ASTRI Astrofisica con Specchi a Tecnologia Replicante Italiana

cta cherenkov telescope array

INAF ISTITUTO NAZIONALE DI ASTROFISICA NATIONAL INSTITUTE FOR ASTROPHYSICS

Universidade de São Paulo
Instituto de Astronomia, Geofisica e Ciencias Atmosferica

MIUR

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

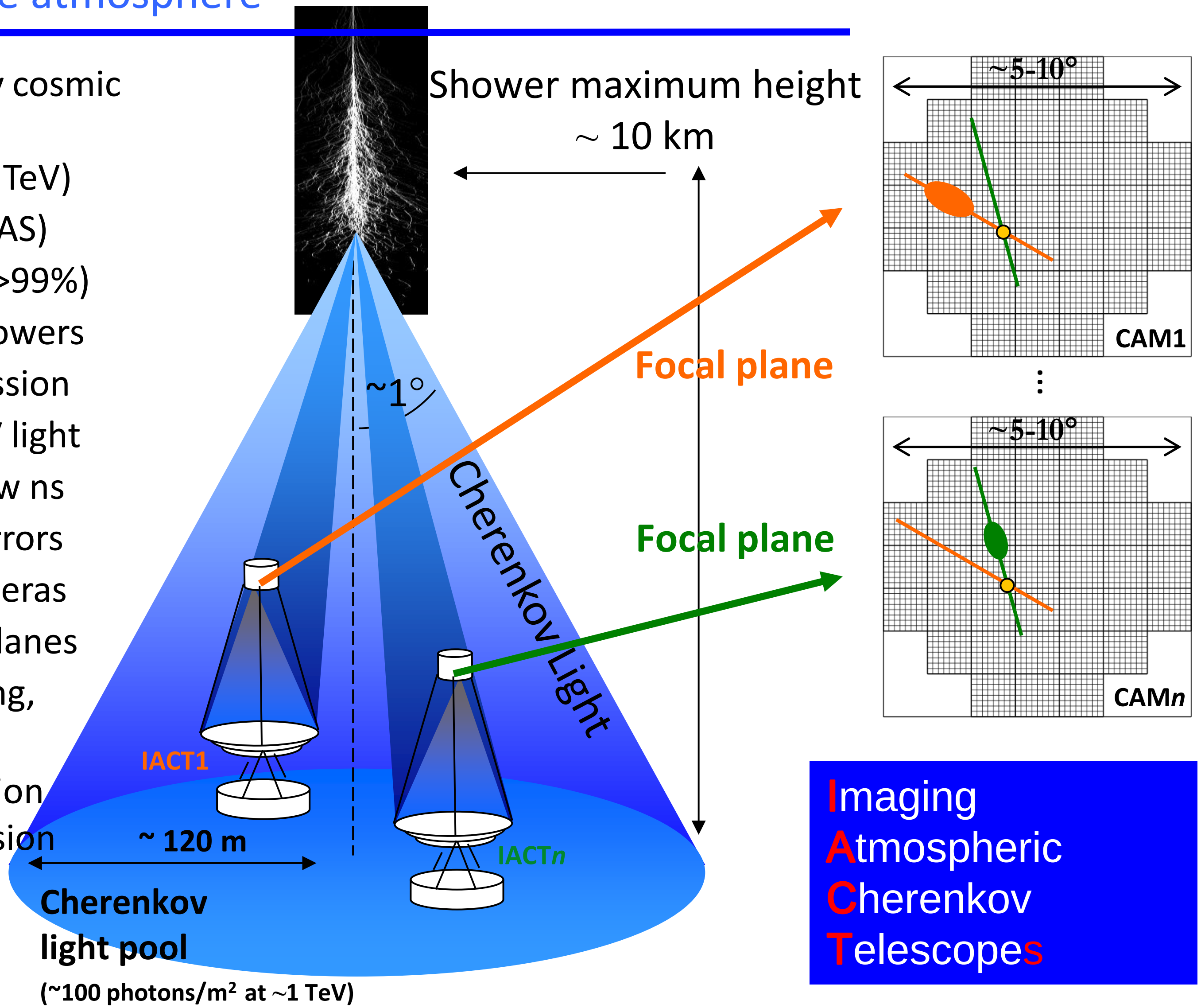
ASTRI data reduction software on low-power and parallel architectures

Michele Mastropietro, S. Lombardi, D. Bastieri, L. A. Antonelli
INAF - Osservatorio Astronomico di Roma & UniPD
for the ASTRI Collaboration and the CTA Consortium

- Imaging Atmospheric Cherenkov Telescopes (IACTs)
  - the Cherenkov Telescope Array (CTA)
  - the ASTRI project in the framework of CTA
- The ASTRI data reconstruction and scientific analysis software
- Algorithms parallelization
- GPU and low-power platform benchmarks using different metrics

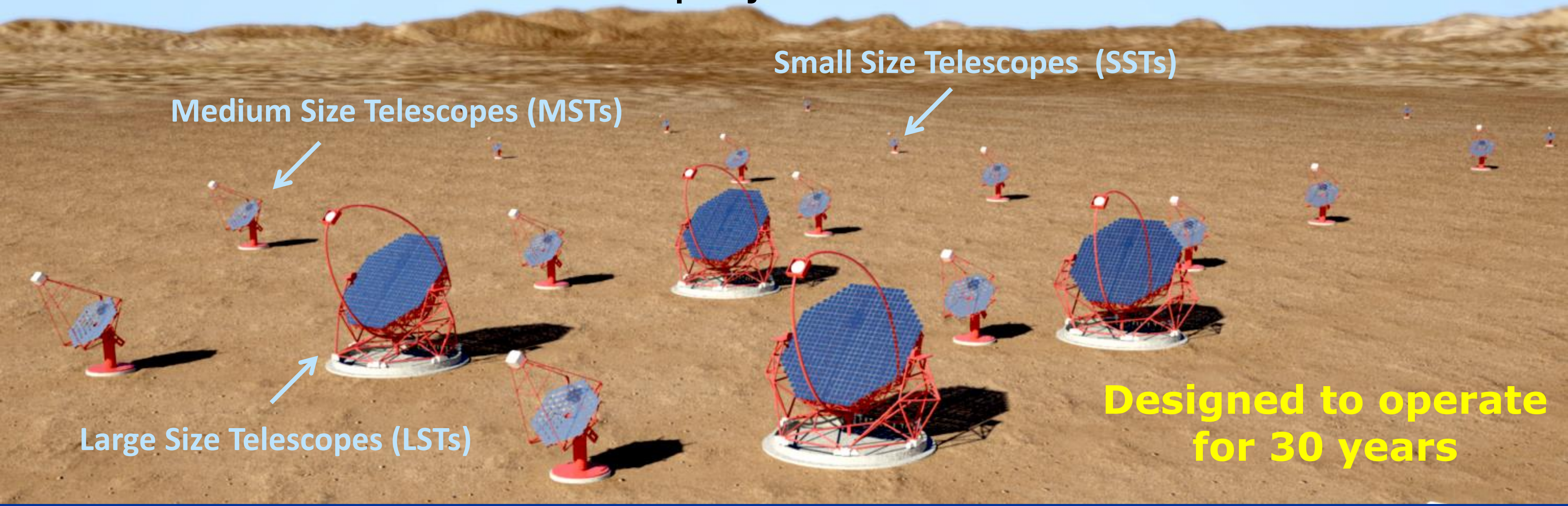
Top of the atmosphere

- Very High Energy primary cosmic rays (p, α, $e^{\pm}$, γ, ...)
  (CTA range: 20 GeV - 300 TeV)
- Extended Air Showers (EAS)
  - Hadronic showers (>99%)
  - Electromagnetic showers
- Cherenkov radiation emission
  - optical and near-UV light
  - Flash duration of few ns
- Light collected by the mirrors
- Light focalized in the cameras
- EAS images in the focal planes
- Image calibration, cleaning, and parameterization
- Stereoscopic reconstruction and background suppression
- Scientific analysis
- Science products
- Observatory products



Imaging Atmospheric Cherenkov Telescopes

- Next generation ground based Gamma-ray Observatory
- Open observatory
- Two sites with total > 100 telescopes (LSTs+MSTs+SSTs)
  - Southern Site: Near Paranal in Chile (selected for negotiations)
  - Northern Site: La Palma, Canary Islands (selected for negotiations)
- 32 countries, ~400M€ project



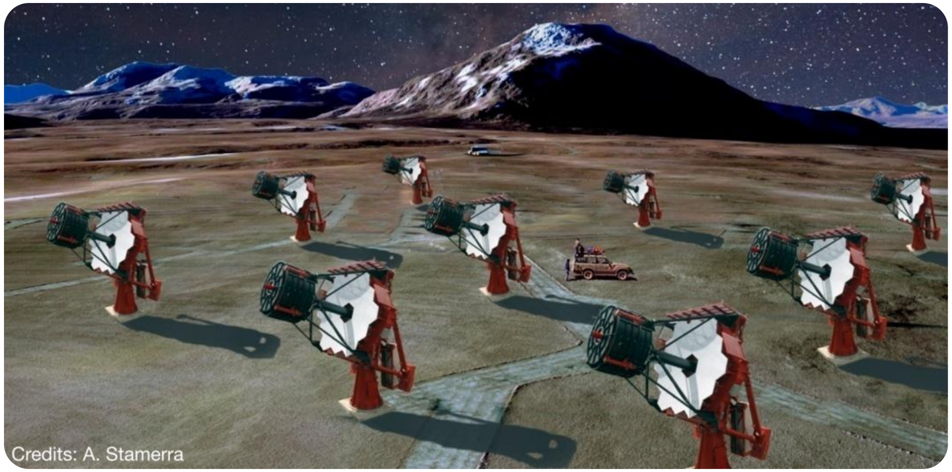
The **ASTRI Project** (led by INAF) has two main goals:

- an **end-to-end prototype of the CTA small-size telescope** in a dual mirror configuration (**ASTRI SST-2M**), inaugurated on 2014 Sept. 24th at the INAF observing station on Mt. Etna (Sicily) and undergoing the scientific and performance validation phase by the end of 2016;
- an **ASTRI mini-array** composed of **9 ASTRI telescopes** proposed to be installed at the chosen CTA Southern site likely in 2018


Inauguration @ Serra La Nave observing station (Mt. Etna, Sicily)
Credits: T. Abegg


Artistic view of the ASTRI mini-array @ CTA Southern site
Credits: A. Stamerra

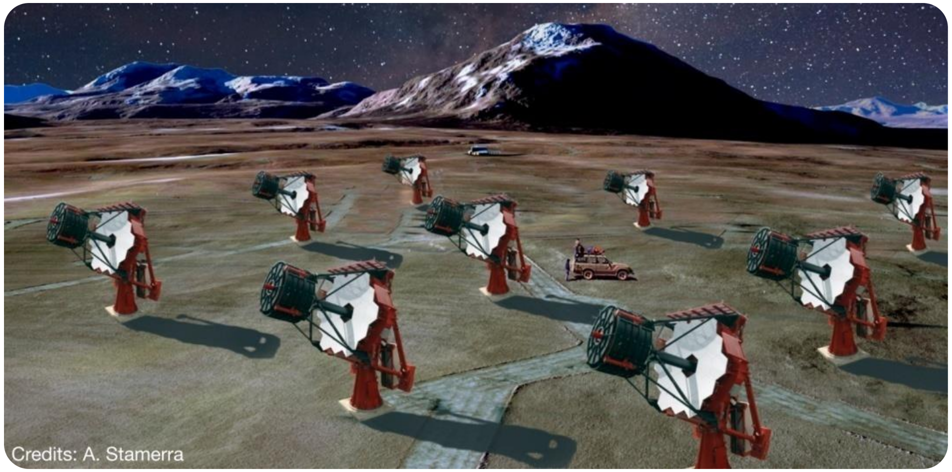
The ASTRI Mini-array



- Proposed to be installed at the southern site of the CTA, as an initial seed of the entire observatory

- In such a location power consumption, data bandwidth, heat dissipation become critical concerns

The ASTRI Mini-array



- The capability of each detector to process, at least partially, its own data before sending them to the central data acquisition could provide a key advantage
- Carrying out preliminary data reduction on the telescope would greatly decrease the bandwidth required by the array installation

The ASTRI Scientific Software

- Two main goals:
  - Having something working and fully functional ready to analyze real data from ASTRI camera in fall 2016
  - Having the low level part of the analysis running on low-power architectures (i.e. ARM+GPUs)

A-SciSoft is organized in four distinct functional breakdown stages:

- **Calibration**
- **Reconstruction**
  - **Telescope-wise**
  - **Array-wise**
- **Analysis**
- **Science**

Calibration (telescope-wise)
Apply data calibration algorithms and perform data quality checks

↓

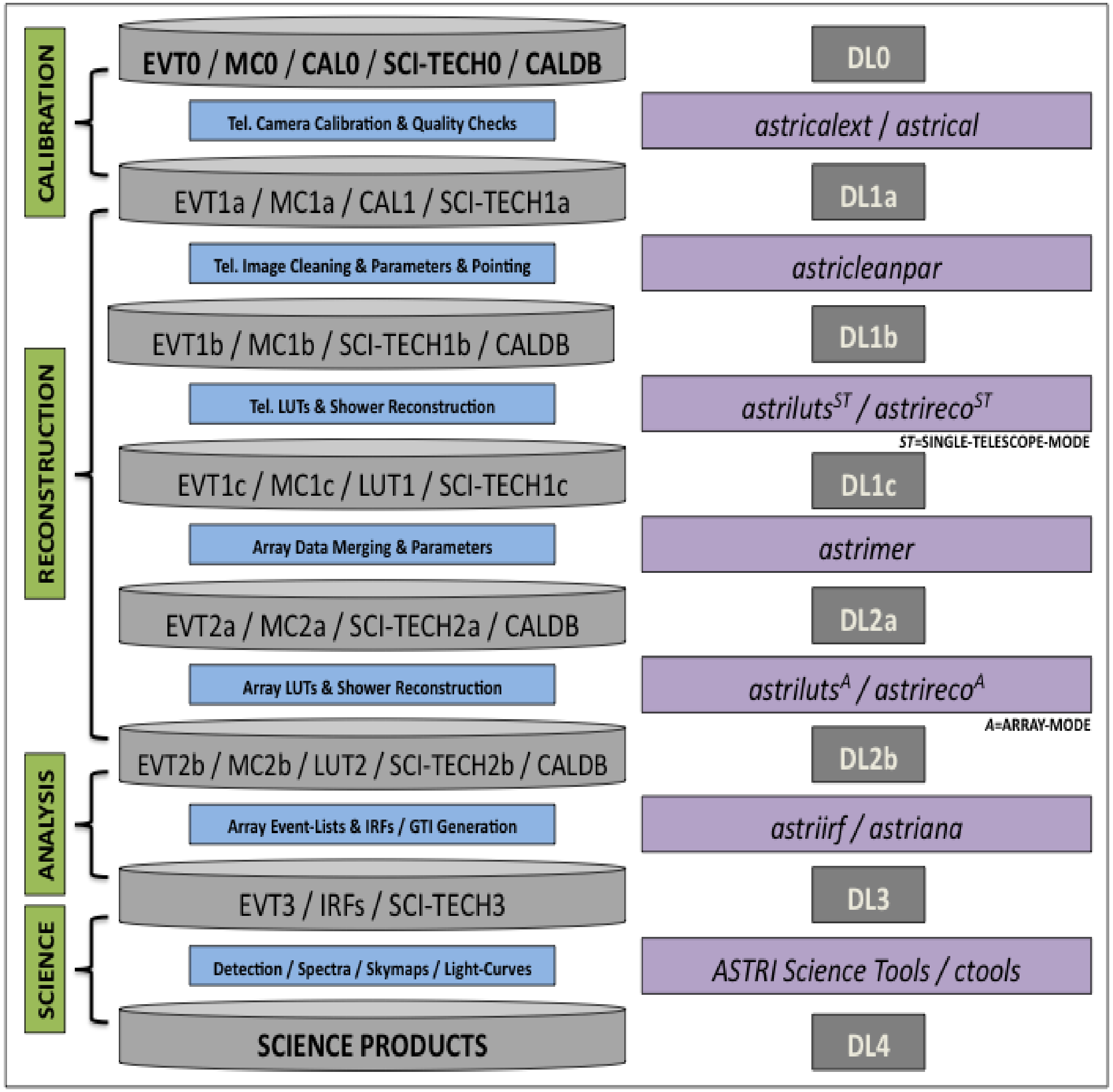
Reconstruction
(telescope-wise)
Image cleaning and parameterization + single-telescope event reconstruction
(array-wise)
Stereo parameters and reconstruction (event direction, energy, classification)

↓

Analysis (event-wise)
Production of gamma event list, IRFs and good time intervals (GTI)

↓

Science
Production of detection plots, spectra, sky maps, light curves



Breakdown stages; Basic components; Executable modules; I/O Data level.

The ASTRI SST-2M Prototype and Mini Array Data Pipeline

- Manages **FITS data** (from DL0 to DL3) adopting CFITSIO/CCFITS libraries;
- Operates on chuncks of events
- It is written in C++, CUDA, Python
- It is developed in independent software modules linked by pipelines written in Python

Pixel-level algorithms easily express parallelism

- **Calibration**

  Essentially an *embarrassingly parallel* Fused Multiply-Add (FMA) operation:

  ```
  $0 = $0 × $1 + $2
  ```

- **Cleaning**

  Two pass cleaning (two thresholds comparison)

  Well suited to parallelism

Calibration

Calibration formula (for each pixel):

PHE = ADC * CC + PED

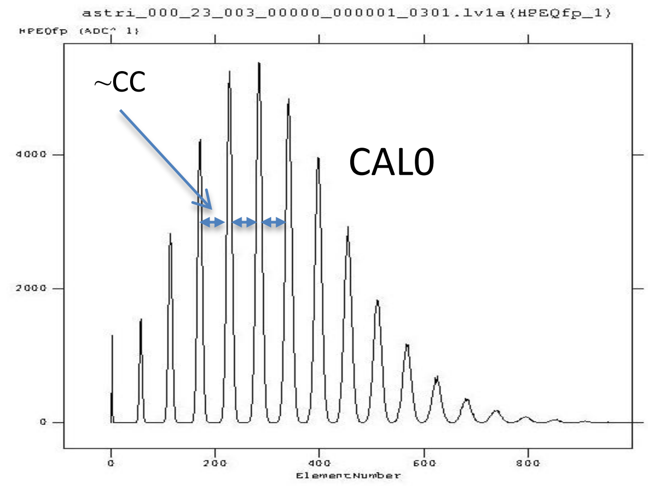
Where:

PHE: photo-electron equivalent

ADC: ADC counts from camera pixel

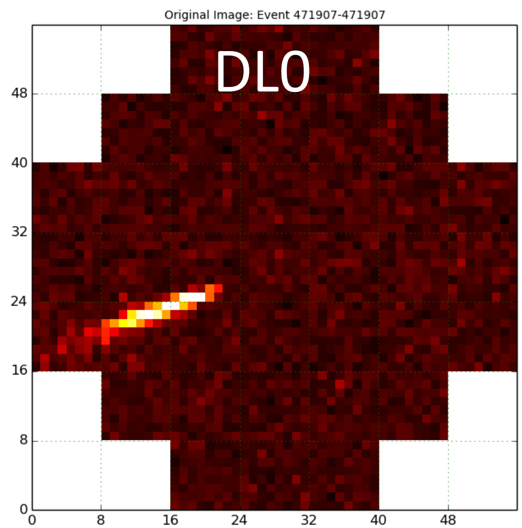
CC: calibration coefficient

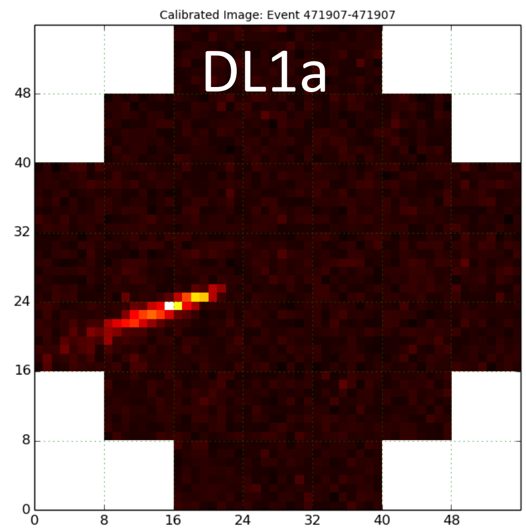
PED: pedestal value



- Extraction of CAL1 from CAL0
- Application of CAL1 to EVT0/MC0
- NSB level extraction
- Data Quality Checks
- Production of EVT1a/MC1a

(TEL.-WISE CALIBRATED DATA)

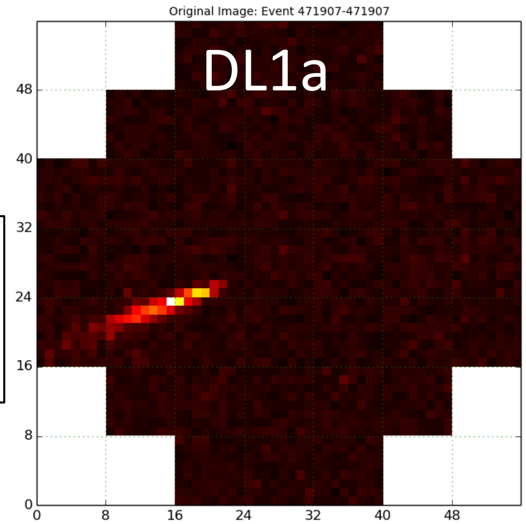


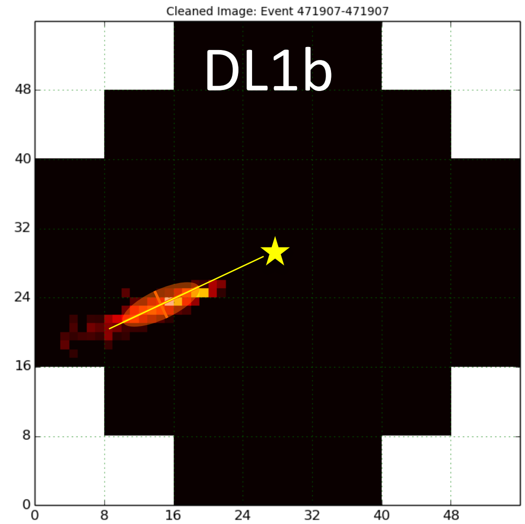


Cleaning

- Two pass cleaning (two thresholds comparison)
  1. Identify seeds of core pixels
  2. Find neighboring pixels constituting the image boundary
- Well suited to parallelism
  - parallelize both per pixel and per event

❖ Image Cleaning and Parameterization
❖ Telescope Pointing Reconstruction
❖ Production of EVT1(b)/MC1(b)





Numerical Precision Validation

- FMA instruction matters! (Available since Haswell)

$$round(X * Y + Z)$$

vs.

$$round(round(X * Y) + Z)$$

- FMA makes computations faster and more accurate!*

Calibration: phe = cc*adc+ped
double precision math

```
In [3]: 0.0444444455*971-43.1333351
Out[3]: 0.022221480500000723

In [4]: 0.0439560451*968-42.6923103
Out[4]: -0.1428586432000003

In [5]: 0.0439560451*970-42.6483536
Out[5]: -0.010989852999998106
```

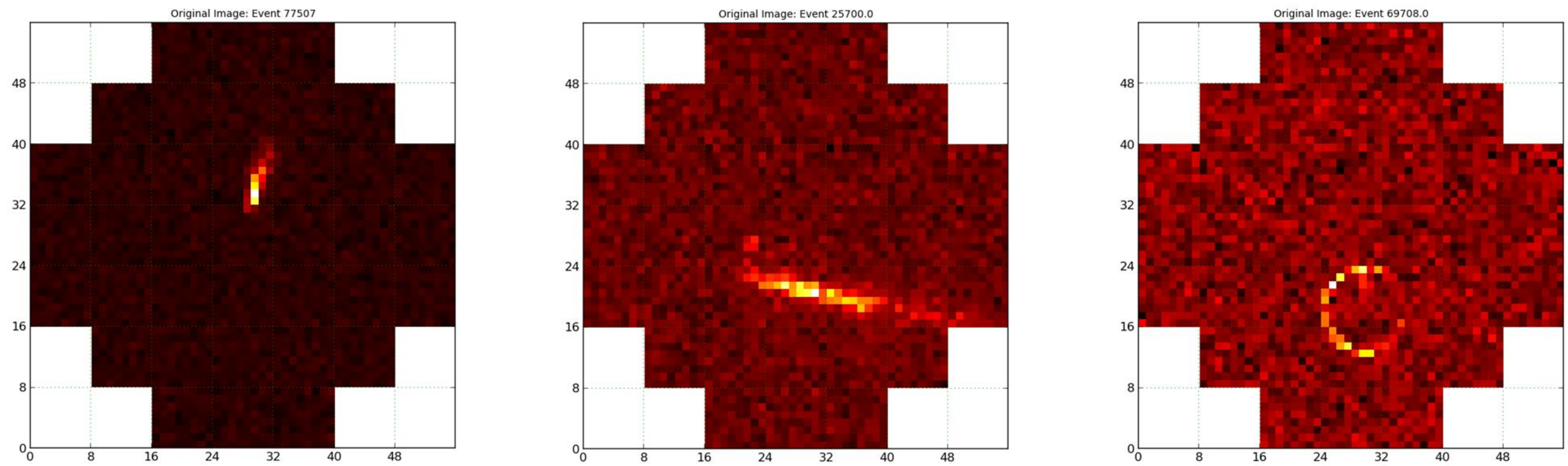
ARS FMA	ARS No FMA
1,311110E+00	1,311111E+00
2,222149E-02	2,222061E-02
2,466666E+00	2,466667E+00
5,164819E-01	5,164833E-01
-1,428587E-01	-1,428604E-01
1,791207E+00	1,791206E+00
3,593406E+00	3,593407E+00
-1,098987E-02	-1,099014E-02
3,021977E+00	3,021976E+00

*Whitehead, N.; Fit-Florea, A. - NVIDIA (2011). "Precision & Performance: Floating Point and IEEE 754 Compliance for NVIDIA GPUs".

Seamless and flexible integration of GPU code

- Build phase does not depend on CUDA toolkit
- Software modules detect GPUs in the system and act accordingly (*thanks C++ polymorphism!)*
- CPU/GPU execution switchable on user request
- Important to enable GPU *Persistence daemon* to minimize driver load latency

Reference Test Case



- 55000 (≈ 500 MB) events of simulated "real data"
- ≈ 110s of nominal acquisition rate (500Hz)
- ≈ 80.5% of events survives cleaning with default settings
- Compliant with format and size agreed with camera hardware team

Development system

Workstation

Dual-processor Intel Sandy Bridge @ 2GHz with 16 physical cores and 128GB of RAM (8GB per core)

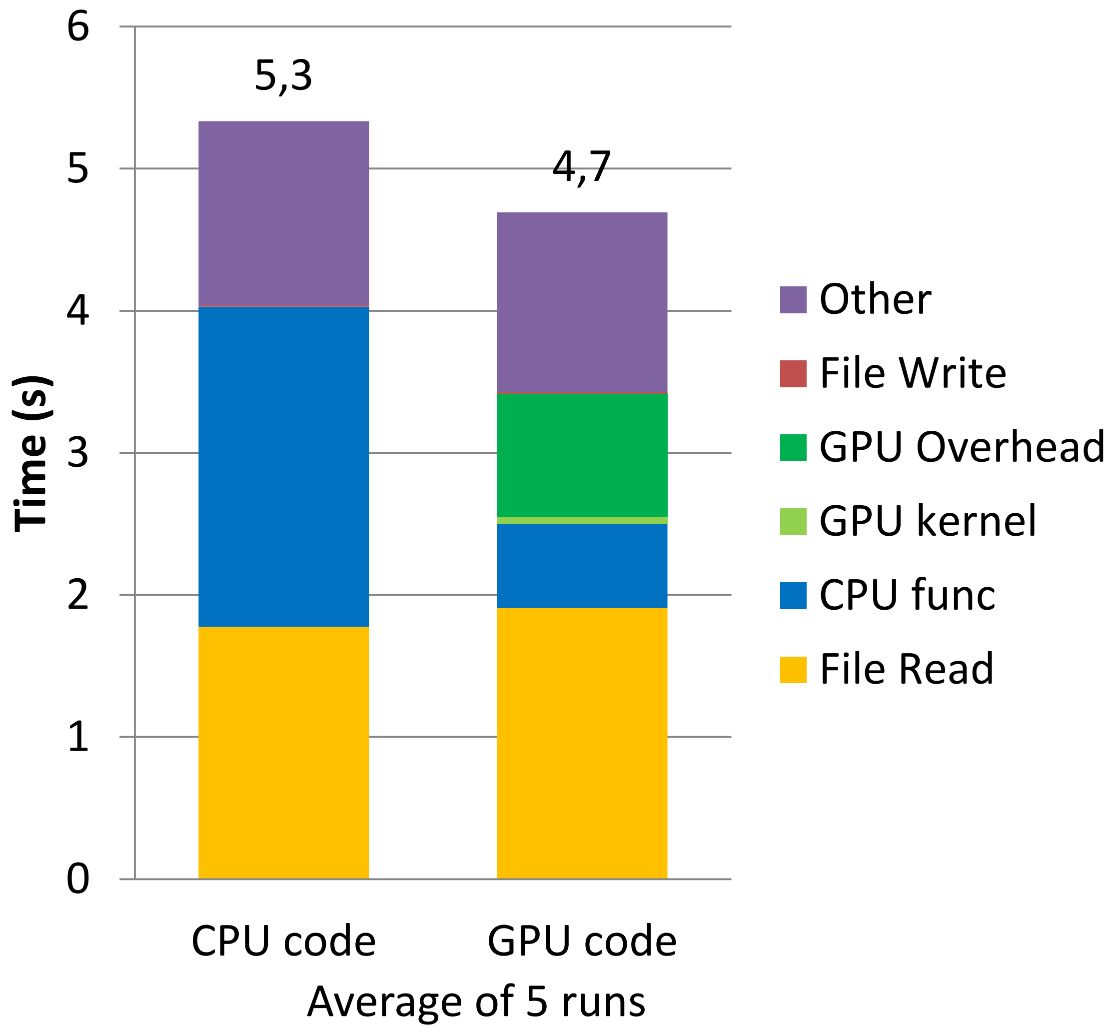
GPU gen3 READY and n.1 installed (up to 3 GPU drives)

8 disk slots of 4TB each (to export 2 different redundant drives of ~12TB each)

direct link and share with the storage system

- Installed @ OAR Monte Porzio Catone
- Accelerator: One NVIDIA Tesla K20c (20-30% slower than K40)

Unified module (up to DL1b)



- Direct processing from DL0 to DL1b
- 73x reduction in size:

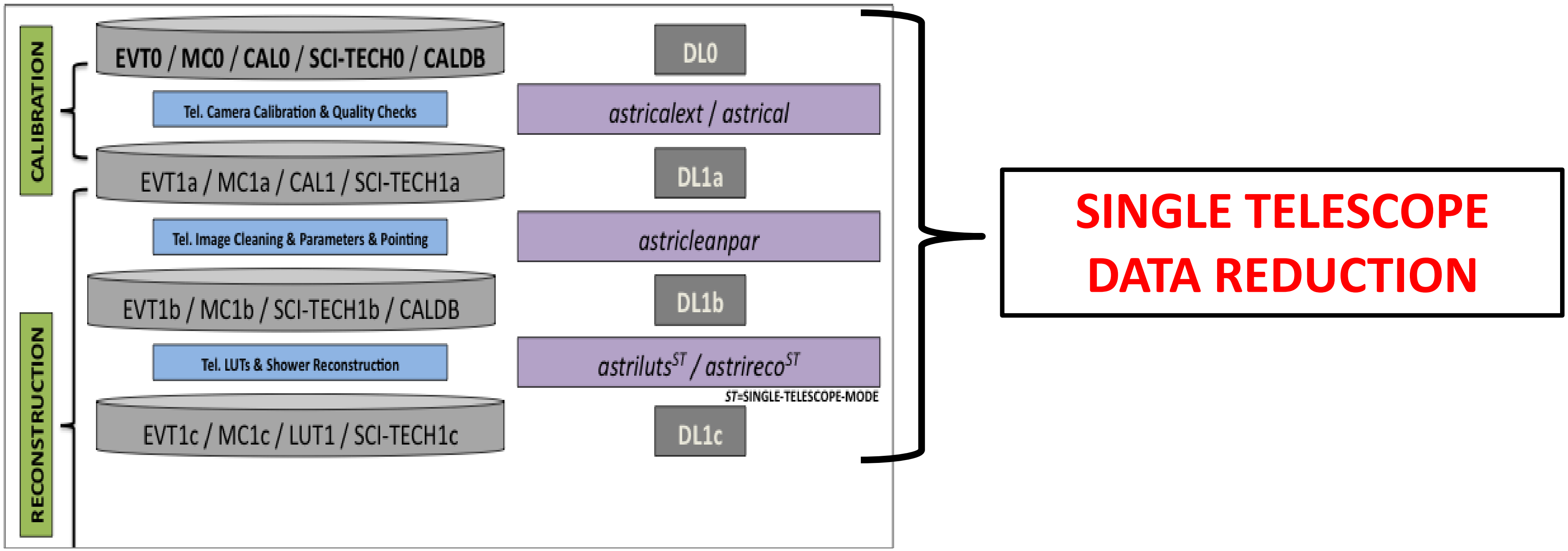
500 MB input

↓

6.9 MB output

- CPU parallel processing with 32 OpenMP threads

Low-power vs Speed

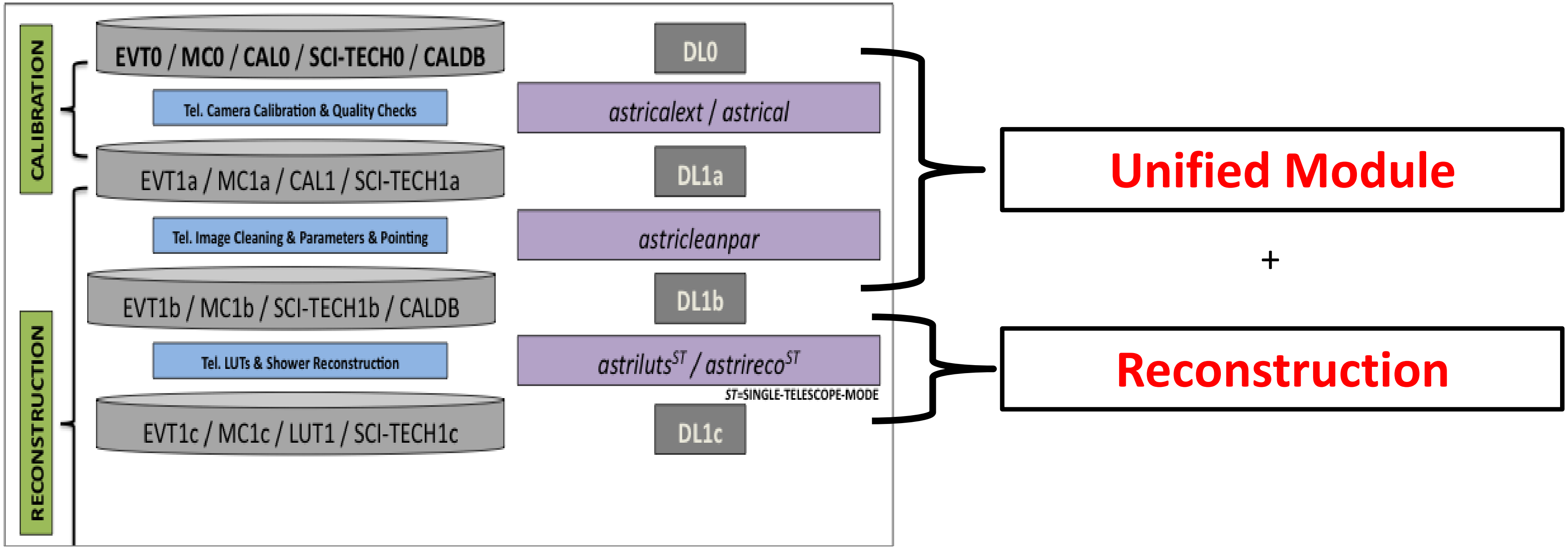


BUT

CTA requirement:

- The observatory should be able to send scientific alert within **30s** after the acquisition of the last event participating to the alert.

Low-power vs Speed



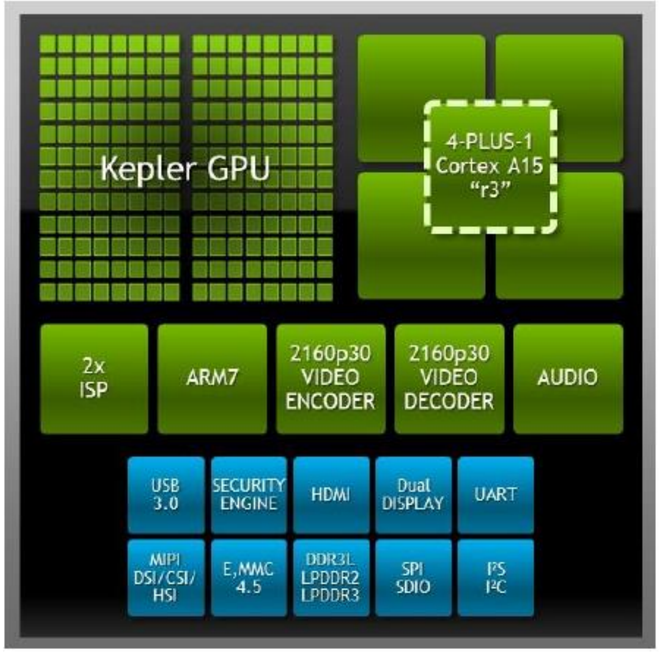
BUT

CTA requirement:

- The observatory should be able to send scientific alert within **30s** after the acquisition of the last event participating to the alert.

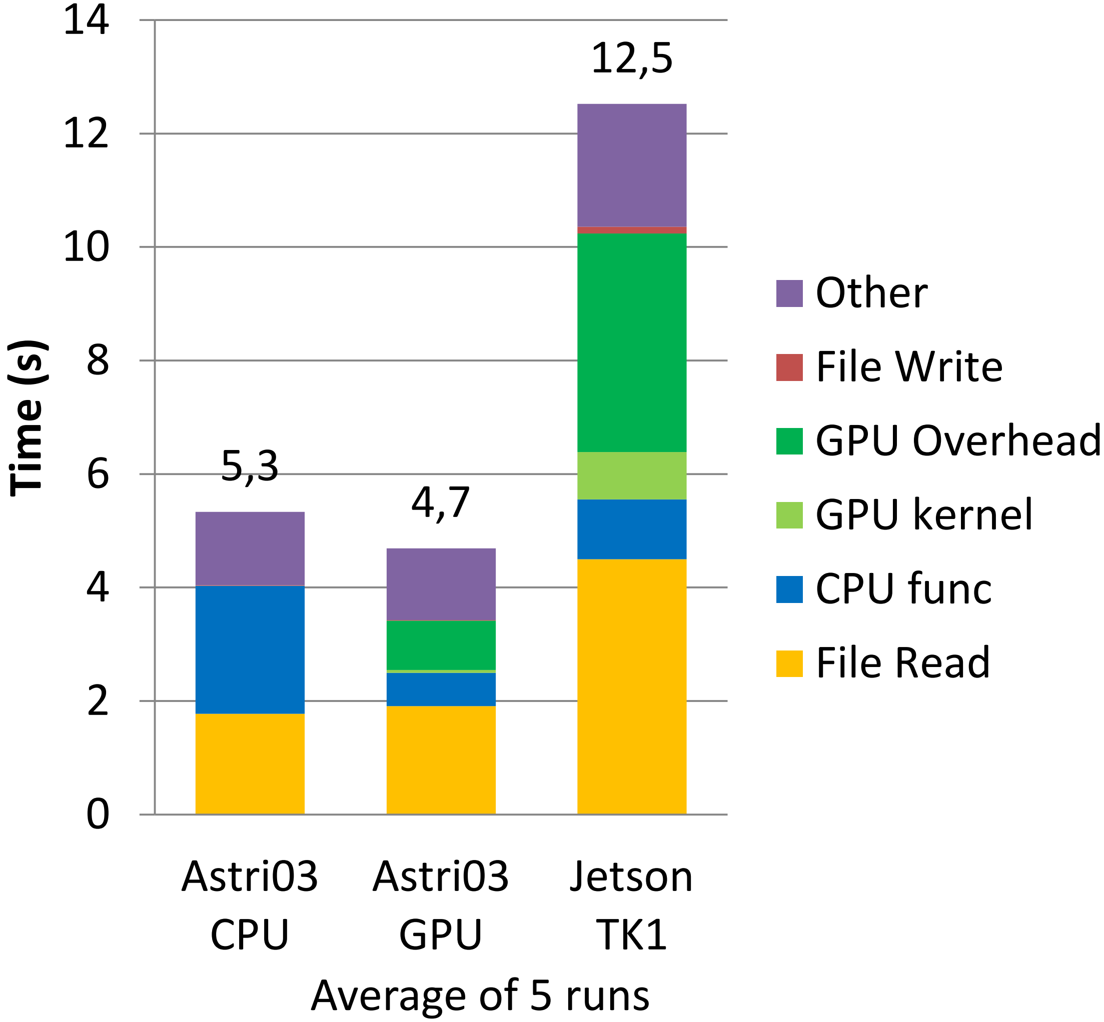
NVIDIA Jetson TK1

- Heterogeneous System-on-Chip
- CPU: Quad-core ARM A15
- GPU: Kepler architecture - 1 Multiprocessor
- RAM: 2GB (memory shared by CPU & GPU)
- OS: Ubuntu 14.04 Linux for Tegra (L4T)
- CUDA 6.5
- I/O: SATA 3Gb/s HDD



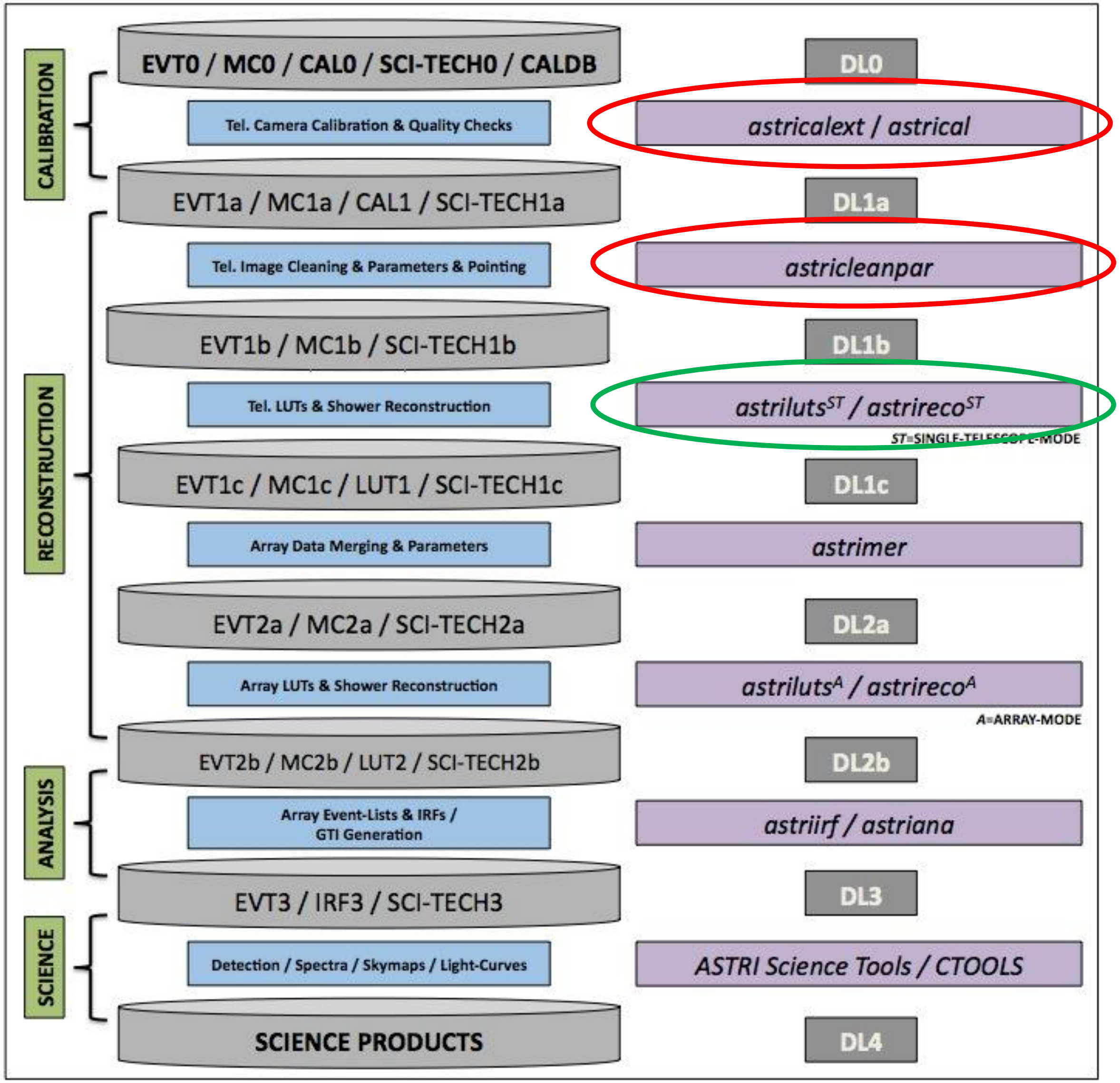
Average power consumption: < 10 W

Low-power Unified module



- Processing from DL0 to DL1b in **12.5 s**
- Including DL1c, total time will be:

  22.5 s < 30 s



astrireco

- Implements random forest application
- Loads pre-trained models (look up tables LUTs)
- Energy, direction and hadronness reconstruction

Execution time: 10s - 4 ARM core (using OpenMP)

{Time,Energy} to solution

- Our computing node: 225W
- Jetson: 10 W peak
- Rough (conservative) estimation:

	Workstation	Jetson	Improvement
Time to solution	6,5 s	22,5 s	0,28 x
Energy to solution	1430 J	225 J	6,3 x
Energy delay product	9295 Js	5062 Js	
Event/s	7690 evt/s	2200 evt/s	
Energy/event	29,6 mJ/evt	4,5 mJ /evt	

Challenges

- Soon tests on real ASTRI SST-2M prototype data @ Mt. Etna (fall 2016)
- Finish porting of clustering on GPU
  - Less time on PCIe transfers
- Try the new Jetson hardware: NVIDIA Jetson TX1
  - More effective usage of Sistem-On-Chip memory
  - Improvements in time foreseen within the same power consumption (10 W)