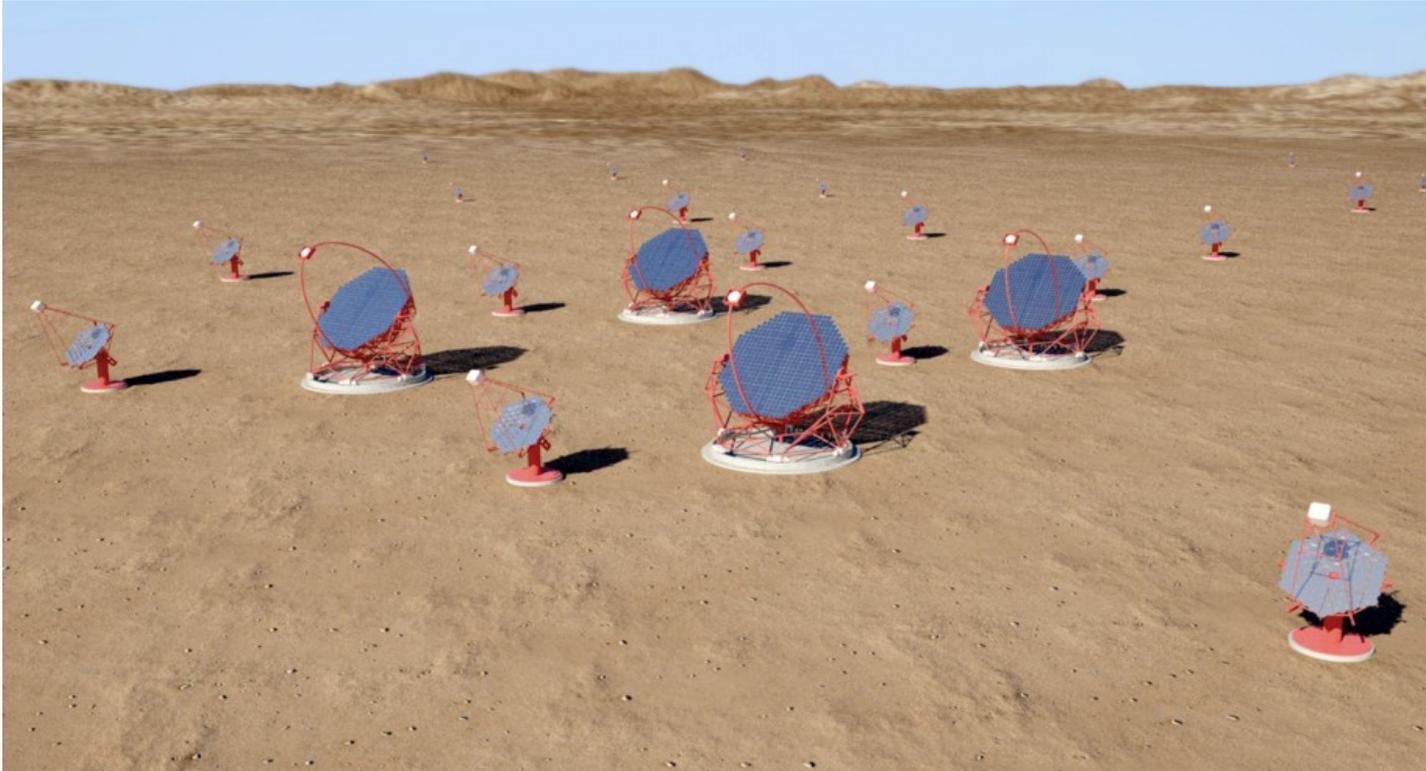


OPC UA Server Development for CTA

Kiran Gothe
Manoranjan

On behalf of HAGAR Team



Reference:

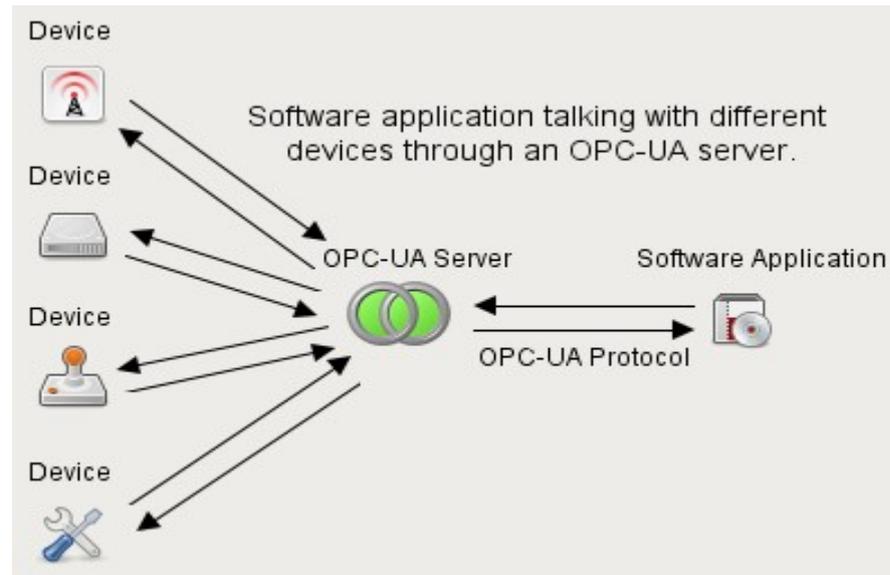
1. <https://opcfoundation.org>
2. Documents on C++ based OPC-UA Server Development by Unified Automation
3. E-Book “OPC Unified Architecture” by Wolfgang et. al.
4. CTA Documents, Presentations etc.

Outline of the talk

- What is OPC UA and why?
- Relevance to CTA
- Server Development for
 - Calibration Box
 - All Sky Camera
- Current Status

OPC-UA

(Open Platform Communications Unified Architecture)



- **OPC-UA** is a new Industry standard for Process control specification released in 2008 by OPC foundation.
- Internationally recognized (**IEC 62541**)
- Lots of products on market place: **>4200 companies** create products based on OPC UA with **+25000 OPC-UA enabled products**
- The purpose of OPC-UA is to provide a way for clients to communicate with different brands of devices without having to implement proprietary device-specific communication protocols used by those devices.
- This is achieved through the use of an **OPC-UA server**. An OPC-UA server handles communication between software applications (client application) and devices.

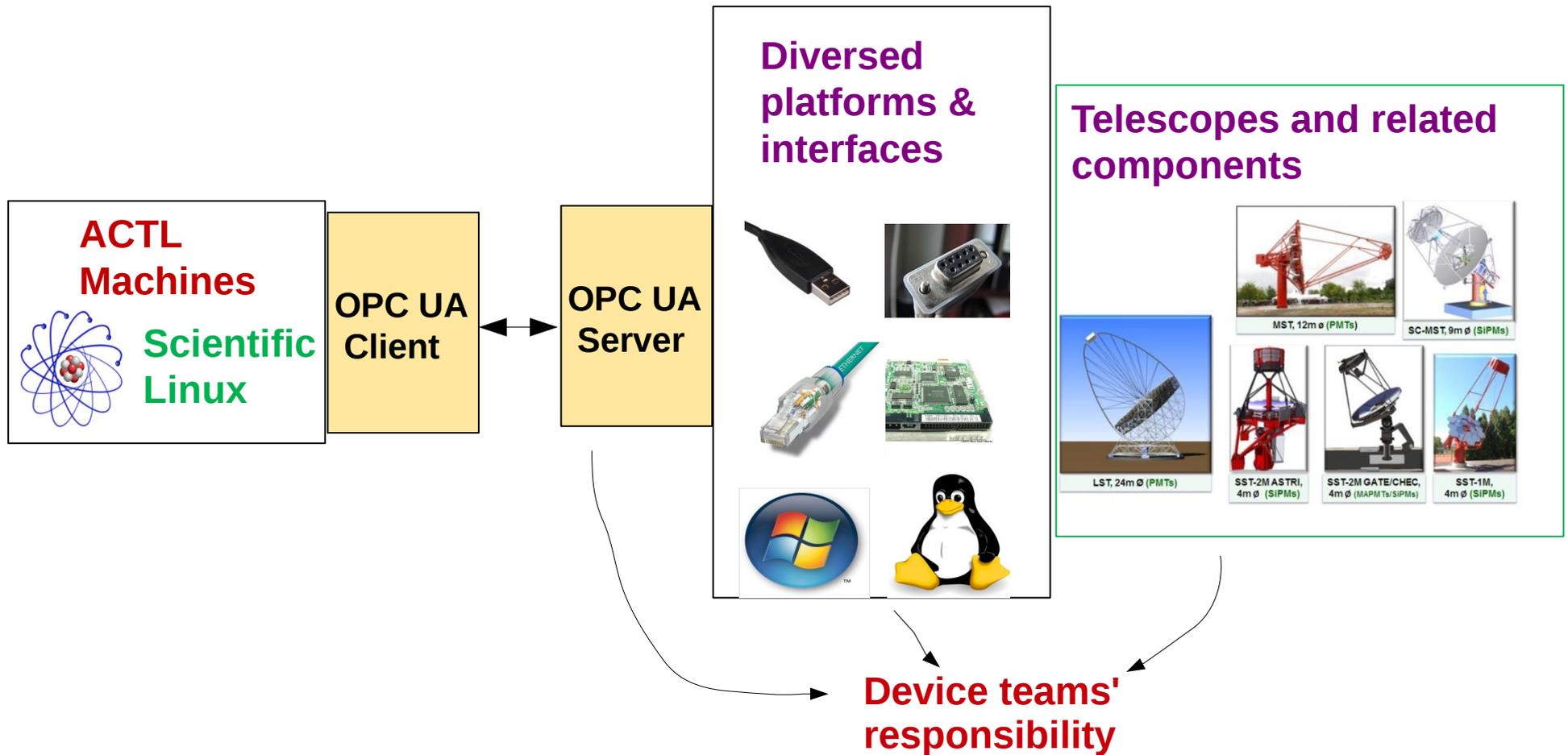
Headline benefits of OPC-UA

- **Platform Independence** : OPC UA functions on any of the following or more:
 - **Hardware platforms**: traditional PC hardware, cloud-based servers, PLCs, micro-controllers (ARM etc.)
 - **Operating Systems**: Microsoft Windows, Apple OSX, Android, or any distribution of Linux, etc.
- **High scalability**
 - from **smart sensors** and **smart actuators** to **mainframes**. The device need only to implement a **subset** of total OPC UA functionality the client need.
- **Improved Security**:
 - Public Key Infrastructure
 - User Authentication
 - User Authorization
- **Improved Modeling**
 - Extensive vocabulary for device modeling
 - ❖ Type information (allowing clients to establish semantic information)
 - ❖ Express relationship between components (allowing clients easier browsing thro' related components)
- **Enterprise Level Data Publishing**
 - Standard SOAP web service (can get through firewall) with high level of security. Even non OPC-UA clients can consume output published by an OPC-UA server.

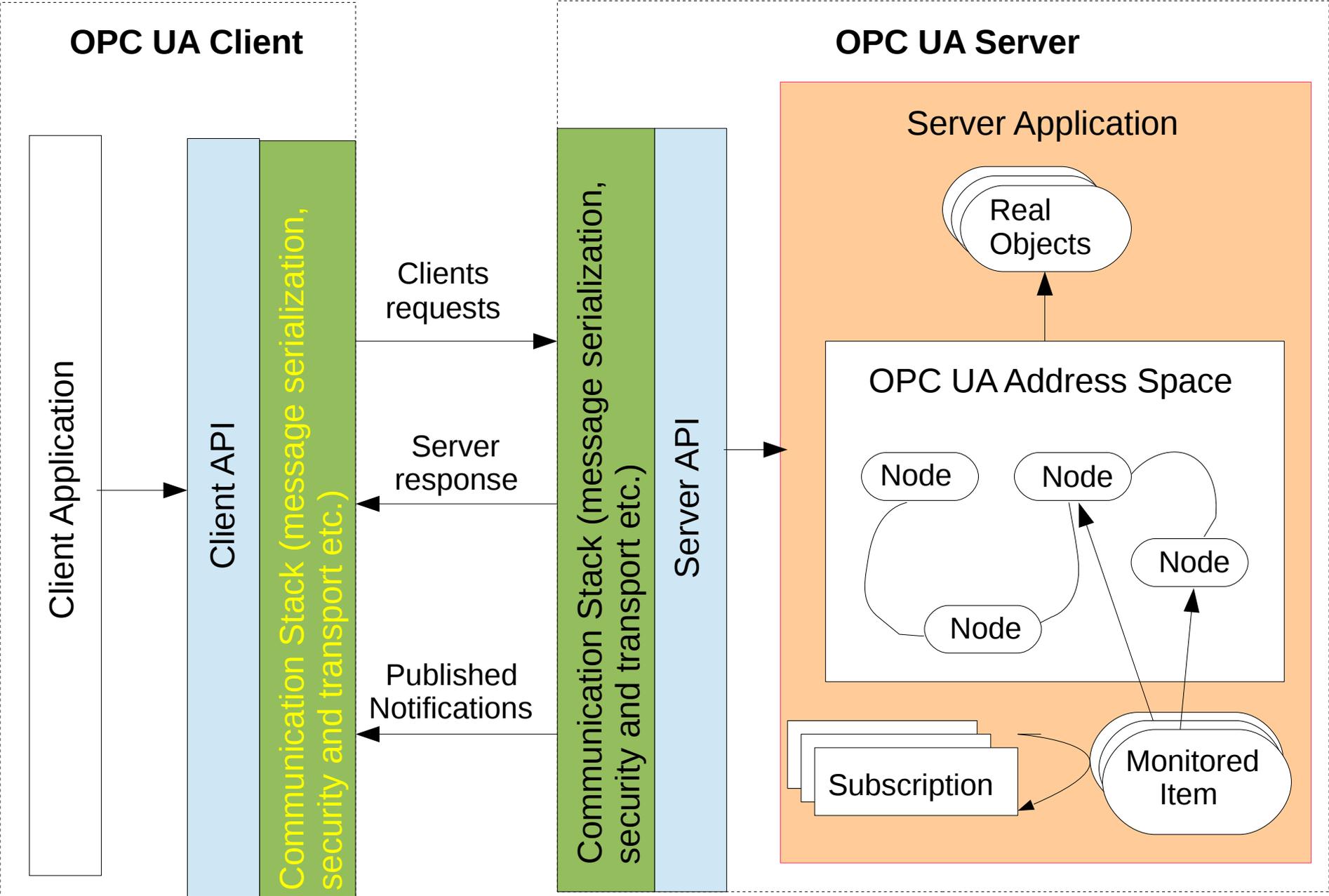
Connection between Hardware systems and Array control (ACTL) software

The CTA a technological 'zoo':

- 6 telescope structures
- 3 optical configurations
- 7 cameras
- Diverse front-end & read-out electronics



OPC Unified Architecture



Important stages in server development

- Creating Address Space
- Connecting Nodes with Real Time data
- Adding support for Methods
- Adding support for Subscription

Resources

- › C++ based OPC UA Client/Server SDK binary (from Unified Automation, Germany) and the license to use it.
- › The device driver
- › General purpose ready-to-use client called UaExpert (from Unified Automation, Germany)
- › LAN

Calibration Box for Large Sized Telescope (LST) in CTA

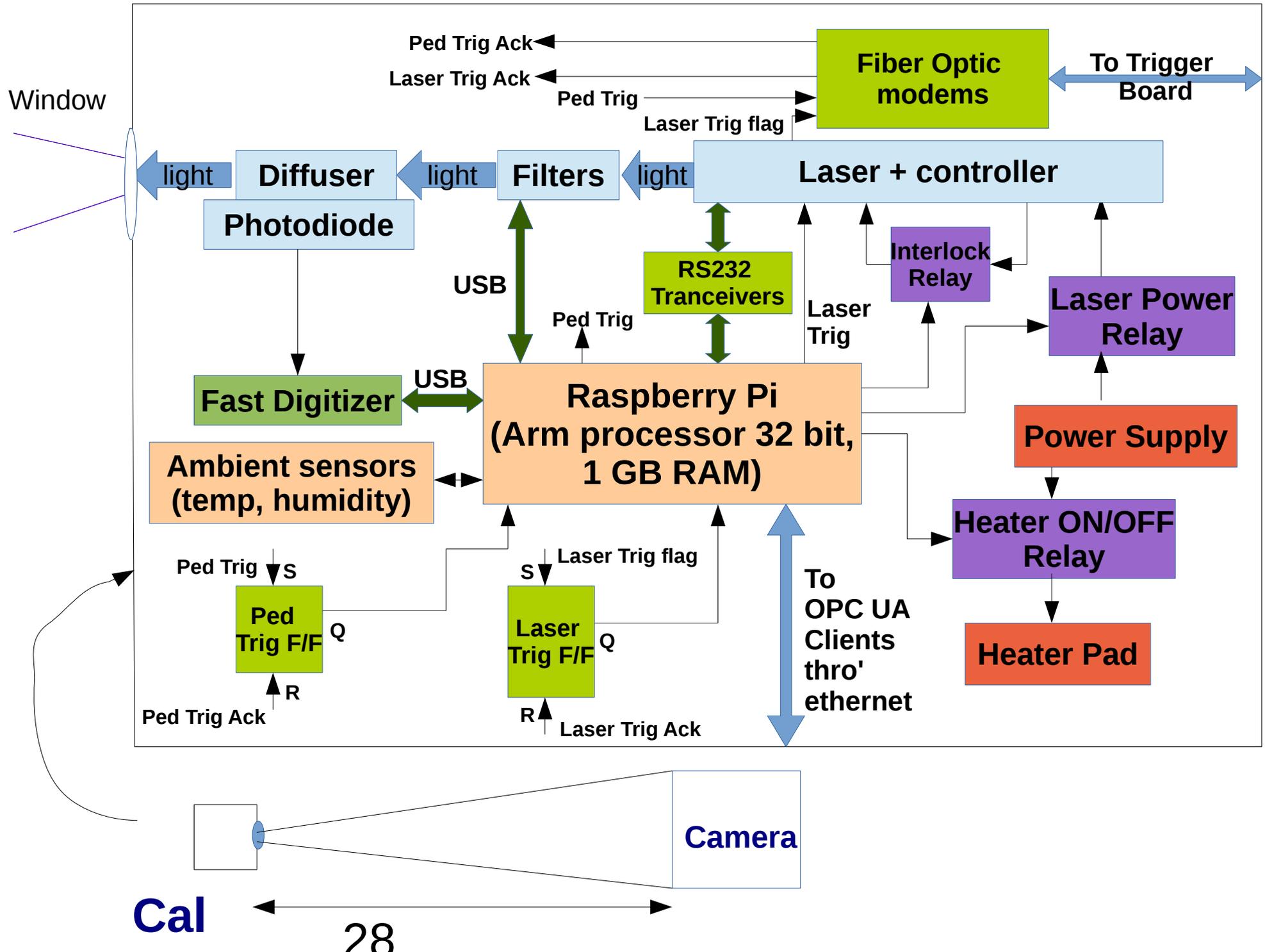
Purpose:

- A periodic calibration of each electronic channel including the PMT, processing & readout electronics to achieve precise measurements.
- This is done by illuminating the camera with a UV light source of known intensity and recording the digital counts for each electronic channel.

Some salient features of the Calibration Box

- One Calibration box for each Large Size telescope(LST).
-
- The Box is placed at a distance of ~28 m from the Camera, in the center of the parabolic mirror structure of LST.
- The laser light source produces a 400 ps pulse of UV light (355 nm), arbitrarily triggered.
- The UV laser pulse rate is adjustable between 1 Hz and 2000 Hz.
-
- The intensity of the light can be modulated by a pair of absorptive neutral density filters.
- The light beam is widespread uniformly to the whole Camera area through a diffusor inside the Calibration Box.
- Sensors for the temperature and humidity measurement inside the box
- All the components and sensors of the calibration box controlled by a single board computer, Raspberry Pi.

Functional Block diagram of Calibration Box



Salient features of the Calibration Box Server

- C++ based OPC-UA server running on the Raspberry Pi
- Device-specific protocols to communicate with the underlying hardware
- OPC-UA protocol to communicate with the software applications (clients)
- A separate thread to
 - monitor and update the status parameters like **temperature, humidity, Laser status & error**
 - Sending the **burst of laser triggers** and **Pedestal triggers** with the periodicity defined by the Clients
- The calibration box information that is exposed to clients: **Power status, Laser status, Laser error, InTemperaure, InHumidity, OutTemperature, OutHumidity, Laser Trig Parameters, Ped Trig Parameters** etc.
- Client has access to the Calibration box information by way of **polling, subscription and event generation**.
- Client writes values of certain nodes in the server address space like **filter setting, trigger parameters, temperature/humidity** outside the box
- **Methods** such as **powering On/Off, setting filters, start/stop firing the laser/pedestal triggers**

All Sky Camera

Deployed at the sight of CTA to monitor the sky condition
Connected to host PC through USB port

The device:

- Model: G2-4200
- CCD: 4 MPx CCD Kodak KAI-4022
- Connectivity with host machine : USB 2.0
- **Resolution: 2056 x 2062**
- Pixel size: 7.4 μm x 7.4 μm
- Readout modes: standard, low noise, ultra low noise
- **Image download time : 5 to 7 s**

User (client) demands:

- A set of images at some **periodic interval ...** decided at run time.
- A set would consist of at least three images each with its own **imaging parameters**
- To have access to the functional state parameters of the camera

Camera Specifications

ChipWidth
ChipHeight
PixelWidth
PixelHeight
cameraDescription

Camera State Parameters

Camera Voltage
Camera Gain
Chip Temperature
Env Temperature

Imaging Parameters

readoutMode;
hBinning;
vBinning;
exposureWidth
exposureHeight
filterNo
ExposureTime

Salient features of the All Sky Camera Server

- A thread to
 - monitor and update the **Camera State Parameters** at a fastest possible rate
 - take pictures and update the picture data as per the imaging parameters received from client
- Information exposed to clients: **Device status, Camera State Parameters, camera specs, imaging parameters[5] and picture data[5]**
- Client has access to the information above by way of **polling, subscription and event generation.**
- Client writes values of certain nodes in the address space like **imaging parameters** for a maximum of 5 pictures to be taken in a set

Current Status

- Server development projects are ON for two devices
 - › All Sky Camera
 - › Calibration Box
- First version of the Software code complete for Calibration Box
- The server for Calibration Box would be tested by integrating the underlying hardware directly.
- The software coding for 'All Sky Camera' is partially done. Once completed, it will have to be ported to the host machine located in the vicinity of the device and then will be tested remotely by logging into the machine.

HAGAR Team Members

B. S. Acharya, Varsha Chitnis, Ramdas Deshmukh, Phunchok Dorjey, Nawang Dorji, Sandeep Duhan, Anthony D'Souza, Kiran Gothe, Manoranjan, Nagesh B. K., Nandkishore Parmar, Sonal Patel, Prasanna Purohit, Shobha K Rao, Bharat B. Singh, Atreyee Sinha, Suresha Upadhyaya

Ex-members :

Lab Saha, Amit Shukla

SINP collaborators :

Anshu Chatterjee, Pratik Majumdar

Thanks!