# The ASTRI Mini-Array Supervisory Control and Data Acquisition software system


**Andrea Bulgarelli – INAF/OAS Bologna**

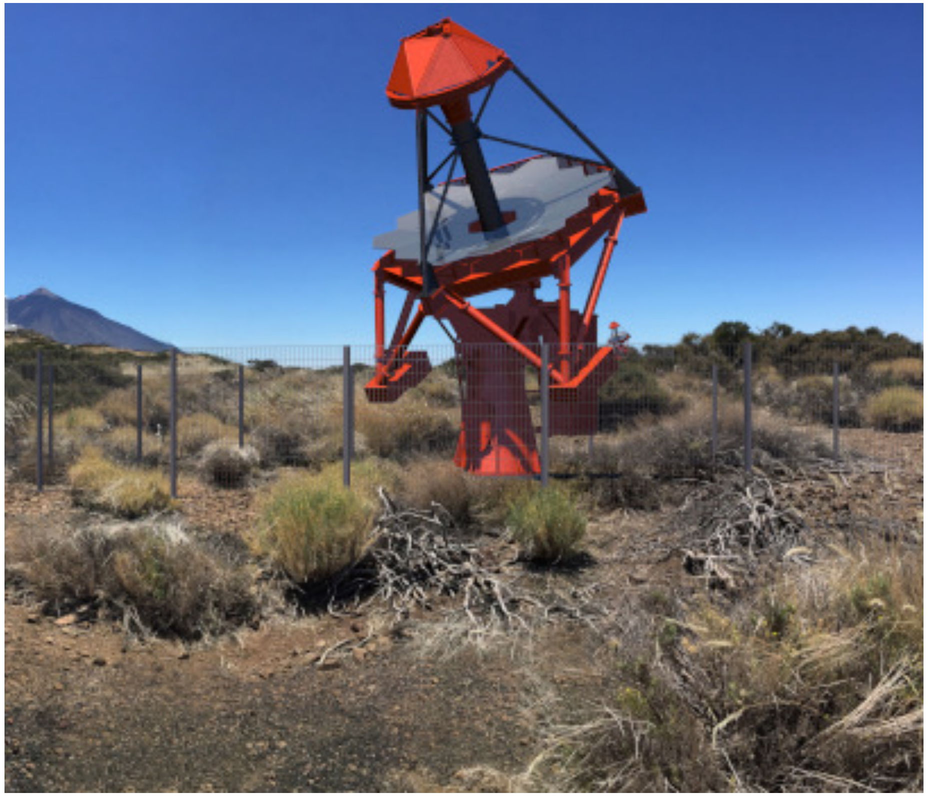
for the ASTRI Project



**USC VIII General Assembly, Galzignano (PD) 16 Oct 2024**
**https://indico.ict.inaf.it/event/2870/**


# Outline

- The ASTRI Mini-Array Project
- The ASTRI Mini-Array System
- Software Architecture
- Software development approach



# The ASTRI Project

ASTRI Mini-Array

**ASTRI: AStrofisica con Tecnologia Replicante Italiana -> Astrophysics with Italian Replicating Technology**

**ASTRI-Horn Prototype**

INAF Project funded by Italian Ministry of Research

End-to-end prototype installed and operational on Mount Etna (Sicily, Italy)

**First detection of a gamma-ray source** (Crab Nebula) **with a dual-mirror, Schwarzschild-Couder Cherenkov telescope** (Lombardi et al., 2020)

Nanni Bignami

## Array Mini-Array

INAF-led Project with several partners:

Univ. of Sao Paulo/FPESP (Brazil), North-West Univ. (S. Africa), IAC (Spain), FGG, ASI/SSDC, Univ. Geneva, Univ. of Padova, Perugia

**Array of 9 ASTRI telescopes being deployed at the *Observatorio del Teide*** (Tenerife, Spain) in collaboration with IAC and FGG-INAF.

**First 4 years → *Core Science***, following 4 years of *Observatory Science.* **Full Science operation → 2026**

# The ASTRI Mini-Array in a nutshell

Mini-Array

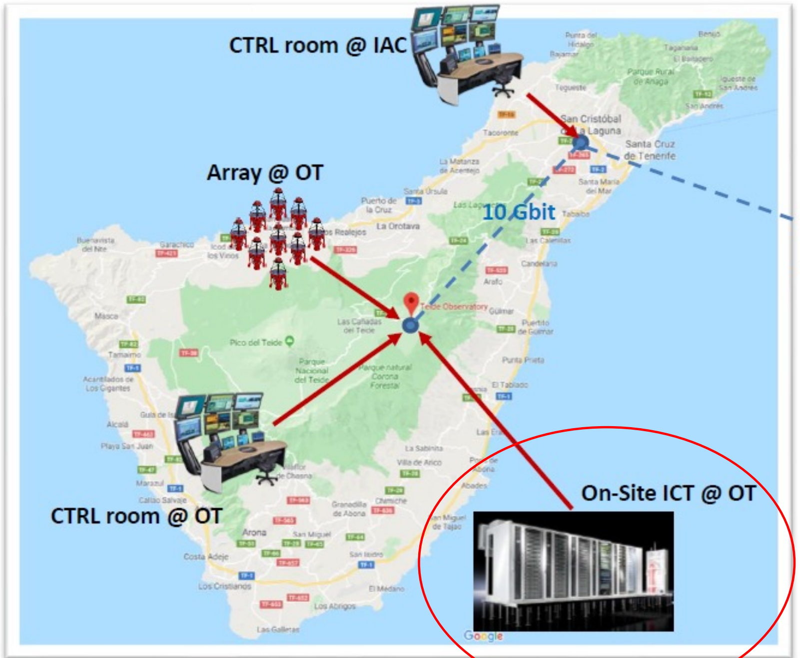
**ASTRI Mini-Array Observing Site (AOS)**

- Teide Observatory: ASTRI Mini-Array system.
- IACTEC in La Laguna: Array Operation Center.

**ASTRI Mini-Array in Italy**

- Data Center in Rome.
- Remote Array Operation Centers.

- ASTRI software allows the Mini-Array to be **operated remotely from the Array Operation Centers (AOCs)**
- **Off-site** at the Data Center in Rome
  - **stereo trigger;**
  - **data processing.**
- **On-site** is foreseen a **quick-look** of data at telescope level




Gianotti's talk on Friday

10 Gbit


A. Bulgarelli

# The ASTRI Mini-Array System

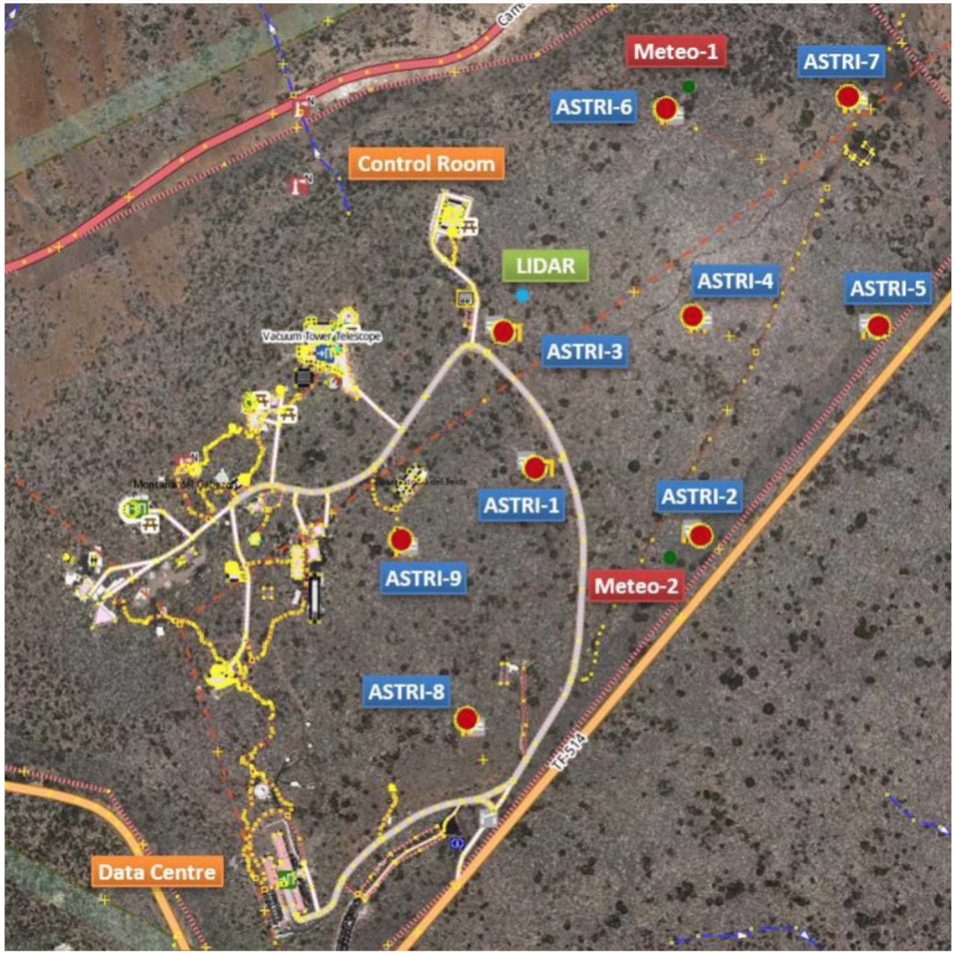
**Array System, nine Telescopes** with their hardware assemblies, including **Cherenkov Camera** and **Stellar Intensity Interferometry Instrument**.

- **ASTRI-1 to ASTRI-9**

**Atmosphere Characterisation System**

- **1 LIDAR** (LIght Detection And Ranging) atmospheric composition, structure, clouds and aerosols
- **3 SQM** (Sky Quality Meter): brightness of the night sky.
- **1 UVSiPM**: diffuse night sky background.

ASTRI software monitors and control all hardware **assemblies** of the ASTRI Mini-Array.




The ASTRI Mini-Array at the Teide Observatory

**Telescope Power Management System**

- **Transformer Station**

**Information Communication Technology**

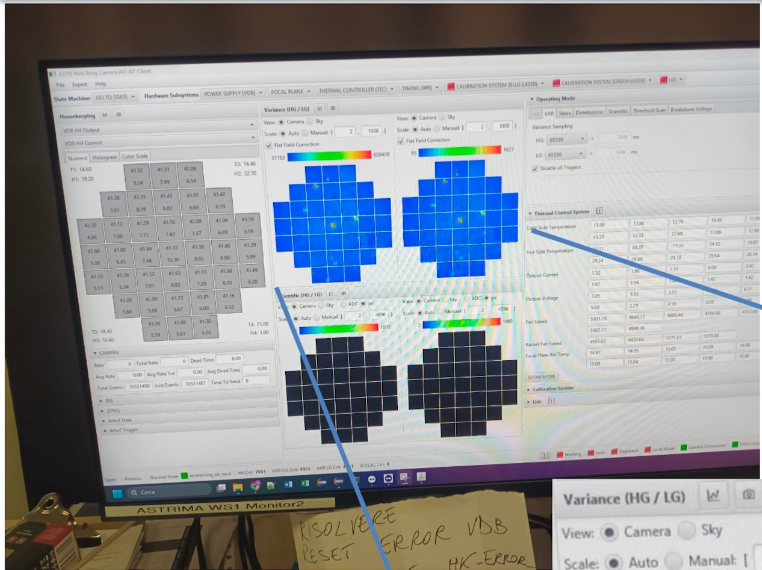
- **On-site Data Centre**
- **Control Room**

**Safety and Security system**

**Environmental Monitoring System**

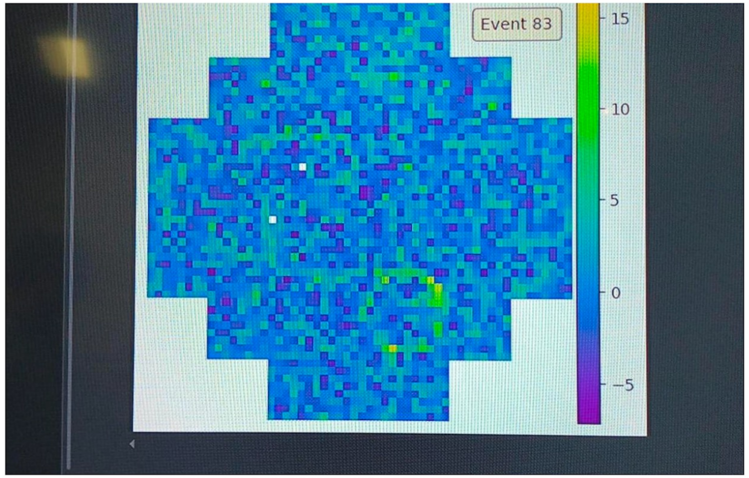
- Weather Stations
  - Meteo-1
  - Meteo-2
- Humidity and rain sensors;
- All-sky camera: cloud coverage

# ASTRI latest achievements



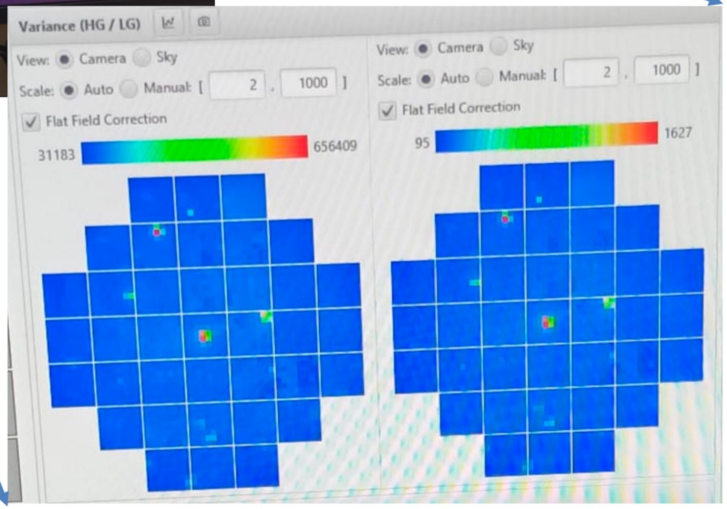
## ASTRI-1 first light !

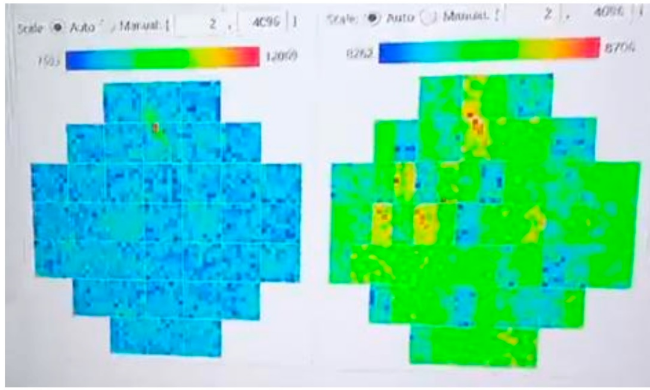
## Sep 2$^{nd}$, 2024




**First muon ring**

Some stars clearly visible in the variance monitor, both in the High and Low Gain channels

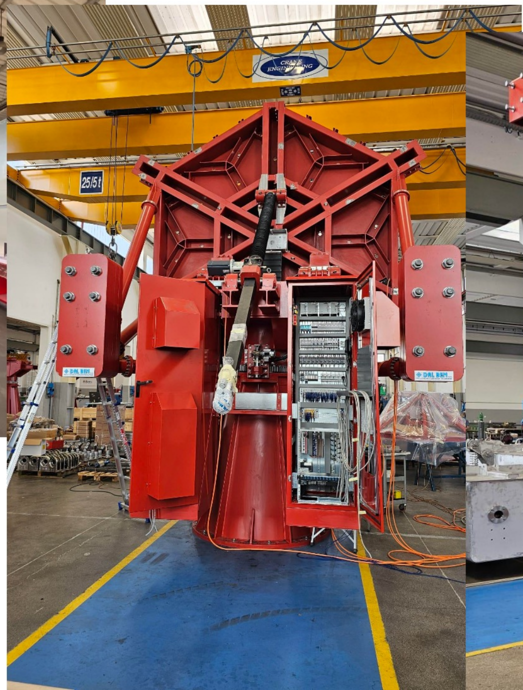


**First cosmic-ray event**

# The remaining 8 telescope structure are on their way

ASTRI Mini-Array





Two cameras in production

CREDITS: Dal Ben SpA and EIE GROUP SRL

# ASTRI Timeline

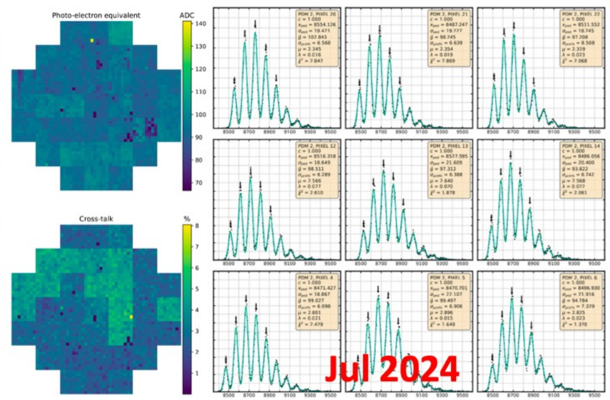
Mini-Array

- Site infrastructure completed
- ASTRI-1 telescope accepted
- Engineering camera on ASTRI-1 delivered at end of July 2024
- On-site ICT delivered mid-October 2024

- First scientific observation October/December 2024
- SCADA ORR November/December 2024

- Two more cameras completed by Spring 2025
- Ready for commissioning late summer 2025
- Scientific operations will start with 3 telescope in 2025
- All telescope completed in 2026

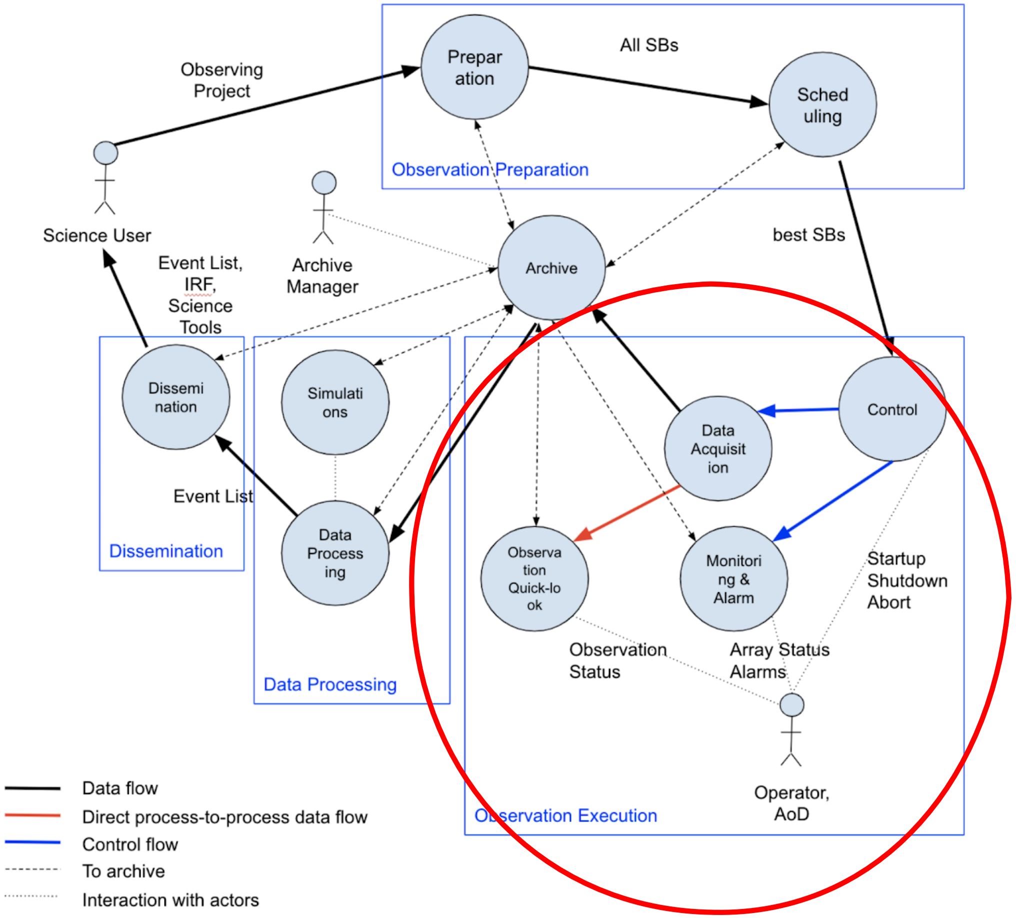



Eager to collaborate with existing and future gamma-ray facilities!
(MAGIC, HESS, VERITAS, HAWC, LHAASO/LACT, CTAO, SWGO…)

# Observing cycle

Mini-Array

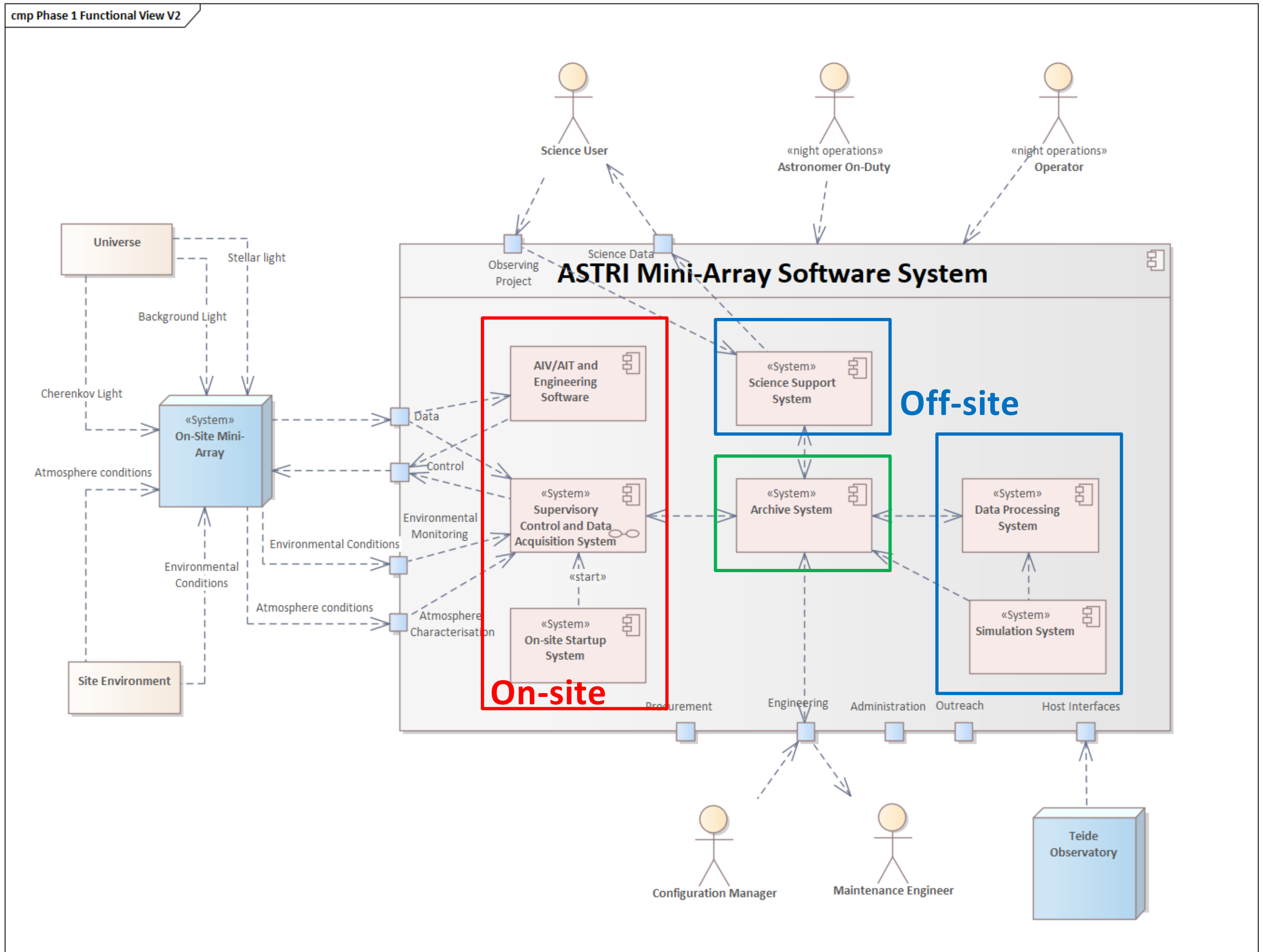



ASTRI software support the full observing cycle

The ASTRI Mini-Array software is envisioned to handle an observing cycle, i.e. the end-to-end control and data flow system. The observing cycle can be divided into the following main phases:

1. Observation preparation: submitting an Observing Project that is turned into Scheduling Blocks (SBs) and Observing Blocks (OBs)
2. Observation execution: execution of Observing Blocks
3. Data Processing: produces calibrated and reconstructed data
4. Dissemination: Data and Science Tools are distributed for a scientific analysis of the Observing Projects.

# General software architecture

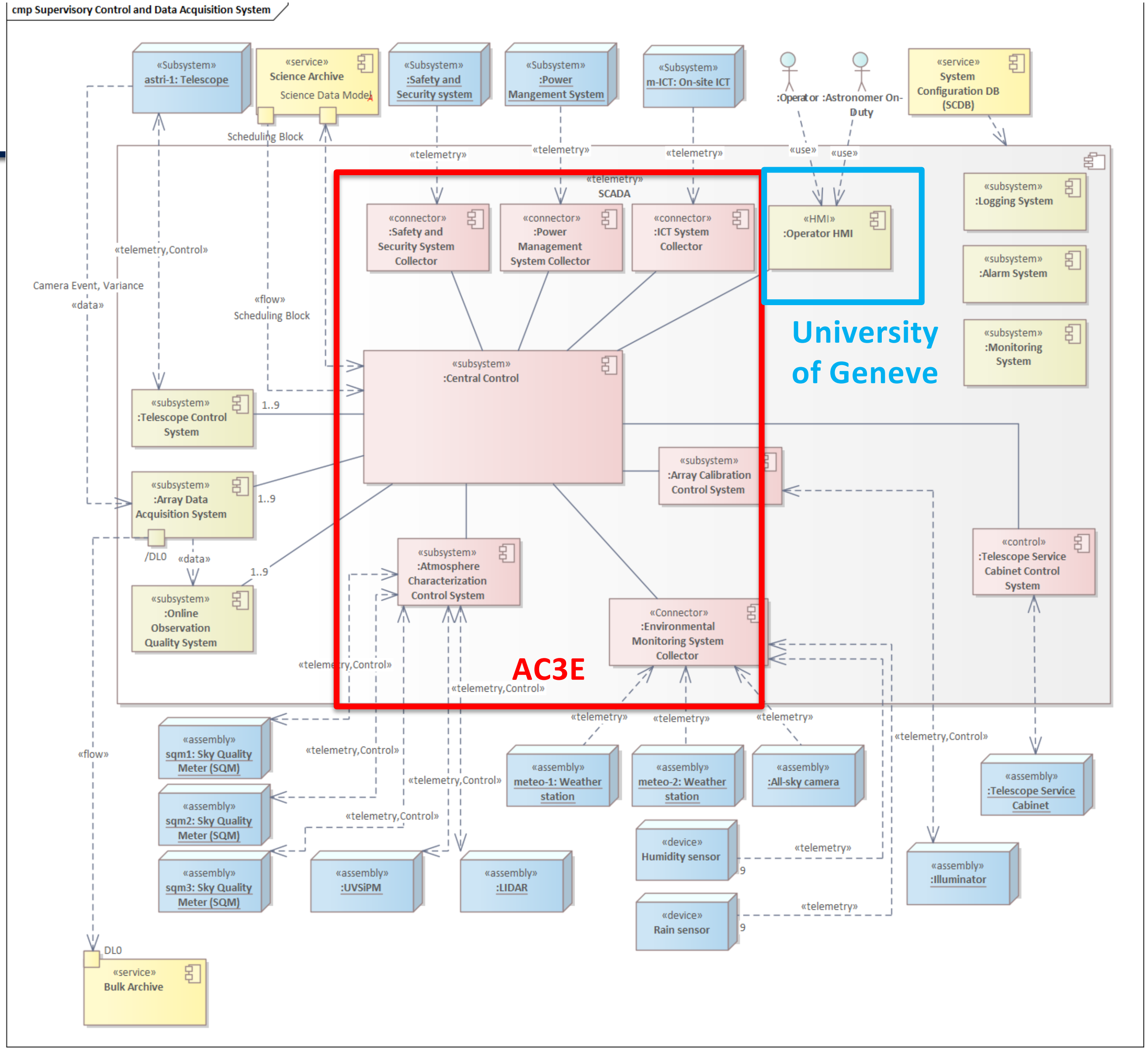



- **Supervisory Control And Data Acquisition (SCADA) System**: *interfaces, controls, monitors, acquires data, manage alarms of all software subsystems and assemblies of the Mini-Array*
- **On-site Startup System**
- **AIV/AIT and Engineering Software**
- **Archive System**
- **Data Processing System**
- **Science Support System**
- **Simulations System**

# Supervisory Control and Data Acquisition System (SCADA)

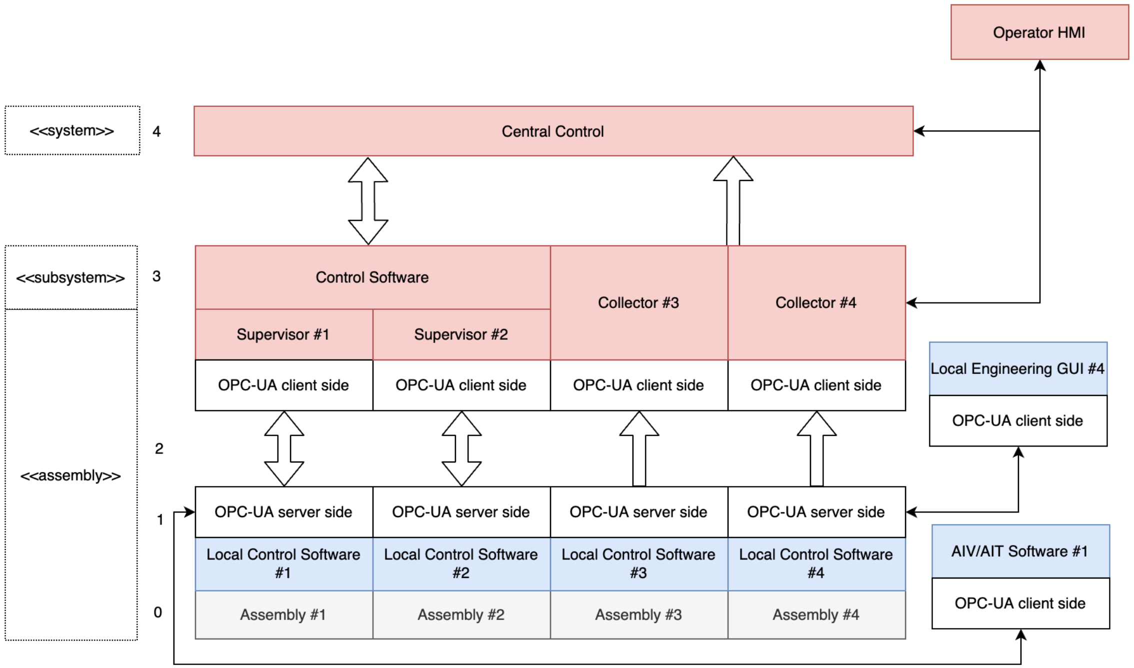
- **SCADA interfaces, controls, monitors, acquires data, manage alarms of all software subsystems and assemblies of the Mini-Array:**
  - automatically execute the whole sequence of operations needed to perform an observation;
  - react to critical conditions in an automatic way to put the array system in a safe state.
- **Software subsystems:**
  - Central Control;
  - Control systems and Collector;
  - Monitoring/Alarm/Logging systems;
  - Data Acquisition and Quality systems;
- Supervised by an Operator with an HMI

Red and green are SCADA subsystem; blue boxes are the ASTRI Mini-Array assemblies; yellow boxes are part of the Archive System.
The << telemetry >> stereotype represents monitoring points, alarms, errors, logs, and status information, << data >> stereotype represents the data flow. The << control >> stereotype represents the control flow.



# Supervisory Control and Data Acquisition System (SCADA)/2

- **Central Control System** coordinates the sequence of operations, all software components and hardware assemblies, the sequences of the startup, shutdown, configuration and the status.
- Control systems, to control, monitor, and manage alarms and the status of the telescopes (**Telescope Control System**, developed by INAF based on the ASTRI-Horn experience), the assemblies used to characterise the atmosphere (**Atmosphere Characterisation Control System**), the calibration system (**Array Calibration Control System**), the telescope service cabinets **(Telescope Service Cabinet Control System);**
- Collectors, to monitor and determine alarms and the status of environmental devices (**Environmental Monitoring System Collector**), of the Information and Communication Technology (ICT) system (**On- site ICT System Collector**), the power system (**Power Management System Collector**), the Safe and Security System (**Safety and Security System Collector**);
- **Operator Human Machine Interface (HMI),** the user interface for the Operator, including an Operator Logbook to save logs of the observations during the night.



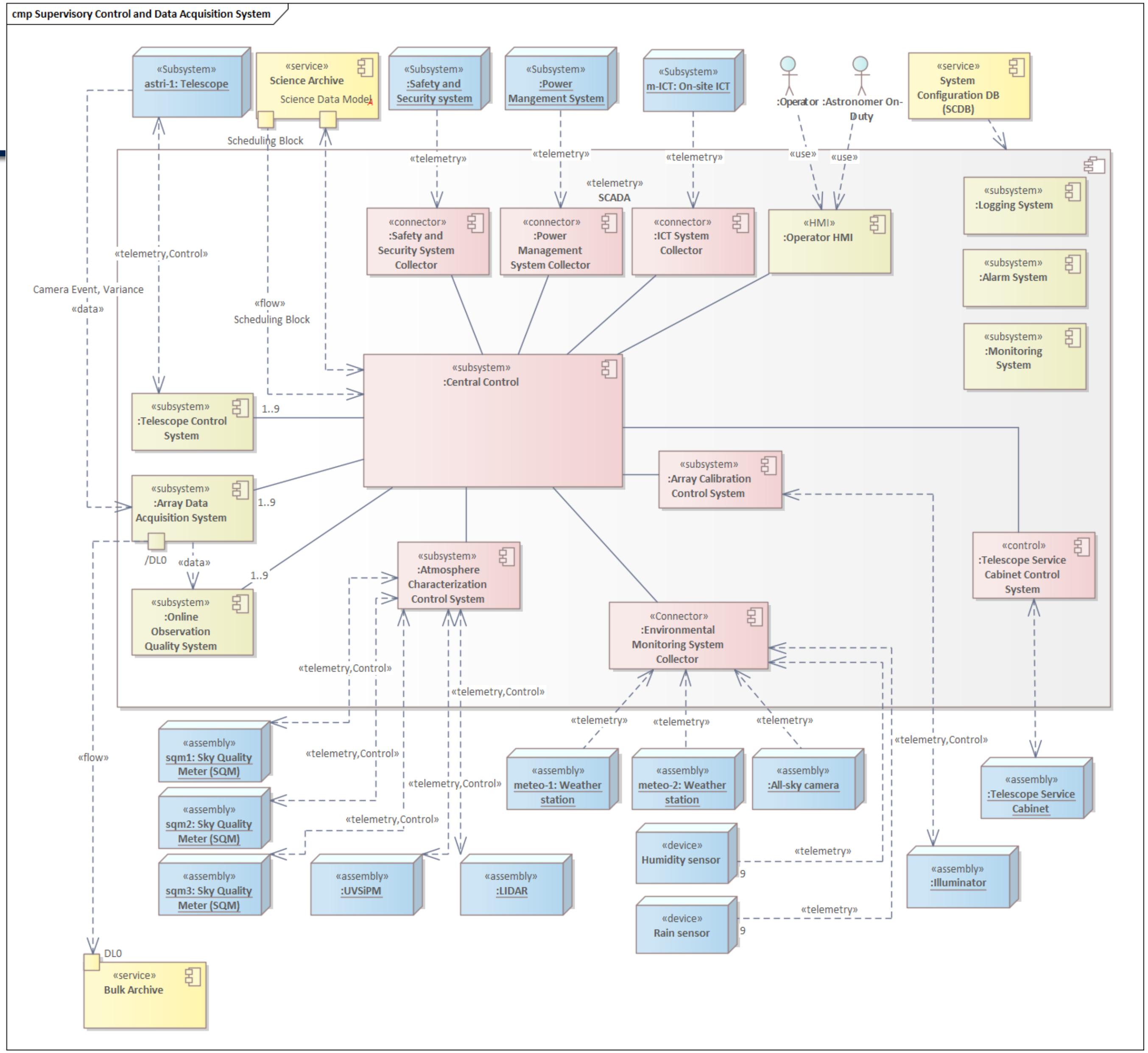

Interfaces between hardware assemblies and SCADA: IEC 62541 standard OPC Unified Architecture protocol, plus

- SNMP for the ICT system
- Modbus for Power Management System

# Supervisory Control and Data Acquisition System (SCADA)/3

- **Logging System, Monitoring System and Alarm System**, developed by INAF, monitor the overall performance of the systems through the acquisition of environmental, monitoring and logging points and alarms from instruments and generates status reports or notifications to the Operator.
- **Telescope Control System**, to control a single telescope, developed by INAF
- **Array Data Acquisition System**, developed by INAF, acquires Cherenkov Cameras and Stellar Intensity Interferometry Instruments data
- **Online Observation Quality System**, developed by INAF, focuses on ongoing problems and the status of the observations

Red and green are SCADA subsystem; blue boxes are the ASTRI Mini-Array assemblies; yellow boxes are part of the Archive System.
The << telemetry >> stereotype represents monitoring points, alarms, errors, logs, and status information, << data >> stereotype represents the data flow. The << control >> stereotype represents the control flow.



# SCADA and software frameworks

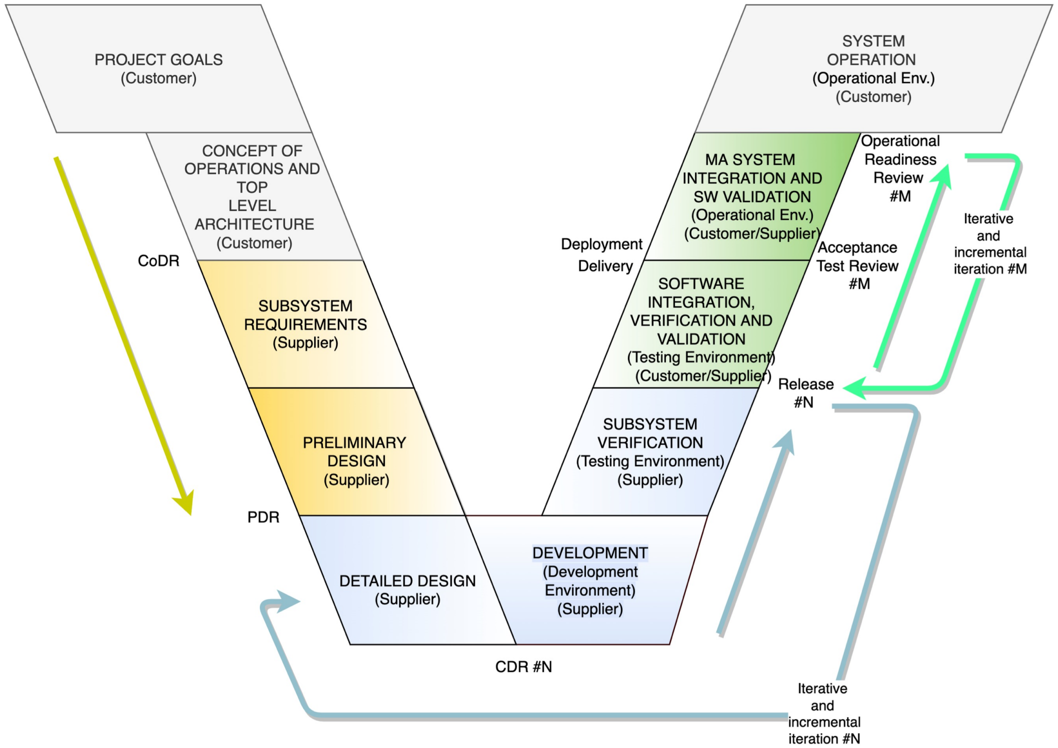
Mini-Array

- SCADA is developed using ALMA Common Software (ACS)†, Kafka, Cassandra, MongoDB.
  - ALMA Common Software (ACS): a container component framework, designed for distributed systems, with standardized paradigms for logging, alarms, location transparency, and support for multiple programming languages: Java, C++ and Python.
    - ACS has been used successfully for ALMA, which manages an array of 66 antennas on the Chajnantor plateau in Chile.
    - ACS is used for ASTRI-Horn, Sardinia Radio Telescope, CTAO.
  - Kafka is used as a data backbone for all SCADA subsystems, except for data acquisition that uses TCP/IP.
  - Cassandra is used for the permanent storage of monitoring and data quality information, synchronized with off-site data center
  - MongoDB for observing plans and system configuration

†https://confluence.alma.cl/display/ICTACS/ACS+Documentation

# Software development life cycle (SDLC): general

- The high complexity of the software management is due to:
  - the high number of hardware assemblies;
  - the high number of software subsystems;
  - many teams
- **Software Development Plan: agile development methodologies**, including:
  - **frequent iterations and releases**;
  - **use case-driven development approach**;
  - **verification**: automated testing and continuous integration;
  - formal reviews when necessary.
- Integrated with **Verification and Validation Test plans** and **Quality Assurance Plan.**




ASTRI Mini-Array Software life-cycle and reviews
V + iterations with incremental releases
use case-driven
Automated verification with CI

# Team organisation

- INAF (OAS, OACT), UNIPG
  - Coordination
  - Telescope Control System
  - Array Data Acquisition System
  - Online Observation Quality System
  - Monitoring, Logging, Alarm system
  - Test bed + ICT

AC3E

Central Control

Startup System

Integration, deployment

University of Geneve

Operator HMI

many teams

# Releases

| SCADA | Dev | Int. and Delivery | Ver@testbed | Deployment@Teide | Ver@Teide | Val@Teide | Release |
|---|---|---|---|---|---|---|---|
| v0.1.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v0.2.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v0.3.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v0.3.1 | ✅ | ✅ | ✅ | ✅🟠 #102 (closed) | ✅🟠 #101 (closed) | ❌ | ✅ |
| v0.3.2 | ✅ | ✅ | ✅ | ✅ interface-control-document!18 (merged) | ✅ #132 (closed) monitoring-system#29 (closed) power-management-system-collector#5 (closed) | ❌ | ✅ |
| v0.4.0 | ✅ | ✅ | ✅ | ✅ | ✅🟠 telescope-control-system#19 (closed) #145 (closed) | ❌ | ✅ |
| v0.4.1 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v0.5.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v0.6.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v0.7.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v1.0.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| v1.1.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ |
| v1.2.0 | ✅ | ✅ | ✅ | ❌ | ❌ | ❌ | ✅ |
| v1.3.0 | ✅ | ✅ | ✅ | ❌ | ❌ | ❌ | ✅ |
| v2.0.0 | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| v2.0.1 | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| v2.0.2 | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| v2.1.0 | ✅ | ⏳ | | | | | |
| v2.2.0 | ⏳ | | | | | | |

**Frequent iterations and releases: 17 integrated releases since Nov 2022 -> one each 1.5 months**

**It is normal for the development and integration/deployment phases to overlap because they are carried out by different teams.**

| | | | | |
|---|---|---|---|---|
| v2.0.1 | ✅ | ✅ | ✅ | ✅ |
| v2.0.2 | ✅ | ✅ | ✅ | ✅ |
| v2.1.0 | ✅ | ⏳ | | |
| v2.2.0 | ⏳ | | | |

# Releases /2

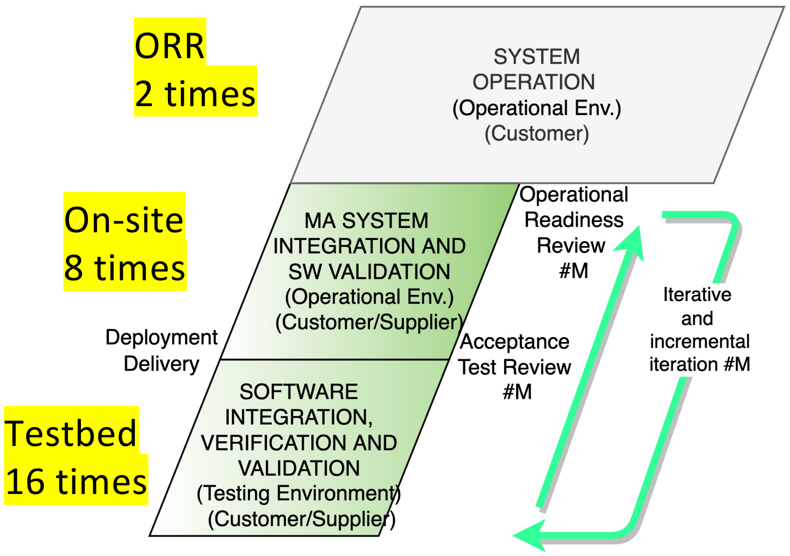
- **SCADA v0.1.0 Released** Feb 28, 2023
- **SCADA v1.0.0 Released** Mar 05, 2024
- **SCADA v2.0.0 Released** Sep 5, 2024

**frequent iterations and releases**

- **SCADA v2.1.0 (WIP)**
  - Deployment @ Teide: Oct 20, 2024
- **SCADA v2.2.0**
  - Deployment @ Teide: Nov 5, 2024 — 1 telescope + camera
- **SCADA v2.3.0**
  - Deployment @ Teide: Jan 21, 2024
- **SCADA v3.0.0**
  - Deployment @ Teide: Feb 27, 2025 — 3 telescopes

# Software development life cycle (SDLC): integration of subsystems

- **At the system level (e.g. SCADA).**
- **Iterations on integration:**
  - **Software integration, verification and validation:**
    - Integration of the subsystems in a **testing environment**;
    - Acceptance Test Review, to verify the completeness of the developed software, documentation, test and analysis reports + preliminary **Validation Test Report**;
  - **Software deployment and system integration (with hardware and other sw systems) in the operational environment:**
    - deployment at the Array Observing Site, or in the off-site Data Center
    - operational Readiness Review (ORR): the software system is ready for operations + final **Validation Test Report**.




- Deployment
  - Kubernetes
  - ACS Virtual Machines

# List and status of subsystems to integrate

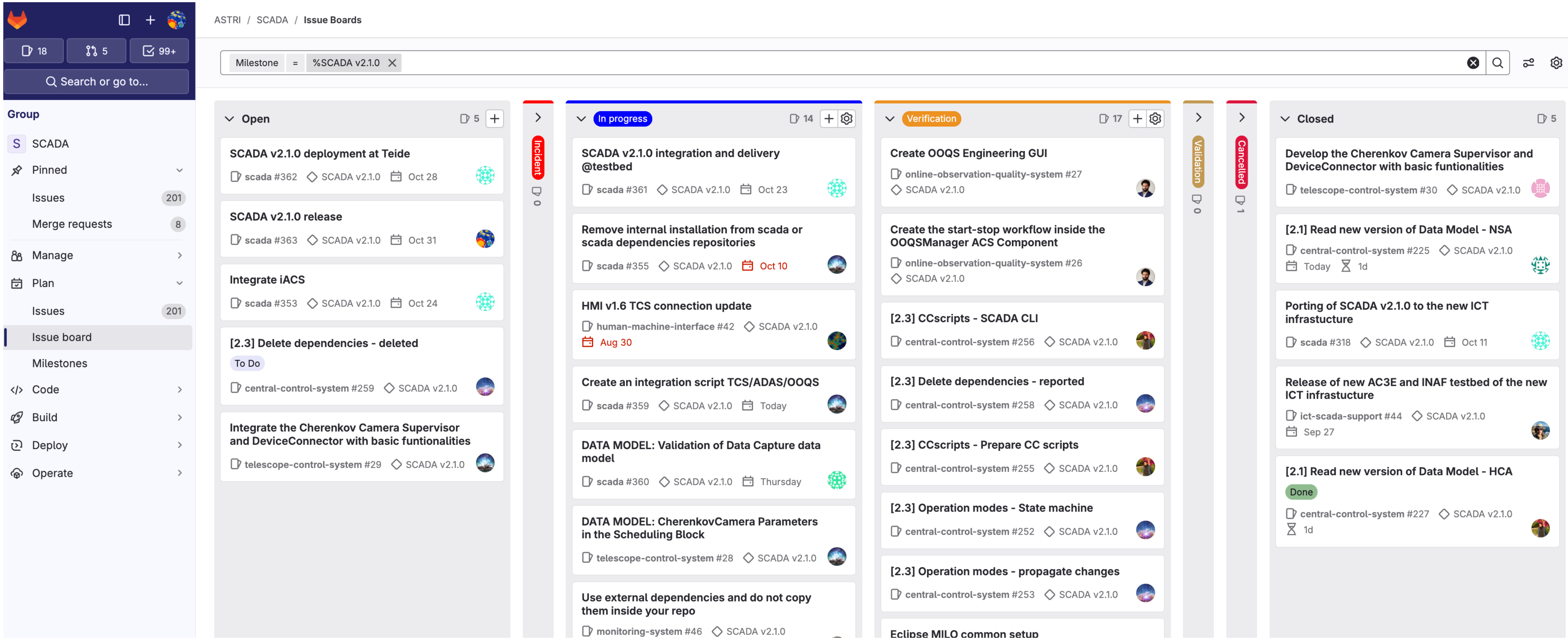
**Delivery status**

| Sub-systems | Tag | Date Update | Status | UCD | SRS | SDD | VERTP | VERTR | SUM |
|---|---|---|---|---|---|---|---|---|---|
| interface-control-idl | v2.12.0 | 2024-09-03 | | na | na | na | na | na | |
| interface-control-document | v4.3.0 | 2024-10-01 | ✅ CHANGED | na | na | na | na | na | |
| astri-manager-components | v1.0.0 | 2024-09-03 | | | | | | | |
| common-device | v1.0.0 | 2024-01-09 | | na | na | na | na | na | |
| ext-lib | v1.2.1 | 2023-06-01 | | na | na | na | na | na | |
| icdtoacs-code-generator | v1.2.0 | 2023-05-29 | | na | na | na | na | na | |
| opcua-simulator | v2.2.0 | 2023-07-03 | | na | na | na | na | na | |
| monitoring-system | v4.8.5 | 2024-09-17 | ✅ CHANGED | 2023-09-13/1.2 | 2022-04-07/1.4 | 2023-09-13/1.2 | 2023-09-13/1.3 | 2023-09-13/1.3 | |
| telescope-control-system | v6.1.2 | 2024-09-12 | | 2023-04-17/2.0 | 2022-04-07/1.3 | 2022-04-07/1.1 | 2023-11-15/4.0 | 2024-02-12/5.1 | |
| human-machine-interface | v1.4.0 | 2024-02-27 | | 2021-06-18/1.1 and discussion | | | | | |
| array-data-acquisition-system | v2.1.1 | 2024-08-19 | | 2024-02-13/1.2 | ⌛ | :hourglass:draft | ⌛ | ⌛ | |
| online-observation-quality-system | v2.2.0 | 2024-09-02 | | 2021-05-01/1.1 | 2022-05-07/1.4 | 2022-05-07/1.1 | 2021-12-22/1.0 | 2021-12-22/1.0 | |
| central-control | v2.2.2 | 2024-09-09 | | | 2023-09-29/2.2 | 2023-09-29/2.2 | ⌛ VERTP draft 1.0 VERTP draft 1.0 | ⌛ draft | 2023-11-07/1.0 |
| ict-system-collector-acs | v1.0.4 | 2024-03-01 | | 2023-09-29/1.0 | na | na | ⌛ draft | ⌛ draft | |
| power-management-system-collector-acs | v1.2.1 | 2024-09-05 | | ⌛draft | | | ⌛ draft | | |
| environmental-monitoring-system-collector | v2.0.2 | 2024-09-11 | | 2023-09-15/1.0 | ⌛ draft | na | 2023-09-14/3.0 | ⌛ draft | |
| onsite-offsite-transfer-system | v0.1.1 | 2023-10-19 | | | | | | | |
| ict-scada-support | v1.5.0 | 2024-03-18 | | | | | | | |
| SCDB manager | v1.1.0 | 2024-01-29 | | | | | ⌛ 2023-11-10/1.0 | ⌛ 2023-11-14/1.0 | ⌛ draft |
| atmosphere-characterisation-control-system | v2.0.1 | 2024-09-05 | | | | | | | |
| central-control-system | v1.5.0 | 2024-09-03 | | | | | | | |
| alarm-system | v1.1.0 | 2024-07-18 | | | | | | | |

| Sub-systems | Tag | Date Update | Status | UCD | SRS | SDD | VERTP | VERTR | SUM |
|---|---|---|---|---|---|---|---|---|---|
| power-management-system-collector | v1.4.0 | 2024-06-20 | | na | na | na | na | na | |
| service-cabinet-control-system | v1.4.0 | 2024-06-20 | | 2022-11-29/1.0 | | | 2023-03-15/1.1 | 2023-03-15/1.1 | |

| Sub-systems | Tag | Date Update | Status | UCD | SRS | SDD | VERTP | VERTR | SUM |
|---|---|---|---|---|---|---|---|---|---|
| aiv-icdutils | v1.1.0 | 2024-06-20 | | na | na | na | na | na | |
| avi-opcuaconnector | v1.0.0 | 2024-06-20 | | na | na | na | na | na | |
| avro-kafka-utils | v1.5.0 | 2023-09-07 | | na | na | na | na | na | |

# Kanban board (INAF Gitlab)

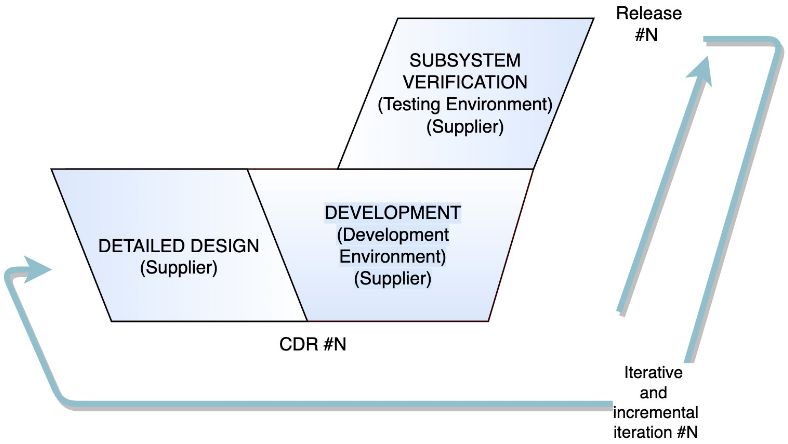





A. Bulgarelli

# Software development life cycle (SDLC): subsystem development

- At the subsystem level (e.g. SCADA Monitoring system).
- **Iterations on subsystem development:**
  - **selection of a use case to develop + Verification Test Plan**, then
    - **detailed design:** Critical Design Review for some iterations;
    - **development;**
    - **verification:** if all automated tests pass, the software is released with the updated documentation + **Verification Test Report**.




use case-driven development approach

- INAF GitLab
- **Enterprise Architect**
- Continuous Integration (CI): GitLab CI environment for automated subsystem verification, dockers.
- SonarQube for quality reports.
- Hardware simulators.

# Use case development approach

**8.2. ASTRI-UC-9.1.1.0-100: The Central Control executes a scheduling block for Cherenkov observation**

**Summary and Scope**: This UC describes the execution of a SB. This UC describes the use case for creating, operating, and destroying an Array. This task is one of the major activities of operating the **Central Control system**. This is the method by which calibrations and scientific observing projects are executed.
The following steps depict events in the lifecycle of a SB. These steps also show how other subsystems are involved in this process. However, the major purpose of including these actions in this document is to document actions that the Central **Control system** performs in this process.

**Authors**: Gino Tosti, Andrea Bulgarelli, Federico Russo,

**Version**: 1.0

**Trigger:** the execution of a SB.

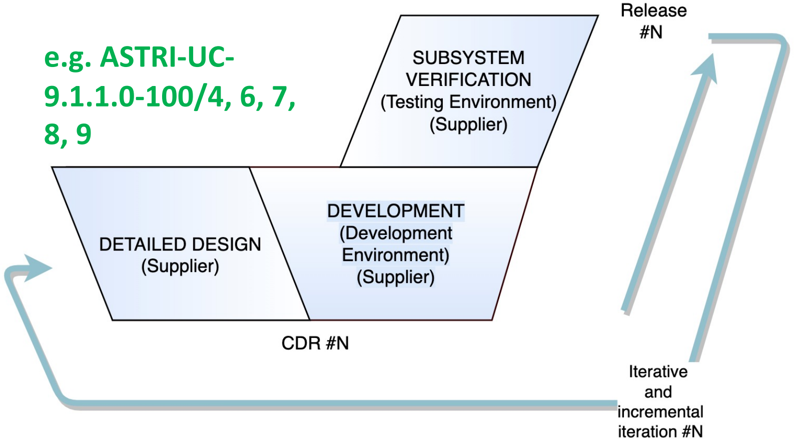
**Frequency**: one or more for each observation night.

**Phase**:

**Assum**
*the old a*

PRE-

e.g. ASTRI-UC-9.1.1.0-100/4, 6, 7, 8, 9




23. **Start execution procedure**. At the beginning of the execution of each science OB, for each OB and for each telescope

*Alternate 1*:
   a. the **observing script** passes to the TCS the target information (RA, DEC, TIME, ....) and commands the TCSs to execute the tracking.
   b. The TCS validates the command by checking the formal correctness of the target information and whether the subsystems can accept that commands in their current status.

*Alternate 2*:
   a. The trajectory is calculated by the **observing script**
   b. The **observing script** sets the trajectory into each TCSs and commands the TCSs to execute the trajectory.
   c. The TCS validates the command by checking whether the subsystems can accept that commands in their current status.

*Exception: Telescope fails to start the tracking TBD.*
*Exception: The TCS can't execute the tracking command as it fails the validation checks.*

24. The **TCS** command the telescope and starts slewing/tracking.

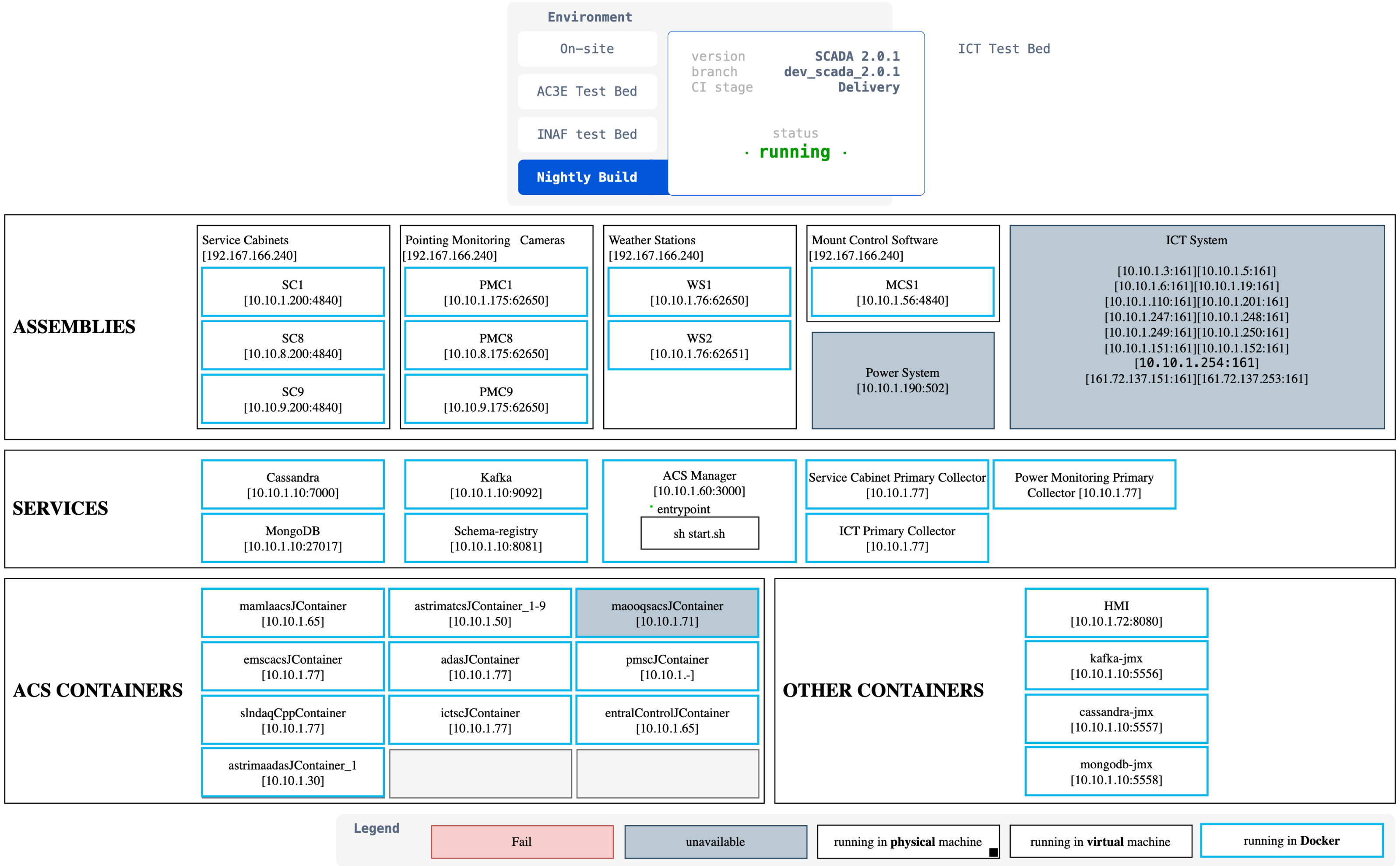
25. The **TCS** checks the **ADAS** is connected with the **Camera BEE** and is in *idle* state

*Exception: The ADAS and Camera BEE are not connected*
*Exception: The ADAS is not in idle state*

**SCADA v1.0.0 -> ASTRI-UC-9.1.1.0-100/3, ASTRI-UC-9.1.1.0-100/4**
**SCADA v1.2.0 -> ASTRI-UC-9.1.1.0-100/4, 6, 7, 8, 9**
**SCADA v1.3.0 -> ASTRI-UC-9.1.1.0-100/11, 16, 17, 18, 19-21, 22, 33d, 34i, 35a, 35b + data acquisition flow 23-29**
**SCADA v2.0.0 -> ASTRI-UC-9.1.1.0-100/18, 33d**
**SCADA v2.1.0 -> ASTRI-UC-9.1.1.0-100/30-32**
**SCADA v2.2.0 -> ASTRI-UC-9.1.1.0-100/33**

# CI, Software Integration and Deployment

ASTRI Mini-Array



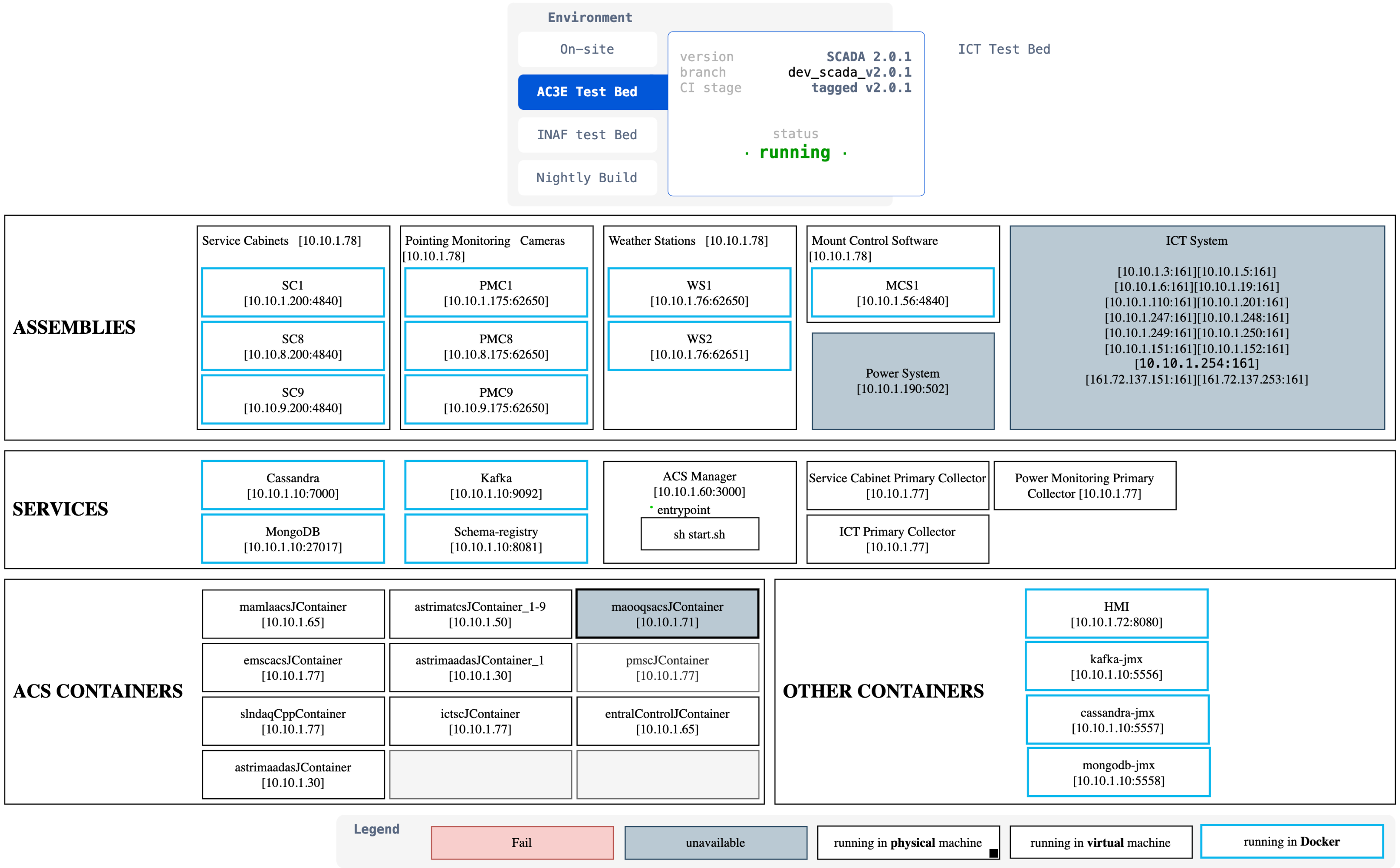

CI pipelines running in a Kubernets cluster



A. Bulgarelli

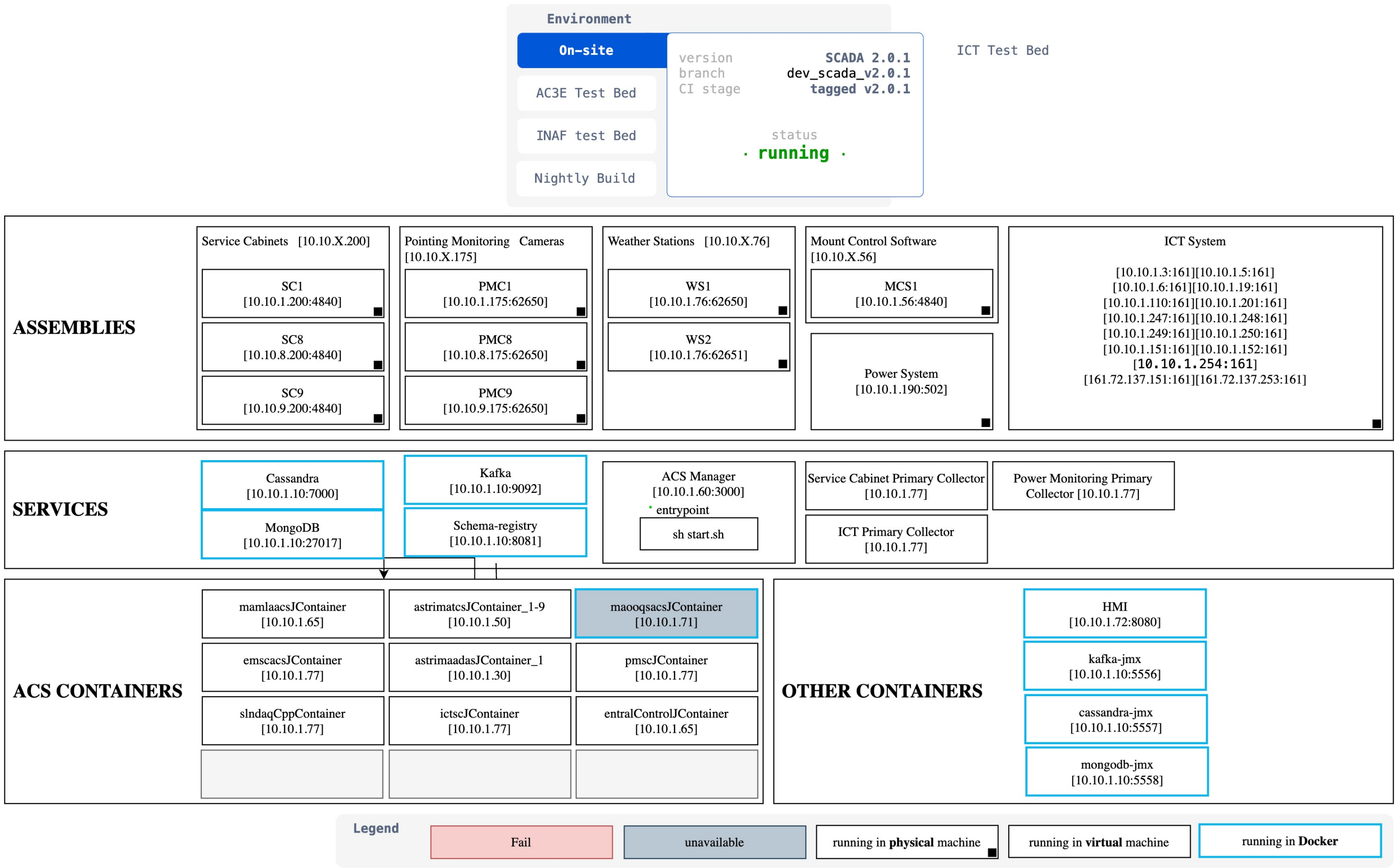
# CI, Software Integration and Deployment

Mini-Array




Two test bed fully representative of the deployment environment + hardware simulators

# CI, Software Integration and Deployment

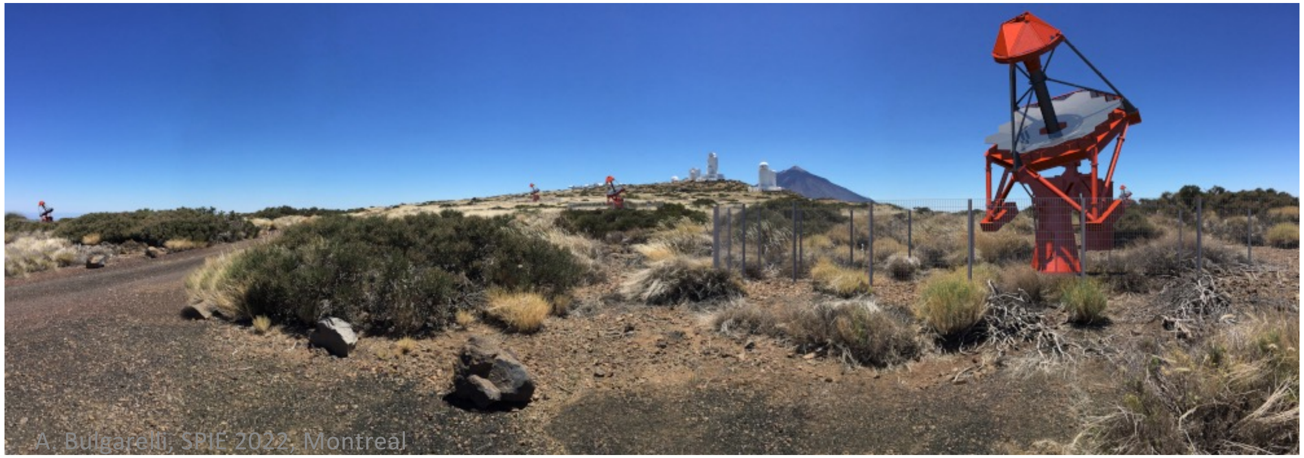



On-site ICT

# Conclusions

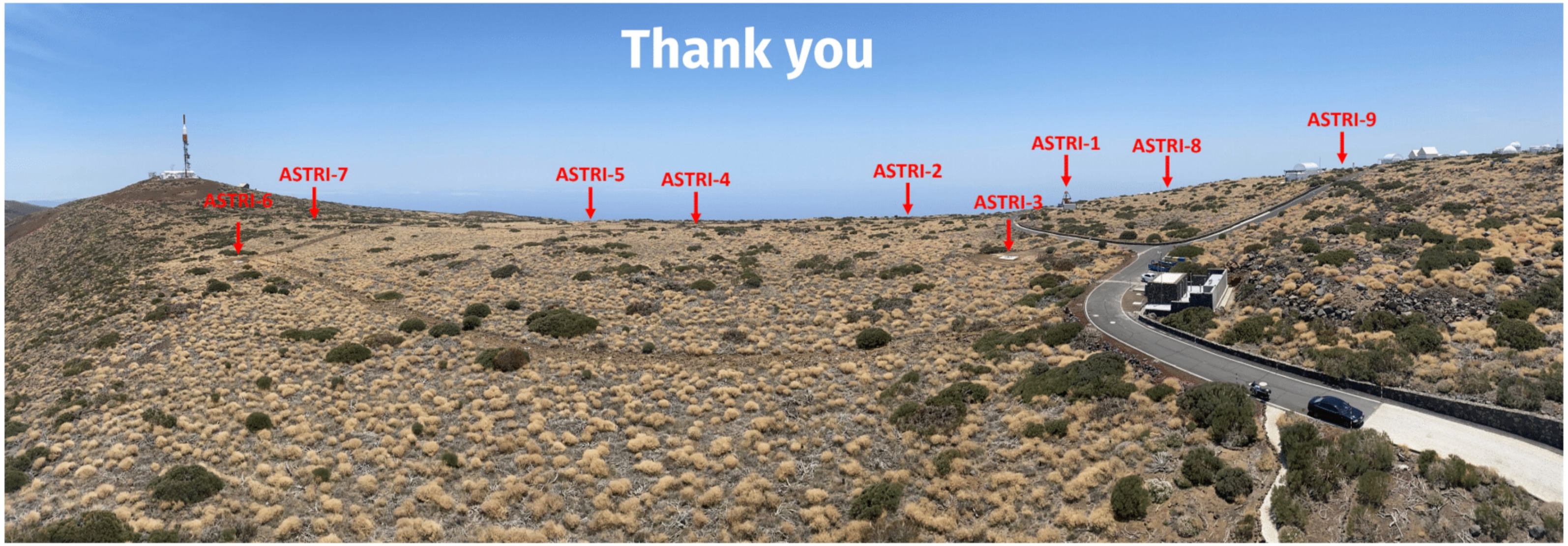
- The ASTRI Mini-Array software development approach allows the ASTRI team to release, integrate and deploy software in an iterative and incremental way:
  - to follow the different phases of the project, from construction to operations;
- SCADA will be used in the following months to acquire data from the first ASTRI-1 telescope @Teide


- Software architecture and development approach for the ASTRI Mini-Array project at the Teide Observatory, A. Bulgarelli at al, Journal of Astronomical Telescopes, Instruments, and Systems, Vol. 10, Issue 1, 017001 (January 2024). https://doi.org/10.1117/1.JATIS.10.1.017001

# ASTRI Mini-Array

View from Vacuum Tower Telescope

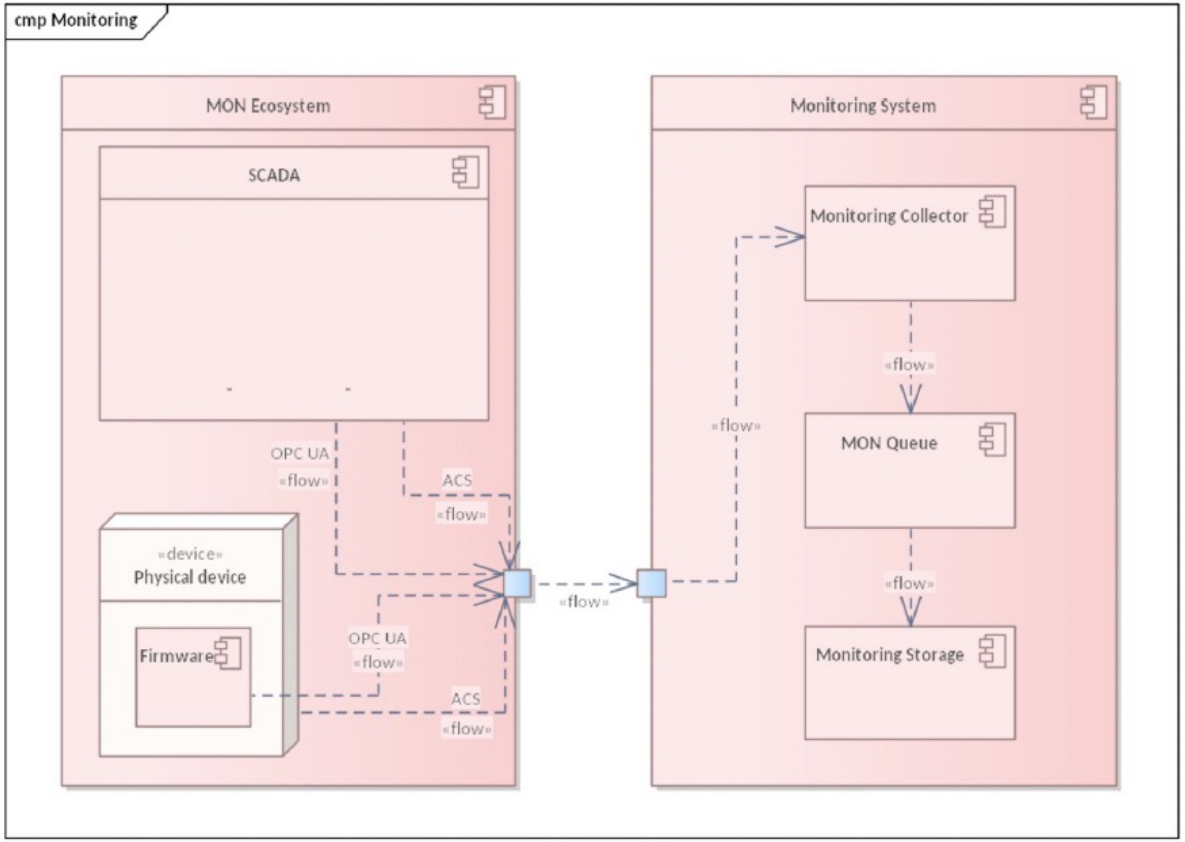



View from Themis Telescope

# Backup slides

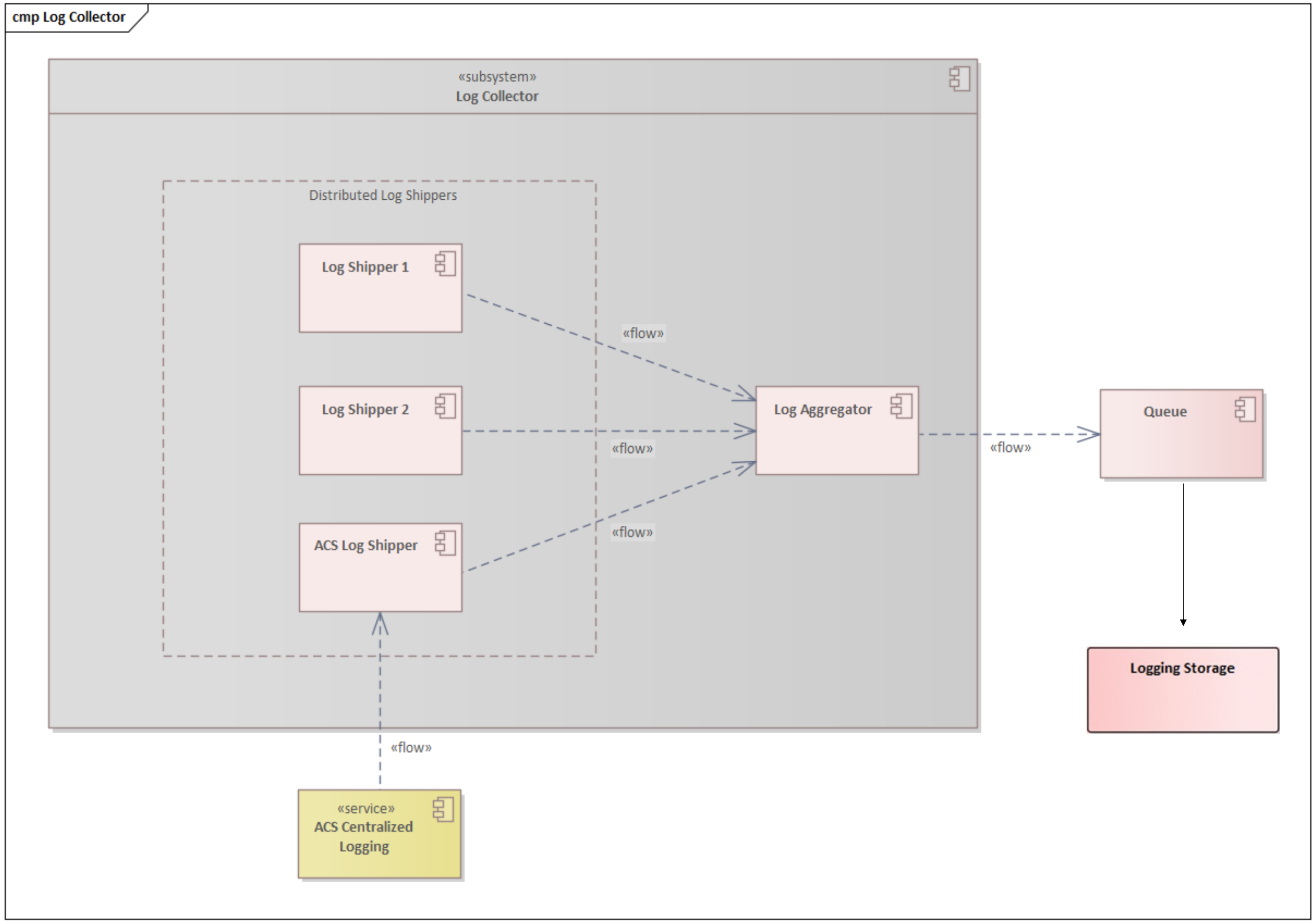
# MONITORING SYSTEM

ASTRI Mini-Array




•**Comprehensive System Monitoring**: Tracks key data points from ASTRI MA's hardware and software components for real-time system health insights.
•**Fault Detection and Diagnosis**: Enables systematic identification and analysis of system faults to enhance operational reliability.
•**Support for Predictive Maintenance**: Facilitates corrective actions and predictive maintenance strategies, reducing system downtime.

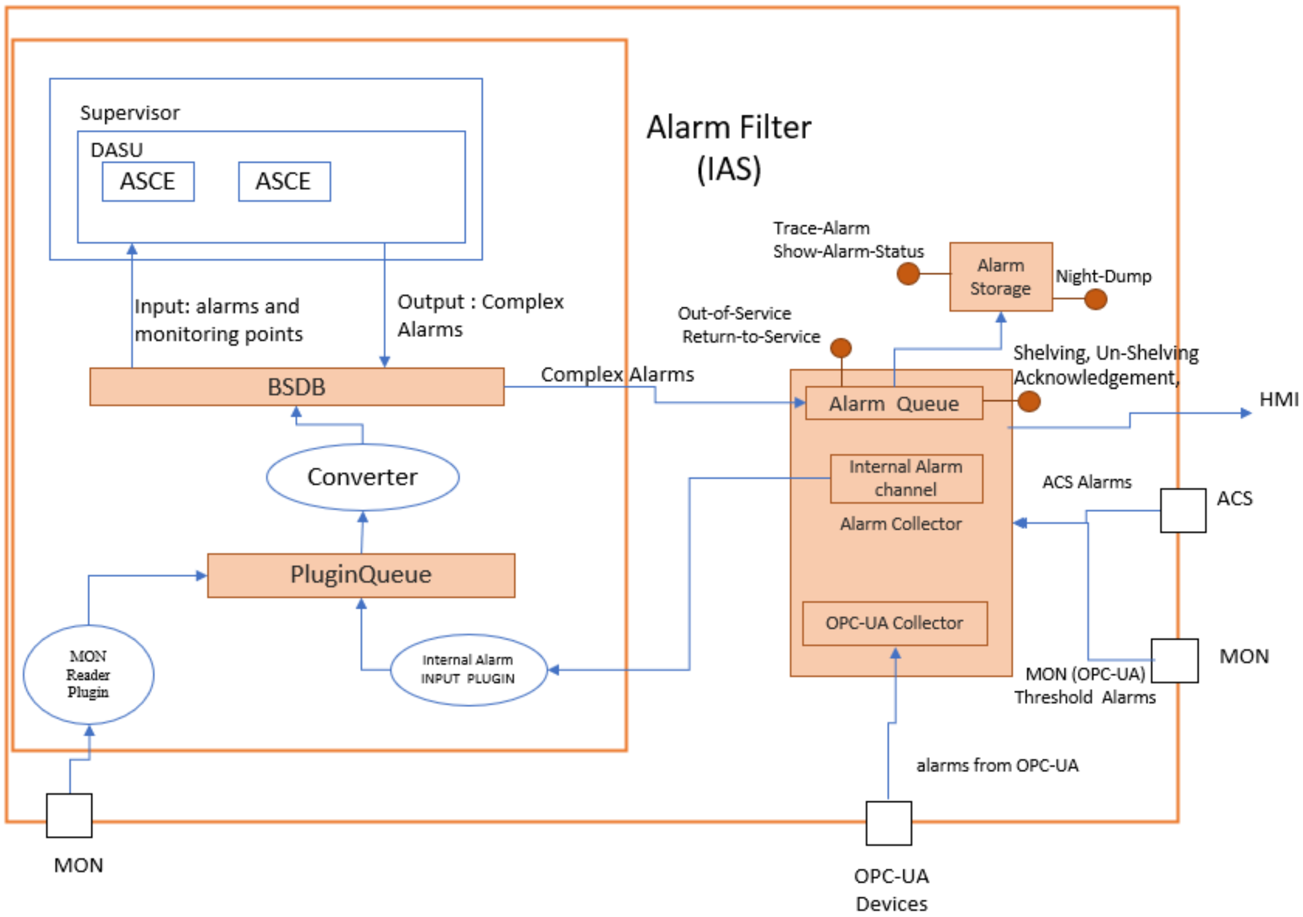
# LOGGING SYSTEM




Comprehensive Log Collection: Gathers logs from subsystems, observation scripts, and low-level software components for complete system visibility.

Integrated ACS Component Logs: Captures logs from ACS components to support system diagnostics and performance monitoring.

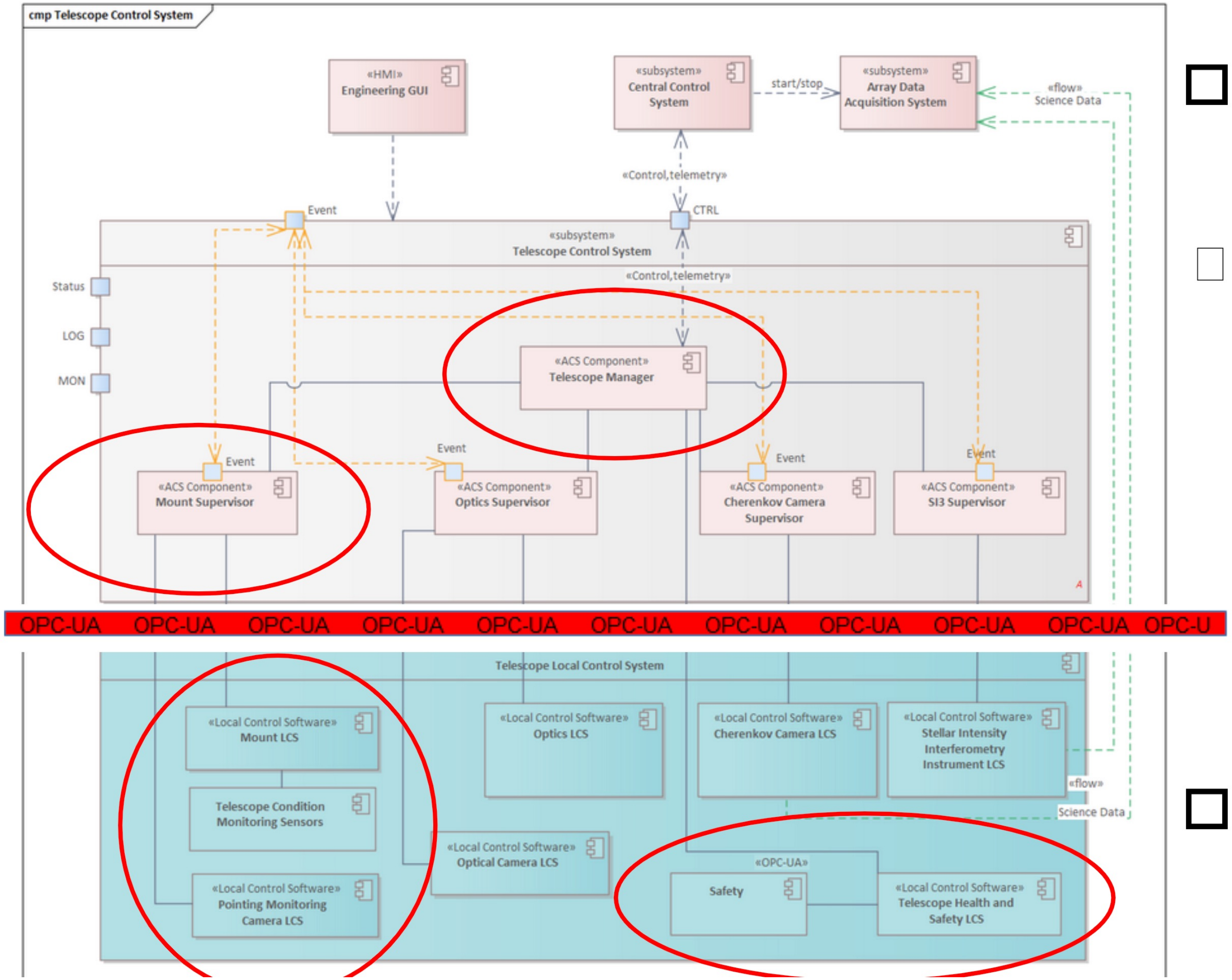
# ALARM SYSTEM




Alarm Detection: Identifies warning and critical alarms based on real-time information to prevent system failures.

Automated Status Evaluation: Ensures proactive system assessment and quick response to operational issues.

# Telescope Control System




- High Level Actors (interface-control-idl)

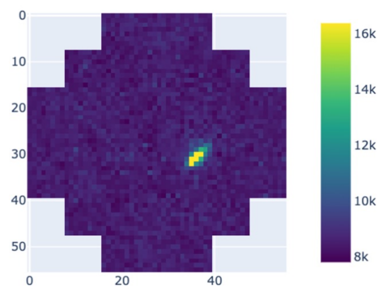
- The current main components include:
  - The Telescope Manager
  - The TcsMaster
  - The MountSupervisor and its Device Connector
  - HealthAndSafetyDeviceConnector
  - The PMCsupervisor and its D.C.

- Local Control Softwares

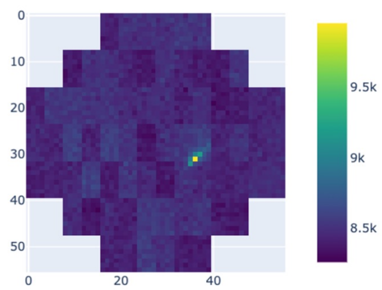
# Online Observation Quality System

## OOQS Engineering GUI

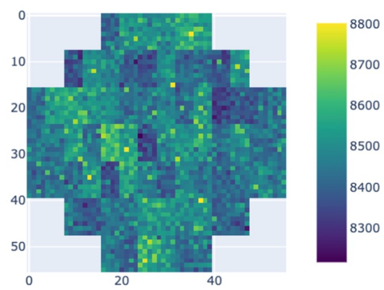
### S22 High Gain



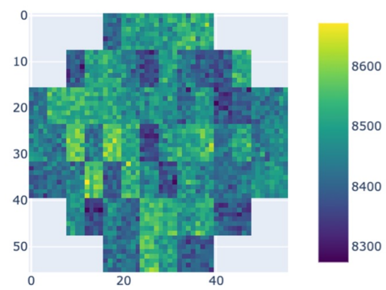

### S22 Low Gain



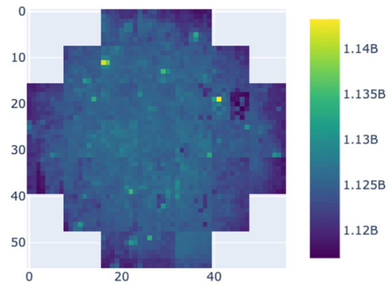

### C11 High Gain



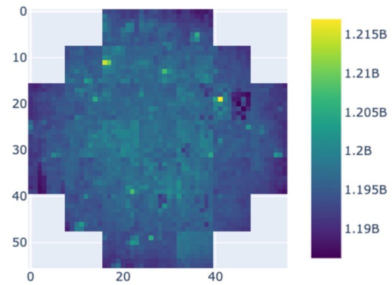

### C14 Low Gain




### VAR 102

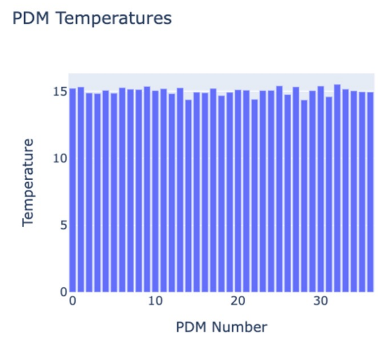



### VAR 103




### PDM Rates


### PDM Temps



# CENTRAL CONTROL SYSTEM

in-progress

- 50% Central Control — ACS
- 80% Environmental Monitoring System Collector — ACS
- 50% Atmosphere Characterisation Control System — ACS
- 40% Power Management System Collector — ACS
- 10% ICT System Collector — ACS

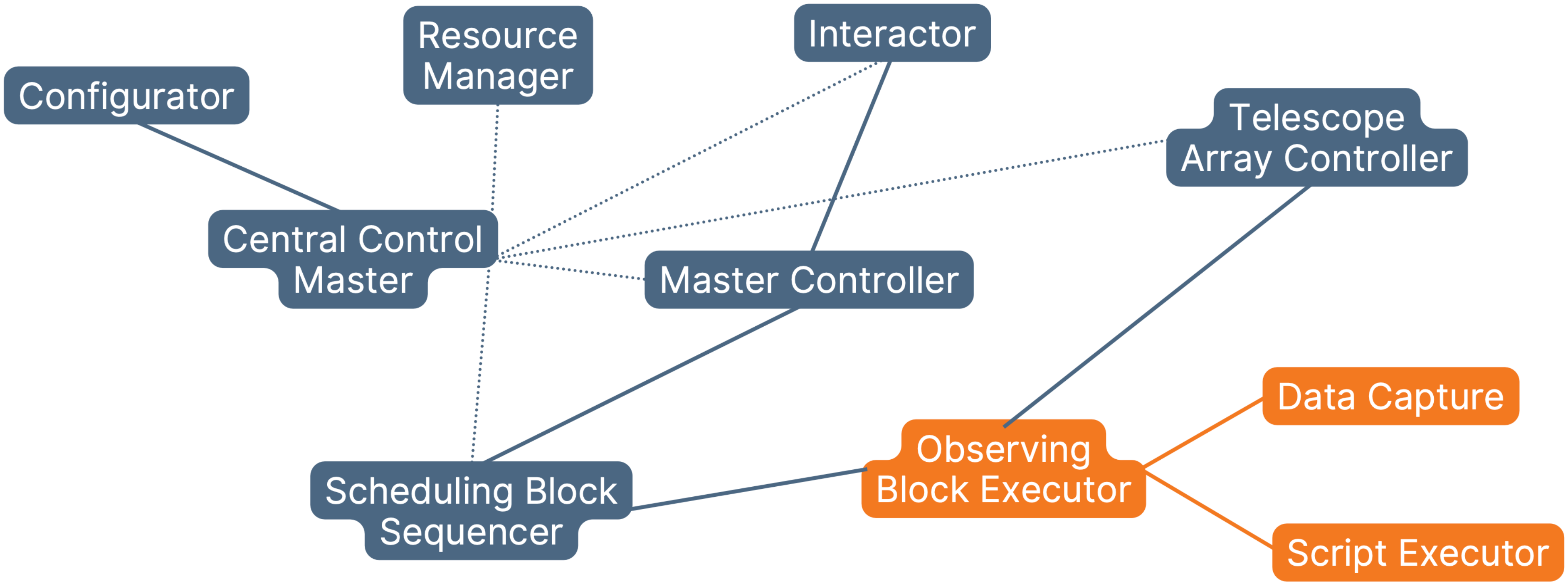
upcoming

- Array Calibration Control System — ACS
- Safety and Security System Collector — ACS
- Telescope Service Cabinet Collector — ACS

AC3E
Advanced Center for Electrical and Electronic Engineering

# CENTRAL CONTROL

legend
<ACS Dynamic Components/>
<ACS Components/>

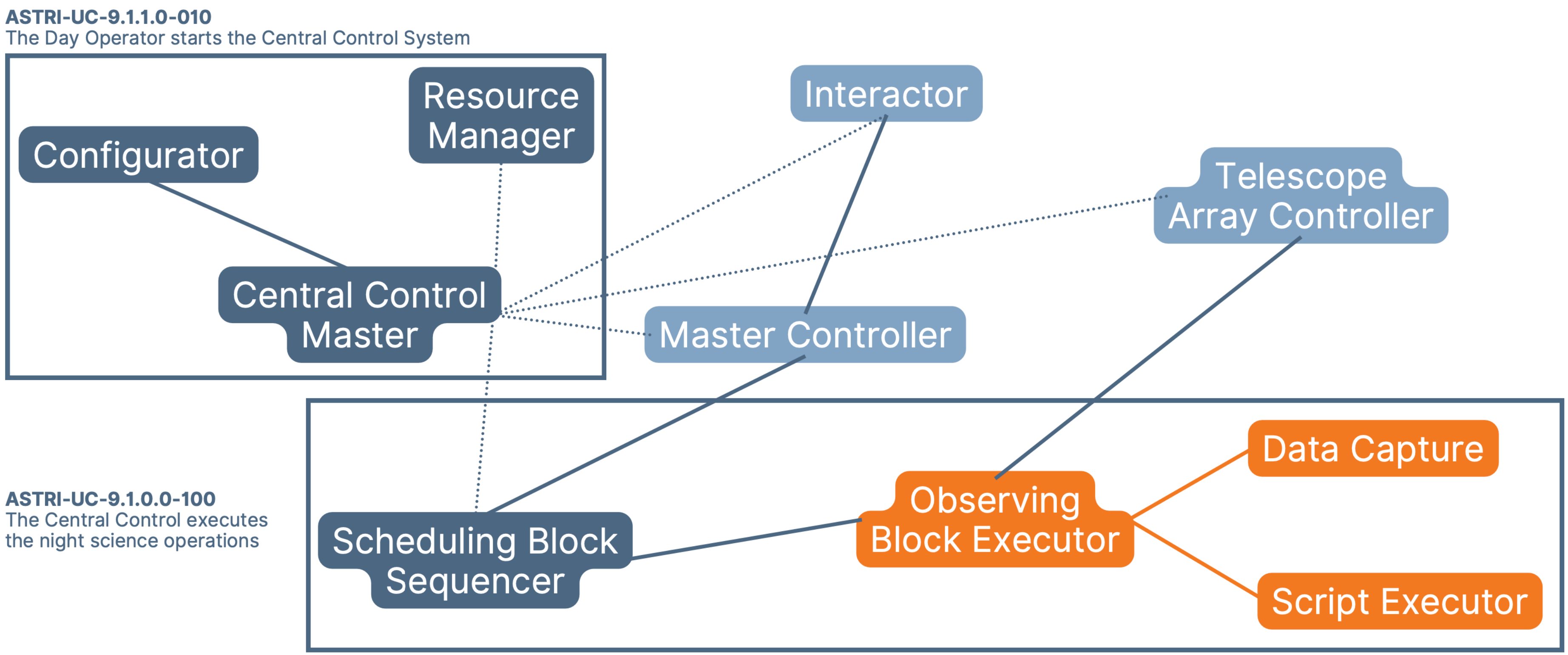


legend

*<ACS Dynamic Components/>*

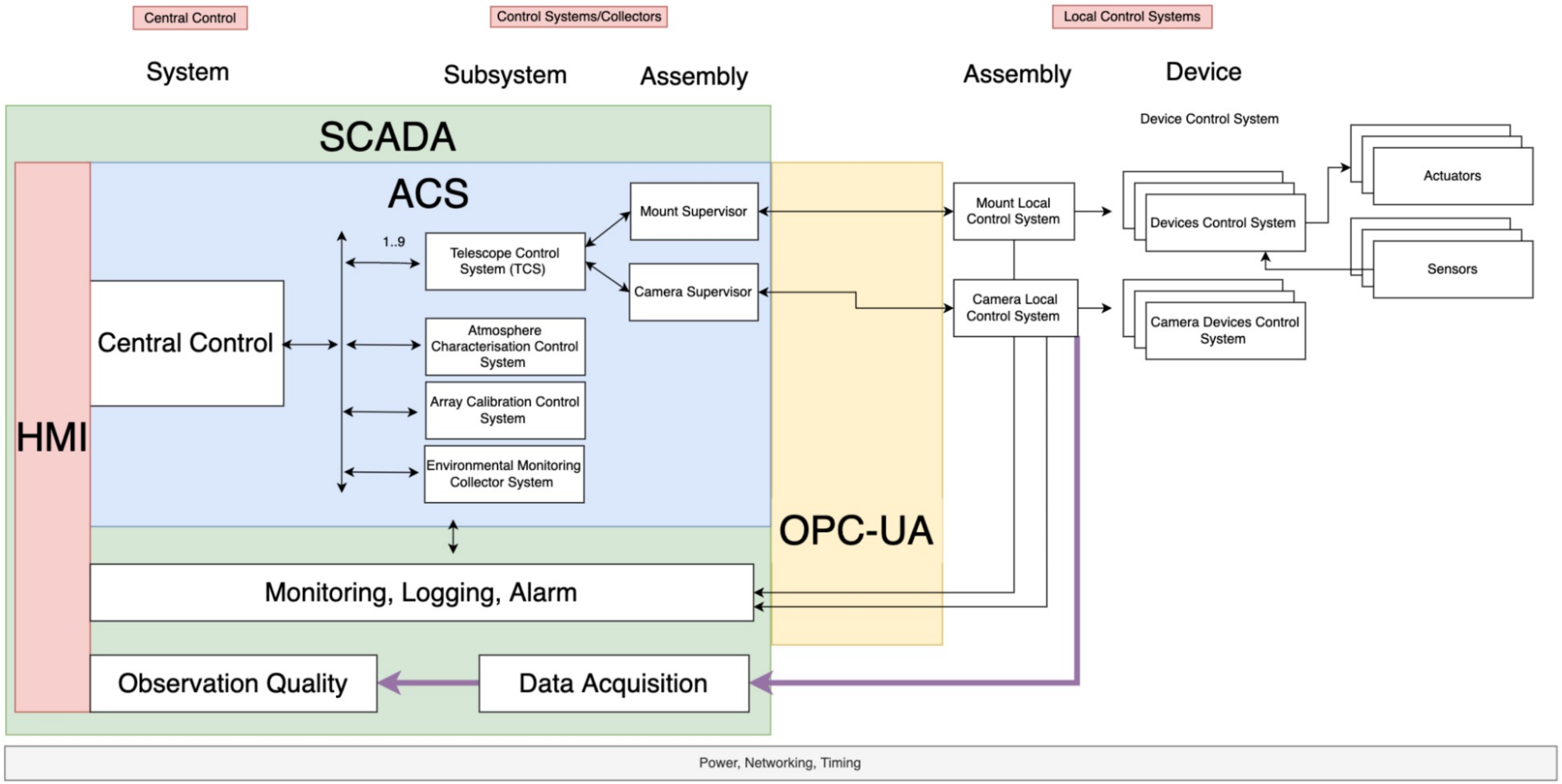
*<ACS Components/>*

# CENTRAL CONTROL V2.0.0



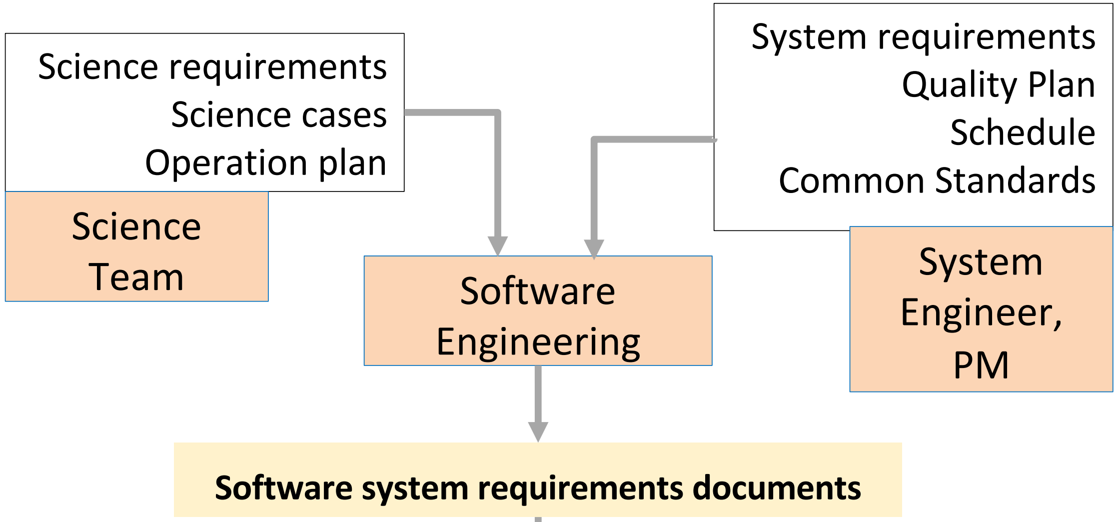
# Supervisory Control and Data Acquisition System (SCADA)/1

- **SCADA controls, monitor, and acquire data for all the assemblies and operations carried out at Array Observing Site in an automated way.**
- Supervised by an Operator.



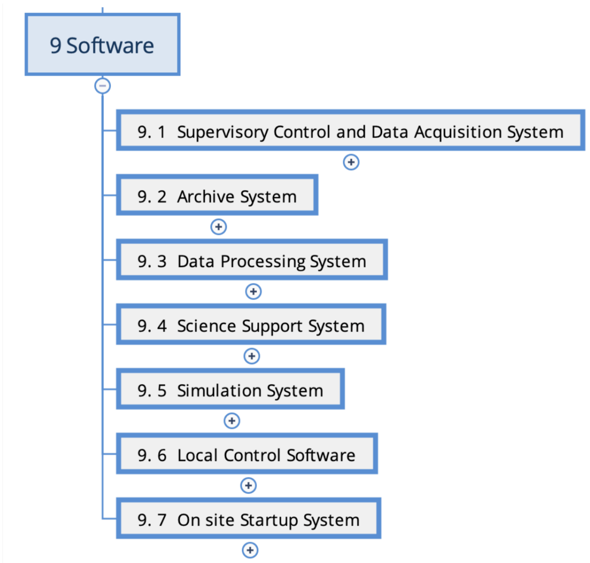
# Software system requirement documents

Mini-Array

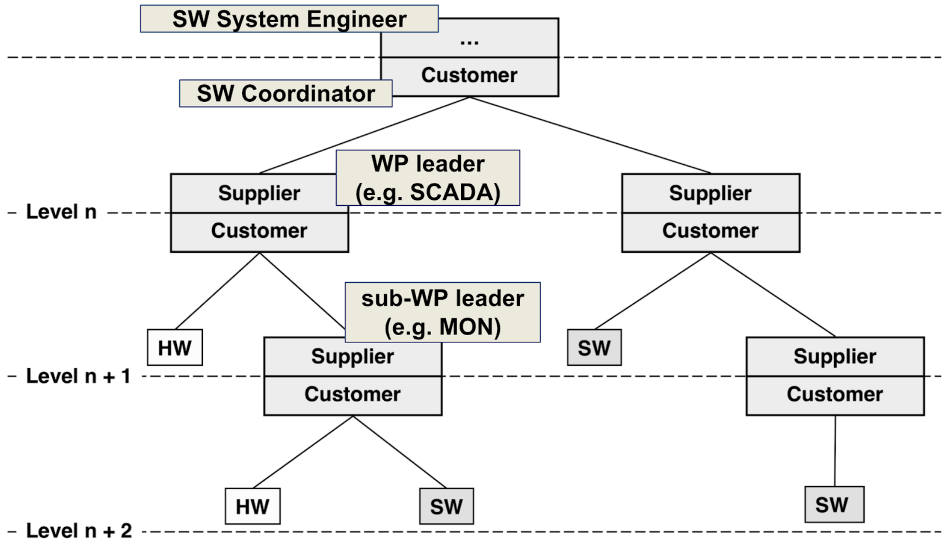



- ASTRI-MA Top Level **Functional Software Architecture**
  - functional architecture and quality requirements
- ASTRI-MA Top Level **Use Cases**
- ASTRI-MA **Data Model**
  - to support functional and behavioural views and support for logical external/internal interfaces
- Derived documents:
  - ASTRI-MA Software **Product Breakdown Structure**
  - ASTRI-MA **Glossary**

# Product Breakdown Structure and customer-supplier relationship




First level of the ASTRI Mini-Array software Product Breakdown Structure



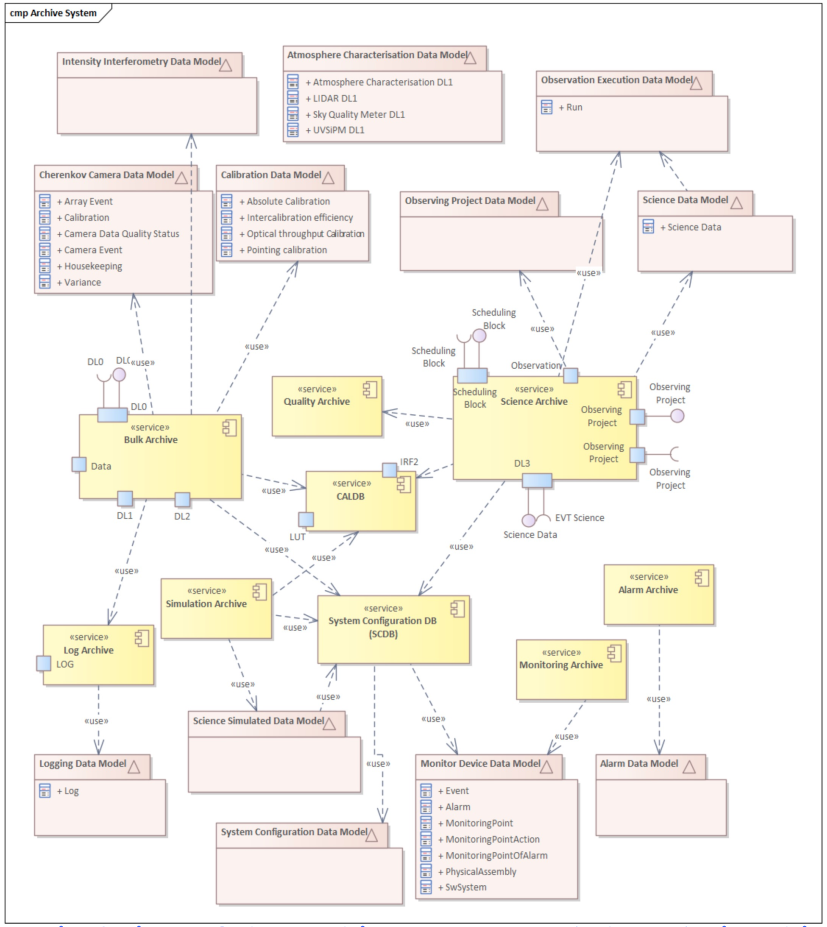

**Software suppliers: software development by INAF and external companies/research institutions:**

- Central Control System and on-site deployment of SCADA by AC3E (SCADA)
- Operator Human Machine Interface by the University of Geneve (SCADA)

- **The customer-supplier relationship of the ASTRI Mini-Array software.**
- INAF has in charge of **software management and coordination** and development of many software subsystems.

# Archive System

- The central repository and the core of the ASTRI workflow.
- Distributed between on-site (temporary) and off-site.
- Management of
  - Observation plans;
  - Scientific data;
  - Monitoring/alarm/logging data;
  - system configuration.
- **The main interface for the SCADA system**

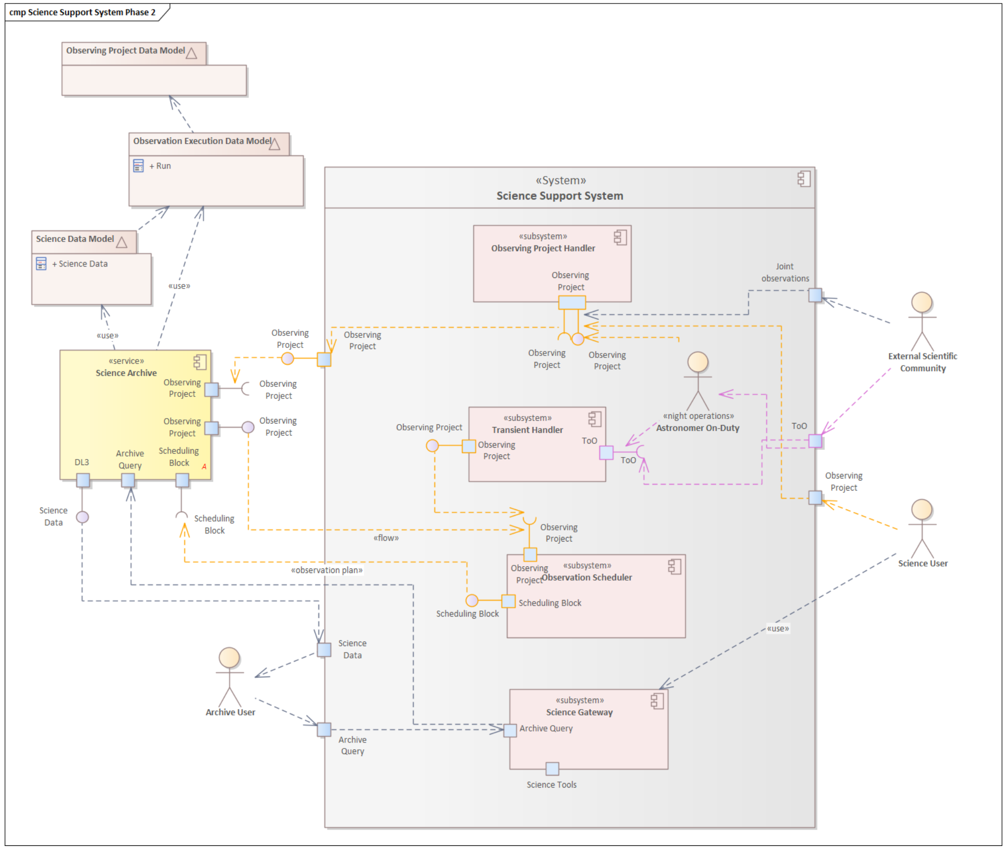



Logical view of the Archive System and the relationship of each archive with the data models

# Science Support System

Mini-Array

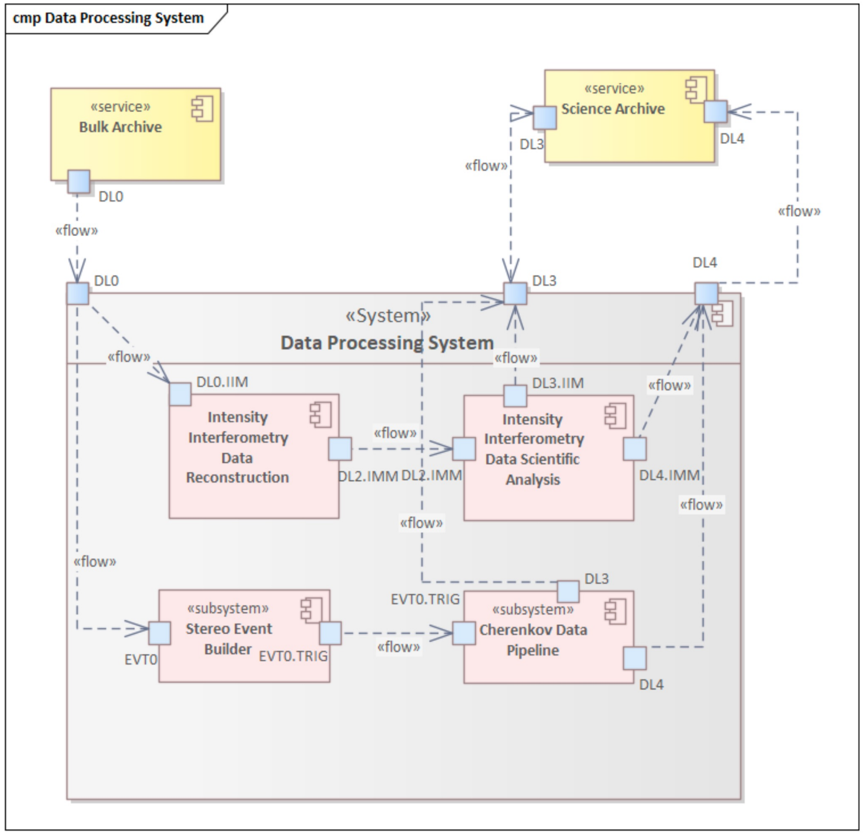
- The Science Support System is the main interface to the Science User.
- It manages observation plans, dissemination of scientific data and science tools.
- **Prepare the observation plan for the SCADA system**
- Main components:
  - **Observing Project Handler**: to submit Observing Projects;
  - **Transient Handler**: to manage ToO observations
  - **Observation Scheduler**: preparation of long-term and short-term observation plans;
  - **Science Gateway**: to retrieve science-ready data, science tools and tools to support the Observing Project preparation.




Science Support System component diagram with the Science Archive and related data models.

# Data Processing System

- The Data Processing System (DPS)
  - performs data reduction and analyses;
  - checks the quality of the final data products.
- Main functions:
  - Stereo Event Builder: perform the off-line software stereo array trigger of Cherenkov data;
  - Cherenkov Data Pipeline;
  - Intensity Interferometry Data Reconstruction and Scientific Analysis;




Data Processing System component diagram. DL0 is the raw data generated by scientific instruments, where IMM is the Intensity Interferometry data and EVT is the Cherenkov data. EVT0.TRIG is the Cherenkov data after the stereo array trigger. DL3 and DL4 are scientific products.