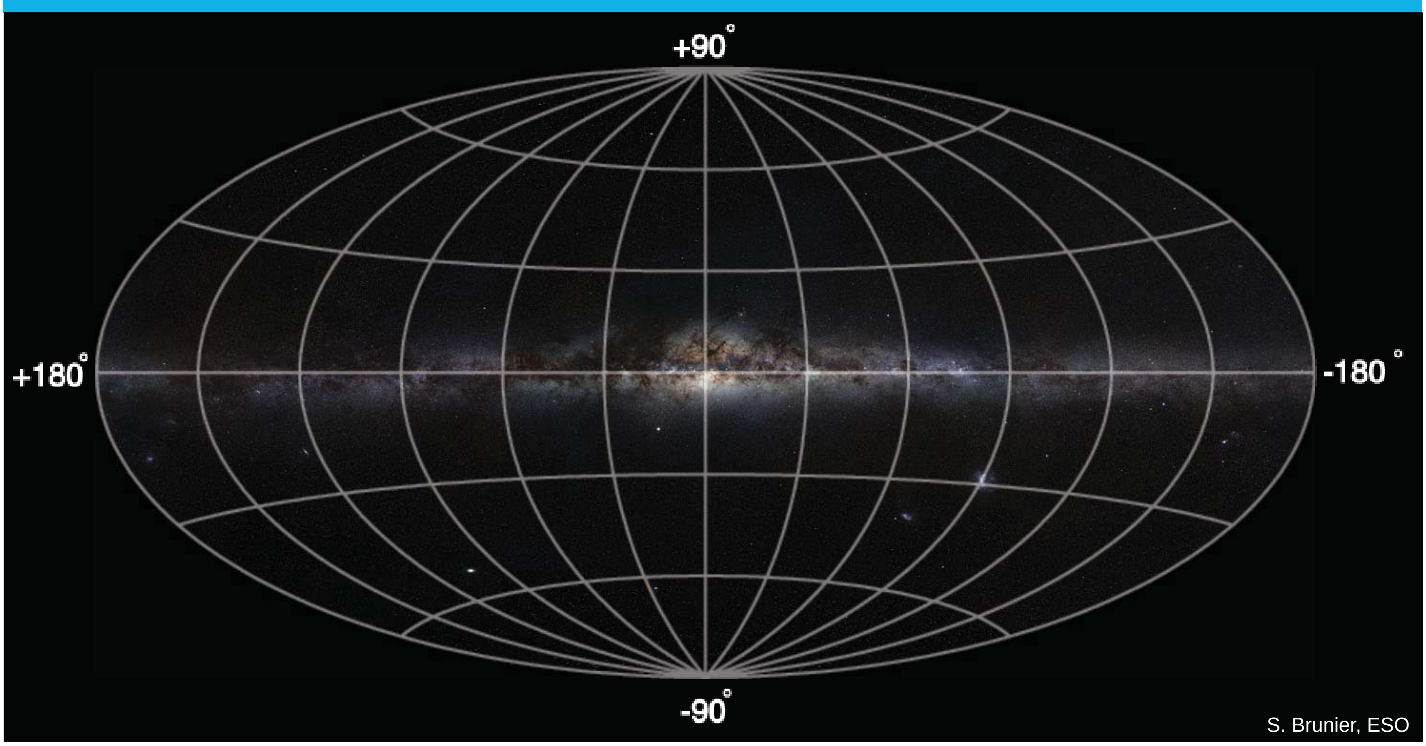
The H.E.S.S. Data Acquisition System.

An overview

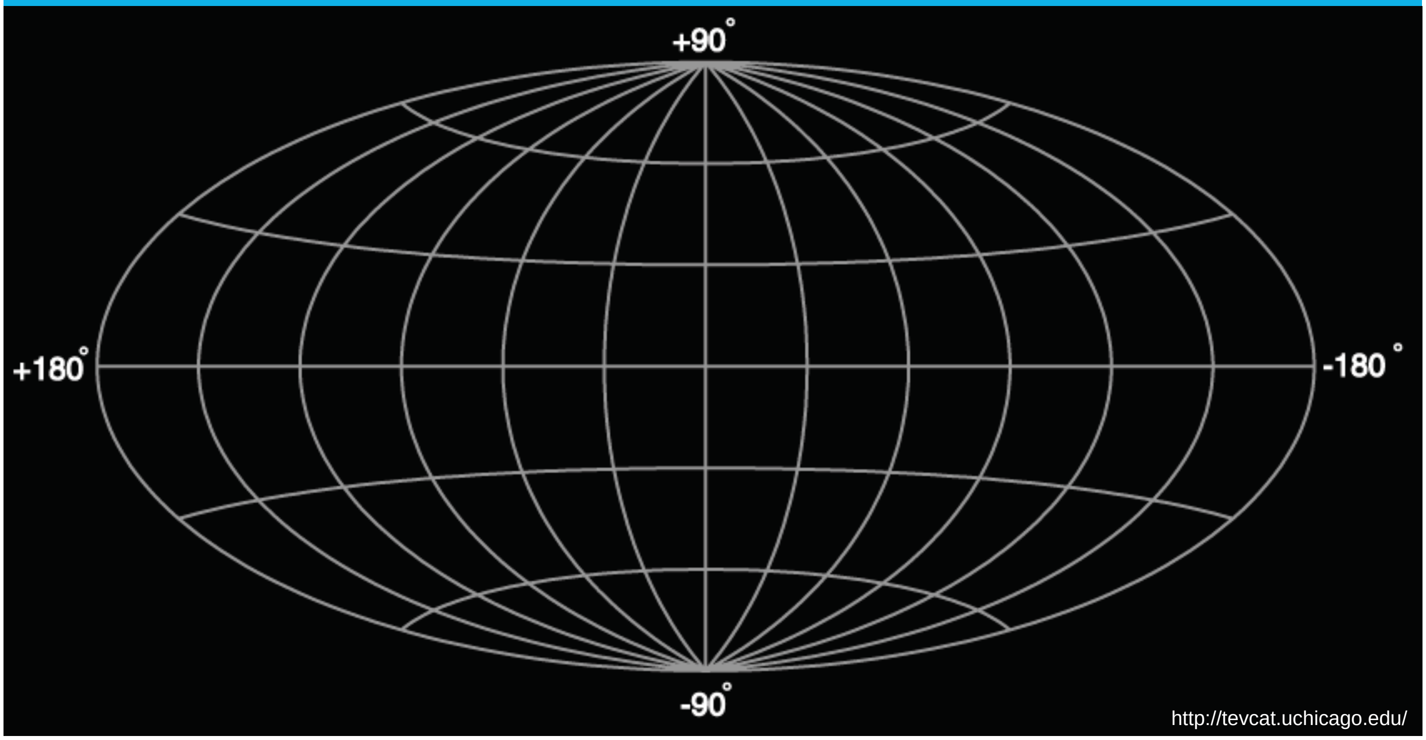
Arnim Balzer
for the H.E.S.S. DAQ group
Technical Seminar
Zeuthen, 26.06.2012

The Optical Sky



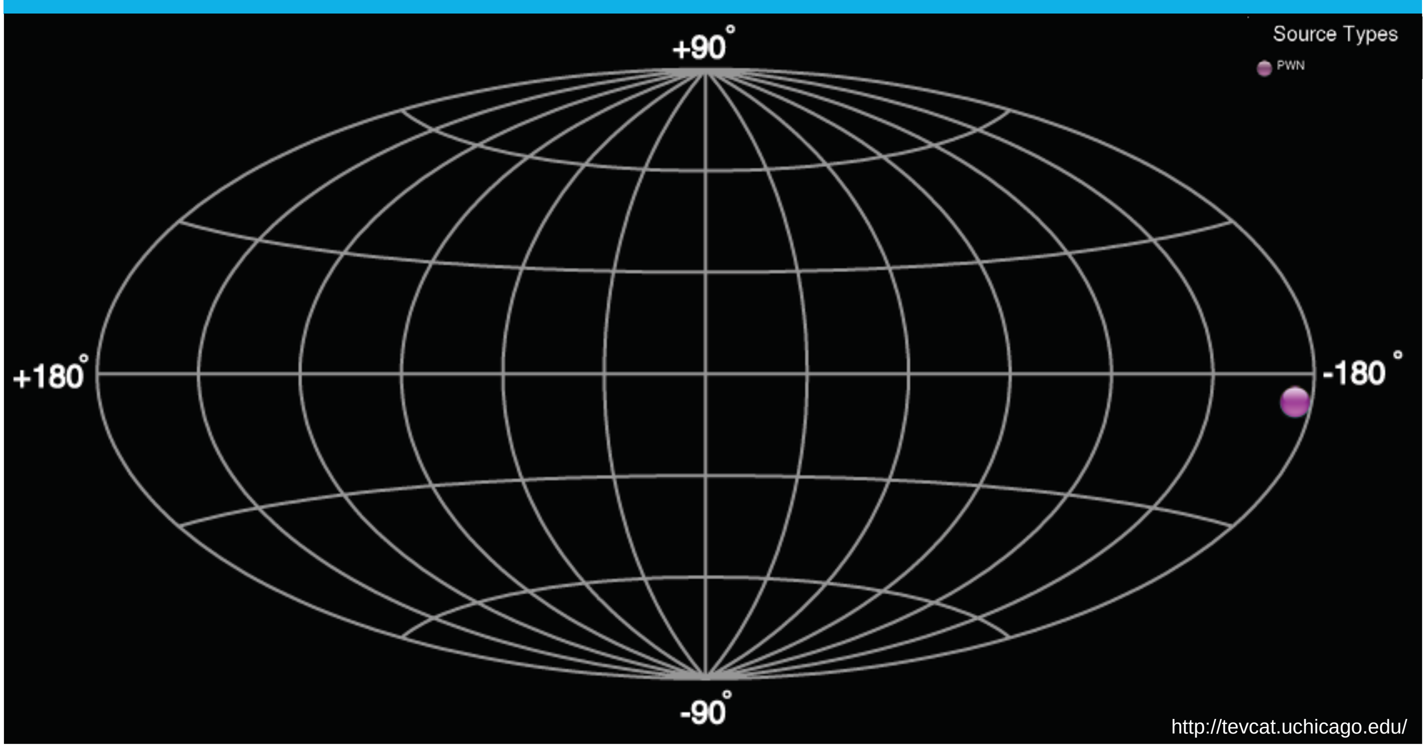
> Energy of radiation: 1 eV

The Very High Energy Gamma-Ray Sky before 1989



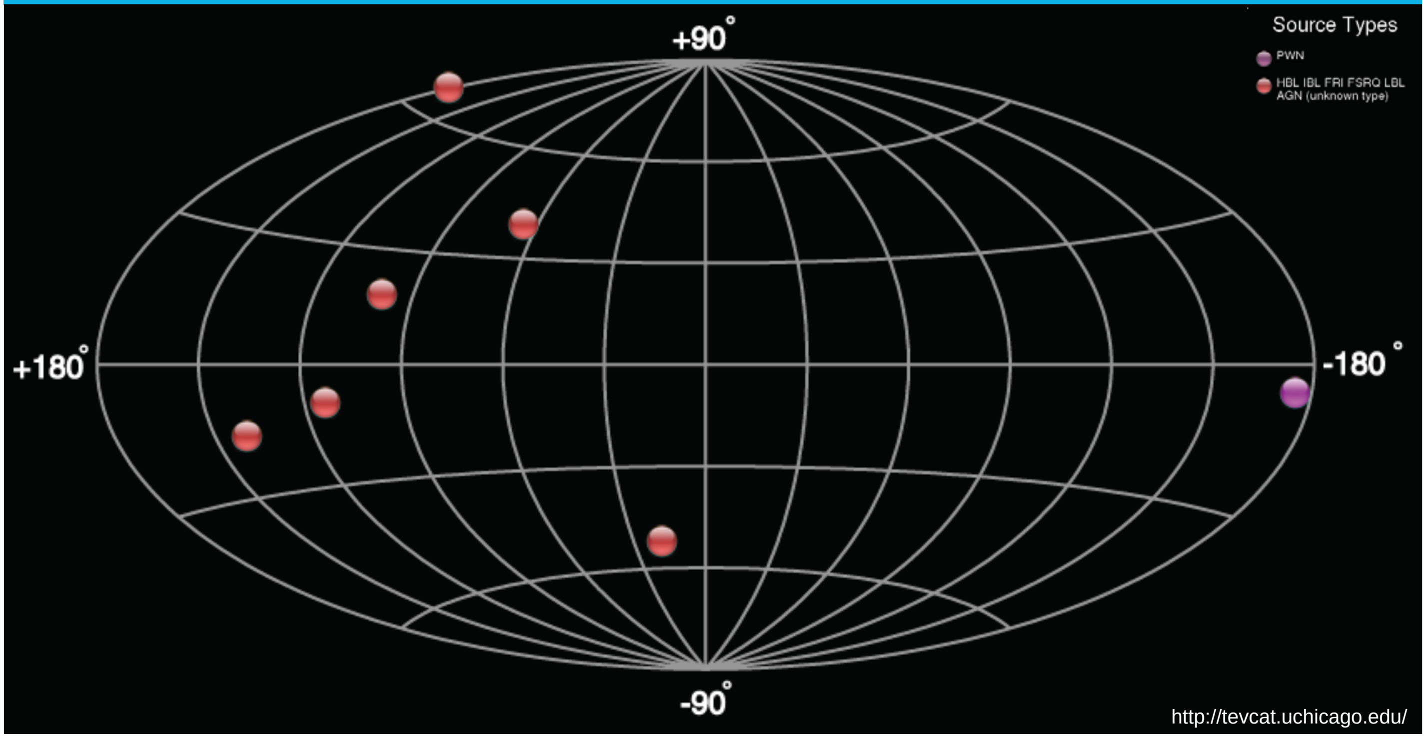
- Energy of radiation: >100 GeV

The Very High Energy Gamma-Ray Sky in 1989



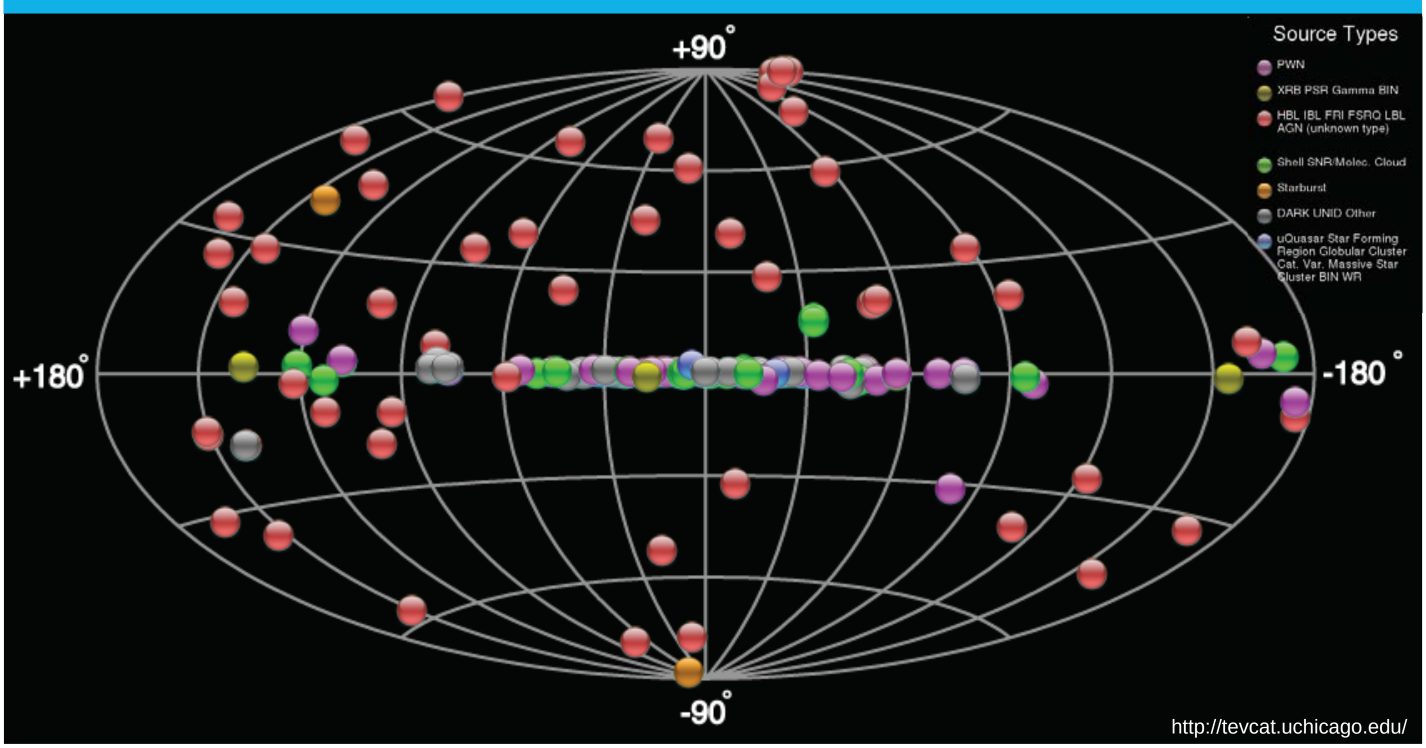
- Energy of radiation: >100 GeV

The Very High Energy Gamma-Ray Sky in 1999



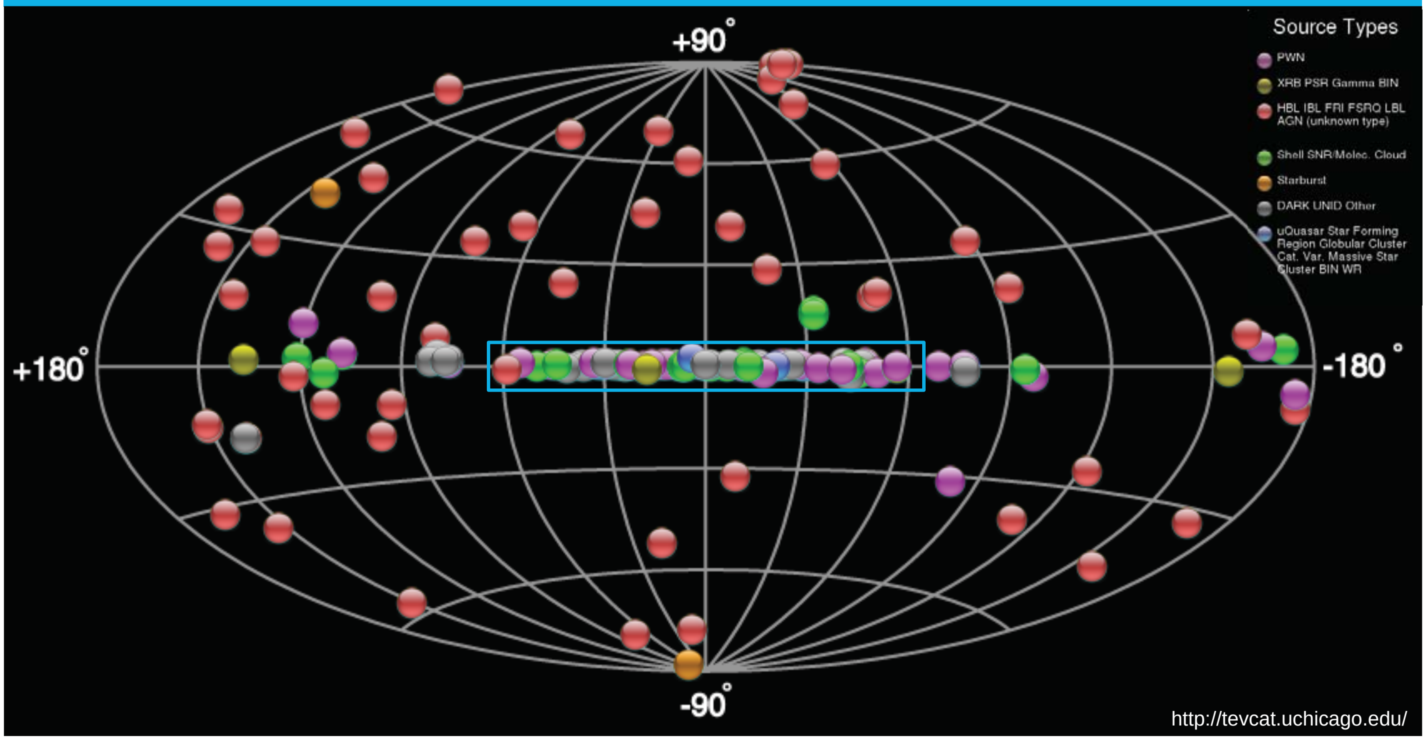
> Energy of radiation: >100 GeV

The Very High Energy Gamma-Ray Sky in 2012

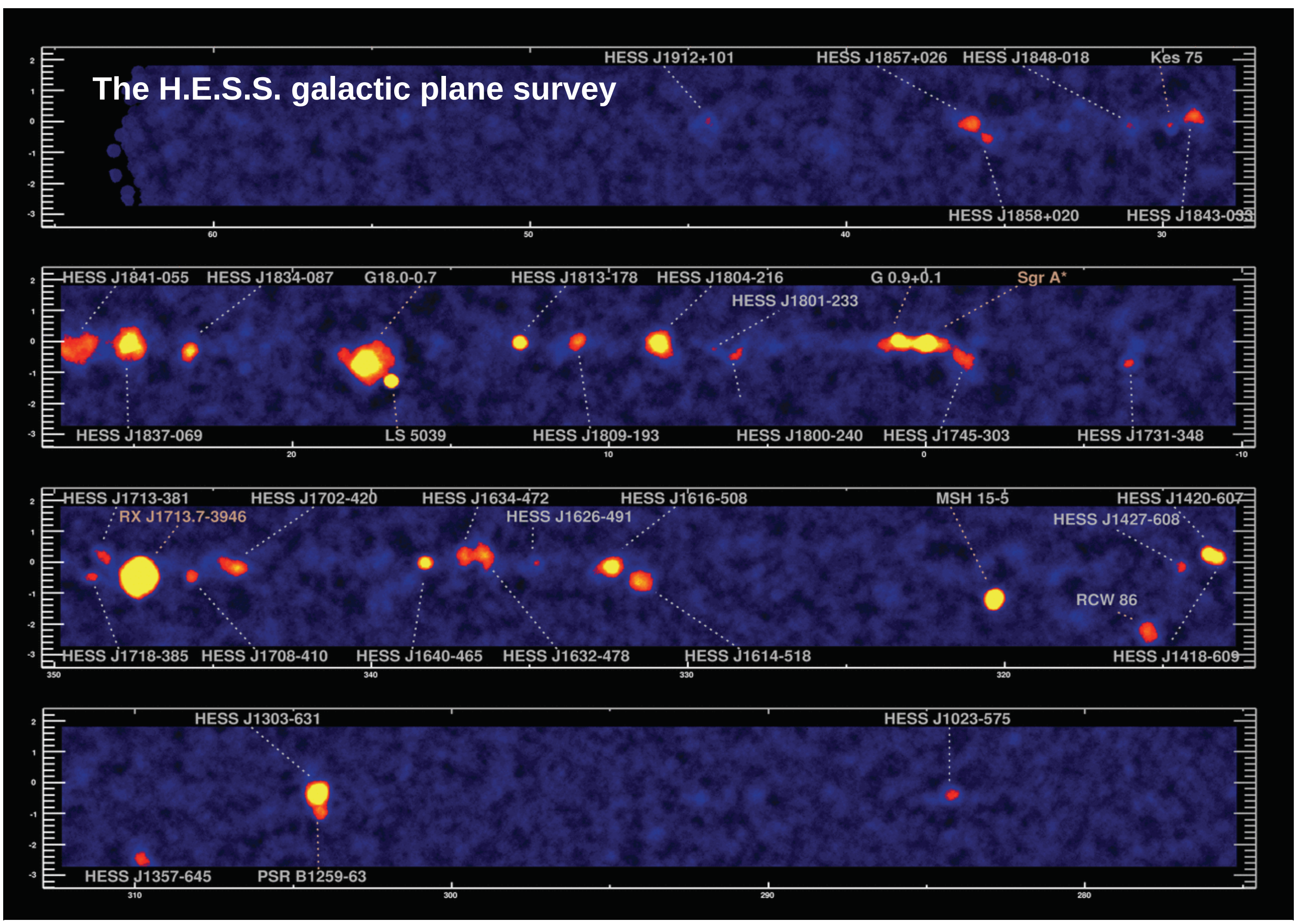


> Energy of radiation: >100 GeV

The Very High Energy Gamma-Ray Sky in 2012

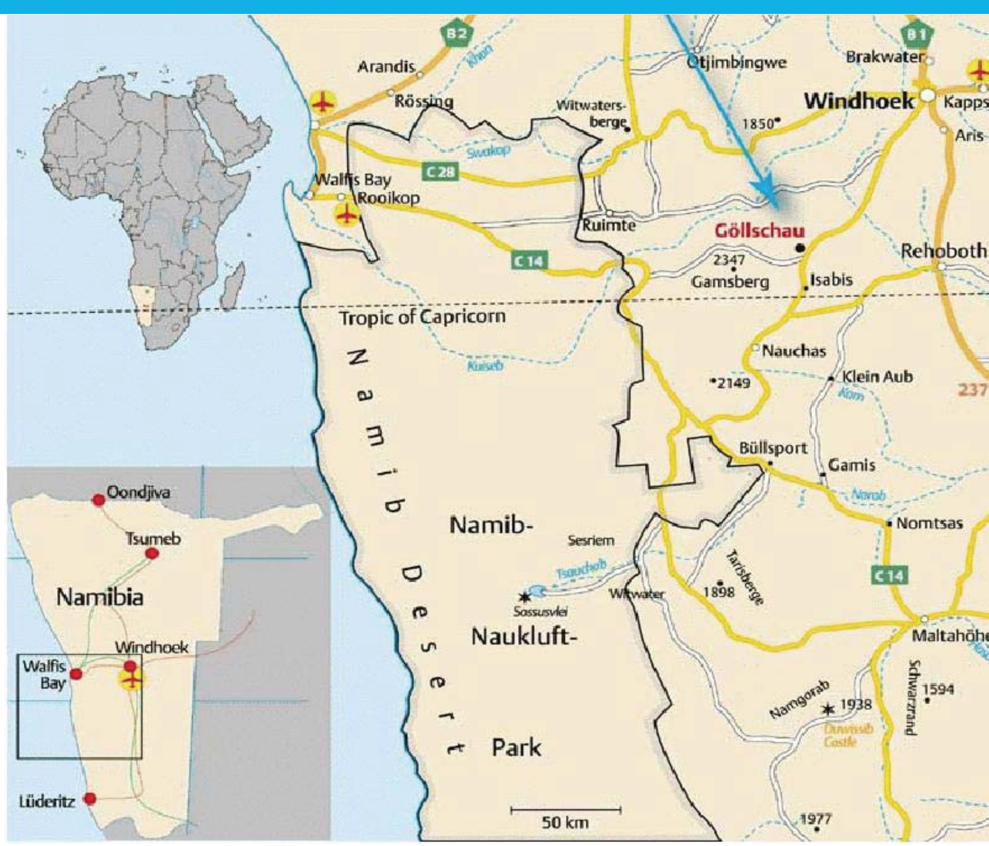


> Energy of radiation: >100 GeV

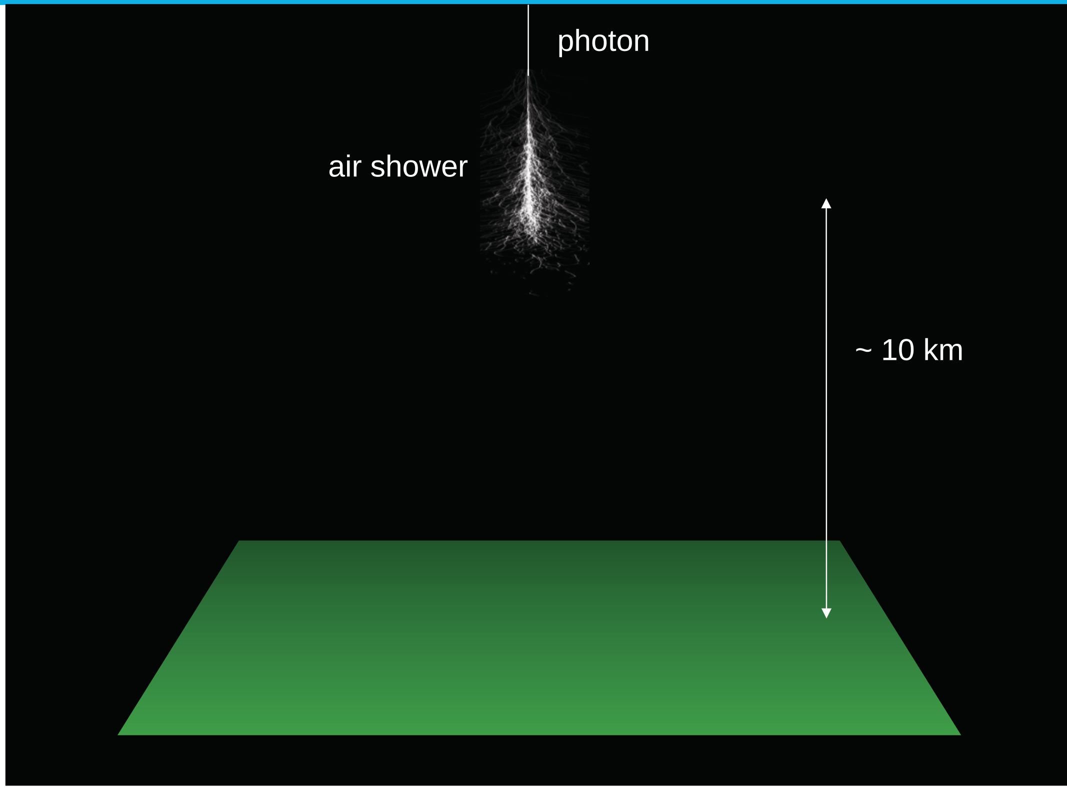


The High Energy Stereoscopic System

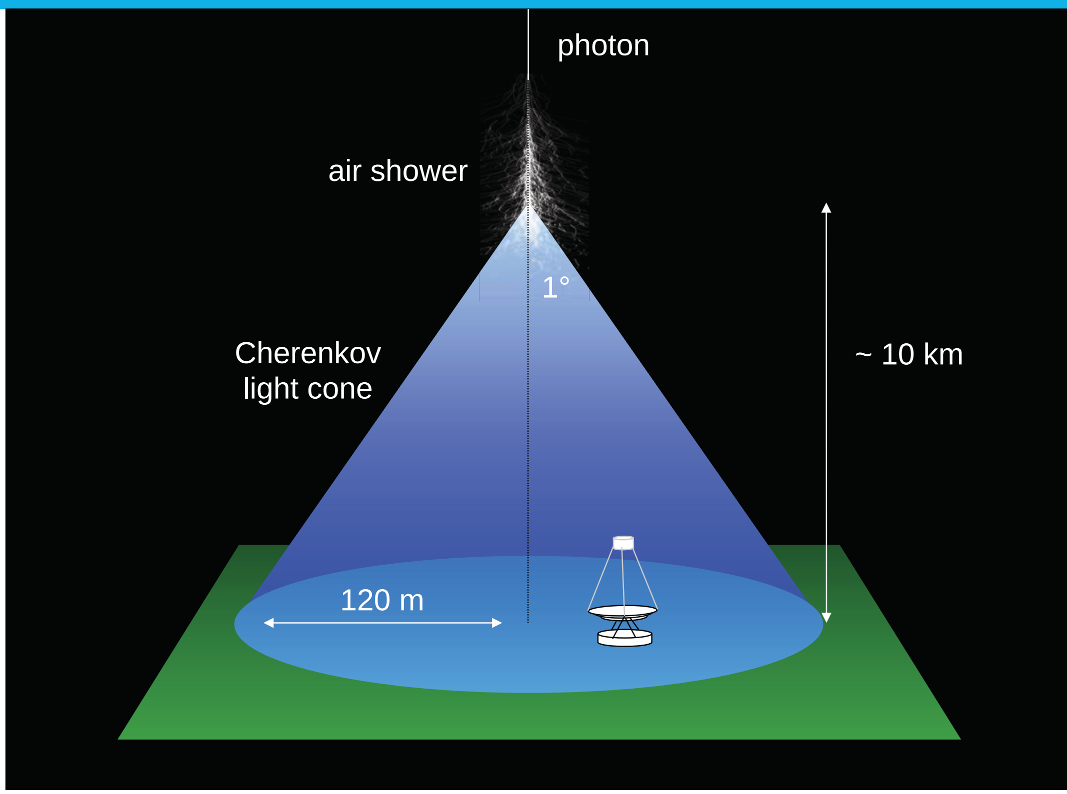
Namibia



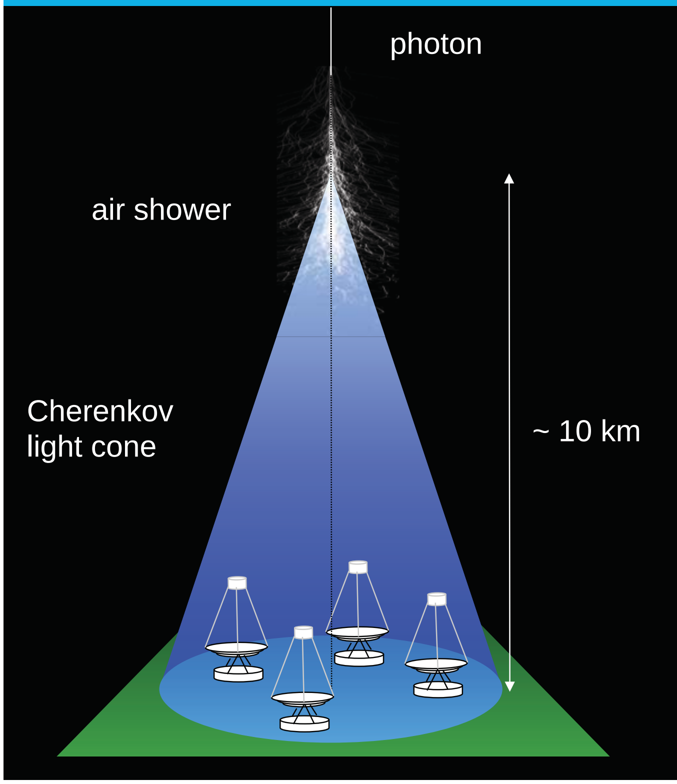
Imaging Atmospheric Cherenkov Technique

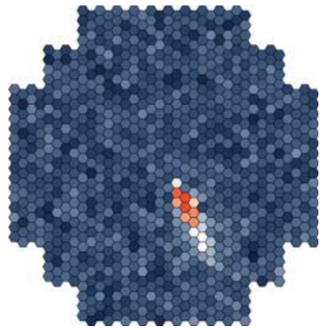


Imaging Atmospheric Cherenkov Technique

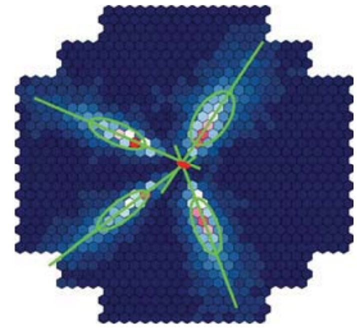


Imaging Atmospheric Cherenkov Technique





Single telescope event



Four telescope event in common camera plane

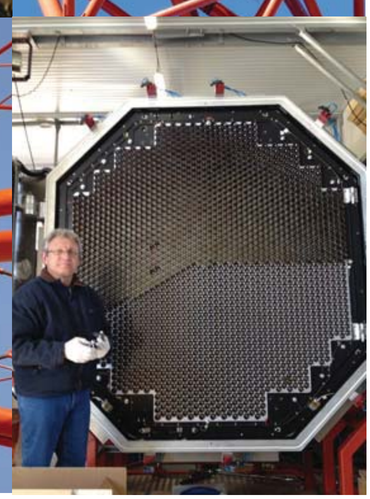
- > Intensity → Energy
- > Orientation → Direction
- > Shape → Primary Particle

A 960 Pixel vs. a 2048 Pixel Digital Camera

H.E.S.S. Phase 1

H.E.S.S. II

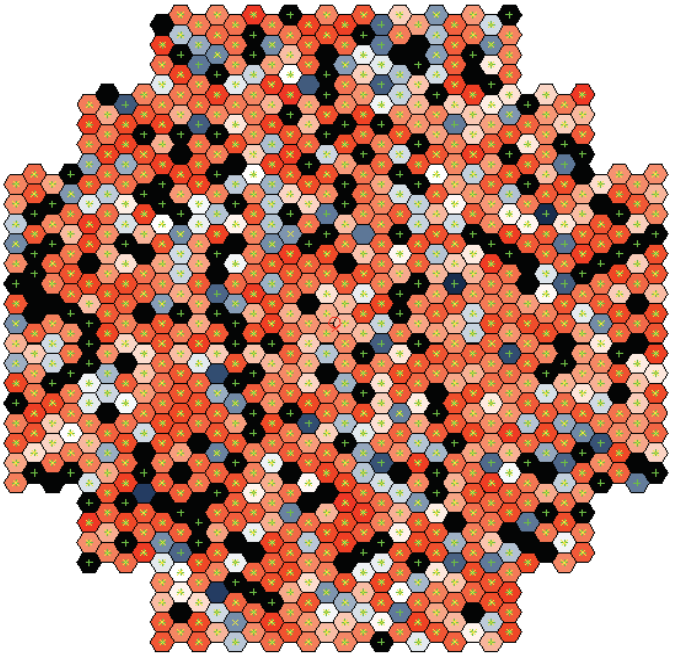
H.E.S.S. Phase 1



H.E.S.S. II

Most Important Camera Feature

> Very short “exposure time”



1/10000 s
(100 µs)

1/100000 s
(10 µs)

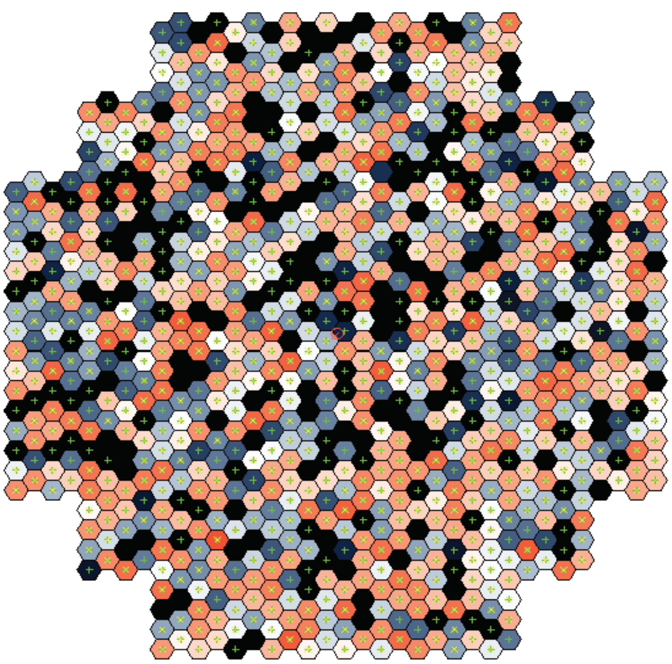
1/1000000 s
(1 µs)

1/10000000 s
(100 ns)

16/100000000 s
(16 ns)

Most Important Camera Feature

> Very short “exposure time”



1/10000 s
(100 µs)

1/100000 s
(10 µs)

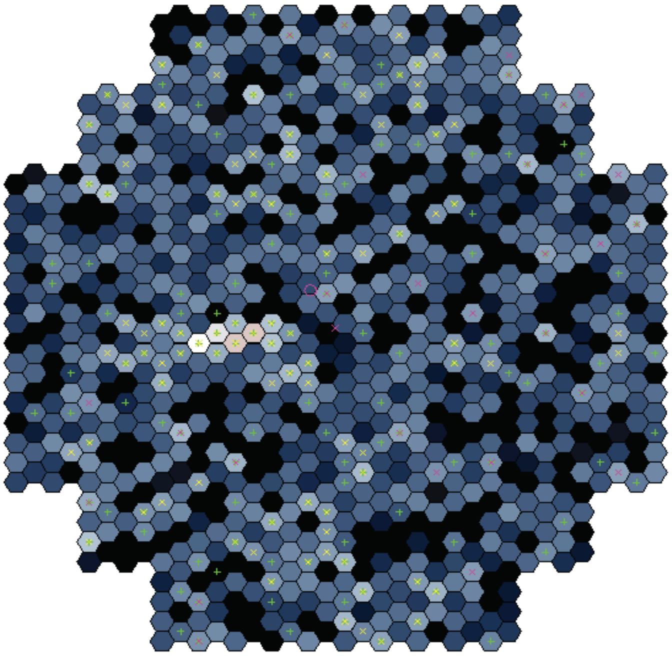
1/1000000 s
(1 µs)

1/10000000 s
(100 ns)

16/100000000 s
(16 ns)

Most Important Camera Feature

> Very short “exposure time”



1/10000 s
(100 μs)

1/100000 s
(10 μs)

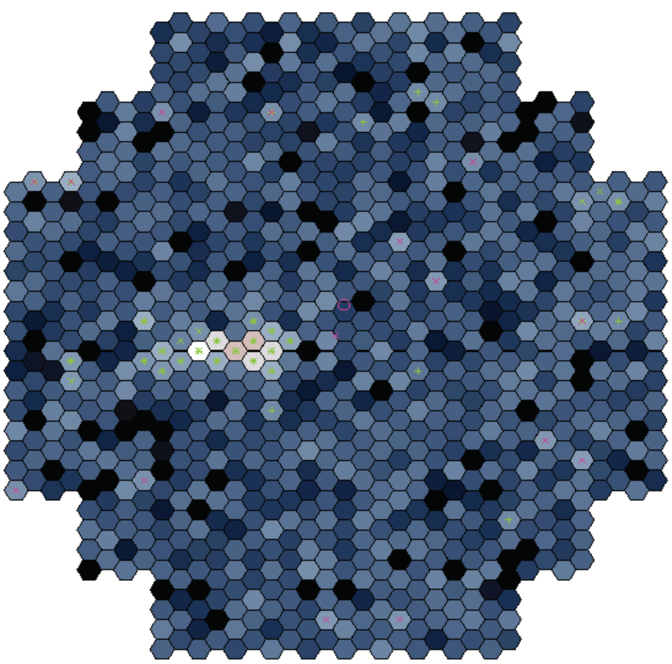
1/1000000 s
(1 μs)

1/10000000 s
(100 ns)

16/100000000 s
(16 ns)

Most Important Camera Feature

> Very short “exposure time”



1/10000 s
(100 μs)

1/100000 s
(10 μs)

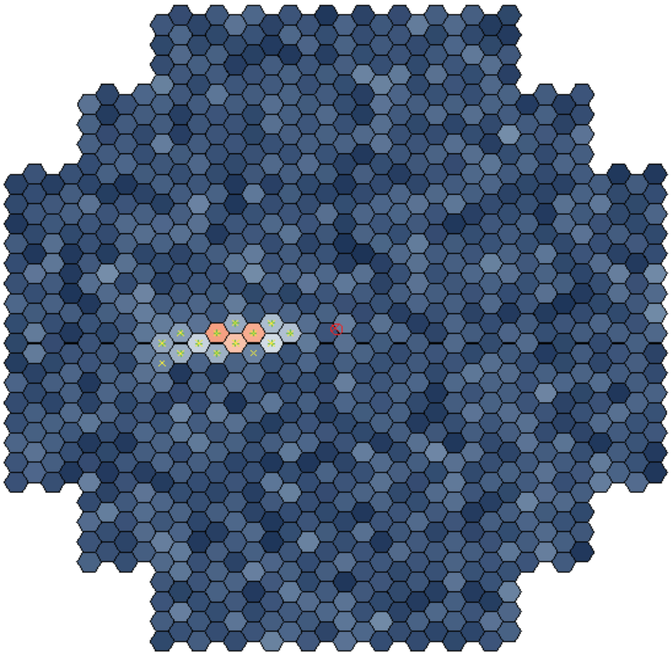
1/1000000 s
(1 μs)

1/10000000 s
(100 ns)

16/100000000 s
(16 ns)

Most Important Camera Feature

> Very short "exposure time"



1/10000 s
(100 μs)

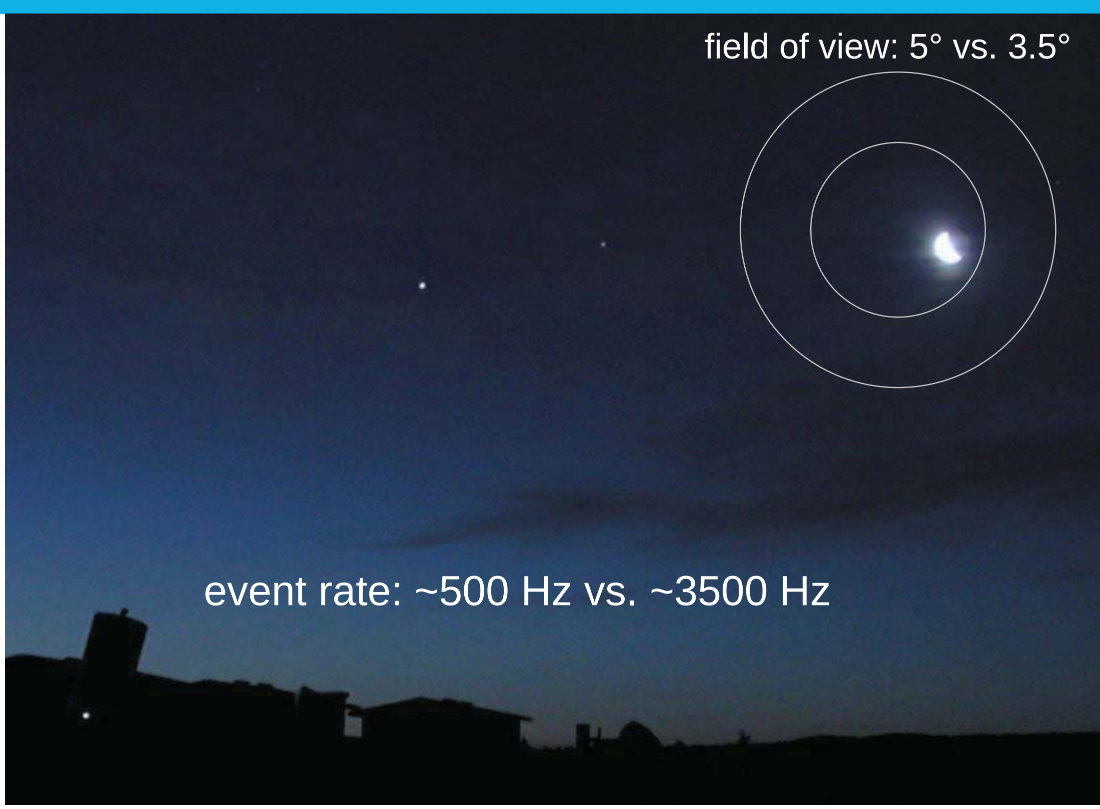
1/100000 s
(10 μs)

1/1000000 s
(1 μs)

1/10000000 s
(100 ns)

16/100000000 s
(16 ns)

The Field of View



Overview

> Members of the DAQ group
> DAQ requirements
> Hardware
  - Server farm & network
  - Cluster monitoring & remote administration
> Software
  - General DAQ design
  - State machine & transitions
  - Communication & messaging
  - Data flow & node switching
  - GUIs & displays
  - Monitoring array performance
  - DAQ virtual machine

Current Members of the H.E.S.S. DAQ Group

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

Daniel Göring
Christian Schneider

ÉCOLE POLYTECHNIQUE ParisTech

Mathieu de Naurois

DESY

Christian Stegmann
Matthias Füßling

HUMBOLDT-UNIVERSITÄT ZU BERLIN

Ullrich Schwanke
Michael Gajdus

Universität Potsdam

Anton Lopatin
Arnim Balzer

Requirements for the H.E.S.S. II Data Acquisition

> Be able to process the data rates of the 5 cameras:

- 500 Hz * 4 Telescopes * 960 Pixels * 2 Channels * 4 byte ≈ 15 MB/s
- 3500 Hz * 1 Telescopes * 2048 Pixels * 2 Channels * (4+2+2) byte ≈ 115 MB/s

> Enough storage for at least 2 month of data taking

- ≈ 3177 GB in 228 days with H.E.S.S. Phase 1 ≈ 420 GB per Month
- ≈ 420 GB per Month * 115 MB/s / 15 MB/s ≈ 3220 GB per Month

> Enough computing power for:

- Event building
- Real time analysis & data quality checks

Server Farm on Site

> 5 Supermicro storage server

- RAID 6 + Hot Spare + Backup
- 12 TB each
- Intel Xeon E5420 4*2.5 GHz
- 8 GB DDR2 RAM

> 10 Supermicro worker nodes

- Intel Xeon E5450 8*3.0 GHz
- 16 GB DDR2 RAM

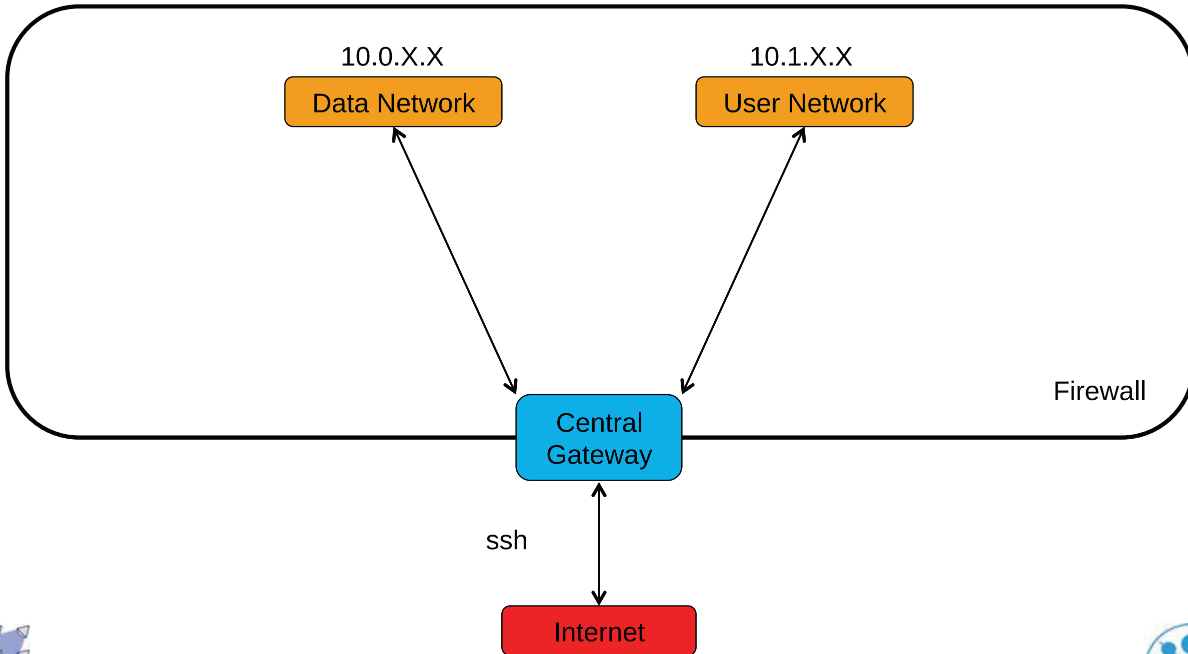
> Operating system:

- Fedora 12 x64

Network Layout

> Keep networks separated

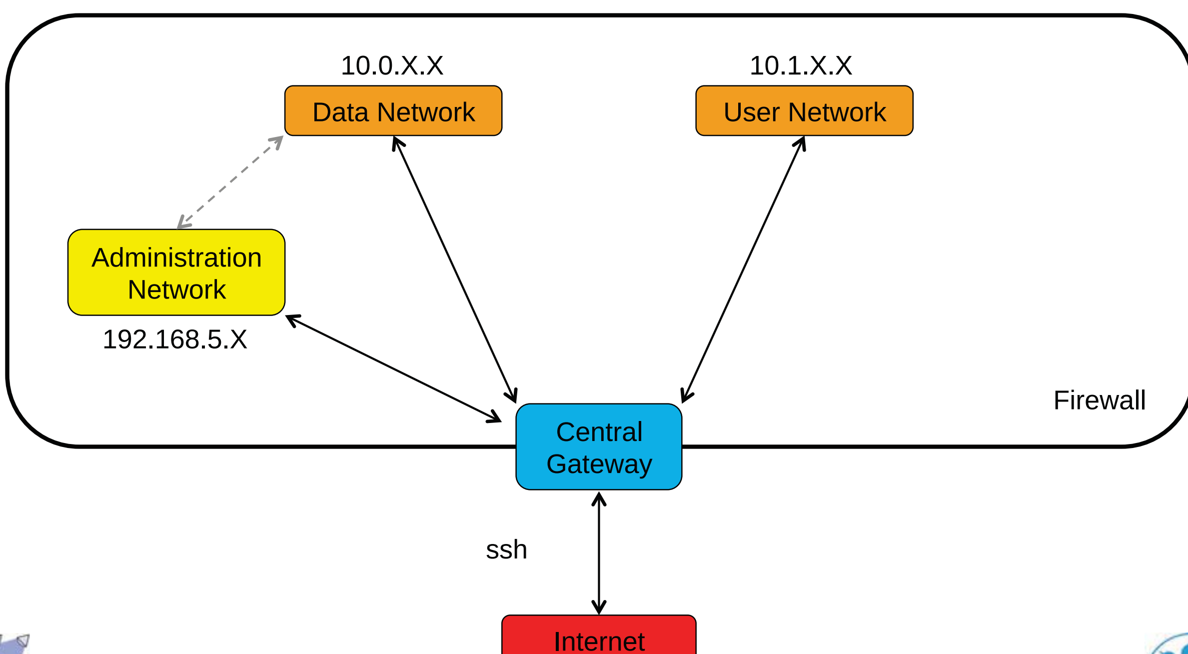
- Especially data and user network



Network Layout

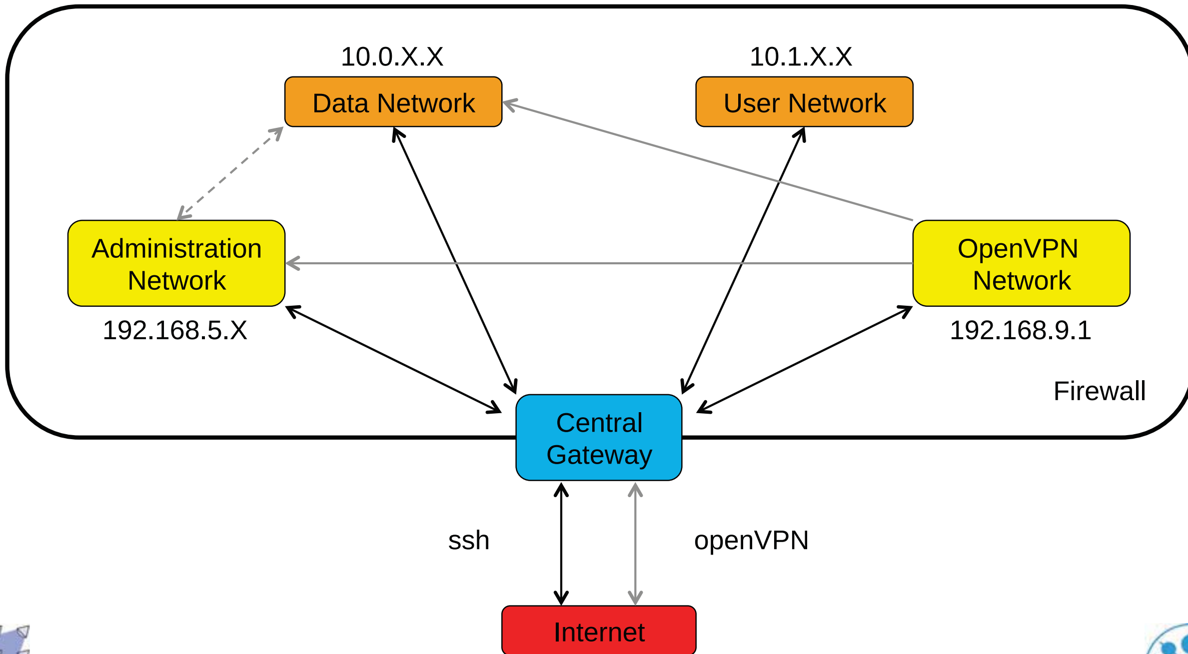
> Keep networks separated

- Especially data and user network



Network Layout

> Keep networks separated
  - Especially data and user network



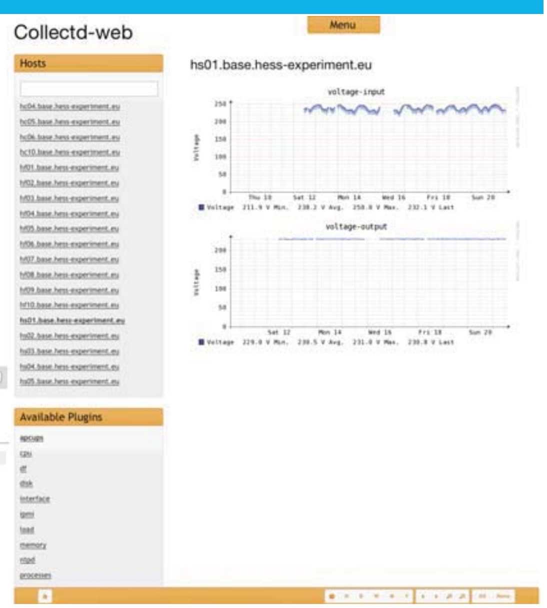
Cluster Monitoring

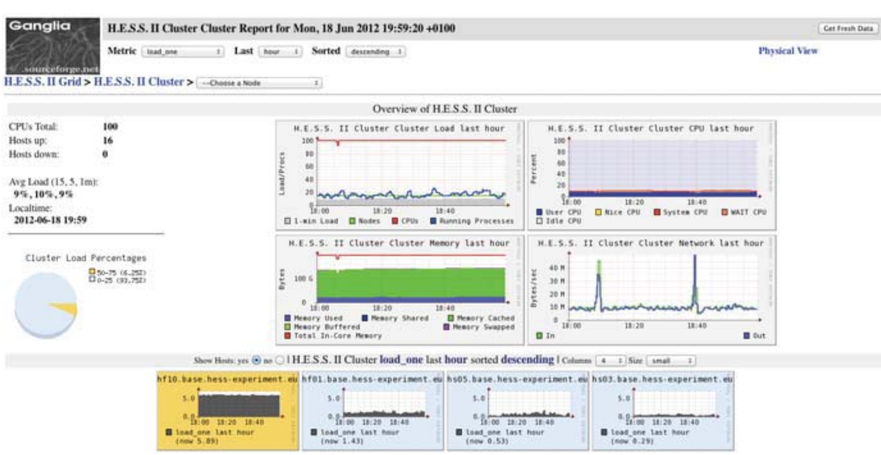
> Ganglia

> Collectd

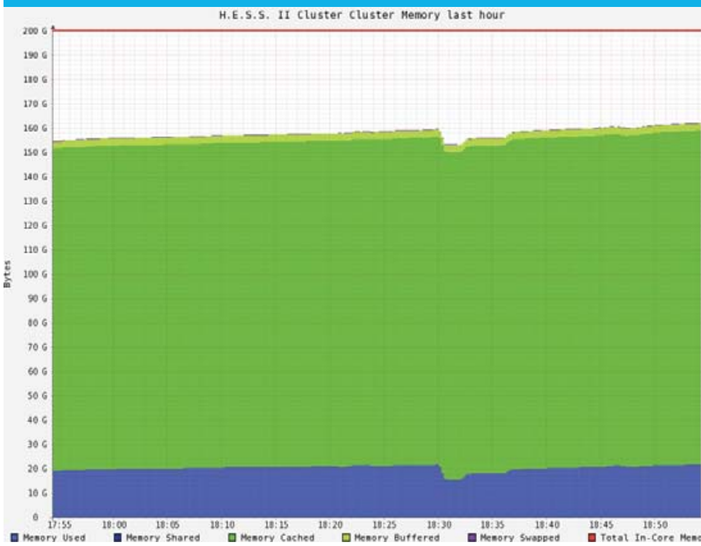
> Logwatch

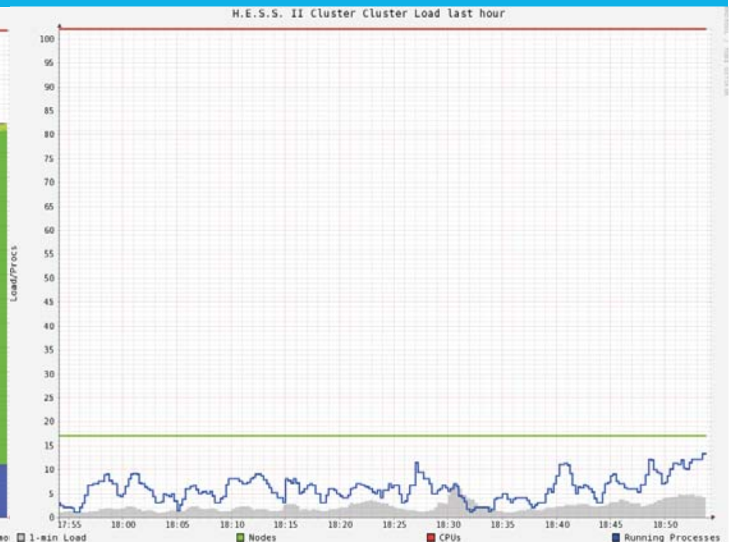
> SNMP traps and emails





Current Cluster Usage





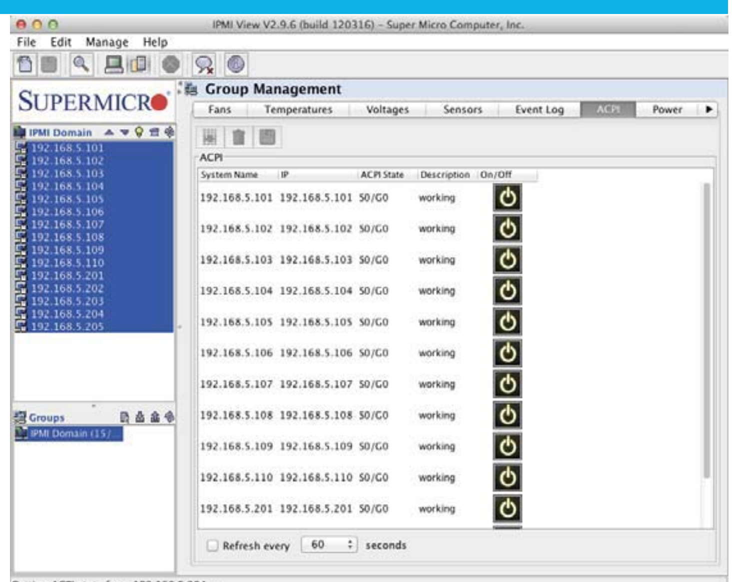
Remote Administration

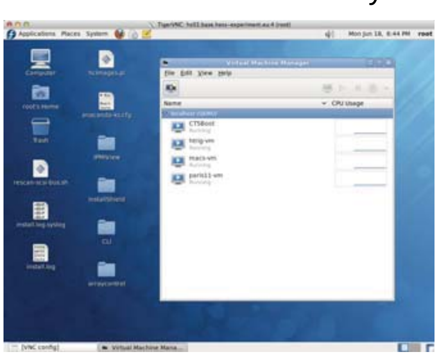
> openVPN for easy access to all important networks

> IPMI cards for remote administration of server and nodes

> VNC server for direct access to machines for problem solving

- Central control pc in control room
- All display machines in control room
- Server and nodes if necessary



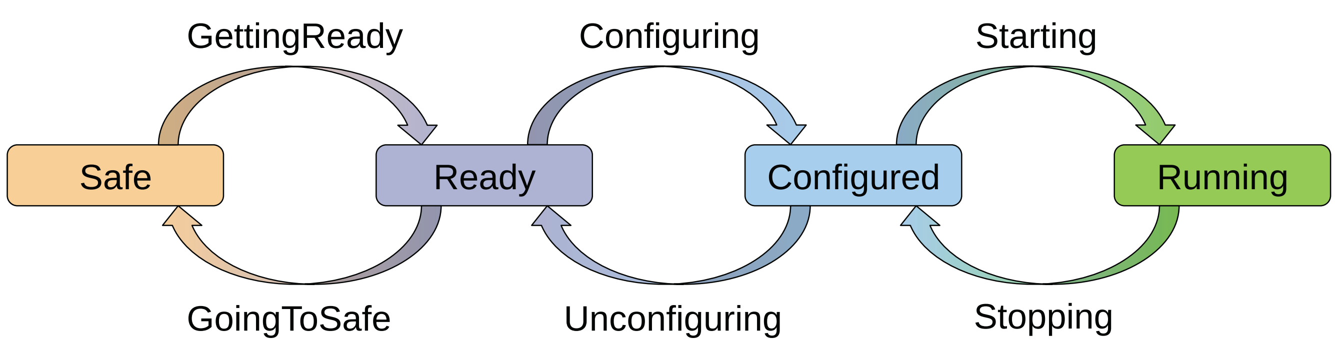


General DAQ Design

- The DAQ is a multi-machine, multi-process, multi-threaded system
  - ≈ 23 machines + dedicated hardware
  - ≈ 158 processes called controller
- Every piece of hardware has its own controller
  - Responsible for activation, management and deactivation of hardware
- Dedicated management controllers called manager
  - Responsible for synchronization and error handling
- Several other controller types with different purposes in use
  - Receiver, displays, sound, alerts, real time analysis, calibration …
- Supported programming languages:
  - C++
  - Python

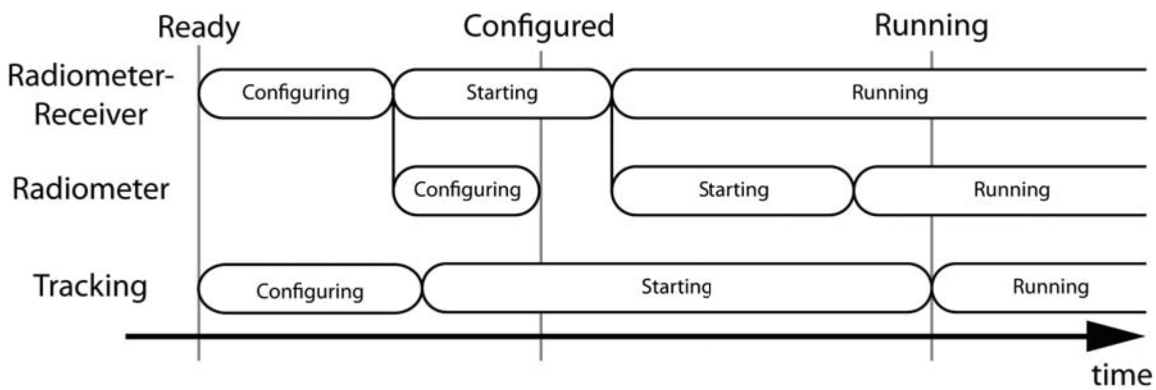
State Machine

- Every controller implements a common state machine
- There is only one transition for a given state and direction!
- Default states:
  - Day time: Safe
  - Night time: Ready
  - Data taking: Running



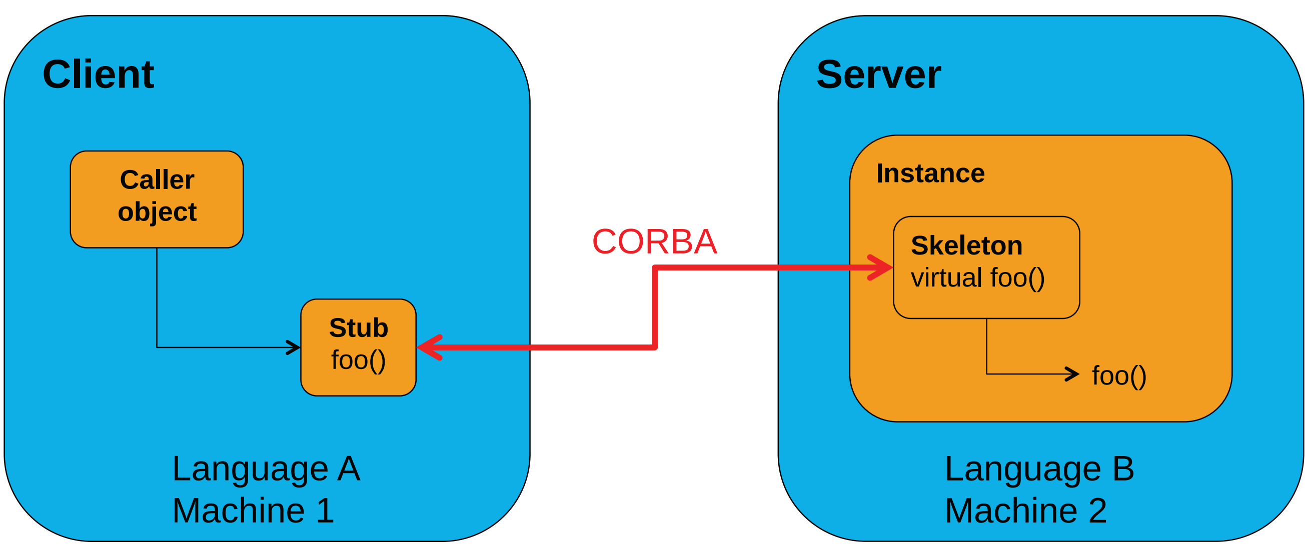
Transition Scheme

- Controller only receive a target state
- Target state does not need to be an adjacent state
- Controller will wait for its dependencies to finish their current state transition before proceeding with their own
- "Immediate" transitions will ignore dependencies!



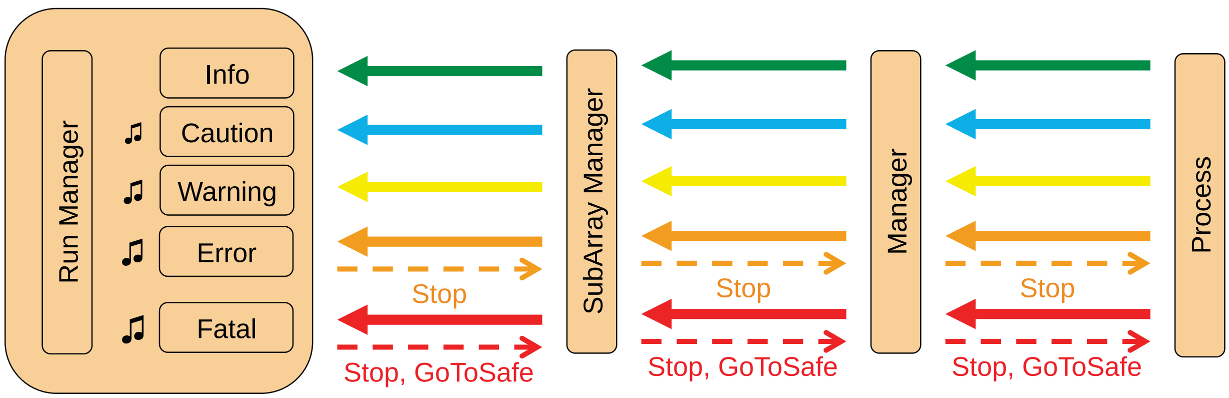
Inter Process Communication

- DAQ uses omniORB implementation of the CORBA inter process communication standard



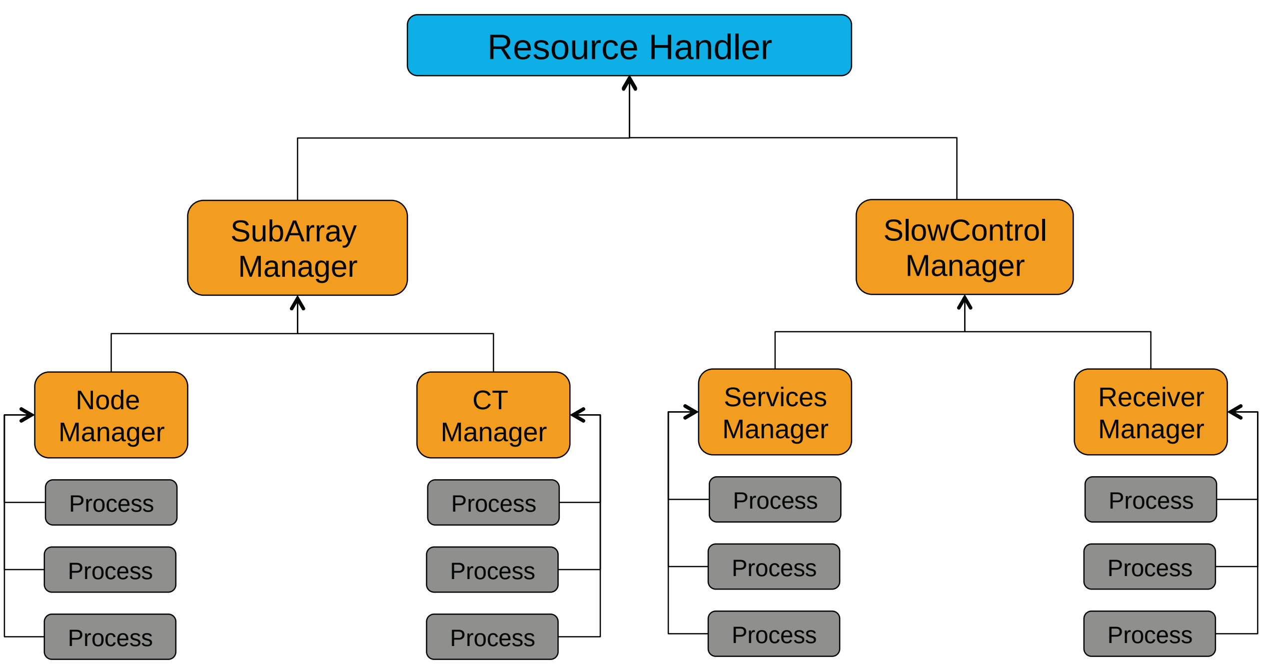
Messaging System

- Processes can subscribe to messages of other processes
- Used to monitor changes in the state of other processes
  - Error handling, …
- Used for logging and shifter information



Chain of Command

- Example for the hierarchy of the processes inside of the DAQ



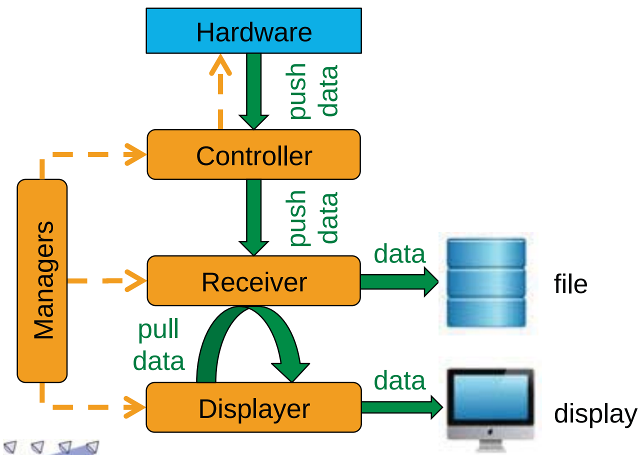
Data Flow

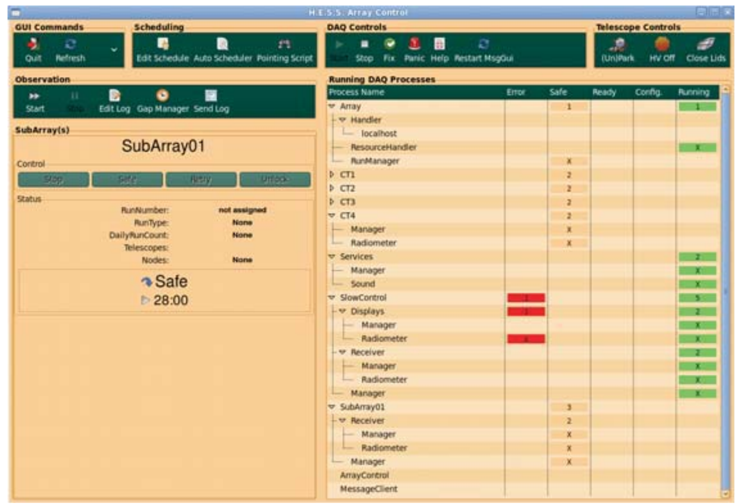
> Own receiver for each hardware controller

> Every receiver may have a displayer

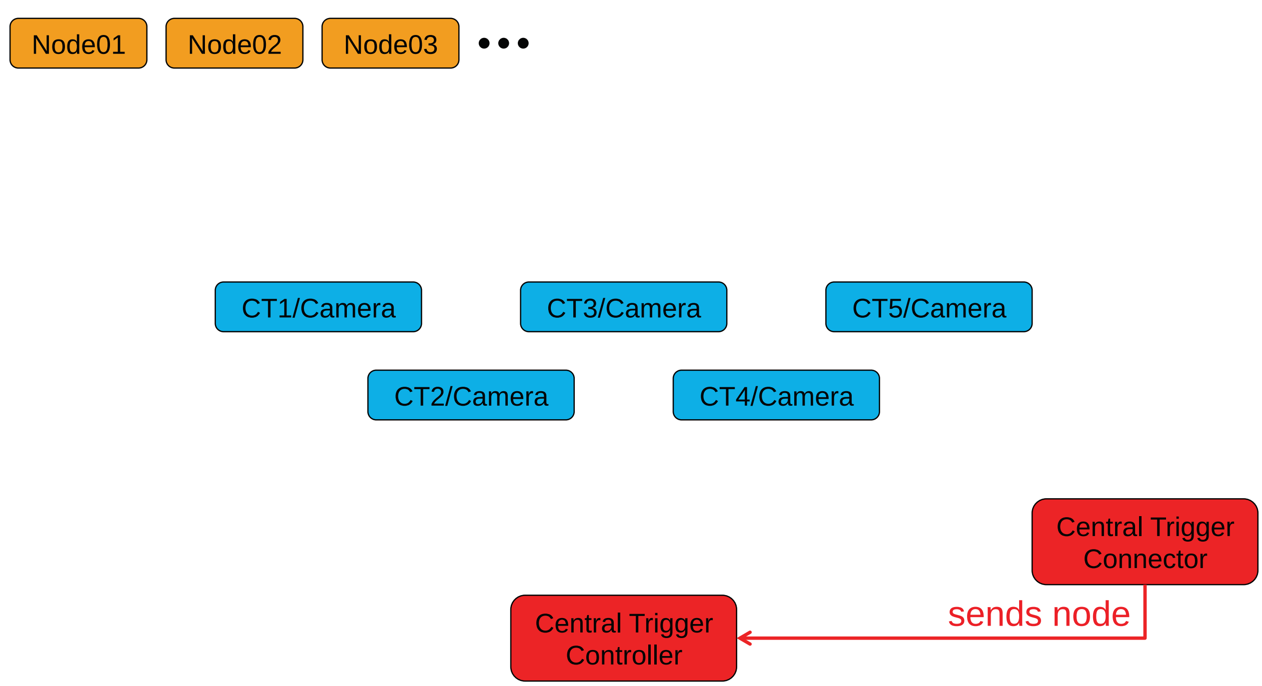
> Data can be "pulled" or "pushed" between processes

- Push: guaranteed data integrity
- Pull: data may be dropped

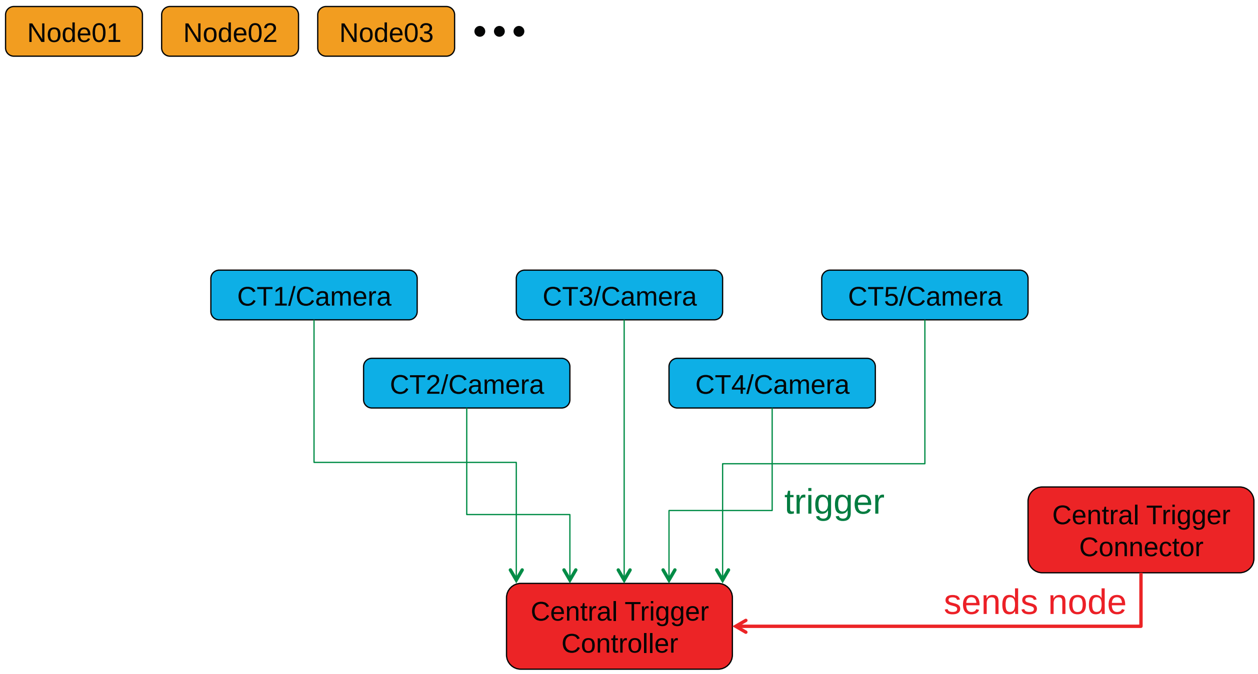




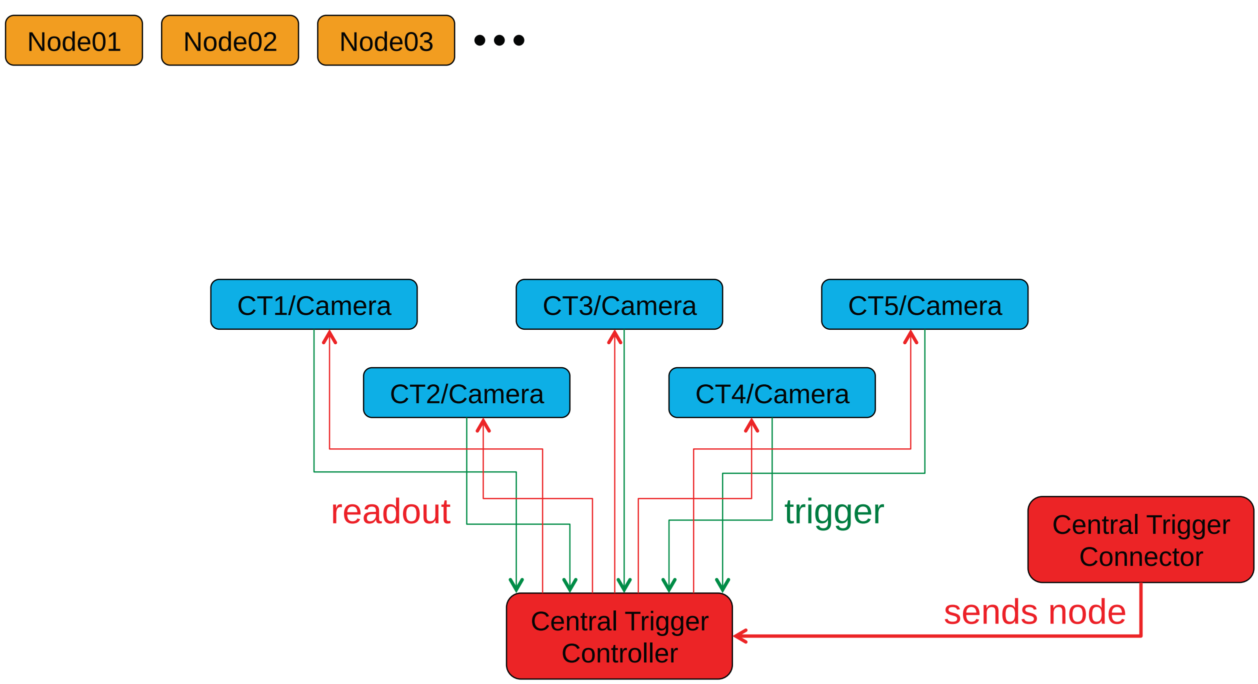
Camera Readout and Node switching



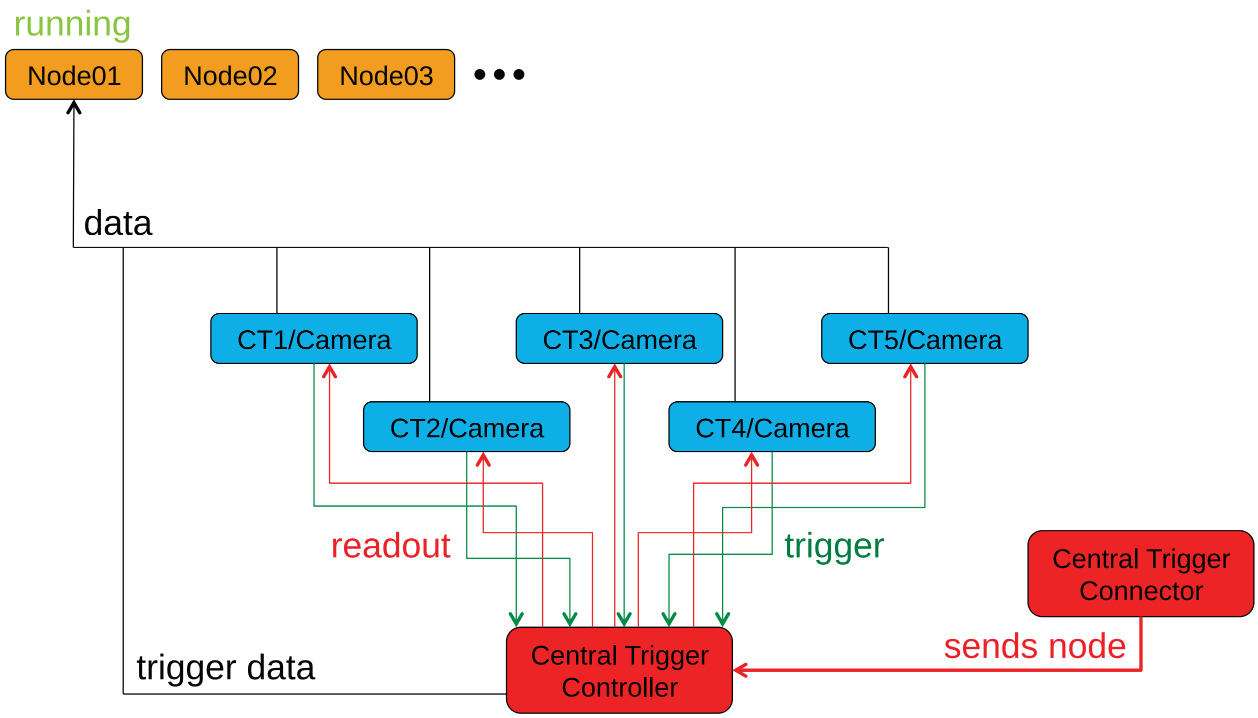
Camera Readout and Node switching



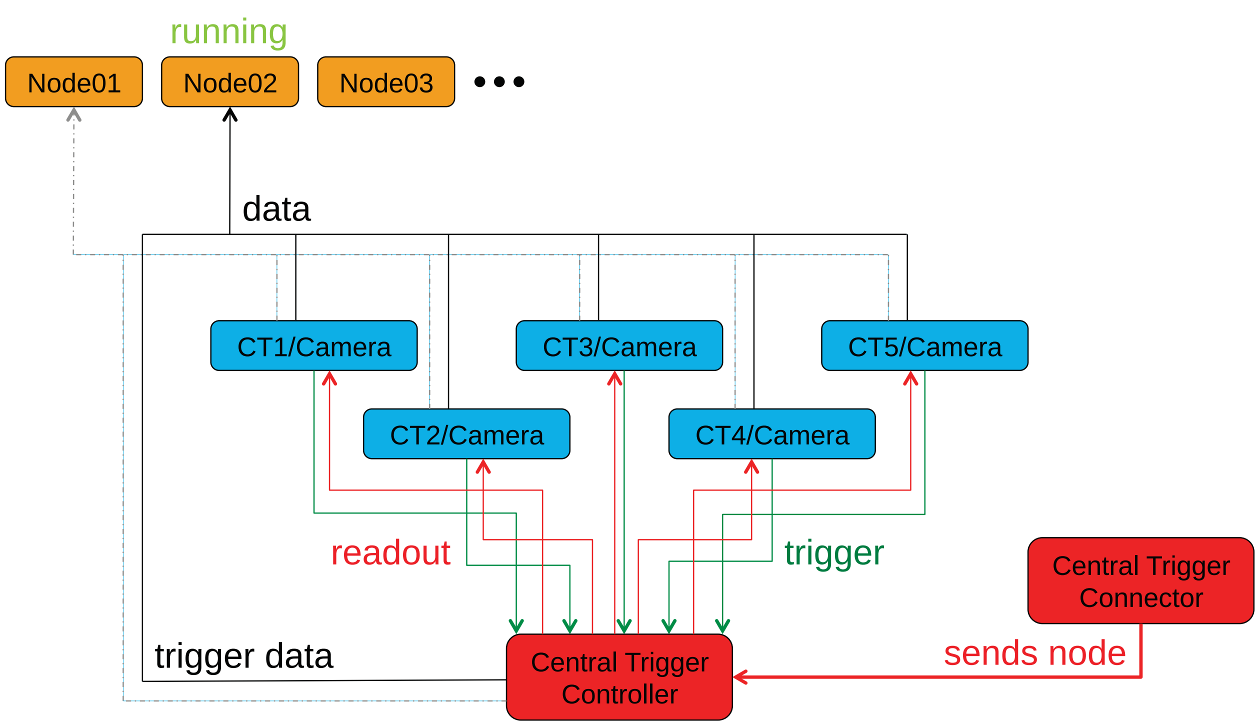
Camera Readout and Node switching



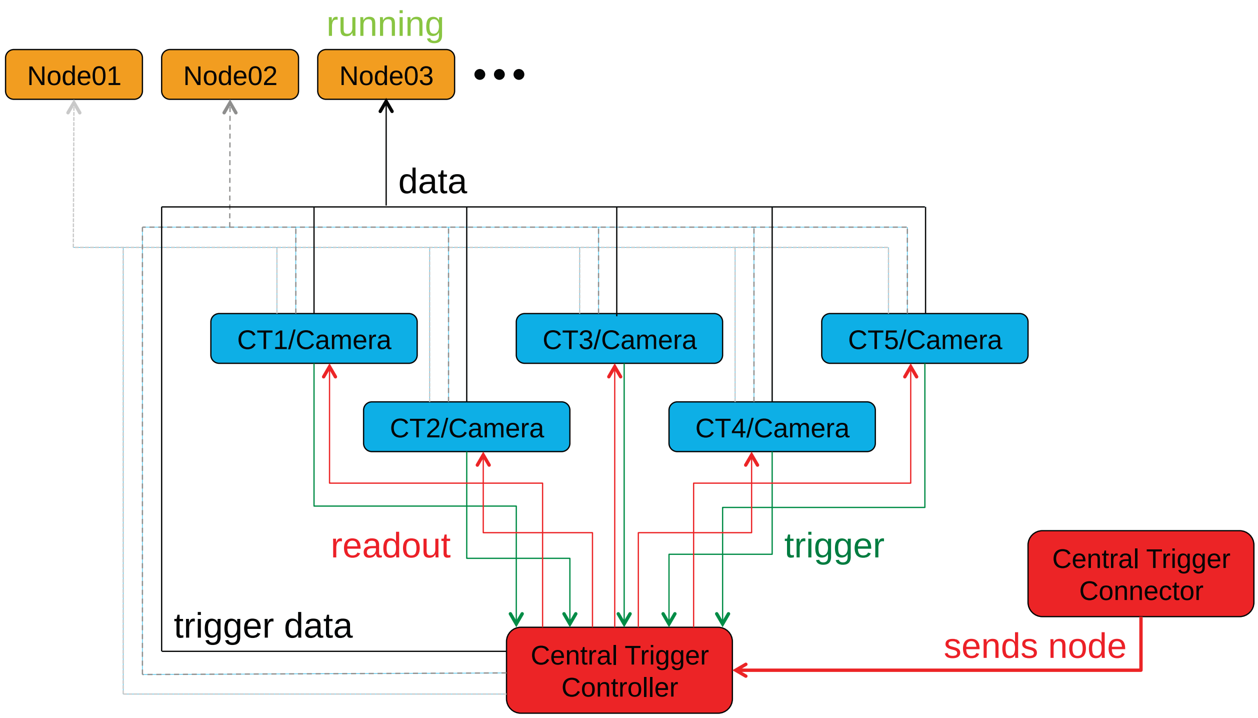
Camera Readout and Node switching



Camera Readout and Node switching



Camera Readout and Node switching



The Displayer

- One generic display controller in the DAQ
- Configured solely via a database
- Uses ROOT object introspection capabilities
- Able to display any data member of any storage class
- Can use every histogram and graphic class of ROOT
- Possibility to specify warning and error limits

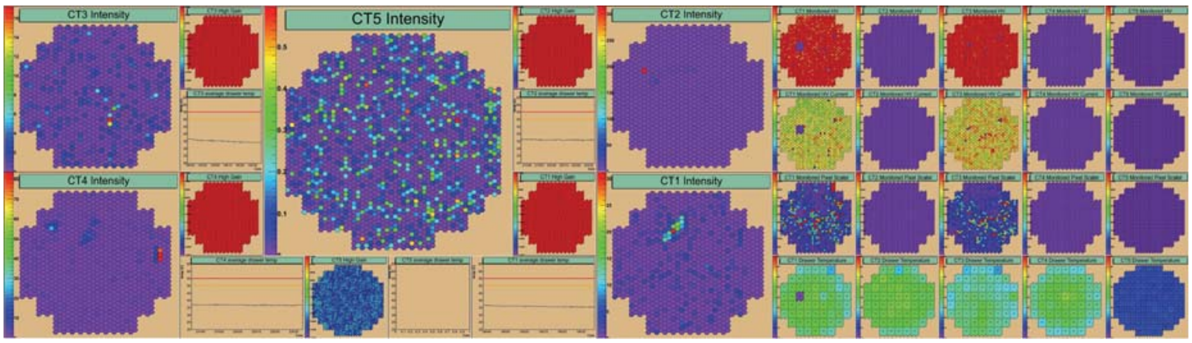
Control Room Displays and Central DAQ GUI

> Dedicated display machines
  - X11 forwards from any machine in the data network

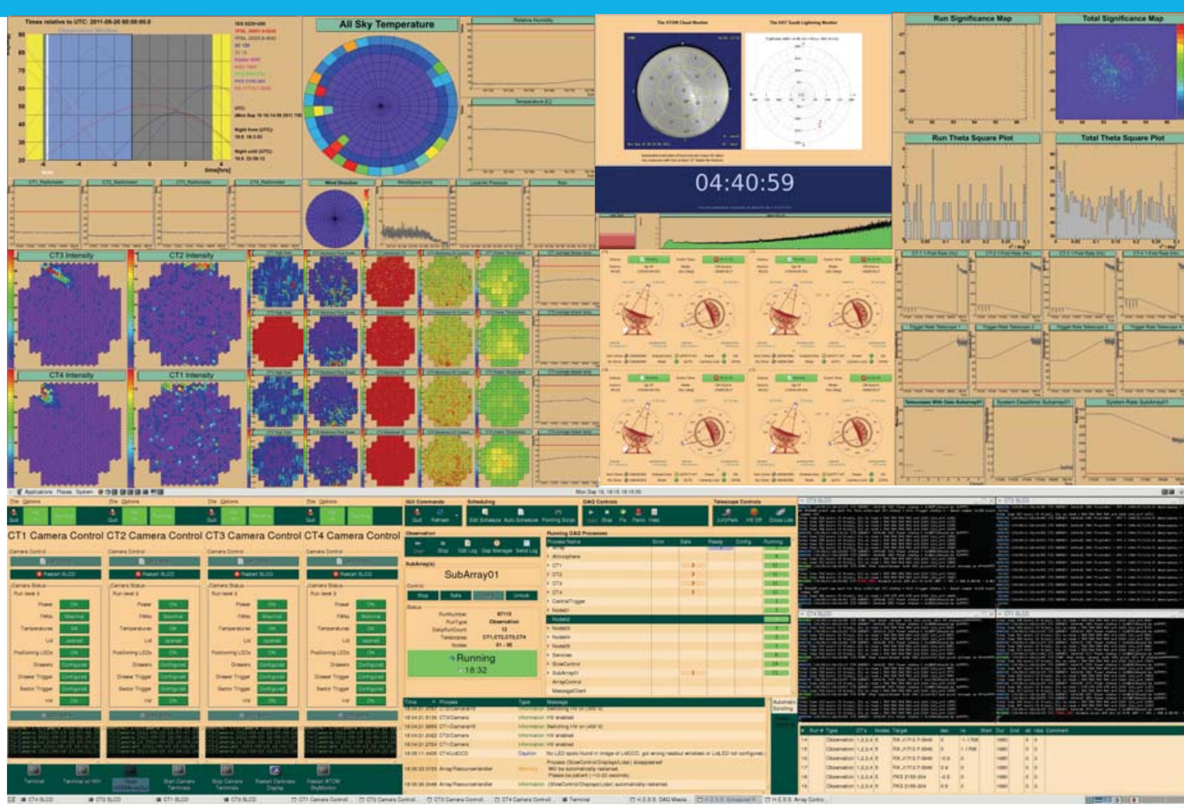
> Most displays use ROOT graphics and histograms

> Some displays written in Python using PyGTK 2.4

> Central DAQ-GUI and camera controllers also in Python



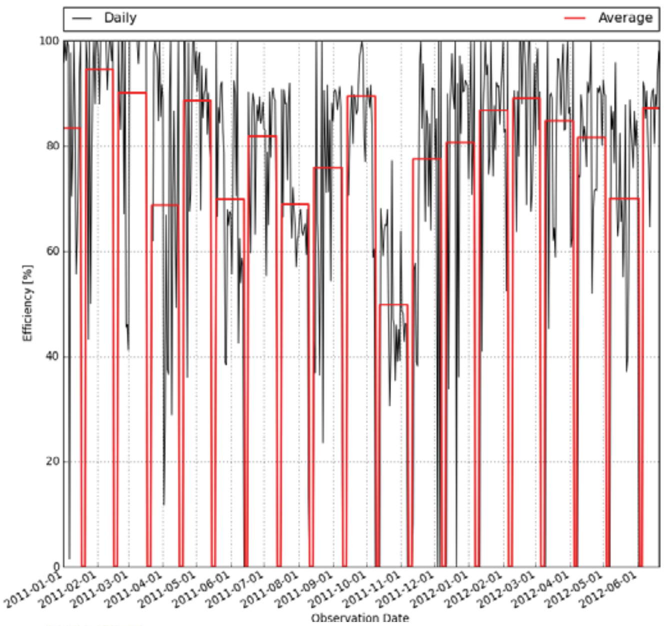
Control Room Displays and Central DAQ GUI



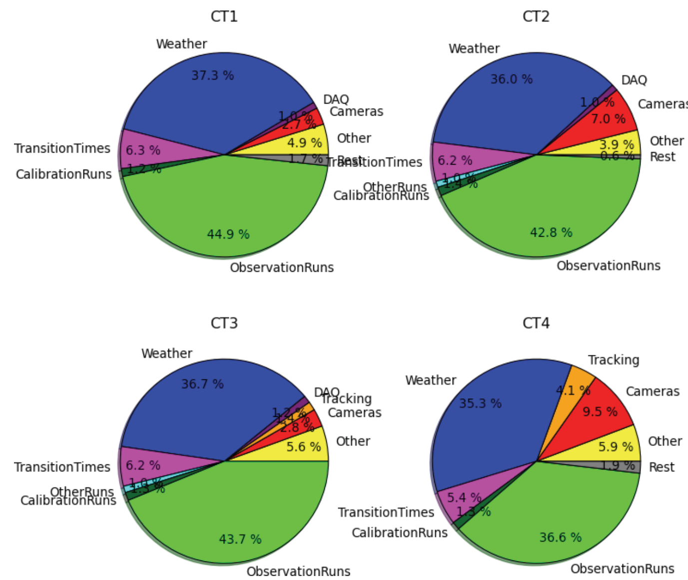
Monitoring the Array Performance

> H.E.S.S. II DAQ operational and in use since Spring 2011

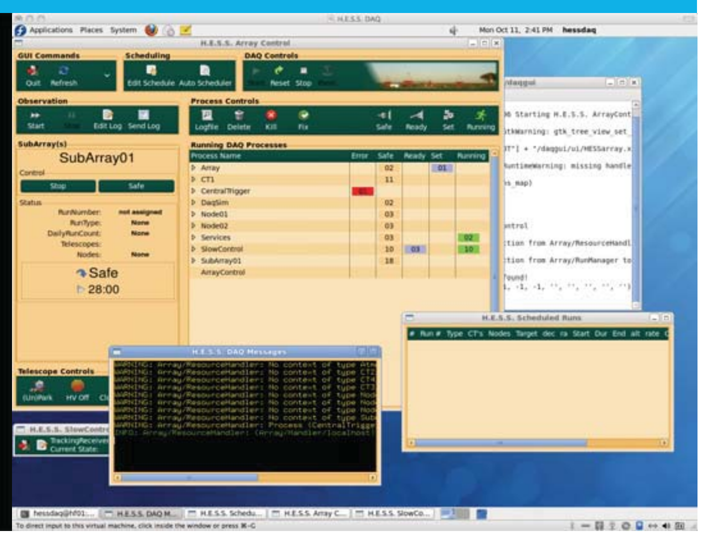
Data taking efficiency



Telescope down time



H.E.S.S. DAQ Virtual Machine



> Common platform for controller development

> Intended to test controllers and hardware

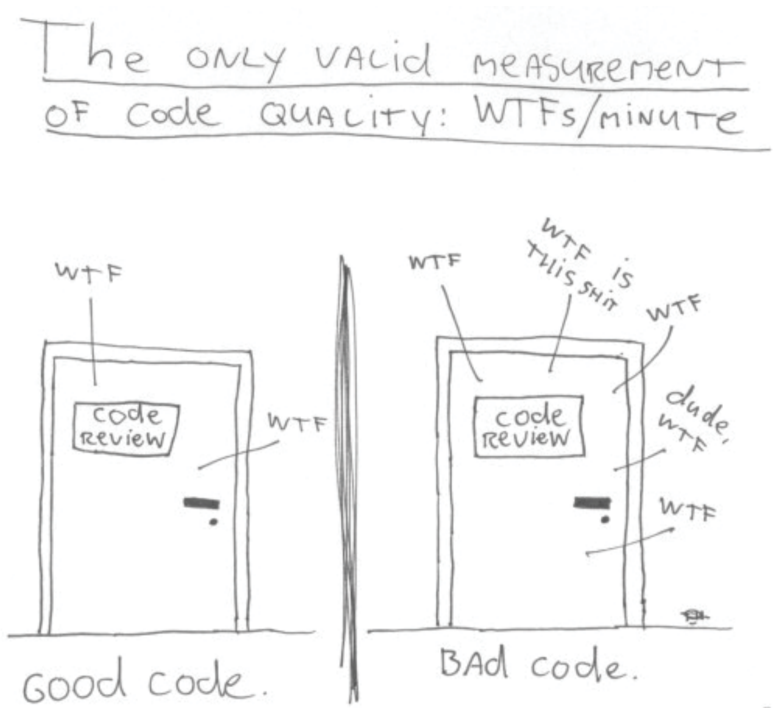
> As close to the system in Namibia as possible

Lessons learned

- Use unit tests, apply test driven development and refactor your code
- Error handling is a key feature
- Keep the software modular to keep test and development setups simple
- Focus on one programming language
- Use websites to display data
- Test software with real hardware
- Choose a language for which a good IDE is available
- Have and use a coding standard
- Use a modern versioning system

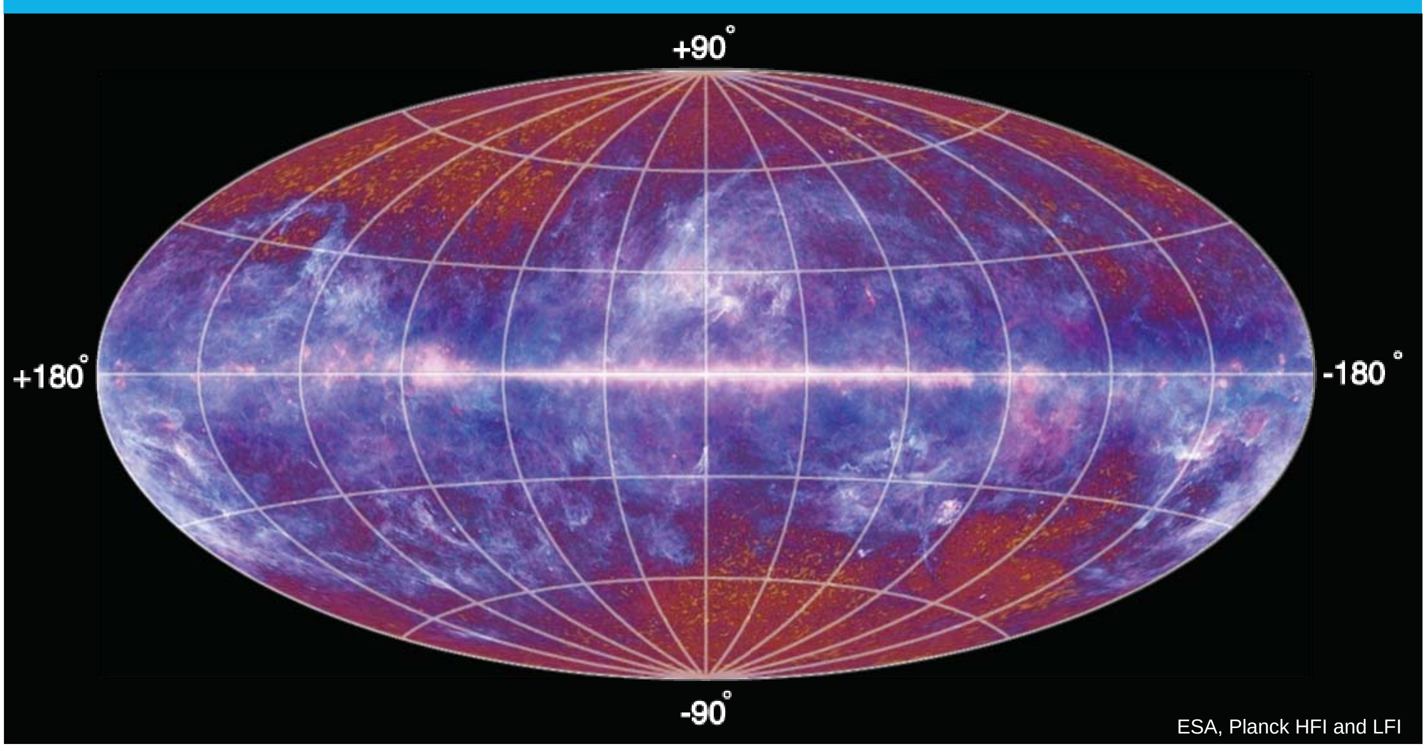
Conclusions

- The H.E.S.S. DAQ is working fine and it is well prepared for H.E.S.S. II



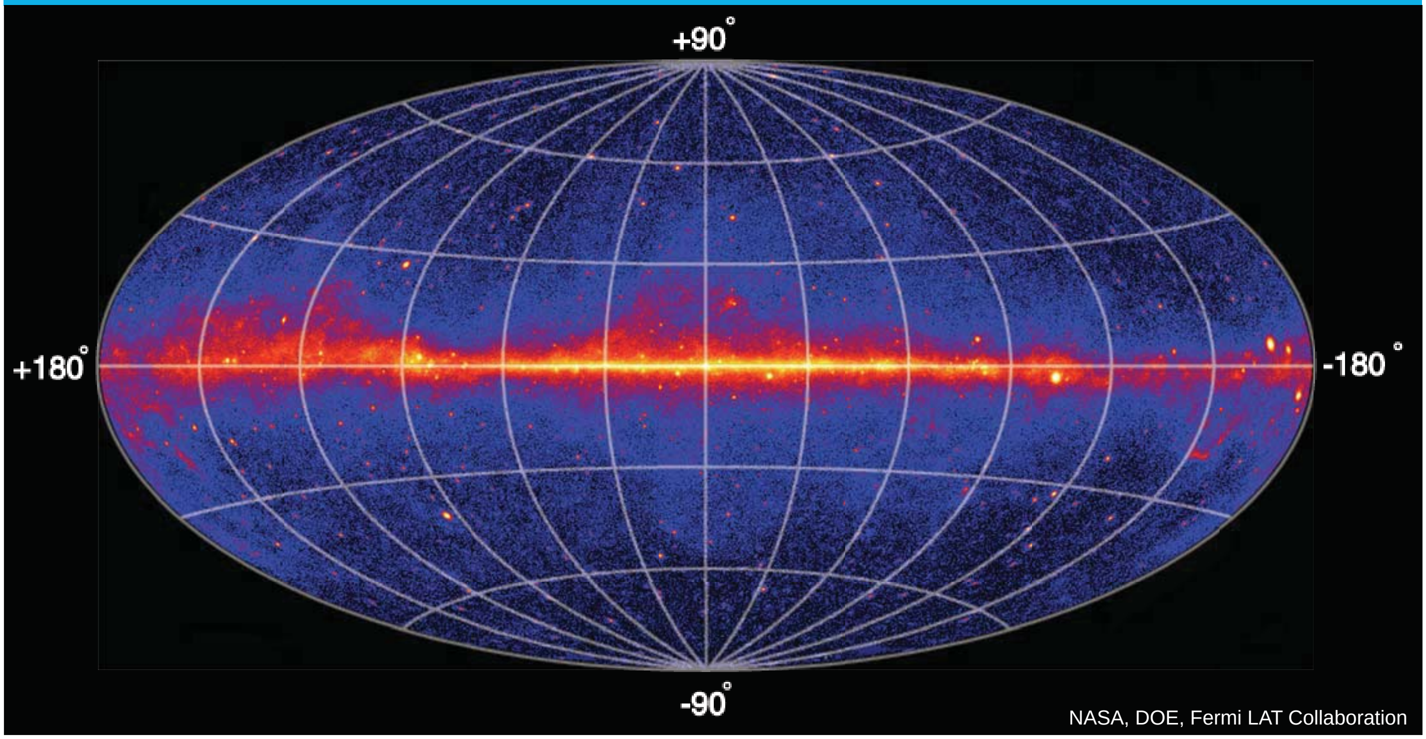
Taken from
"Clean Code"
by Robert C. Martin

The Microwave Sky

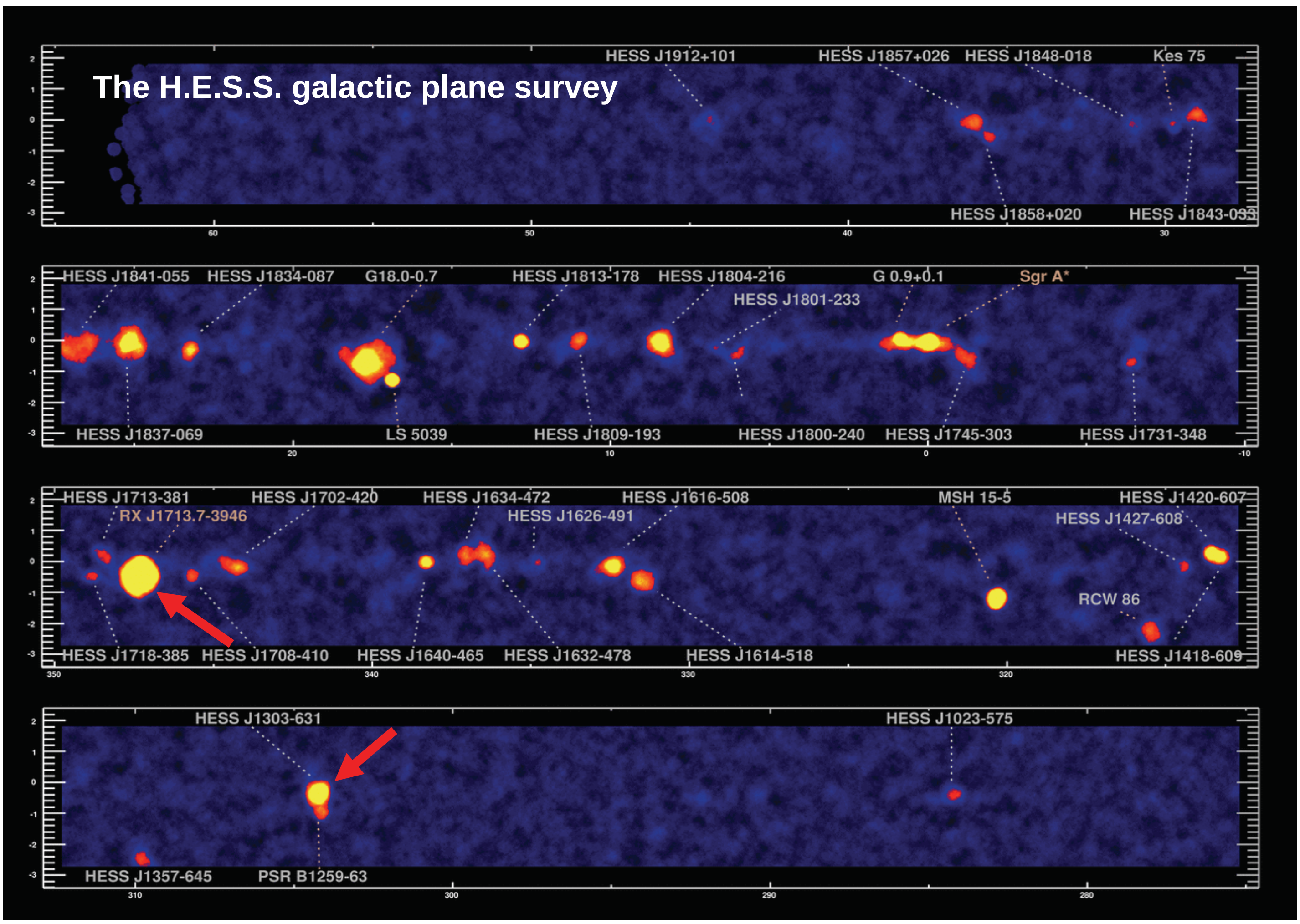


> Energy of radiation: 0.000001 eV = 1 µeV

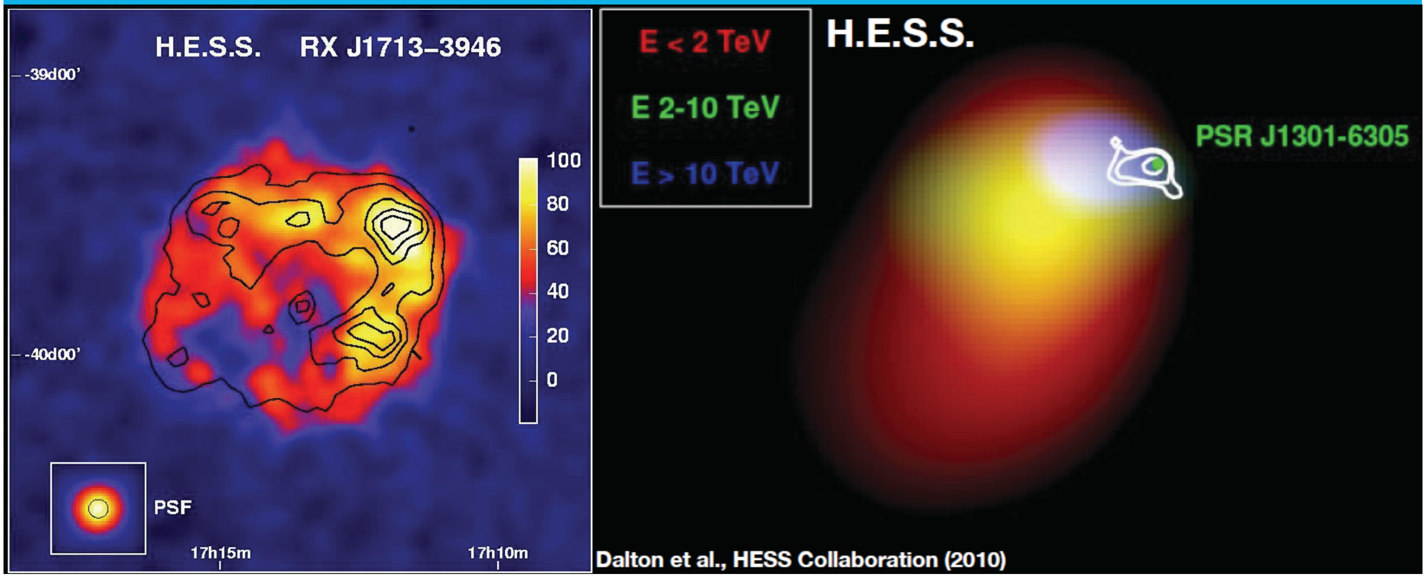
The High Energy Gamma-Ray Sky



> Energy of radiation: 1000000000 eV = 1 GeV



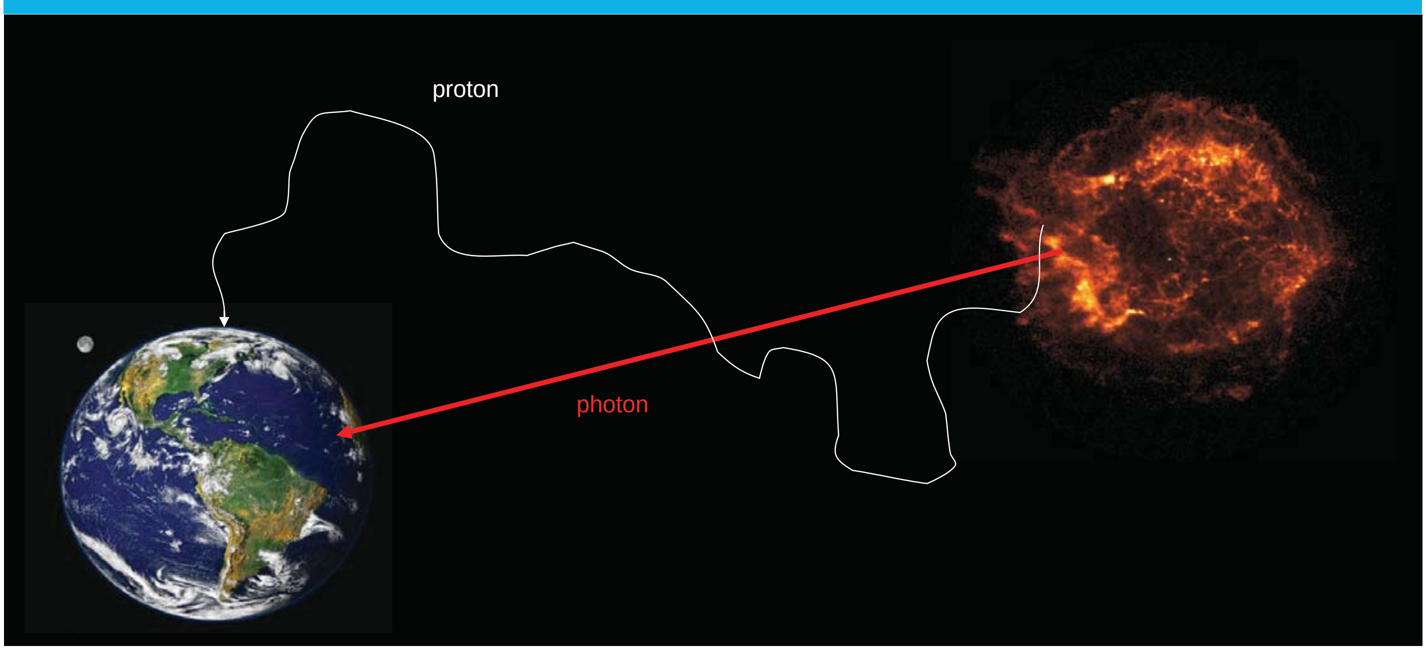
Two Sources Up Close



RX J1713-3946
shell like supernova remnant

HESS J1303–631
pulsar wind nebula

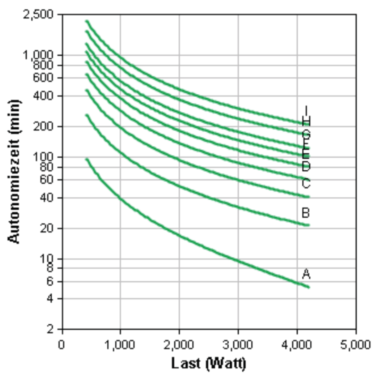
Propagation of Particles

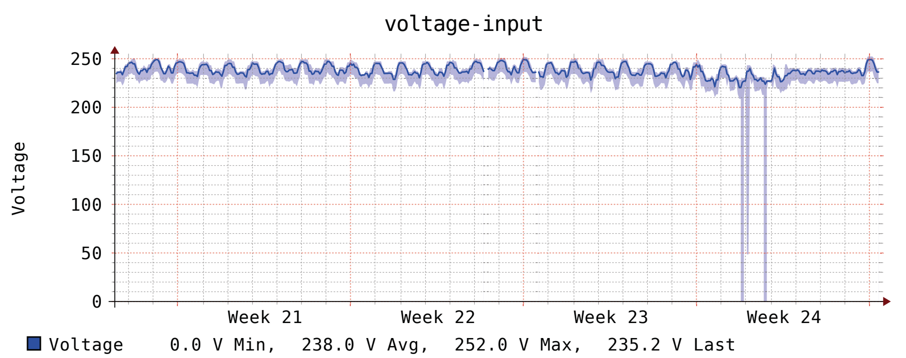


> Gamma radiation points back to its origin

Uninterruptible Power Supplies

> 3 APC Smart-UPS RT 6000 VA
  - Each with a APC Smart-UPS battery pack
  - currently at 35% of max. capacity





Additional Equipment

> 2 Meinberg M300 GPS time server
  - Used for ntp services for the farm

> 2 Transtec 7600 LTO-4 tape drives
  - Transport of data to Europe every month

> 6 HP ProCurve 2510G switches
  - 3 in use, 3 spare

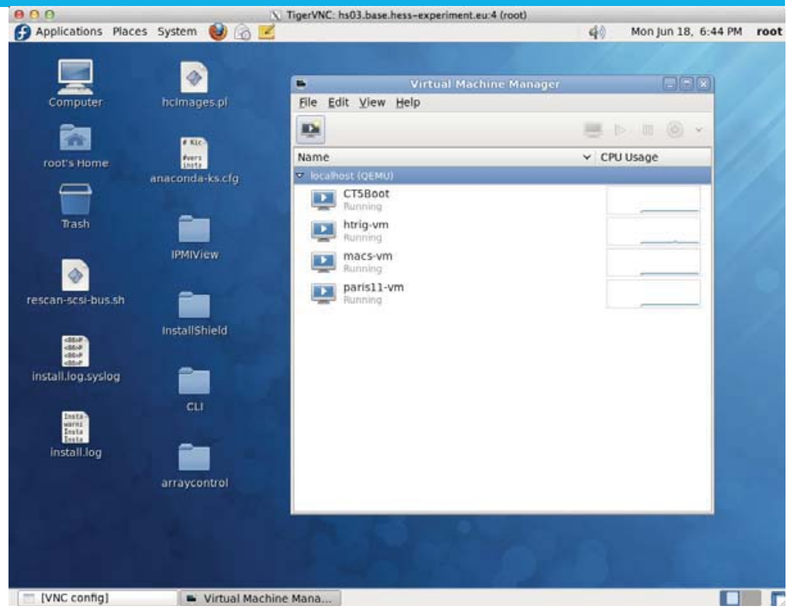
Hardware Virtualization

> Types of machines:
  - Two camera boot servers
  - Central trigger boot server
  - Machine for backward compatibility to 32 bit software and old kernel

> Running on one of the servers
  - RAID6
  - Backup

> Almost no maintenance

> High availability



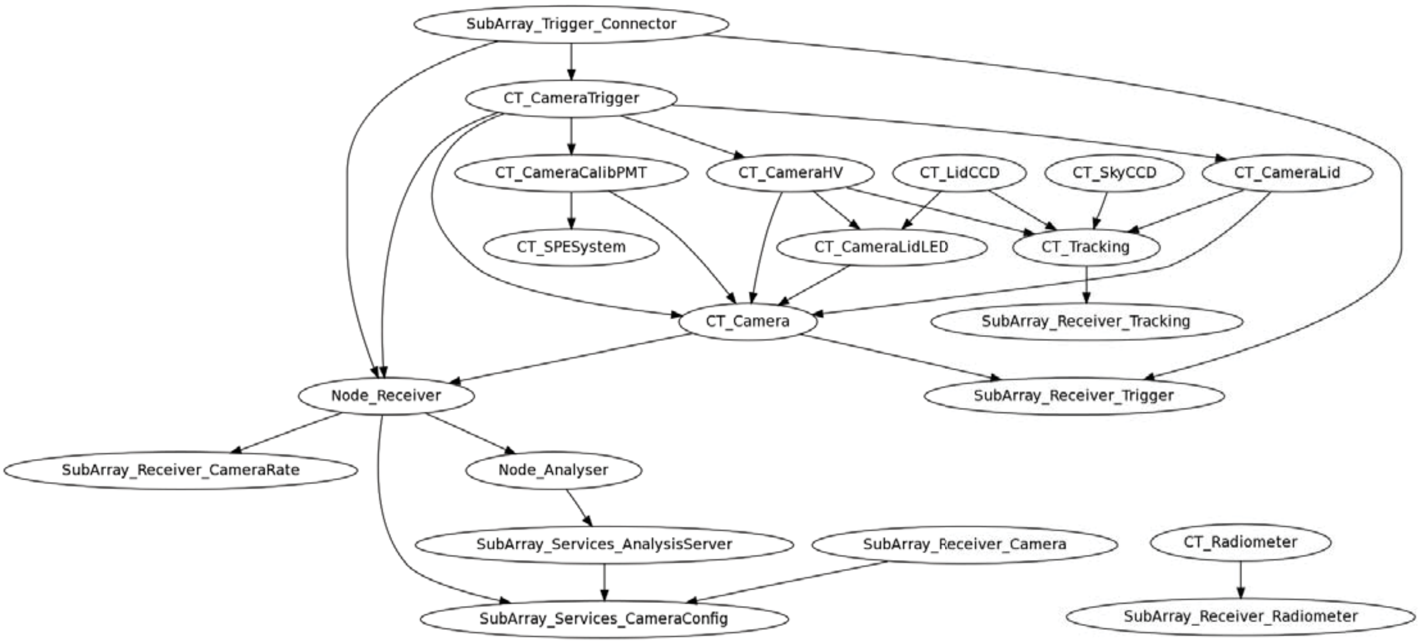
Handling Controller Dependencies

> A controller can depend on any number of other controllers
  - Global database table, must be applicable for every run type

> Controller participation in a run can be specified in a database table
  - Required: Controller must take part in the run, no errors allowed
  - Optional: Controller can take part in the run, errors or crashes do not stop data taking
  - Disturbing: Controller is send to safe state and does not take part in the run

> Controller states monitored and enforced by manager processes

Current Controller Dependencies



Data Storage and Access

> Data is stored using ROOT object serialization mechanism

> ROOT graphics and histograms used in online and offline software

> Data storage classes are experiment specific

> Storage classes can be send within the DAQ using CORBA

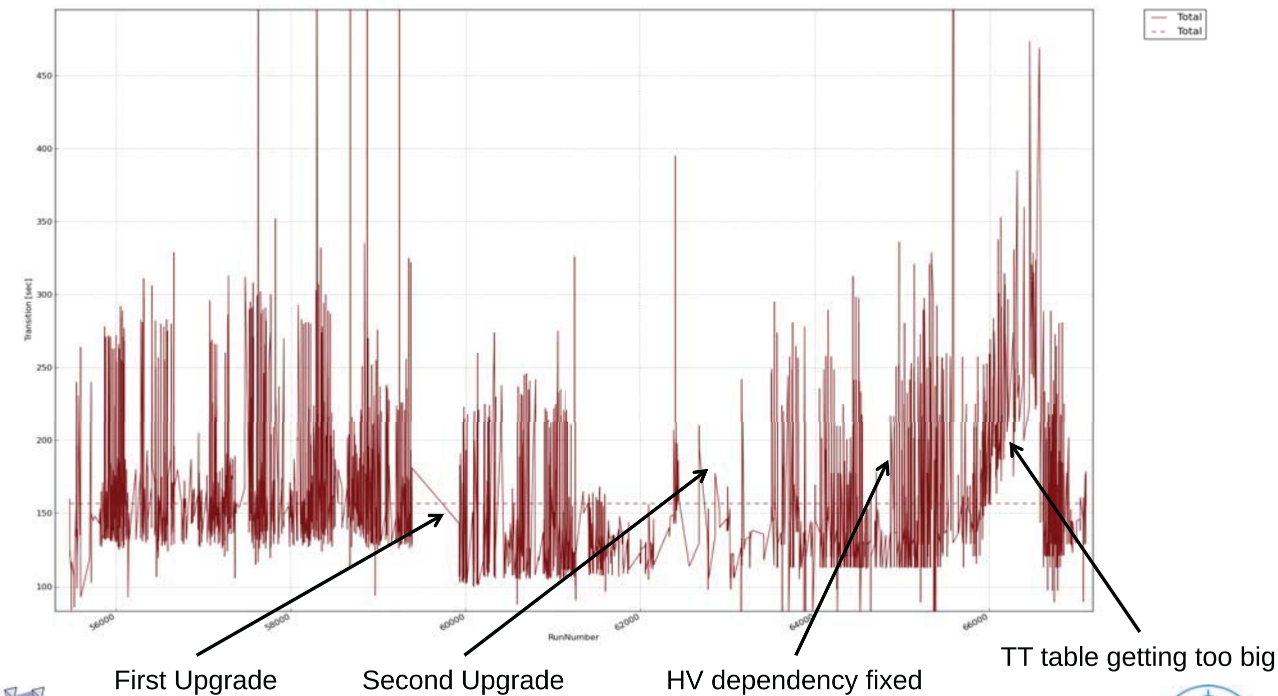
> ROOT object introspection capabilities used for generic access to easily display data in real time

- Access to every member of every storage class
- Displays configured via a database

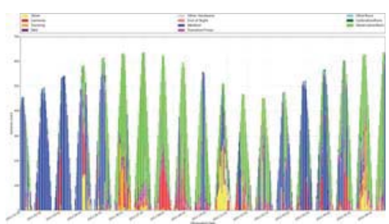
> Data calibration, reconstruction, … algorithms can be used in online and offline software

Monitoring the DAQ Performance

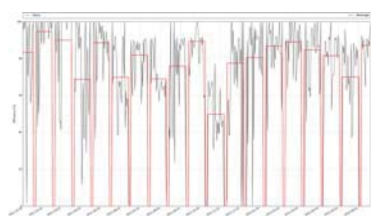
Total transition durations: 01.01.2010 until 30.09.2011

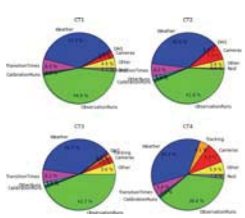


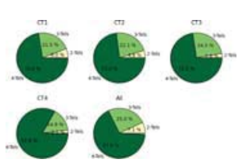
Transition Time Tools – Available Plots

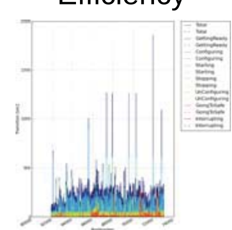

Performance

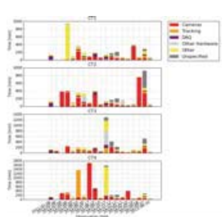

DAQ restarts over time

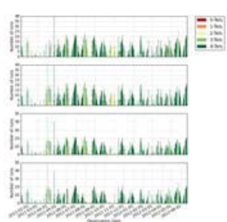

Efficiency

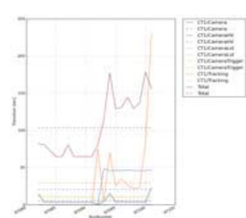

Tel Down Time


Tel Participation


Controller Performance


Down Time per Period


Participation per Night


Controller Comparison