

Status of Tragaldabas Slow Control

Marcos Seco

Universidade de Santiago de Compostela

June 29, 2017



1 Status

- Monitoring
- Slow Control

2 New Slow Control Software

- General Architecture
- DAQ+Sensors
- DB

Monitoring

- Probes: Currents, Humidities, Pressures, “Trigger Rate”
- Portal: Planned for mid May/June 2016
- Proposed: sensor matrix
 - Custom solution
 - NagVis

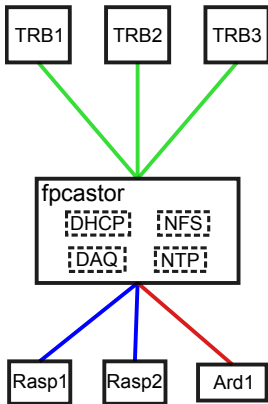
Monitoring

- Probes: Currents, Humidities, Pressures, “Trigger Rate”
- **Portal:** Iker Gonzalez is working in a version based on OpenMCT
- **Proposed:** sensor matrix
 - Custom solution
 - NagVis

Monitoring

- Probes: Currents, Humidities, Pressures, “Trigger Rate”
- **Portal:** Iker Gonzalez is working in a version based on OpenMCT
- **Proposed:** sensor matrix
 - Custom solution
 - NagVis
 - Pending on the work on OpenMCT

Slow Control

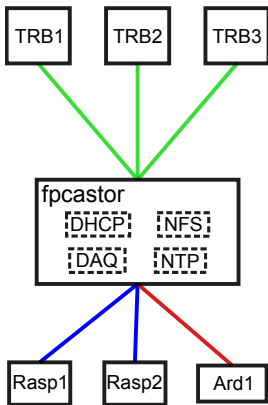


- USB (Plane Curr., temps,...)
- ETH (DHCP, NTP)
- ETH (DHCP, NFS, DAQ)

• Dead Times

- DAQ loses synchronization at times
- Parallel restart of the TRB's (~70s)
- Test to check DAQ sync status

Slow Control



- USB (Plane Currs., temps,...)
- ETH (DHCP, NTP)
- ETH (DHCP, NFS, DAQ)

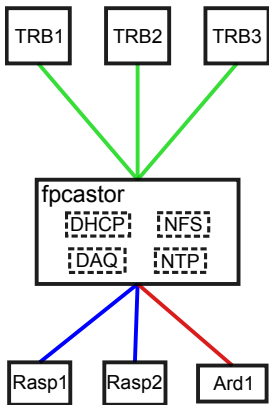
• Dead Times

- DAQ loses synchronization at times
- Parallel restart of the TRB's (~70s)
- Test to check DAQ sync status
- Hardware deadtime of 1ms.

Slow Control

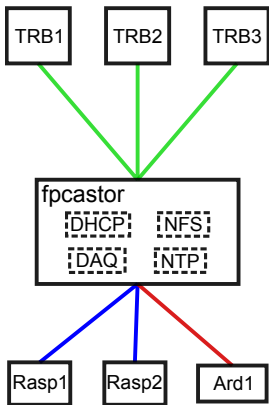
- Dead Times

- Test to check DAQ sync status: **Implemented**



- USB (Plane Curr., temps,...)
- ETH (DHCP, NTP)
- ETH (DHCP, NFS, DAQ)

Slow Control

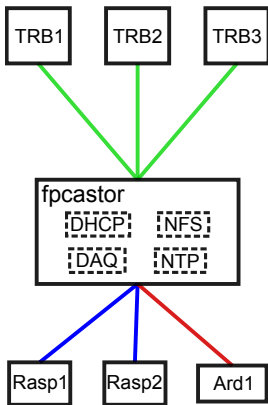


- USB (Plane Curr., temps,...)
- ETH (DHCP, NTP)
- ETH (DHCP, NFS, DAQ)

• Dead Times

- Test to check DAQ sync status: **Implemented**
- The system is now restarted on **sync lost**

Slow Control

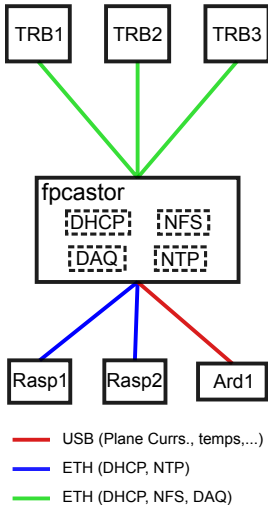


- USB (Plane Curr., temps,...)
- ETH (DHCP, NTP)
- ETH (DHCP, NFS, DAQ)

• Dead Times

- Test to check DAQ sync status: **Implemented**
- The system is now restarted on **sync lost**
- Reduced hardware deadtime **100 μ s.**

Slow Control



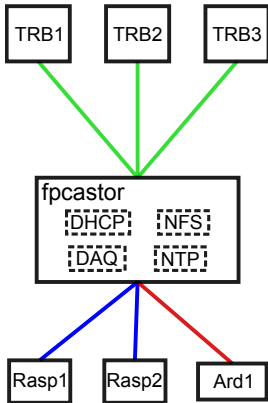
- Dead Times

- Test to check DAQ sync status: **Implemented**
- The system is now restarted on **sync lost**
- Reduced hardware deadtime **100 μ s.**

- DAQ software chain

- mix of bash and perl scripts.
- Plan to migrate everything to python.

Slow Control



- USB (Plane Curr., temps,...)
- ETH (DHCP, NTP)
- ETH (DHCP, NFS, DAQ)

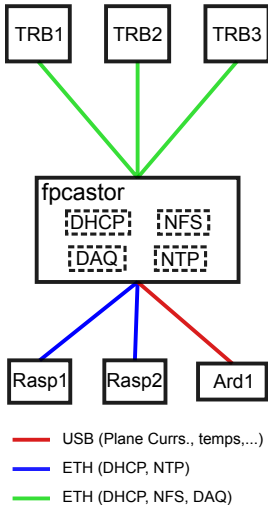
- Dead Times

- Test to check DAQ sync status: **Implemented**
- The system is now restarted on **sync lost**
- Reduced hardware deadtime **100 μ s.**

- DAQ software chain

- Migration started.

Slow Control



• Dead Times

- Test to check DAQ sync status: **Implemented**
- The system is now restarted on **sync lost**
- Reduced hardware deadtime **100 μ s.**

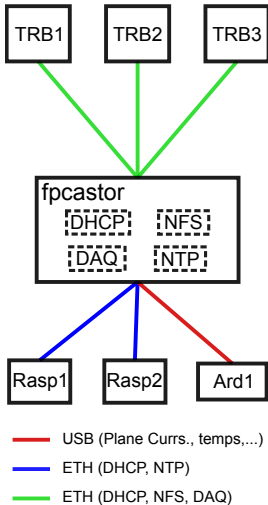
• DAQ software chain

- Migration started.

• Availability

- Only one host for NFS, DHCP, DAQ \implies SPOF
- Possible solutions for NFS:
 - DRBD+Heartbeat
 - CVMFS

Slow Control



• Dead Times

- Test to check DAQ sync status: **Implemented**
- The system is now restarted on **sync lost**
- Reduced hardware deadtime **100 μ s**.

• DAQ software chain

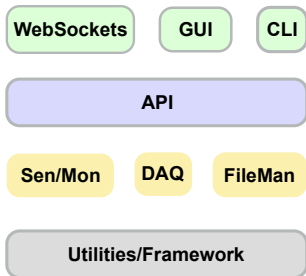
- Migration started.

• Availability

- Only one host for NFS, DHCP, DAQ \implies SPOF
- Possible solutions for NFS:
 - DRBD+Heartbeat
 - CVMFS
- Use of Raspberry Pi's as DAQs

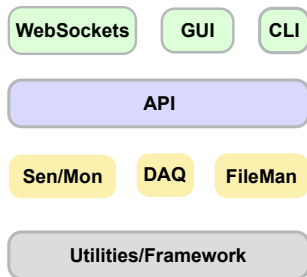
General Architecture

- General Principles:
 - I **do not** want to **reinvent the wheel**.



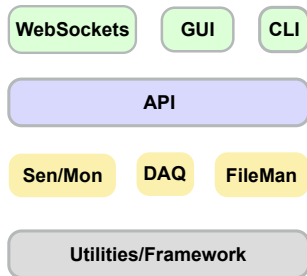
General Architecture

- General Principles:
 - I **do not** want to **reinvent the wheel**.
 - System must be **modular**



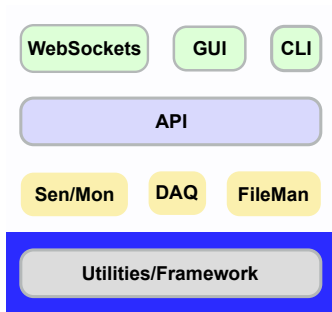
General Architecture

- General Principles:
 - I **do not** want to **reinvent the wheel**.
 - System must be **modular**
 - System must be relatively **simple**



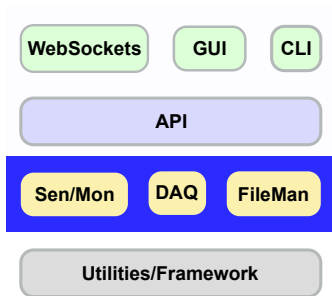
General Architecture

- General Principles:
 - I **do not** want to **reinvent the wheel**.
 - System must be **modular**
 - System must be relatively **simple**
- Layers:
 - Utilities/Framework: logs, comms., messages, DB's ...



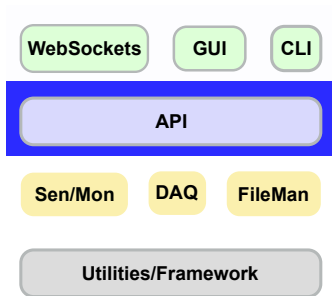
General Architecture

- General Principles:
 - I **do not** want to **reinvent the wheel**.
 - System must be **modular**
 - System must be relatively **simple**
- Layers:
 - Utilities/Framework: logs, comms., messages, DB's ...
 - Work: Perform all the **necessary actions** to acquire the data, monitorize the system or keep track of generated files.



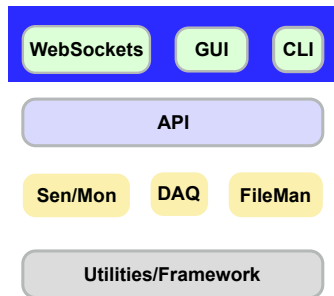
General Architecture

- General Principles:
 - I **do not** want to **reinvent the wheel**.
 - System must be **modular**
 - System must be relatively **simple**
- Layers:
 - Utilities/Framework: logs, comms., messages, DB's ...
 - Work: Perform all the **necessary actions** to acquire the data, monitorize the system or keep track of generated files.
 - API: middle layer to **generate interfaces**.



General Architecture

- General Principles:
 - I **do not** want to **reinvent the wheel**.
 - System must be **modular**
 - System must be relatively **simple**
- Layers:
 - Utilities/Framework: logs, comms., messages, DB's ...
 - Work: Perform all the **necessary actions** to acquire the data, monitorize the system or keep track of generated files.
 - API: middle layer to **generate interfaces**.
 - Service: Interfaces to **communicate** with the outside world



DAQ+Sensors

