

The logo for the Cherenkov Telescope Array Observatory (CTAO), featuring the letters 'CTAO' in a bold, white, sans-serif font. A small, stylized starburst graphic is positioned between the 'A' and 'O'.

ACADA
COLLABORATION



The Monitoring, Logging and Alarm System of ACADA for the Cherenkov Telescope Array Observatory (CTAO)

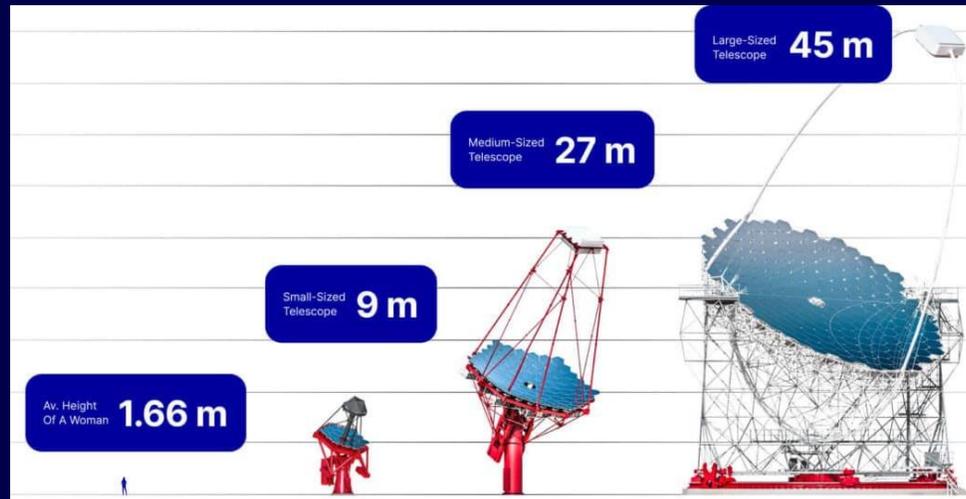
A Scalable Infrastructure for Observatory-Wide Data Collection, Storage, and Diagnostics

Kevin Munari - INAF CATANIA - PRESENTER
Federico Incardona - INAF CATANIA - PRESENTER
Alessandro Costa, Sebastiano Spinello INAF CATANIA

USC-C General Assembly
Savoia Excelsior Palace, Trieste (IT)
March 11, 2026

CTAO

More than **60** telescopes deployed at the **2** hemispheres of Earth



Large-Sized Telescope (LST): ~45 m; ~50 tonnes; ~20 sec for repositioning (20 GeV → 3 TeV)

Medium-Sized Telescope (MST): ~27 m; “workhorse” of CTA with sensitivity in its core energy range (80 GeV → 50 TeV)

Small-Sized Telescope (SST): ~9 m; sensitive to the highest energies (2 → 300 TeV)

CTA N + CTA S = **CTA Observatory (CTAO)**: the first ground-based gamma-ray *proposal-driven* observatory open to the *worldwide* astronomical and particle physics communities

ACADA in CTAO

The core software infrastructure of CTAO

REL 1 → July 2023

REL 1.5 → Dec 2024

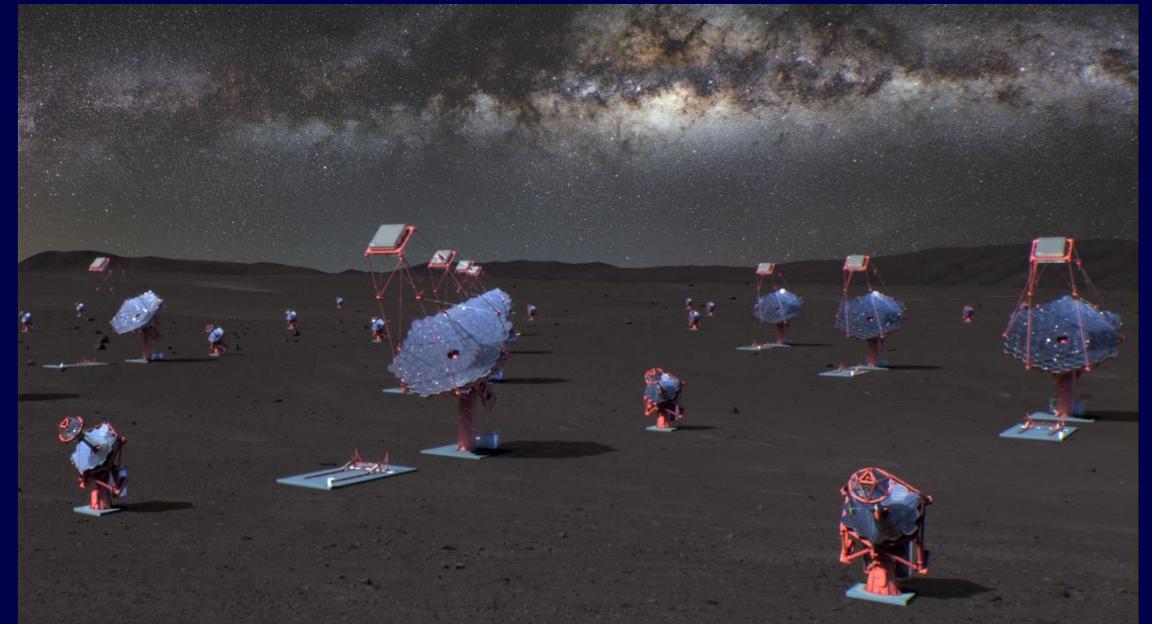
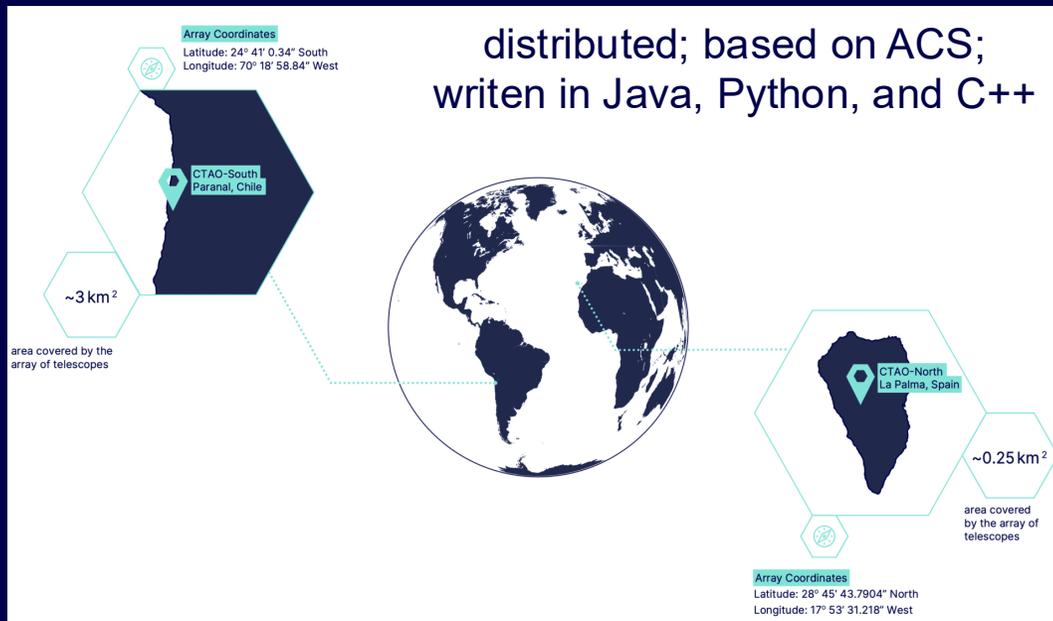
REL 2 → expected July 2026

Supervises distributed array elements at **CTA-N** and **CTA-S**.

Orchestrates telescope **observation** execution.

Manages **data acquisition** and **compression** of raw data.

Provides **automatic science alerts** and **human-machine interface**.



ACADA in CTAO: numbers

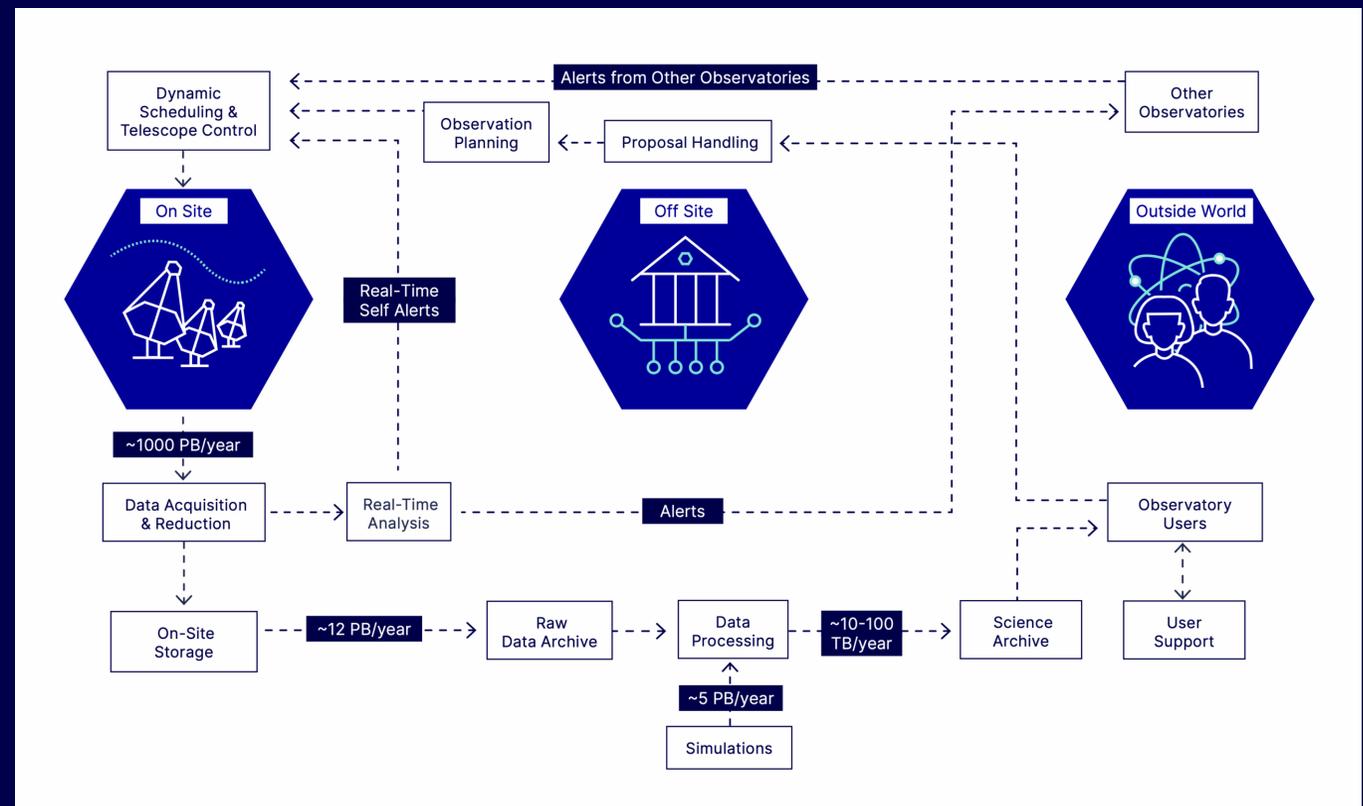
50+ experts to build ACADA

8 years project (2021 -> 2029)

1000 PB/year or raw data

12 PB/year after compression

2000 computer cores (overall)



MONAAS in ACADA

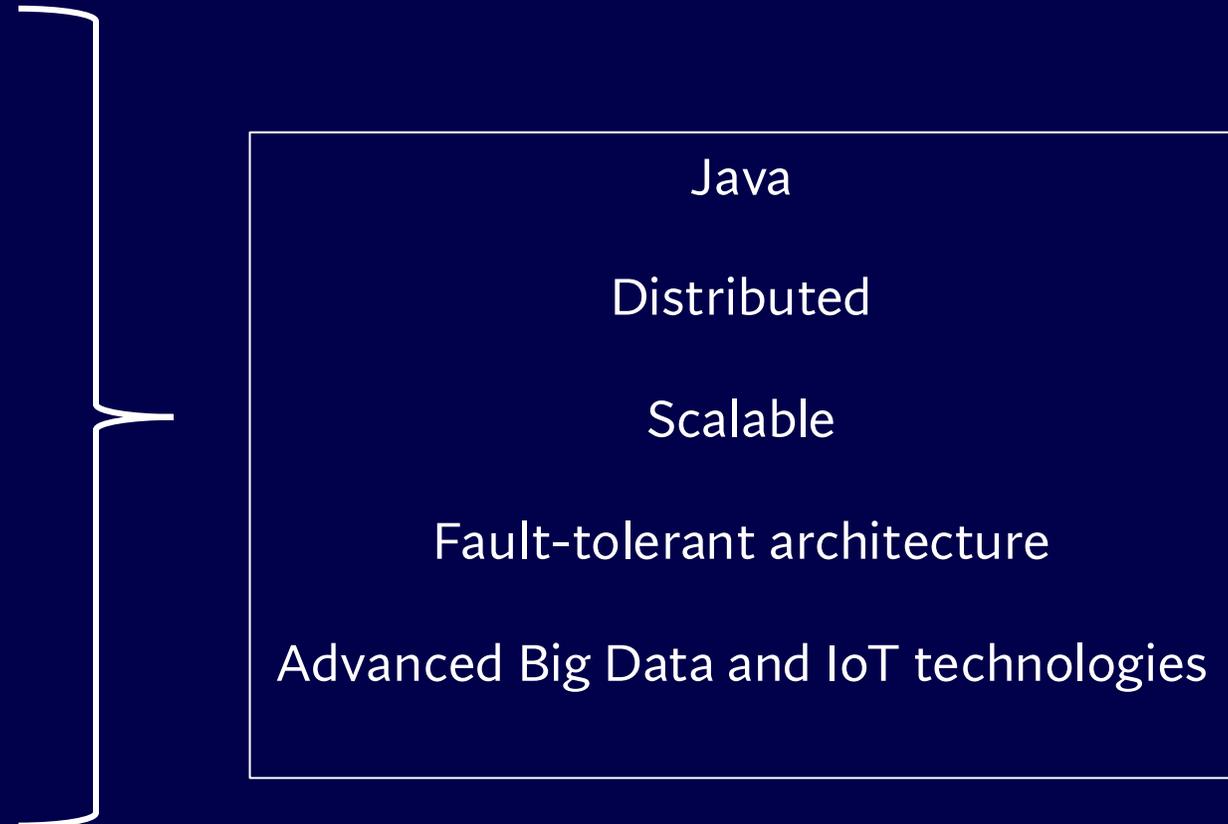
The core systems for CTAO inspection and diagnosis

Monitoring and Logging System (MON)

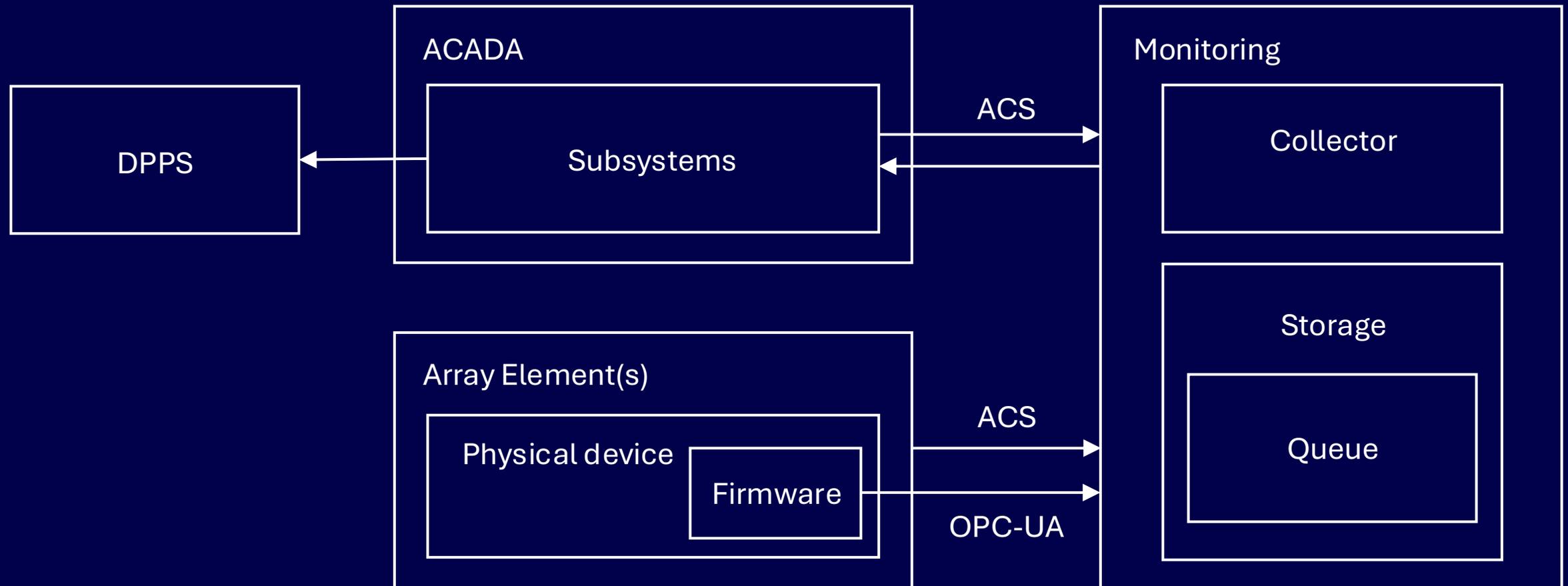
- Collect, elaborate, distribute, and archive ancillary data coming from telescopes, auxiliary, and calibration instruments
- → 5 Hz; 26 Mbps for Monitoring; 1 Gbps for Logging

Array Alarm System (AAS)

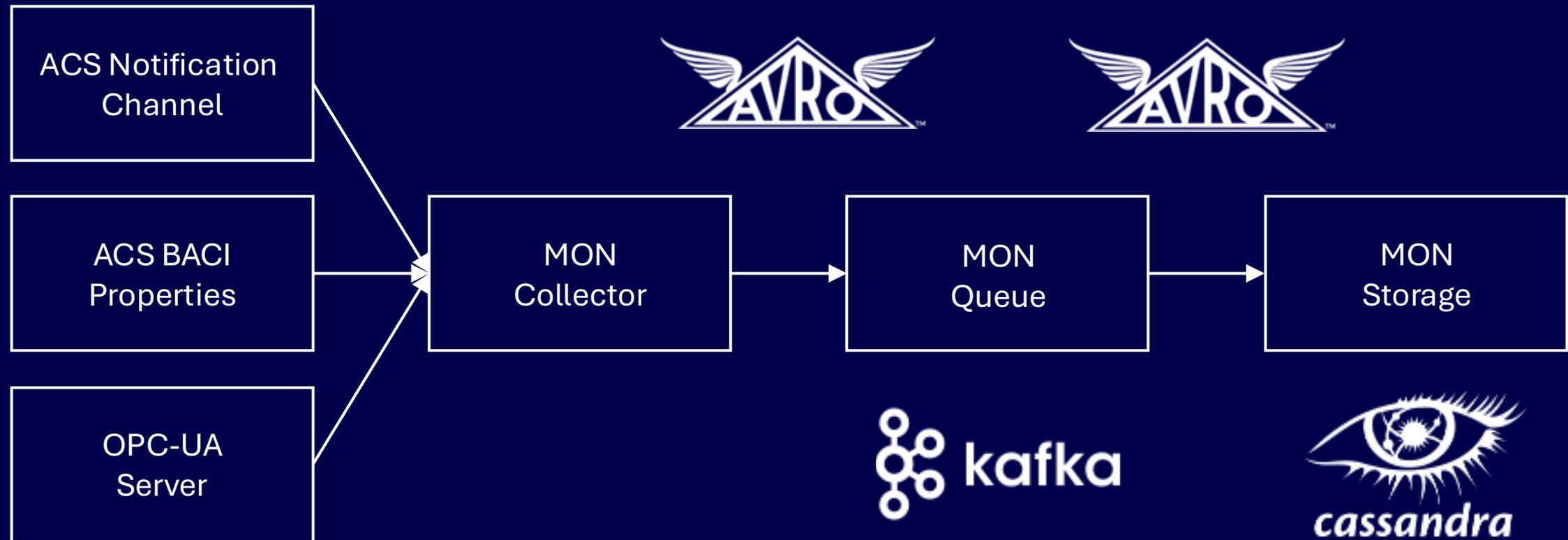
- Collect, filter, shelve, reduce, acknowledge, and combine alarms from Array Elements (telescopes and instruments)



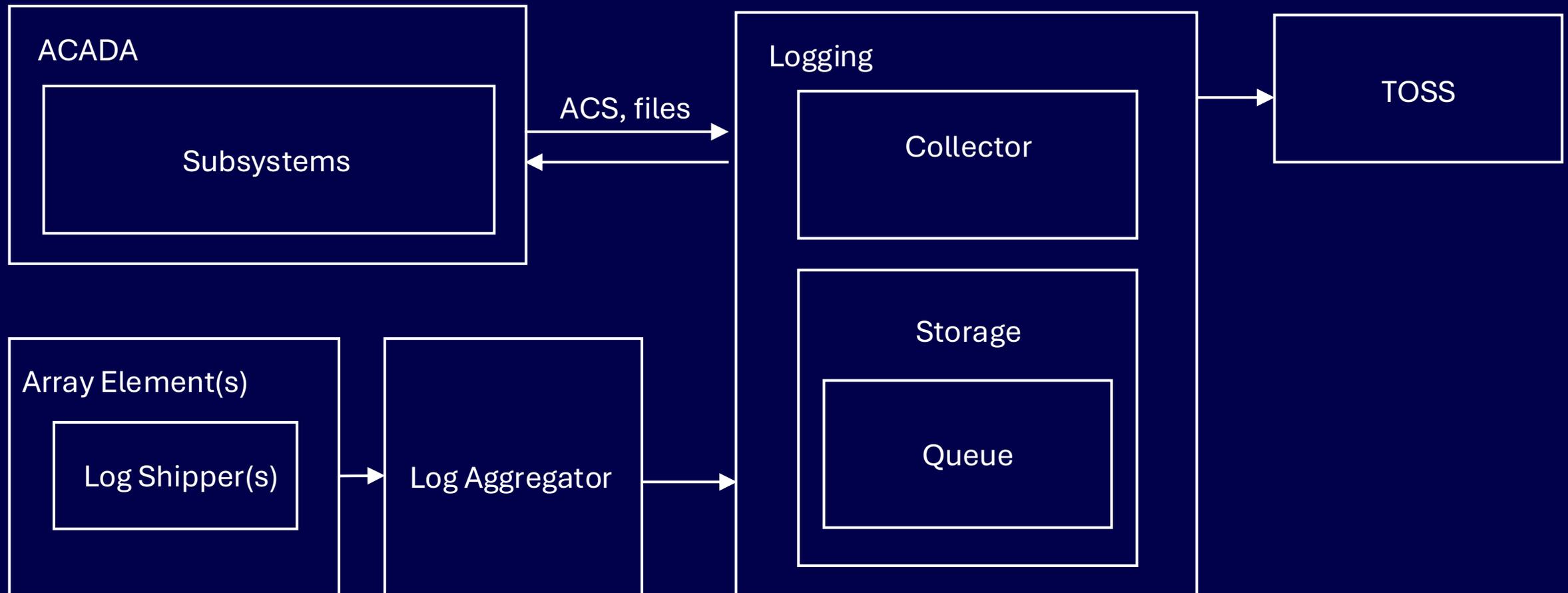
Monitoring deployment



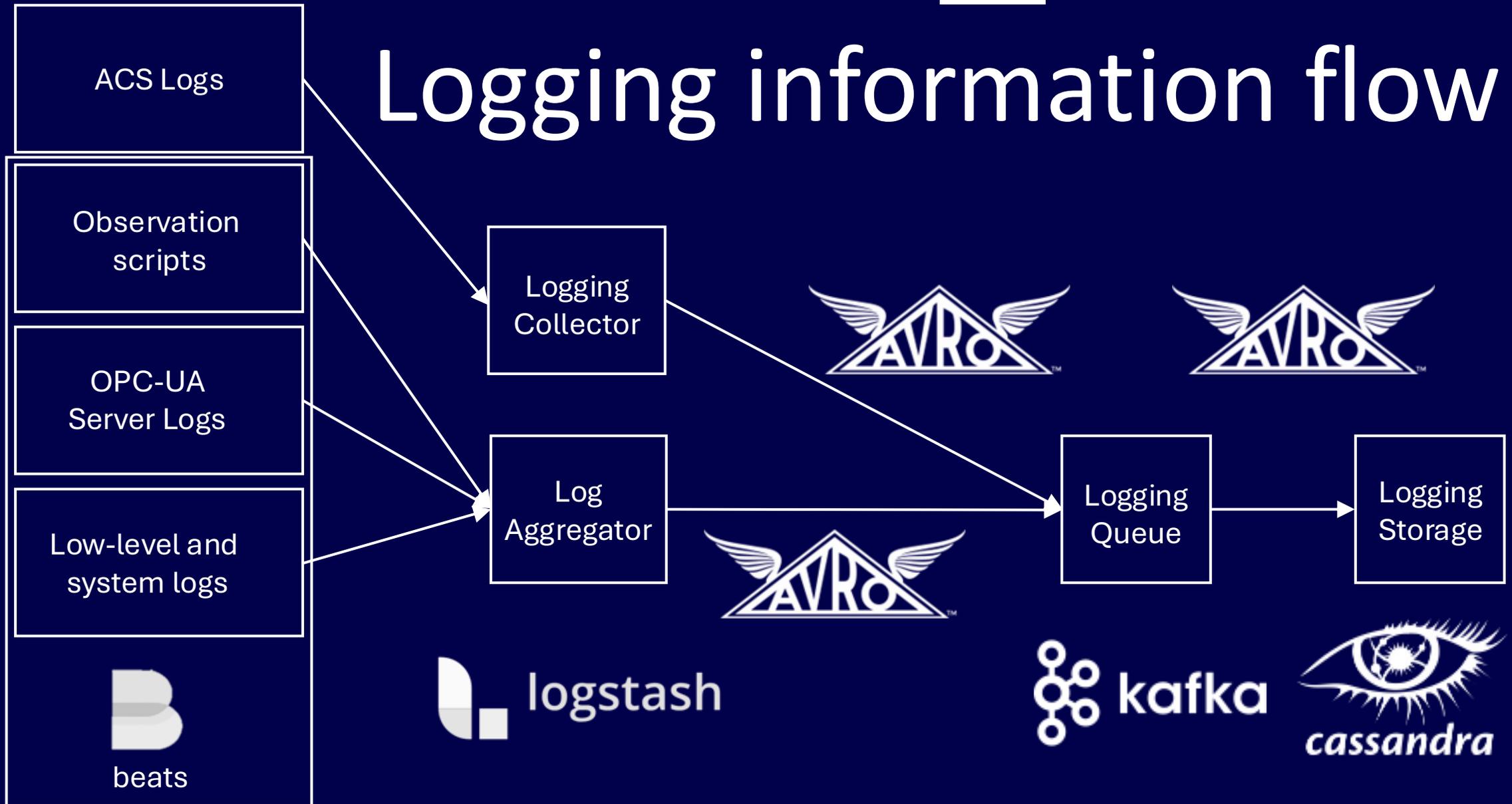
Monitoring information flow



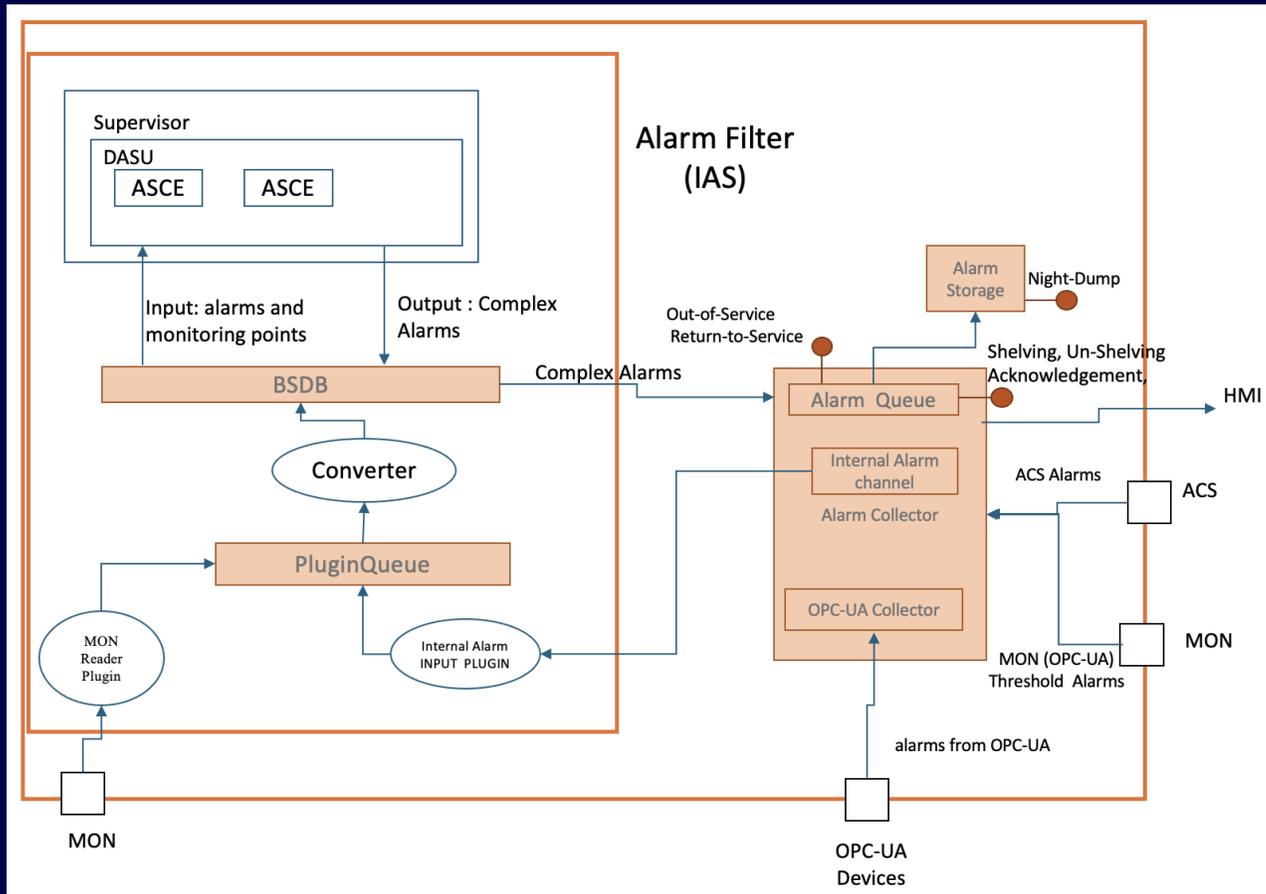
Logging deployment



Logging information flow

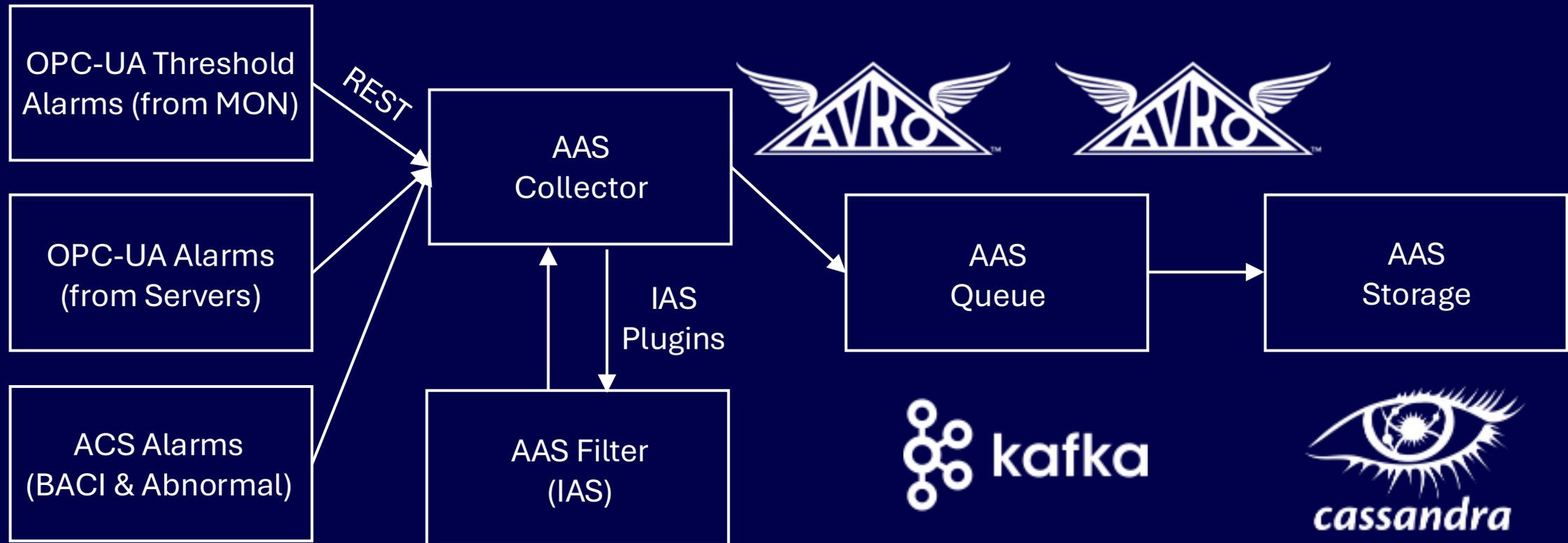


Alarm deployment



Implemented through the
Integrated Alarm System
(IAS)

Alarm information flow



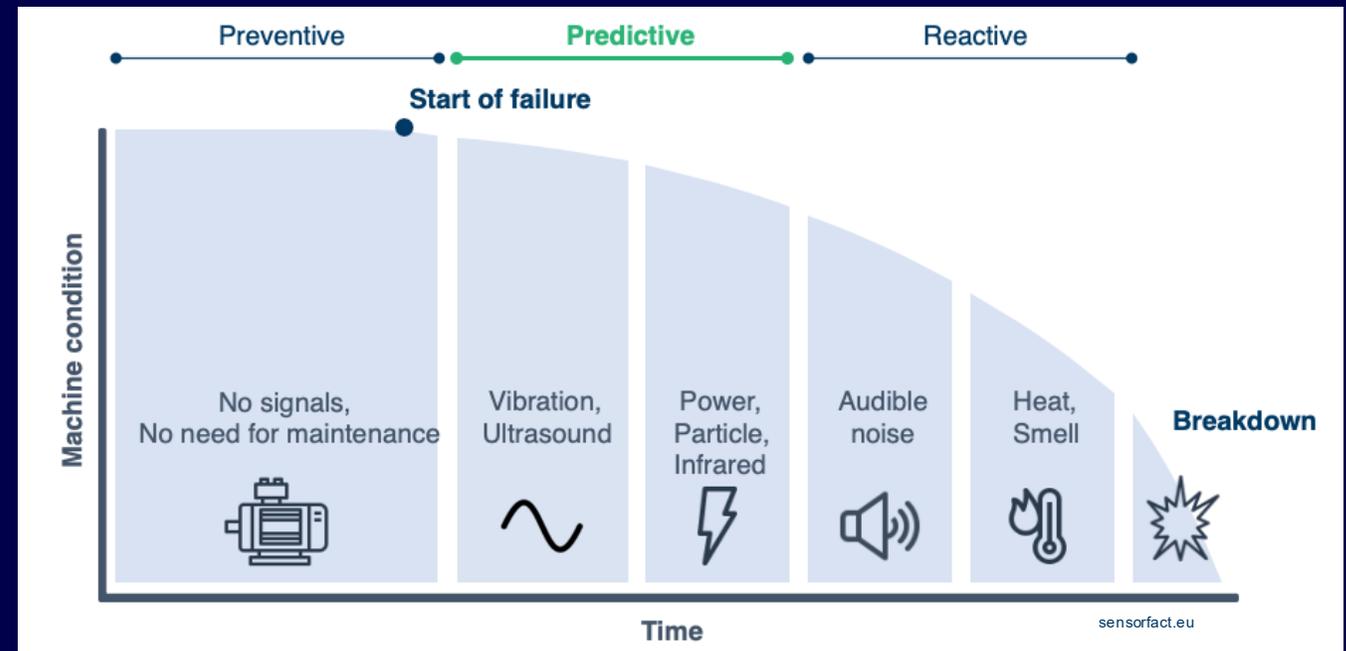
Towards predictive maintenance



```

2020-11-11 13:52:28 DEBUG Adapters:38 - Adapter: mongo/v1 (Mongo)
2020-11-11 13:52:31 INFO app - Started listening...
2020-11-11 13:52:27 ERROR Oracle:22 - Failed to write to DB. Error
2020-11-11 13:52:27 ERROR Users:24 - Failed to create JSON (E: mis
2020-11-11 13:52:14 DEBUG BatchWriter:28 - Successfully connected
    
```

Enables the **replacement** of a component **just in time** (when it is close to failure) **improving the system life cycle**:



lower the facility inactivity



increase the operational time

extend the components' lifespan

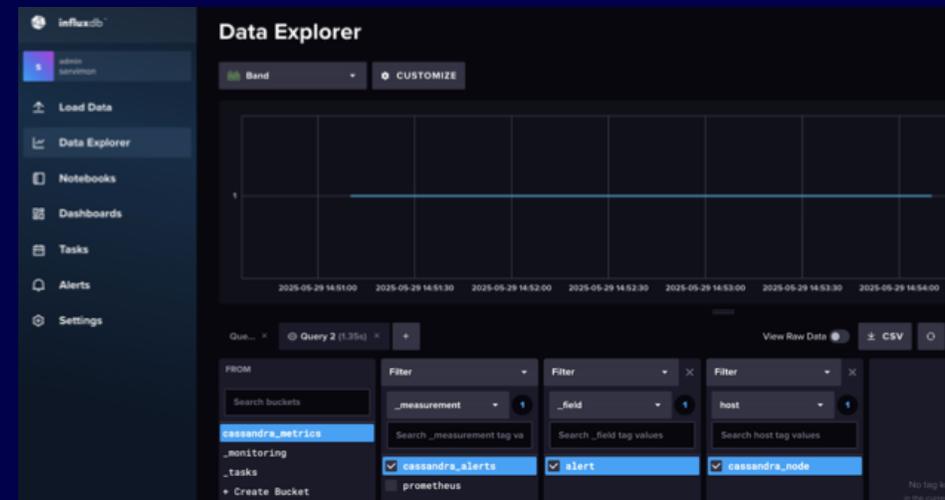
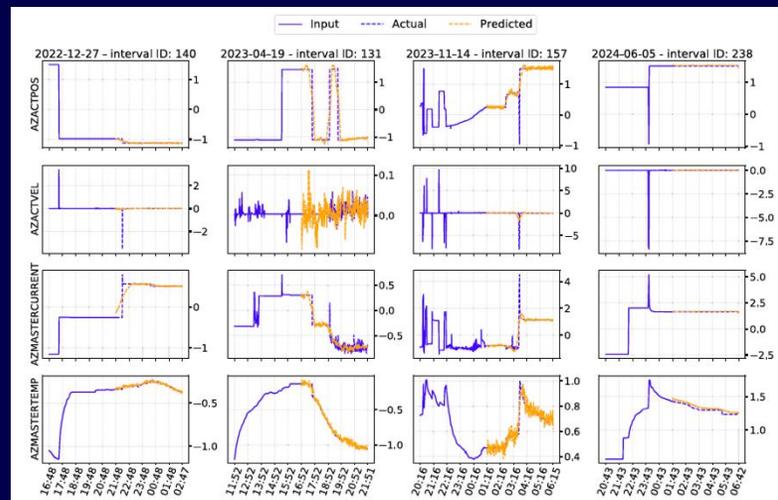


reduce the maintenance costs

Towards predictive maintenance

[1] K. Munari, K., A. Costa, F. Incardona, E. Mastriani, S. Spinello, S. Germani, P. Bruno, *Enhancing CTAO Monitoring and Alarm Subsystems in Distributed Environments Using ServiMon*, Proceedings of 39th International Cosmic Ray Conference — PoS(ICRC2025), <https://doi.org/10.22323/1.501.0775>

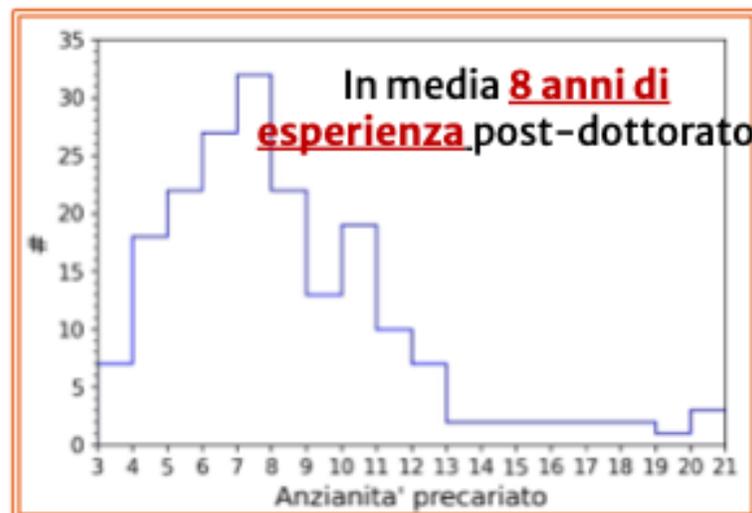
[2] F. Incardona, A. Costa, F. Farsian, F. Franchina, G. Leto, E. Mastriani, K. Munari, G. Pareschi, S. Scuderi, S. Spinello, G. Tosti, *Multivariate time-series forecasting of ASTRI-Horn monitoring data: A Normal Behavior Model*, Astronomy and Computing, 55 (2026), <https://doi.org/10.1016/j.ascom.2026.101071>.



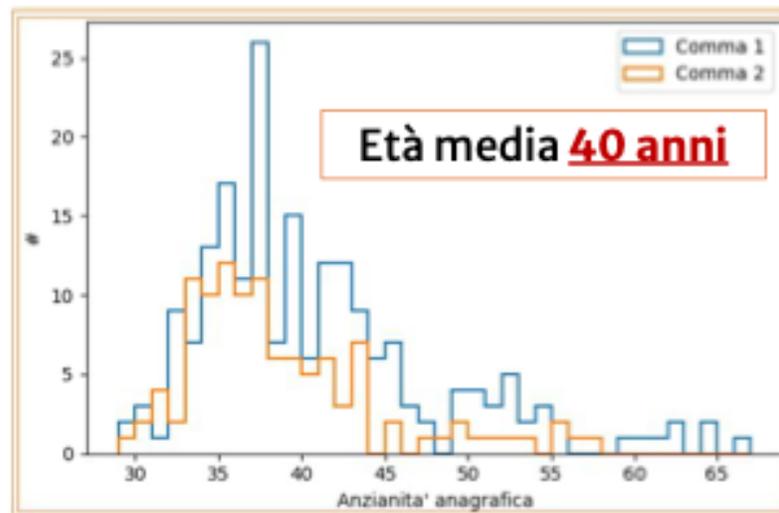
Thank you

La situazione del personale precario in INAF è **INSOSTENIBILE!**

1.200 Tempo Indeterminato Vs **650** precari: più di 1 precario ogni 2 persone di ruolo



Plot di un campione rappresentativo dei precari INAF al 31/12/2024



Dei **650**, **287** possono essere stabilizzati:
173 tramite chiamata diretta (comma 1)
114 tramite concorsi riservati (comma 2)

Entro l'anno, l'attuale situazione determinerà l'esodo di > 100 lavoratori altamente qualificati e il MUR se ne lava le mani

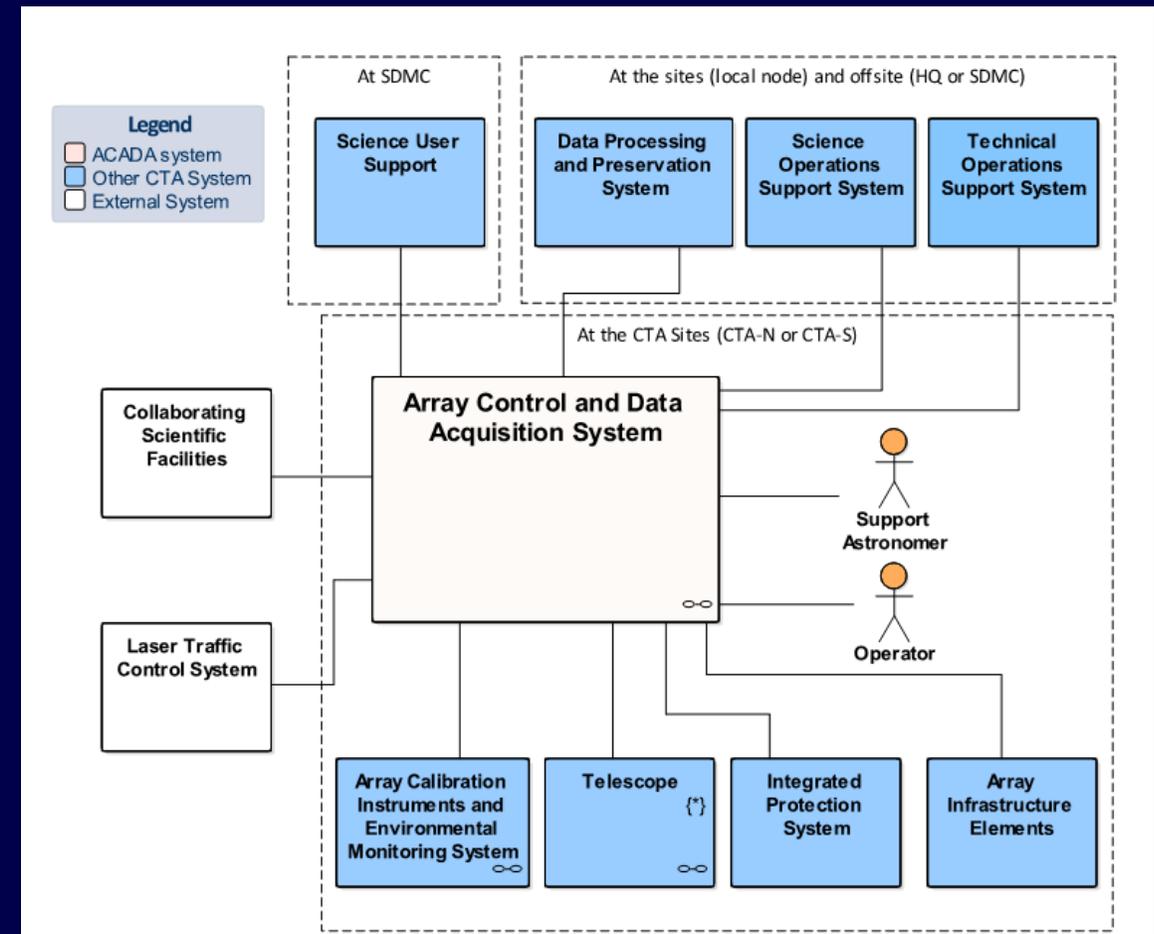
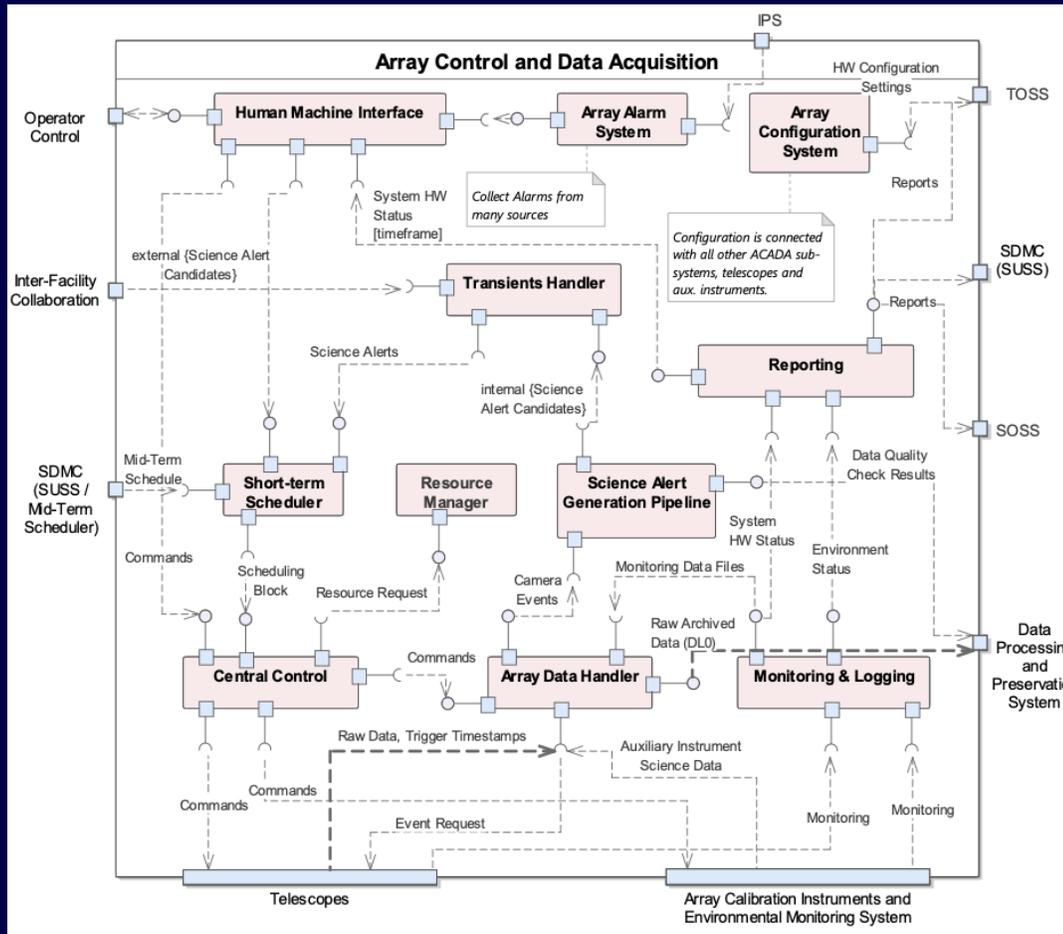
È **URGENTE** che INAF **PROCEDA ORA** con le **STABILIZZAZIONI TRAMITE MADIA:** unica soluzione per questa emergenza



Molti colleghi (972) hanno già firmato, per sostenerci e aggiungere il nome alla lista del QR,
contattaci a retestabilizzandi1.inaf@gmail.com



ACADA in CTAO: context



Monitoring features and configs

Configuration:

ACS: ACS Manager discovery + ACS CDB entries

OPC-UA: ACADA CDB → attempts every 2 minutes

Callback-Based Monitoring

Remote monitors

Asynchronous callback delivery

Monitoring instance per property

Sampling Control

Configurable sampling interval

→ protects network bandwidth

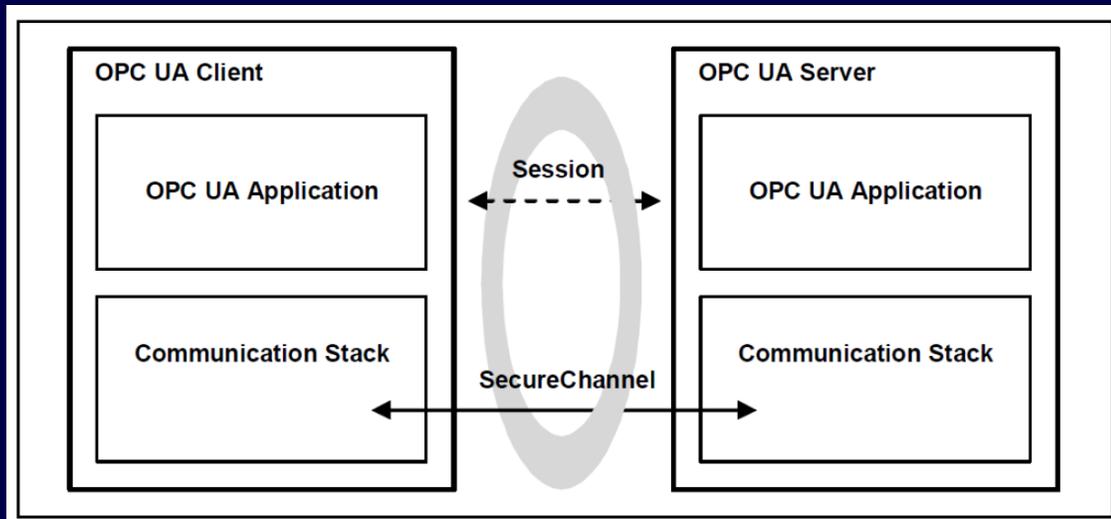
Minimum trigger interval

Per-property configuration (not global)

On-Change Mode

Callback triggered only on significant value changes

Optimized for event-driven monitoring

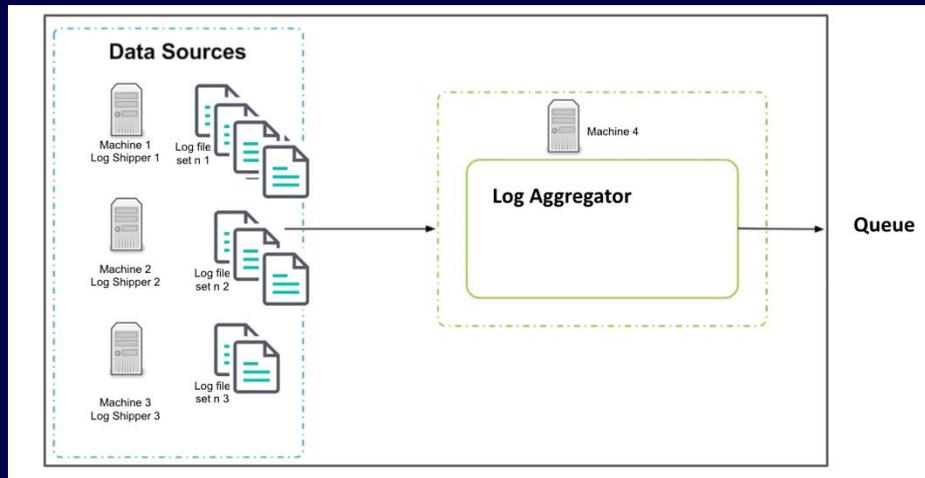


Logging features and configs

Configuration:

Client configuration: logging framework, ACS

Server configuration: Filebeat, Logstash, ACS
CDB, ACADA CDB



Avro LogEvent schema (main fields)

recordId, sourceTimestamp, logEntryType, message

Log Levels

TRACE, DEBUG, INFO, NOTICE, WARNING,
ERROR, CRITICAL, ALERT, EMERGENCY

Logging Analyzer (future development)

analyzes logs to trigger additional alarms/warnings

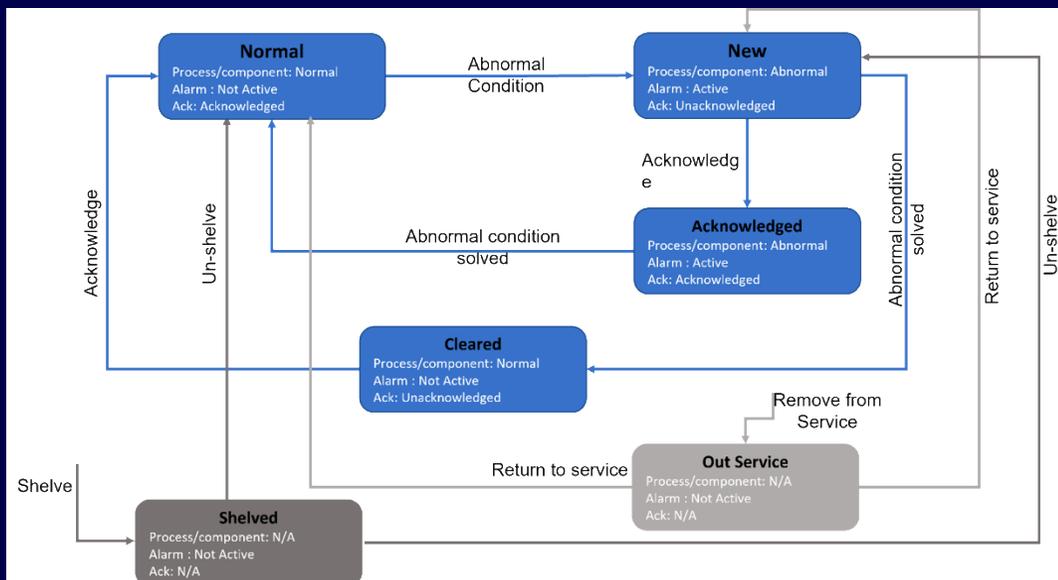
Alarm features and configs

Configuration:

ACS: ACS CDB

OPC-UA: ACADA CDB

IAS: Plug-in, IAS CDB, transfer functions



Alarm Collection

Consumes ACS alarms

Receives OPC-UA threshold alarms

Receives complex alarms from IAS

Publishes events to Kafka

Supports: Shelving, Acknowledgement, OOS

Alarm Filtering (IAS-based)

Implements Transfer Functions

Generates complex alarms

Executes operational scripts (high priority)

Logging system overview (distributed collection)

- Handles many log sources:
 - ACS logs, observation scripts, firmware/low-level logs, syslogs, OPC-UA server logs
- **Logging Collector** is distributed:
 - **log shippers** on each host harvest files
 - buffering + filtering at the edge
 - forward to a central **Log Aggregator**
- Aggregator routes to **Logging Storage** and **Logging Analyzer**

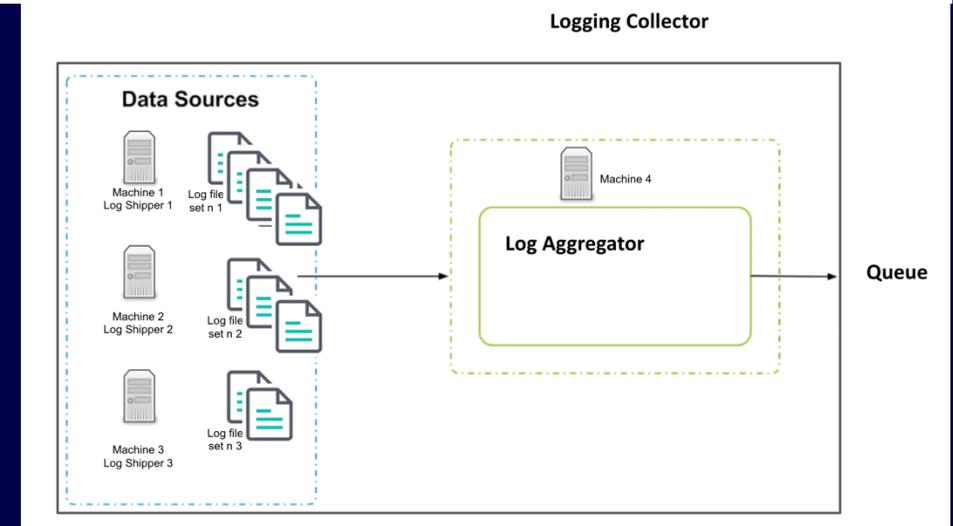
Log event model, severity, and alarms

- Log events follow an Avro **LogEvent** schema (core fields):
 - recordId, sourceTimestamp, logEntryType, message
- Severity taxonomy:
 - TRACE, DEBUG, INFO, NOTICE, WARNING, ERROR
 - CRITICAL, ALERT, EMERGENCY → should reach operators via HMI
- **Logging Analyzer**:
 - analyzes logs to trigger additional alarms/warnings

Table 5: Monitoring Value Avro Schema

```

{
  "namespace": "cta.log.kafka",
  "type": "record",
  "name": "LogEvent",
  "fields": [
    # recordId unique key assigned to the log event
    {"name": "recordId", "type": "string"},
    # sourceTimestamp specifies the exact time when the log entry was submitted.
    {"name": "sourceTimestamp", "type": "double"},
    # logEntryType: "TRACE", "DEBUG", "INFO", "NOTICE", "WARNING", "ERROR", "CRITICAL", "ALERT", "EMERGENCY"
    {"name": "logEntryType", "type": "string"},
    {"name": "message", "type": "string"},
  ]
}
    
```



Logical architecture (high level) (2)

Component	Description
Monitoring Collector	The Monitoring Collector is a component that collect and serialize monitoring information from ACADA internal processes and from all Controllable Systems and Devices on a CTA site in an asvncronous fashion.
Monitoring Storage	A component inside the MON that stores monitored quantities according to changes in value based on configurable threshold criteria.
Monitoring Value Inspector	A component that provides the capability to resample the monitoring information coming in the form of irregular and unevenly spaced time series data to a consistent and regular frequency.
Logging Collector	This component is responsible for logging information from relevant software components such as: Software logs provided by elements using the control framework, Software logs of the observation scripts, Software logs produced by low-level firmware.
Logging Analyzer	Logging Analyzer is a component that analyses logging data information to trigger further alarms, as well as warnings for the technical crew.
Logging Storage	The Logging Storage is a component inside the MON that stores logging events quantities according to desired log entry level.
Monitoring and Logging Supervisor	This component <u>receives startup and shut-down</u> commands by the and passes them to the MON systems. It provides also the MON status information to the RM

Monitoring inputs and configuration

- Monitoring covers two main source families:
 - ACS Monitoring Properties
 - OPC-UA Monitoring Points
- Configuration:
 - ACS: uses **ACS Manager discovery + CDB entries** (validated XML delivered by element teams)
 - OPC-UA: uses the ACADA –CDB

Callback-Based Monitoring

Remote monitors created via:

```
create_monitor(...)
```

```
create_postponed_monitor(...)
```

Asynchronous callback delivery

Monitoring instance per property

Sampling Control

Configurable sampling interval

```
→ set_timer_trigger(...)
```

Enforces minimum trigger interval

```
→ protects network bandwidth
```

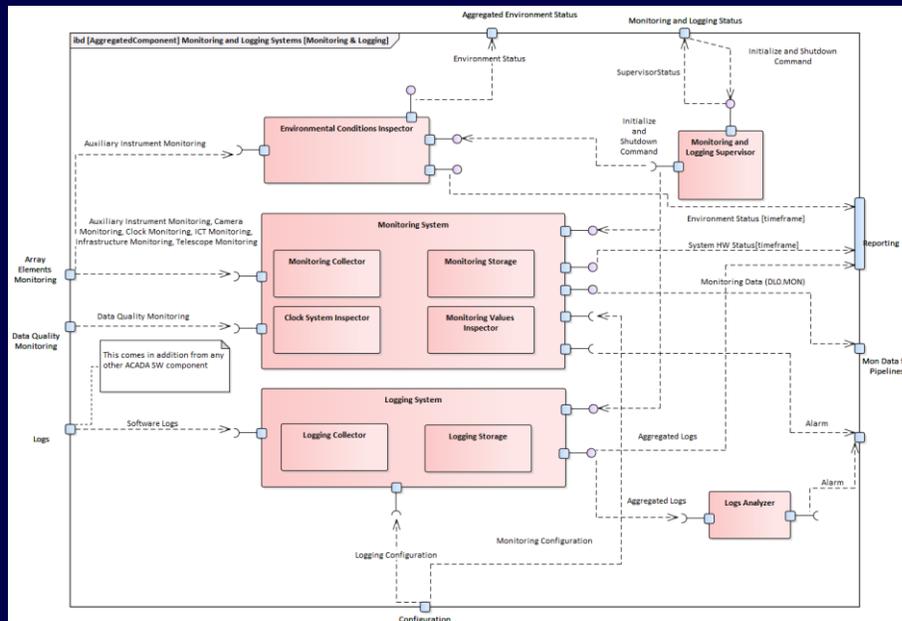
Per-monitor configuration (not global)

On-Change Mode

Timer disabled (timer = 0)

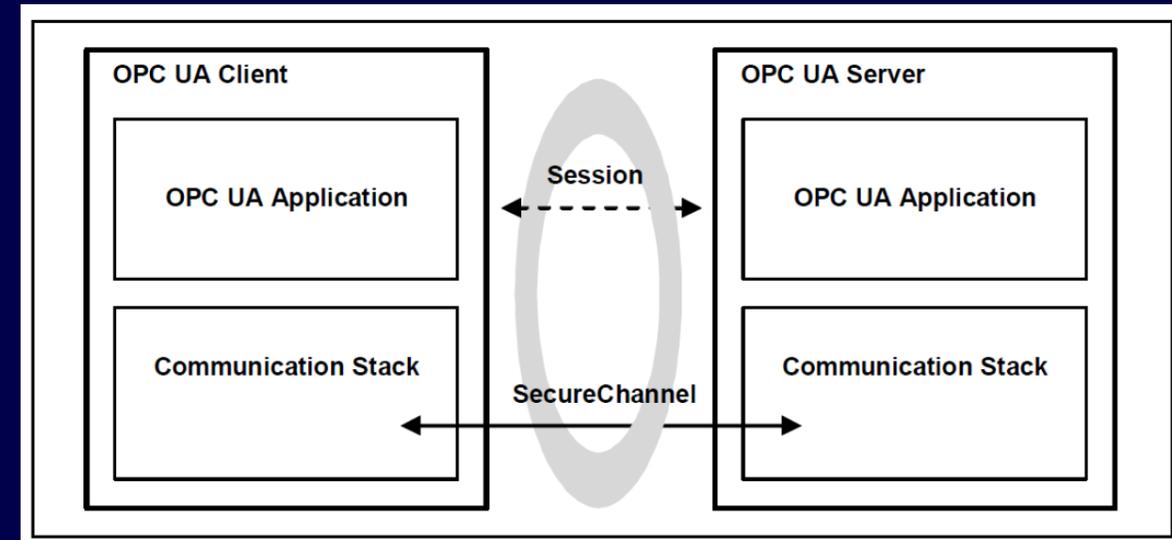
Callback triggered only on significant value changes

Optimized for event-driven monitoring



Monitoring Collector (OPC-UA mode)

- Client–server approach based on **OPC-UA sessions**
- Discovery:
 - direct connection or via **Local Discovery Server**
- Security and robustness:
 - secure channel
 - client certificate generation/usage
- Collection model:
 - **subscriptions + monitored items**
 - asynchronous tasks per monitored server/config



Key design decisions and technology stack

- **Implementation choices:**

- **Java** for MON components (robustness, ACS alignment)
- **OPC-UA sessions** for monitoring (secure, reliable communication)
- **Eclipse Milo** as OPC-UA SDK (pure Java)
- **Filebeat** (log shipper) preferred over logging libraries for resilience

- **Core technologies:**

- Kafka clients, Avro, Cassandra, SLF4J (and supporting libraries)

Log event model, severity, and alarms

- Log events follow an Avro **LogEvent** schema (core fields):
 - recordId, sourceTimestamp, logEntryType, message
- Severity taxonomy:
 - TRACE, DEBUG, INFO, NOTICE, WARNING, ERROR
 - CRITICAL, ALERT, EMERGENCY → should reach operators via HMI
- **Logging Analyzer:**
 - analyzes logs to trigger additional alarms/warnings

AAS Architecture Overview (1)

Alarm Collector

- Consumes ACS alarms
- Receives OPC-UA threshold alarms
- Receives complex alarms from IAS
- Publishes events to Kafka
- Supports: Shelving, Acknowledgement, OOS

Alarm Filter (IAS-based)

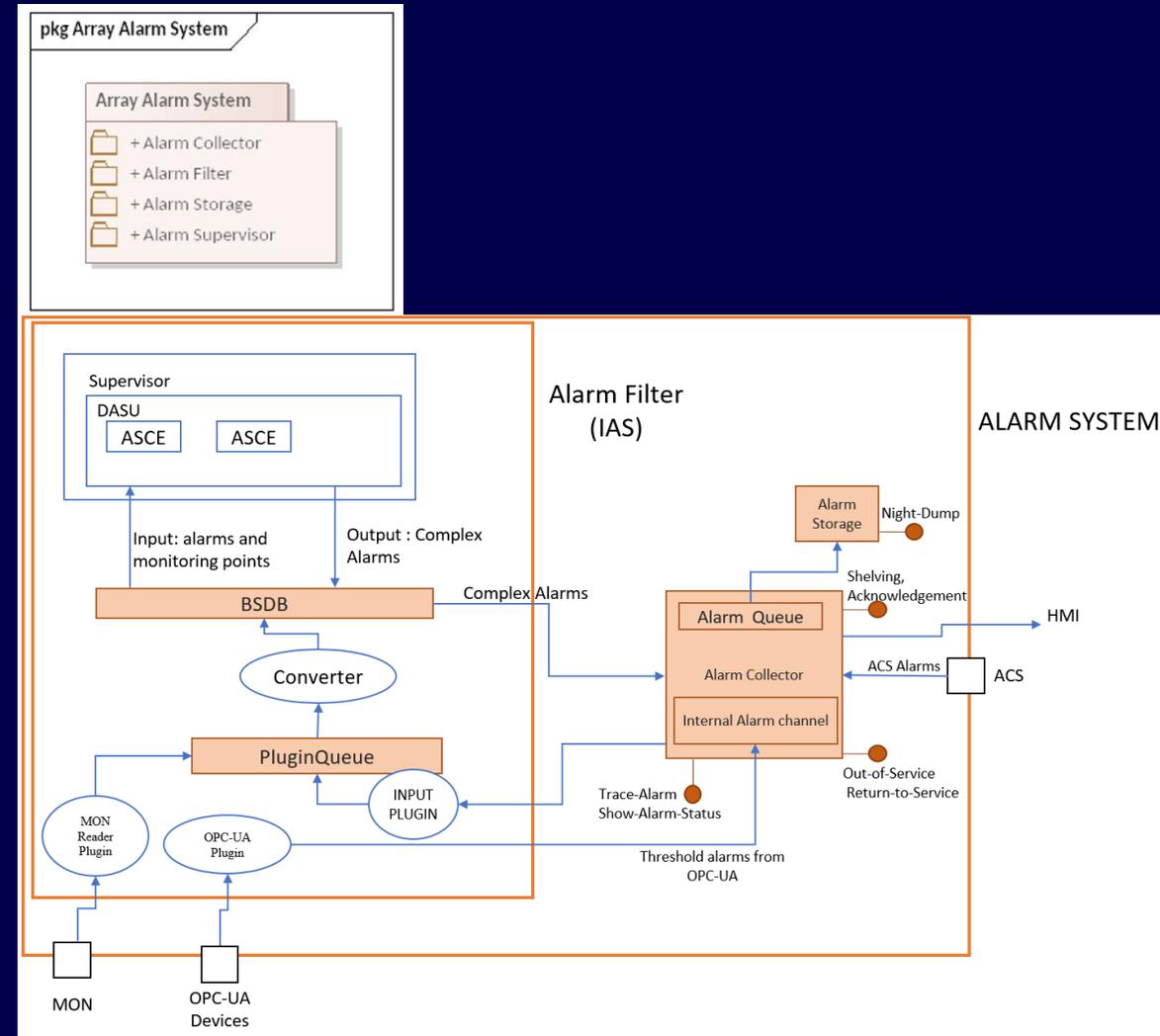
- Implements Transfer Functions
- Generates complex alarms
- Executes operational scripts (high priority)

Alarm Storage

- Consumes Kafka topics
- Batch insertion into Cassandra
- 24h data availability for TOSS

Alarm Supervisor

- Manages lifecycle of AAS components
- Interfaces with Resource Manager



AAS Architecture Overview (2)

Component	Description
Alarm Collector	Delivers the capabilities for gathering ACS or OPC-UA alarms raised by Array Elements or ACADA components. The Alarm Collector obtains ACS alarms generated by ACS through the OPC-UA Values Reader Plugin and the Alarm Filter (IAS). The design of the Alarm Collector encompasses features like Shelving, Un-Shelving, and Acknowledgement, providing operators the ability to interact with the status of each alarm. Moreover, it incorporates functions such as Out-Of-Service and Return-to-Service.
Alarm Filter	The implementation of the Alarm Filter component utilizes the IAS [RD18], [RD20]. The Alarm Filter provides means to create alarms according to a set of defined rules. The Alarm Filter makes use of the Alarm Rules Database a database defining the alarm reduction rules for the Alarm Filter.
Alarm Storage	The purpose of the Alarm Storage is to retrieve alarm data from the Kafka topics supplied by the Alarm Collector and conduct the insertion query into the Cassandra database. It provides a local repository to store alarms. It provides TOSS with the last 24 hours alarm events
AAS Supervisor	Supervisor component for the AAS, connected with the supervision tree provided by RM. It provides also the AAS status information to the RM.

AAS Architecture Overview (3)

