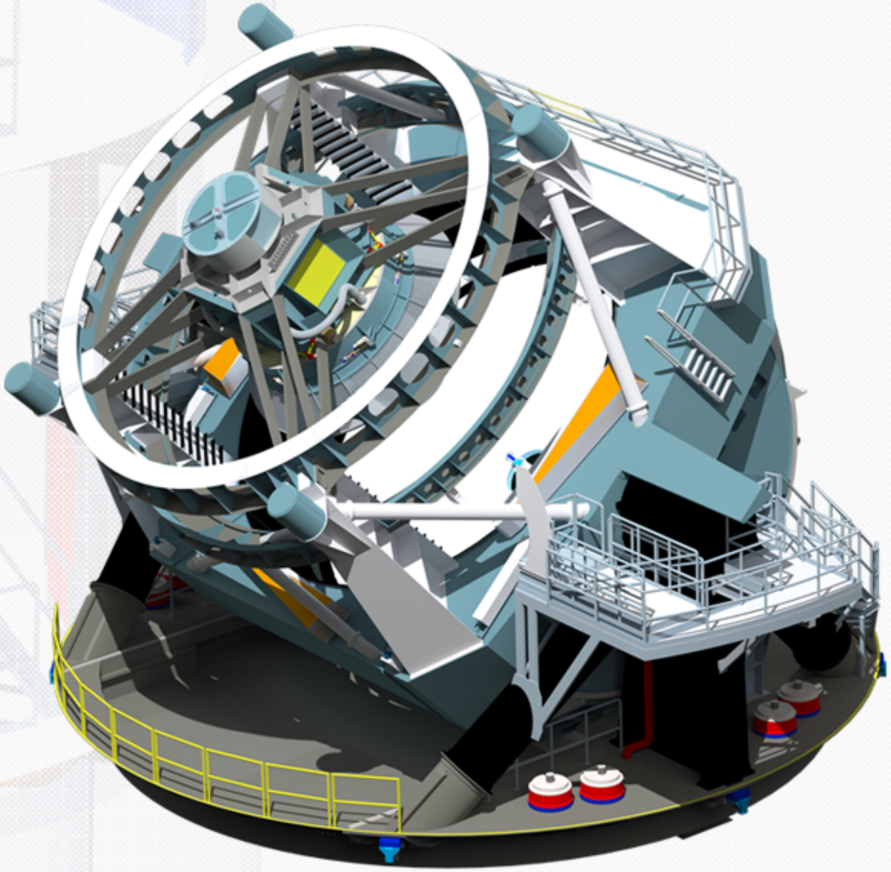
Basic afw and Data Butler Concepts

Kian-Tat Lim

Data Management Project Engineer and System Architect

October 5, 2015



DM Boot Camp

October 5, 2015

Why A Stack?

- Build a toolkit and framework
- Composable parts
- Standardized interfaces
- Scalability
- Portability

Task Abstraction

- Tasks (algorithms) operate on objects, not physical representations
- Escaping from binary program + file metaphor
- Allows Tasks to be invoked in many contexts
  - Command line, large-scale production, SUI/T, SDQA
- Allows data to be stored in many formats
  - Filesystem, tape, object store, database
  - Local, remote

What Is "afw"?

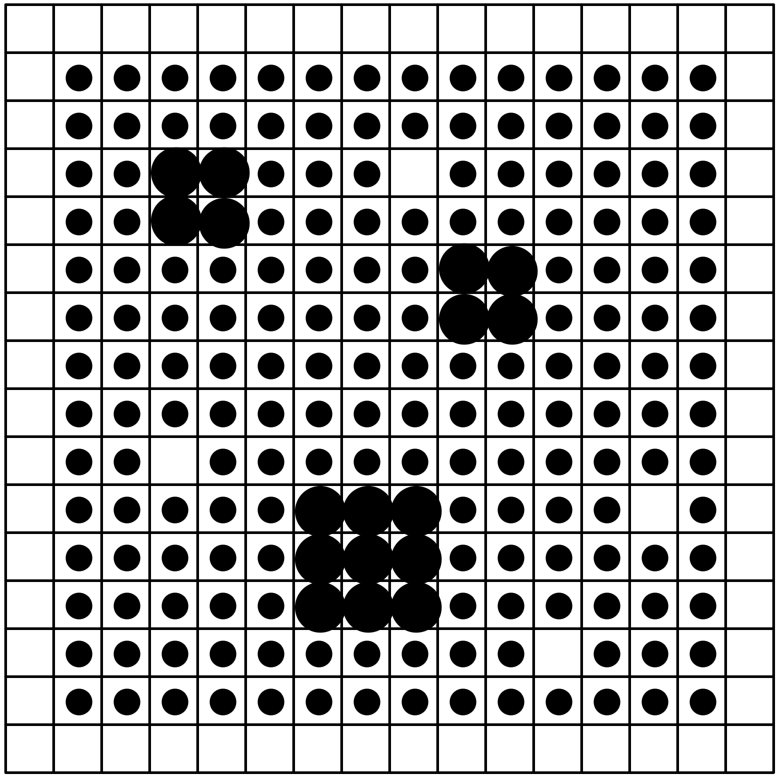
- "Applications framework"
- Applications is really Science Pipelines
- Framework is really a library or toolkit
- Therefore:
  - Library of astronomical image processing objects that can be used to build algorithms and pipelines

Basic afw Objects

- Image
- Exposure
- Skymap
- Table/Record

Image

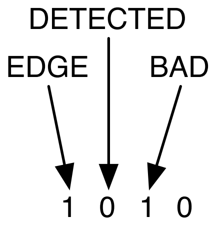
- 2-D rectangular array of pixels
- Pixel types: uint16, int32, float, double



Mask

- 2-D rectangular array of bit masks
- Each bit has a name associated with it
- Can add custom bits (up to capacity of bit mask)

1000	1000	1000	1000	1000	1000	1000	1000
1000	0000	0000	0000	0000	0000	0000	1000
1000	0000	0000	0000	0100	0100	0000	1000
1000	0000	0000	0000	0100	0100	0000	1000
1000	0000	0000	0000	0000	0000	0000	1000
1010	0010	0010	0000	0000	0000	0000	1000
1000	0000	0000	0000	0000	0000	0000	1000
1000	1000	1000	1000	1000	1000	1000	1000



DETECTED

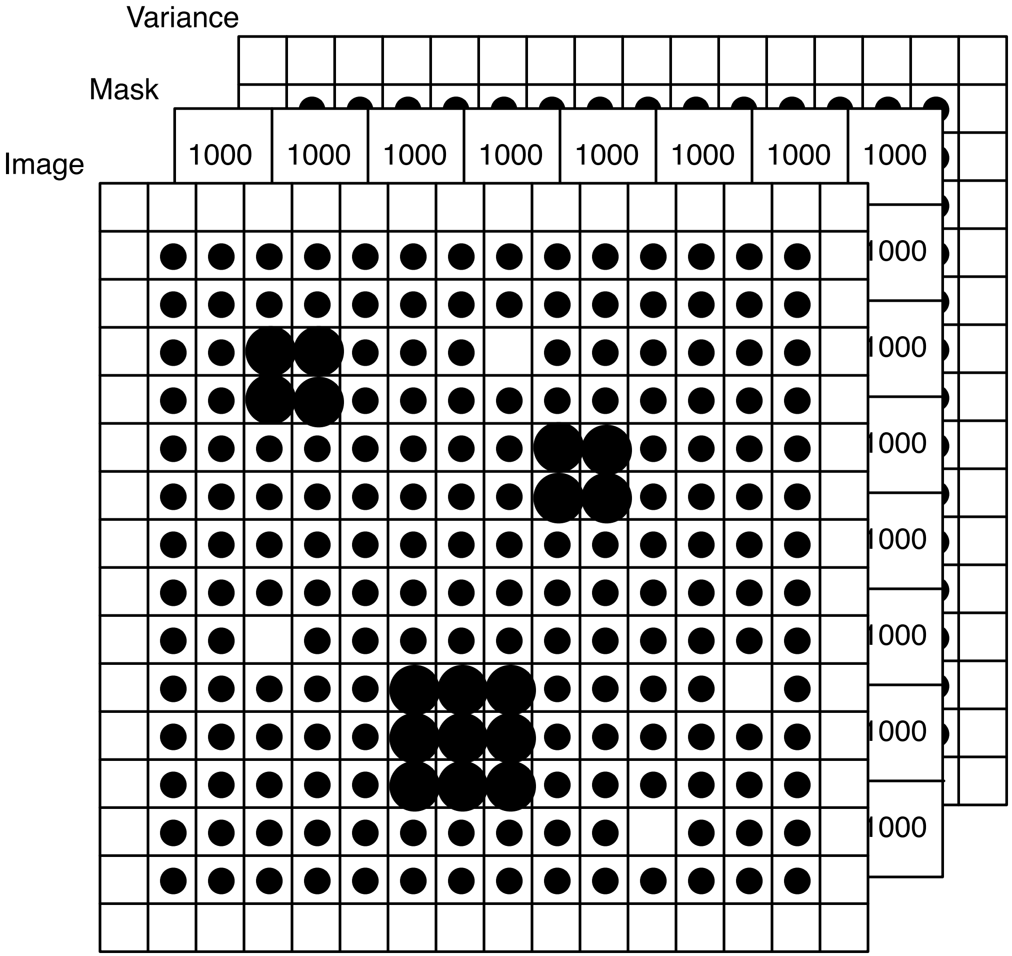
EDGE

BAD

1 0 1 0

Exposure

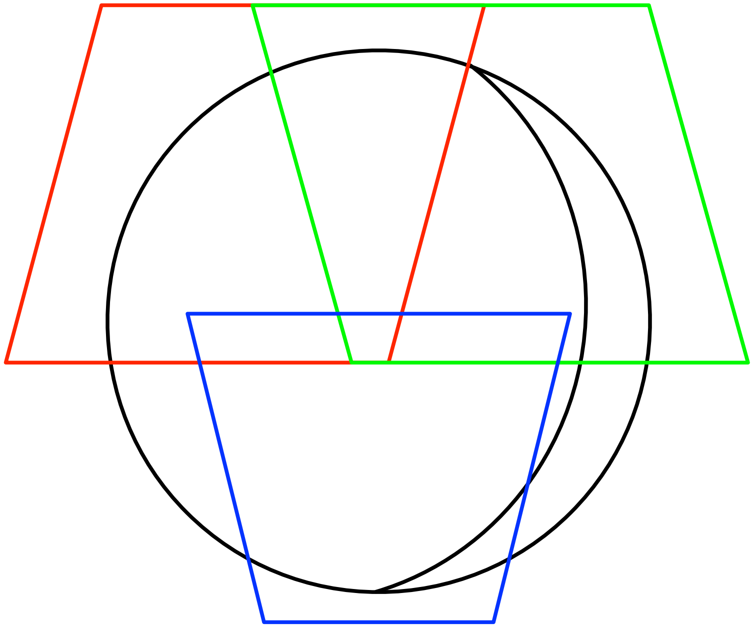
- Image, mask, variance plane, metadata about image



Key	Value

Skymap

- Overlapping tangent plane projections (*tracts*)
  - Dodecahedral, declination band, specific positions, rings, HEALPixes
- Each divided into overlapping rectangular segments (*patches*)
  - Inner regions exactly tile tract, overlap border *within* tract



Exposure Objects

- Wcs
- Psf
- Filter
- Calib
- Detector
- PropertyList

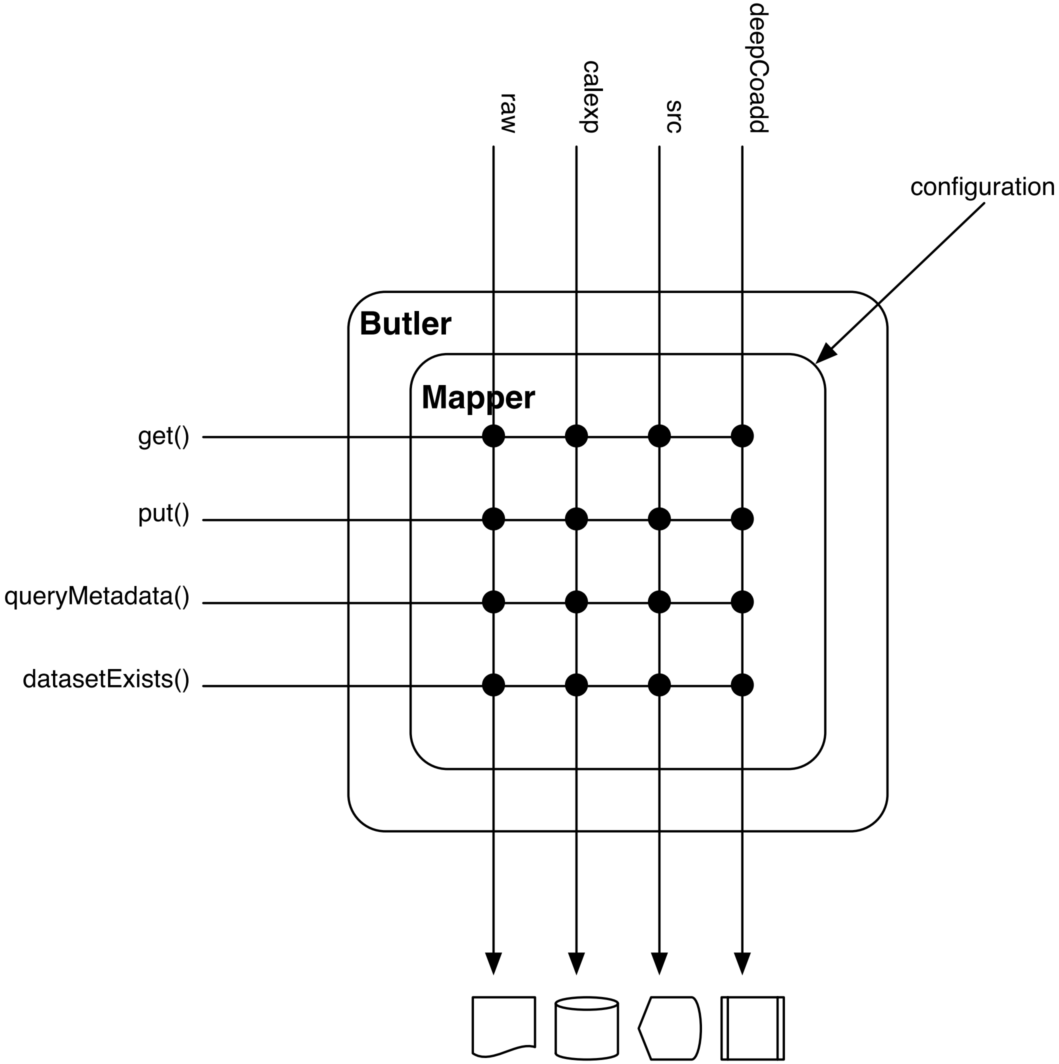
Table/Record

- Like a relational (SQL) table or a FITS table
- Columns of varying types, defined by a schema
- Rows (*records*) for different measurements

Data Butler

- Butler gets data, puts data, lists data
- Butler manages translations between physical formats and internal afw objects
- Butler does not do I/O itself -- it is a framework for allowing I/O to be configured

Butler Is A Router

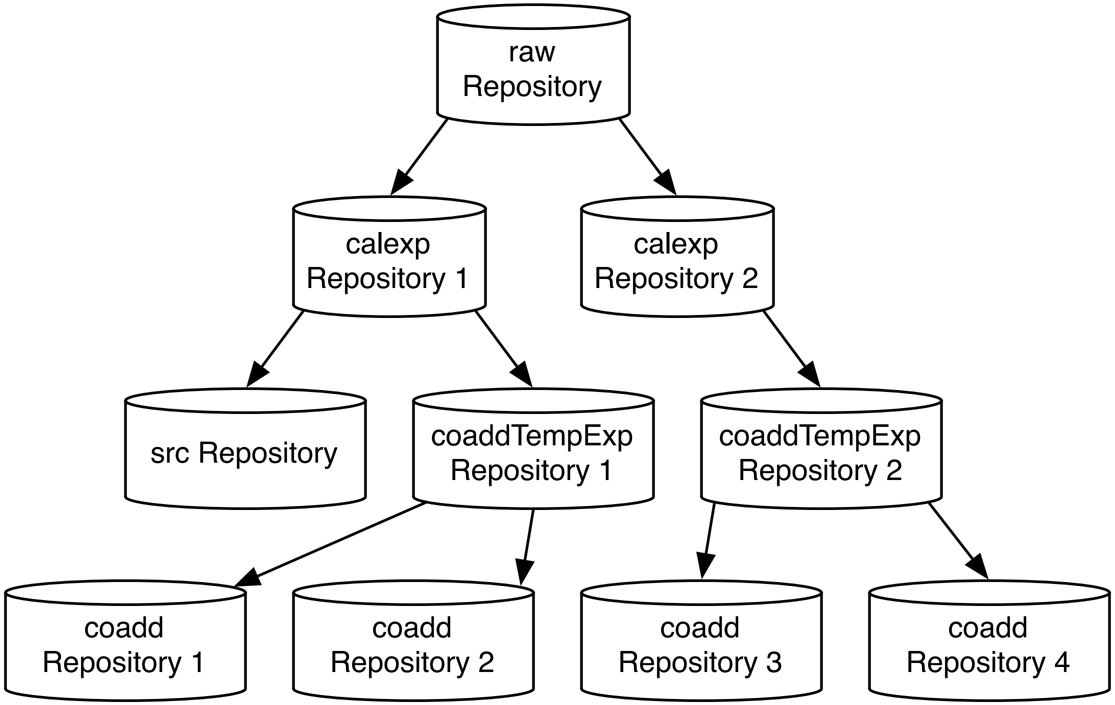


Repository

- A *repository* is conceptually a collection of datasets referenced by a path or URL
- In practice, it is a directory tree
- Repository structure defined by its *mapper* (and its mapper's configuration)
- Mapper selected by `_mapper` file in repository
- Mappers typically written by camera/observatory experts

Repository Chaining

- Output repositories are automatically chained to "parent" input repositories
  - Datasets from anywhere in the parent chain can be retrieved



Dataset

- Anything that can be persisted or retrieved
  - Some datasets can only be retrieved, possibly only persisted
- Range from numbers to images with metadata to entire catalogs of source measurements

Identifying Datasets

- Dataset type
  - Not Python or C++ type
  - Each mapper can have its own list of dataset types
- Data id
  - Key/value pairs
  - Each dataset type can have its own list of keys
  - Different mappers can require different keys
- Documentation of dataset types and data id keys is sparse
  - `obs_*` package policy files
- Data reference (for programming)
  - Combination of Butler and data id
  - Can be applied to multiple dataset types (if appropriate)

Common Data Id Keys

- visit, ccd, filter
- tract, patch

Partial Identification

- Only unique key/value pairs need be provided in a data id
  - Others are looked up in a *registry* database
  - Registry typically generated from raw input data
- "Rendezvous"
  - Information from one dataset can be used to look up another
  - Used to determine calibration images to apply

Common Dataset Types

- Images and Exposures
  - raw (postISRCCD, visitim, icSrc, icMatch)
  - calexp
  - coaddTempExp
  - deepCoadd_calexp (deepCoadd)
- Calibration Exposures
  - bias, dark, flat, fringe
- Detection and Measurement Tables
  - src (srcMatch), src_schema, transformed_src, transformed_src_schema

Other Dataset Types

- Identifiers
  - ccdExposureId
- Results
  - calexpBackground, psf, apCorr
- Processing Information
  - processCcd_config, processCcd_metadata

Summary

- *Tasks* use the *Butler* interface to operate on *datasets*, often *Images* or *Exposures*, identified by *data ids* within *repositories*, producing new datasets, often new Exposures or *Tables*, in an output repository chained to the input repository. The Butler uses a camera-specific *Mapper* to define the repository structure, available datasets, and data id keys.