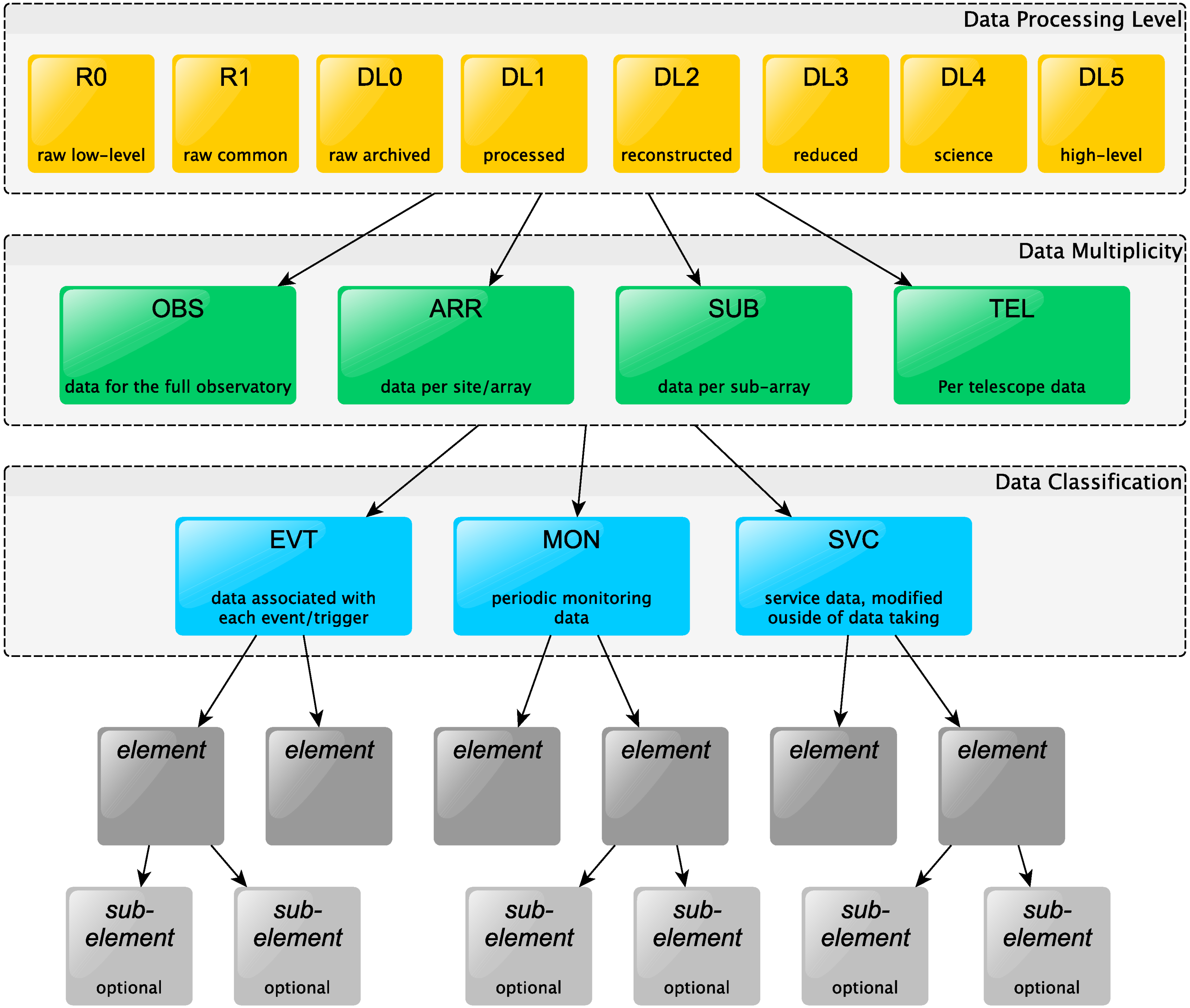
IRFS FOR CTA SCIENCE

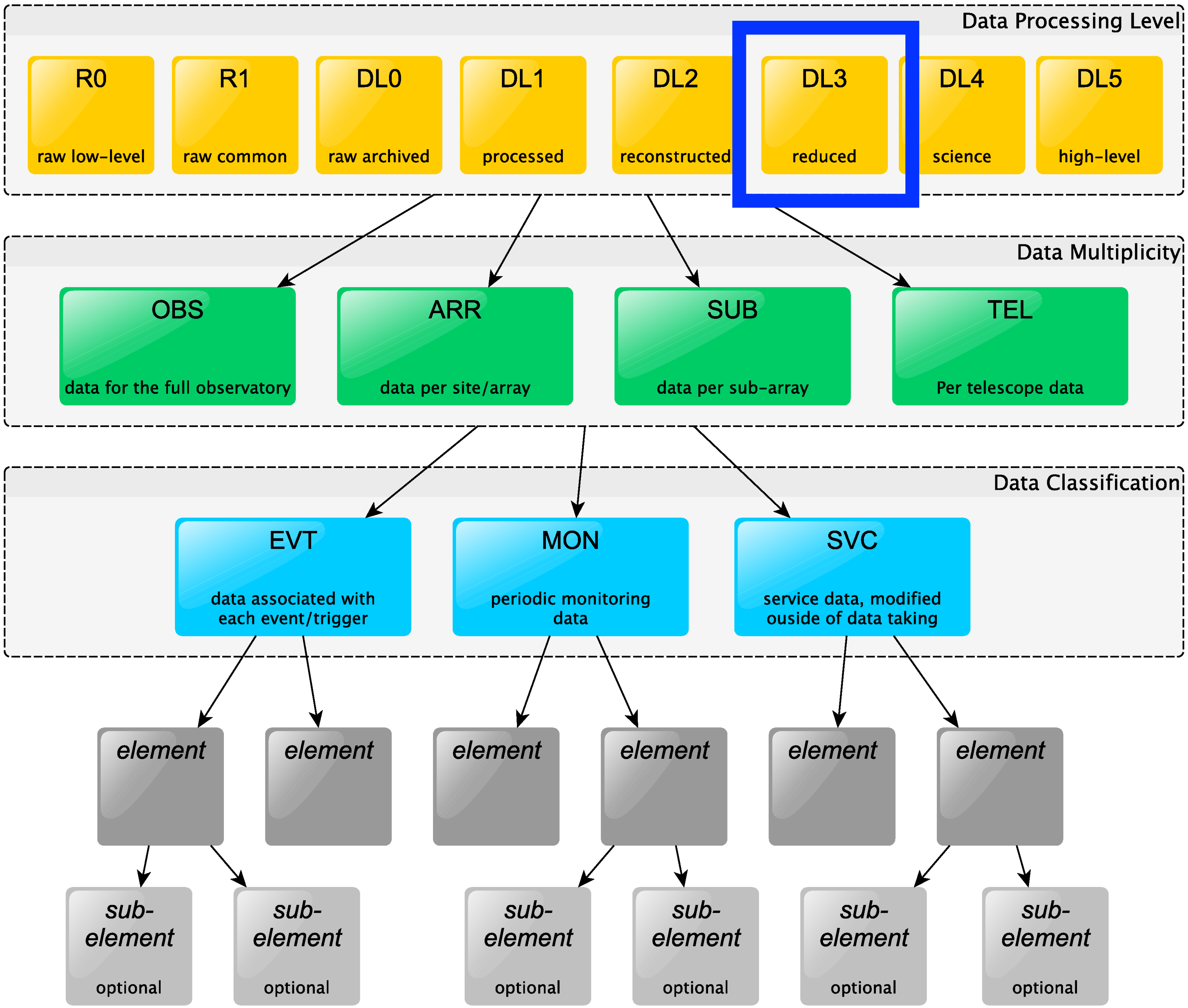
Karl Kosack
DPPS Coordinator, CTAO
CEA Saclay

Gammapy Meeting, Paris Feb 2018

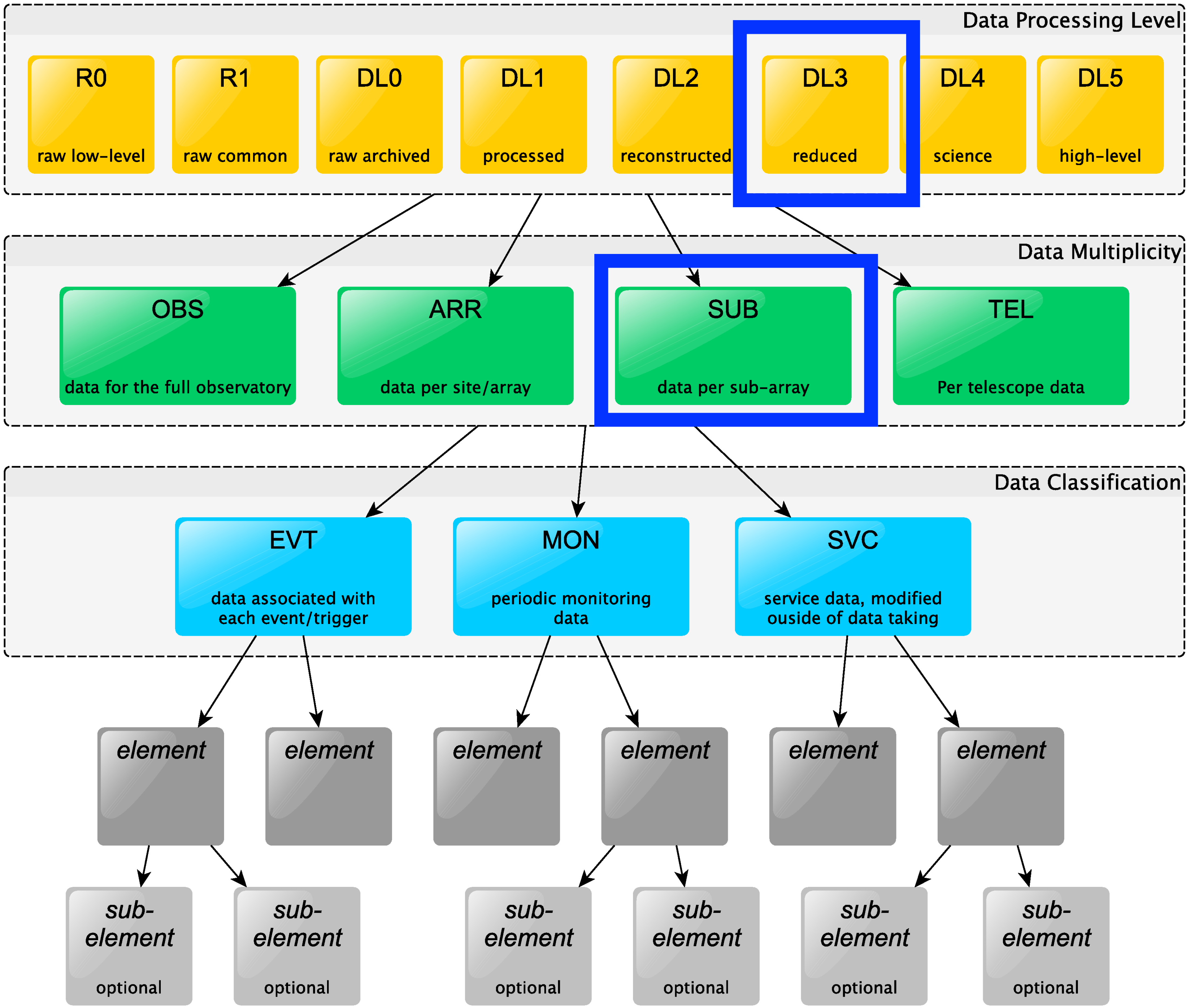
Recall: Data Naming in CTA



Recall: Data Naming in CTA

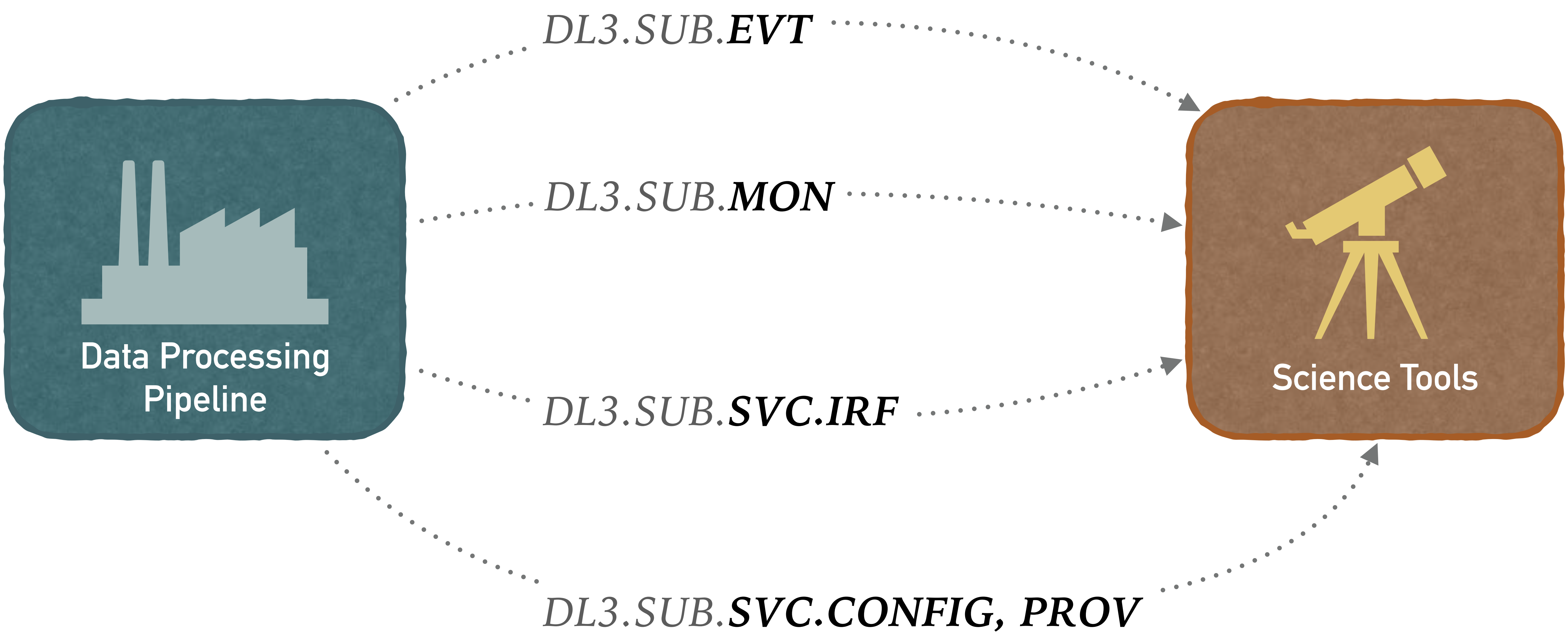


Recall: Data Naming in CTA



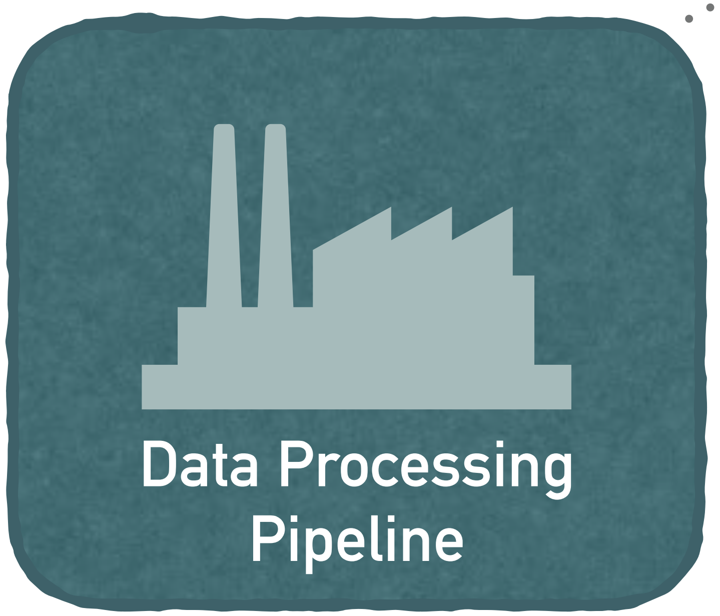
Pipeline to Science Tools Interface

For a given *obs_id*, we will produce:



Pipeline to Science Tools Interface

For a given *obs_id*, we will produce:



*DL3.SUB.**EVT***

***event list:** RA, Dec, time,*

[particle_class, event_class, …]

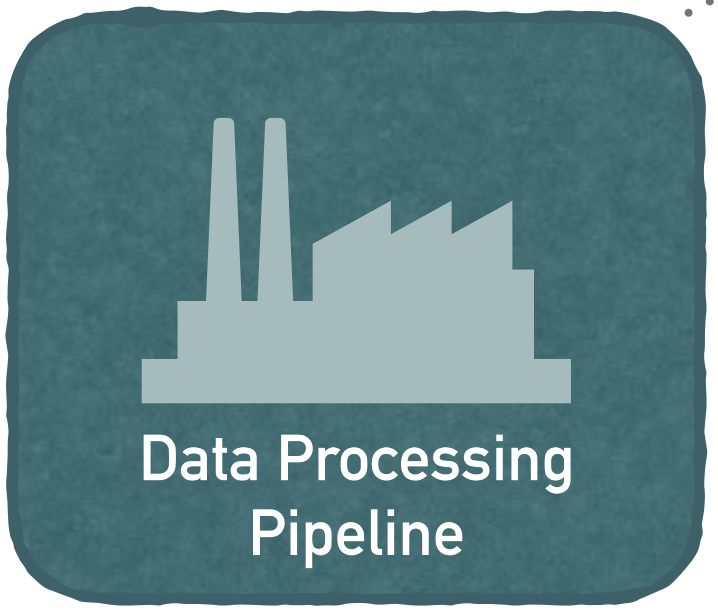
*DL3.SUB.**MON***

*DL3.SUB.**SVC.IRF***

*DL3.SUB.**SVC.CONFIG, PROV***

Pipeline to Science Tools Interface

For a given *obs_id*, we will produce:



*DL3.SUB.**EVT***
***event list:** RA, Dec, time,*
[particle_class, event_class, …]

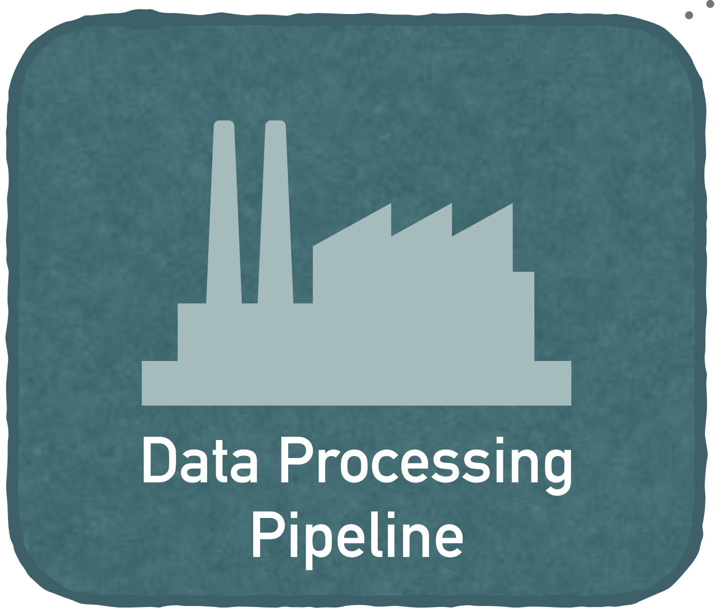
*DL3.SUB.**MON***
Data Quality, Weather,
***Pointing, etc.** as func of time*

*DL3.SUB.**SVC.IRF***

*DL3.SUB.**SVC.CONFIG, PROV***

Pipeline to Science Tools Interface

For a given *obs_id*, we will produce:



*DL3.SUB.**EVT***
***event list:** RA, Dec, time,*
[particle_class, event_class, …]

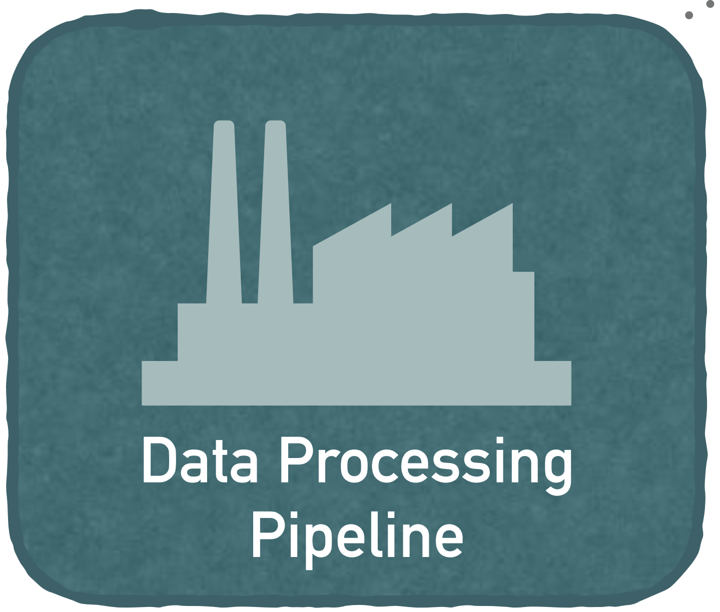
*DL3.SUB.**MON***
Data Quality, Weather,
***Pointing, etc.** as func of time*

*DL3.SUB.**SVC.IRF***
Effective Area, PSF, Energy
Migration, etc.

*DL3.SUB.**SVC.CONFIG, PROV***

Pipeline to Science Tools Interface

For a given *obs_id*, we will produce:



*DL3.SUB.**EVT***
event list: *RA, Dec, time, [particle_class, event_class, …]*

*DL3.SUB.**MON***
Data Quality, Weather, Pointing, etc. *as func of time*

*DL3.SUB.**SVC.IRF***
Effective Area, PSF, Energy Migration, etc.

*DL3.SUB.**SVC.CONFIG, PROV***
Observation and Subarray configuration, provenance data

IRF Parameters and Dependencies

IRF Parameters and Dependencies

Pipeline:

- Camera Calibration Algorithm
- Waveform Integration Method
- Image Cleaning Method
- Reconstruction Algorithm
- Discrimination algorithm
- Discrimination cuts
- Classification Algorithm
- Classification Result
- Atmosphere & Array calibration

IRF Parameters and Dependencies

Pipeline:

- Camera Calibration Algorithm
- Waveform Integration Method
- Image Cleaning Method
- Reconstruction Algorithm
- Discrimination algorithm
- Discrimination cuts
- Classification Algorithm
- Classification Result
- Atmosphere & Array calibration

Observation Parameters:

- (alt, az) of pointing
- (r, ϕ) position in FOV
- NSB level
- observation config (subarray, etc.)

IRF Parameters and Dependencies

Pipeline:

- Camera Calibration Algorithm
- Waveform Integration Method
- Image Cleaning Method
- Reconstruction Algorithm
- Discrimination algorithm
- Discrimination cuts
- Classification Algorithm
- Classification Result
- Atmosphere & Array calibration

Observation Parameters:

- (alt, az) of pointing
- (r, ϕ) position in FOV
- NSB level
- observation config (subarray, etc.)

Event Parameters

- Reconstruction class / type
  - a 3-LST event has different reconstruction than a 3-SST event
  - a contained event (inside array) reconstructs better than an uncontained event, etc.
- Particle class (e-, gamma, …)

Obs-wise vs gridded Simulations

Observation-wise Simulations

- fold as much knowledge of atmosphere and instrument into monte-carlo, and produce an IRF per observation
- not so simple to get enough statistics, but generally lowers systematics

Gridded Simulations:

- systematics depend on dimensionality and phase-space binning & interpolation
- difficulty or impossible to include systematic effects like broken pixels, or camera gradients
- statistics generally higher, since each bin can be reused for multiple observations

We will likely have both!

- on-site: can only support gridded
- best solution may depend on science case: high stats vs low systematics…
- *Not clear if Science Tools need to differentiate between the two…*

Obs-wise vs gridded Simulations

Observation-wise Simulations

- fold as much knowledge of atmosphere and instrument into monte-carlo, and produce an IRF per observation
- not so simple to get enough statistics, but

Gridded Simulations:

- systematics depend on dimensionality and
- difficulty or impossible to include systema camera gradients
- statistics generally higher, since each bin observations

We will likely have both!

- on-site: can only support gridded
- best solution may depend on science case: high stats vs low systematics…
- *Not clear if Science Tools need to differentiate between the two…*

Keep in mind:

All the simulations we produce now are gridded very coarsely (not sufficient for a final analysis)

The work to produce an grid or obs-wise IRF is *non-trivial and requires vast computing resources*

Cannot support every science-tools user wish! *(changing cuts, etc) → only a set of fixed configurations*

IRF-related tasks (for ASWG)

IRF-related tasks (for ASWG)

What parameters should be:

- dimensions of the IRF?
- implicit in the Monte-Carlo simulation setup?
- corrections to events or IRFs? and where are they applied?

IRF-related tasks (for ASWG)

What parameters should be:

- dimensions of the IRF?
- implicit in the Monte-Carlo simulation setup?
- corrections to events or IRFs? and where are they applied?

How are IRFs parameterized?

- binned profile? What binning?
  - after some smoothing? Fitting and resampling?
  - how to interpolate or extrapolate?
  - what to do with missing or low-stats / high-uncertainty values?
- functional representation?
- How are uncertainties (stat + sys) stored and used?

IRF-related tasks (for ASWG)

What parameters should be:

- dimensions of the IRF?
- implicit in the Monte-Carlo simulation setup?
- corrections to events or IRFs? and where are they applied?

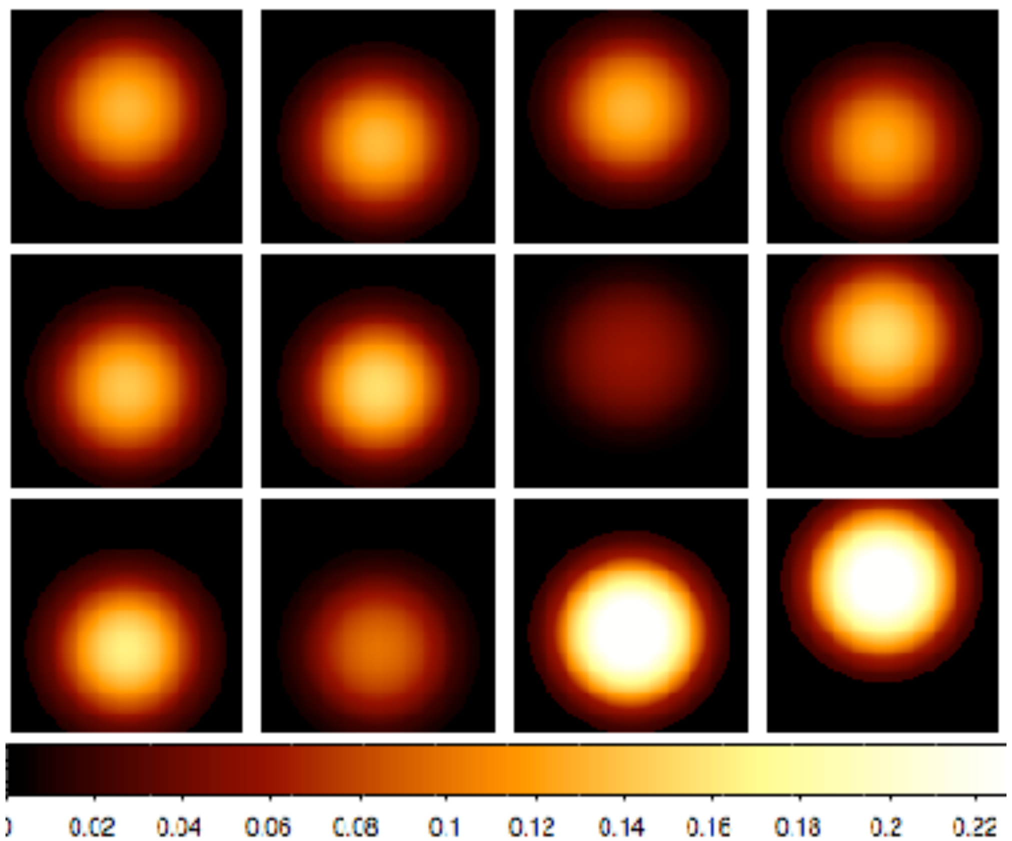
How are IRFs parameterized?

- binned profile? What binning?
    - after some smoothing? Fitting and resampling?
    - how to interpolate or extrapolate?
    - what to do with missing or low-stats / high-uncertainty values?
- functional representation?
- How are uncertainties (stat + sys) stored and used?

How are IRFs factorized?

- *Now*: decouple ($A_{eff} \cdot \varepsilon$), PSF, $M(E_{tr} \rightarrow E_{reco})$, R_{bg}
- Are there correlations between these? Cross-terms?
- Can we really provide R_{bg} per observation or just the background acceptance *shape*? (i.e. *can we include true exposure in the IRF, and require the science tools to calculate it (as now)*?
- We could multiply many of these quantities together and e.g. store Aeff, ε, and M together in a single matrix (normalization of M is such that integration gives back Aeff, for example) → then we have just 3 matrices: **M**' + PSF + R_{bg}

A note on exposures



The exposure normalized to the number of background events, for a set of HESS Crab runs, ***all of exactly 28 minutes!***

in IACTs, the exposure depends not only on the optics and time, but on the *atmosphere,* which changes in time

- zenith angle and azimuth : changes depth of atmosphere, usually included as a dimension in the IRF
- atmosphere profile: can be folded into simulation, particularly if run-wise
- seeing / transparency : can be estimated, but no idea yet if this is robust enough for a global correction?
- aerosol layers: measured by LIDAR and FRAM, could be corrected, but where and how reliably?

A note on exposures

† in my experience, both ctools and gammapy ran out of memory when I tried a run-wise fit for a large dataset like the Galactic Center survey…

A note on exposures

Where should we apply exposure correction?

- so far: nowhere (data challenge data has no atmosphere variation)
- Options are:
  - **to IRFs** (need detailed and robust way to estimate it that is *not* from the background rate, since that is analysis-dependent - e.g. you need to know in advance where sources are)
  - **in the Science Tools:** exposure correction is calculated using the background model or off-source event region, as with current instruments

† in my experience, both ctools and gammapy ran out of memory when I tried a run-wise fit for a large dataset like the Galactic Center survey…

A note on exposures

Where should we apply exposure correction?

- so far: nowhere (data challenge data has no atmosphere variation)
- Options are:
  - ➤ **to IRFs** (need detailed and robust way to estimate it that is *not* from the background rate, since that is analysis-dependent - e.g. you need to know in advance where sources are)
  - ➤ **in the Science Tools:** exposure correction is calculated using the background model or off-source event region, as with current instruments

Drawbacks to each:

- **IRF Correction**: clear need for *Time Intervals* that are shorter than observations! Science tools would need to take this into account and associate IRF or correction factor per TI, or we need to break down observation files into one per TI (could be a lot).

  We don't (yet) know the systematics of applying an analysis-independent correction, even if it is being promised by the CCF group.

- **In Science Tools**: you *cannot* simply sum all runs together and then model the background and sources: need to do it runwise, with exposure normalization as a free parameter *per run*. **This is not how most 1DC analyses were done!†**

† in my experience, both ctools and gammapy ran out of memory when I tried a run-wise fit for a large dataset like the Galactic Center survey…

Science Tools IRF Considerations

IRF data format:

- storage of N-dim data (sparse? fixed or variable bins? …)
- detailed provenance (ensure we are using the right IRF for this event list, etc)
- per *file? Time Interval/GTI?*
- What file format? FITS (de-facto, preferred for now), but other options exist (HDF5) with advantages of better metadata and compression.

IRF API:

- expect changes to format, factorization, and parameterization with ***each pipeline pass*** (≈yearly)
- design software to be able to adapt to these changes and ensure backward compatibility (reproduce old results)
- try as much as possible to make the treatment of IRFs abstract.
- Don't expect everything to be in the IRFs… (at least not for early data). Some corrections like exposure may be necessary in the science tools until a method to correct them independent of background rate is developed