CTA Observatory - Computing and Data Management

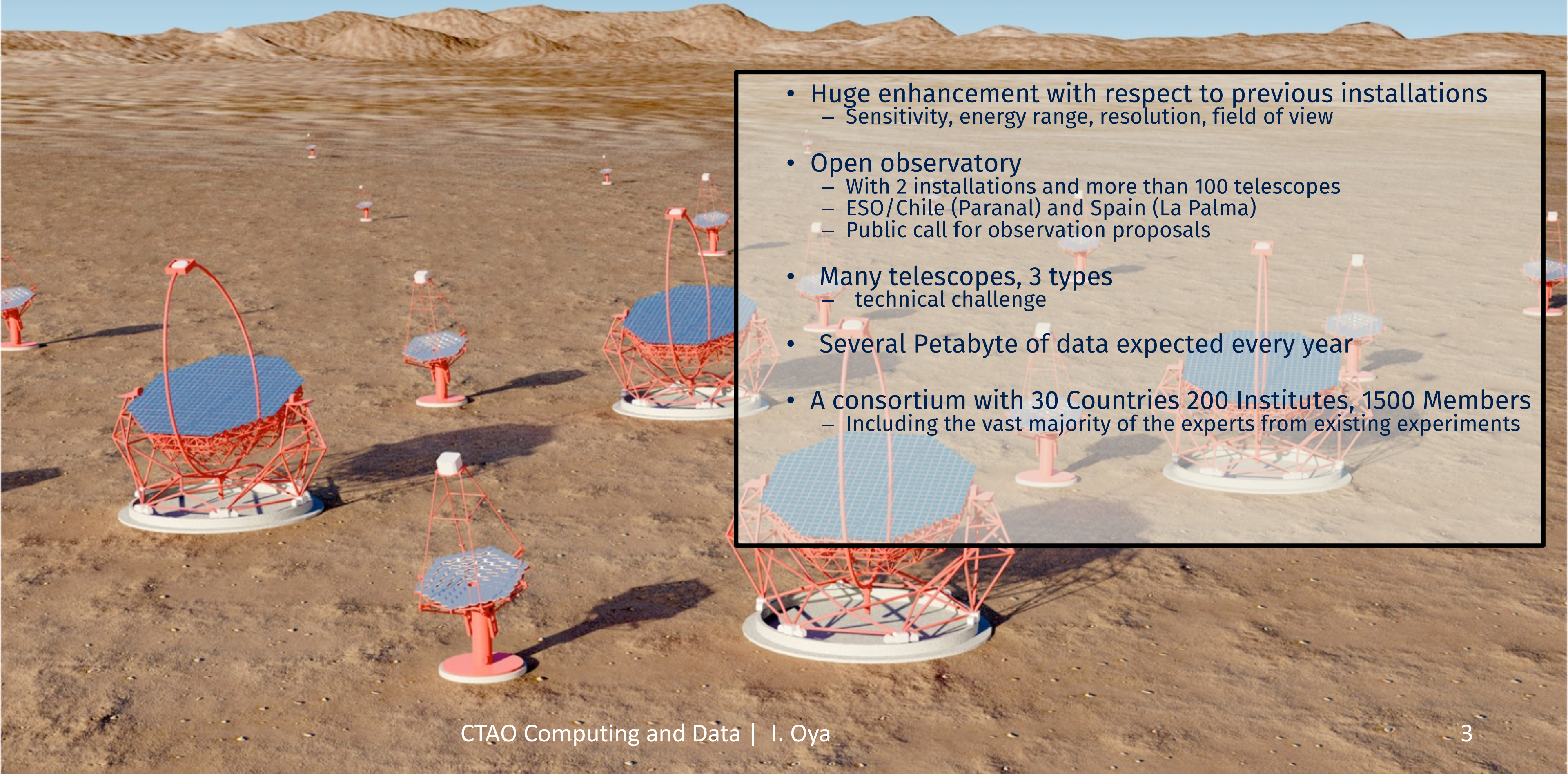
I. Oya

CTA Observatory gGmbH

Zeuthen Data Science Seminar, 2023-01-24.

Contents

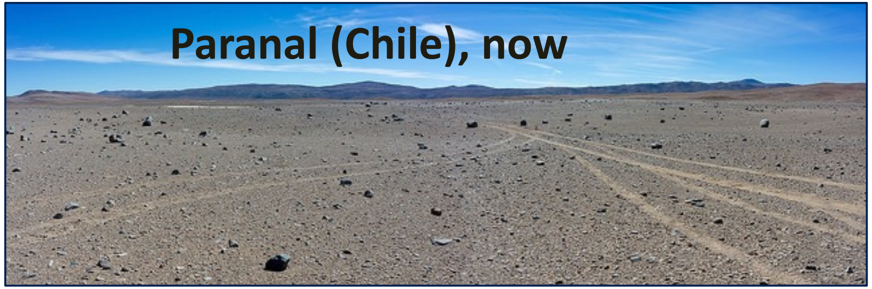
The Cherenkov Telescope Array Observatory

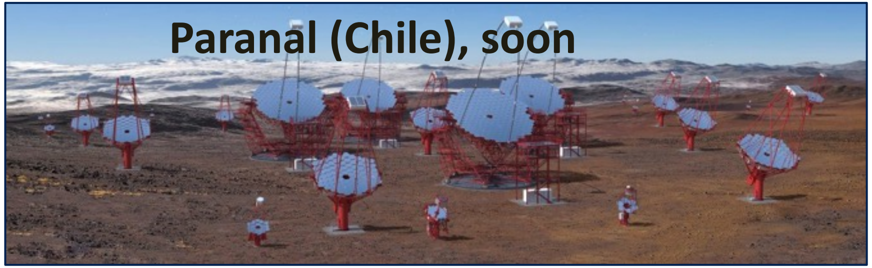


- Huge enhancement with respect to previous installations
  - Sensitivity, energy range, resolution, field of view
- Open observatory
  - With 2 installations and more than 100 telescopes
  - ESO/Chile (Paranal) and Spain (La Palma)
  - Public call for observation proposals
- Many telescopes, 3 types
  - technical challenge
- Several Petabyte of data expected every year
- A consortium with 30 Countries 200 Institutes, 1500 Members
  - Including the vast majority of the experts from existing experiments

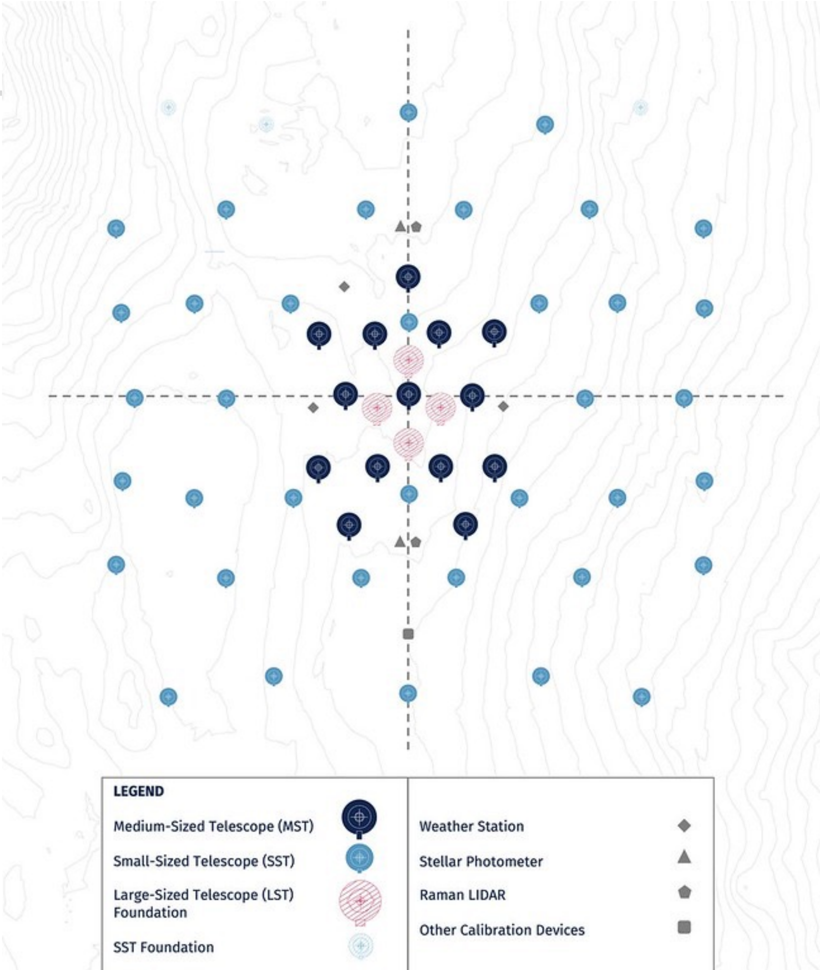
CTAO: a data driven observatory

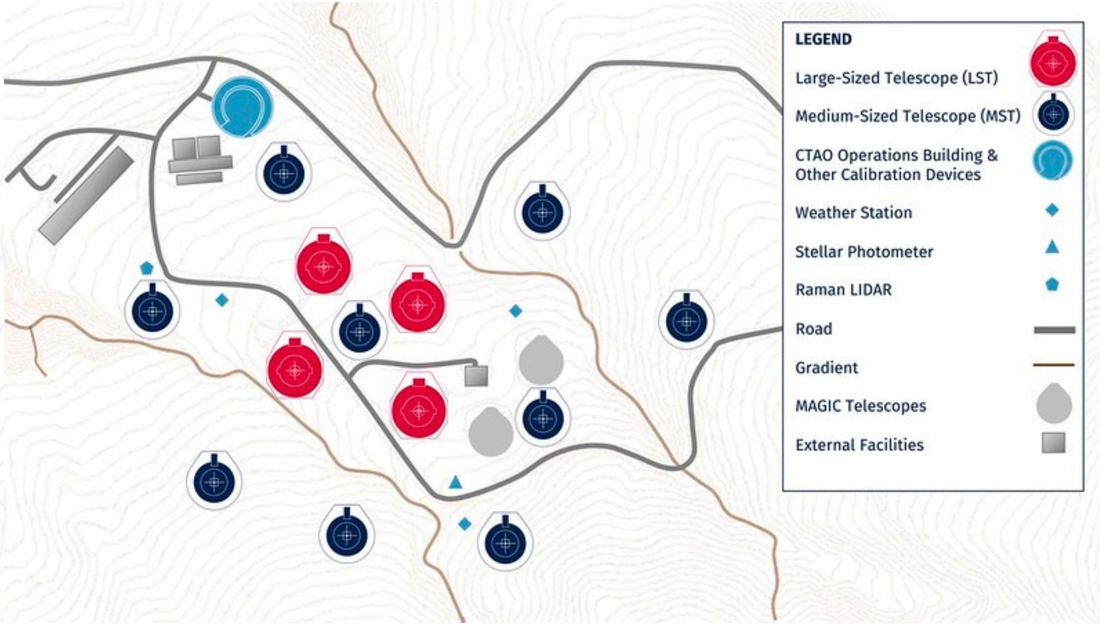
CTAO at Paranal & La Palma



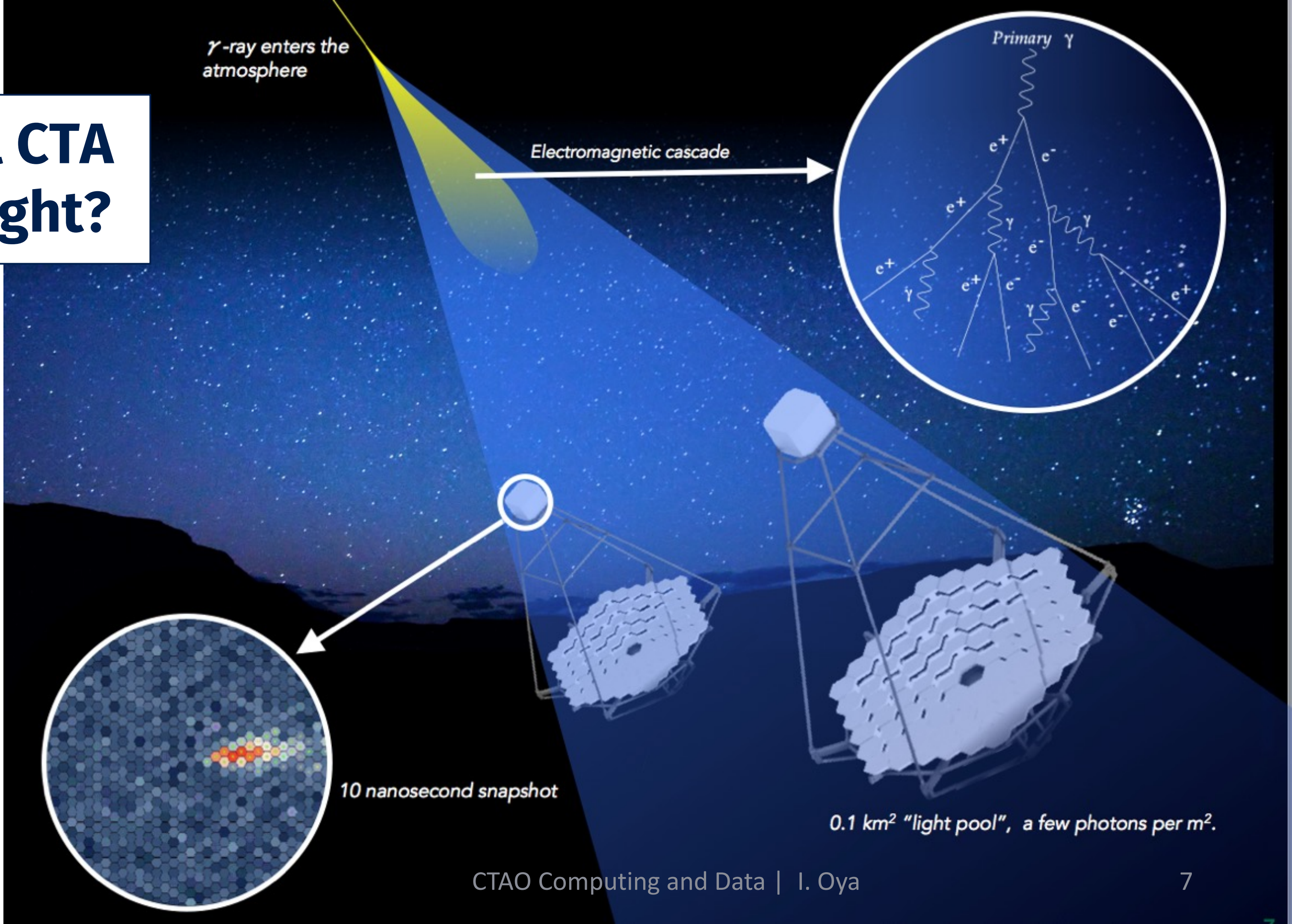


CTAO at Paranal & La Palma (II)



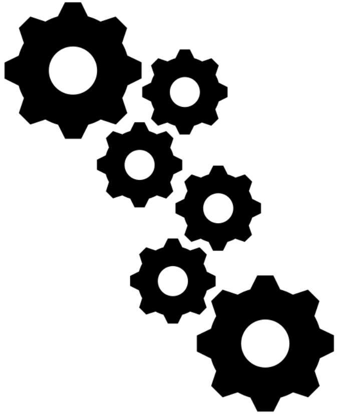


How will CTA detect light?



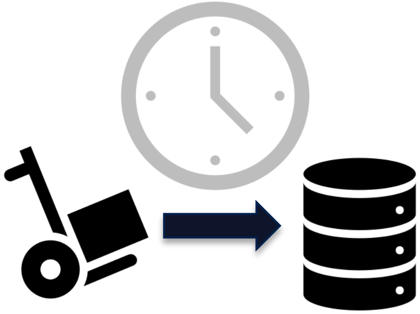
CTA Observatory Operations

- An Open Observatory / User Facility
  - For the first time in this waveband
  - Annual AoO, TAC ranking, long-term schedule
  - Proposal preparation support, tracking, helpdesk, …
  - Public science data archive – After proprietary period
- Two Telescope Arrays & HQ & SDMC – one Observatory
  - Inter-site coordination
  - Uniform approach to science operations
- Main Challenges
  - Sub-array operation, wide field of view, instrument response generation, background modelling, rapid alert generation and response, data volume, science operations during construction
- A Software Instrument
  - Software plays a critical role in all steps of the Observatory

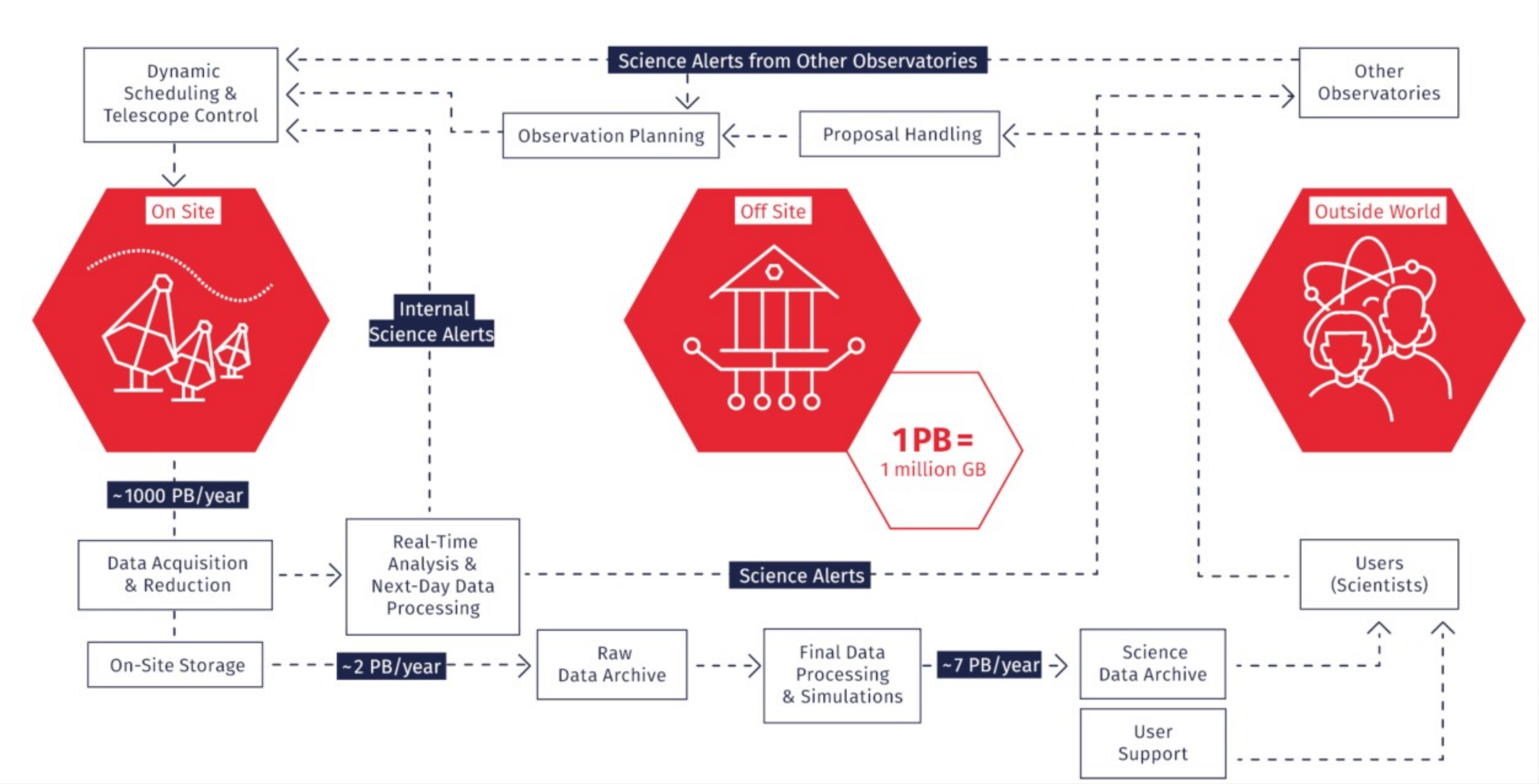


CTA Observatory Operations – Commitment

- "Data" is the final product of CTAO
  - Everything else is "just support to get data"
- Operating CTA as an Observatory impacts many areas
  - Provider – Customer relationship
    - CTAO = Supplier, Science User = Customer
  - "Observatory" means "Commitment"
    - To deliver data of defined scope, quality and within time
    - To treat science users equally and fair
    - To keep the observatory at the forefront of state-of-the-art research
    - To keep our services up and running 24/7/365
  - 30 years life time
    - Significant maintenance effort
  - Limited operations budget
    - Build, operate and maintain simple and robust CTAO system



CTAO OPERATIONS

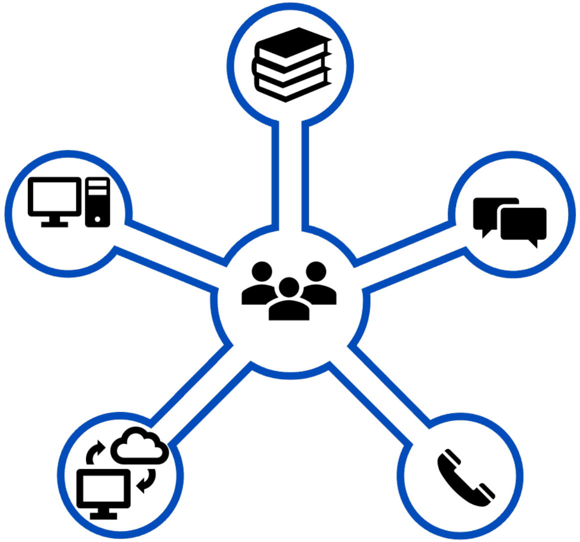


Observation Execution	Data Processing and Preservation	Science User Support

Initial calib./reduction →Transmission from site →Bulk data archive →Science data archive

Science Operations – Implications

- Science Operations = Data processing + User Support Services + Computing Facilities Support
- Main elements for efficient and sustainable science operations include:
  - Observation planning and execution software
  - Processing software
  - Computing facilities and storage
  - Expert people to run the software (e.g. MC Simulations)
- Prepare and deliver science data to users
- User Support Services



Science Operations – Implications

- Science Operations = Data processing + User Support Services + Computing Facilities Support
- Main elements for efficient and sustainable science operations include
  - Observation planning and execution software
  - Processing software
  - Computing facilities and storage
  - Expert people to run the software (e.g. MC Simulations)
- Prepare and deliver science data to users
- User Support Services

- Call for Proposals
- Proposal Handling
- Science Portal, FAQ, Newsletter
- PI interactions, incl. science alerts
- Provide tools for proposal preparation, data analysis
- Science planning
- Data processing and data quality monitoring
- Data archiving
- MC simulations
- Provide instrument response functions
- Science Analysis tools
- …

CTA Observatory and PI

Science ops start and end with the PI...

... and the rest will be handled by the CTAO Computing services (and the operations team using it)

CTAO Computing Challenges (I)

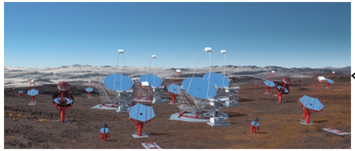
- Schedule observations
  - up to 8 sub-arrays concurrently at each site
  - optimize instrumentation usage
- Control and supervise
  - handle O(100) telescopes of several, quite different types
  - 8 subarrays
- Provide a software-based central trigger
- Sub-nanosecond accuracy timestamping of events
- Control and supervise LIDARS, ceilometers, and other auxiliary instruments
- React on a few second timescales to internal and external science alerts

CTAO Computing Challenges (II)

- Reducing On-site Data Volume
  - 1000 PB/y directly coming out from cameras (mostly noise)
  - Impossible to transfer via internet link (see later)
- Storing and Processing Big Science Data:
  - PBs of data
  - Distributed computing model
  - Data volume is too big to separate storage from computing
- Simulating CTA
  - Development of extensive air showers
  - Propagation of Cherenkov light
  - Cherenkov photon ray-tracing through the telescope optics to camera
  - Photosensors simulation and camera electronics
- Huge, distributed, and diverse team of developers
  - 100+ developers, different countries, from research institutions and SW developer companies, senior SW engineers and students, …

CTAO Computing Systems

Instruments



main software systems

ACADA (Array Control and Data Acquisition)

DPPS (Data Processing and Preservation System)

SUSS (Science User Support System)

Science Users

"Central information hubs for coordination"

TOSS (Technical Operations Support System)

SOSS (Science Operations Support System)

infrastructure

Array Clock

On-site ICT (Information and Communications Technology)

Off-site ICT (Information and Communications Technology)

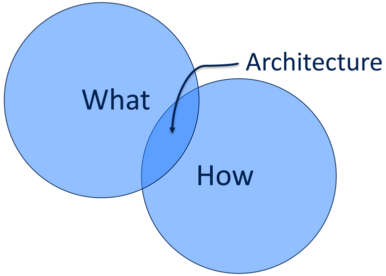
own and managed

CTA Data Centers

service-level agreements

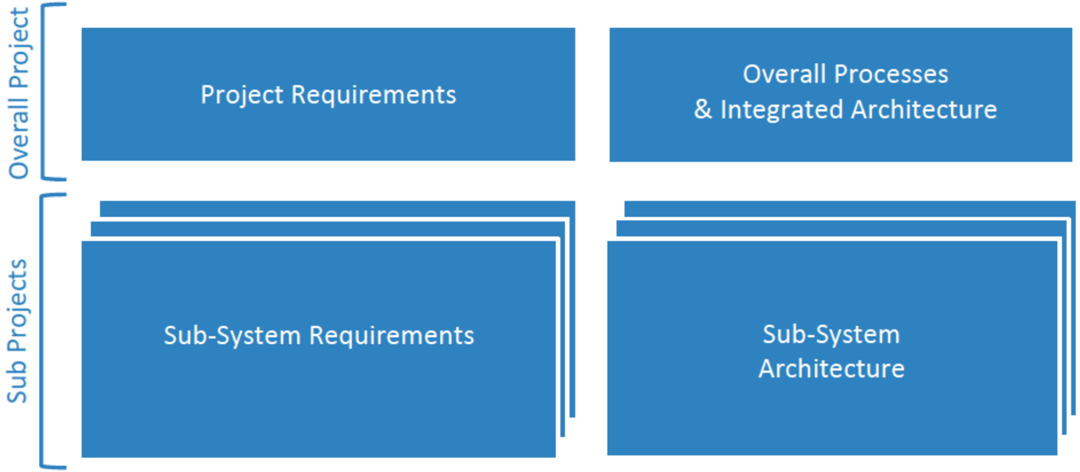
Collaborating Data Centers

We developed a formal system architecture, data modelling, and systems engineering process

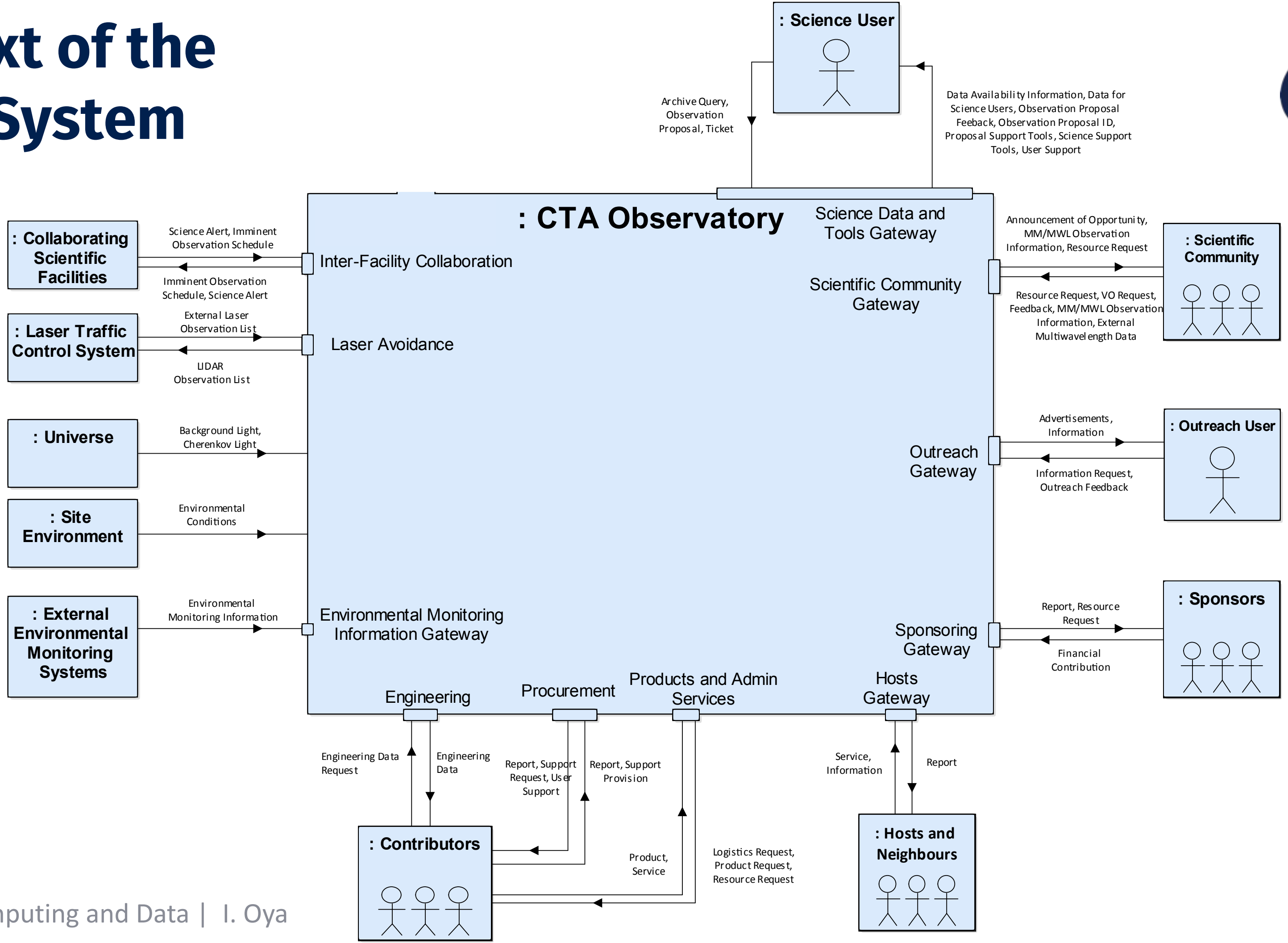


The CTA Observatory System Architecture

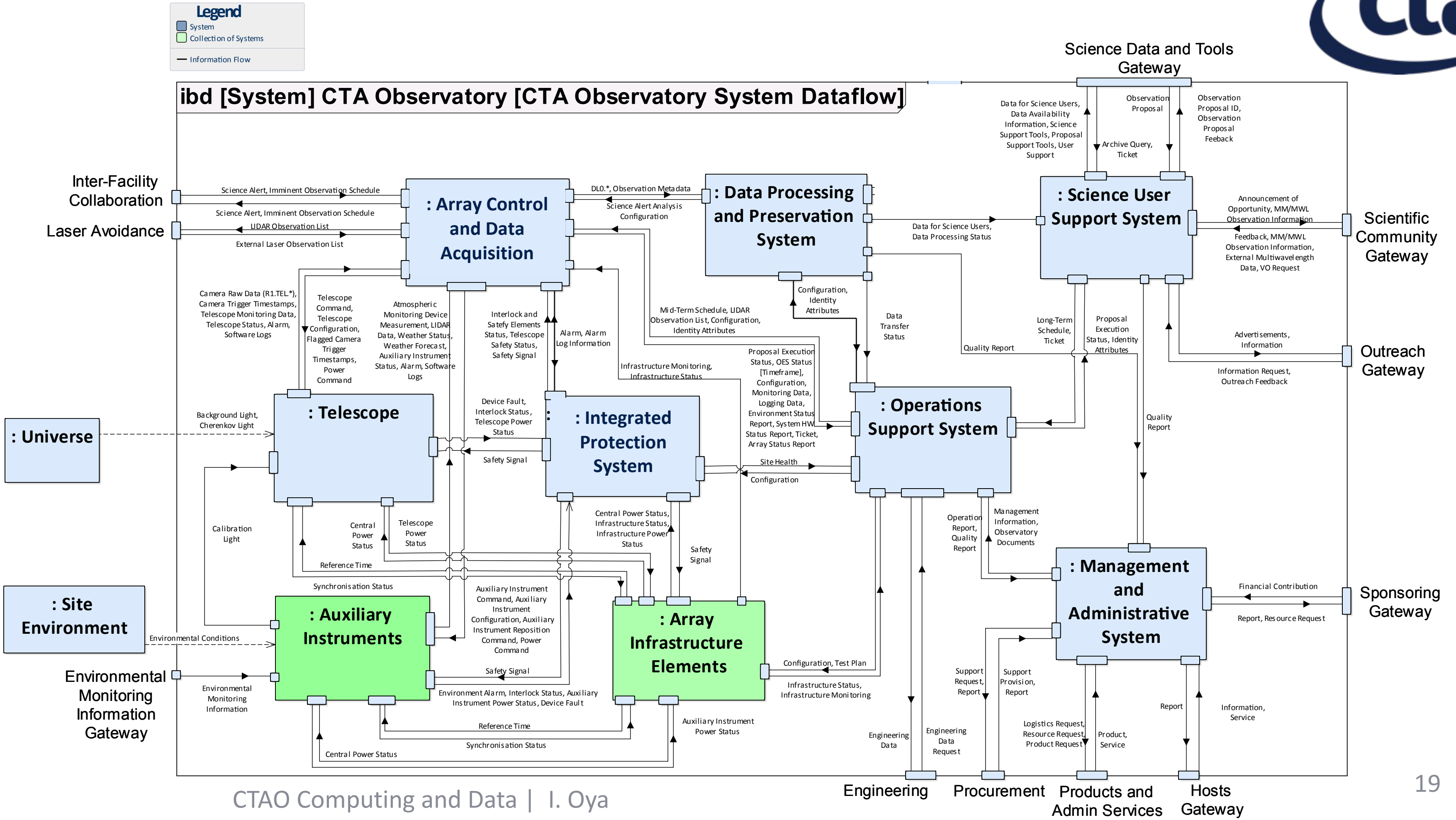
- Architecture summarizes all decisions taken in the project, with views representing the resulting system structures
- **Different layers**: the whole CTAO System, and sub-systems
- **Different abstraction levels:** e.g. functional, logical viewpoints
- **Includes everything**: systems, stakeholders, processes, data, interfaces…
- **Model-based and formal approach**:
  - SysML/UML notation
  - Implemented in Sparx Enterprise Architect



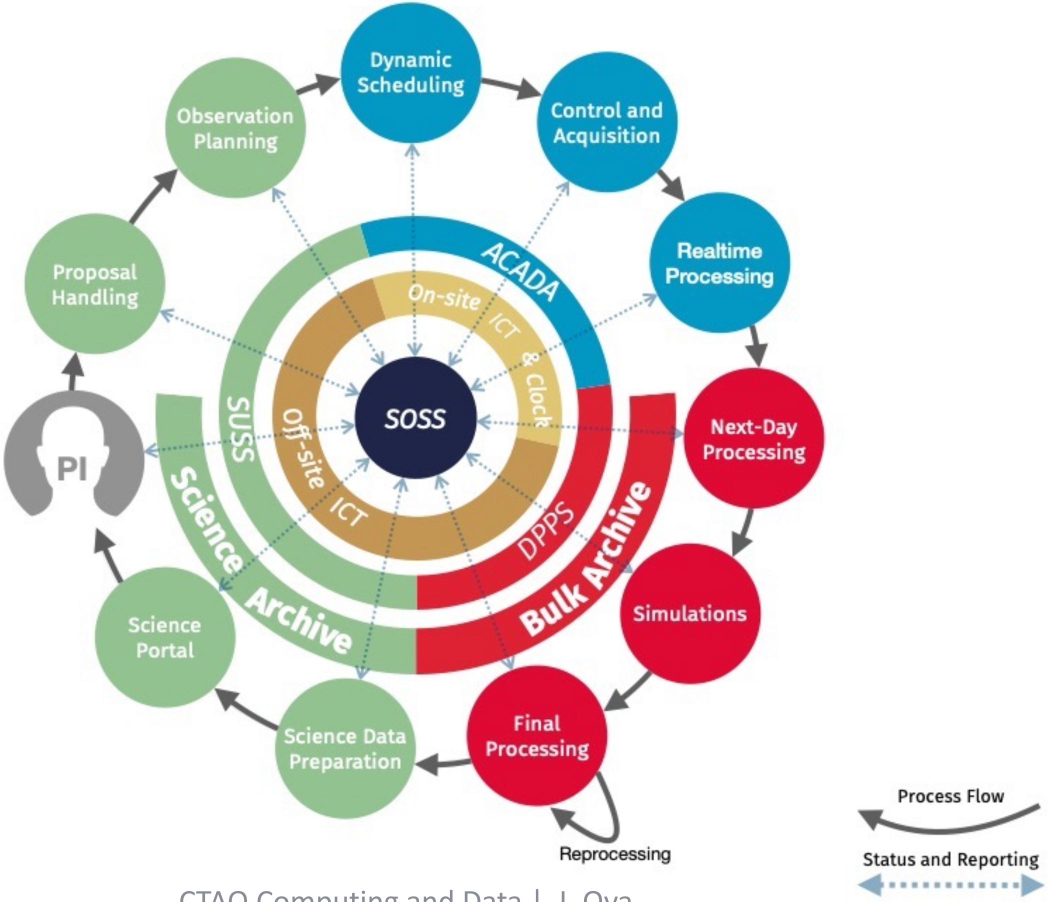
Context of the CTAO System



The CTA System Structure



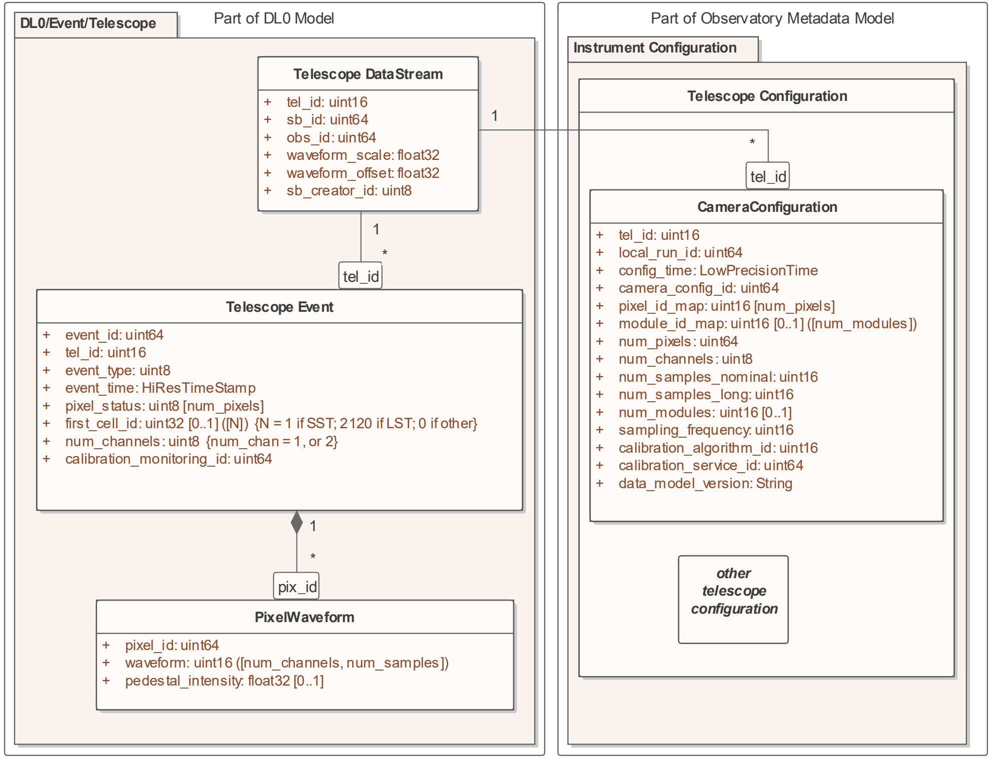
Science operations and software systems



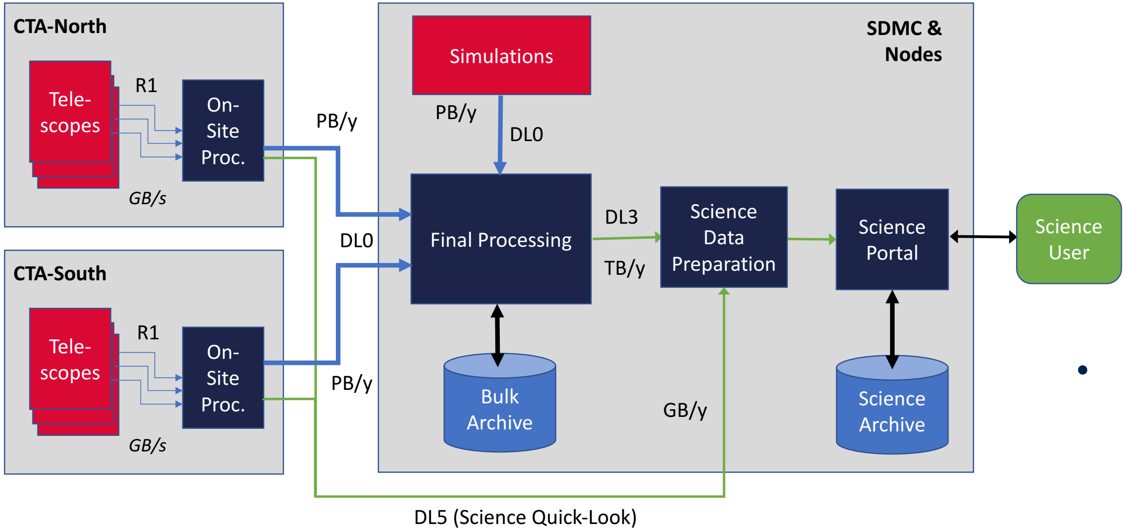
Data Modelling

- Modelling the data is a fundamental prerequisite to:
  - define how data will be handled and preserved by the CTAO computing systems
  - specify the interfaces between the systems
  - to implement the software systems
  - identify the computing infrastructure needs
- CTAO contains lot of different types of data → data modelling is a significant effort

example of raw archived data (DL0) Camera Event Data

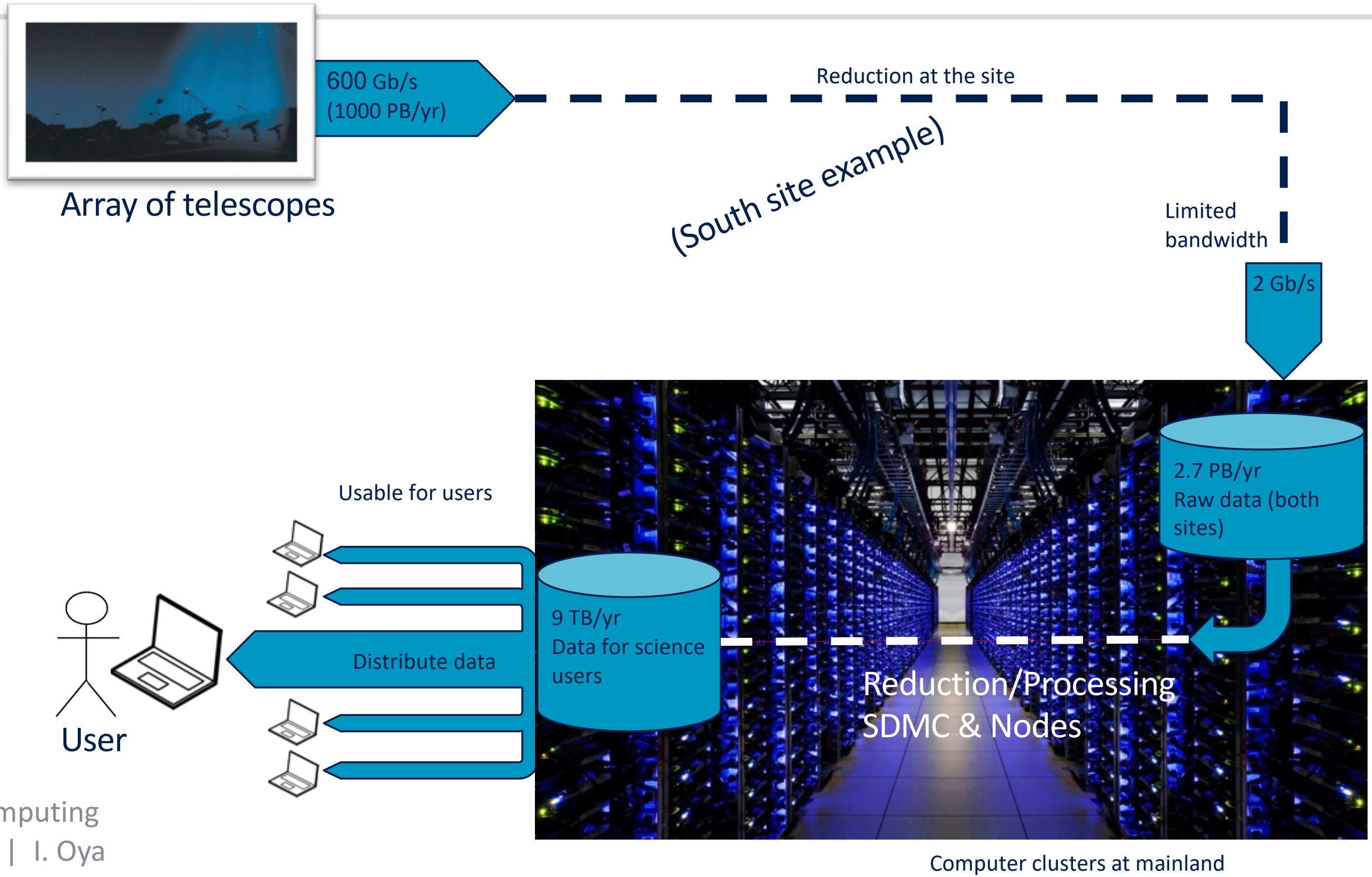


Data Flow & Analysis Categories

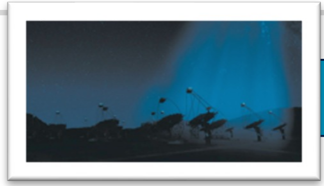


- **Three data product categories**
  - *Cat A*: at the sites, real time, robust, generate internal science alerts, online data quality
  - *Cat B*: At the sites, offline (e.g. next day), tailored version of “C” → better calibration than Cat. A, ready by next morning
  - *Cat C*: offsite, final analysis results with maximum sensitivity, much more computing (and time) required
- **Strong data reduction**
  - Reduced, from the photon-detector data up to the data delivered to *non-expert* science users
  - Seven data reduction levels → from raw instrument data advanced science data results and catalogues

Data Flow – From the Cameras to the User



Solving the Data Volume Problem (I)



Array of telescopes

600 Gb/s
(1000 PB/yr)

Reduction at the site

Limited bandwidth

2 Gb/s

2.7 PB/yr
Raw data (both sites)

Note: larger bandwidths may become eventually available

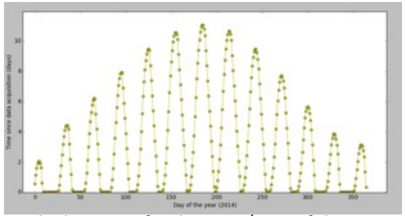
Reduction at the sites

(South site example)

Central trigger → 130 PB/y → 3% full waveform signal, remaining signal integrated → 21 PB/y → Compression, event selection… → ~ 2.7 PB/y

Still need factor 10 reduction due to available bandwidth

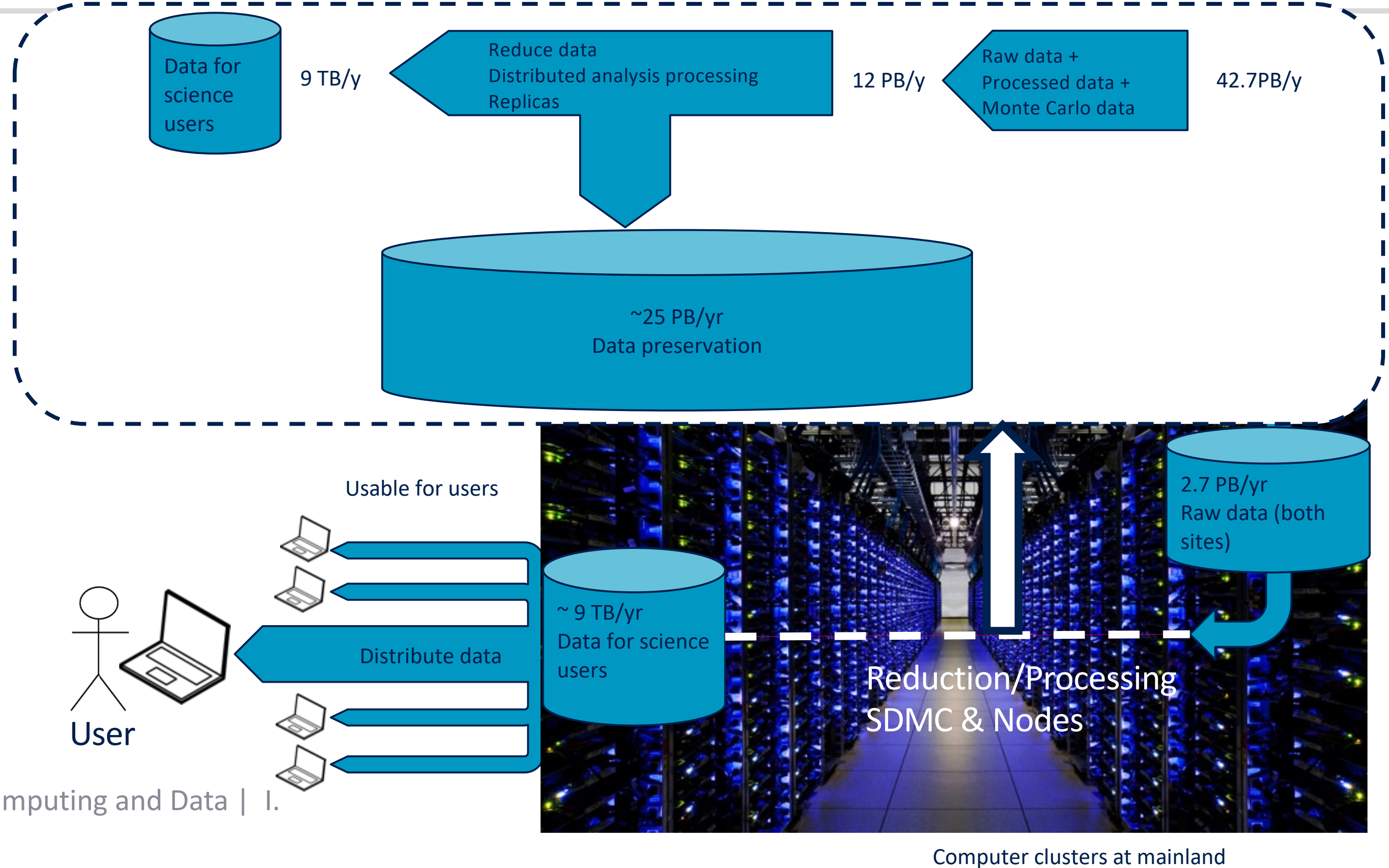
Time since Data Acquisition(days)



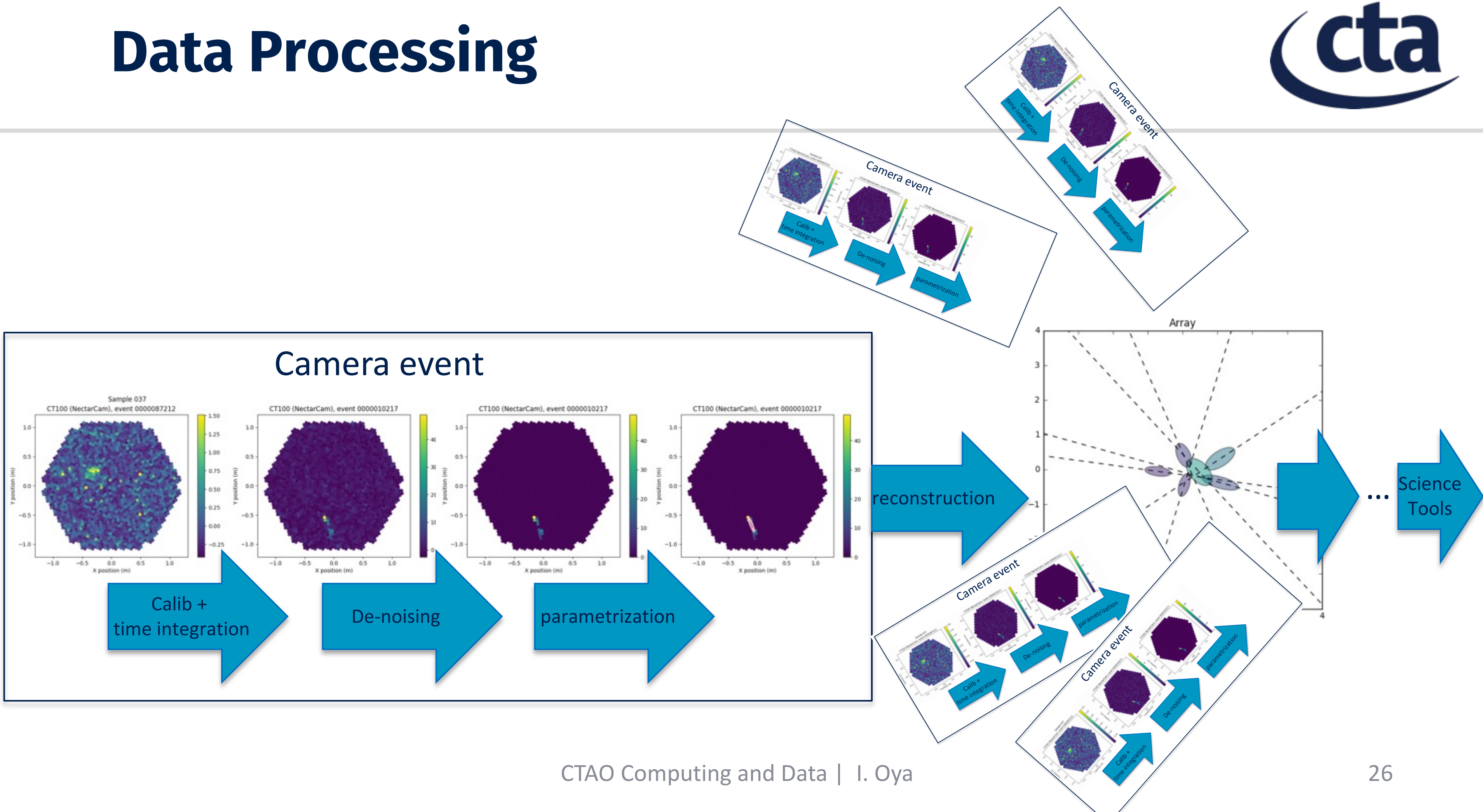
Daily data transfer duration/ Day of the year

Take into account variations in data volume acquire/day of the year…

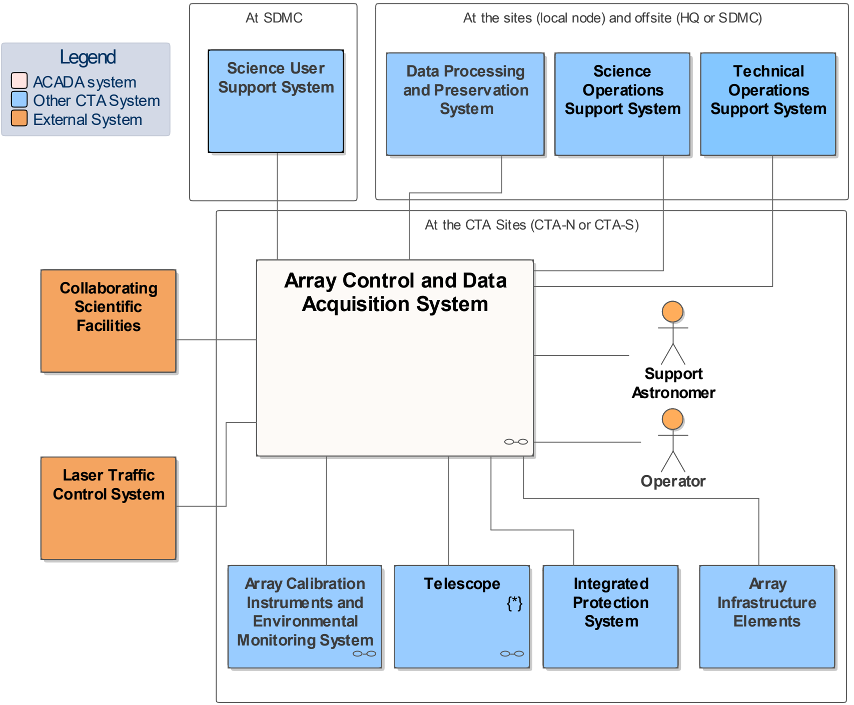
Solving the Data Volume Problem (II)

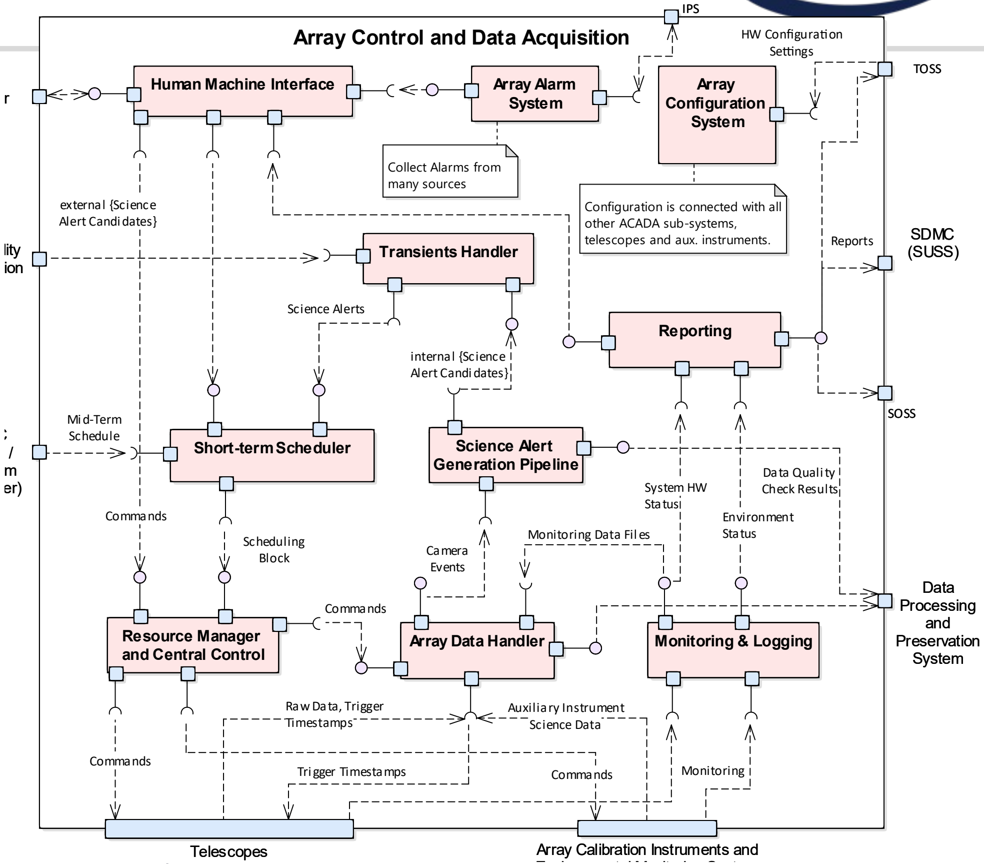


Data Processing



ACADA – Array Control and Data Acquisition



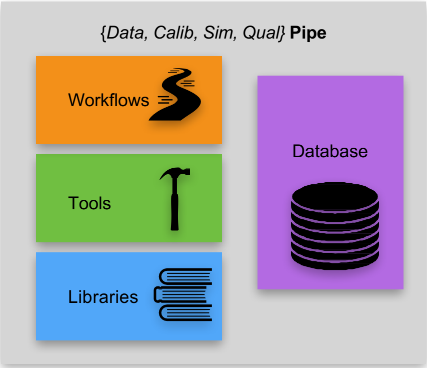


Data Processing and Preservation system

- DPPS Contains
  - Pipelines
    - Process, calibrate, simulate, quality
  - management systems
    - Transfer, archive, Workload, Operations
- Distributed system, made up of:
  - Data Processing and Preservation Nodes
    at multiple data centers (on- and off-site)
  - central management software and interfaces at SDMC
- No single Node is responsible for all data processing or preservation

  → Fault Tolerance + Expandability + Cost Minimization

Inside each Pipeline System:



Example management system technologies

FTS GridFTP

RUCIO GAMAS eXtreme DataCloud

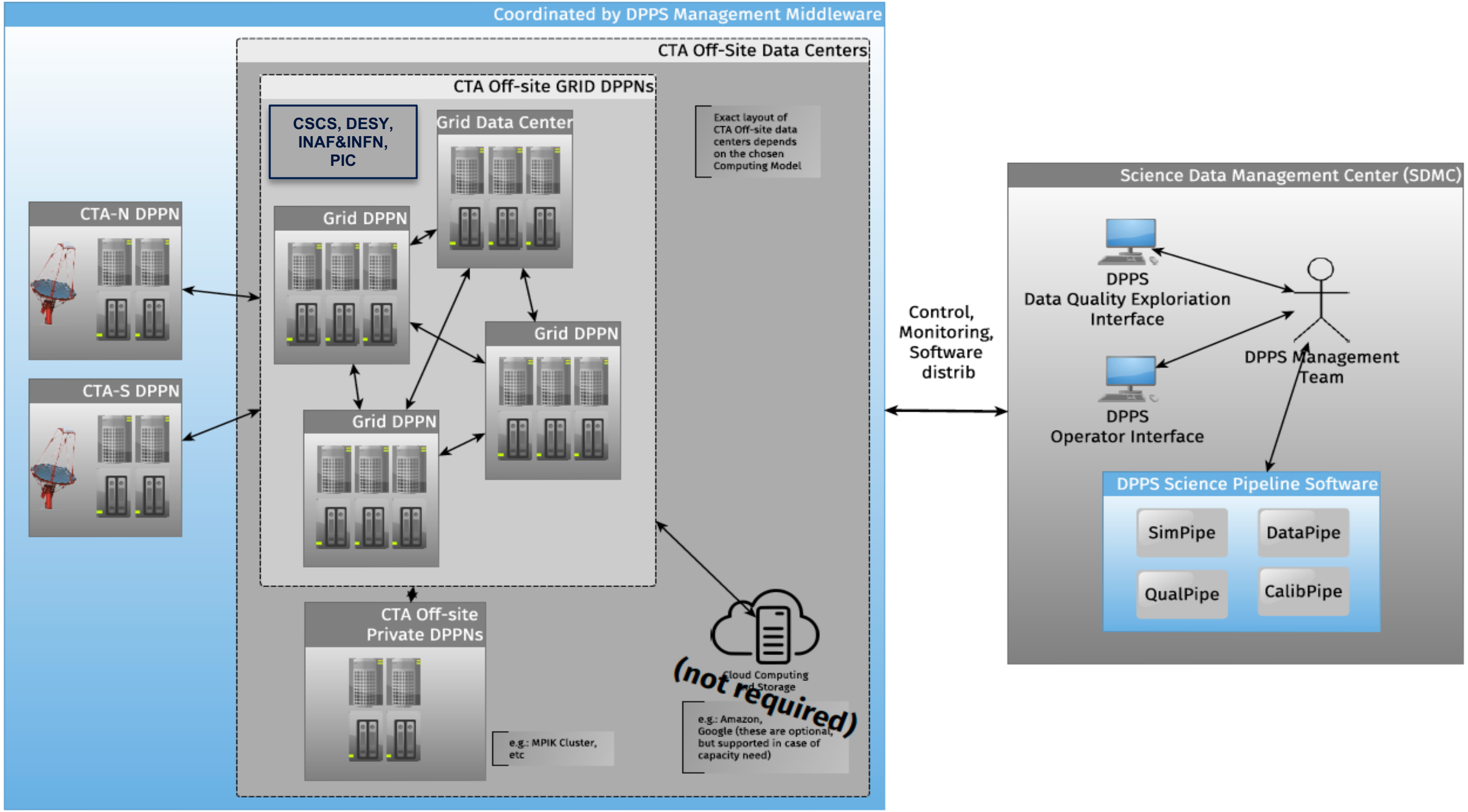
DIRAC THE INTERWARE -DMS

DIRAC THE INTERWARE -WMS

DIRAC THE INTERWARE -web

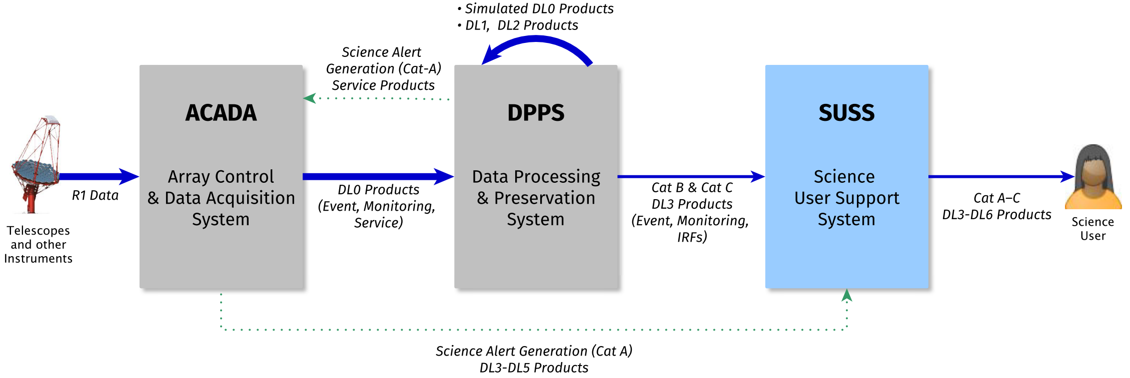
(but no full prototype)

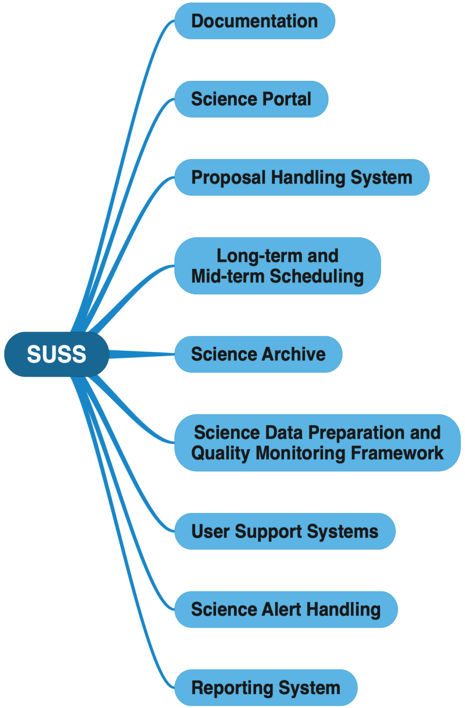
Data Centres and DPPS



Science User Support System

- Support high-level science operations workflows of the CTAO
- provide the Science Users with
  - the Observatory's high-level data products,
  - software products to analyse them,
  - means to request CTAO observations,
  - and all services that support the interactions between the CTAO and its users



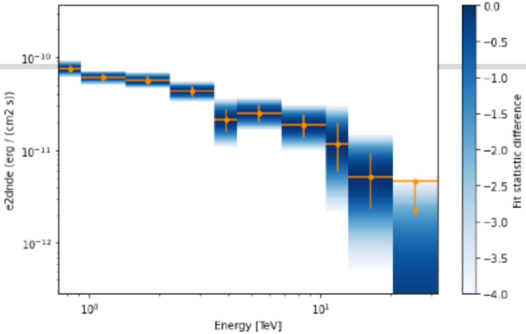


CTAO Science Tools - Based on $\gamma\pi$

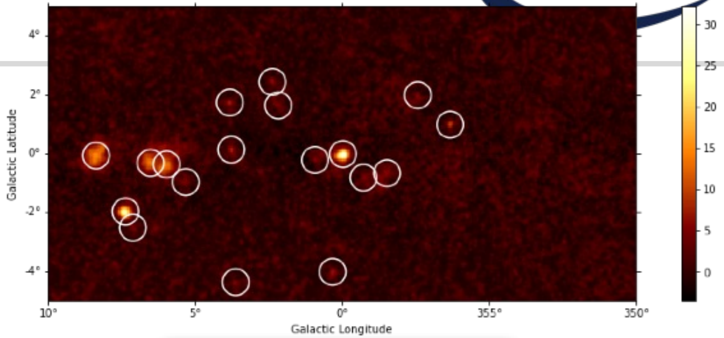
See: Getting started

support for two analysis workflows:

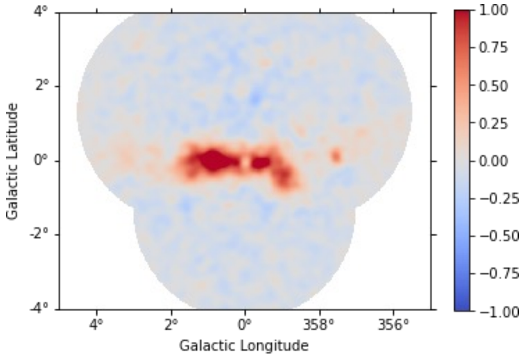
- config-driven high-level interface
- user library API



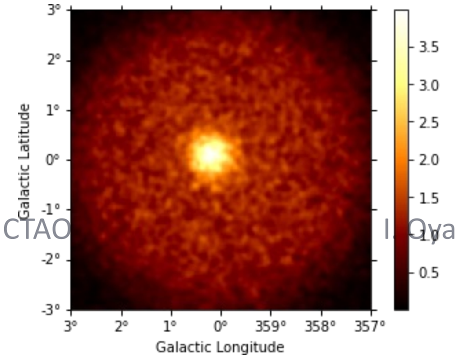
1D spectral analysis



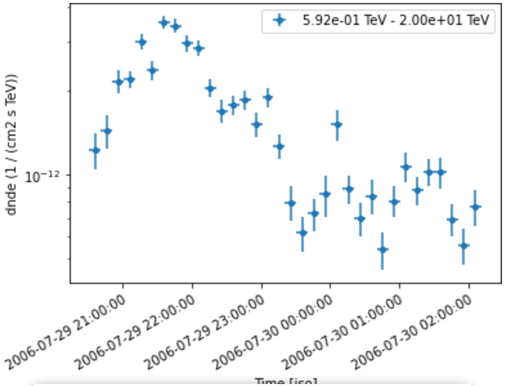
Source detection



3D analysis



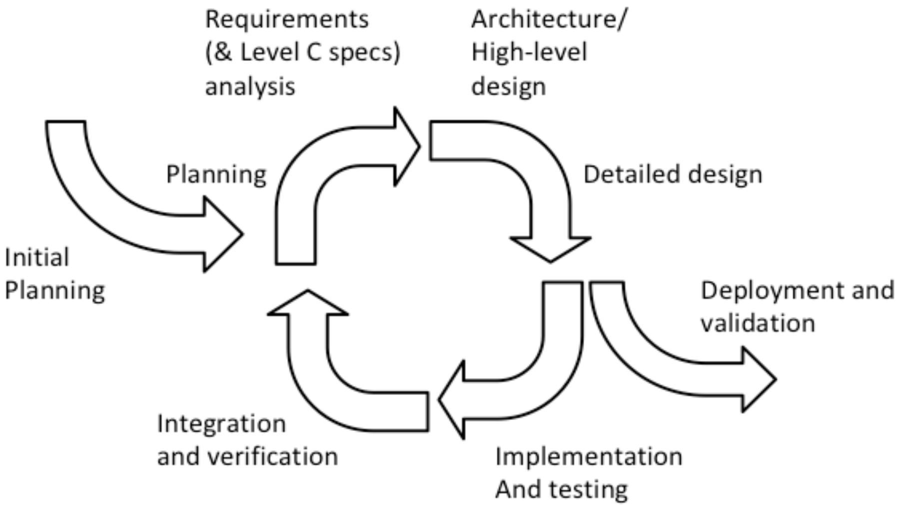
observation simulation



light-curve extraction

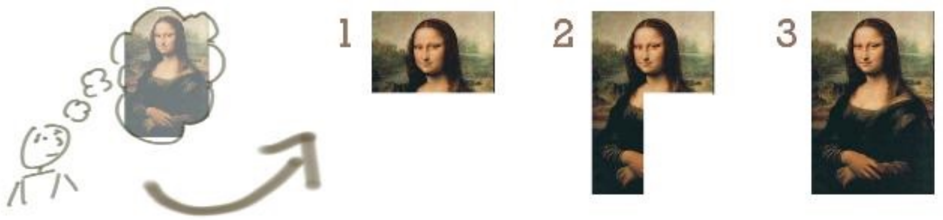
Manufacturing and construction approach – ACADA example

- About 50 people involved in 6 countries and 9 sub-teams
- Incremental and iterative development approach



Incremental and Iterative Software Development life-Cycle (SDLC)

- Incremental development:

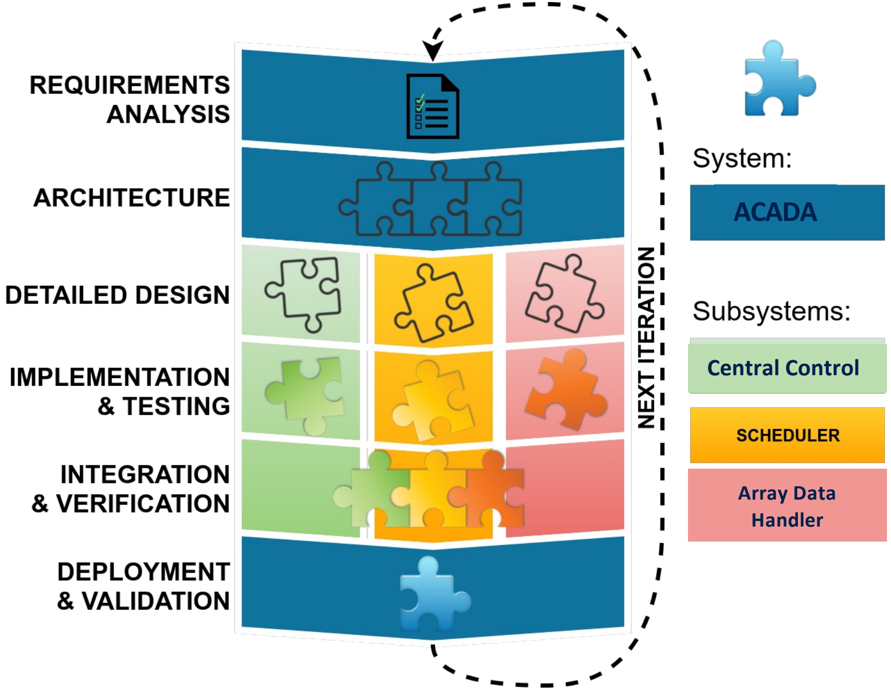


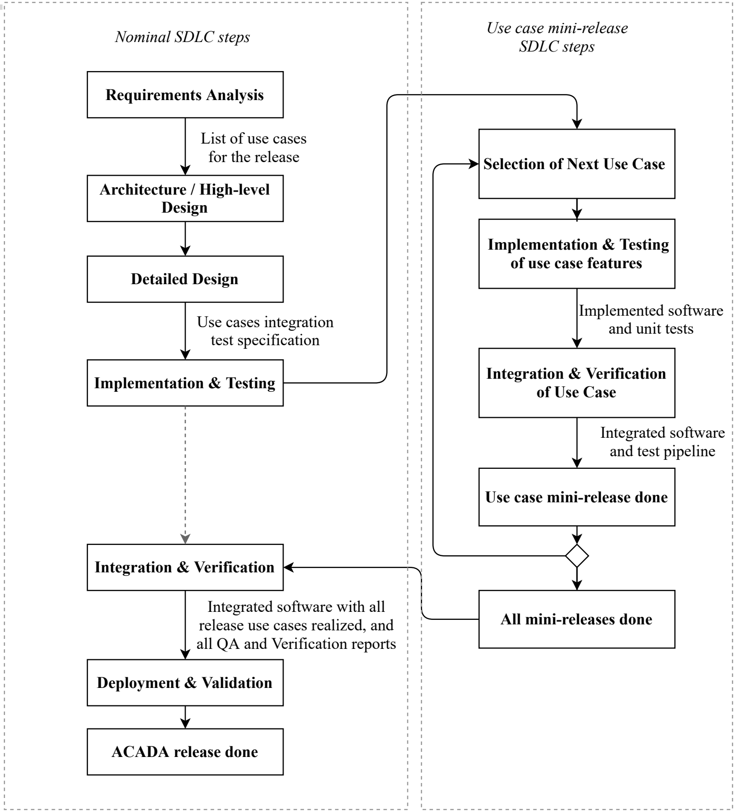
- Iterative development:

Source: Jeff Patton
https://jpattonassociates.com/dont_know_what_i_want/

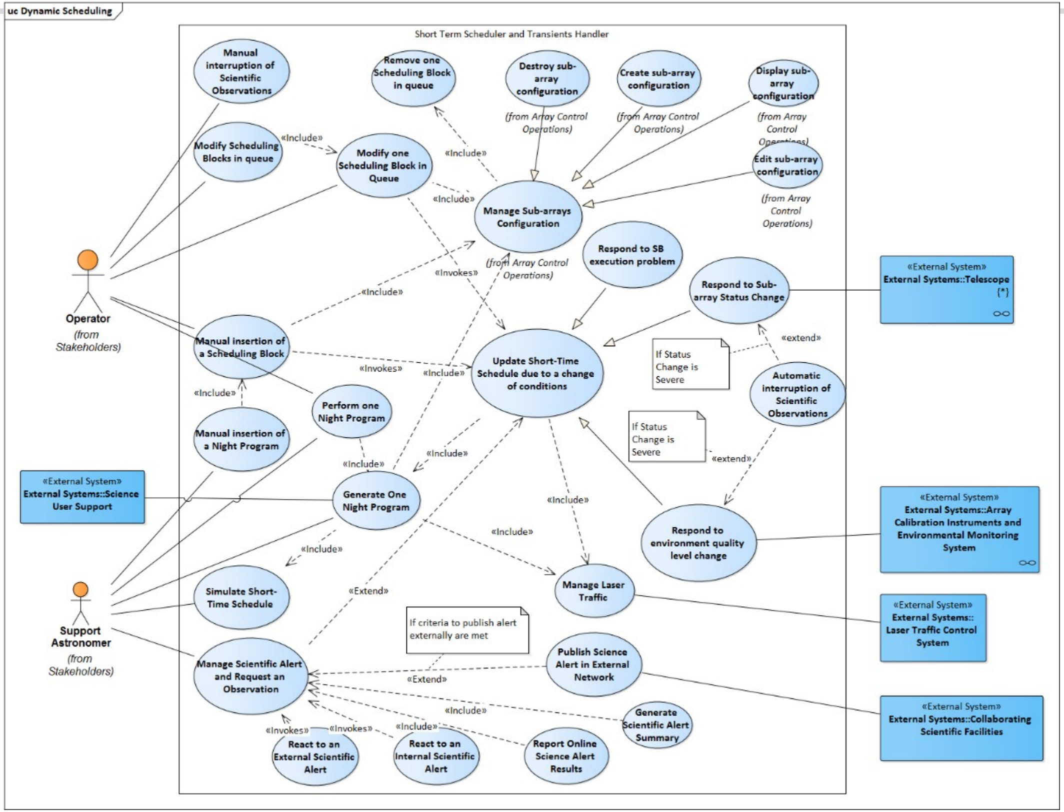
SDLC– ACADA example

Inside every ACADA release…





Use-cases define features of the system



- Each CTAO subsystems has a list of use cases that identifies its main features
- In the case of ACADA, we have around 100 of these features:
  - Acquire Camera data, Execute Observation, Start-up System, etc.

Example of an ACADA use case

ACADA-UC-100-1.2 Perform One Night Program

Summary and Scope: ACADA receives the night program and generates the optimized schedule. The Operator prepares the array for the Observations. When it is time ACADA starts to execute the scheduling Blocks as scheduled. The Operator supervises the execution. When the execution of the last SB is finalized the Operator is informed and a night report is generated.

PRE-CONDITION CONSTRAINT
Pre-Conditions: ACADA is in *Technical* mode. The Warmup procedure has been performed and ACADA is running and is in the operative status (ACADA-UC-150-1.1) Telescopes and Auxiliaries are in the Initialized states The Night Schedule has been generated and loaded. STS has access to the status of the Telescopes and Auxiliaries from the RM (ACADA-UC-180-2.2) STS has access to the weather information from the RM No Scheduling blocks are in execution or in queue.
Trigger: The operator determines that the shift for the Night Observation should start.

SCENARIOS

Basic Path. Main Scenario

1. The Operator commands the initialization of the array (ACADA-UC-150-2.1).
2. A soon as the initialization procedure is ended the STS checks for each SB the availability and operational states of the involved telescopes (all "attached" telescopes are required to be in "Standby" operational state).

Alternate: *2a.* Not every telescope belonging to the sub-array is in *Standby* state but the minimal configuration for the observation is reached. *Join back Main Scenario at 3*

Exception: *2a*. The minimal telescope configuration for the execution of a scheduling block is not reached. *Join back Main Scenario at*

3. STS checks (in parallel) the availability and operational states of the auxiliary instruments (all "attached" auxiliary instruments are required to be in Standby operational state).

Alternate: *3a.* Not every auxiliary device is available, but the minimal configuration is reached for the observation. *Join back Main Scenario at*

Exception: *3a*. The minimal auxiliary configuration for the scheduling block execution is not reached. *Join back Main Scenario at*

4. The Operator asks CC via the HMI to switch to *Science* operational mode (ACADA-UC-200-2.1)

Alternate: *4a*. The operational mode remains set to *Technical*. *Join back Main scenario at*

5. The Operator commands the Startup of the array (ACADA-UC-150-2.2)
6. As soon as the procedure is ended and the RM notifies the STS, the STS checks for the availability and operational state of the telescopes (all "attached" telescopes are required to be in "Ready" operational state).

Alternate: *6a.* Not every telescope belonging to the sub-array is in *Ready* state, but the minimal configuration is available fo the observation. *Join back Main Scenario at*

Exception: *6a*. The minimal telescope configuration for the execution of the scheduling block is not reached. *Join back Main Scenario at*

7. The Operator is informed that the Array Elements arrived at the *Ready* state.
8. If the resource availability has changed during the array start up procedure the STS updates the SBs execution list based the availability of resources, execution time (start and stop), environmental conditions, laser traffic program and science Alerts contained in the schedule (ACADA-UC-100-2.1).

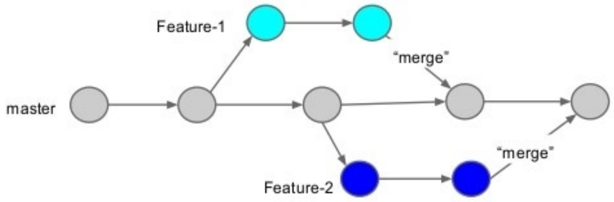
More details omitted …

Using Gitlab for work organization (I)

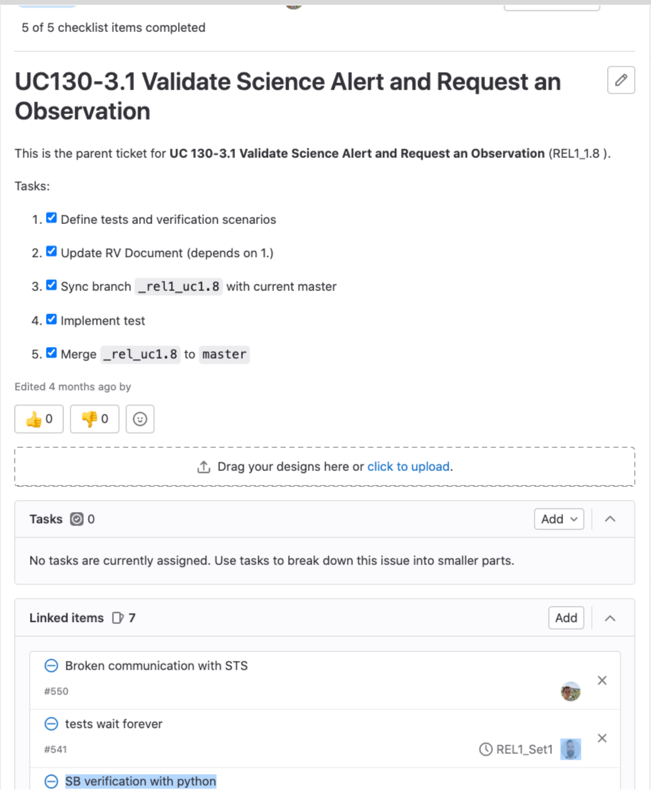
- Use GitLab tickets with priority
- “Main” Tickets to follow-up the implementation
- Finalized merge-request with passed tests marks the finalization of the task

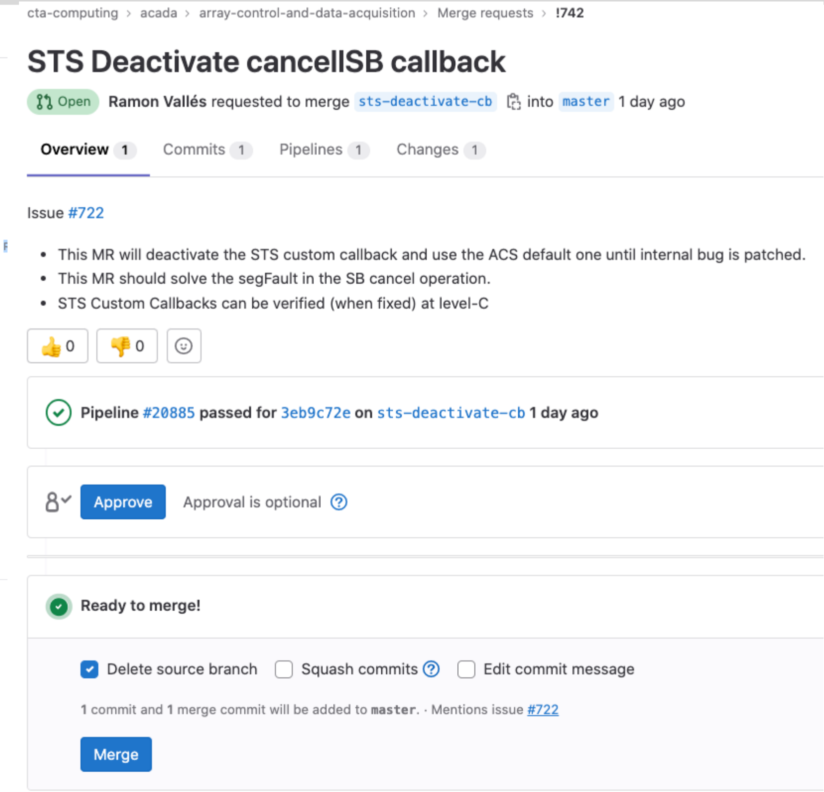
Feature Branch Workflow

- Separate branches for feature development
- Merge back to master when feature ready



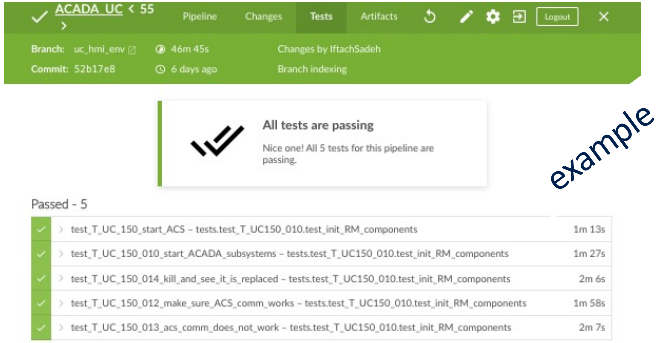
Using Gitlab for work organization (II)

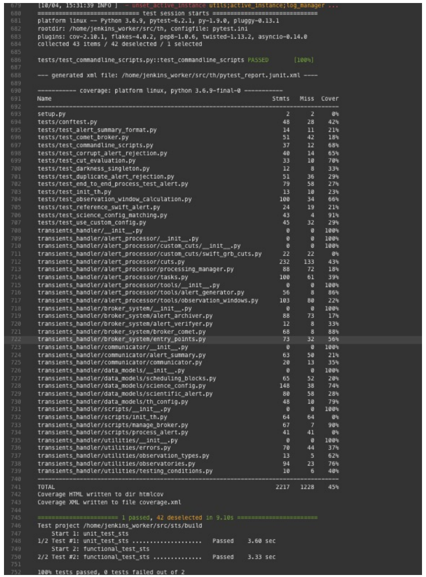


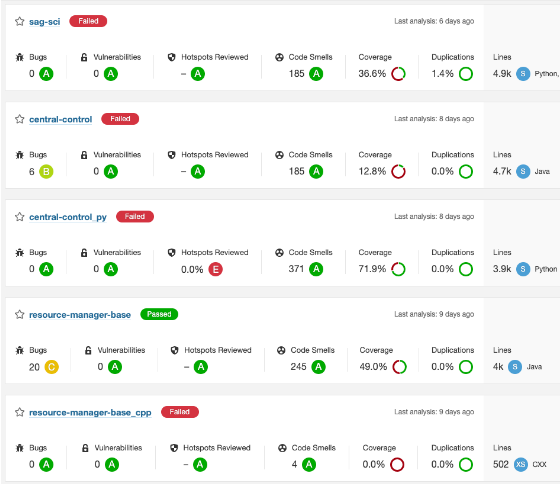


Testing and Quality assurance - ACADA

- Every use-case/requirement has its own test(s)
- In addition, lower-level unit tests
- Certain level of code test coverage must be reached for every release
- Static code analysis quality gates
- Quick, nightly, and long test suites
- Using Jenkins and SonarQube instances provided by DESY in the CTAO Tests Cluster

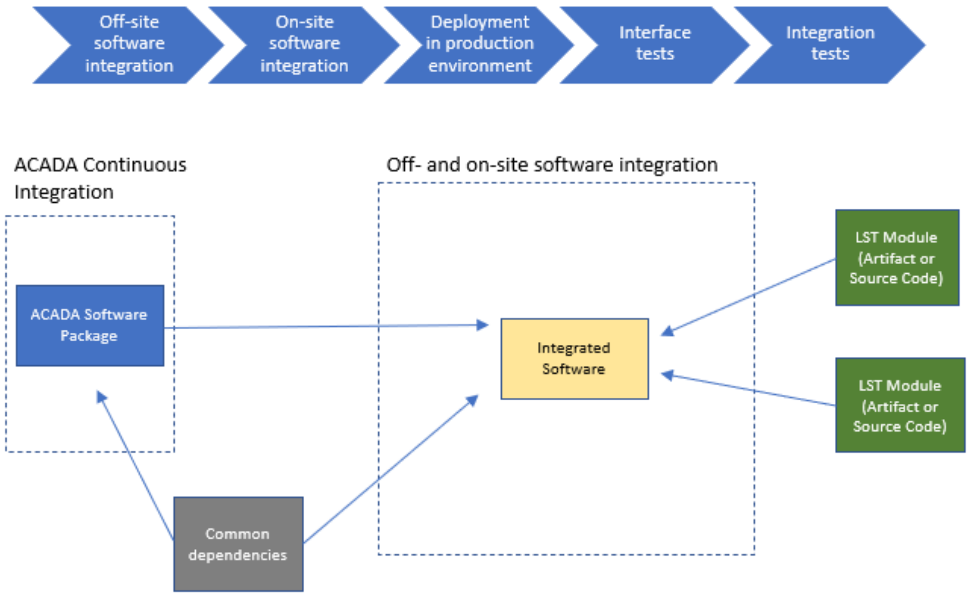






Software and hardware integration

- Test and integrate before deploying into the production environment
- Getting hardware/production environment time is never easy
- Travelling to the sites is expensive, time consuming, and bad for the environment
- There are some things you may not be able to test easily with the instruments:
  - E.g. error and alarm conditions
- <u>Rely on mock-ups and simulators</u>
- Using the CTAO SDMC test cluster at DESY
- We plan to test the ACADA release 1 with the first CTA telescope at La Palma this summer



Role of the CTAO Science Data Management Centre at Zeuthen

- In charge of science operations and making CTAO's science products available to the worldwide community
- Operate CTAO offsite science operations and data management software. In particular:
  - Data Processing and Preservation System (DPPS)
  - Science User Support System (SUSS)
- Maintain and support all CTAO software

CTAO SDMC – Now

- 8 CTAO staff already allocated at the “Seevilla” building on campus
- Other Computing Department colleagues in remote
- CTAO Software coordination

CTAO SDMC and DESY Interfaces

- DESY is leading several ACADA subsystems
- DESY providing the GitLab, test cluster and other services to CTAO
- DESY's datacentre is one of the four official CTAO Offsite data centres.
- DESY is supporting DPPS, SUSS, Onsite-ICT, Clock system via:
  - In-kind contributions
  - Seconded roles
- ½ of us are ex-DESY*ers*

Conclusions and Outlook

- CTAO will work as an observatory
  - Service work & commitment
- CTAO requires well designed software systems in order to manage its many telescopes <u>and the data</u> they will produce as a single efficient system
  - An architecture-driven solution is fundamental
  - Data modeling, Software frameworks, standard technologies and high-performance solutions are required
- Several PB/y raw data to be handled
- CTAO science operations will be managed by the SDMC at Zeuthen
- CTAO software in construction and first versions are being released
- Will perform the first integration with a telescope on summer 2023