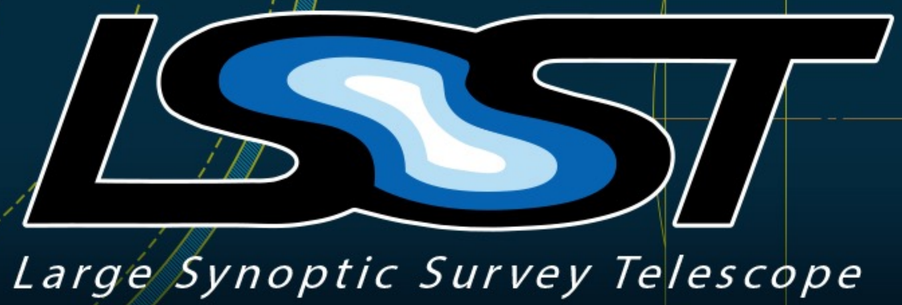
Data Access Use Cases & Requirements

Tim Jenness, for the Butler Working Group

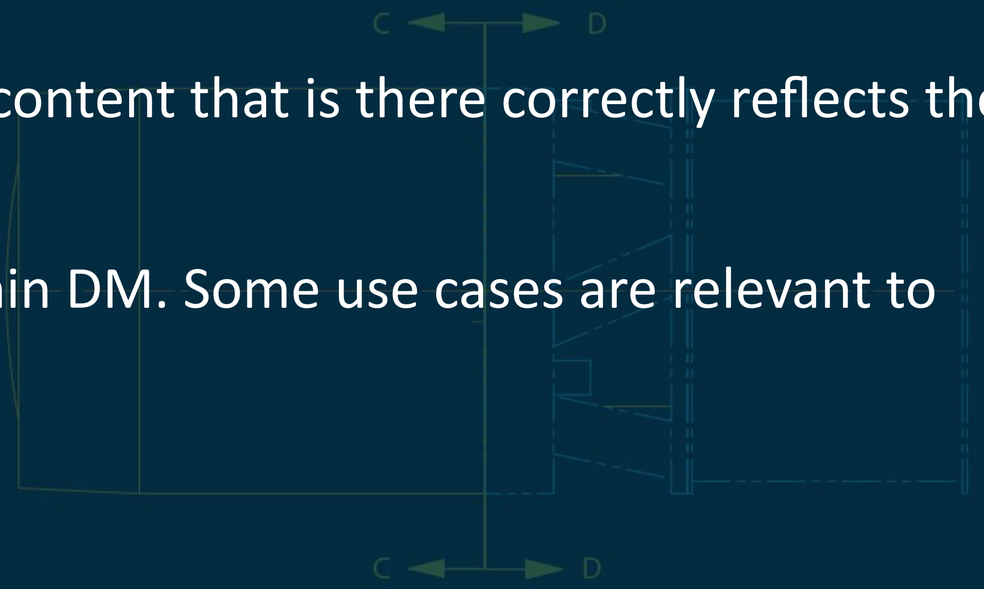
Butler Review



December 15th 2017

Data Access Use Cases: LDM-592

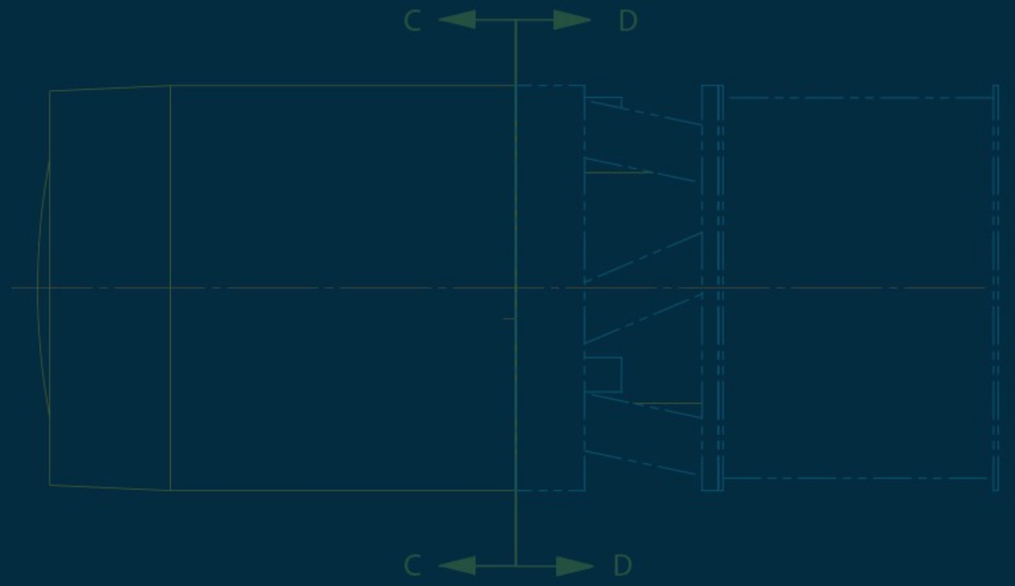
- Use Cases have been developed covering aspects of Data Access relating to the Butler, but the use cases have not been restricted solely to the Butler part; context is provided.
- Use Cases are not comprehensive and could usefully be expanded to drive requirements in other parts of the system.
- The key question for the use cases is whether the content that is there correctly reflects the system we are building.
- Use Cases are labeled by the originating team within DM. Some use cases are relevant to multiple teams.
- We list here some representative use cases.



Use Case: ARCH3

Batch Processing Data With E F D And Updated W C S And Visit Metadata

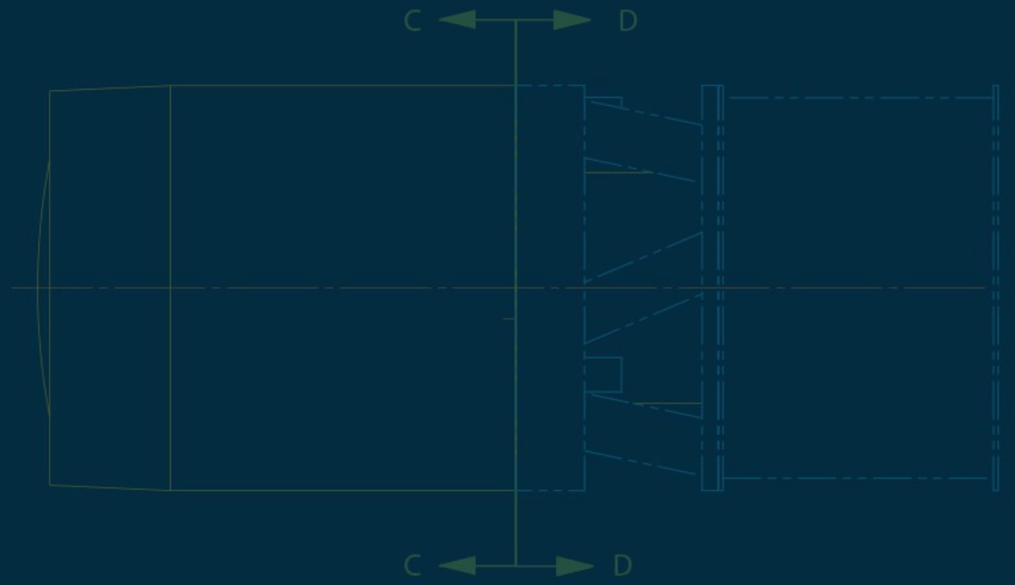
- Requires processing jobs have access to specific items of EFD telemetry.
- Describes the facility for the raw file to have augmented headers over the original raw.
- Describes the need to use a previously calculated WCS with a raw file.



Use Case: ARCH4

Submitting Batch Jobs Via A Notebook

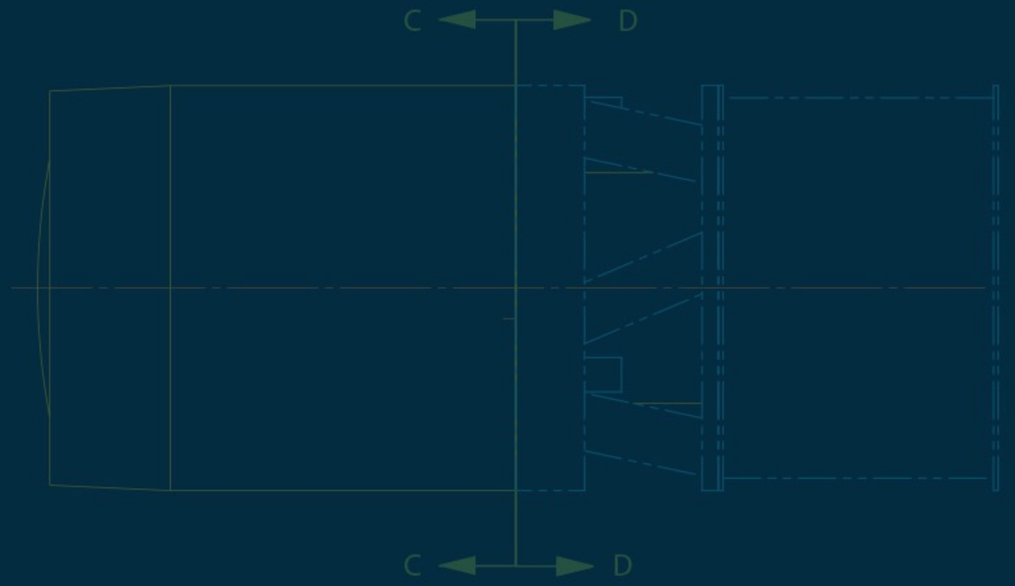
- A batch job submitted from a notebook will run on multiple nodes.
- Each node will write out results.
- Notebook user wants to see an integrated output data repository once the jobs complete, not have to work out how to combine the individual outputs.



Use Case: AP1

Prompt Processing: Process Raw Images And Generate Alerts

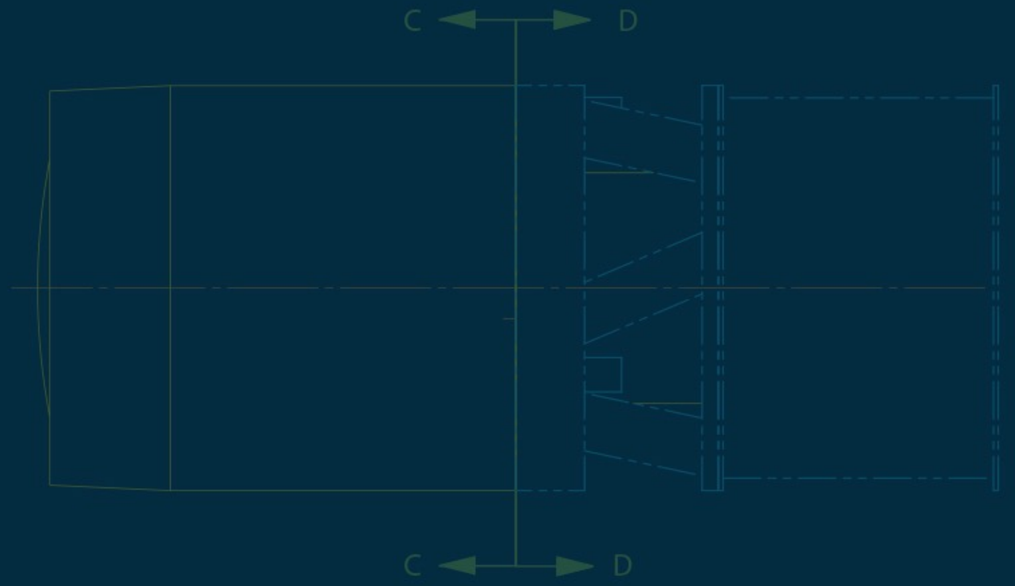
- Standard prompt processing of standard visit.
- Two snaps processed.
- DIAObjects are retrieved using the butler from L1 database and must have access to DIAObjects from the previous visit.
- Alerts issued to alert broker.



Use Case: COMM4

Comparison Of The Night's Data With Archived Data

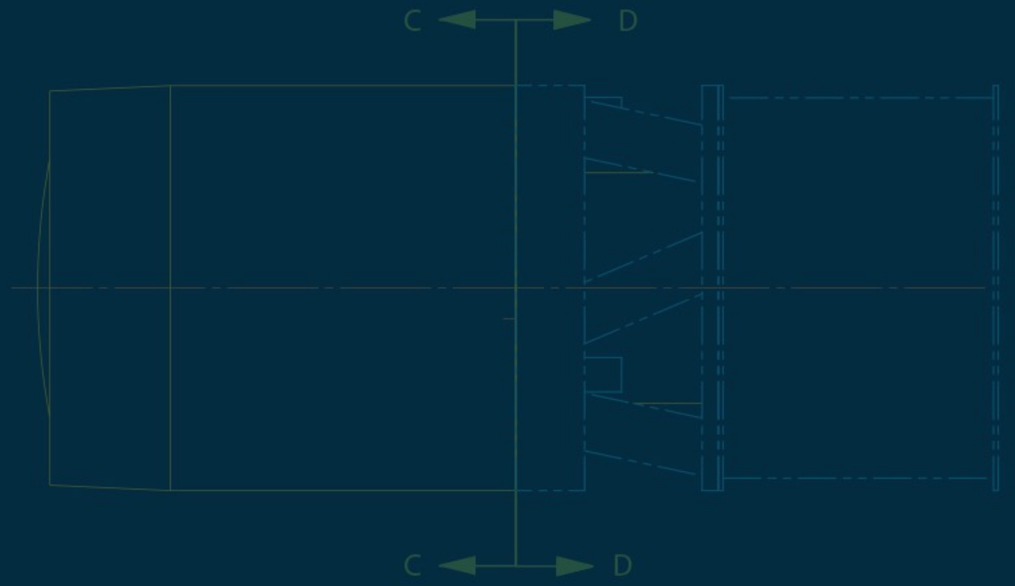
- Commissioning Scientists comparing an analysis carried out on the current night's data on the commissioning cluster, to data taken at a previous time separated by many months and stored in semi-permanent storage at NCSA.



Use Case: DAX8

Repository Migration And Repository Versioning

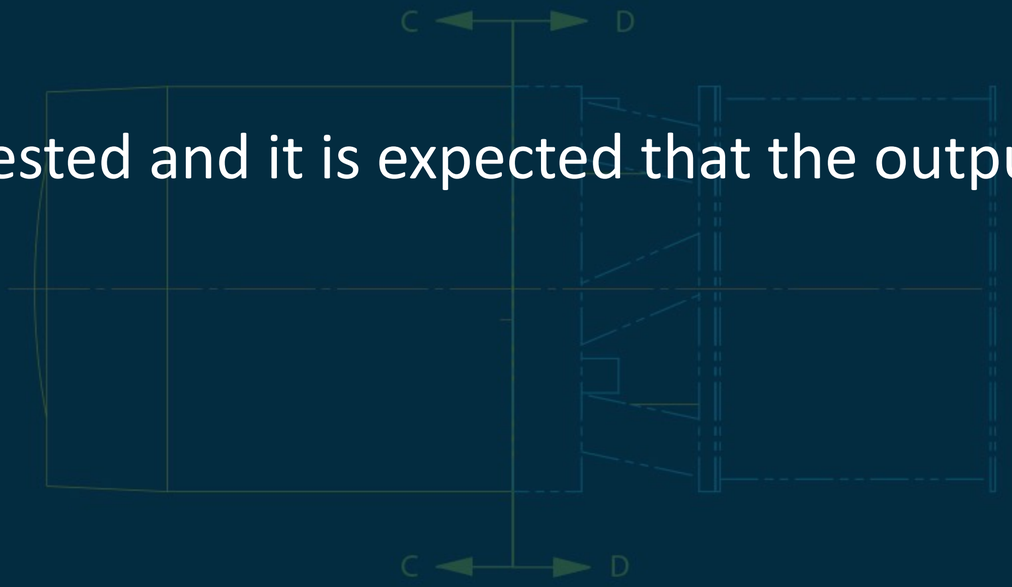
- Upgrading old repositories by migrating them to the new version using a one-off script.
- Butler will not natively support all previous repository versions.



Use Case: LDF1 / SCIVAL1

Submit A Processing Run To The Batch Processing Service

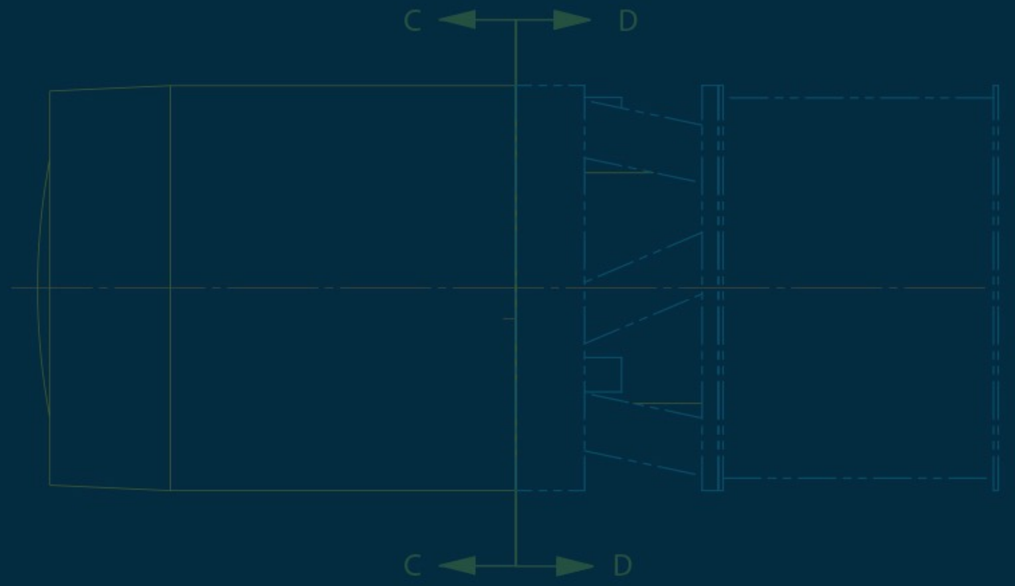
- LSST Data Facility batch processing will work entirely on local files.
- A “pre-flight” step will query the SuperTask to determine which image and catalog files are needed.
- Those files are retrieved from the data backbone and stored on the node for processing.
- Once the job completes the results are harvested and it is expected that the output file names are unique and predictable.



Use Case: SQR2

Data Discovery Based On Observation/Processing Metadata

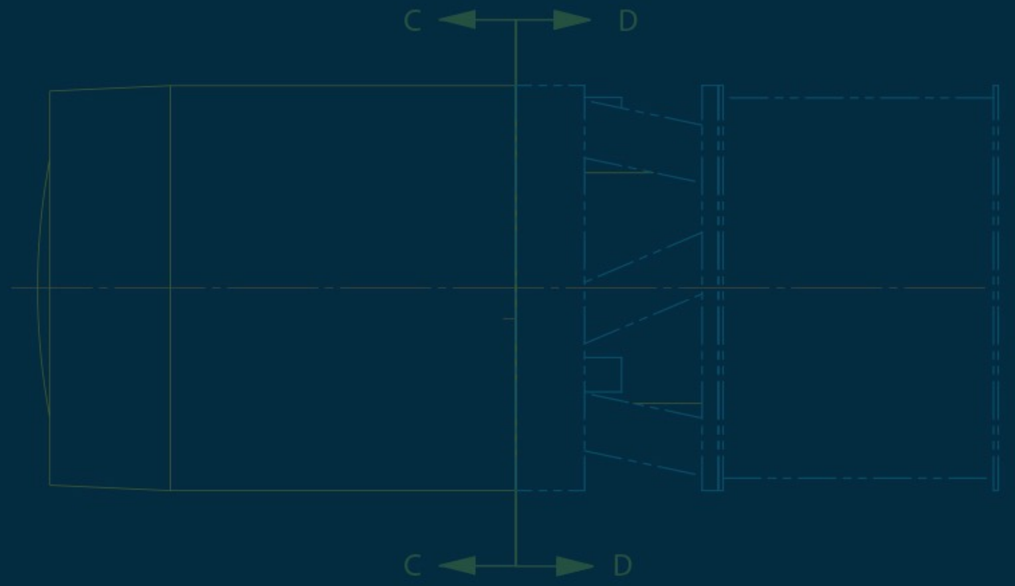
- Retrieving data that meet a specified seeing cut.



Use Case: SQR14

Share Datasets Via V O Space

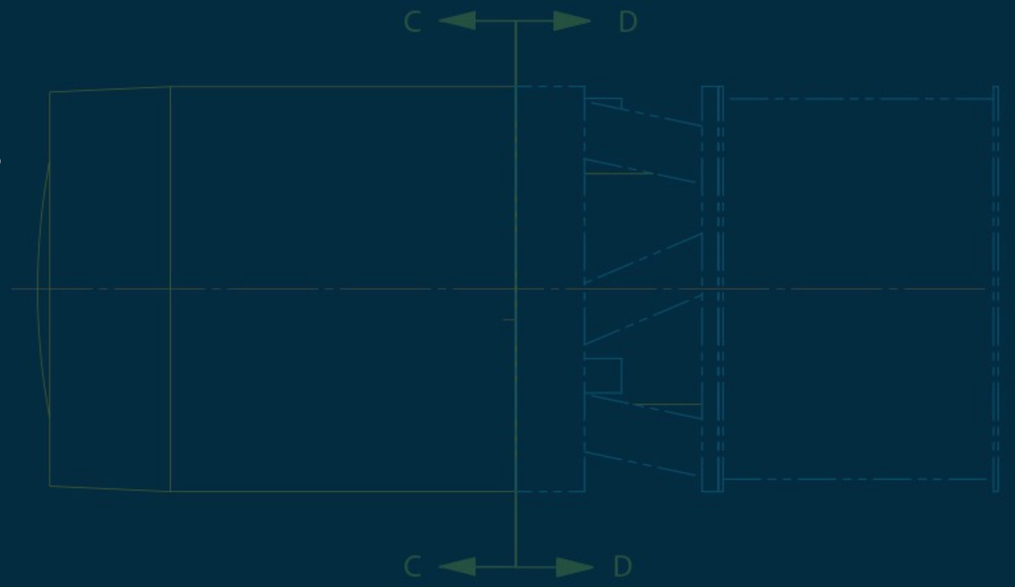
- Data written to a user workspace in the LSST Science Platform should be accessible to a butler user on the VOSpace on their laptop, or accessible to other VOSpace users on the LSP who have been given access. The butler must not download all the files locally, but must only download the files that will be opened.



Requirements: LDM-556

Butler And Supertask

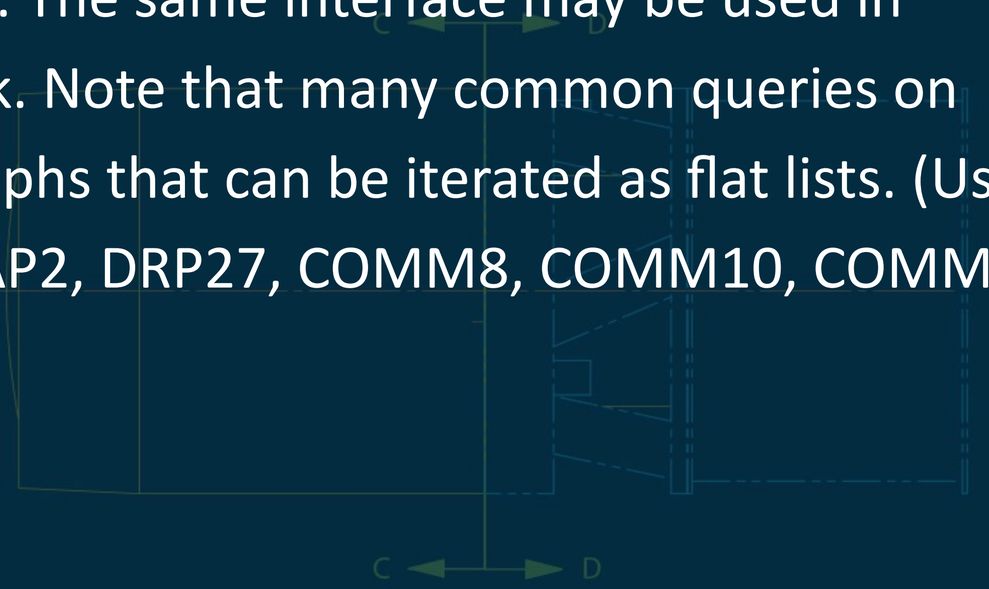
- LDM-556 contains SuperTask requirements as well as Butler requirements. Neither have been formally baselined.
- Working Group determined some requirements that should be considered as LDF requirements. Those are at end of document and should be migrated to the LDF requirements model.
- Butler requirements are linked to Use Cases.
- Butler Requirements have been prioritized.



Requirement: DMS-MWBT-REQ-0083

Consistent Discovery Interface (1A)

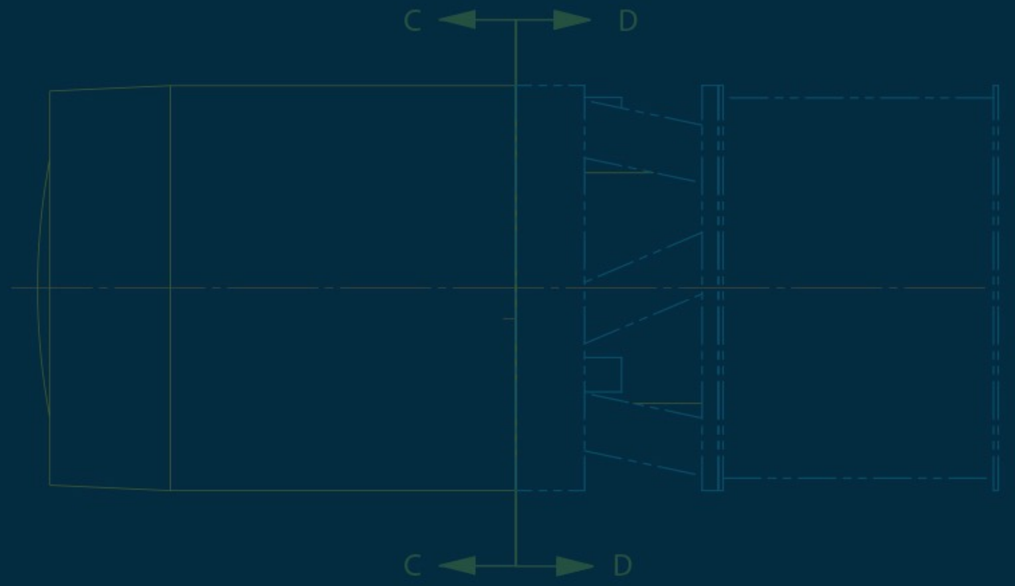
- The Data Discovery System shall provide a consistent interface for obtaining a graph that represents the DataUnits and Datasets in a Data Repository that match user-specified criteria.
- This is an interface expected by SuperTask preflight, and we need to make it consistent in all contexts in which SuperTasks will be launched. The same interface may be used in (possibly interactive) analysis and validation work. Note that many common queries on DataRepository contents may result in simple graphs that can be iterated as flat lists. (Use-Cases: DRP1, DRP7, SCIVAL1, SCIVAL2, SCIVAL3, AP2, DRP27, COMM8, COMM10, COMM13)



Requirement: DMS-MWBT-REQ-0053

Enabling Supertasks To Execute (1A)

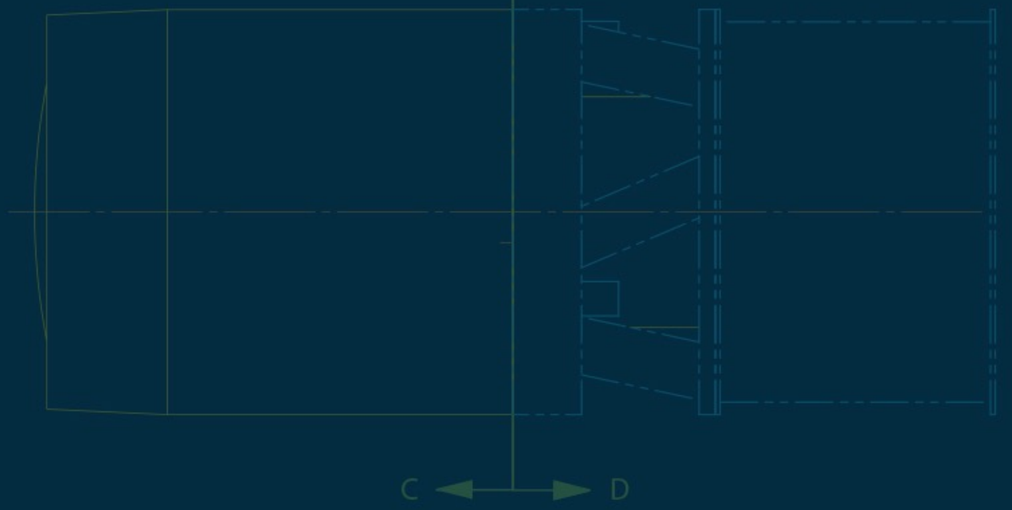
- It shall be possible for the Data Input System to construct a ConcreteDataset from a set of files stored locally on disk (without a remote database connection).
- For example, the batch processing system will have retrieved a valid set of files from the Data Backbone and copied them to a local disk. (UseCases: LDF1, LDF3)



Requirement: DMS-MWBT-REQ-0081

Multiple Chained Input Data Repositories (1A)

- The Data Discovery System shall be able treat multiple input DataRepositories as a single coherent logical repository.
- This could be a local on disk repository and a remote repository, with the the Data Discovery system scanning each in turn. Each dataset read in will contain provenance describing the Data Repository it came from. (UseCaseS: COMM4, LDF104)



Requirement: DMS-MWBT-REQ-0012

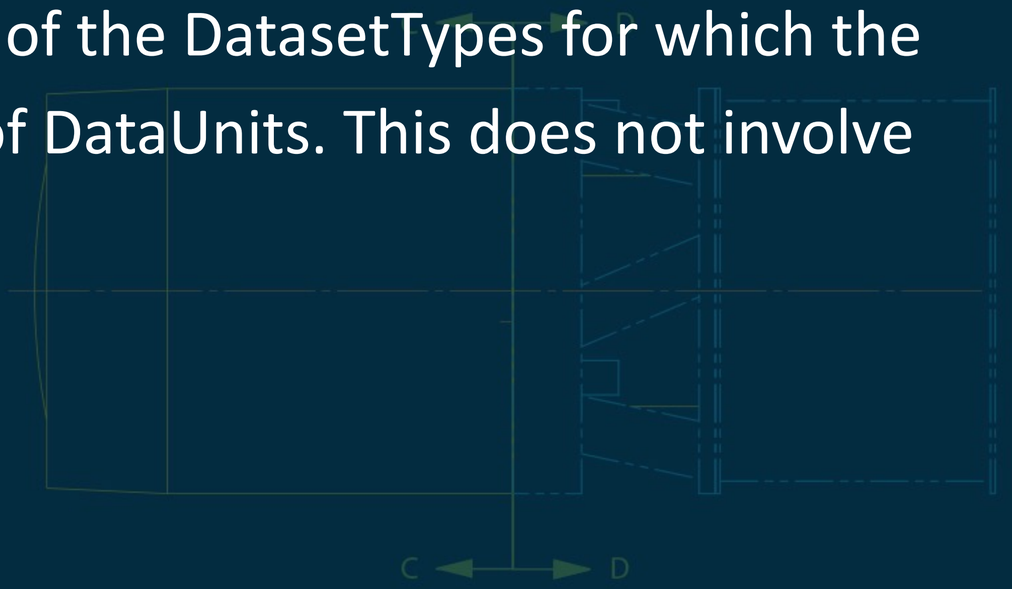
Data Repository Layering (1A)

- A DataRepository (A) shall be usable as an input for processing in a context (B), with its contents appearing as part of the Data Repository used to hold the outputs of the processing, for certain combinations of (A) and (B).
- Generally speaking, smaller-scale processing runs initiated by users with fewer permissions should be able to build on larger-scale processing runs initiated by users with more permissions. This requirement probably cannot be satisfied efficiently by always copying the full original input data repository (A) to the final output repository (B); it almost certainly implies some kind of on-demand transfer or aliasing. (UseCases: DRP1, DRP2, DRP3, DRP7, DRP8, SCIVAL1, SCIVAL2, SCIVAL3, SCIVAL4)

Requirement: DMS-MWBT-REQ-0010

Subsetting A Data Repository Without Data Transfer (1A)

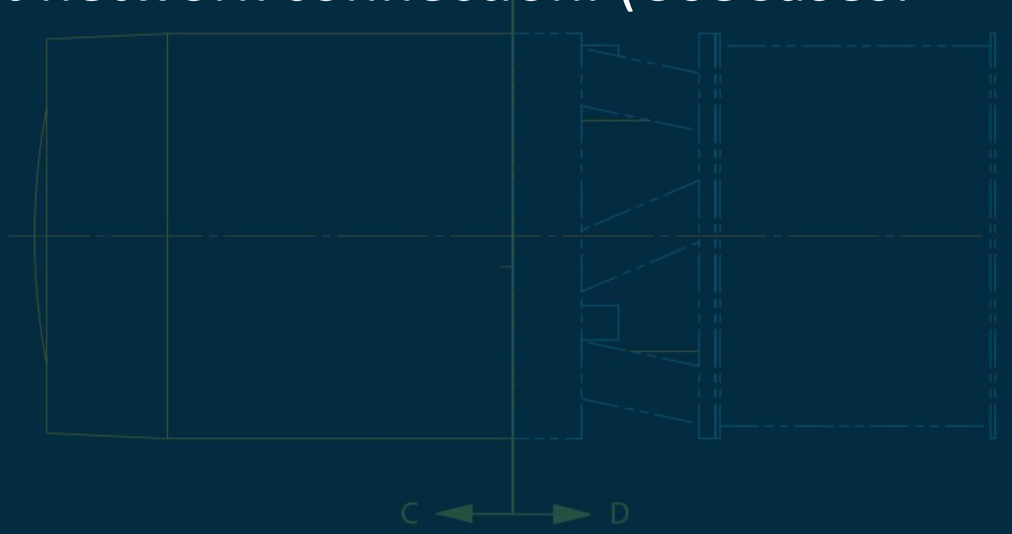
- It shall be possible to easily create a new DataRepository which is a view of a sub-section of an existing DataRepository, given a list of DataUnits and a list of DatasetTypes
- That is, given a list of DataUnits and a list of DatasetTypes, create a new DataRepository with enough information to access any of the DatasetTypes for which the necessary Data Units exist for the input list of DataUnits. This does not involve copying datasets. (UseCases: SQR1, AP1dev)



Requirement: DMS-MWBT-REQ-0011

Subsetting A Data Repository With Data Transfer (1A)

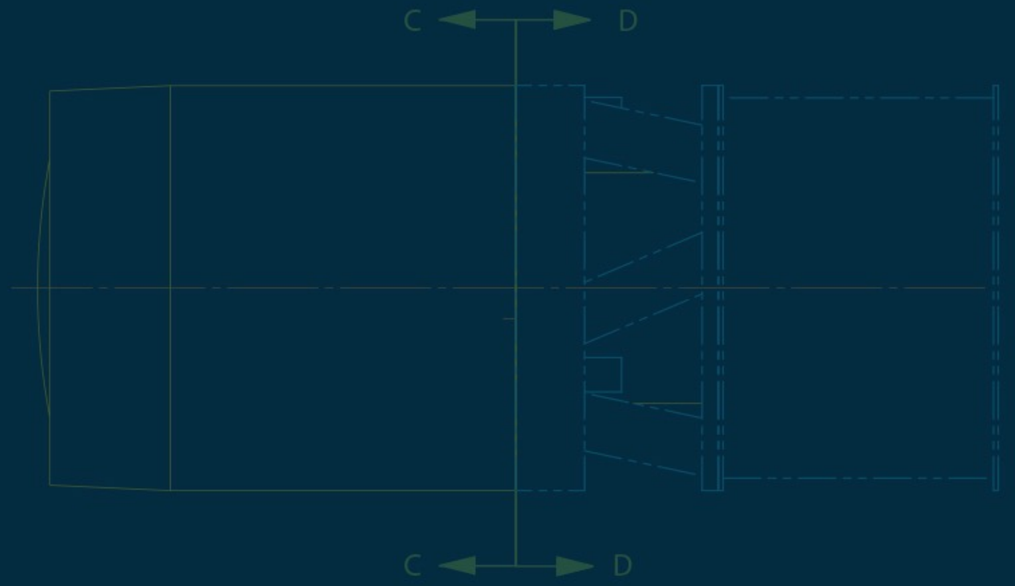
- It shall be possible to easily create a new DataRepository which contains a copy of a sub-section of an existing DataRepository, given a list of DataUnits and a list of DatasetTypes.
- This would transfer the files but not load them into Python objects, thus allowing for instance a processing run to be done without network connection. (UseCases: DRP16, DAX1, LDF103, SQR1)



Requirement: DMS-MWBT-REQ-0089

Filter By Data Quality (1B)

- The Data Discovery System shall be able to filter search results based on data quality assessments.
- For example, ask for raw data that has been flagged as bad to not be included in a coadd. (UseCases: LDF102, LDF1)



Requirement: DMS-MWBT-REQ-0088

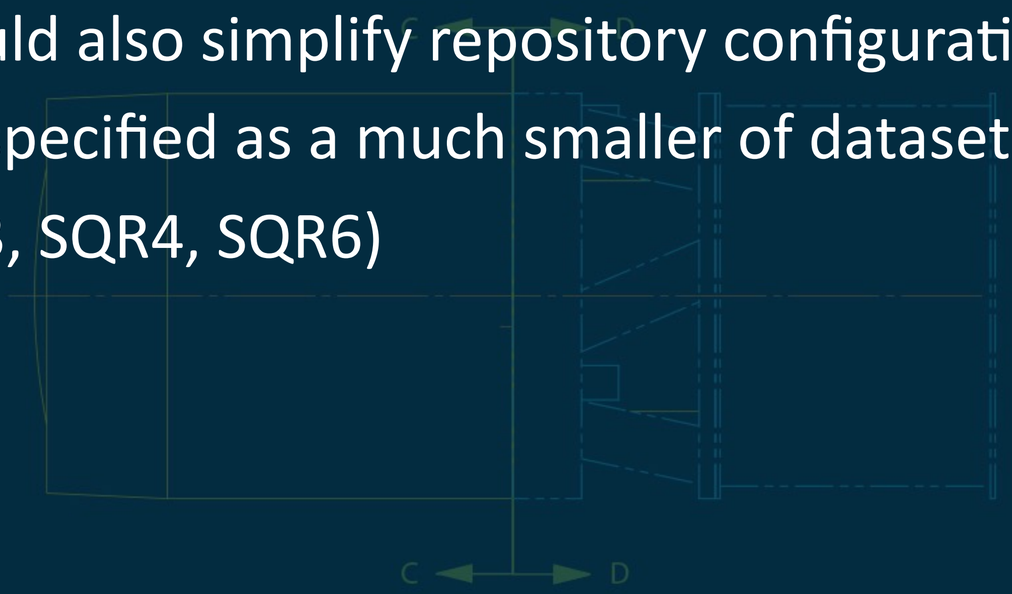
Filter By Non-Dataset Ref Database Entries (1A)

- The Data Discovery System shall be able to filter search results based upon specified filters that need non-DatasetRef database entries.
- This includes joins with LDF Operator specific tables not normally known to the Data Discovery System. These cuts could include, but will not be limited to, seeing, data quality flags, and airmass. (UseCases: LDF1, SQR2, COMM7, SQR1, SQR2, LDF102)

Requirement: DMS-MWBT-REQ-0059

Creation Of New Dataset Types (1A)

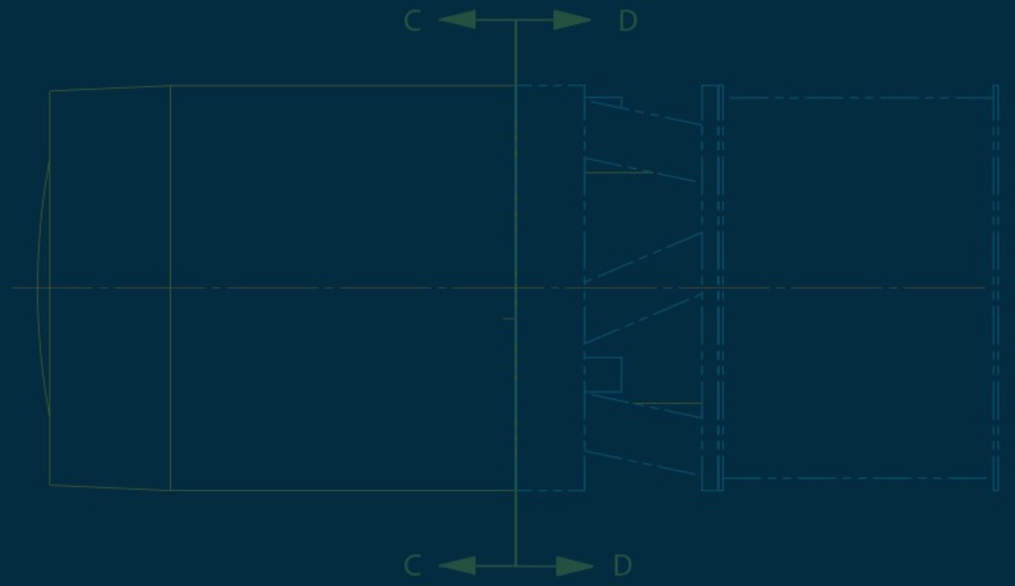
- The Data Output system shall allow a new DatasetType to be registered with a DataRepository, programmatically and at Supertask preflight-time, allowing Datasets of that DatasetType to be added to that DataRepository thereafter.
- This allows persisting config and metadata from a new supertask or command- line task without editing any obs packages. It could also simplify repository configuration as many predefined dataset types could be specified as a much smaller of dataset prototypes. (UseCases: DRP12, SCIVAL1, AP3, SQR4, SQR6)



Requirement: DMS-MWBT-REQ-0002

Versioning Of Data Repositories

- The Data Input/Output system shall be able to describe the version of a DataRepository.
- (UseCases: DAX8)



Requirement: DMS-MWBT-REQ-0003

Repository Version Migration

- The Data Input/Output system shall be able to perform persistent migrations of a DataRepository to bring the Data Model of that DataRepository up to parity with the Data Model expected by the current Data Input/Output System interfaces.
- This is a tool for creating a repository from a repository with an old version to use the current version. Silent in-place updates of a repository should not occur. (UseCases: DAX8)