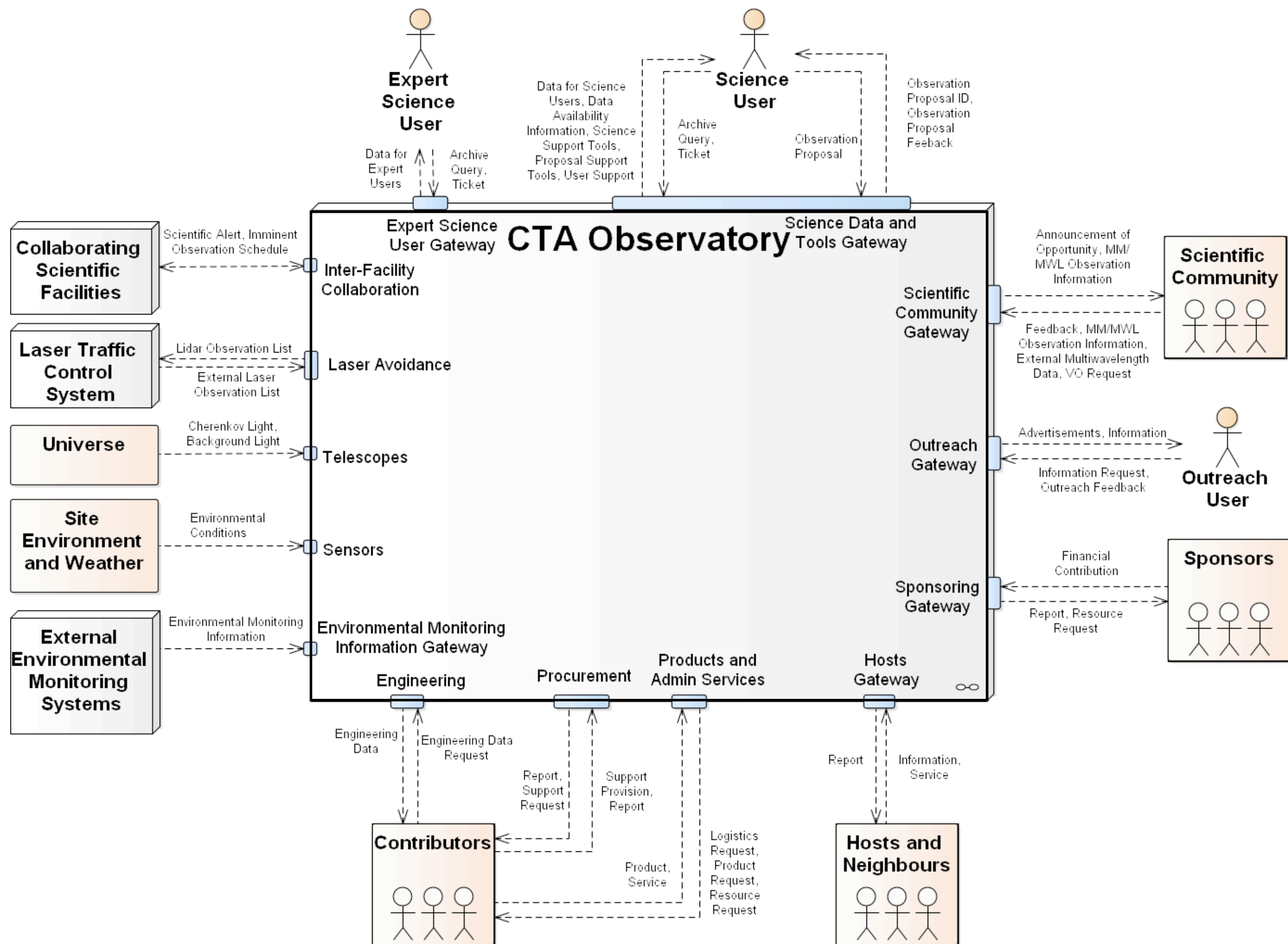


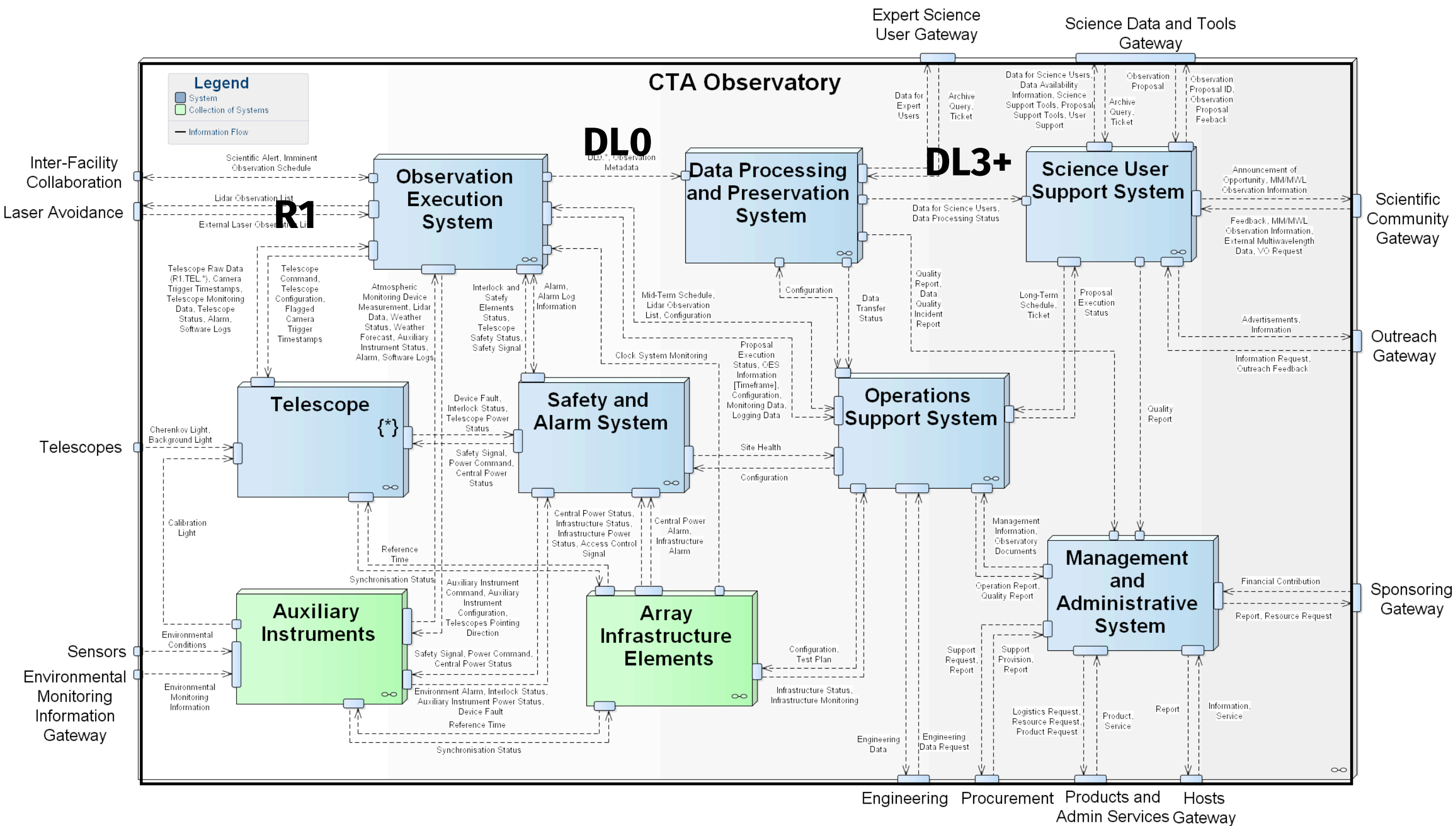


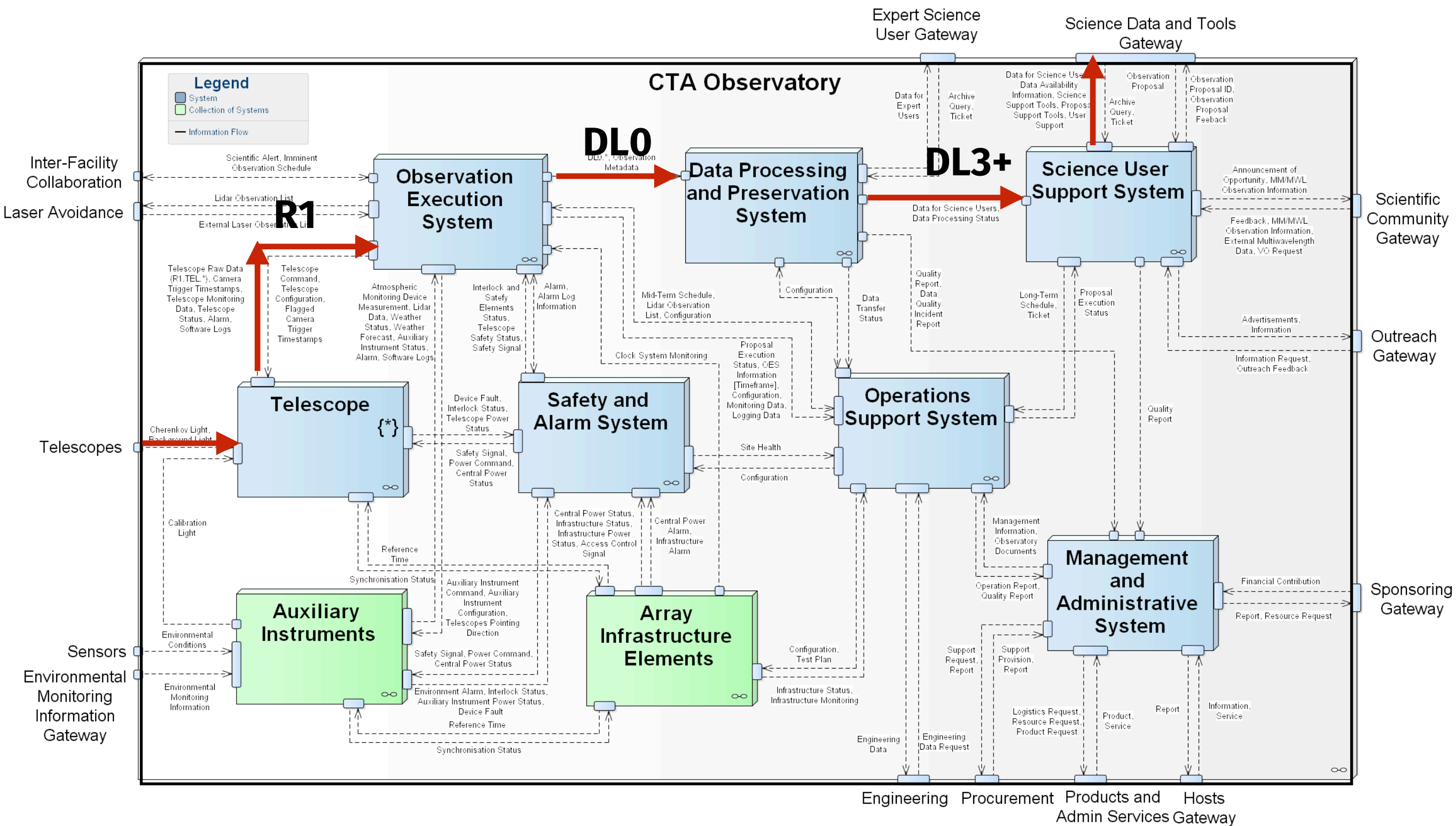
DPPS & OES

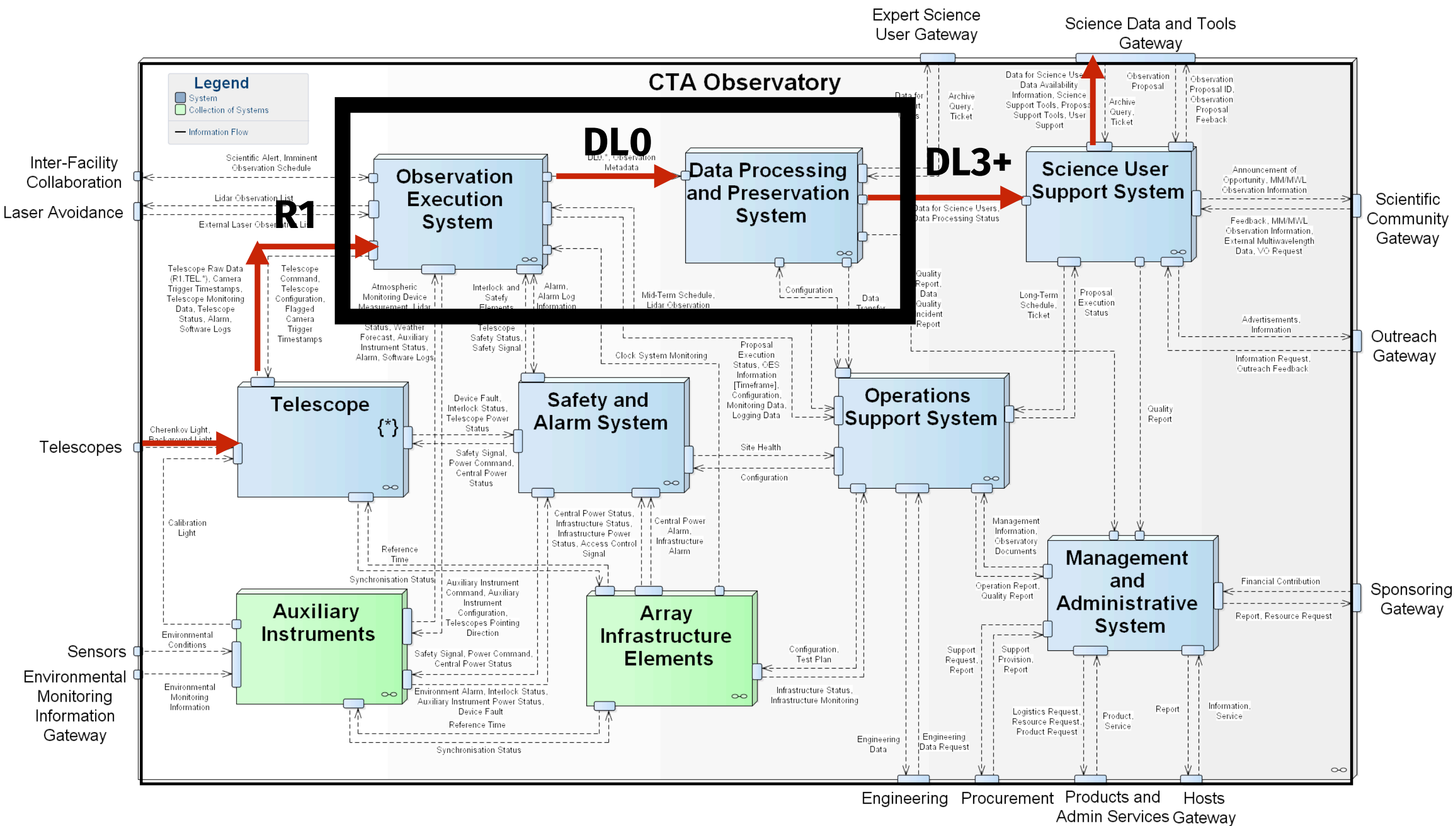
Data Processing and Observation Execution

Karl Kosack

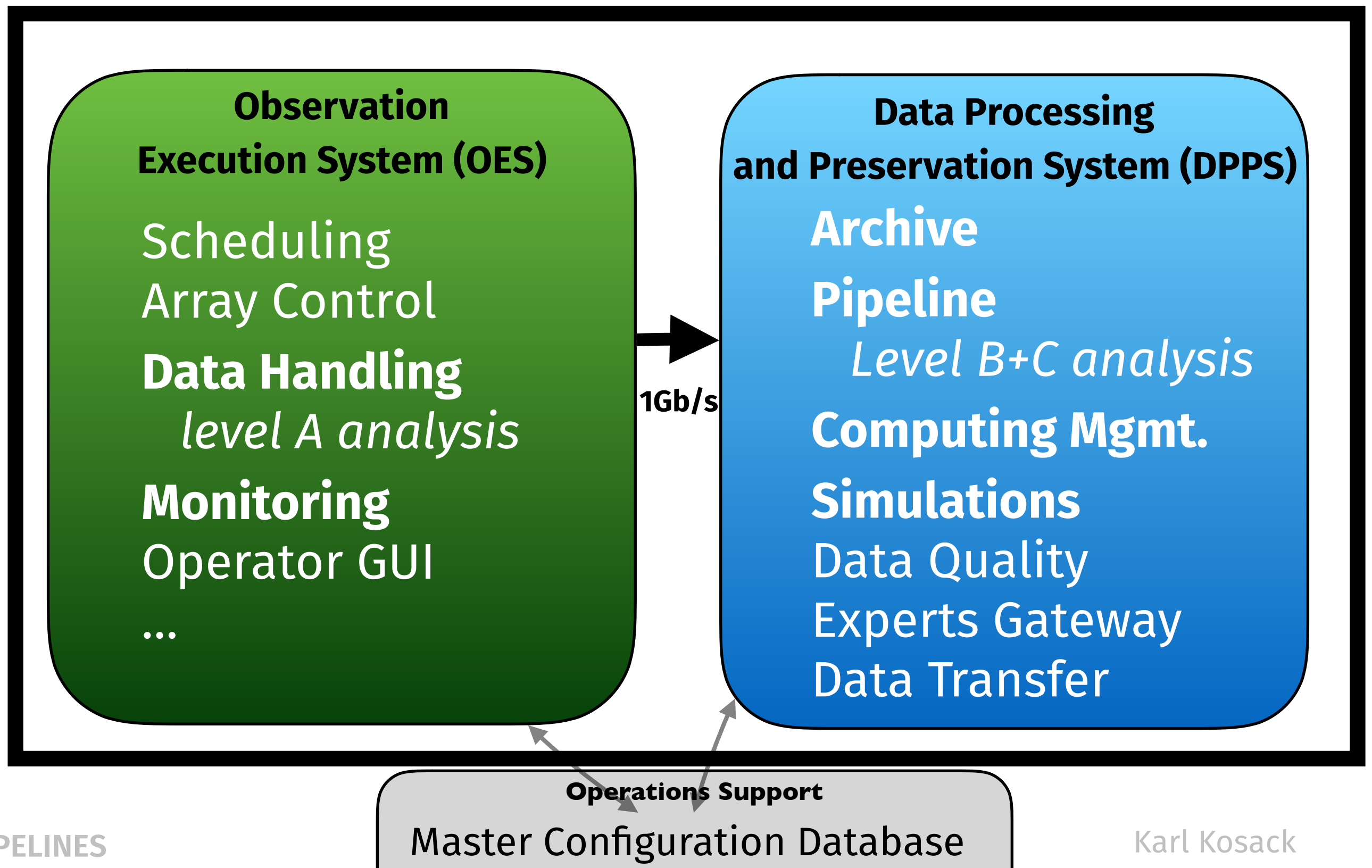








Internal Data-related systems



Data Levels: amount of processing

MCs are somewhere here right now



R0 (*raw low-level*) camera data transmitted from telescope to central servers. R0 content and format is internal to each camera and is specified and coordinated between individual camera teams.

R1 (*raw common*) data output by an individual camera functional unit to the camera DAQ functional unit. This is the first level of data seen by the ACTL system and is therefore as common as possible between all cameras/hardware. Exceptionally, some R1 data may be stored for engineering purposes.

Pipeline starts here
Will need to eventually produce data here to be compatible with "real" CTA data



DL0 (*raw archived*) all archival data from the data acquisition hardware/software. This is the first level of data that are **stored in the bulk archive**. This includes both camera event data and technical data from other subsystems, such as non-camera devices or software.

Reconstructed Events
(many reconstructions and parameters, no more telescopes)



DL1 (*processed*) processed DL0 data that may still include some TEL data and parameters derived from them. For example this includes calibrated image charge, Hillas parameters, and a usable telescope pattern. This is only optionally stored in the archive.

DL2 (*reconstructed*) reconstructed shower parameters such as energy, direction, particle ID, and related signal discrimination parameters. At this point, no TEL information is stored. For each event this information may be repeated for multiple reconstruction and discrimination methods. This is only optionally stored in the archive. At this point, telescope-wise info is generally dropped.

Science Data:
Classified Events
(final reconstruction),
IRFs

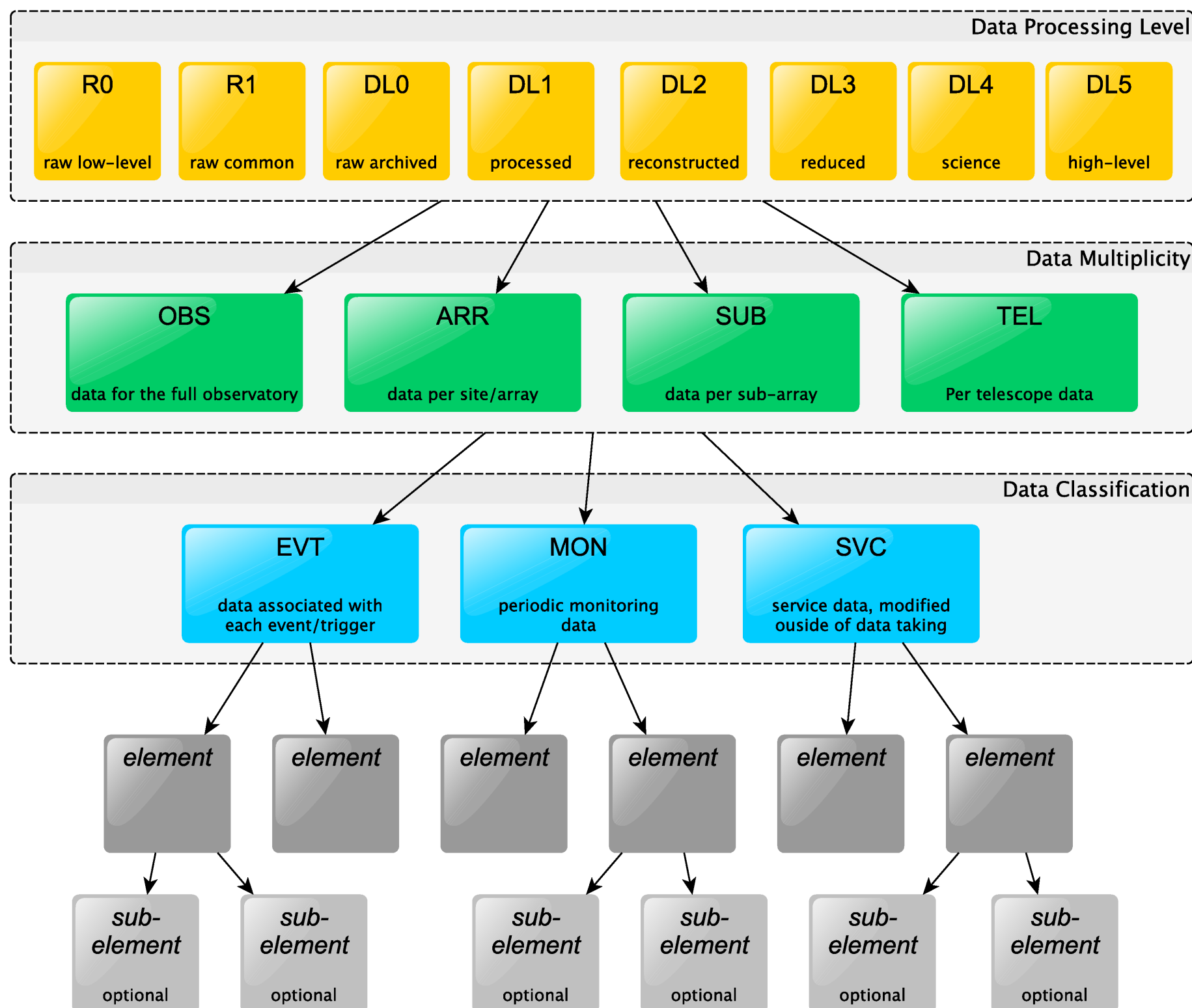


DL3 (*reduced*) Sets of selected (e.g. gamma-ray candidates, electron candidates, selected hadron candidates, etc.) events with a single final set of reconstruction and discrimination parameters, along with associated instrumental response characterizations and any technical data needed for science analysis.

DL4 (*science*) binned data products like spectra, sky maps, or light curves, along with associated data (source models, fit results, etc).

DL5 (*high-level*) high-level or "legacy" observatory data, such as CTA survey sky maps or the CTA source catalog.

Data Naming



For example:

DLO.TEL.EVT:
all eventwise
data from
cameras

DLO.SUB.EVT:
central trigger
data

DLO.TEL.MON:
telescope
monitoring data

DLO.TEL.SVC:
telescope
configuration/
calibration
tables

DL0.*.EVT Data Format



Document under prep:

- ▶ define the minimal R1.*.EVT schema
- ▶ define the *maximal* DL0.*.EVT schema (limited by link)
- ▶ call for data format prototypes
- ▶ define evaluation and down-selection procedure

Associated data volume simulator notebook

Name	Type	Shape	Unit	Description
event_id	uint64	(1)	-	The event ID assigned by the SWAT (primary key)
trig_time_s	uint32	(1)	s (TAI)	telescope trigger time, seconds since reference
trig_time_qns	uint32	(1)	0.25 ns	telescope trigger time, quarter-nanoseconds since trig_time_s
trig_type	uint8	(1)	-	trigger type identifier, encoding the trigger class and sub-class
image	int16	(N_{pix})	DC	the time-integrated image
pix_status	uint8	(N_{pix})	-	See definition below (§2.4)
trace	int16	(K_{tr}, N_{samp})	DC	traces for all pixels that have the DVR flag (bit 2) high in pix_status

Note that this is about the limit of what we can store per event and still transfer data off site!

Some comparison criteria:



Maintenance/ longevity

Criterion
Format is open standard?
Documentation: complete API docs?
Is the format documented sufficiently to re-implement a reader if the original API is lost?
When Created?
Standard defined by whom?
Used outside of CTA?
Currently maintained-by?
Long-term maintenance plan?
Multiple implementations?
Manpower needed to to change computing environment/compiler
Dependencies

Data and Schema flexibility

Criterion
Schema defined how?
Self-descriptive?
Supports required data types
Can store per-item units/scales
table-like API
Support and performance of variable-length fields
Supports multi-dimensional array types
Supports nested data types
Supports rich meta-data/headers
Supports arbitrary meta-data (no schema needed)
Can add/change meta-data in existing file?
Multiple datasets in a single file?
Supports schema evolution
Manpower needed to update data model

Technical and Architecture

Criterion
Machine architecture independence?
Multi-compiler support
Multi-platform support
API Language bindings
Detection of data corruption
Recovery from data corruption
Endianness support
Efficient row (event-wise) data access?
Efficient column (one data item for all events) access?
Parallel read access?
Parallel write access?
Takes advantage of data pre-fetching?
Data are loaded contiguously in memory?
Cache friendliness?
Support for vectorization (SIMD)?
Requires all events in file to be in memory?

Performance

Criterion
Column-wise chunk size (N-rows)
Write performance, Row-Wise (MB/s)
Write performance, Column-Wise (MB/s)
Read performance, Row-wise (MB/s)
Read performance, Column-wise (MB/s)
Compression: internal or external?
Compression: ratio achieved?
Compression: recommended method?
Peak Memory Usage:

ctapipe status



Trace Integration:

- ▶ Support so far 4 methods (*NeighborPeakIntegrator* as default)
- ▶ Missing MARS/EventDisplay style *GradientPredictionIntegrator*: to be implemented (see #

Image Processing:

- ▶ Hillas (code recently cleaned up, sped up, but still needs some refactoring)
- ▶ Tail Cuts: now supports standard + MARS definition, support for optimized thresholds per camera
- ▶ Wavelet Cleaning: 2D method working and ready to be merged into standard pipeline (waiting for better implementation of backend code). 3D method to be done.

Reconstruction:

- ▶ Plane-Intersection: implemented and working
- ▶ ImPACT template model: implemented and working, but needs template generation for full production

General Framework:

- ▶ Mostly ok, but still a lot of (re)design needed. Small dev meeting / telecon soon. (restarting normal telecons)

Upcoming Refactorings



IO sources (need so far to support 4+ file formats)

- ▶ EventFileReader and x_event_source() being merged into common interface (see PR#613)
- ▶ future: support both single-event, and table of events, for different use cases

Merging in of Pipeline steering scripts and related classes

- ▶ EventPreparer (helper to produce DL1 outputs)
- ▶ scripts for:
 - reconstruction,
 - training of energy and classification
 - cut optimization + IRF production + sensitivity and diagnostics

Unification of DL2 outputs

- ▶ currently "proprietary" HDF5 tables (Tino), just need some work to make compliant with the CTA standards (add event_id, rename some columns, etc.)

More to come, as CTA standards and Architecture are developed...