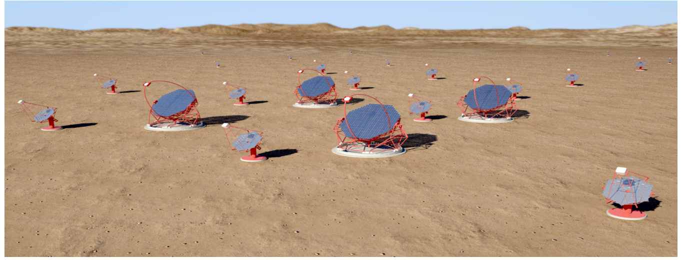
The readout and control system of the mid-size telescope prototype of the Cherenkov Telescope Array

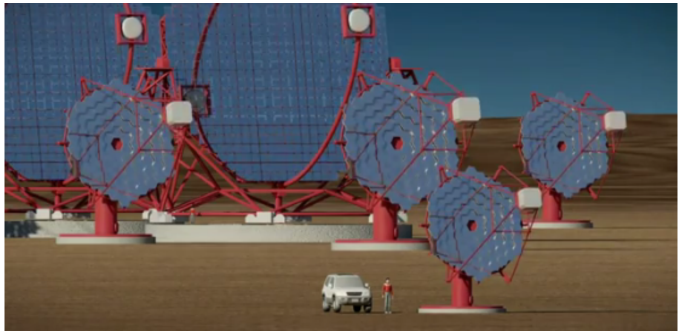
Igor Oya, O. Anguner, B. Behera, E. Birsin, M. Fuessling, D. Melkumyan, T. Schmidt, U. Schwanke, R. Sternberger, P. Wegner, S. Wiesand

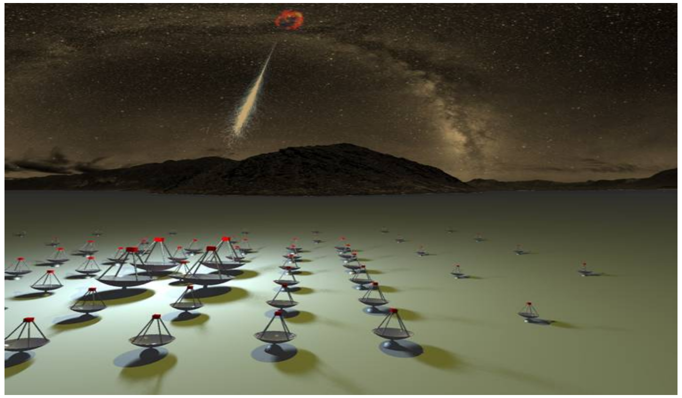
On behalf of the CTA consortium

Cherenkov Telescope Array (CTA)

- Next generation VHE gamma-ray observatory
- Northern and Southern arrays for full-sky coverage
- Energy range → 10's of GeV to above 100 TeV
- Several types of Cherenkov Telescopes will provide:
  - Better Sensitivity
  - Better Energy Range
  - Better Angular Resolution
- The main science goals of the CTA project:
  - Understanding the origin of cosmic rays and their role in the Universe
  - Understanding the nature and variety of particle acceleration around black holes
  - Searching for the ultimate nature of matter and physics beyond the Standard Model

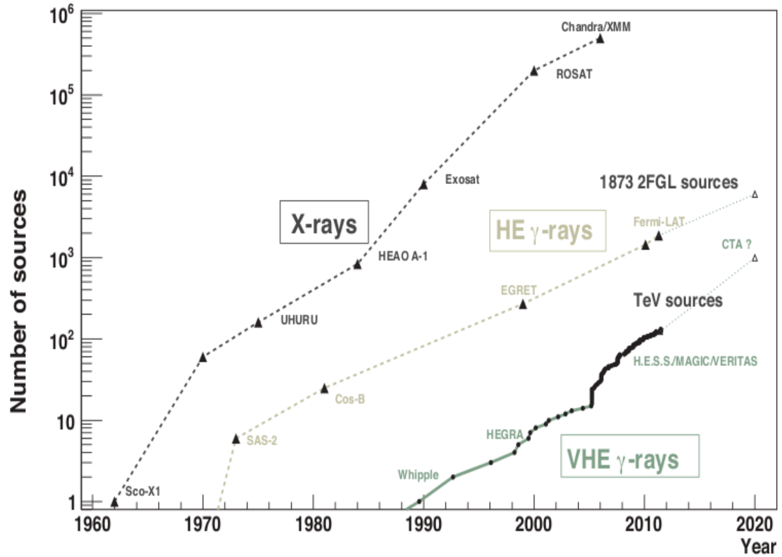


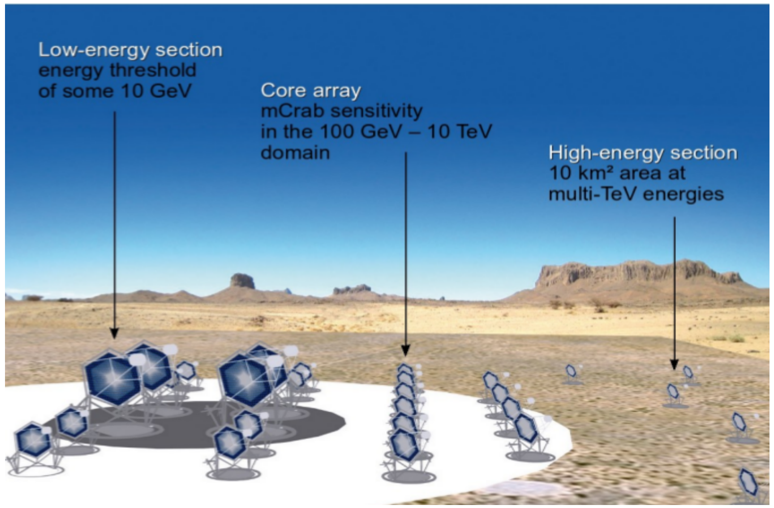




Cherenkov Telescope Array (CTA)

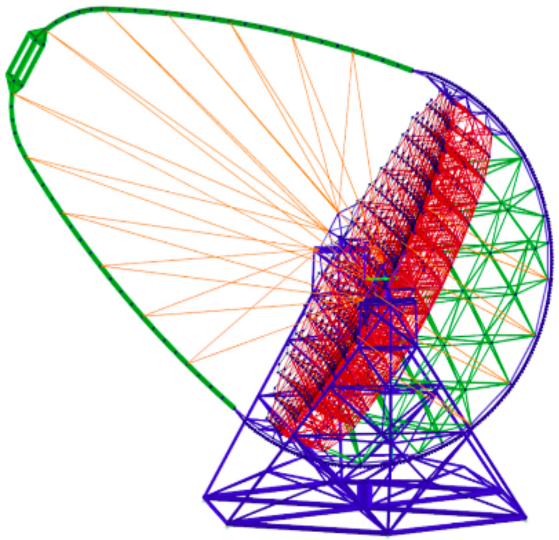
- CTA will be the most sensitive array operating in VHE (E > 10 GeV) domain
- Number of detected VHE source will be greatly increased
- Enhanced angular resolution will allow detailed morphology studies
- CTA will detect weak sources within minutes, rather than hours.
- (Northern and Southern Array) - Total cost ~200M€

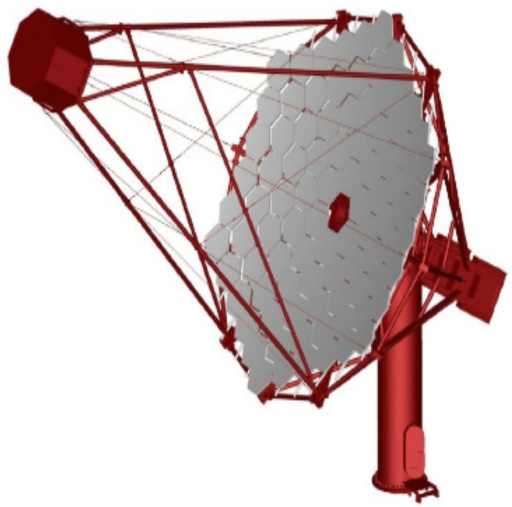


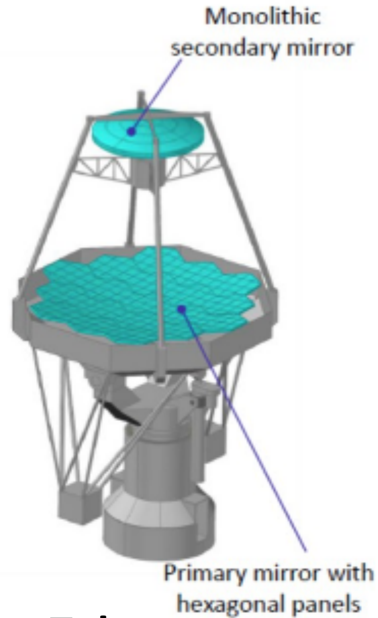


CTA telescope types

- Highly sensitive within a wide energy range (10's of GeV to >100 TeV)







Large Size Telescope (LST)

- 23 m diameter
- E < 100 GeV
- Parabolic Mirror
- ~4 Telescopes
- 5° Field of View (FoV)
- Improved photon collection

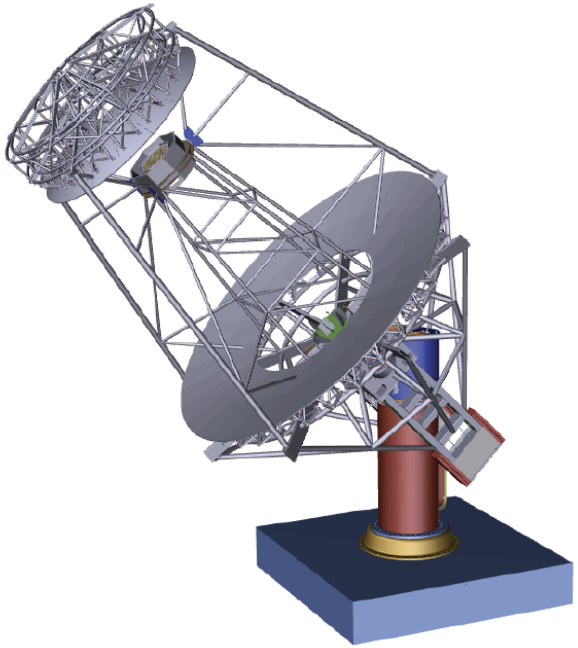
Mid Size Telescope (MST)

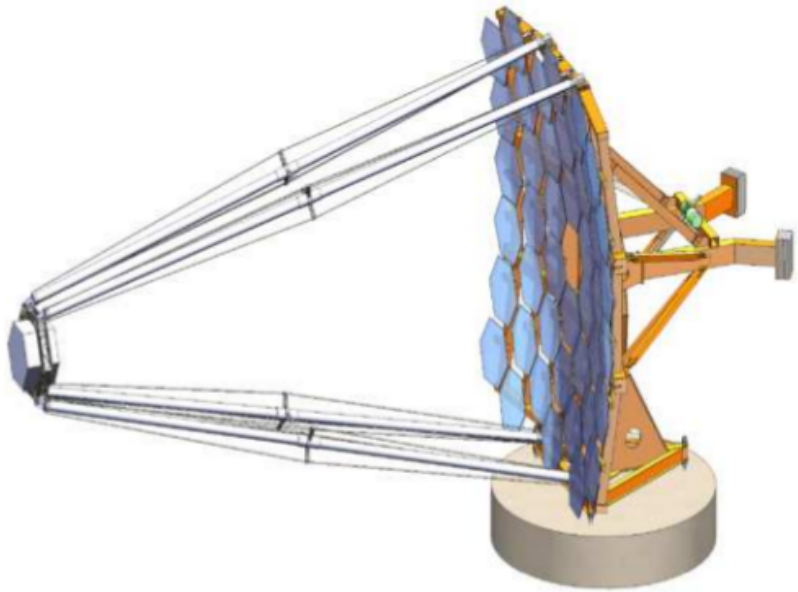
- 12 m diameter
- 100 GeV < E < 10 TeV (The Core Energy Range)
- Davies-Cotton design
- ~25 Telescopes
- 7 - 8° FoV
- Improved Gamma / hadron separation

Small Size Telescope (SST)

- 4-7 m diameter
- >10 TeV
- Schwarzschild-Couder
- ~30 Telescopes
- 10° FoV
- Improved effective area
- Southern site only

CTA telescopes, further Designs





MST

- CTA-USA Design (AGIS*)
- Schwarzschild-Couder
- 9.7 m diameter (primary mirror)
- 8° FoV
- 20-30 telescopes

*Advanced Gamma-ray Imaging System

SST

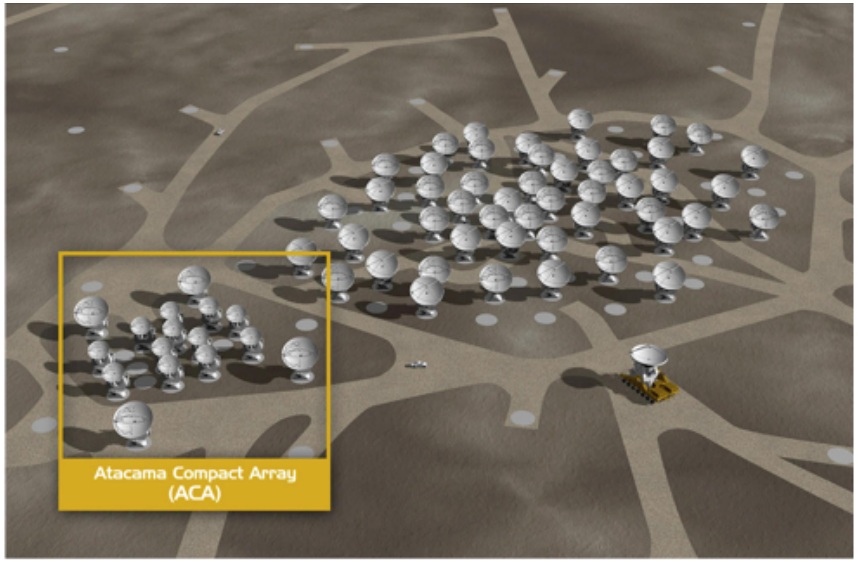
- Davies-Cotton Design
- Large FoV needs relatively larger camera
- Costs more
- Less complex design

The Array control challenge

- CTA will be an open astronomical observatory:
  - Announcements of opportunities
  - Proposal-driven operation
  - Professional operators
- Diverse operation modes:
  - Pointed observations
  - Scans
  - Sub-Arrays
  - Targets of opportunity
  - Multi-wavelength campaigns
  - Alerts from/to other observatories
- Pseudo real-time analysis

- Select and time-tag stereoscopic showers at high efficiency while suppressing background.
- Read-out and store all the information needed for offline reconstruction.
- Support diverse operation modes
- Ensure optimal use of available observation time
- Handle hardware and software failures in a flexible and smart way

CTA array control approach



- Attempt to re-use the experience from current-generation experiments
- Use commodity hardware for processing, networking, and triggering whenever possible
- Promote software standard for the communication of subsystems
- Use a well-defined software framework for all processes in the distributed ACTL system
- Re-use developments made for observatories with similar requirements – in particular for ALMA
- Ensure seamless integration of offline reconstruction code

Software technologies for CTA

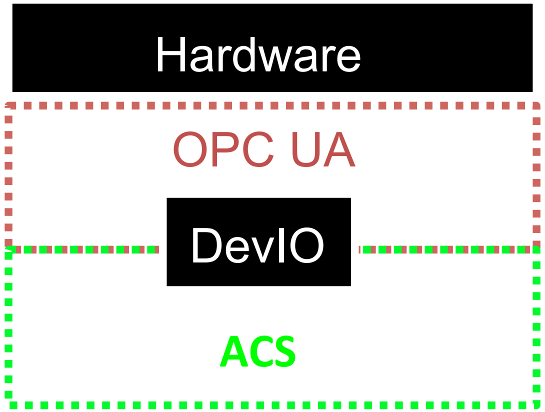
ACS: ALMA Common Software

- Framework for distributed applications used for the control of the ALMA array, one of the largest astronomical projects, similar in many aspects to CTA
- C++, Java and Python implementations
- Framework for a distributed system
- Environment for implementing CORBA applications
- Uses Container/Component Model

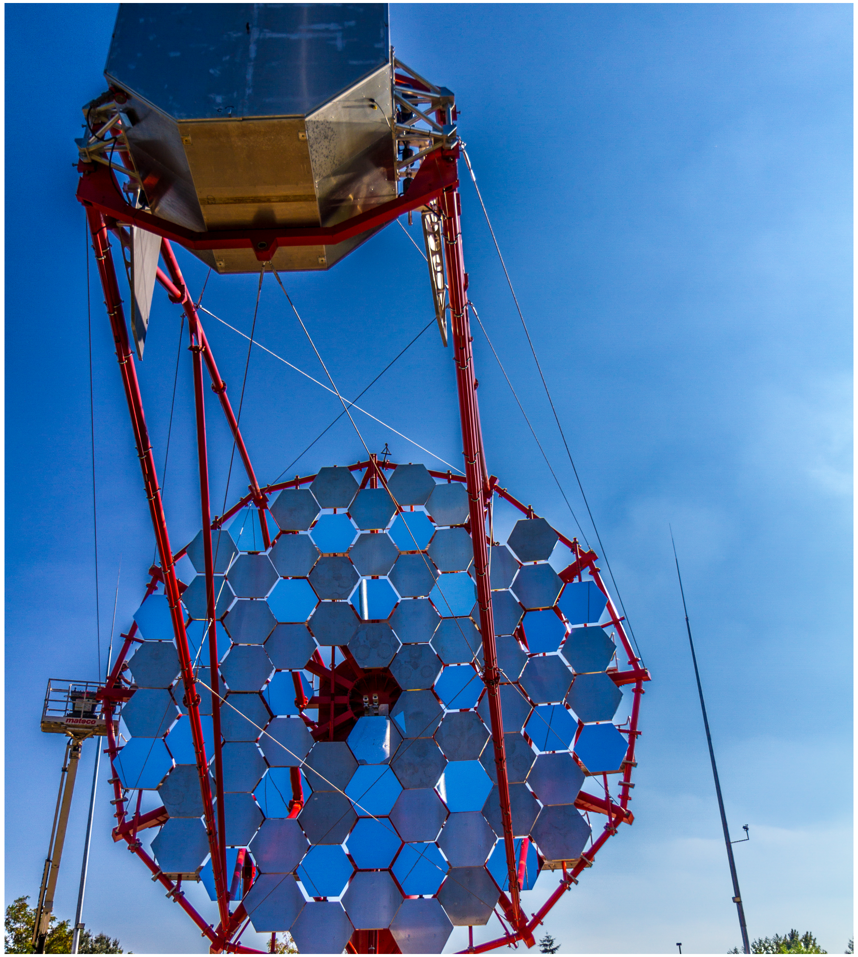
OPC UA: OPC Unified Architecture

- Open Platform Communications specification from the OPC Foundation
- A cross-platform service-oriented architecture for process control
- Multi-platform implementation, including portable ANSI C, C++, C#, Java and .NET implementations
- High scalability: from smart sensors and actuators to mainframes and servers on embedded systems

DevIO: *is an ACS simple and generic abstraction of hardware monitor and control point, based on the Bridge design pattern.*



The MST prototype: characteristics



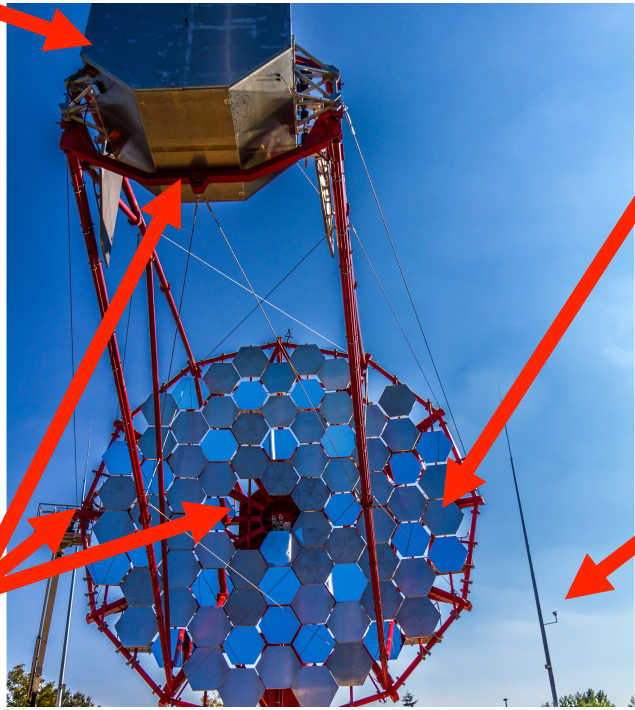
- Located at Berlin-Adlershof
- *Davies-Cotton* design
- Effective mirror area: ~ 100 m2
- Dish diameter D=12m
- Focal length F=16m (>1.3xD)
- Dish curvature radius=19.2m (=1.2×F)
- Camera FoV 8°
- Hexagonal mirrors, 1.2m, 35 kg/m2
- Slewing speed: Rotate to any point in the sky above 30° in elevation≤90 s
- Optical PSF (80% containment diameter) ≤ 1 pixel diameter out to 80% of the camera radius
- Post-calibration pointing precision ≤ 7 arcsec rms in space (5" in elevation and 5"/ sin(elevation) in azimuth)
- Lifetime expectancy (bearings and structure) ~ 30 years
- Control software based on ACS and OPC UA

The MST prototype instrumentation

Camera dummy
OPC UA server
Java ACS component

<u>Drive System</u>
PLC with **OPC UA** server
Java ACS component

<u>5 CCD units</u>
OPC UA server
Java ACS component
Video display application
Document oriented database



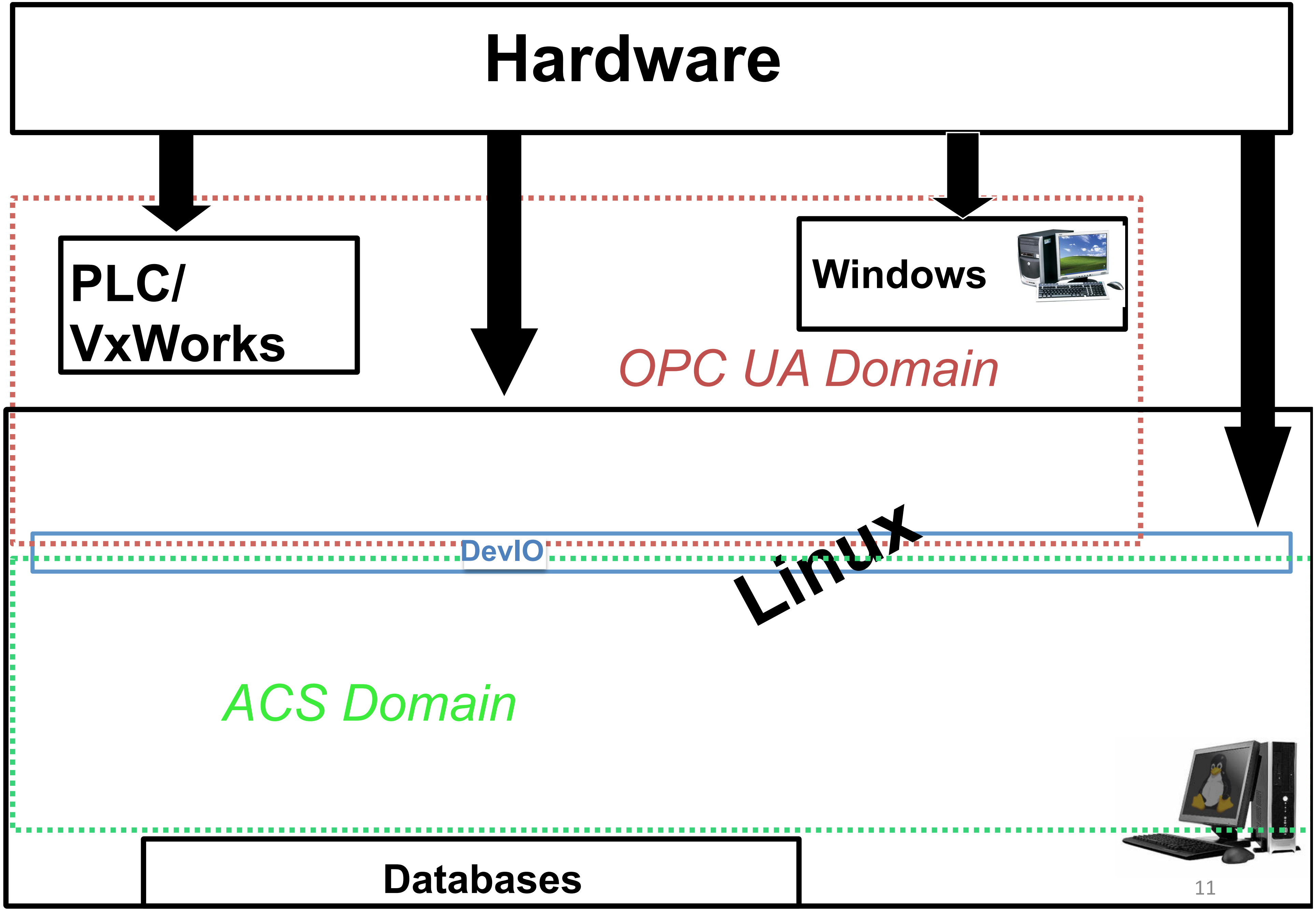
<u>Active mirror control</u>
Different designs:
- Xbee wireless
- CAN-bus

ACS components, common interface

<u>Weather station</u>
OPC UA server
Java ACS component

<u>Telescope Control server</u>
Dell PowerEdgeR720xd
2 CPUs (12 CPU cores)
128GB main memory
6TB local disk space.
Scientific Linux 6.4

MST array control, general view



The CCD Cameras (I)

Devices

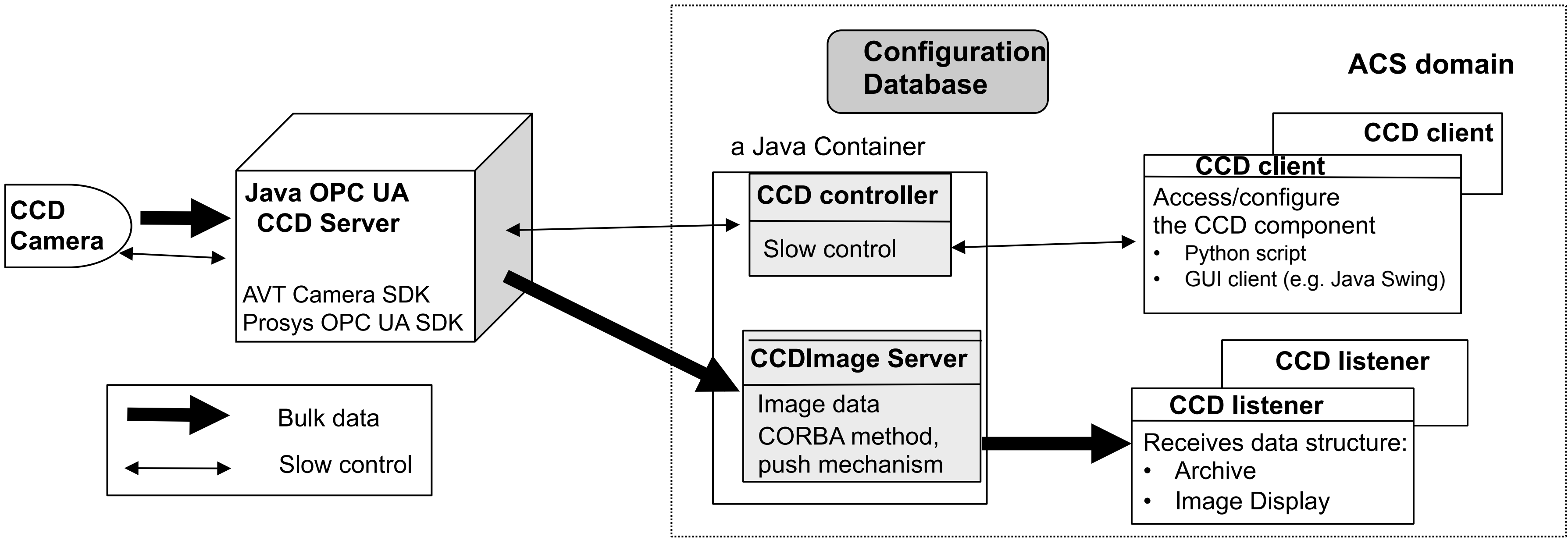
- AVT Prosilica GC1350
- Sony ICX205 ½" Progressive scan CCD
- Maximum 1360 x 1024 Pixels
- Pixel size =4.65 µm
- 8/12 bit depth
- Maximum rate 20 Hz
- Interface: GigE Vision (IEEE 802.3, up to 1000 Mbit/s on 1000BASE-T)

CCD Control software requirements

- 5 CCDs, several or all must operate concurrently. (3 already installed)
- Some of them acquire data at a rate of 10 Hz
- Images need to be correlated with other device measurements (weather station, drive, non optical sensors)

The CCD Cameras (II)

- OPC UA server provides the camera information model (Data&State Model + Basic functions), uses the SDKs from the CCD camera manufacturer for hardware accessing
- Using the Java DevIO and Prosys Java SDKs for OPC UA and ACS communication.
- Image data transferred (push) by a CORBA method call on one or more "listeners" (ACS components or CORBA objects): Idea to test several other data transfer technologies
- Images inserted in the GridFS and linked into the MongoDB together with meta-data

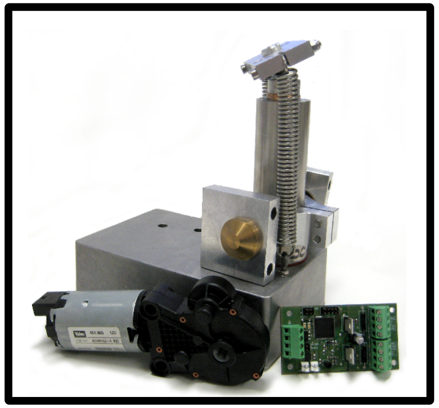


The drive system

- From Bosch-Rexroth
- Composed, at a lower level, of the control of 6 drives (2 for azimuth and 4 for elevation)
- Communicating via a Bosch-Rexrot programmable logic controller (PLC) → Native OPC UA server inside the firmware can be directly accessed via the OPC UA DevIO mechanism
- Pointing to any position and to track any astronomical object.
- Control component implemented as a Java ACS component

The AMC system (I)

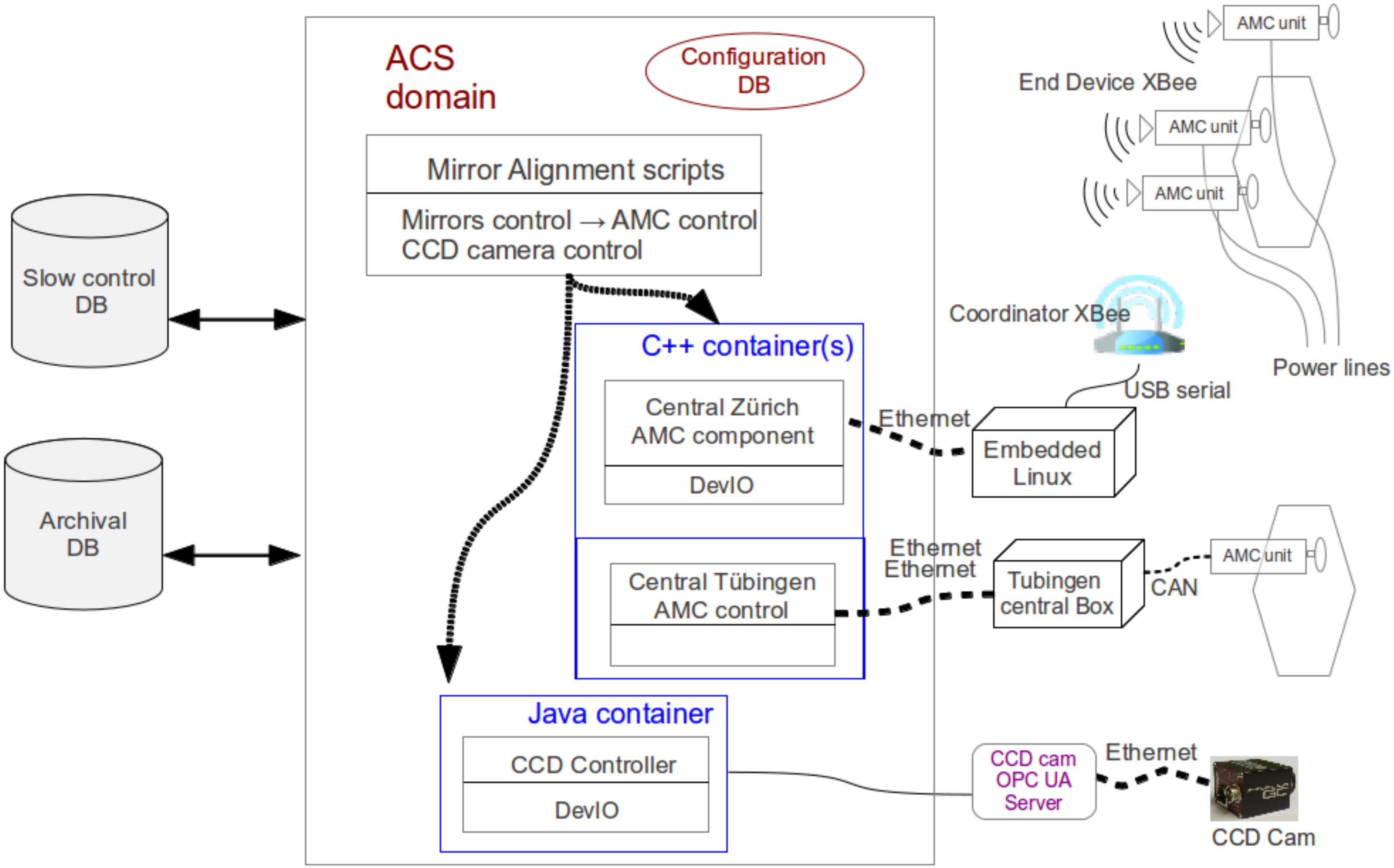
- Use of a tessellated reflector composed of individual mirrors
- Each individual mirror facet is moved by an AMC unit which allows perfect mirror alignment.
- This allows online and off-line re-alignments .
- The dish of the MST prototype is covered by a combination of real and dummy mirrors.
- AMC units of two different designs installed.
  - Zurich: XBee WPAN
  - Tübingen: Can-BUS + Ethernet gateway



Tübingen Actuators

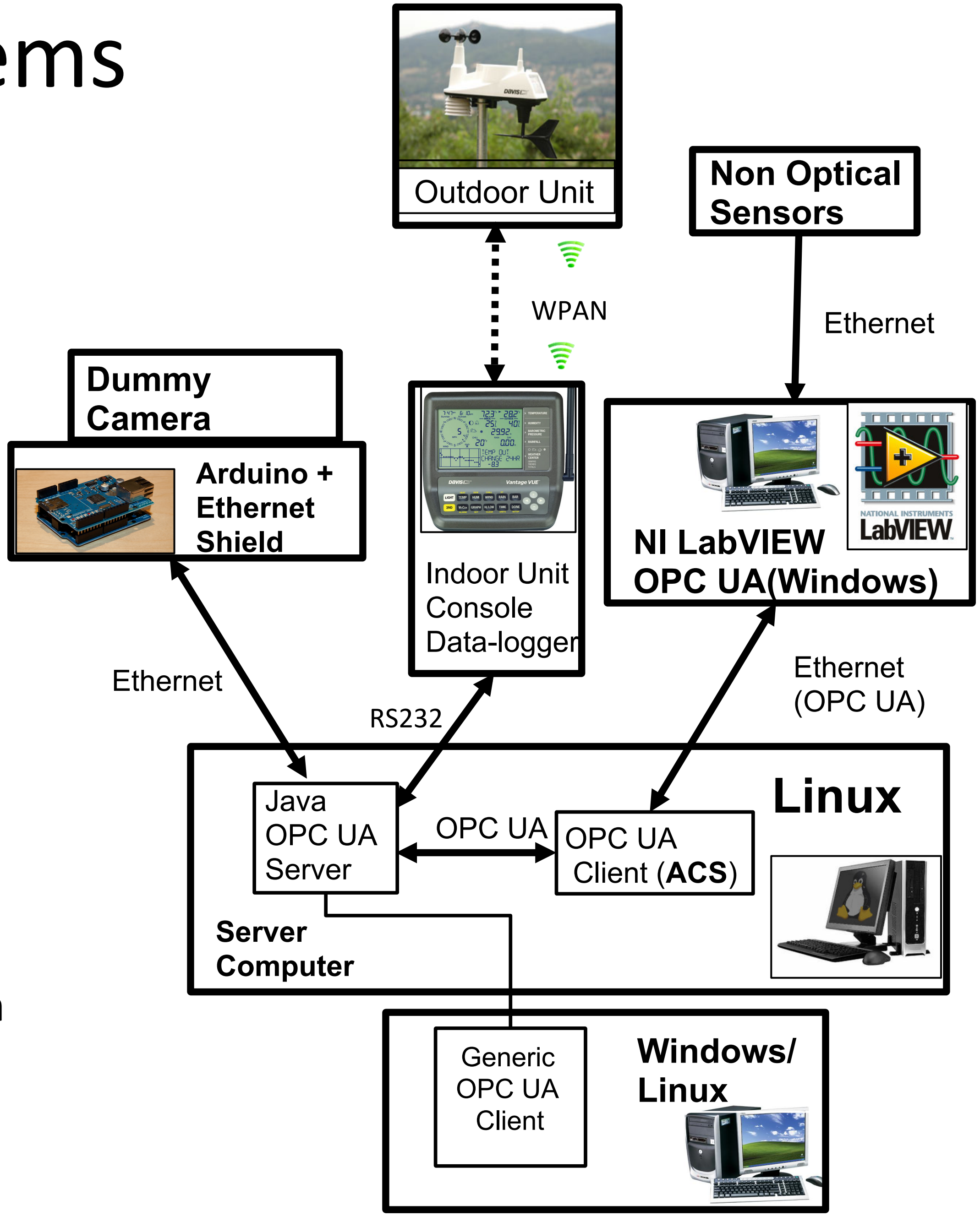
Zurich Actuators

The AMC system (II)



Other systems

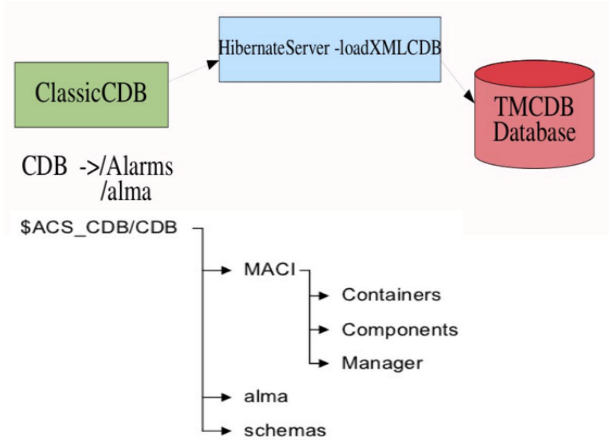
- Weather station
  - Java OPC UA server
  - Java ACS component
  - Running at 24/7 since 3 months
- 'Dummy Camera'
  - Contains sensors, LEDs, and Camera Lids
  - Uses an Arduino controller, Java OPC UA Server and ACS component
- Extra sensors
  - NI Labview on Windows with emdedded OPC UA server



Data persistence and archives

Configuration database (CDB)

- Device configuration and default values
- Front End: ACS TMCDB
- Backend: MySQL

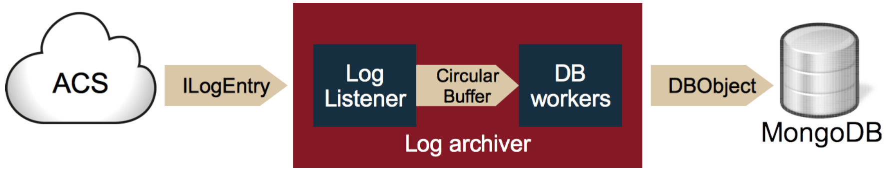


CCD Data storage

- Store the CCD images and headers in the same database
- Front-End: CCD camera readout
- Back-End: MongoDB, images in GridFS

Logging/alarms storage

- Front-End: ACD logging and alarm providers
- Backend: MongoBD



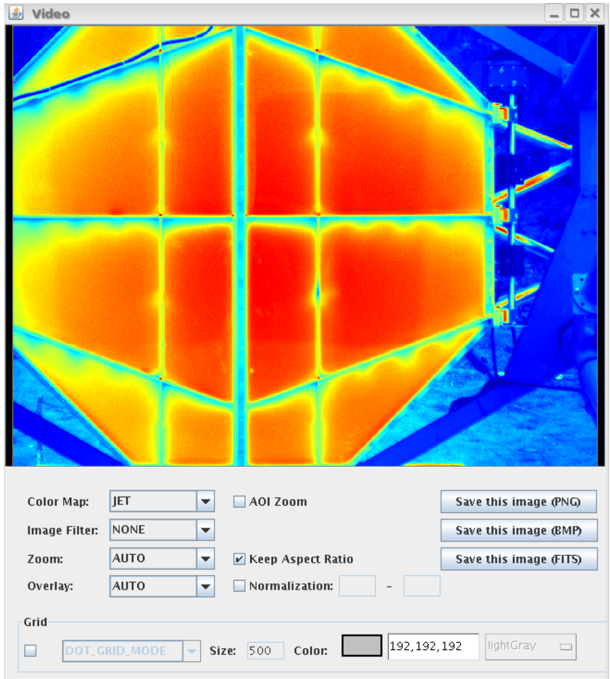
Monitoring storage:

"Property Recorder"

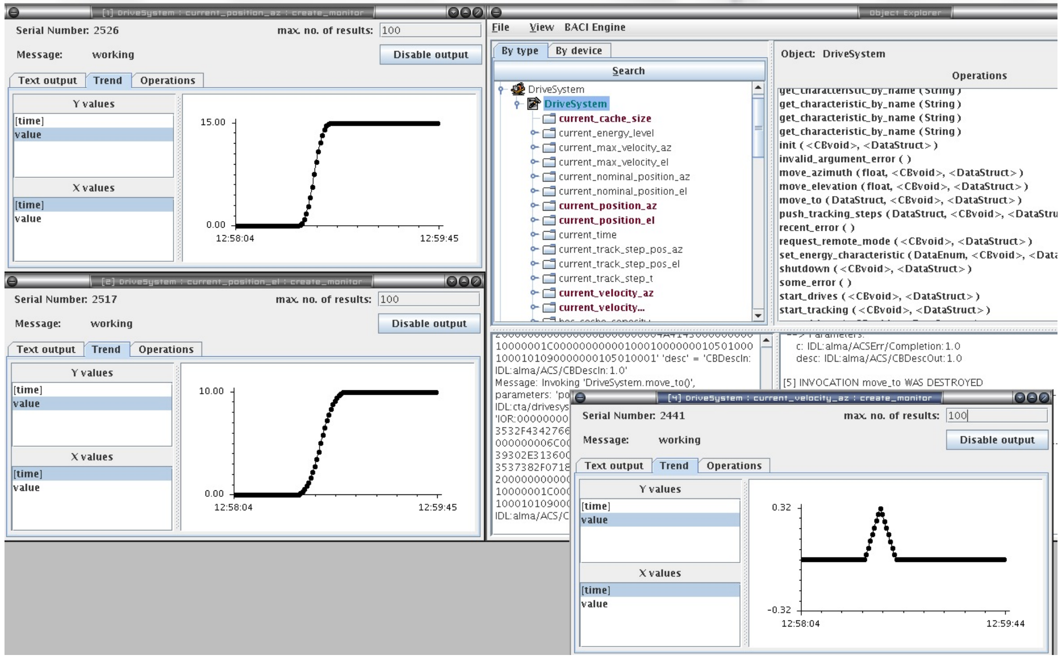
- Monitoring points in device properties
- Periodic and/or value change monitors
- Front-End: Python ACS applications
- Back-End: MongoDB and MySQL

First integrated operations

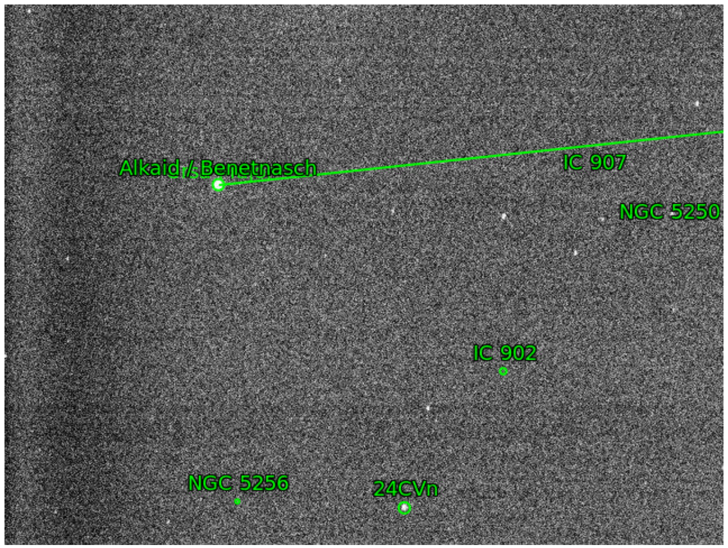
- Move system and display and record CCD images
- Track stars and acquire images with a CCD camera
- Store the images in the MongoDB/GridFS
- Read offline images from the DB and analyze



Lid CCD image display



Drive system control and monitoring GUIs



Identified stars from a CCD image

Conclusions and outlook

- A control and readout software for a telescope is being implemented using the ACS and OPC UA technologies → The CTA ACTL concept is being validated
- First operations have been successfully performed
- Higher level functionality under design and development:
  - Operator GUI
  - Interface to automatic scheduling tools. → Automatic execution of an observing program
  - Web browser based displays for system monitoring plots. *Django* and *Grails* under consideration
- CTA central array control under design → MST test bed