UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

Corryvreckan

The Maelstrom for Your Test Beam Data

8th BTTB Workshop

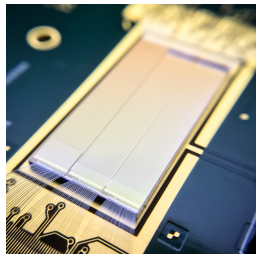
Tbilisi State University, January 28th, 2020

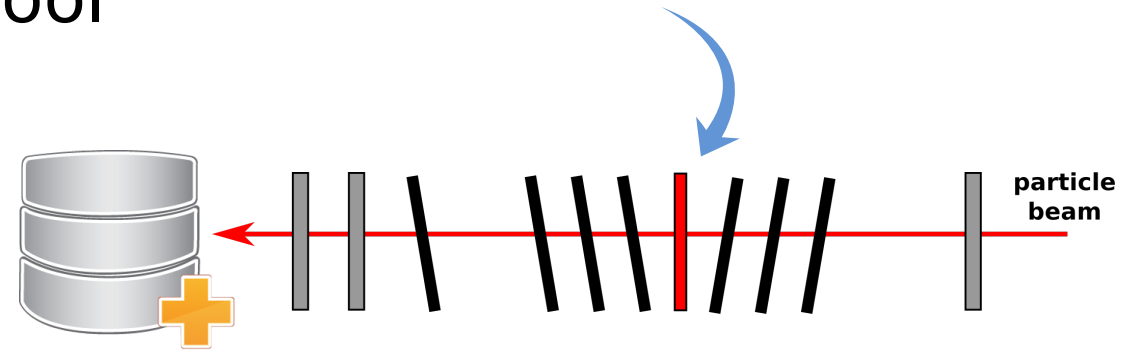
Jens Kröger, Heidelberg University & CERN

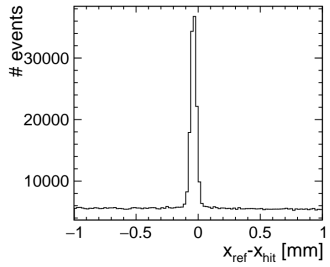
on behalf of the CLICdp collaboration and the Corryvreckan Developers

Corryvreckan

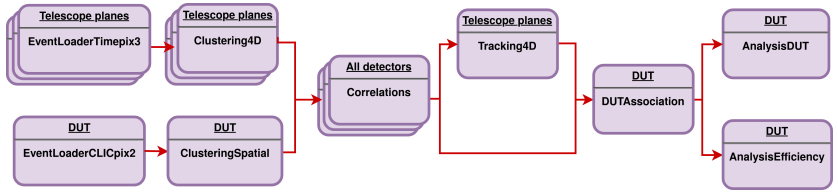
A **reconstruction and analysis** tool for pixel sensor test beam data







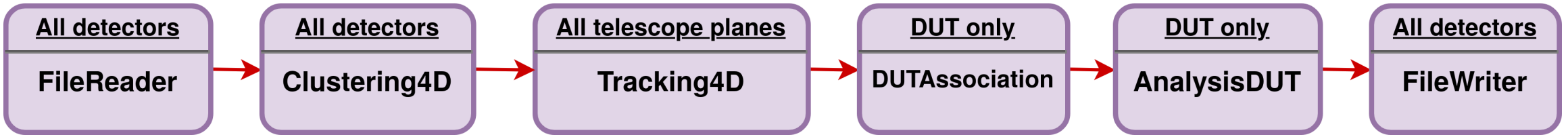
- modular structure
  - framework core
  - modules for specific tasks
- highly flexible and configurable
- easy to understand
  - written in modern C++
  - comprehensive documentation (> 100 pages!)



The Modular Approach

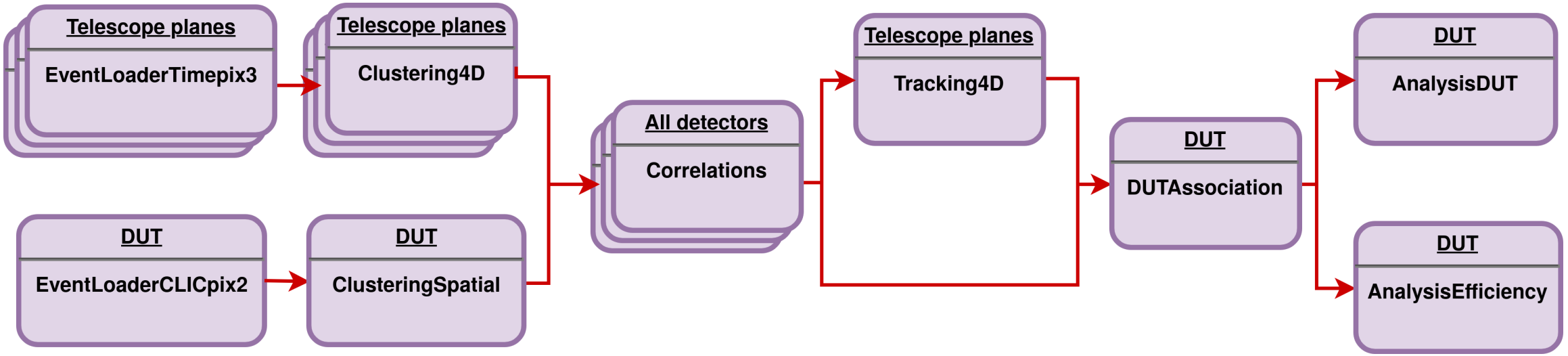
- separation between
  - framework core
  - implementation of algorithms → **`[Modules]`**
- modules:
  ”plug-and-play” algorithms for specific tasks
- objects are stored temporarily:
  - events, pixels, clusters, tracks

- select suitable modules for
  - event building
  - clustering
  - tracking
  - analysis (also multiple DUTs)
  - ...
- quick to set up and easy to configure



The Modular Approach

- can create more complex reconstruction chains
- apply different modules to different devices in the same reconstruction



Configuring Corryvreckan

- TOML style = easy to read
- support of physical units (e.g. 25um)
- need 2 files:
  - **configuration** file: analysis parameters
  - **geometry** file: detector description

example configuration

jens: ~/software/corryvreckan/testing

```
File Edit Options Buffers Tools Conf Help
[Corryvreckan]
log_level = "WARNING"
log_format = "DEFAULT"

detectors_file = "geometries/geometry_timepix3_telescope.conf"
histogram_file = "histograms_run9444.root"

[Metronome]
event_length = 200us

[EventLoaderTimepix3]
input_directory = "data/timepix3tel_ebeam120"

[Clustering4D]
time_cut_abs = 100ns

[Tracking4D]
min_hits_on_track = 6
momentum=120GeV
time_cut_abs = 200ns
track_model = "gbl"
spatial_cut_abs = 200um, 200um
volume_scattering_length=304m

-UU-:----F1  test_timepix3tel_ebeam120_gbl.conf
For information about GNU Emacs and the GNU sys
.
```

example geometry

jens: ~/software/corryvreckan/testing

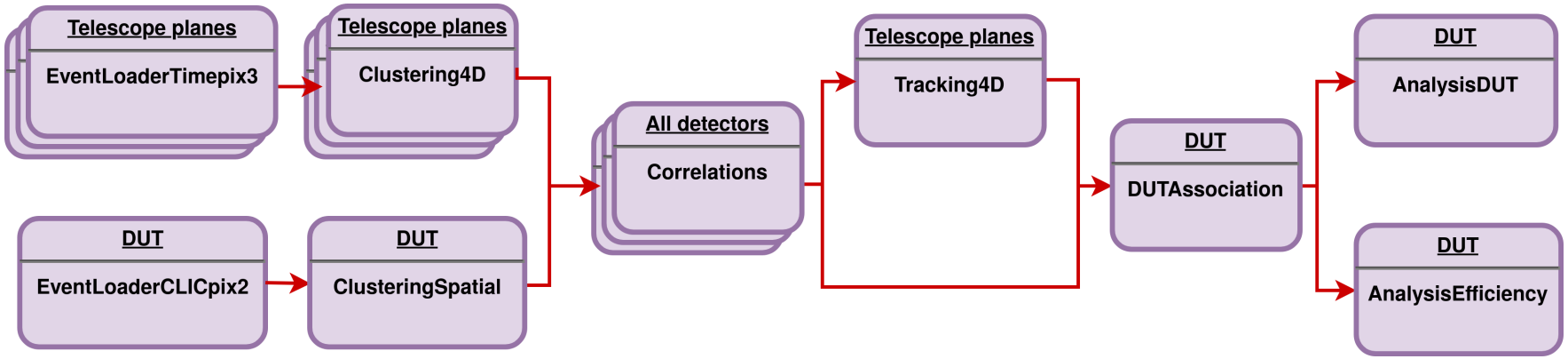
```
File Edit Options Buffers Tools Conf Help
[W0013_D04]
number_of_pixels = 256,256
orientation = 10.7471deg,186.437deg,-1.33797deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 923.402um,296.86um,0
spatial_resolution = 4um,4um
material_budget = 0.01068
time_resolution = 20ns
type = "Timepix3"

[W0013_E03]
number_of_pixels = 256,256
orientation = 11.0172deg,186.658deg,-1.06937deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = -249.154um,408.592um,21.5mm
spatial_resolution = 4um,4um
material_budget = 0.01068
time_resolution = 20ns
type = "Timepix3"

-UU-:----F1  geometry_timepix3_telescope.conf   Top L1    Git
For information about GNU Emacs and the GNU system, type C-h\
 C-a.
```

Flexible Event Building

- event building:
  - arrange data from different devices in **“time slices”** (events) for reconstruction/analysis

- flexible: can combine devices with different readout schemes
  - frame-based, data-driven, triggered, ...

- full analysis chain on event-by-event basis

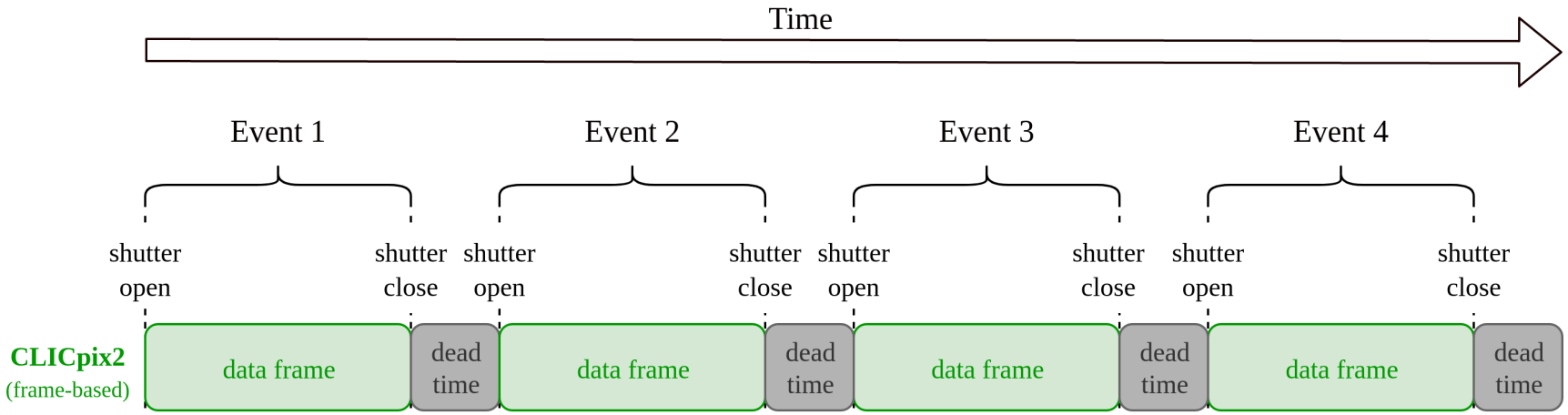


Examples – Flexible Event Building

The 1st module defines the event!

- **frame-based**:
  - frame start and end define an event
  - *example*: CLICpix2 → shutter open/close

```
[EventLoaderCLICpix2]
input_directory = "path/to/data_clicpix2"
```

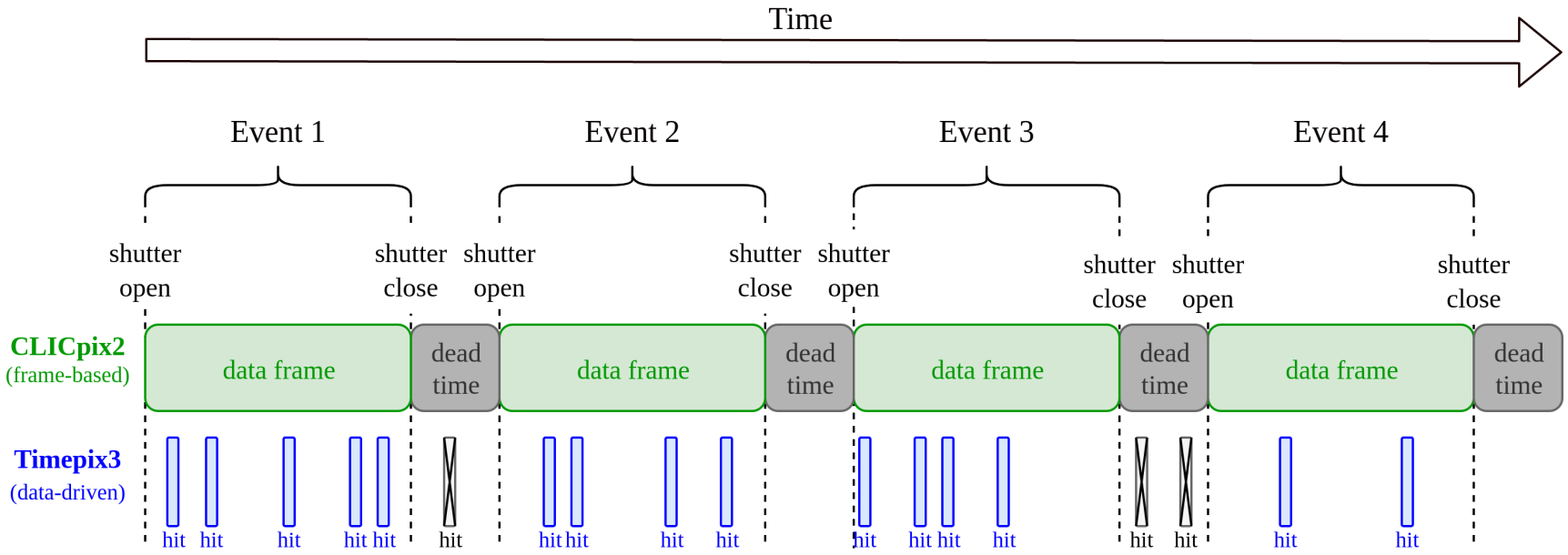


Examples – Flexible Event Building

The 1st module defines the event!

- **frame-based**:
  - frame start and end define an event
  - data from data-driven and/or triggered devices are added if within event
  - *example*: CLICpix2 + Timepix3 telescope

```
[EventLoaderCLICpix2]
input_directory = "path/to/data_clicpix2"

[EventLoaderTimepix3]
input_directory = "path/to/data_timepix3"
```

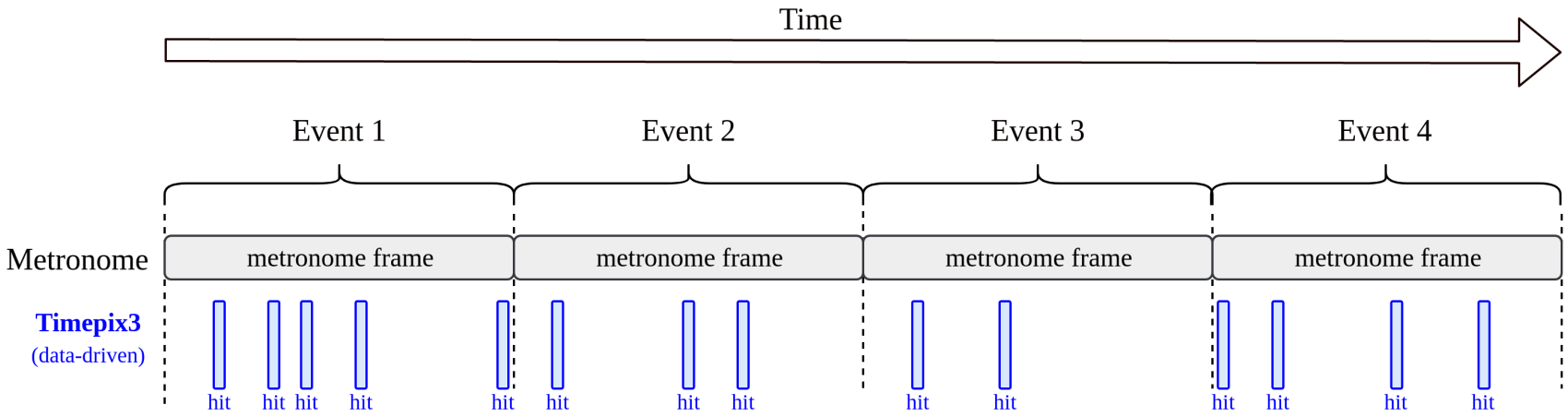


Examples – Flexible Event Building

The 1st module defines the event!

- **data-driven:**
  - **[Metronome]** modules defines events with configurable length
  - all hits within this “time slice” are added to the event
  - *example*: Timepix3 telescope

```
[Metronome]
event_length = 20us

[EventLoaderTimepix3]
input_directory = “path/to/data_timepix3”
```

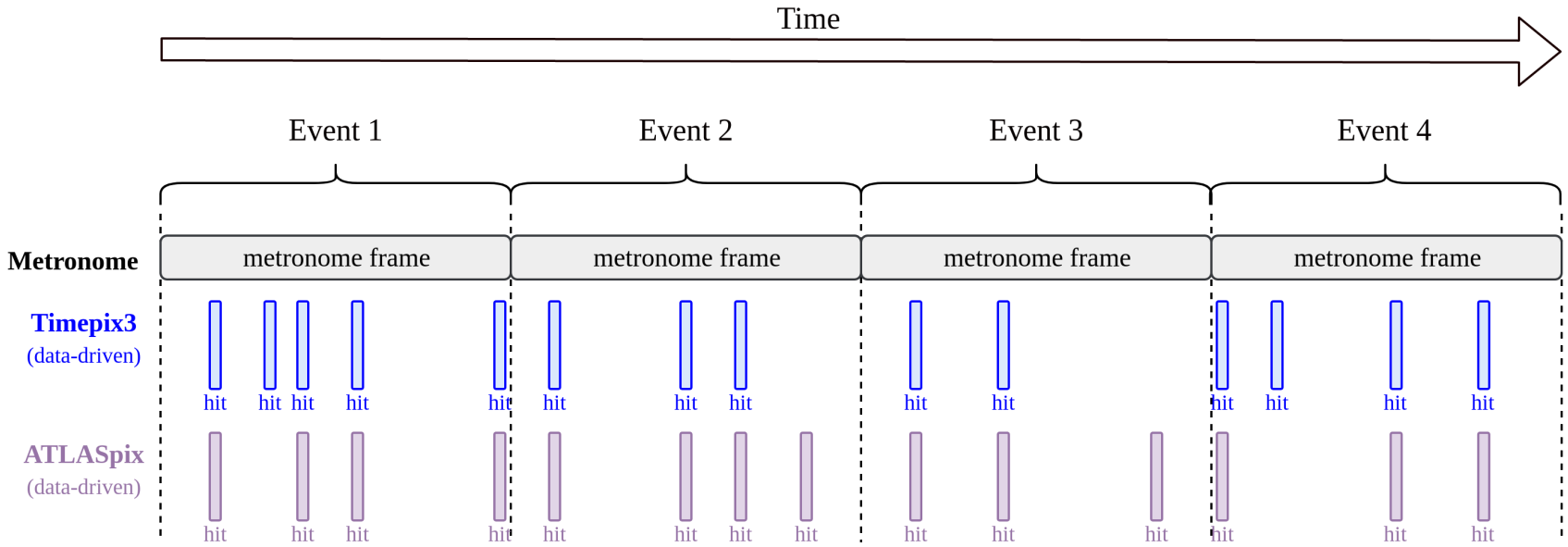


Examples – Flexible Event Building

The 1st module defines the event!

- **data-driven:**
  - [Metronome] modules defines events with configurable length
  - all hits within this “time slice” are added to the event
  - *example*: Timepix3 telescope + ATLASpix

```
[Metronome]
event_length = 20us

[EventLoaderTimepix3]
input_directory = “path/to/data_timepix3”

[EventLoaderATLASpix]
input_directory = “path/to/data_atlaspix”
```

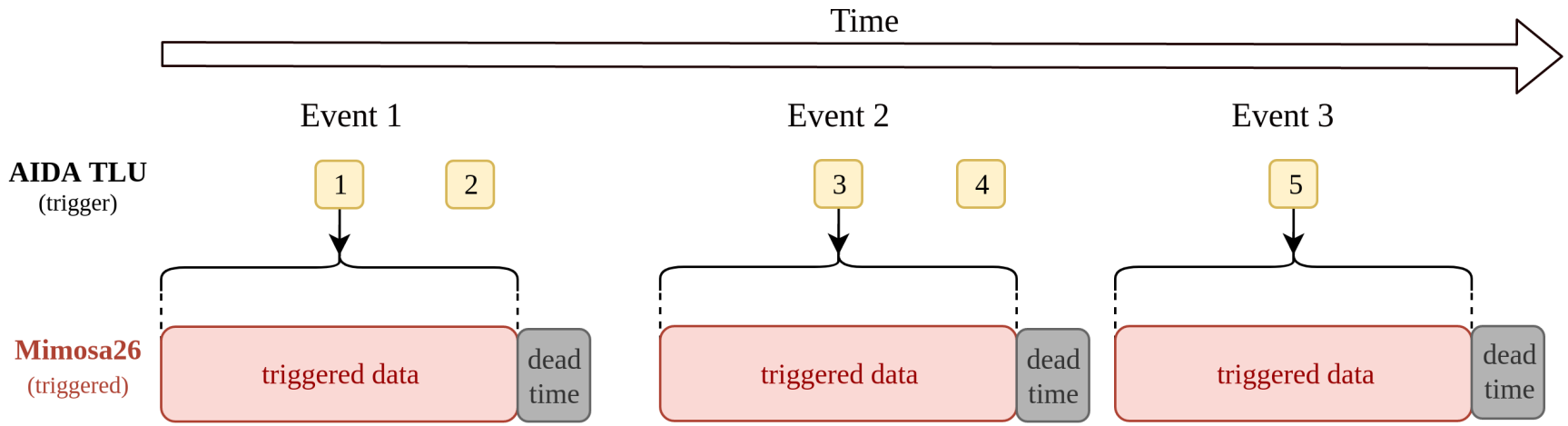


Examples – Flexible Event Building

- **combined (1)**
  - trigger defines event with configurable length
  - data from triggered devices are added based on trigger ID
  - *example*: AIDA TLU + Mimosa26 telescope

The 1st module defines the event!

```
[EventLoaderEUDAQ2]
name = "TLU_0"
adjust_event_times =
    ["TluRawDataEvent", -115us, +230us]]

[EventLoaderEUDAQ2]
type = "MIMOSA26"
```

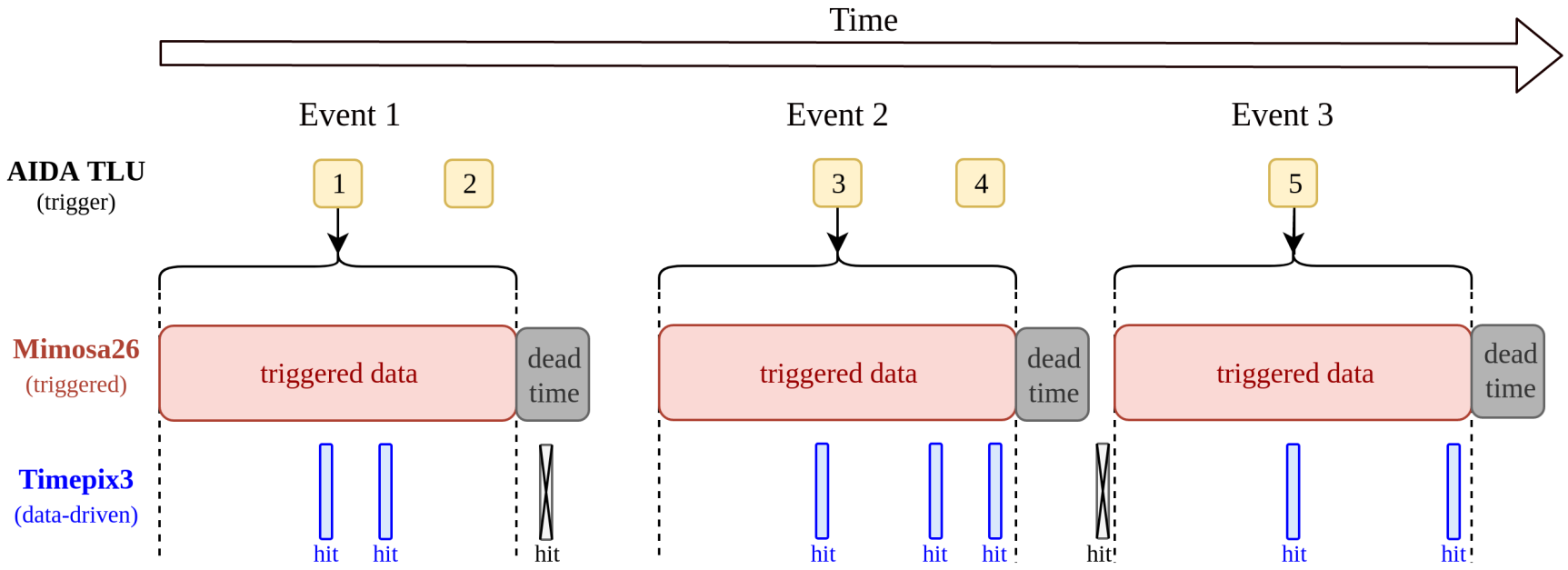


Examples – Flexible Event Building

- **combined (2)**
  - trigger defines event with configurable length
  - data from triggered devices are added based on trigger ID
  - data from data-driven devices are added if inside event
  - *example*: AIDA TLU + Mimosa26 telescope + Timepix3

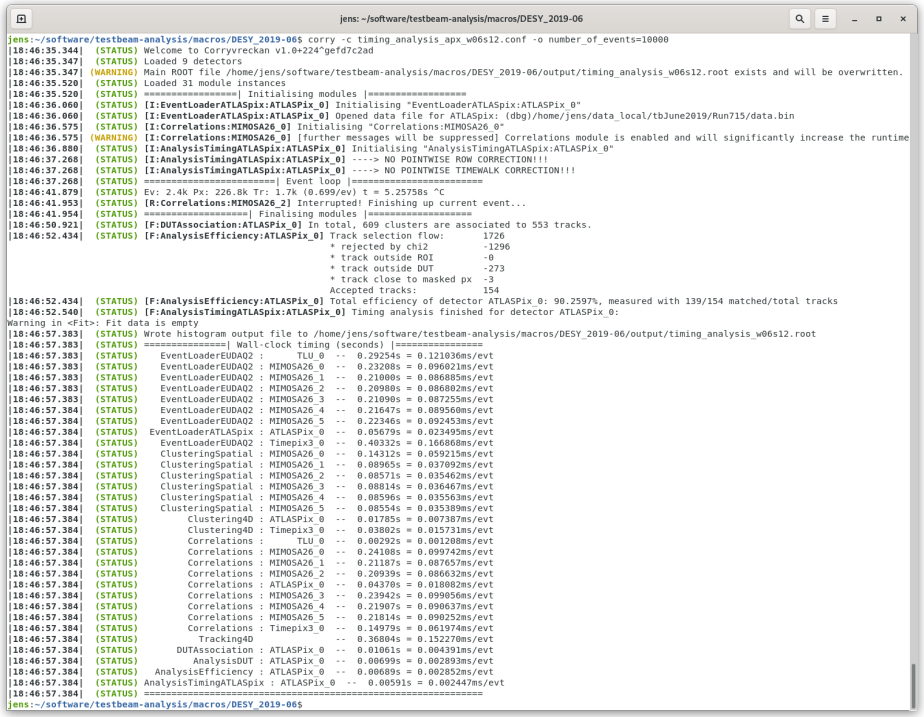
The 1st module defines the event!

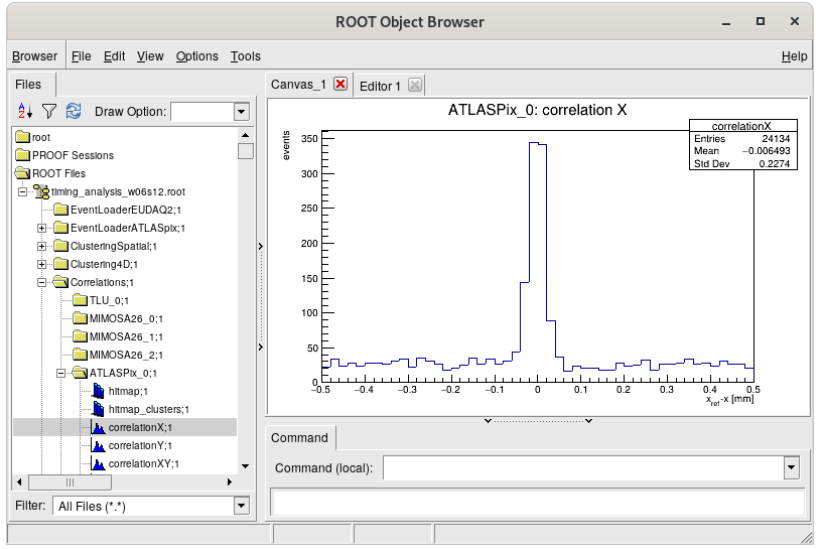
```
[EventLoaderEUDAQ2]
name = "TLU_0"
adjust_event_times =
    ["TluRawDataEvent", -115us, +230us]]

[EventLoaderEUDAQ2]
type = "MIMOSA26"

[EventLoaderEUDAQ2]
name = "Timepix3_0"
```



The Output

- terminal output:
  - configurable verbosity
  - descriptive logging/warning/errors
- output files:
  - ROOT files of Corryvreckan objects (events, pixels, clusters, tracks)
  - ROOT file of analysis histograms from each module used
- Online Monitoring
  - view histograms while analysis is running
  - continuously updated



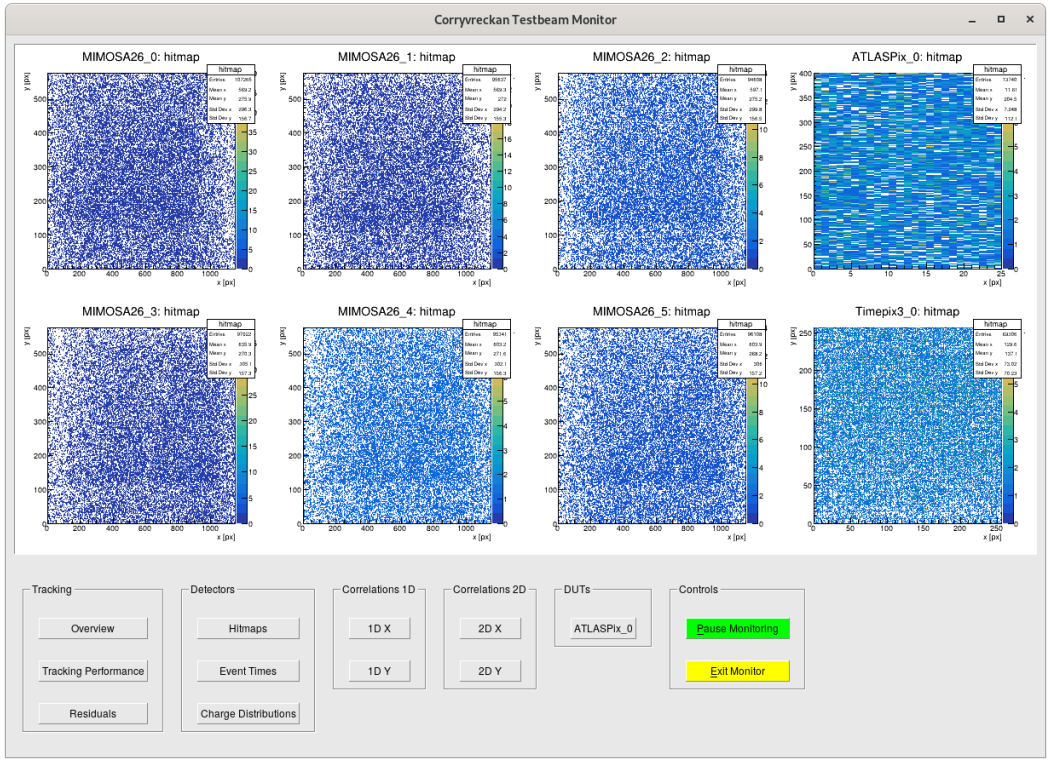


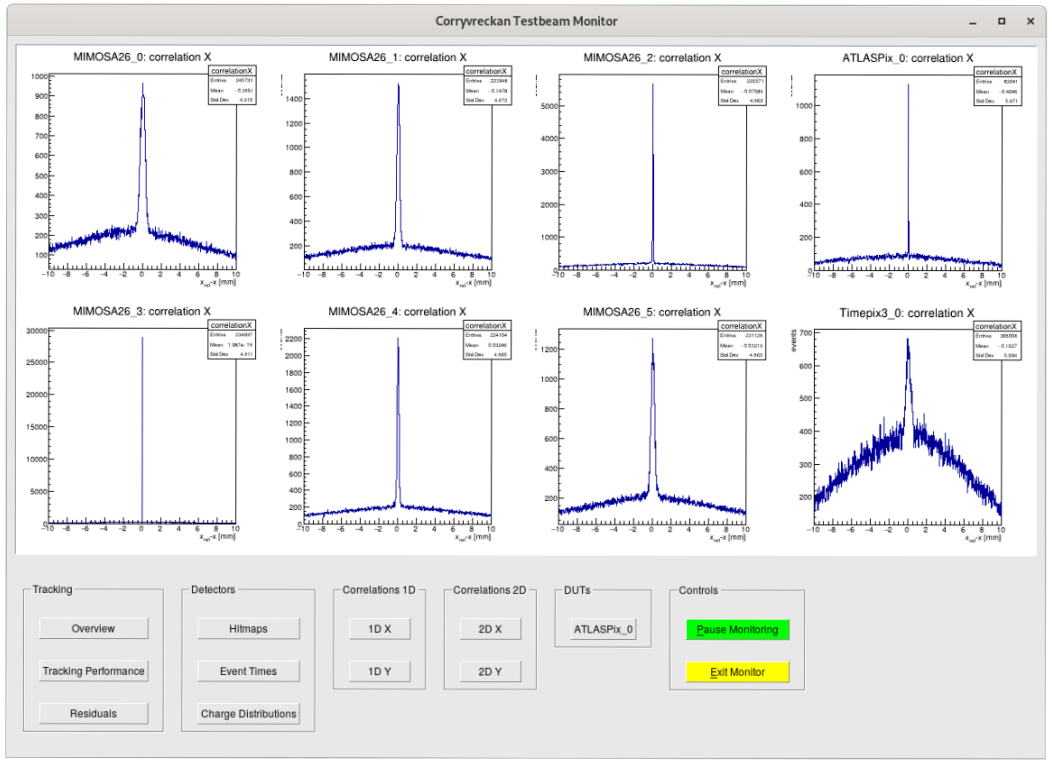
The Online Monitor

- display histograms while reconstructing (and even **while taking data**):
  - any histogram from current analysis can be chosen in configuration

```
[OnlineMonitor]
update = 200
dut_plots = [["EventLoaderEUDAQ2/%DUT%/hitmap", "colz"],
             ["EventLoaderEUDAQ2/%DUT%/hPixelTimes"]]
```

- useful for data quality checks

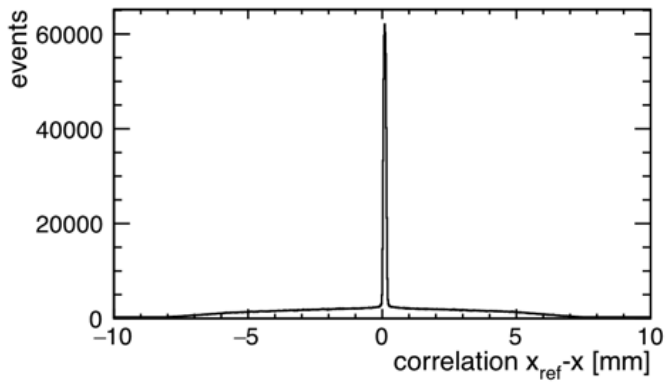




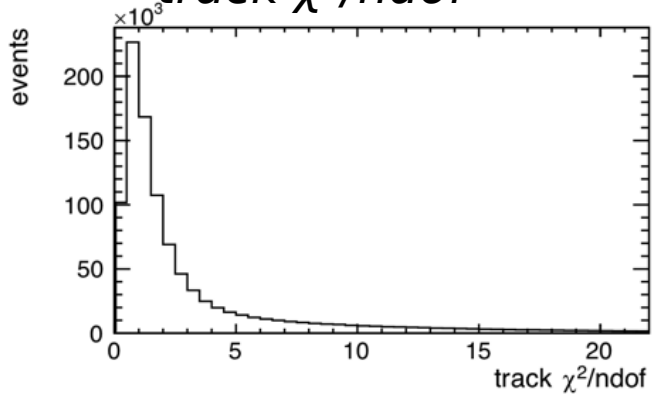
Some Example Results

- DUT analysis plots:
  - ATLASpix in Timepix3 telescope at CERN SPS

1D correlation in X

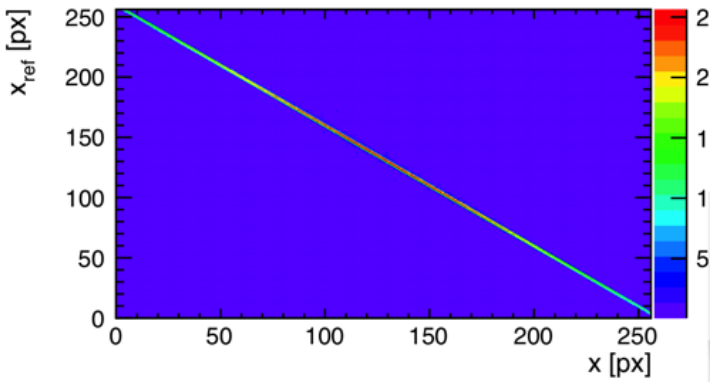


track χ^2/ndof

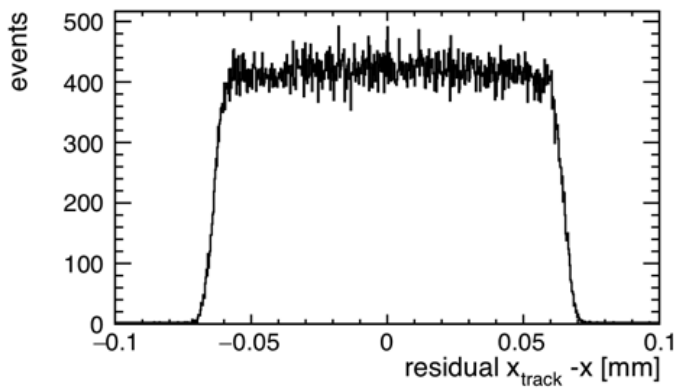


- data converters implemented for:
  - ATLASpix
  - CLICpix/CLICpix2
  - Timepix1/Timepix3

2D correlation in X



track-DUT residual in X

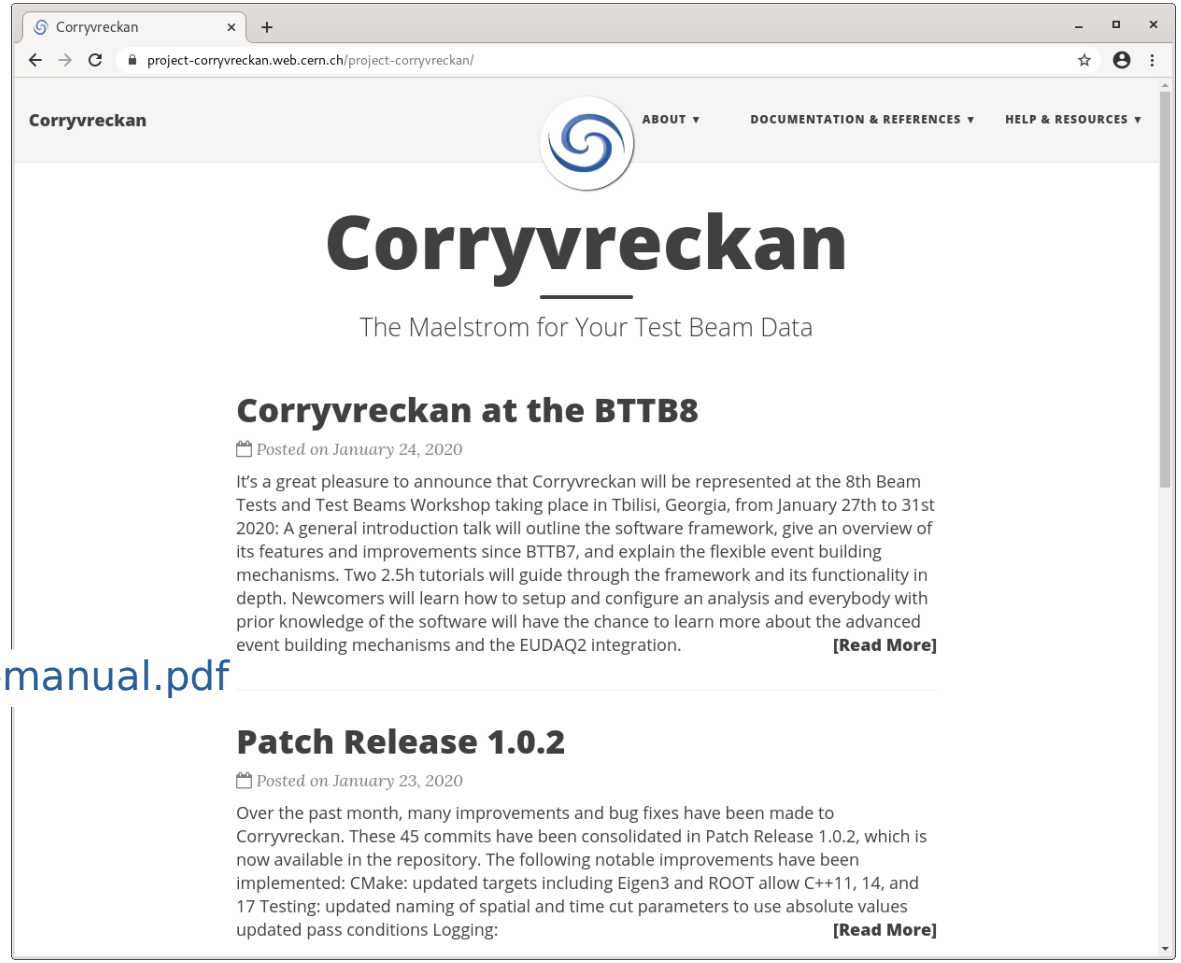


  - **EUDAQ2:** AIDA TLU, Mimosa26 telescope, ATLASpix, CLICpix2, CLICTD, Timepix3

What’s new since BTTB7?

Official Release 1.0 last December

- new project website: https://cern.ch/corryvreckan
  - News on releases
  - Installation/Getting Started
  - Links to code repository, issue tracker, forum etc.
- completed/extended documentation
  - published user manual as CLICdp Note https://cern.ch/corryvreckan/usermanual/corryvreckan-manual.pdf
- many notable features – see next slides



Notable Features (1)

- 4D Tracking
  - spatial and time cuts to associate clusters to tracks, improves track quality
  - absolute and relative cuts (derived from spatial/time resolution)

- various alignment methods
  - track χ^2, track-dut residual, Millepede

- EUDAQ2 integration https://github.com/eudaq/eudaq/
  - include **AIDA TLU** as auxiliary device
  - process data recorded with **EUDAQ2 DAQ:**
    AIDA TLU, Mimosa26 telescope, Timepix3, ATLASpix, CLICpix2, CLICTD

Notable Features (2)

- GBL Track Reconstruction (preliminary)
  - general-broken-line fitting takes multiple Coulomb scattering in the detector planes into account → improved track quality for low-momentum beams like at DESY

- job submission tool
  - automated processing of several data files and/or scans of reconstruction parameters
  - runs locally or in batch mode for HTCondor on lxplus/naf/...

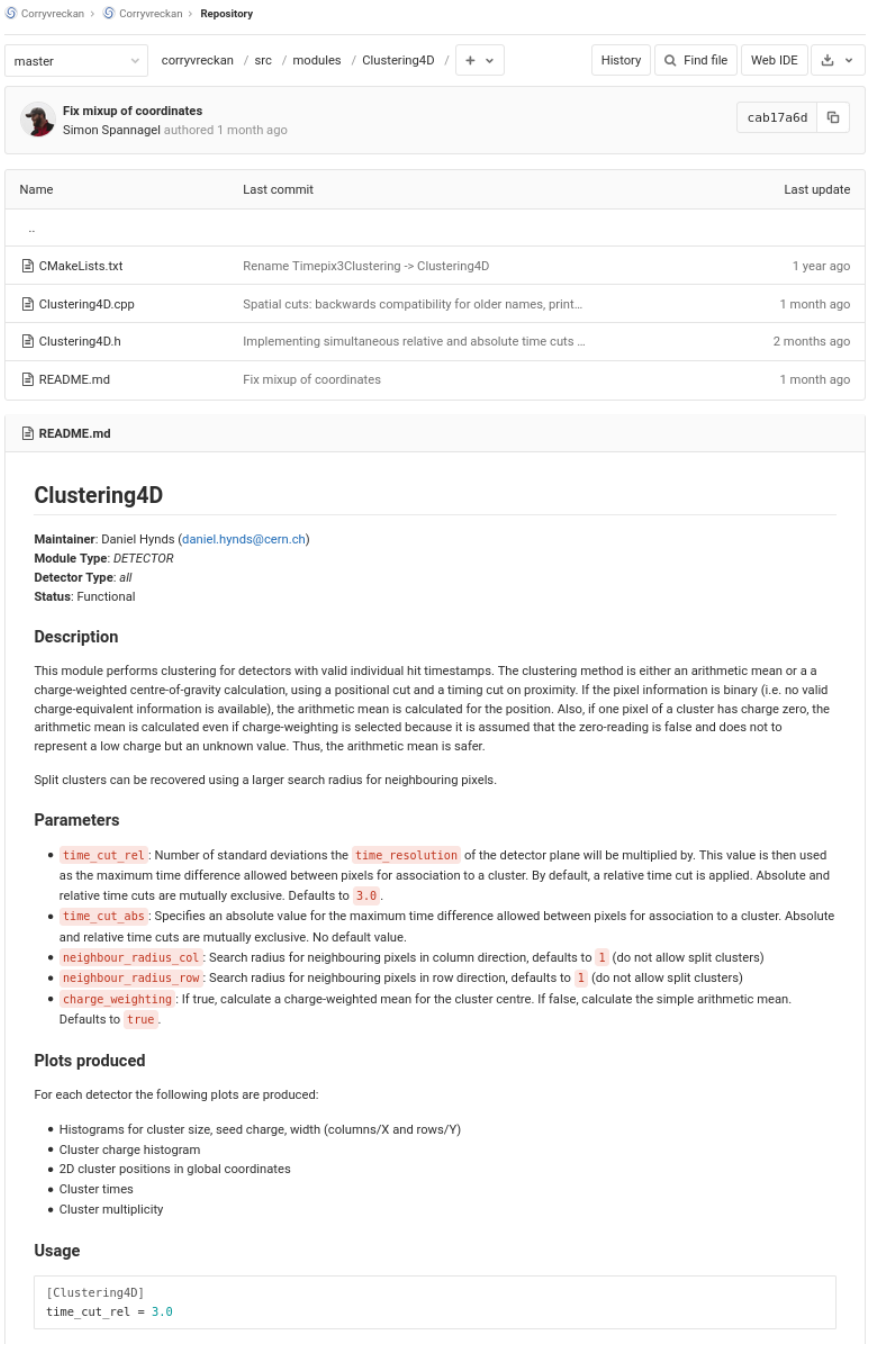
- read in data from **Allpix Squared**
  https://cern.ch/allpix-squared
  - simulate test beam setup in Allpix Squared
  - analyse with same reconstruction framework as real data

Documentation

- **online documentation** in repo
  https://gitlab.cern.ch/corryvreckan/corryvreckan
  - every modules has a README

- extensive user manual
  https://cern.ch/corryvreckan/usermanual/corryvreckan-manual.pdf
  - full description of framework
  - installation instructions
  - “Getting started”, FAQs
  - module descriptions (fetched from repo)
  - published as CLICdp note:
    https://cds.cern.ch/record/2703012

- Doxygen code reference
  https://cern.ch/corryvreckan/reference/
  - more details on code

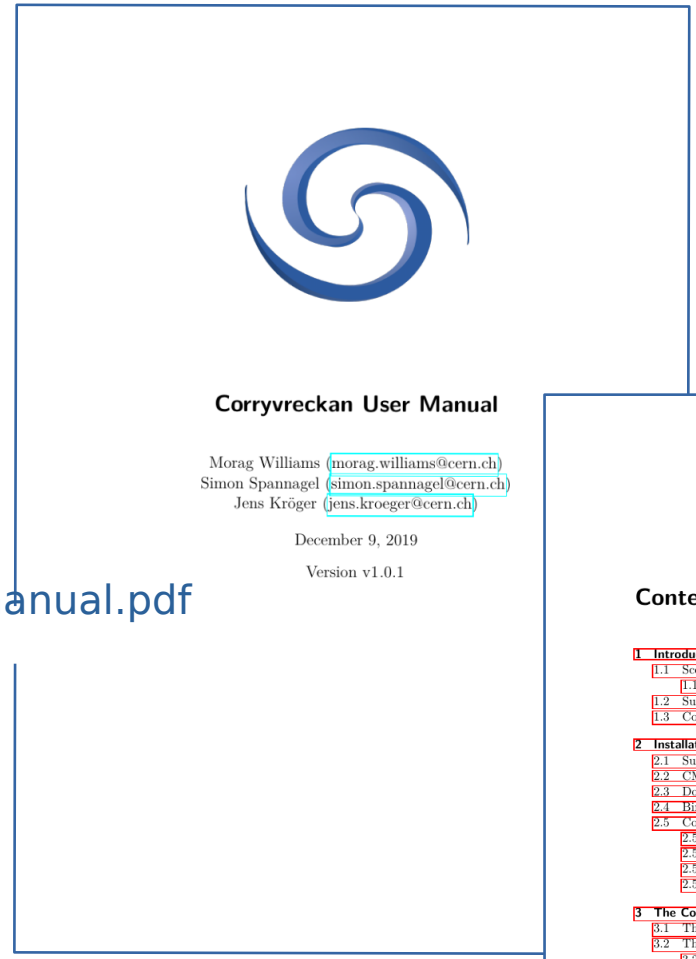


Documentation

- online documentation in repo
  https://gitlab.cern.ch/corryvreckan/corryvreckan
  - every modules has a README

- **extensive user manual**
  https://cern.ch/corryvreckan/usermanual/corryvreckan-manual.pdf
  - full description of framework
  - installation instructions
  - “Getting started”, FAQs
  - module descriptions (fetched from repo)
  - published as CLICdp note:
    https://cds.cern.ch/record/2703012

- Doxygen code reference
  https://cern.ch/corryvreckan/reference/
  - more details on code

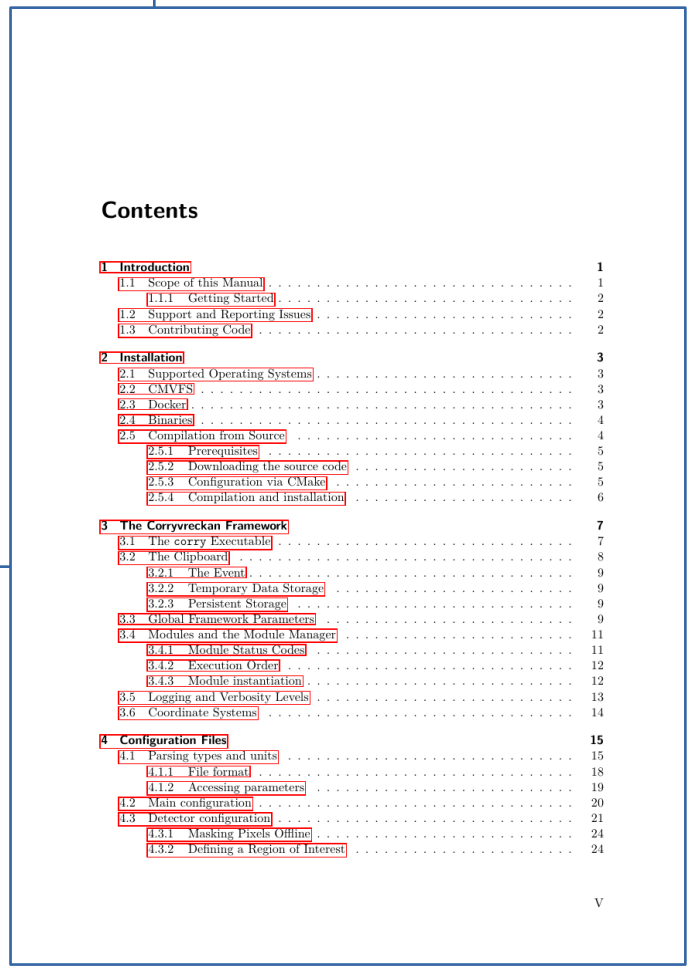


Corryvreckan User Manual

Morag Williams (morag.williams@cern.ch)
Simon Spannagel (simon.spannagel@cern.ch)
Jens Kröger (jens.kroeger@cern.ch)

December 9, 2019

Version v1.0.1



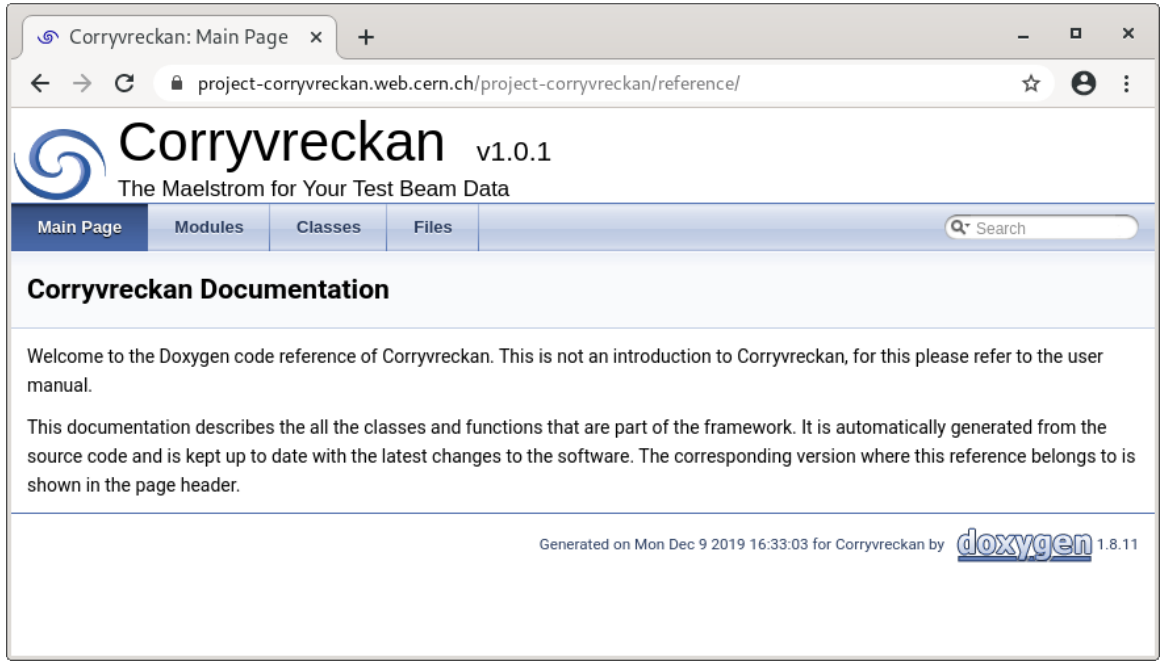
Contents

Documentation

- online documentation in repo
  https://gitlab.cern.ch/corryvreckan/corryvreckan
  - every modules has a README

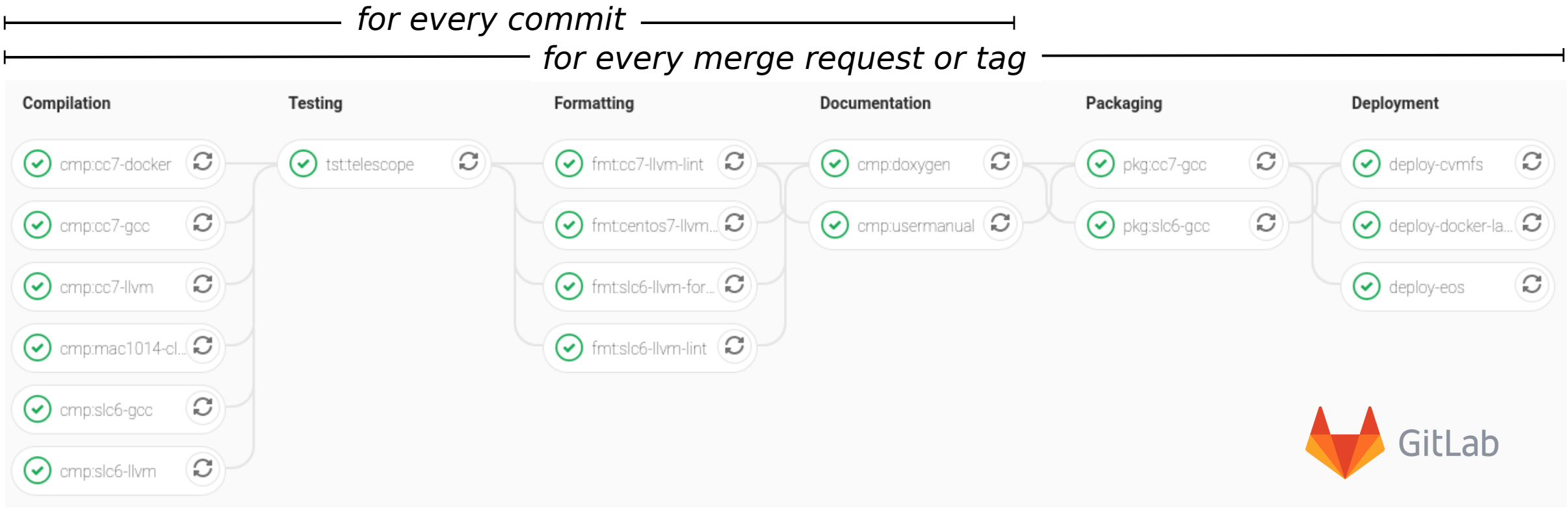
- extensive user manual
  https://cern.ch/corryvreckan/usermanual/corryvreckan-manual.pdf
  - full description of framework
  - installation instructions
  - “Getting started”, FAQs
  - module descriptions (fetched from repo)
  - published as CLICdp note:
    https://cds.cern.ch/record/2703012

- **Doxygen code reference**
  https://cern.ch/corryvreckan/reference/
  - more details on code



GitLab Continuous Integration

- ensures compilation, formatting, functionality (all stages explained in backup)
- pipeline runs through for every commit



Summary – Corryvreckan

reconstruction and analysis tool
for pixel sensor test beam data

- “plug-and-play” modules
- highly flexible/configurable
  → many different event building options
- comprehensive documentation
- growing number of users
- contributions welcome

Learn more:


Join my tutorials:
https://indico.cern.ch/event/813822/contributions/3663099/


Visit our website:
https://cern.ch/corryvreckan


Check out the repository:
https://gitlab.cern.ch/corryvreckan/corryvreckan


Discuss in the forum:
https://corryvreckan-forum.web.cern.ch/


Contact us:
corryvreckan.info@cern.ch

Backup

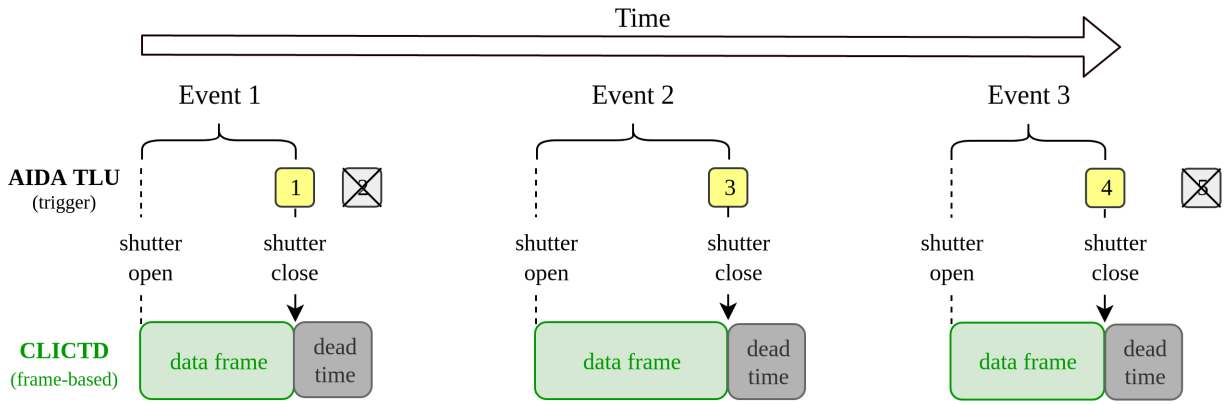
in case there are some questions...

More Examples – Flexible Event Building

- **combined**
  - trigger used to close shutter
  - *example*:
    AIDA TLU + CLICTD

The 1st module defines the event!

```
[EventLoaderEUDAQ2]
name = "CLICTD_0"

[EventLoaderEUDAQ2]
name = "TLU_0"
```

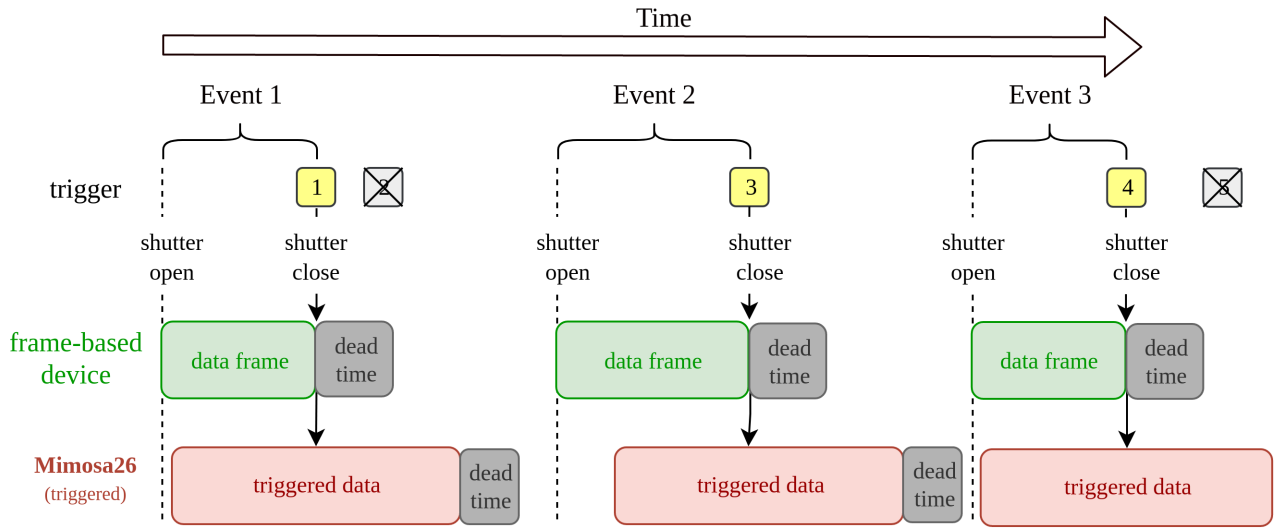


More Examples – Flexible Event Building

- **combined**
  - trigger used to close shutter
  - data from triggered devices are added based on trigger ID
  - *example*:
    AIDA TLU + CLICTD + Mimosa26 telescope

The 1st module defines the event!

```
[EventLoaderEUDAQ2]
name = "CLICTD_0"

[EventLoaderEUDAQ2]
name = "TLU_0"

[EventLoaderEUDAQ2]
type = "MIMOSA26"
```

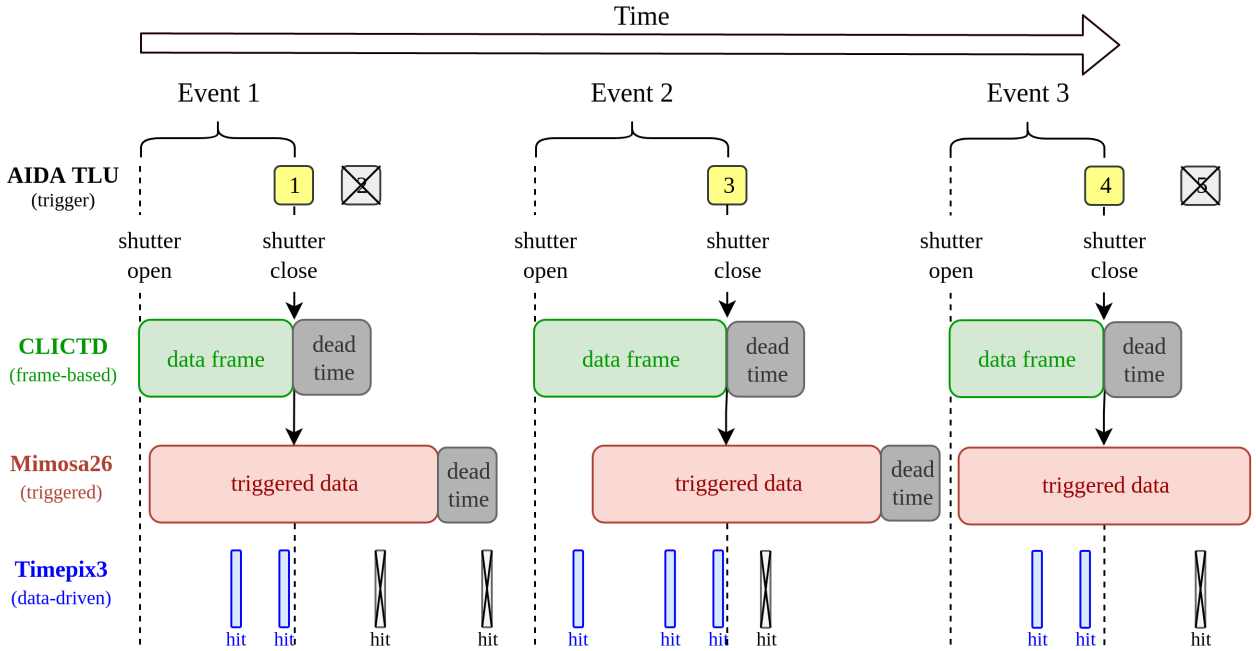


More Examples – Flexible Event Building

- **combined**
  - trigger used to close shutter
  - data from triggered devices are added based on trigger ID
  - data from data-driven devices are added if inside event
  - *example*:
    AIDA TLU + CLICTD
    + Mimosa26 telescope + Timepix3

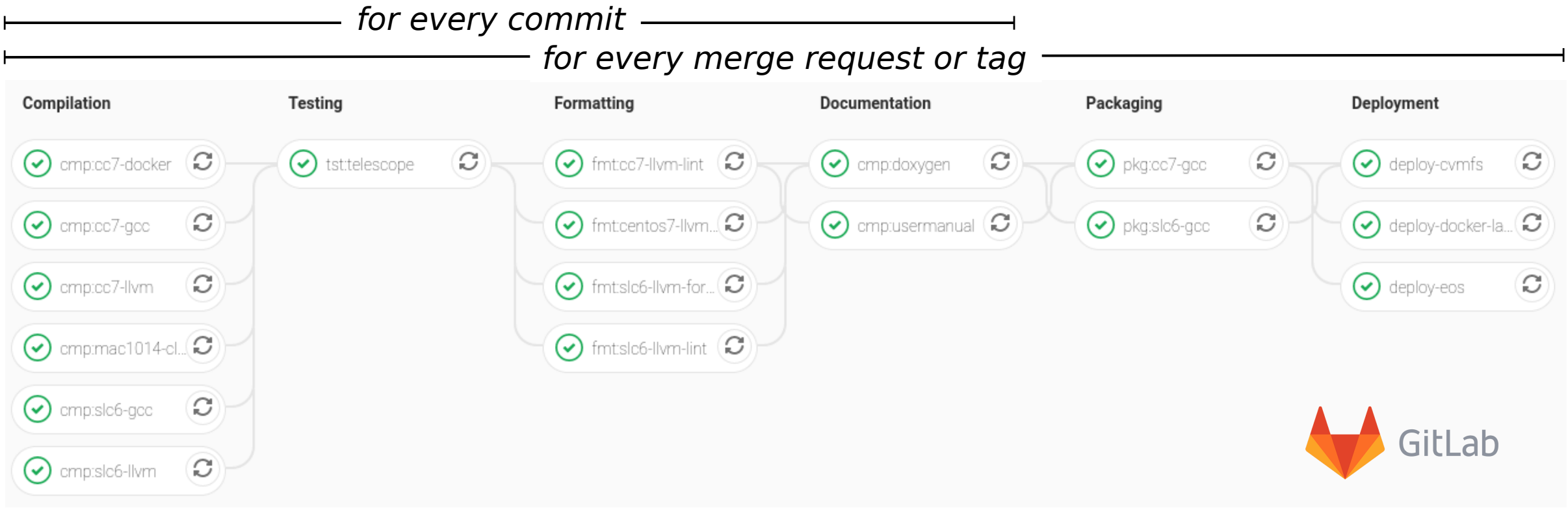
The 1st module defines the event!

```
[EventLoaderEUDAQ2]
name = "CLICTD_0"

[EventLoaderEUDAQ2]
name = "TLU_0"

[EventLoaderEUDAQ2]
type = "MIMOSA26"

[EventLoaderEUDAQ2]
name = "Timepix3_0"
```



GitLab Continuous Integration

- ensures compilation, formatting, functionality (all stages explained in backup)
- pipeline runs through for every commit

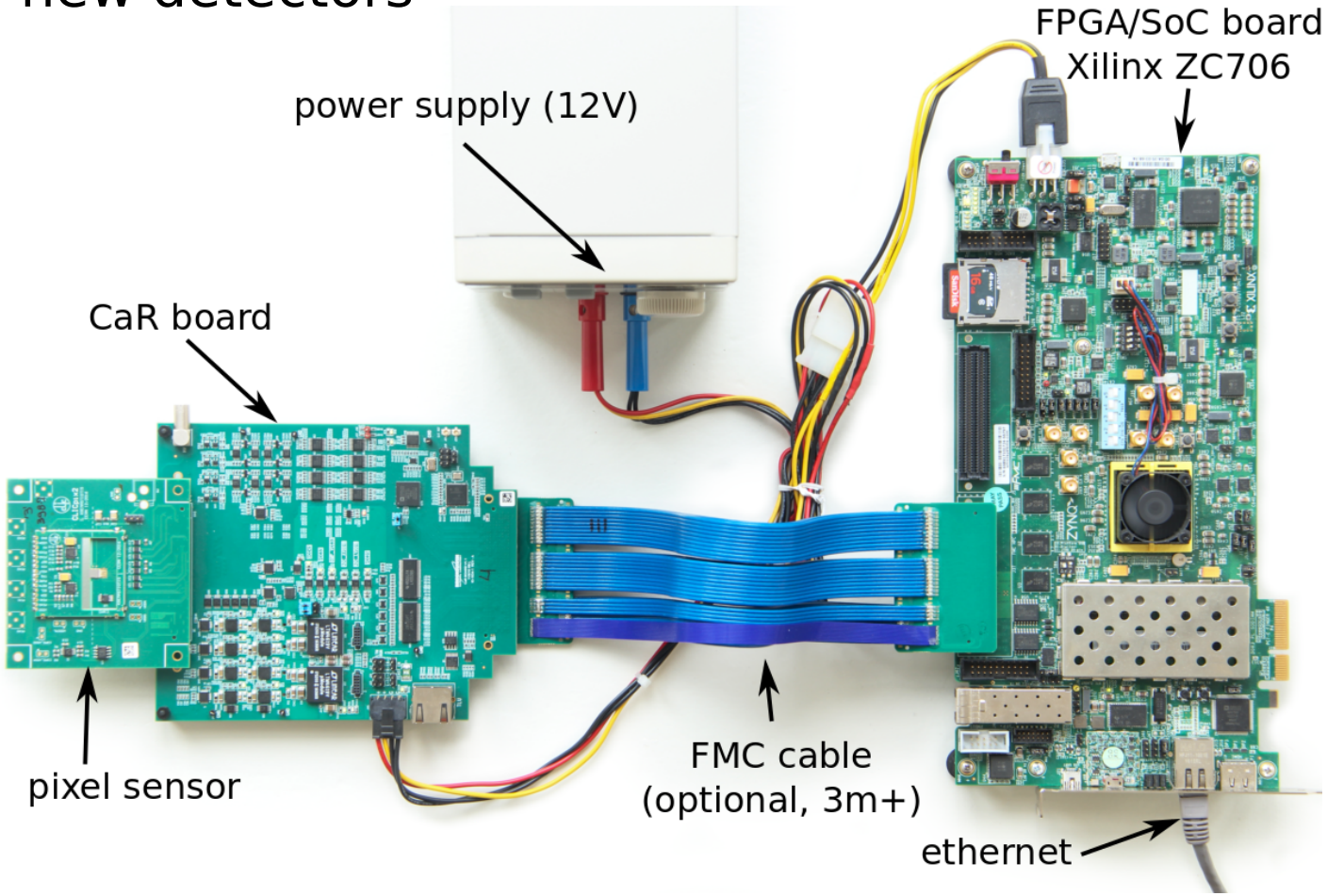


GitLab Continuous Integration – all stages

- **Compilation**
  - compile source code on Scientific Linux 6, CentOS7, and Mac OS X with GCC, Clang, and AppleClang
- **Testing**
  - analyse test data sets and compare output to pass conditions
- **Formatting**
  - check format against defined syntax rules (e.g. tabs ↔ whitespaces) to avoid changes caused e.g. by different indentation, and apply linting
- **Documentation**
  - compile user manual from LaTeX sources and generate Doxygen code reference
- **Packaging**
  - generate release tarballs
- **Deployment**
  - publish new version of CVMFS, new docker image in registry, new user manual and code reference on the website and release tarballs

Caribou – the readout system

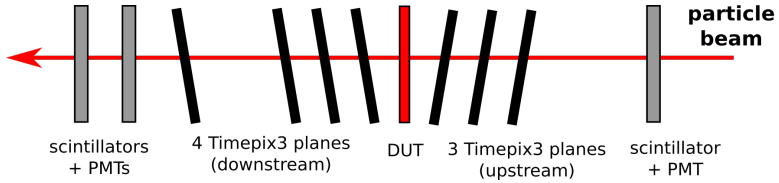
- versatile, open-source, linux-based
- fast & simple implementation of new detectors
  → “fast prototyping”
- **universal:**
  - FPGA board
  - Control & Readout (CaR) board
  - “most of the” firmware/software
- **chip-specific:**
  - chip board
  - “some” firmware/software blocks

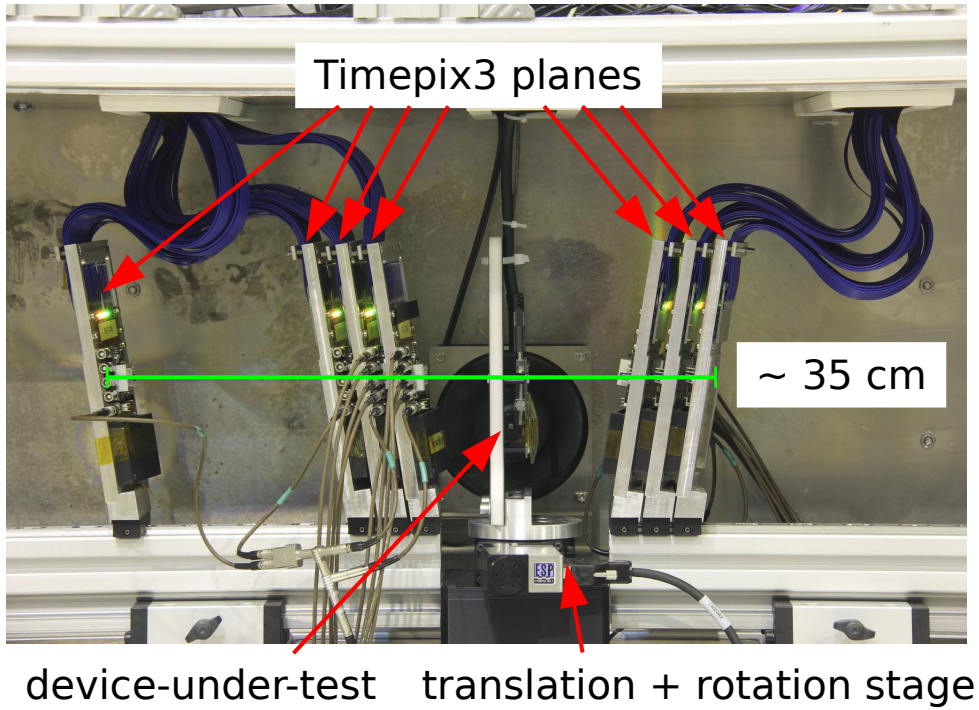


SPS vs. DESY II

SPS:

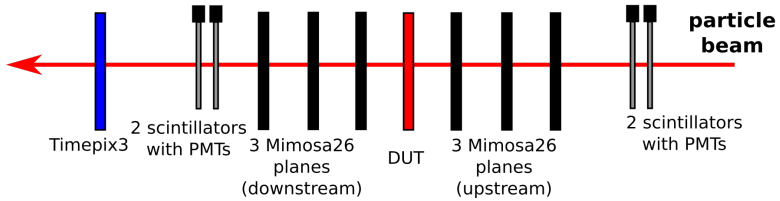
- typical beam condition:
  120 GeV pions @ few MHz
- telescope in operation
  2014-2018



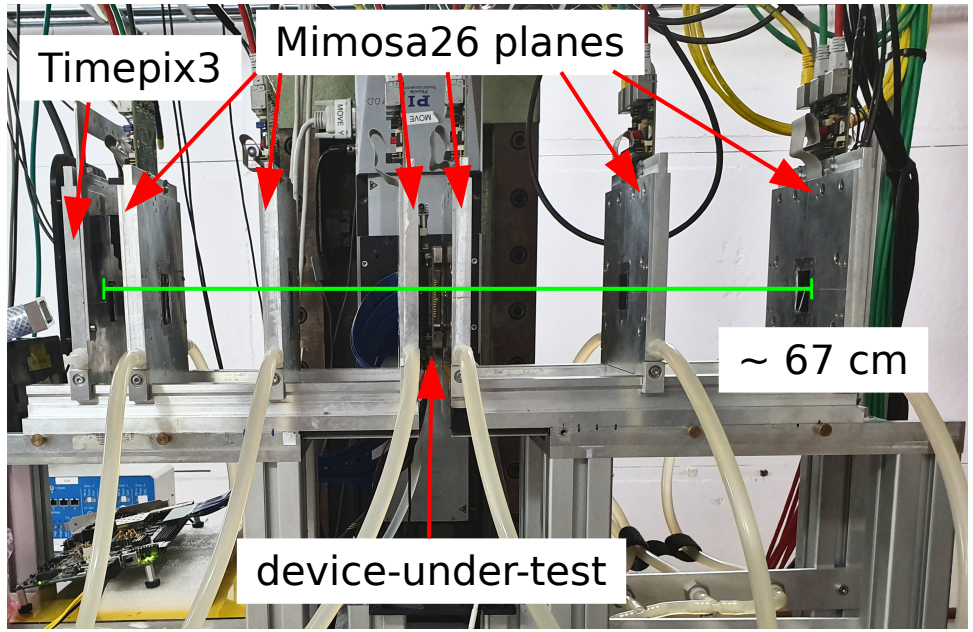


DESY:

- typical beam condition:
  5.4 GeV electrons @ few kHz
- use for CLICdp testbeam campaigns during LHC LS2 2019-2020



→ much lower rate & energy

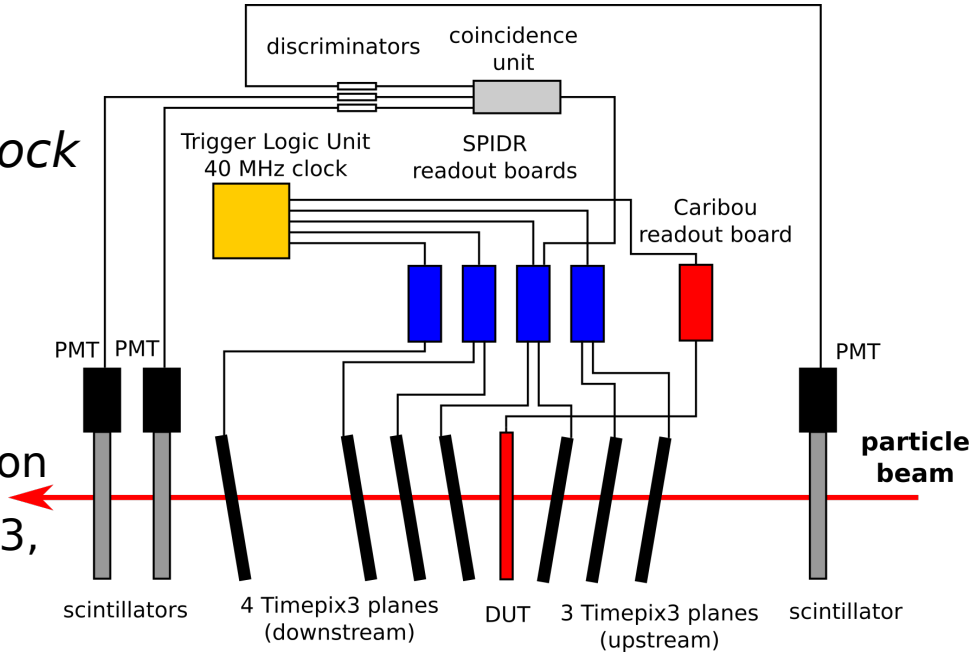


SPS vs. DESY II – Readout

continuous readout, timestamps synchronous with global clock

SPS:

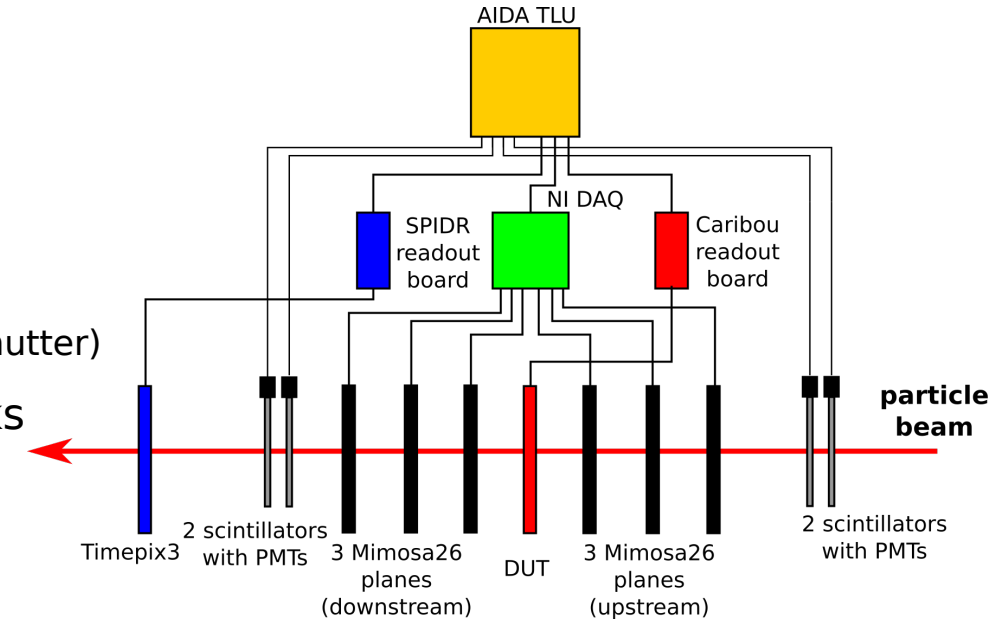
- Trigger Logic Unit (TLU) → provides global clock (time sync.)
- 3 scintillators + PMTs → trigger timestamps
- 7 Timepix3 planes → excellent spatial + timing resolution
- DUT → Investigator, Cracow SOI, Timepix3, CLICpix, CLICpix2, ATLASpix



DESY:

- AIDA TLU → provides global clock (time sync.) + triggers Mimosa Readout
- 4 scintillators + PMTs → input to TLU
- 6 Mimosa26 planes → good spatial res. (2x 115µs bins rolling shutter)
- Timepix3 → used to assign ns timestamp to tracks
- DUT → CLICpix2, ATLASpix, CLICTD

→ additional subsystem

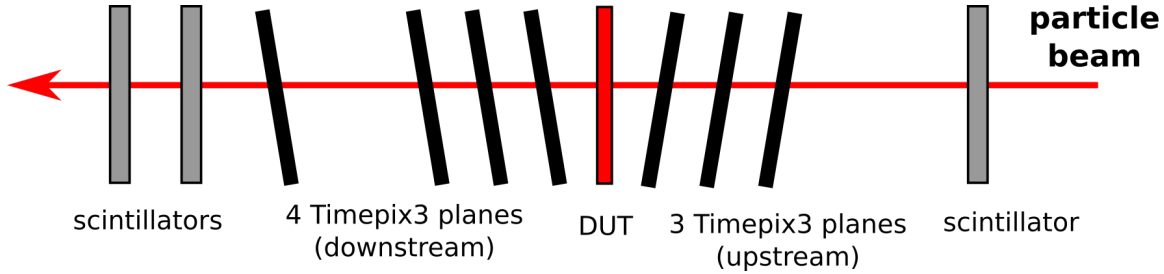


SPS vs. DESY II – Differences in the Analysis

Tracking:

- **SPS**:
  - ➔ 7 Timepix3 hits with precise timestamp
  - ➔ track timestamp = average TPX3 timestamp
- **DESY**:
  - ➔ Mimosa26 hits (3x 115µs) with multiple trigger timestamps
  - ➔ require Timepix3 for unambiguous track time
  - ➔ track timestamp = TPX3 timestamp

SPS:
all sensors provide hit timestamps for tracking



DESY:
unambiguous track timestamp only with TPX3