Telescope Camera

2nd CTAO school, La Palma

Daniel Kerszberg

Disclaimer

- Give you some elements of what Cherenkov cameras are made of and how they work
- [Make you realize why you, as physicists, should care about hardware]
- Trigger some questions in you in the next days when you visit different telescope/see the real thing with your own eyes
- “Why me?” and my own biases:
    - I was in your shoes exactly 10 years ago
    - I did my PhD in H.E.S.S., 6 years postdoc in LST (camera) and MAGIC, now working on NectarCam (one of the MST’s camera with FlashCam) and LSTCam
    - “I’m not a camera hardware expert but a camera software control expert” → I didn’t contribute to the development of any of the hardware inside any camera, but I spent the last 7 years talking endless hours (days and nights) to the experts that did

Outline

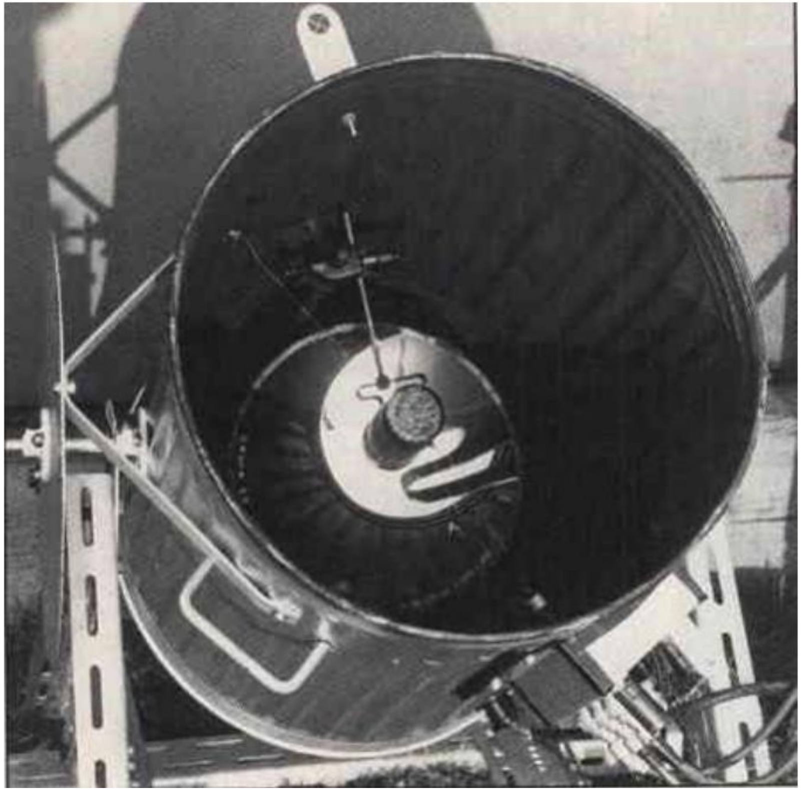
1. A bit of history
2. Camera design
    a. what one can see (hardware)
    b. what one can not see (software)
3. The example of the LST cameras
    a. its challenge (reach the lowest energies)
    b. and how to overcome it
4. Take-away message and conclusions

Outline

1. **A bit of history**
2. Camera design
    a. what one can see (hardware)
    b. what one can not see (software)
3. The example of the LST cameras
    a. its challenge (reach the lowest energies)
    b. and how to overcome it
4. Take-away message and conclusions

Detecting Cherenkov radiation in the atmosphere

- Done in 1953 by W. Galbraith and J.V. Jelley with one mirror and one PMT



A bin, a mirror, and a PMT

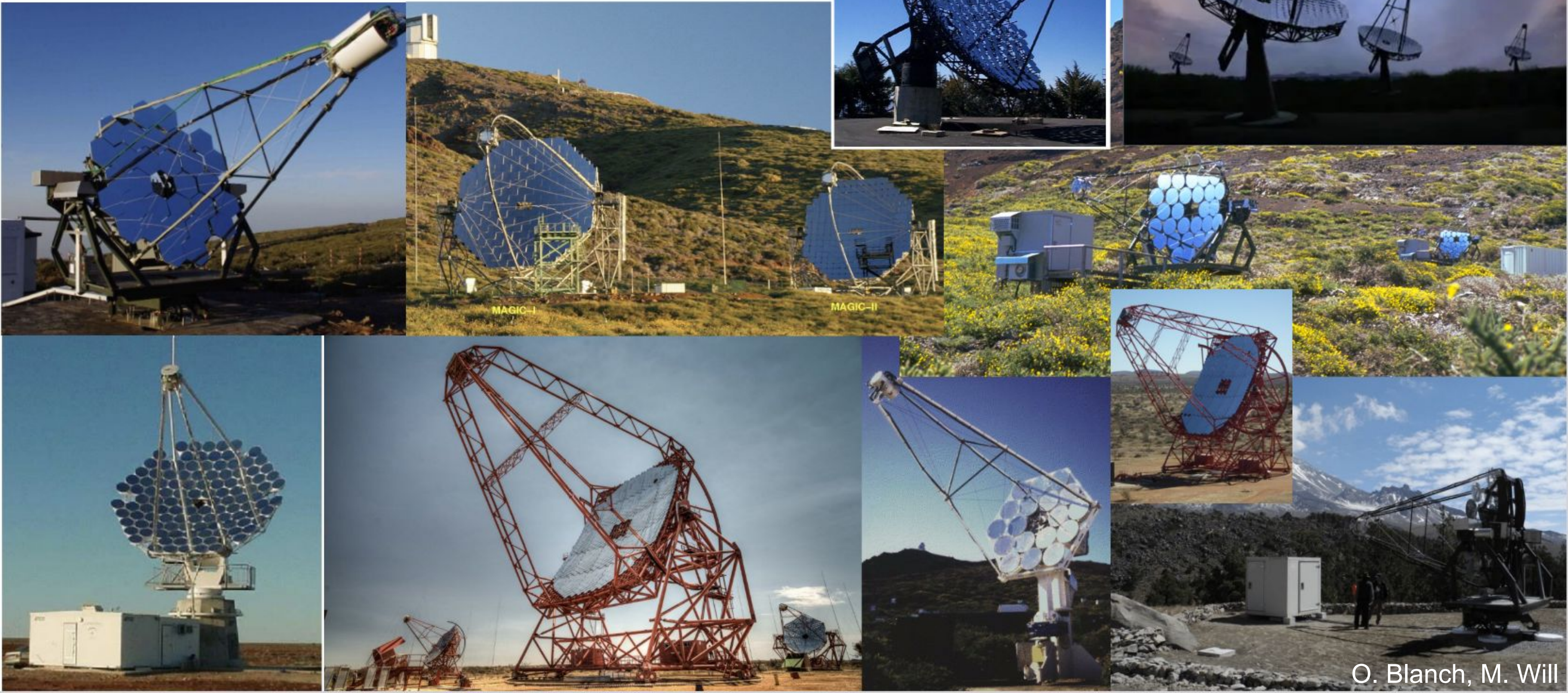
Detecting a gamma-ray source using atmospheric Cherenkov light

- Started in 1967-68 as a 10m diameter telescope with a single PMT camera
- Detected the Crab Nebula at 9 sigma in 1989 using a 37-pixel camera with each pixel size being 0.5°
  + Hillas parameters technique!

The Whipple telescope

Image Air Cherenkov Telescope

Quite some of them have already been built



O. Blanch, M. Will

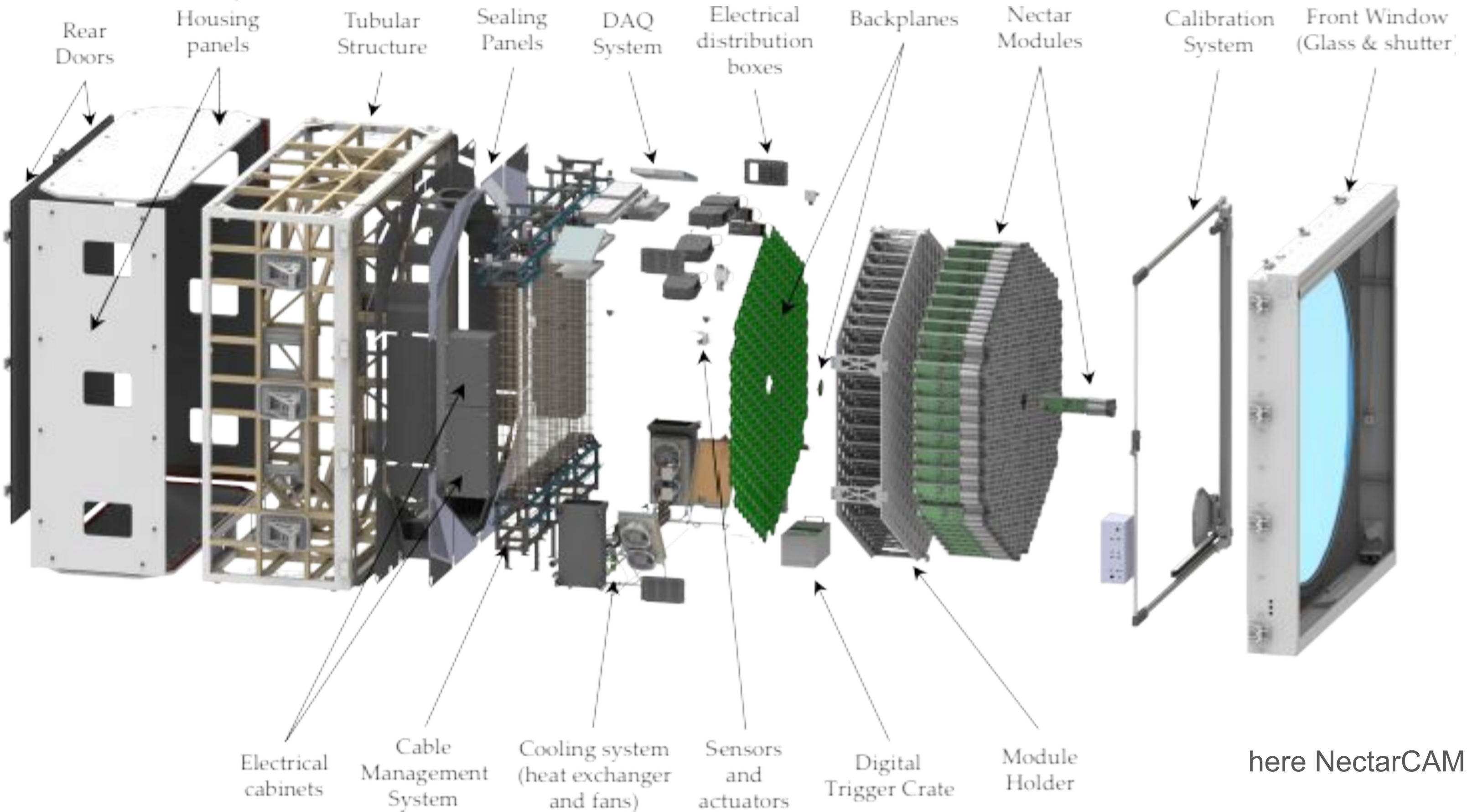
Outline

1. A bit of history
2. **Camera design**
    a. what one can see (hardware)
    b. what one can not see (software)
3. The example of the LST cameras
    a. its challenge (reach the lowest energies)
    b. and how to overcome it
4. Take-away message and conclusions

Which design to pick?

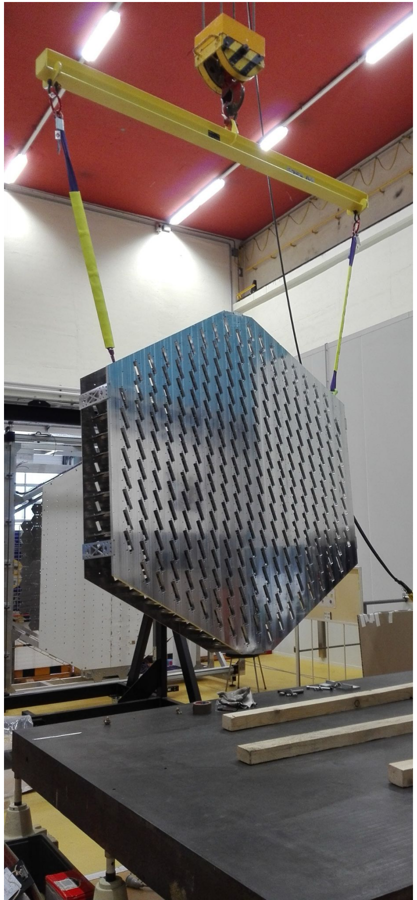
- There is no such thing as best material or best hardware
- They all have pros and cons
- The only valid questions are:
    - Does this piece of hardware fits well with the other elements/parts planned to be used?
    - Knowing the limitations/specifications of such component (heat production, noise…), can I achieve what I want with this choice of hardware?
- Weight the pros and cons, test with prototypes, and then take a decision
- Knowing the limitations of your equipment allows you to correct for some effects (requires to understand them)

Camera design



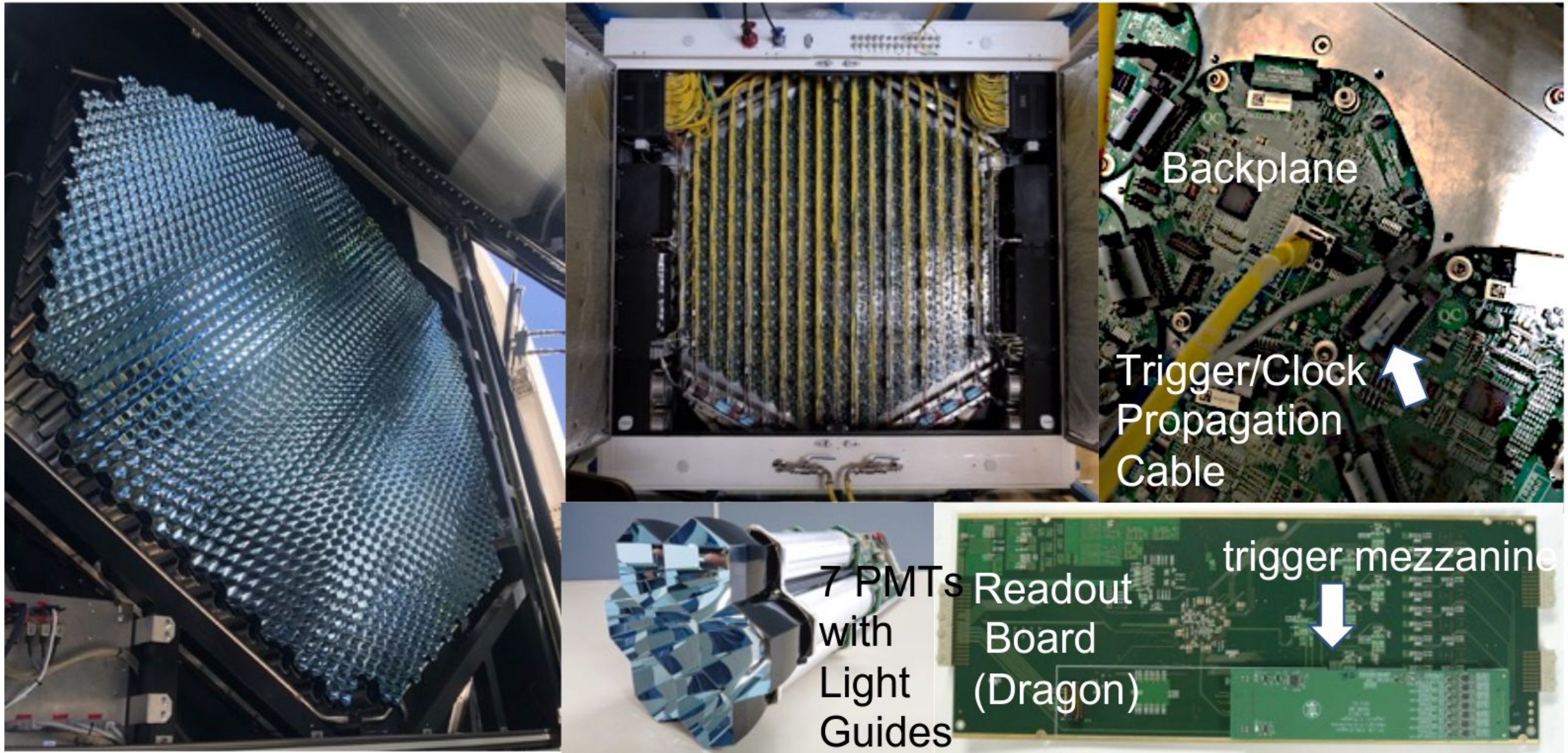
here NectarCAM

Mechanics



LST-1 camera

“Hexagons are the Bestagons”, CGP Grey, https://www.youtube.com/watch?v=thOifuHs6eY



What you can see

- Frame
- Cables
- Cooling system
- Power distribution system
- Environmental sensors
- Electronics
- PMTs
- Light guides

What you can not see

- Firmware
- Software
- Components configurations
- Components calibrations
- Integration
- Interfaces

Schematic of camera software components

Power

Environmental monitoring
And (some) safety

Pixel/Module control
and monitoring

Shutter

Trigger

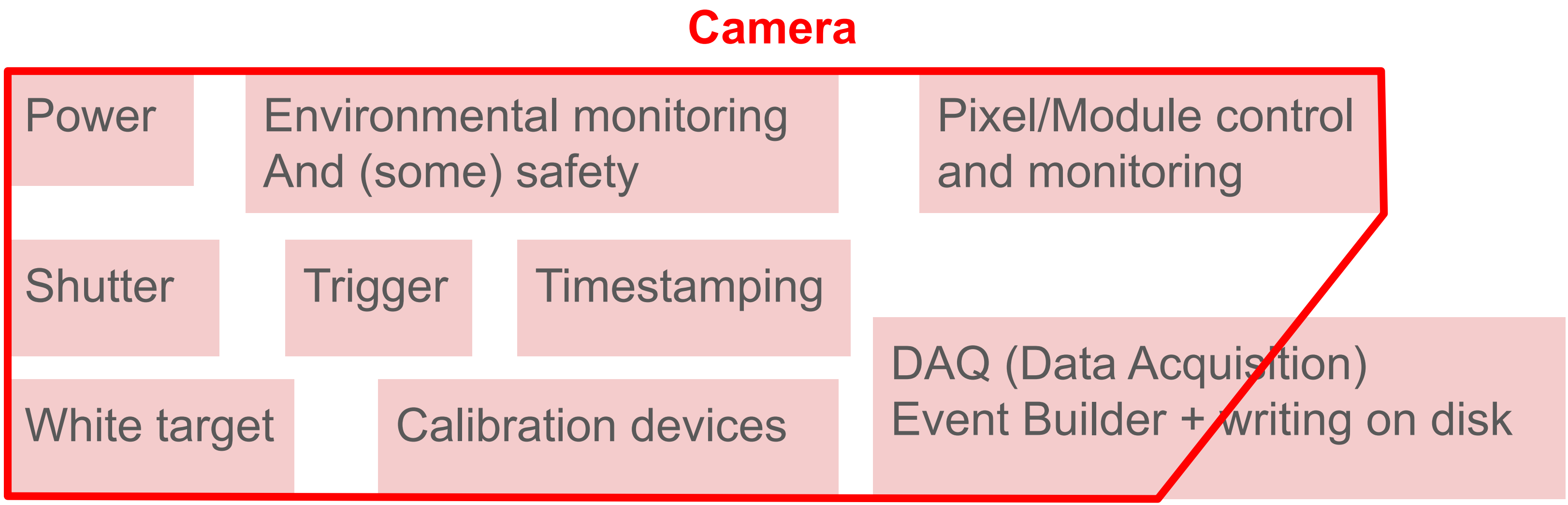
Timestamping

DAQ (Data Acquisition)
Event Builder + writing on disk

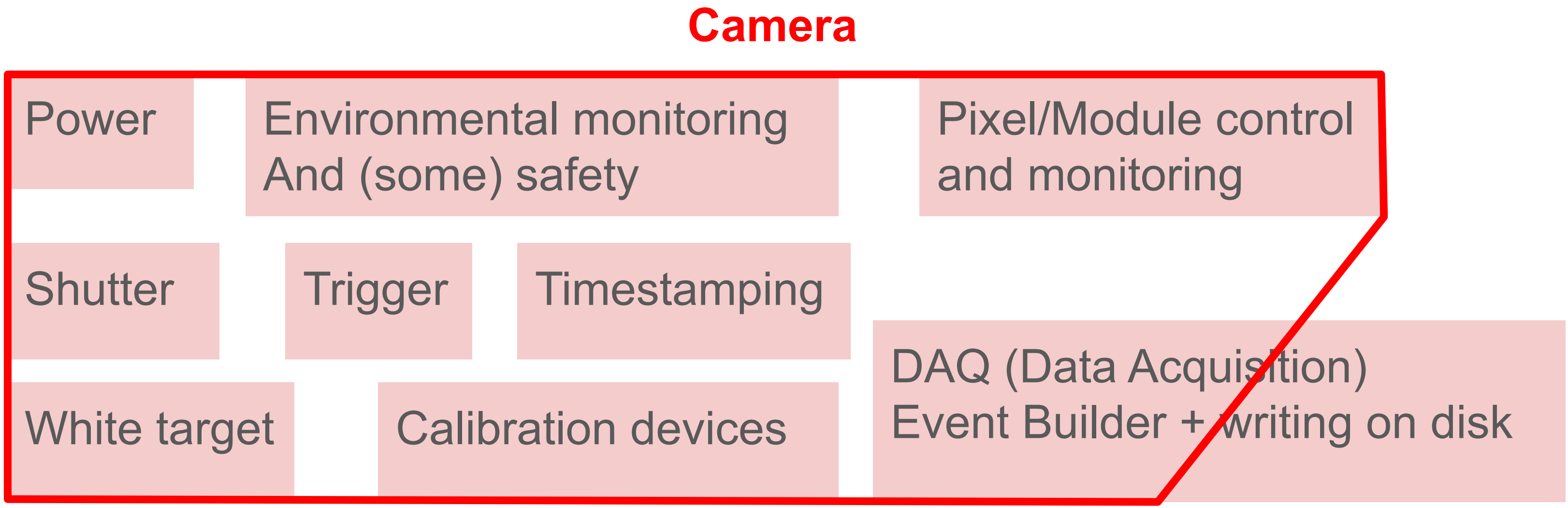
White target

Calibration devices

Schematic of camera software components

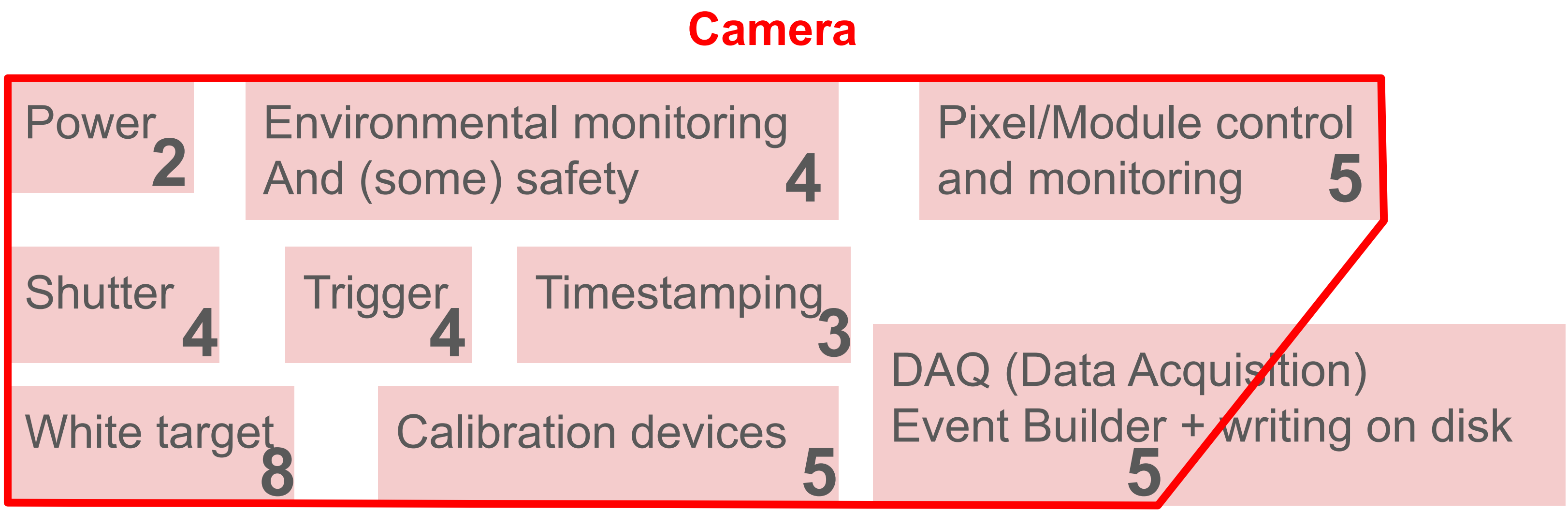


Schematic of camera software components



How many states per component?

Schematic of camera software components



How many states per component? →The LST camera example

Camera state machine

Ca


OFF
switchon
switchoff
backfromtpoint
SAFE
TPOINT
reset
movetotpoint
UNDEFINED
gotostandby
backtosafe
backfromtpoint
recover
STANDBY
TPOINT
movetotpoint
gotoready
backtostandby
READY
startrun
stoprun
OBSERVING

What about hardware/software errors or failures?

- They are to be expected, a system behaving perfectly 100% of the time is not possible. True for hardware and software!
- What can we do ?
    - Plan for them, now how to recover from known possible errors/problems
    - Plan for a way to “restart from scratch” in the worst case scenario that you lost control

- Be ready for nominal operation takes ~20% of the time (or less)
- Be ready to recover from any errors/problems takes ~80% of the time (or more)

Camera availability

[in reality it’s telescope (structure + camera) availability]

You already have bad weather, (full) moon, day time, power cuts, fire, volcanoes… Out of the remaining time how much time do you want to be observing?

→ 100%!

In reality what should you aim for?

→ 50%? 90%? 95%? 99%? 99.9%?

Outline

1. A bit of history
2. Camera design
    a. what one can see (hardware)
    b. what one can not see (software)
3. **The example of the LST cameras**
    a. its challenge (reach the lowest energies)
    b. and how to overcome it
4. Take-away message and conclusions

How to reach the lowest energies?

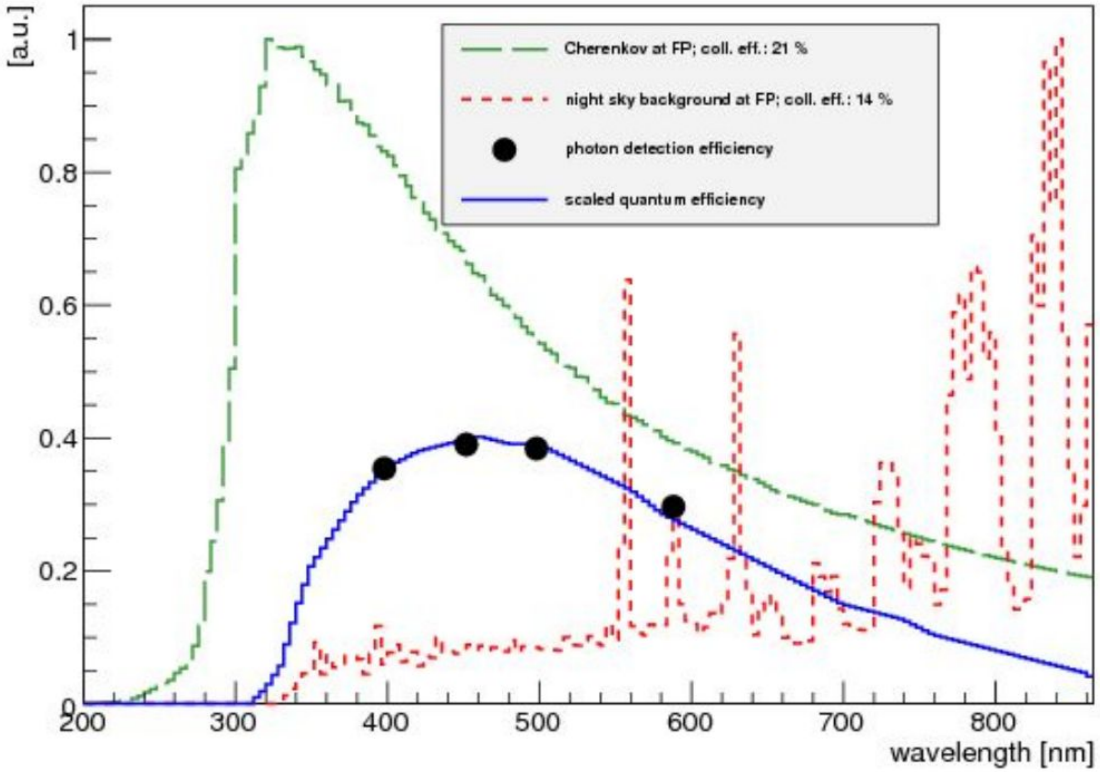
How to reach the lowest energies?

At camera level,
it’s a trigger problem:
- noise → thresholds

How to reach the lowest energies?

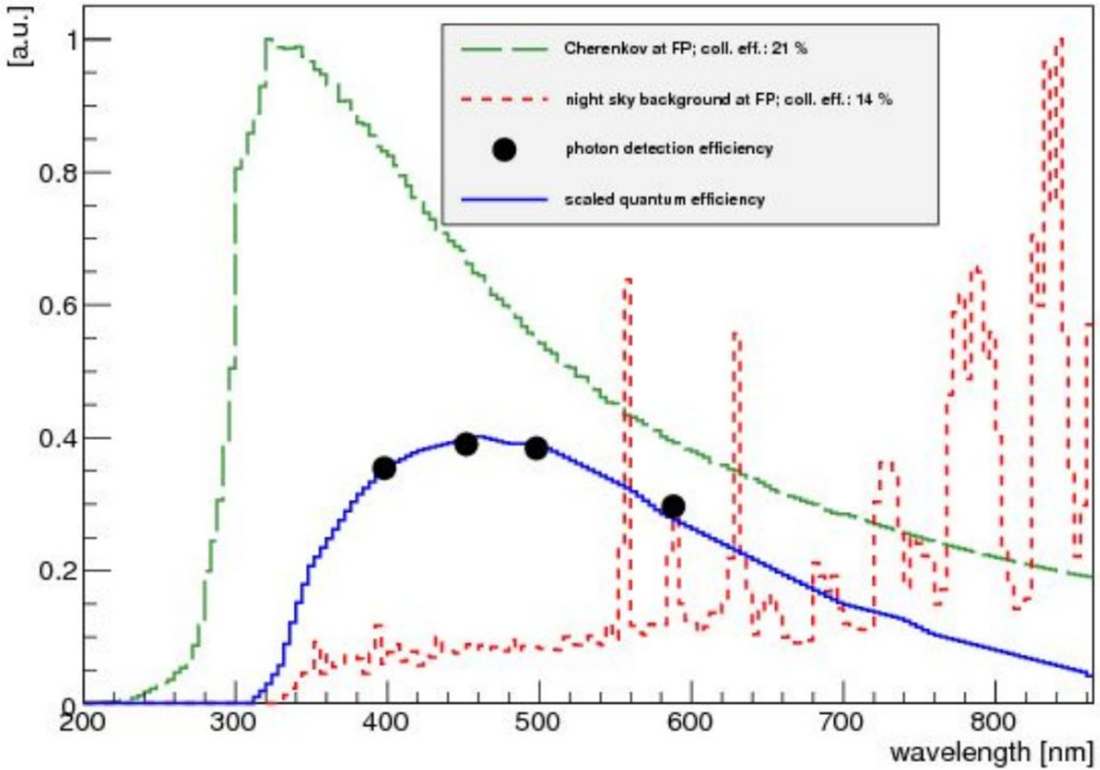
At camera level,
it’s a trigger problem:
- noise → thresholds

- Night Sky Background
  (NSB)



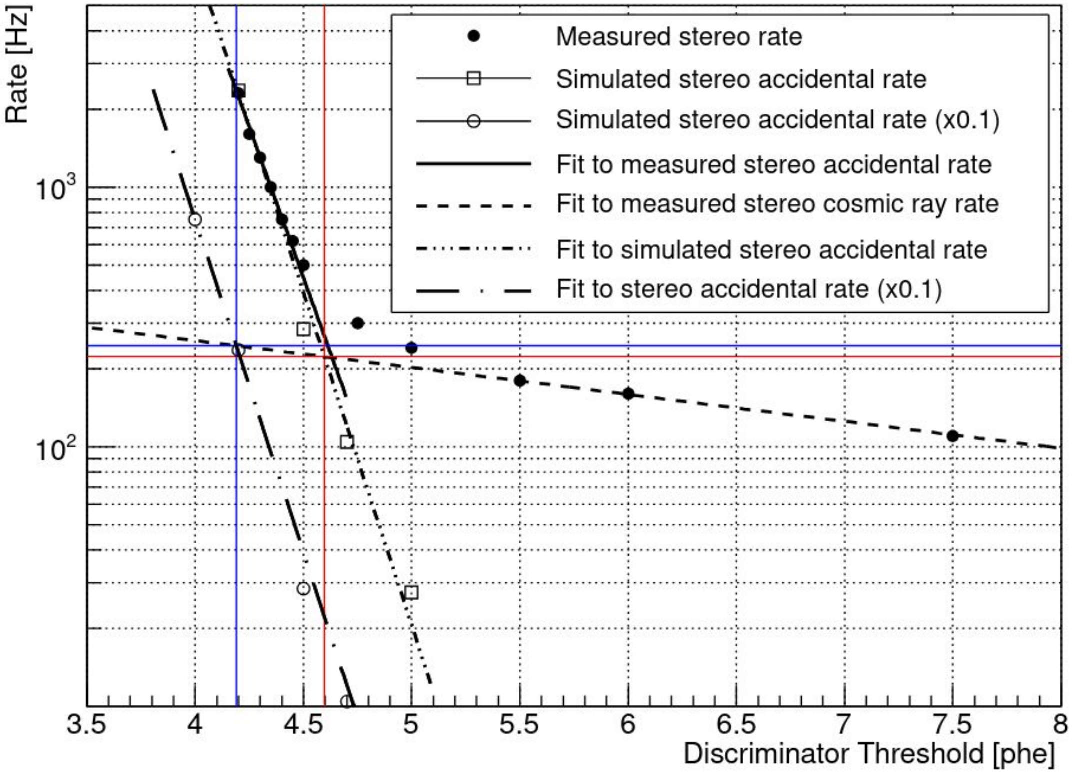
How to reach the lowest energies?

At camera level,
it's a trigger problem:
- noise → thresholds
  + trigger strategy
- Night Sky Background
  (NSB) → stereoscopy

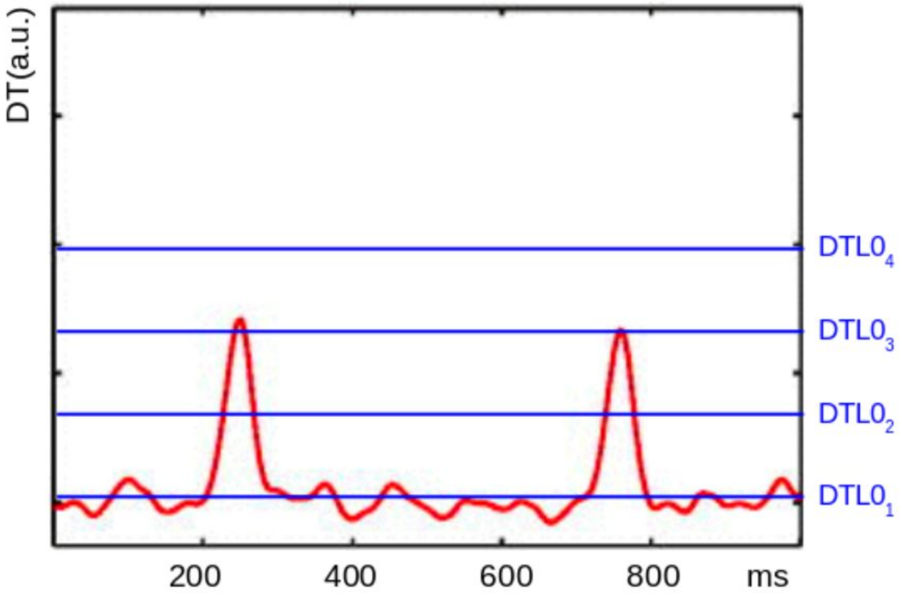


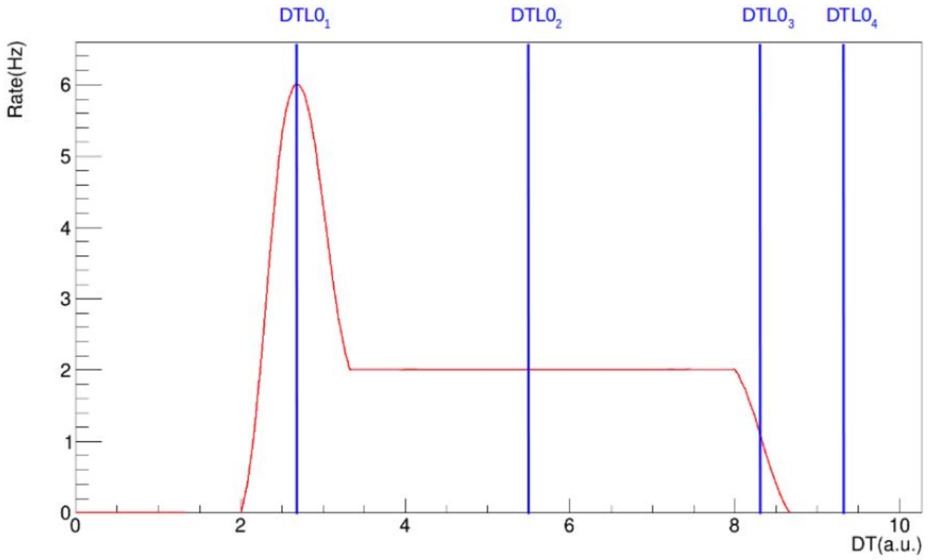
How to reach the lowest energies?

At camera level,
it's a trigger problem:
- noise → thresholds
  + trigger strategy
- Night Sky Background (NSB) → stereoscopy



Triggering



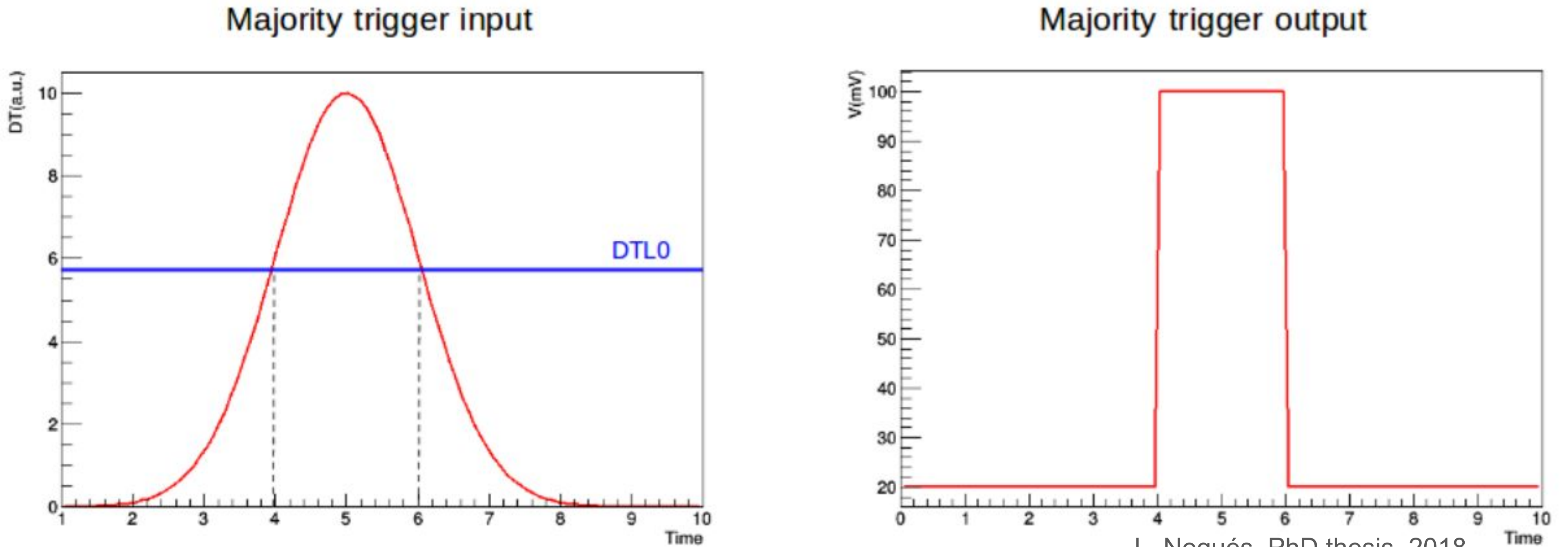


L. Nogués, PhD thesis, 2018

Trigger at pixel level

Trigger at individual pixel level, called “Level 0” or L0

- Signal is above or below a certain level
- If above, send (where?) a trigger signal → called majority trigger



L. Nogués, PhD thesis, 2018

Trigger at the level of “neighbouring pixels”

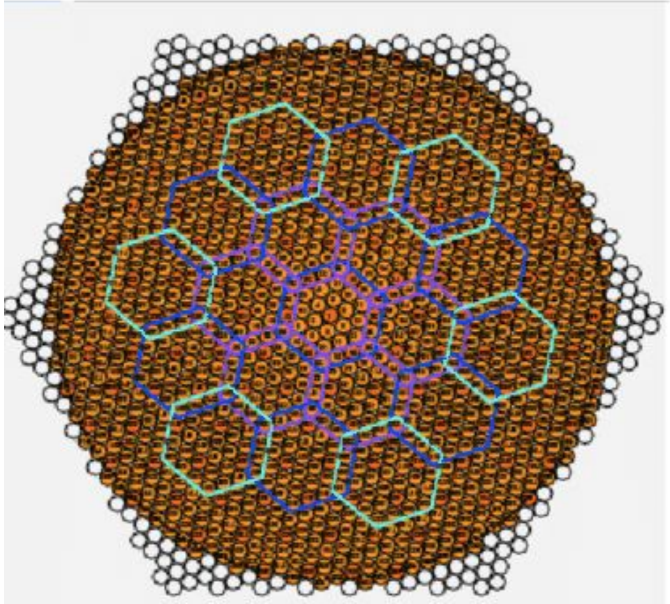
Trigger in group of neighbouring pixels, called “Level 1” or L1

Different strategies are possible:

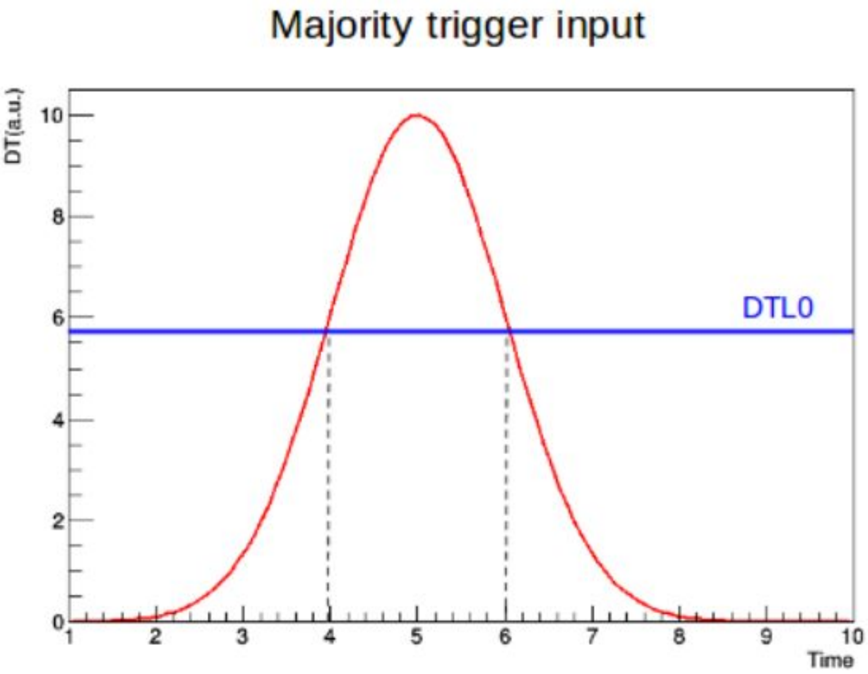
- Send a trigger if n pixels signals overlap
- Send a trigger if n close-compact (ie every pixel in the group is in contact with at least other two) next-neighbour (dubbed nNN) pixel signals overlap
- …

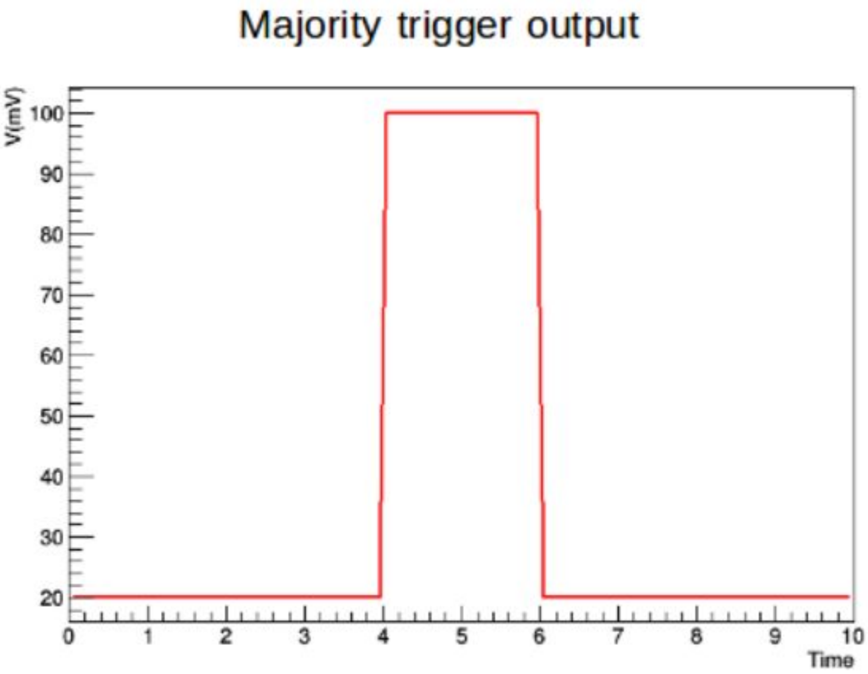
For instance, one of MAGIC’s trigger is 3NN

But then why the L1 trigger region looks like that? ——>



Can we do better than the majority trigger for low energies?





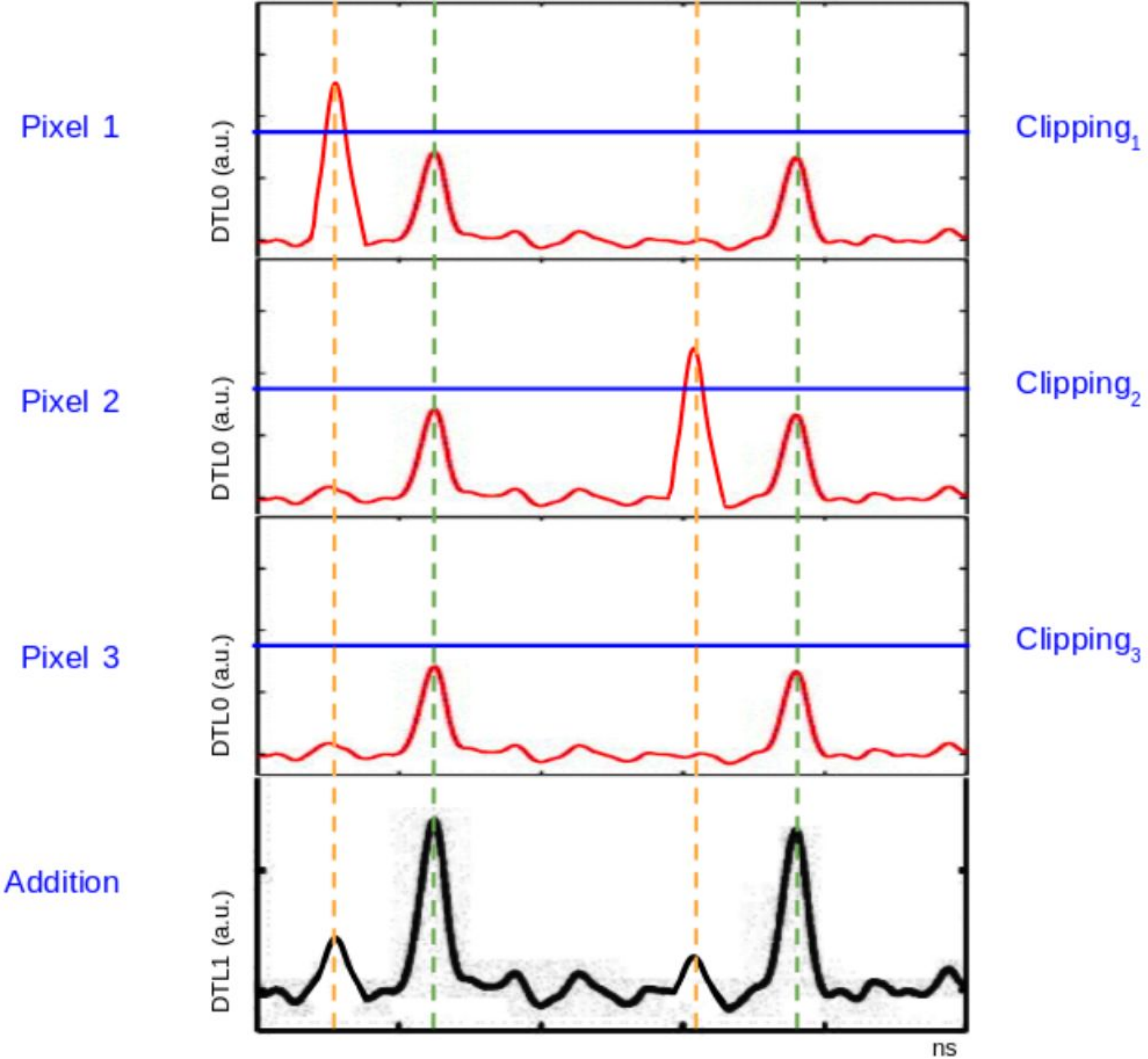
L. Nogués, PhD thesis, 2018

Majority trigger vs Sum trigger

- Majority trigger
  - The width of the L0 signal can the same for all pixels (e.g. MAGIC) or corresponding to the time over threshold (e.g. LST, not used)
  - The output voltage of the L0 signal is fixed to reduce impact of after-pulses from the PMTs
- Sum-trigger
  - Directly adds the analog signals coming from the pixels and sends the summed signal to the L1 trigger of the neighbouring modules, e.g. in LST it’s in groups of 21 pixels (3 modules) during regular operation (the value is configurable!)
  - May use an attenuator to equalize the amplitude of the signals from the different pixels, and a clipping circuit to reduce impact of after-pulses from the PMTs

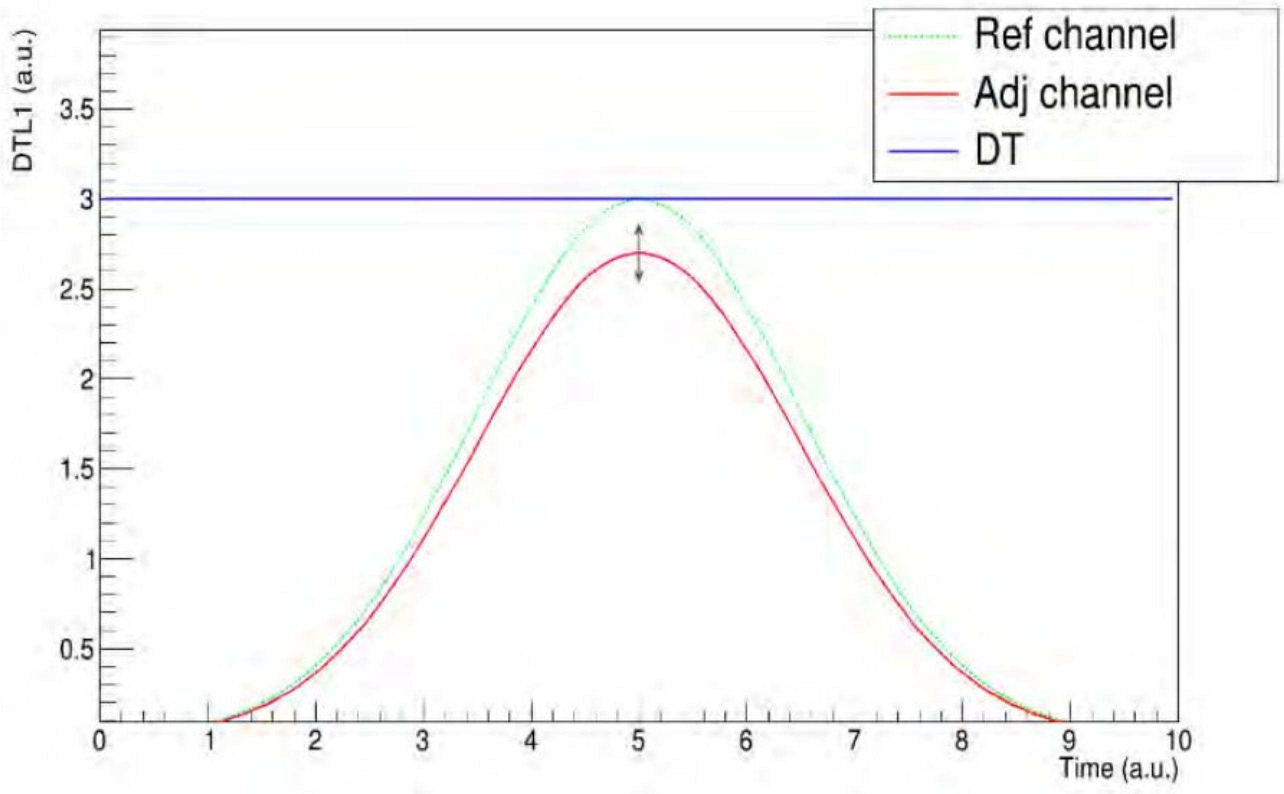
Sum trigger

An example:



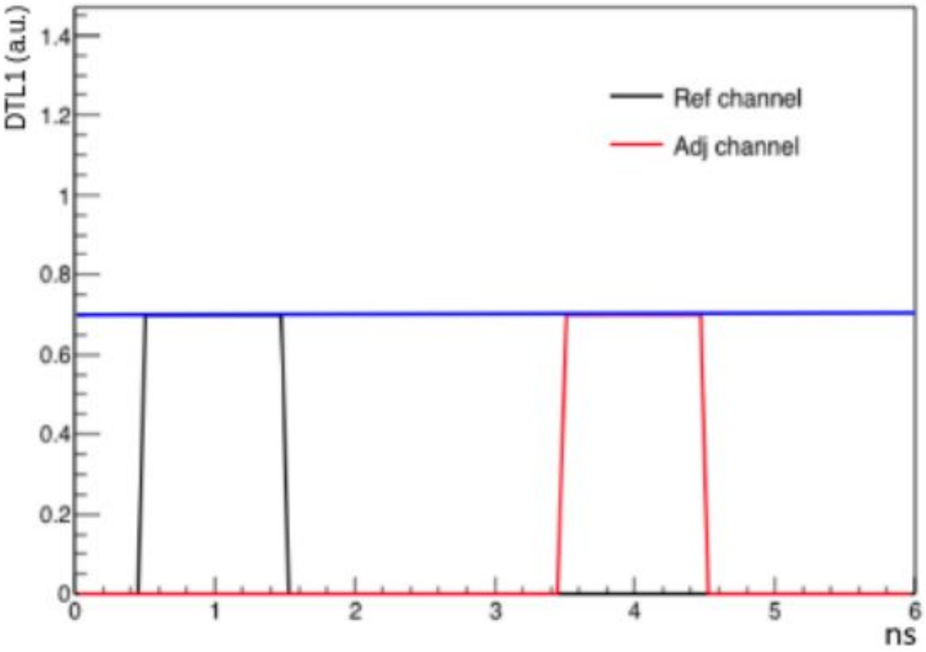
L. Nogués, PhD thesis, 2018

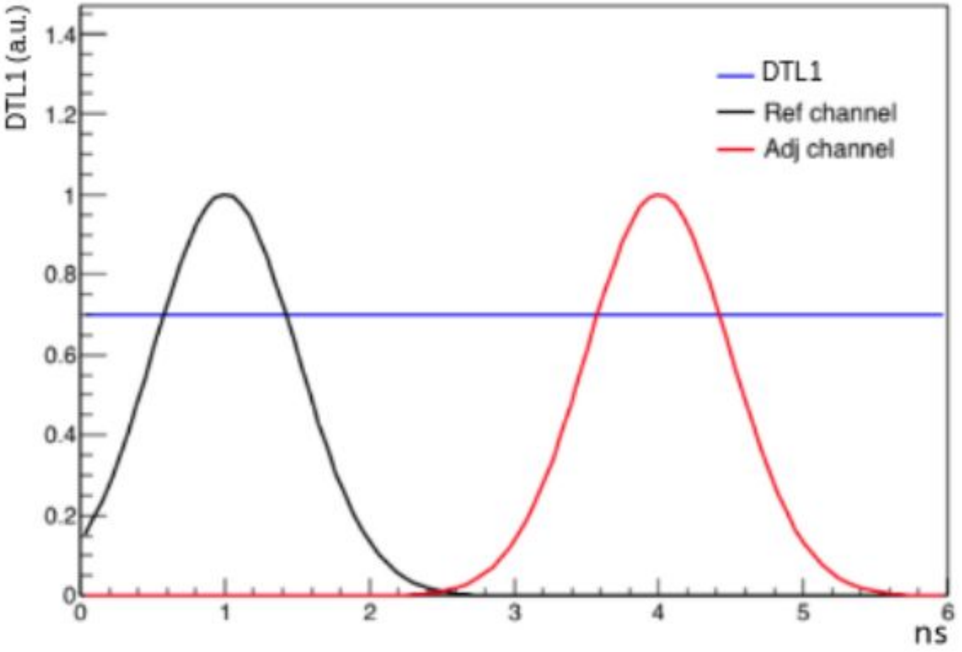
Calibration of the trigger signal in gain



L. Nogués, PhD thesis, 2018

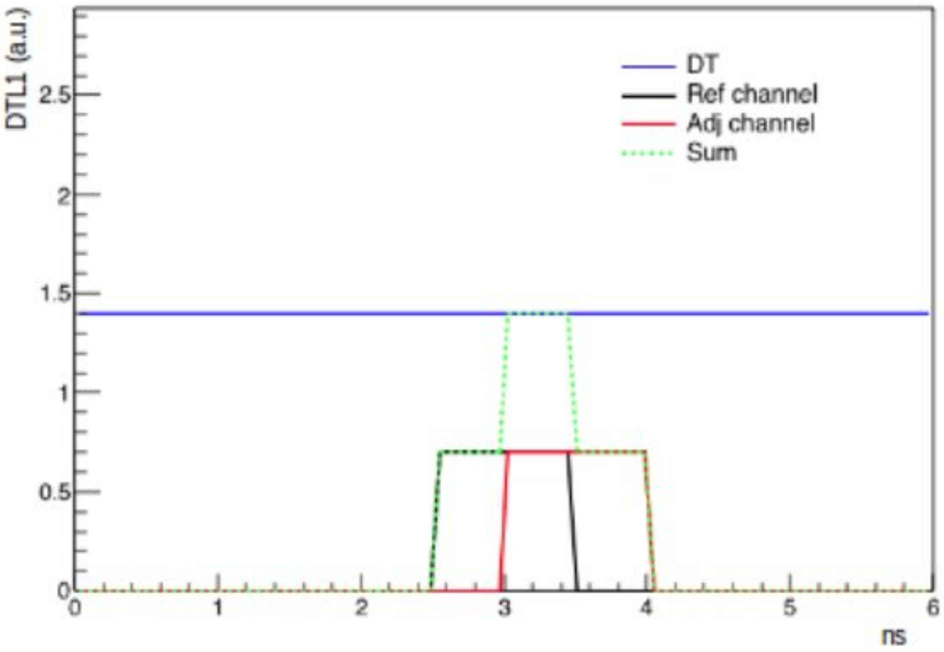
Calibration of the trigger signal in time

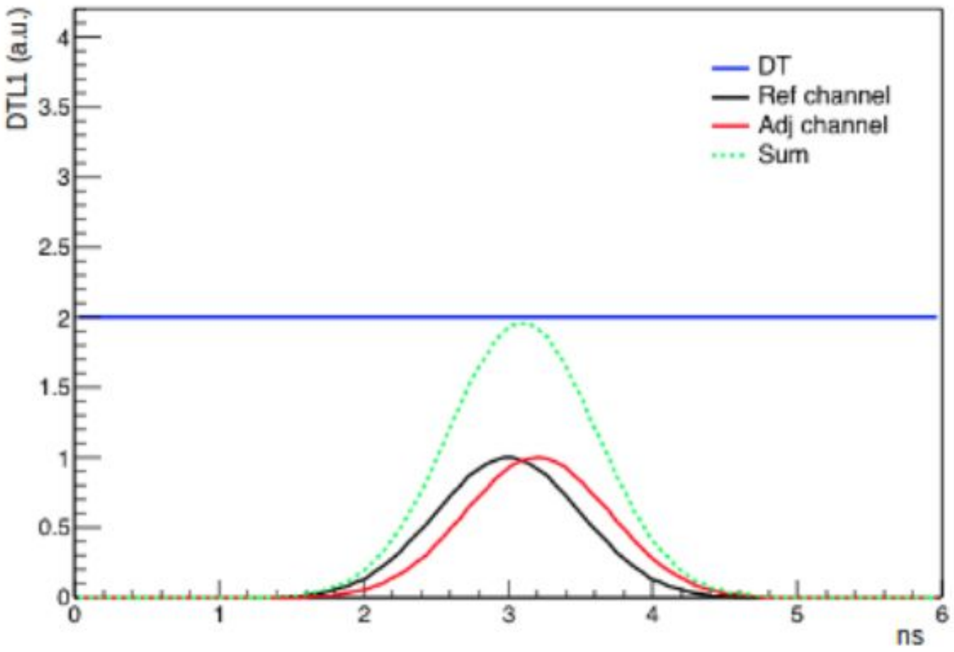




L. Nogués, PhD thesis, 2018

Calibration of the trigger signal in time

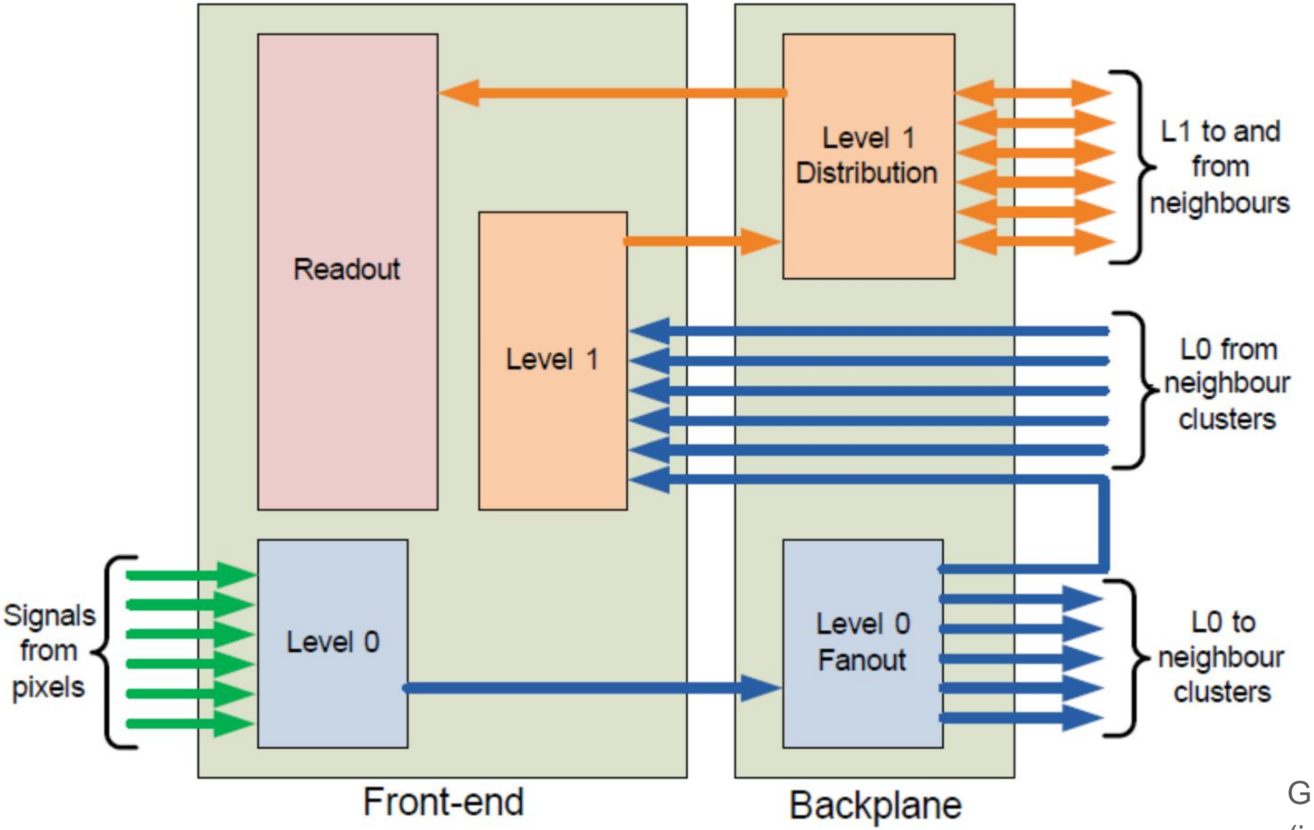




L. Nogués, PhD thesis, 2018

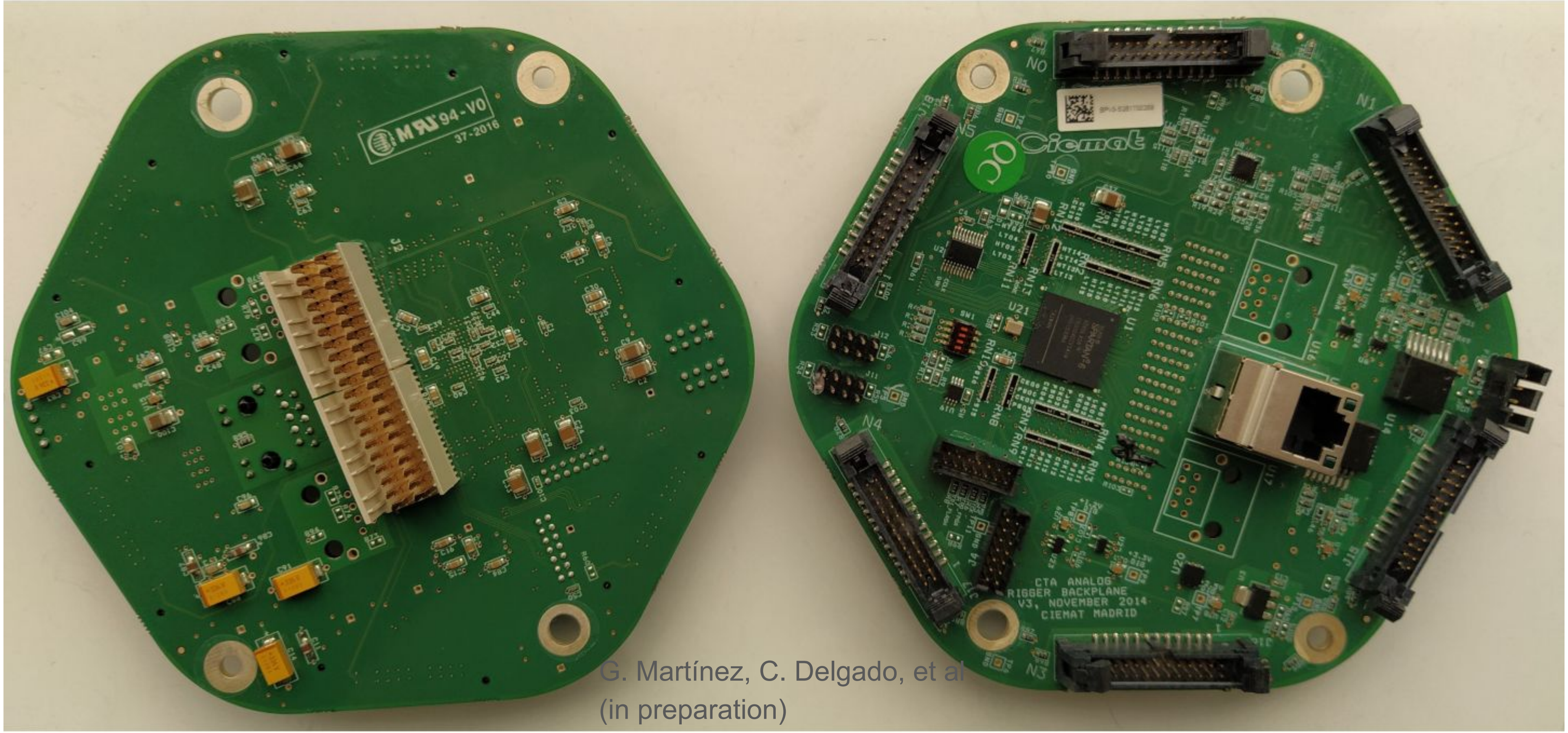
Trigger and clock distribution

Trigger and clock distribution



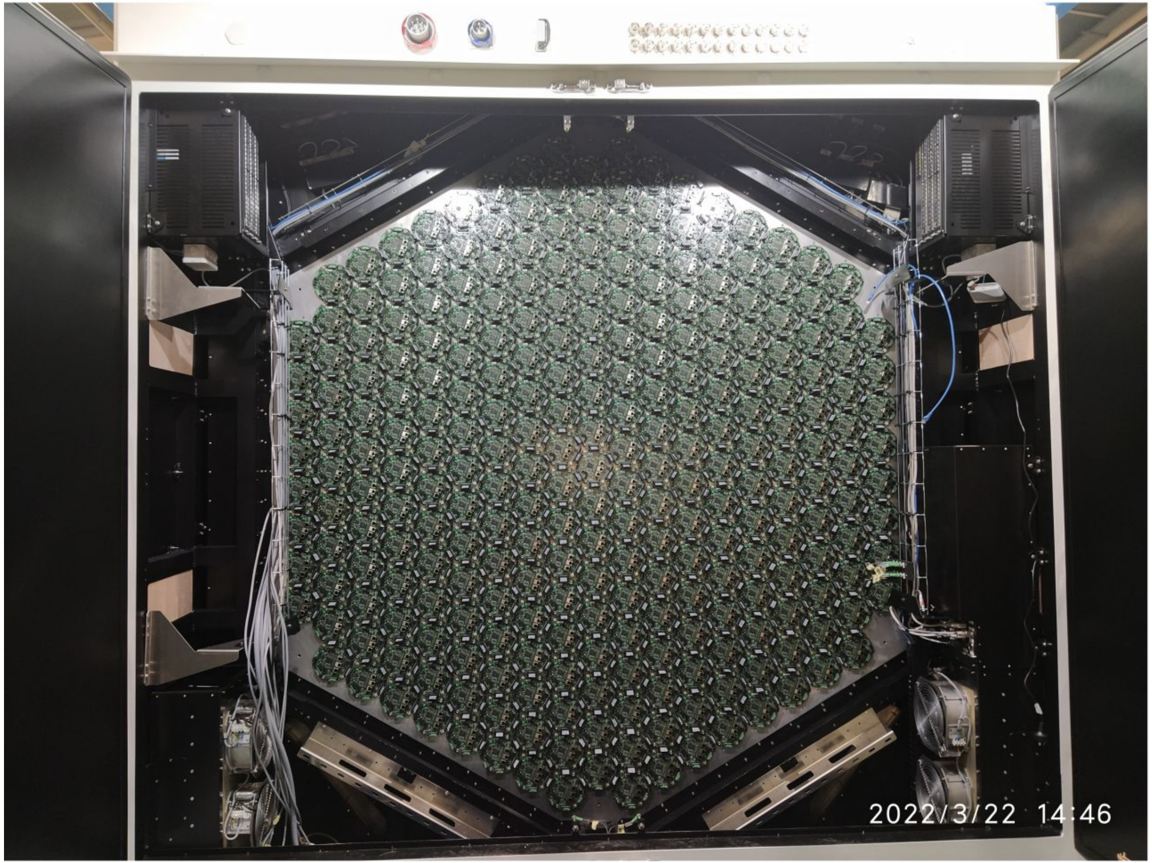
G. Martínez, C. Delgado, et al
(in preparation)

One LST backplane



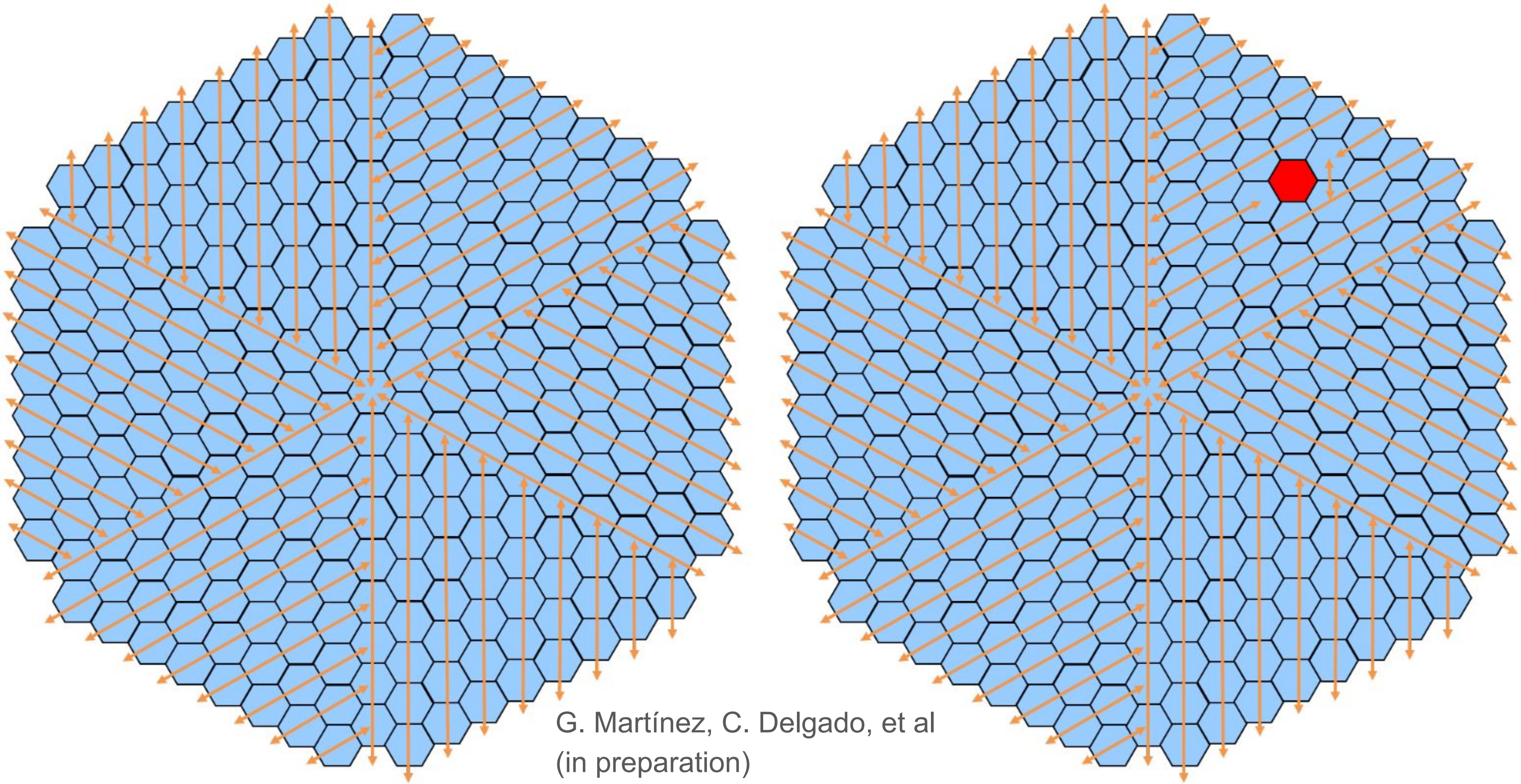
G. Martínez, C. Delgado, et al
(in preparation)

A LST camera fully equipped with backplanes



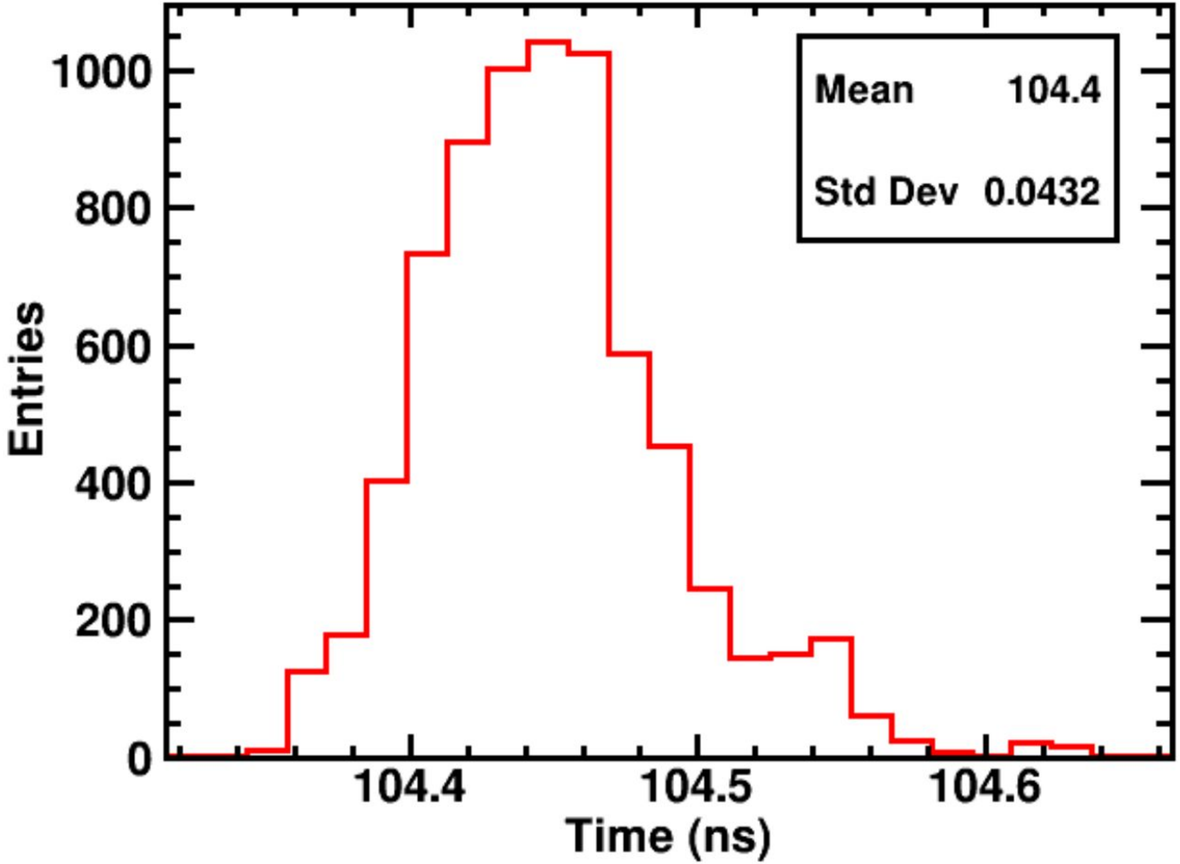
G. Martínez, C. Delgado, et al
(in preparation)

Trigger and clock distribution routes



G. Martínez, C. Delgado, et al
(in preparation)

Dispersion of the arrival time at backplane level after calibration



G. Martínez, C. Delgado, et al
(in preparation)

How do you delay a signal?

- Take the example of the delay calibrations, how do you think it works in reality? How to we delay a signal by X nanoseconds?

How do you delay a signal?

- Take the example of the delay calibrations, how do you think it works in reality? How to we delay a signal by X nanoseconds?

- Physical delay lines in the electronic boards
  → had to be incorporated in the camera design before production
  → the idea that you want/need such precision in timing had to be there very early, same for the realization that you will need to calibrate such delays

- Who writes the code that does such calibrations? Where does it live? How and when do you use it? How do you use its output?

Outline

1. A bit of history
2. Camera design
   a. what one can see (hardware)
   b. what one can not see (software)
3. The example of the LST cameras
   a. its challenge (reach the lowest energies)
   b. and how to overcome it
4. **Take-away message and conclusions**

What if I tell you I can make a camera…

- With 2% more quantum efficiency but it works only 50% of the time?

What if I tell you I can make a camera…

- With 2% more quantum efficiency but it works only 50% of the time?
- With 2% more quantum efficiency but it is available for operation 0.5% of the time less because it’s somehow more unstable?

What if I tell you I can make a camera…

- With 2% more quantum efficiency but it works only 50% of the time?
- With 2% more quantum efficiency but it is available for operation 0.5% of the time less because it’s somehow more unstable?
- Which can start a run 20 seconds faster than before?

What if I tell you I can make a camera…

- With 2% more quantum efficiency but it works only 50% of the time?
- With 2% more quantum efficiency but it is available for operation 0.5% of the time less because it’s somehow more unstable?
- Which can start a run 20 seconds faster than before?
    - 20 seconds x 2 runs per hour x 1500 hours of observations per year = 16,7 hours per year
    - 16,7 x 30 years of operation = 500 hours in total

What if I tell you I can make a camera…

- With 2% more quantum efficiency but it works only 50% of the time?
- With 2% more quantum efficiency but it is available for operation 0.5% of the time less because it’s somehow more unstable?
- Which can start a run 20 seconds faster than before?
  - 20 seconds x 2 runs per hour x 1500 hours of observations per year = 16,7 hours per year
  - 16,7 x 30 years of operation = 500 hours in total
- Which can start a run 20 seconds faster than before but is available for operation 0.5% of the time less because it’s somehow more unstable?

Take-away message

- Building a camera is a (very) long process that involves designing, proto-typing, manufacturing, testing, integrating…
  E.g 10-15 years for all CTAO cameras

- The scientific objectives drive design choices… but can't overcome "physical" limitations (space, accessibility, heat dispersion, affordability…)

- There are many design possibles!

- It's a collaborative effort: you need to talk to your colleagues otherwise it won't succeed

(My) conclusion

- I hope some things are more clear now than 1h30 ago
- I also hope you have more questions now than 1h30 ago ;-)
  → let’s answer to them in front of a real camera or during operations :-)
  (or during any break of course, or by email in 1 month or…)
- I hope you are convince of why, as physicist, you should care about hardware/software and why you are needed to get the best possible instrument
- I hope in the next years you find ways to get involved in designing, proto-typing, manufacturing, testing, integrating…
  → It’s really fun and it will make you better physicists!