



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386



GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung



Corryvreckan

The Maelstrom for Your Test Beam Data



8th BTTB Workshop

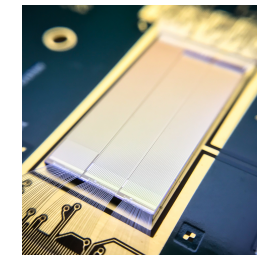
Tbilisi State University, January 28th, 2020

Jens Kröger, Heidelberg University & CERN

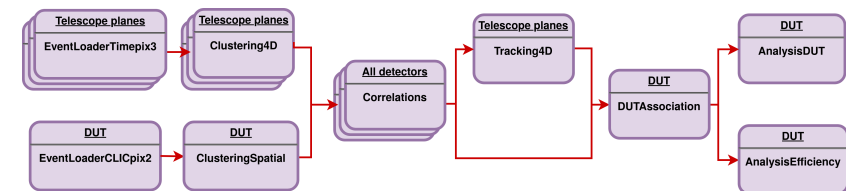
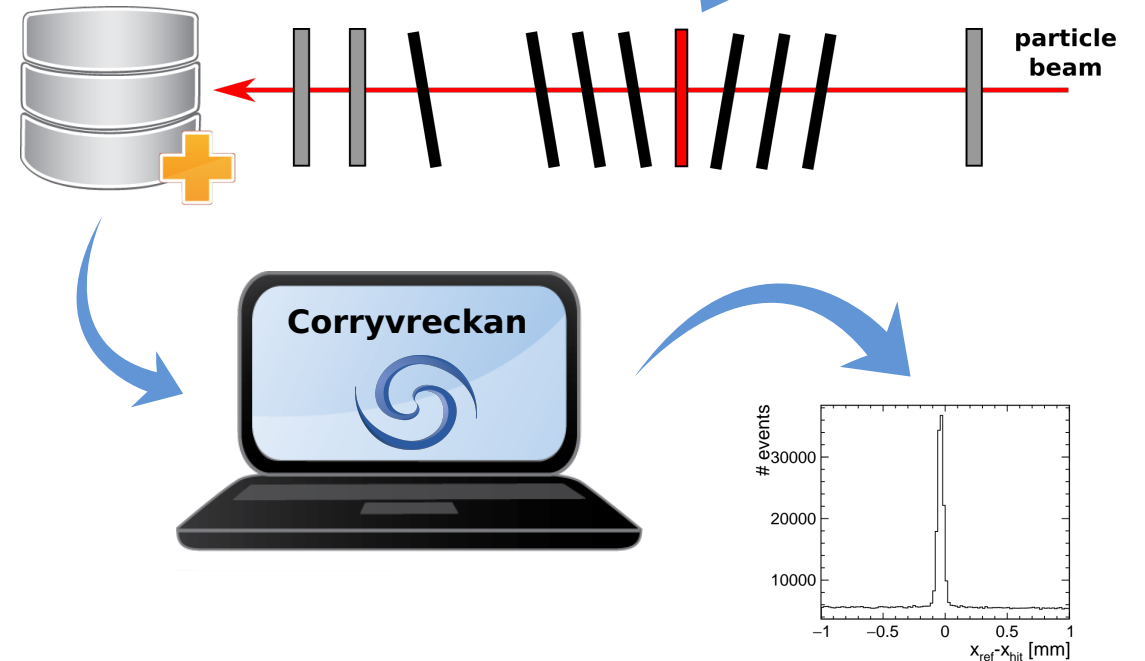
*on behalf of the CLICdp collaboration
and the Corryvreckan Developers*

Corryvreckan

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

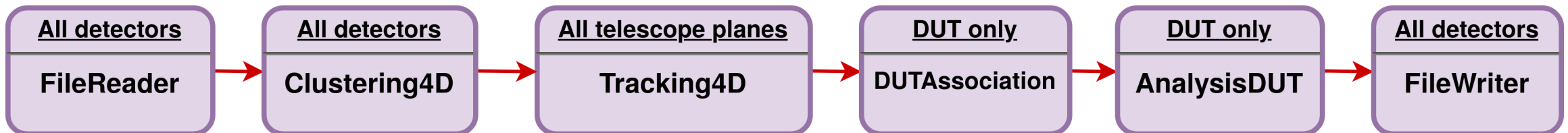


- modular structure
 - framework core
 - modules for specific tasks
- highly flexible and configurable
- easy to understand
 - written in modern C++
 - comprehensive documentation (> 100 pages!)



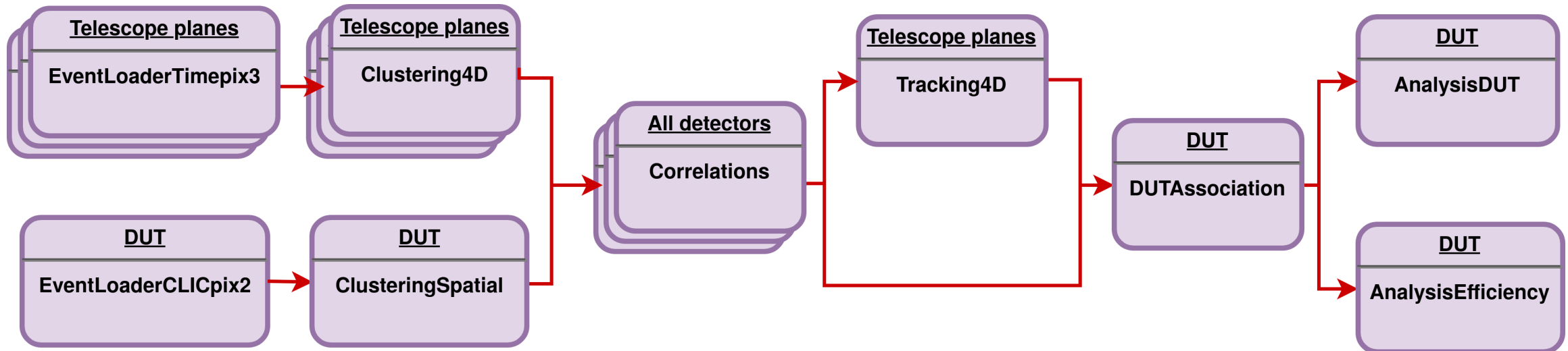
The Modular Approach

- 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

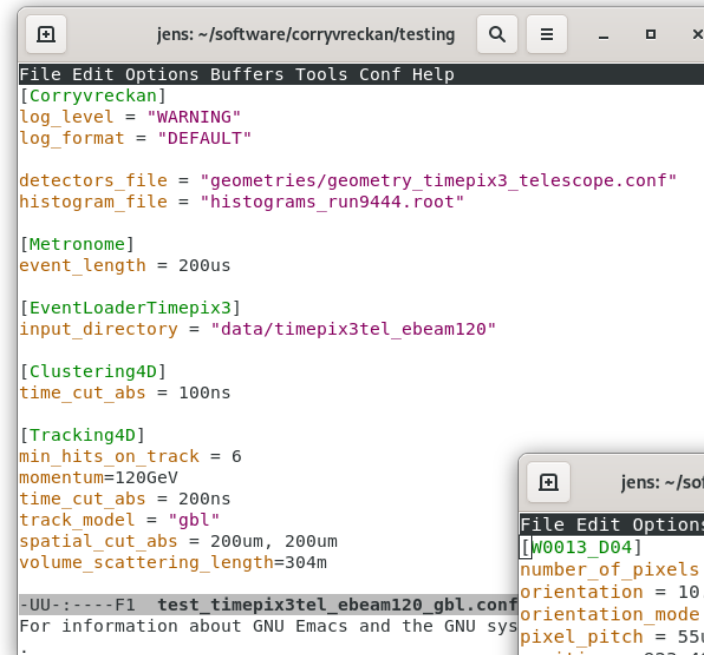
- can create more complex reconstruction chains
- apply different modules to different devices in the same reconstruction



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

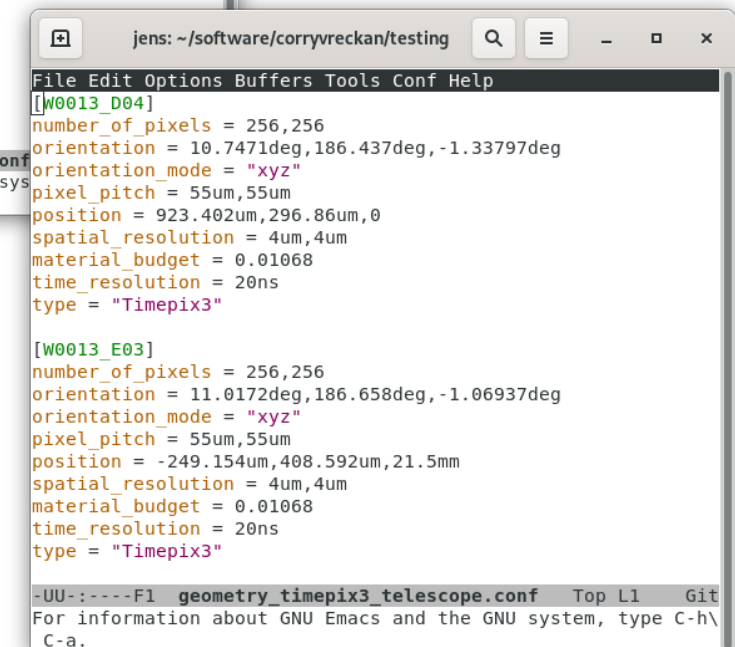
[EventLoaderTimepix3]
input_directory = "data/timepix3tel_ebeam120"

[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



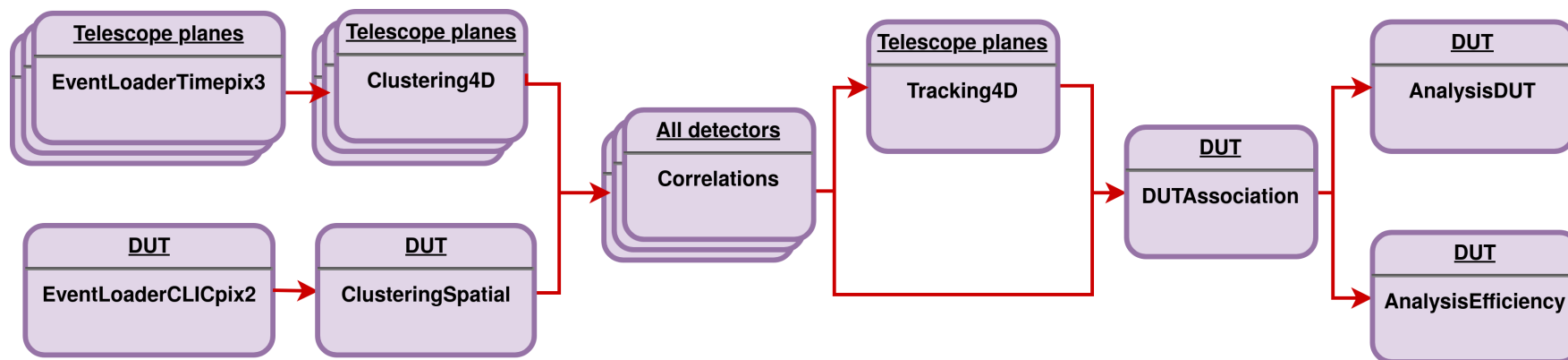
```
jens: ~/software/corryvreckan/testing
File Edit Options Buffers Tools Conf Help
[W0013_D04]
number_of_pixels = 256,256
orientation = 10.7471deg,186.437deg,-1.33797deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 923.402um,296.86um,0
spatial_resolution = 4um,4um
material_budget = 0.01068
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



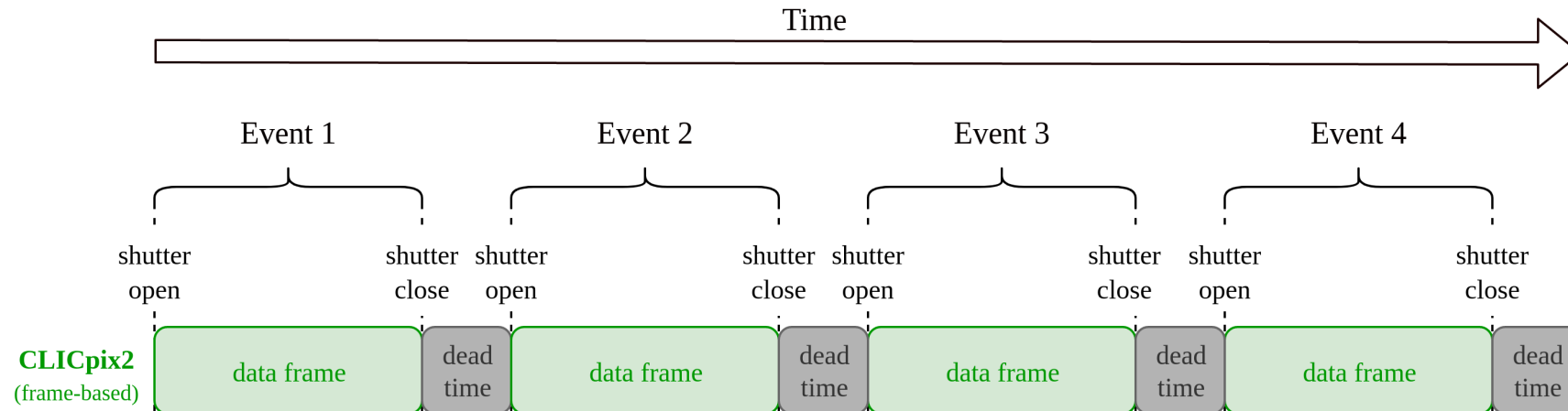
Examples – Flexible Event Building

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

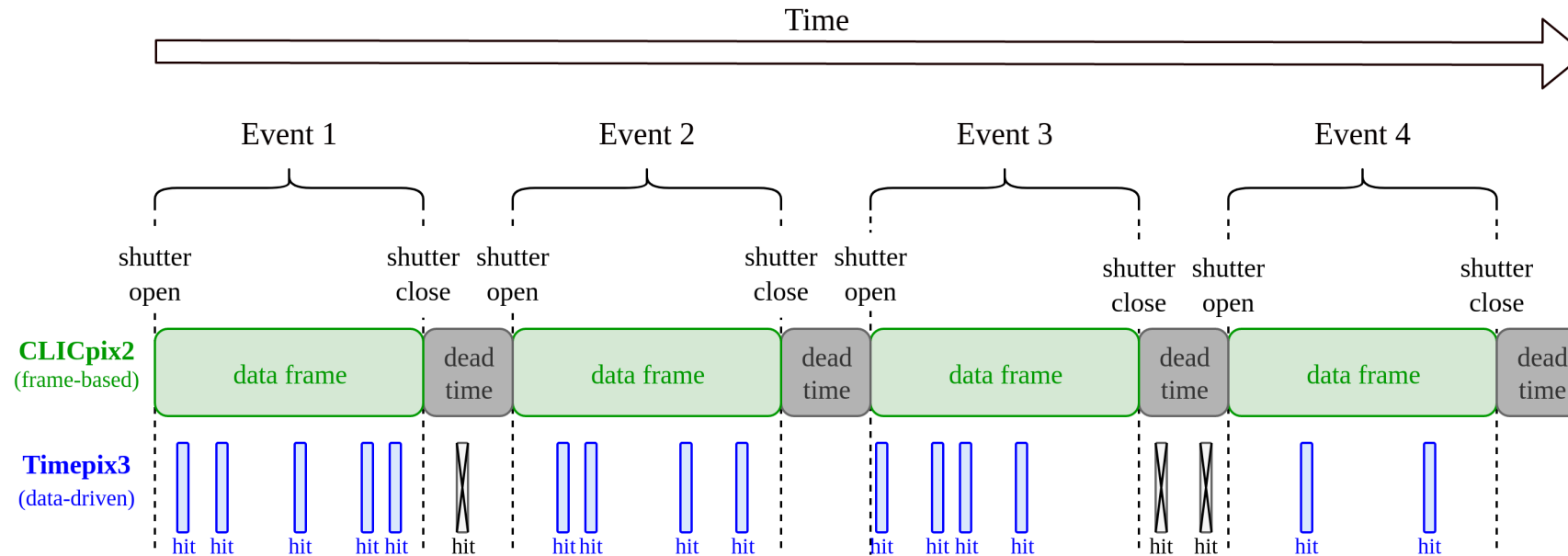
The 1st module defines the event!

- **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!

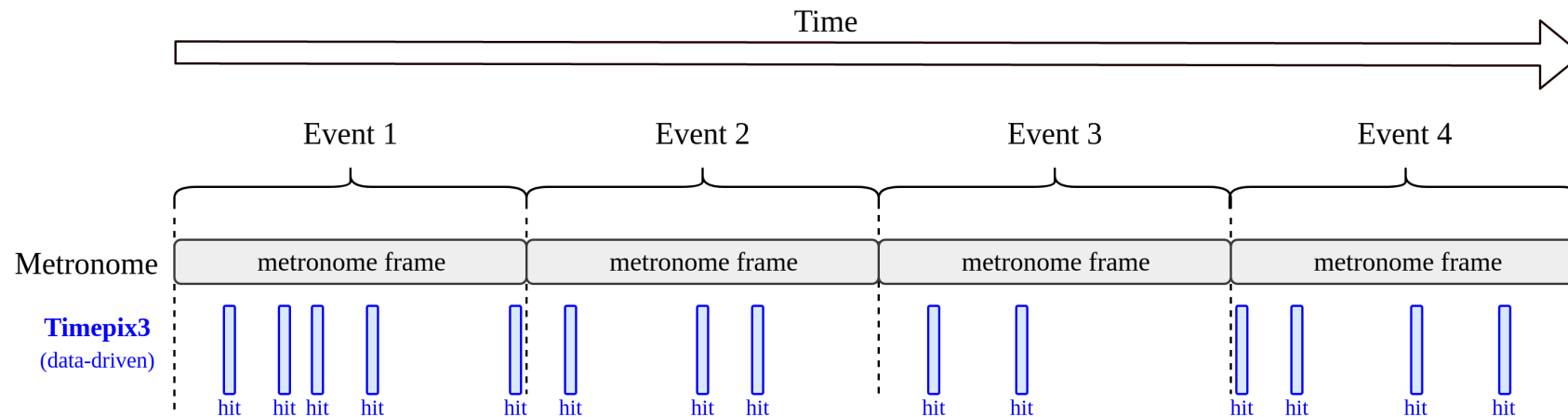
- **data-driven:**

- [Metronome] module 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!

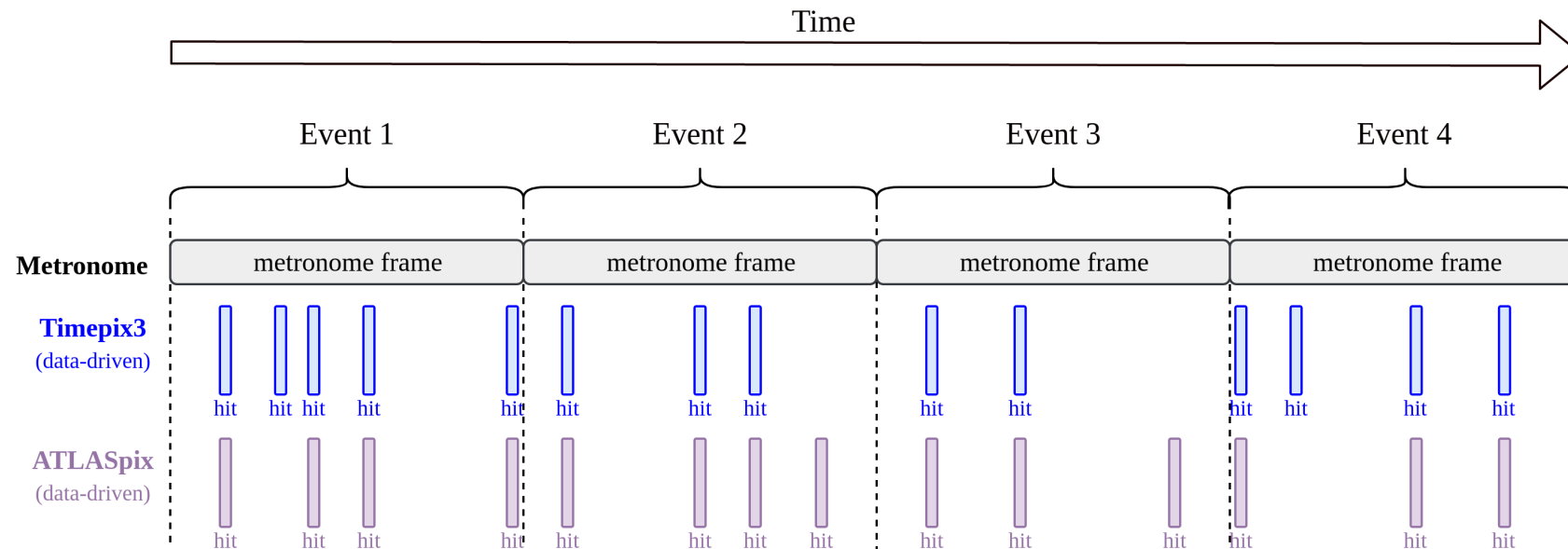
- **data-driven:**

- [Metronome] module 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

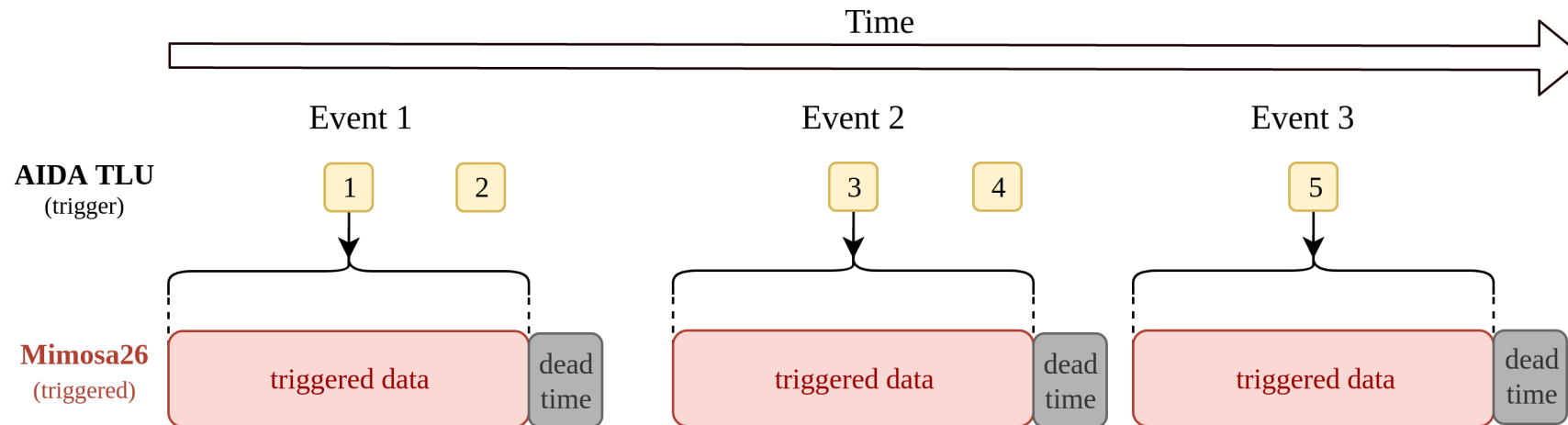
The 1st module defines the event!

- **combined (1)**

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

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

[EventLoaderEUDAQ2]
type = "MIMOSA26"
```



Examples – Flexible Event Building

The 1st module defines the event!

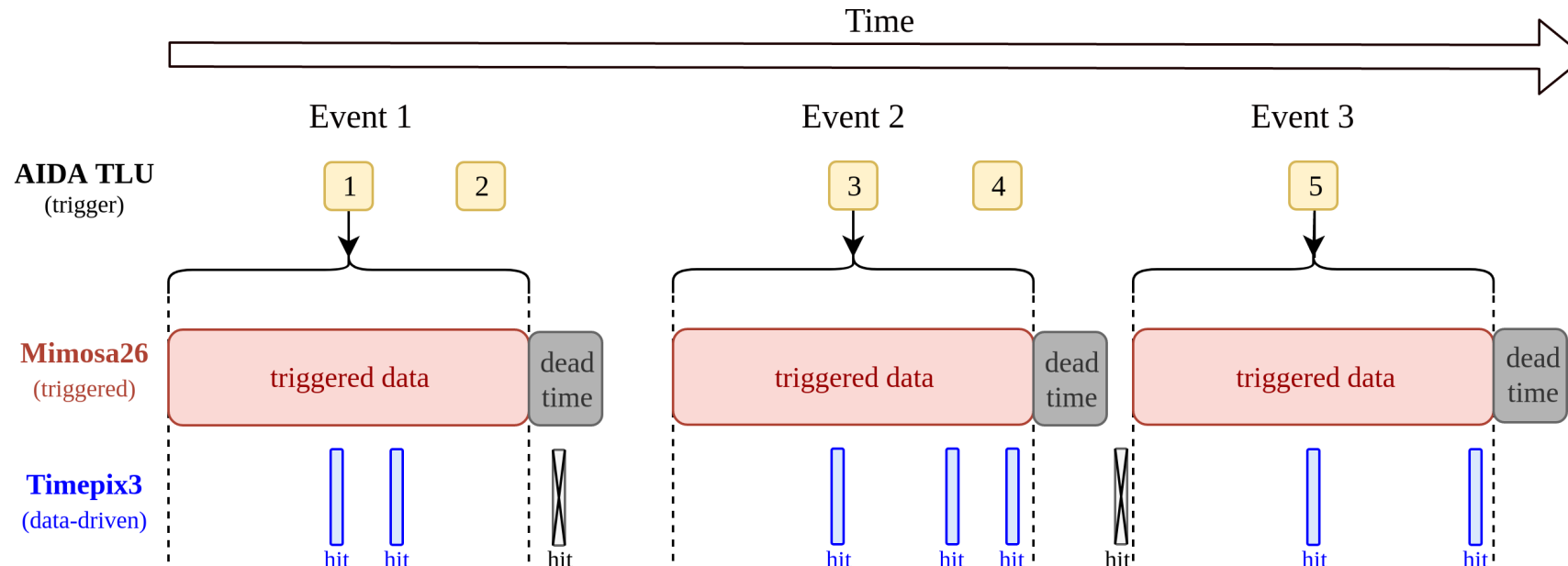
- **combined (2)**

- 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*

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

[EventLoaderEUDAQ2]
type = "MIMOSA26"

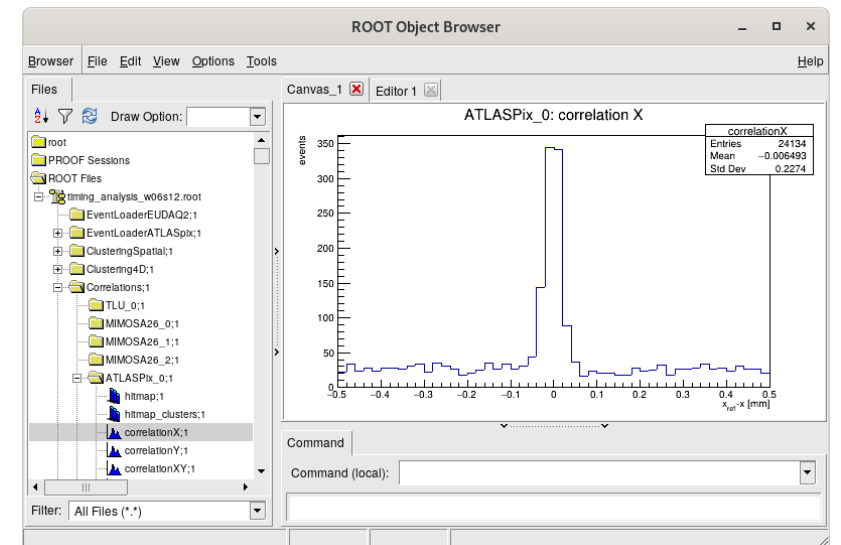
[EventLoaderEUDAQ2]
name = "Timepix3_0"
```



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

```
jens:~/software/testbeam-analysis/macros/DESY_2019-06
jens:~/software/testbeam-analysis/macros/DESY_2019-06$ corry -c timing_analysis_apx_w06s12.conf -o number_of_events=10000
18:46:35.344 (STATUS) Welcome to Corryvreckan v1.6+224gef47c2ad
18:46:35.347 (STATUS) Loaded 0 detectors
18:46:35.347 (WARNING) Main ROOT file /home/jens/software/testbeam-analysis/macros/DESY_2019-06/output/timing_analysis_w06s12.root exists and will be overwritten.
18:46:35.520 (STATUS) Loaded 31 module instances
18:46:35.520 (STATUS) ===== Initialising modules =====
18:46:36.000 (STATUS) [E:EventManager:ATLASpIx_0] Initialising "EventManager:ATLASpIx_0"
18:46:36.000 (STATUS) [I:EventLoaderATLASpIx:ATLASpIx_0] Opened data file for ATLASpIx: (dbg)/home/jens/data_local/tbJune2019/Run715/data.bin
18:46:36.575 (STATUS) [I:Correlations:MIMOSA26_0] Initialising "Correlations:MIMOSA26_0"
18:46:36.575 (WARNING) [I:Correlations:MIMOSA26_0] [further messages will be suppressed] Correlations module is enabled and will significantly increase the runtime
18:46:36.600 (STATUS) [I:AnalysisTimingATLASpIx:ATLASpIx_0] Initialising "AnalysisTimingATLASpIx:ATLASpIx_0"
18:46:37.268 (STATUS) [I:AnalysisTimingATLASpIx:ATLASpIx_0] ----> NO POINTWISE ROW CORRECTION!!!
18:46:37.268 (STATUS) [I:AnalysisTimingATLASpIx:ATLASpIx_0] ----> NO POINTWISE TIMEWALK CORRECTION!!!
18:46:37.268 (STATUS) ===== Event loop =====
18:46:41.079 (STATUS) Ev: 2.4k Px: 226.8k Tr: 1.7k (0.699/ev) t = 5.25758s ^C
18:46:41.953 (STATUS) [I:Correlations:MIMOSA26_2] Interrupted! Finishing up current event...
18:46:41.954 (STATUS) ===== Finishing modules =====
18:46:50.921 (STATUS) [F:DUTAssociation:ATLASpIx_0] In total, 609 clusters are associated to 553 tracks.
18:46:52.434 (STATUS) [F:AnalysisEfficiency:ATLASpIx_0] Track selection flow: 1726
* track outside ROI -1296
* track outside DUT -273
* track close to masked px -3
Accepted tracks: 154
18:46:52.434 (STATUS) [F:AnalysisEfficiency:ATLASpIx_0] Total efficiency of detector ATLASpIx_0: 90.2597%, measured with 139/154 matched/total tracks
18:46:52.540 (STATUS) [F:AnalysisTimingATLASpIx:ATLASpIx_0] Timing analysis finished for detector ATLASpIx_0:
warning in «Fit»: Fit data is empty
18:46:57.383 (STATUS) Wrote histogram output file to /home/jens/software/testbeam-analysis/macros/DESY_2019-06/output/timing_analysis_w06s12.root
18:46:57.383 (STATUS) ===== Wall-clock timing (seconds) =====
18:46:57.383 (STATUS) EventLoaderEUDAQ2 : TLU_0 -- 0.29254s = 0.121036ms/evt
18:46:57.383 (STATUS) EventLoaderEUDAQ2 : MIMOSA26_0 -- 0.23208s = 0.096021ms/evt
18:46:57.383 (STATUS) EventLoaderEUDAQ2 : MIMOSA26_1 -- 0.21008s = 0.086805ms/evt
18:46:57.383 (STATUS) EventLoaderEUDAQ2 : MIMOSA26_2 -- 0.20999s = 0.086805ms/evt
18:46:57.383 (STATUS) EventLoaderEUDAQ2 : MIMOSA26_3 -- 0.21090s = 0.087255ms/evt
18:46:57.384 (STATUS) EventLoaderEUDAQ2 : MIMOSA26_4 -- 0.21847s = 0.089560ms/evt
18:46:57.384 (STATUS) EventLoaderEUDAQ2 : MIMOSA26_5 -- 0.22346s = 0.092453ms/evt
18:46:57.384 (STATUS) EventLoaderATLASpIx : ATLASpIx_0 -- 0.05679s = 0.023495ms/evt
18:46:57.384 (STATUS) EventLoaderEUDAQ2 : Timepix3_0 -- 0.40322s = 0.166808ms/evt
18:46:57.384 (STATUS) ClusteringSpatial : MIMOSA26_0 -- 0.14312s = 0.059215ms/evt
18:46:57.384 (STATUS) ClusteringSpatial : MIMOSA26_1 -- 0.08965s = 0.037692ms/evt
18:46:57.384 (STATUS) ClusteringSpatial : MIMOSA26_2 -- 0.08571s = 0.035462ms/evt
18:46:57.384 (STATUS) ClusteringSpatial : MIMOSA26_3 -- 0.08814s = 0.036467ms/evt
18:46:57.384 (STATUS) ClusteringSpatial : MIMOSA26_4 -- 0.08596s = 0.035563ms/evt
18:46:57.384 (STATUS) ClusteringSpatial : MIMOSA26_5 -- 0.08554s = 0.035309ms/evt
18:46:57.384 (STATUS) Clustering4D : ATLASpIx_0 -- 0.01785s = 0.007307ms/evt
18:46:57.384 (STATUS) Clustering4D : Timepix3_0 -- 0.03802s = 0.015731ms/evt
18:46:57.384 (STATUS) Correlations : TLU_0 -- 0.00292s = 0.001208ms/evt
18:46:57.384 (STATUS) Correlations : MIMOSA26_0 -- 0.24308s = 0.099742ms/evt
18:46:57.384 (STATUS) Correlations : MIMOSA26_1 -- 0.21187s = 0.087657ms/evt
18:46:57.384 (STATUS) Correlations : MIMOSA26_2 -- 0.20939s = 0.086632ms/evt
18:46:57.384 (STATUS) Correlations : ATLASpIx_0 -- 0.04379s = 0.018025ms/evt
18:46:57.384 (STATUS) Correlations : MIMOSA26_3 -- 0.23942s = 0.099565ms/evt
18:46:57.384 (STATUS) Correlations : MIMOSA26_4 -- 0.21907s = 0.090637ms/evt
18:46:57.384 (STATUS) Correlations : MIMOSA26_5 -- 0.21814s = 0.090252ms/evt
18:46:57.384 (STATUS) Correlations : Timepix3_0 -- 0.14979s = 0.061746ms/evt
18:46:57.384 (STATUS) Tracking4D : -- 0.36804s = 0.152270ms/evt
18:46:57.384 (STATUS) DUTAssociation : ATLASpIx_0 -- 0.01061s = 0.004391ms/evt
18:46:57.384 (STATUS) analysisSDT : ATLASpIx_0 -- 0.00609s = 0.002835ms/evt
18:46:57.384 (STATUS) AnalysisEfficiency : ATLASpIx_0 -- 0.00689s = 0.002825ms/evt
18:46:57.384 (STATUS) AnalysisTimingATLASpIx : ATLASpIx_0 -- 0.00591s = 0.002447ms/evt
jens:~/software/testbeam-analysis/macros/DESY_2019-06
```



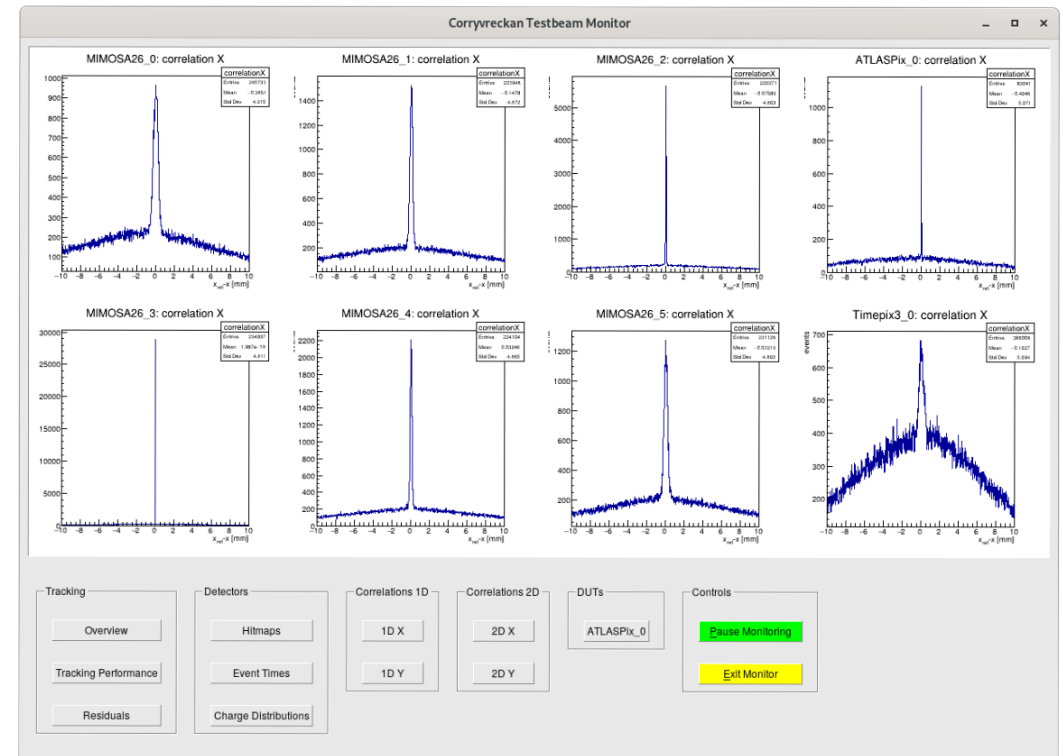
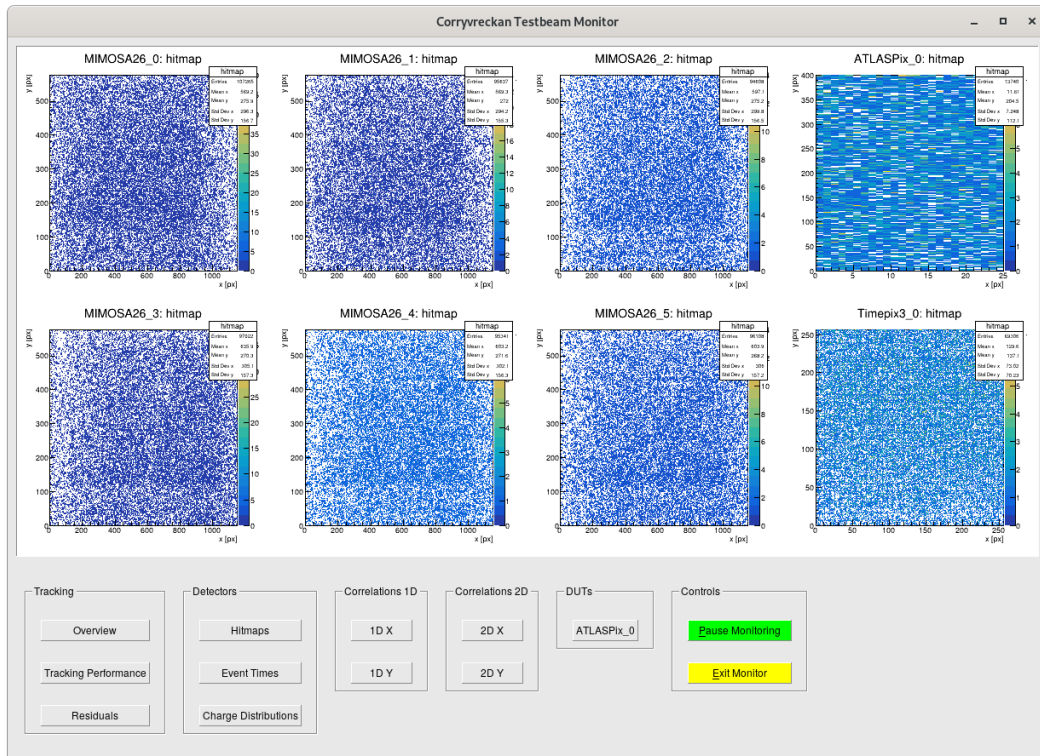
The Online Monitor

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

```
[OnlineMonitor]
```

```
update = 200
```

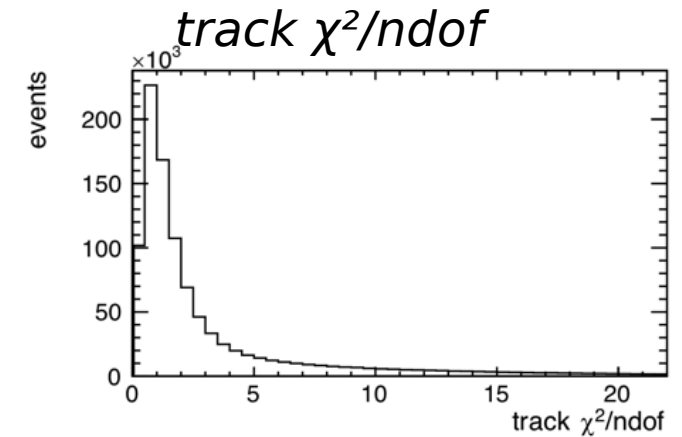
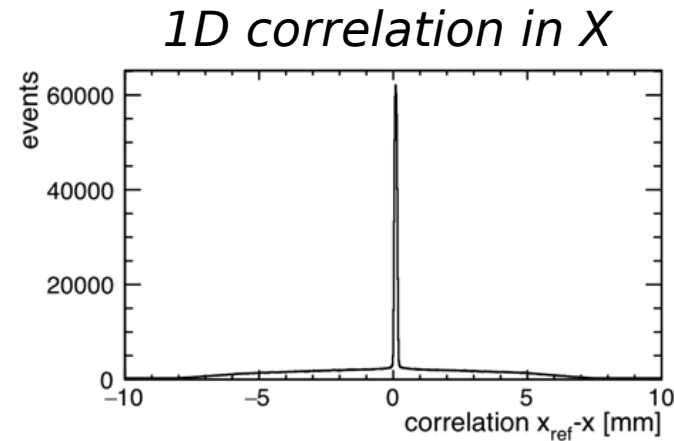
```
dut_plots = [{"EventLoaderEUDAQ2/%DUT%/hitmap", "colz"},  
             [{"EventLoaderEUDAQ2/%DUT%/hPixelTimes"}]]
```



Some Example Results

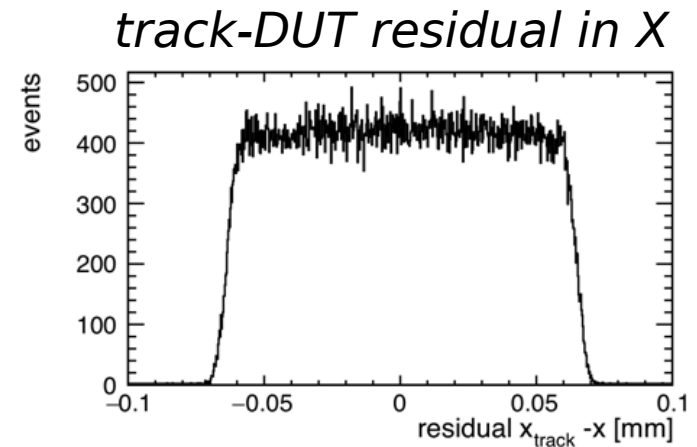
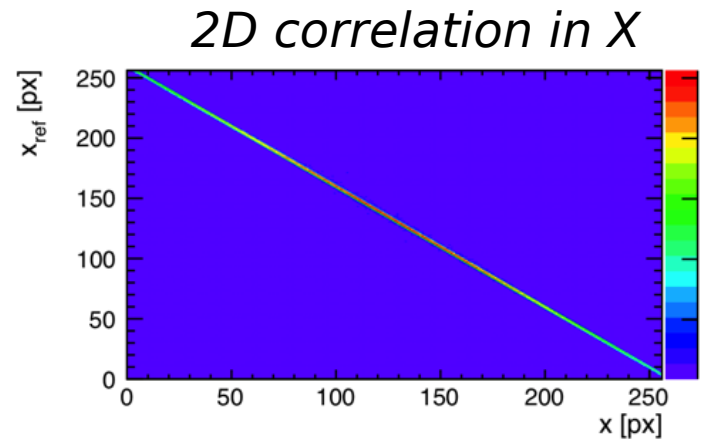
- DUT analysis plots:

- ATLASpix in Timepix3 telescope at CERN SPS



- data converters implemented for:

- ATLASpix
- CLICpix/CLICpix2
- Timepix1/Timepix3

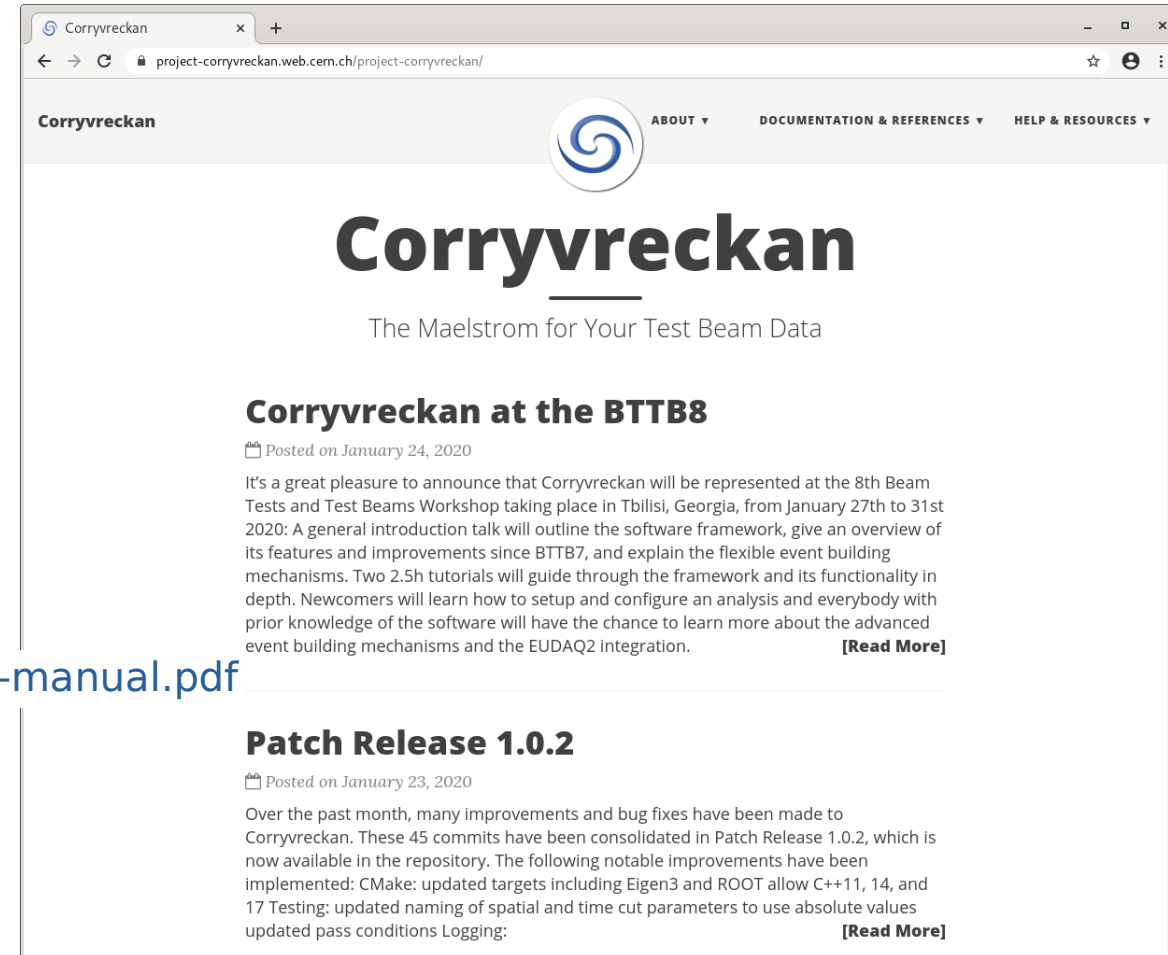


- **EUDAQ2:** AIDA TLU, Mimosa26 telescope, ATLASpix, CLICpix2, CLICTD, Timepix3

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, Mimosas26 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

The screenshot shows a GitLab repository page for the 'Clustering4D' module. At the top, there's a breadcrumb trail: 'Corryvreckan > Corryvreckan > Repository'. Below that, the repository path is shown as 'master / corryvreckan / src / modules / Clustering4D /'. There are buttons for 'History', 'Find file', 'Web IDE', and a download icon. A commit message is visible: 'Fix mixup of coordinates' by Simon Spannagel, authored 1 month ago. Below the commit message is a table listing files and their last commit details:

Name	Last commit	Last update
--		
CMakeLists.txt	Rename Timepix3Clustering -> Clustering4D	1 year ago
Clustering4D.cpp	Spatial cuts: backwards compatibility for older names, print...	1 month ago
Clustering4D.h	Implementing simultaneous relative and absolute time cuts ...	2 months ago
README.md	Fix mixup of coordinates	1 month ago

Below the table, the 'README.md' file is expanded, showing the following content:

Clustering4D

Maintainer: Daniel Hynds (daniel.hynds@cern.ch)
Module Type: DETECTOR
Detector Type: all
Status: Functional

Description

This module performs clustering for detectors with valid individual hit timestamps. The clustering method is either an arithmetic mean or a charge-weighted centre-of-gravity calculation, using a positional cut and a timing cut on proximity. If the pixel information is binary (i.e. no valid charge-equivalent information is available), the arithmetic mean is calculated for the position. Also, if one pixel of a cluster has charge zero, the arithmetic mean is calculated even if charge-weighting is selected because it is assumed that the zero-reading is false and does not to represent a low charge but an unknown value. Thus, the arithmetic mean is safer.

Split clusters can be recovered using a larger search radius for neighbouring pixels.

Parameters

- **time_cut_rel**: Number of standard deviations the **time_resolution** of the detector plane will be multiplied by. This value is then used as the maximum time difference allowed between pixels for association to a cluster. By default, a relative time cut is applied. Absolute and relative time cuts are mutually exclusive. Defaults to **3.0**.
- **time_cut_abs**: Specifies an absolute value for the maximum time difference allowed between pixels for association to a cluster. Absolute and relative time cuts are mutually exclusive. No default value.
- **neighbour_radius_col**: Search radius for neighbouring pixels in column direction, defaults to **1** (do not allow split clusters)
- **neighbour_radius_row**: Search radius for neighbouring pixels in row direction, defaults to **1** (do not allow split clusters)
- **charge_weighting**: If true, calculate a charge-weighted mean for the cluster centre. If false, calculate the simple arithmetic mean. Defaults to **true**.

Plots produced

For each detector the following plots are produced:

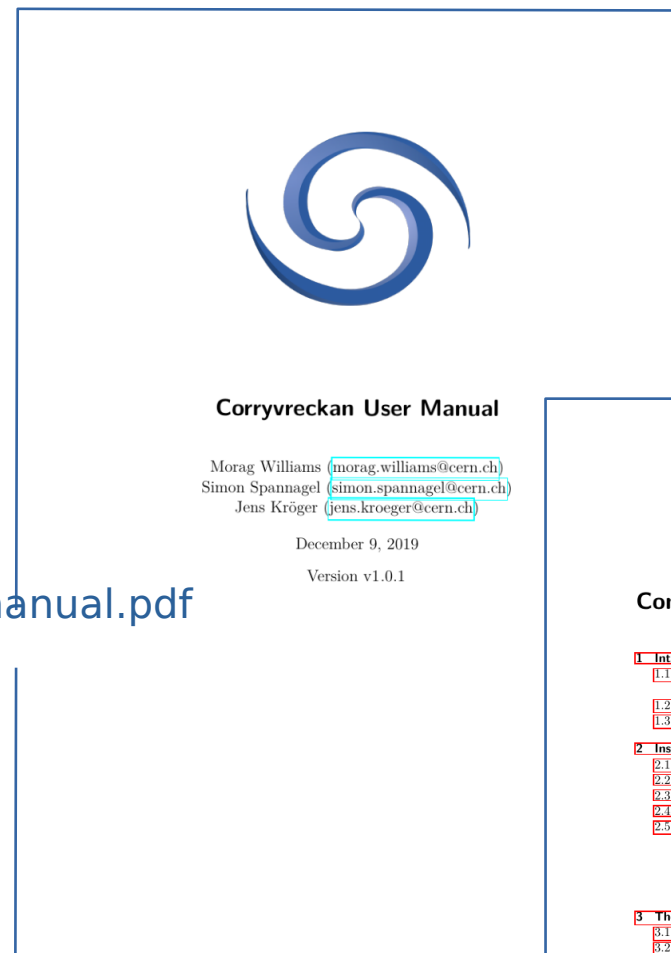
- Histograms for cluster size, seed charge, width (columns/X and rows/Y)
- Cluster charge histogram
- 2D cluster positions in global coordinates
- Cluster times
- Cluster multiplicity

Usage

```
[Clustering4D]
time_cut_rel = 3.0
```


Documentation

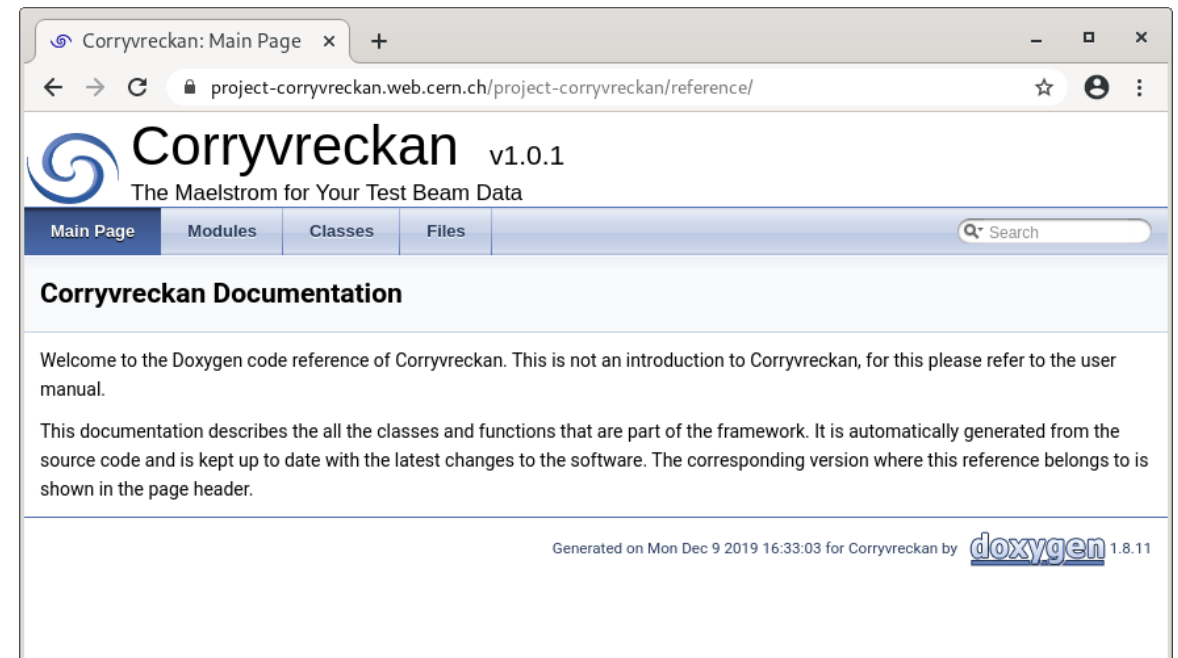
- online documentation in repo
<https://gitlab.cern.ch/corryvreckan/corryvreckan>
 - every modules has a README
- **extensive user manual**
<https://cern.ch/corryvreckan/usermanual/corryvreckan-manual.pdf>
 - full description of framework
 - installation instructions
 - “Getting started”, FAQs
 - module 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



Contents	
1 Introduction	1
1.1 Scope of this Manual	1
1.1.1 Getting Started	2
1.2 Support and Reporting Issues	2
1.3 Contributing Code	2
2 Installation	3
2.1 Supported Operating Systems	3
2.2 CMVFS	3
2.3 Docker	3
2.4 Binaries	4
2.5 Compilation from Source	4
2.5.1 Prerequisites	5
2.5.2 Downloading the source code	5
2.5.3 Configuration via CMake	5
2.5.4 Compilation and installation	6
3 The Corryvreckan Framework	7
3.1 The Corry Executable	7
3.2 The Clipboard	8
3.2.1 The Event	9
3.2.2 Temporary Data Storage	9
3.2.3 Persistent Storage	9
3.3 Global Framework Parameters	9
3.4 Modules and the Module Manager	11
3.4.1 Module Status Codes	11
3.4.2 Execution Order	12
3.4.3 Module instantiation	12
3.5 Logging and Verbosity Levels	13
3.6 Coordinate Systems	14
4 Configuration Files	15
4.1 Parsing types and units	15
4.1.1 File format	18
4.1.2 Accessing parameters	19
4.2 Main configuration	20
4.3 Detector configuration	21
4.3.1 Masking Pixels Offline	24
4.3.2 Defining a Region of Interest	24

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

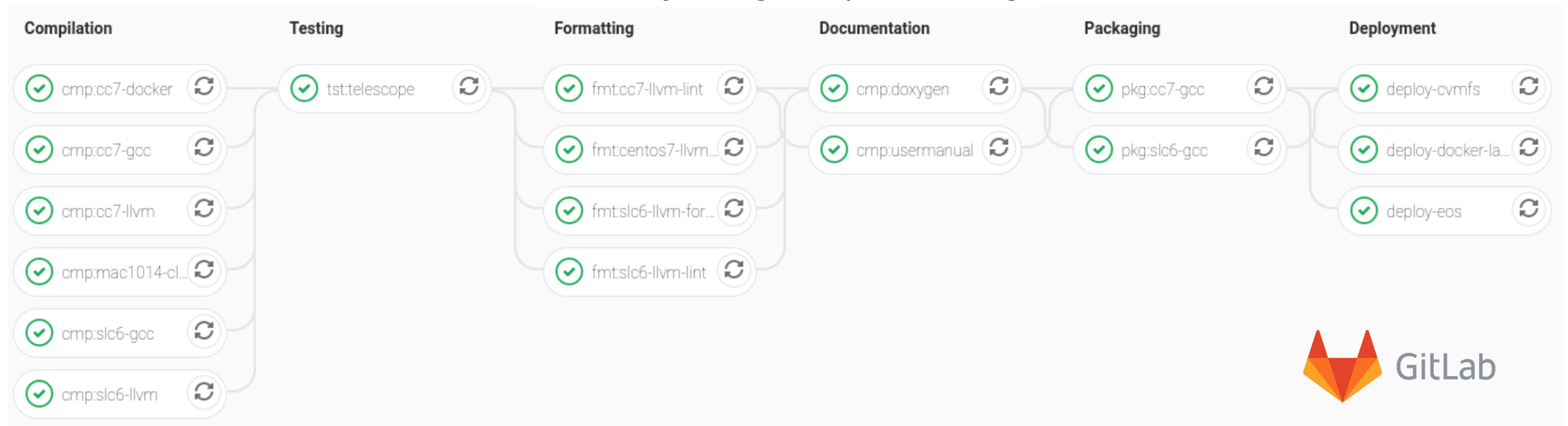


GitLab Continuous Integration

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

for every commit

for every merge request or tag



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

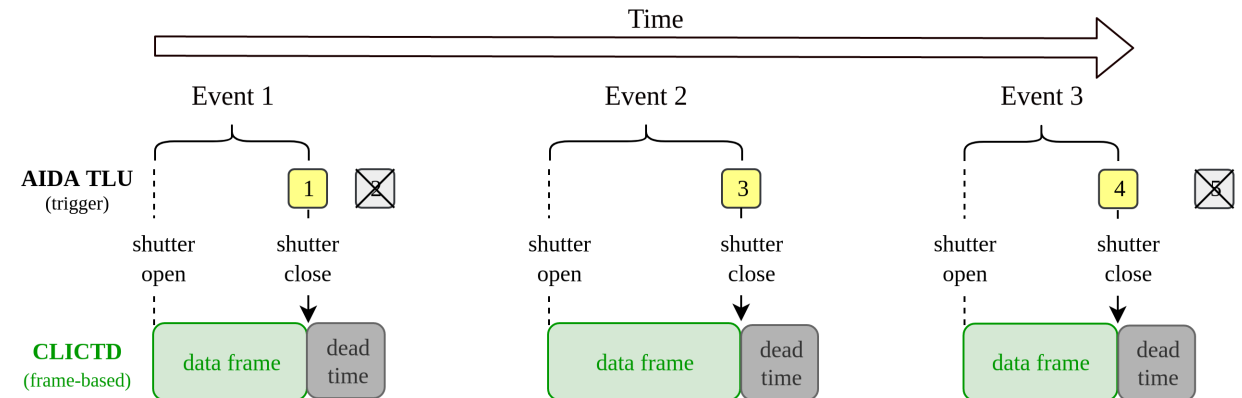
in case there are some questions...

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

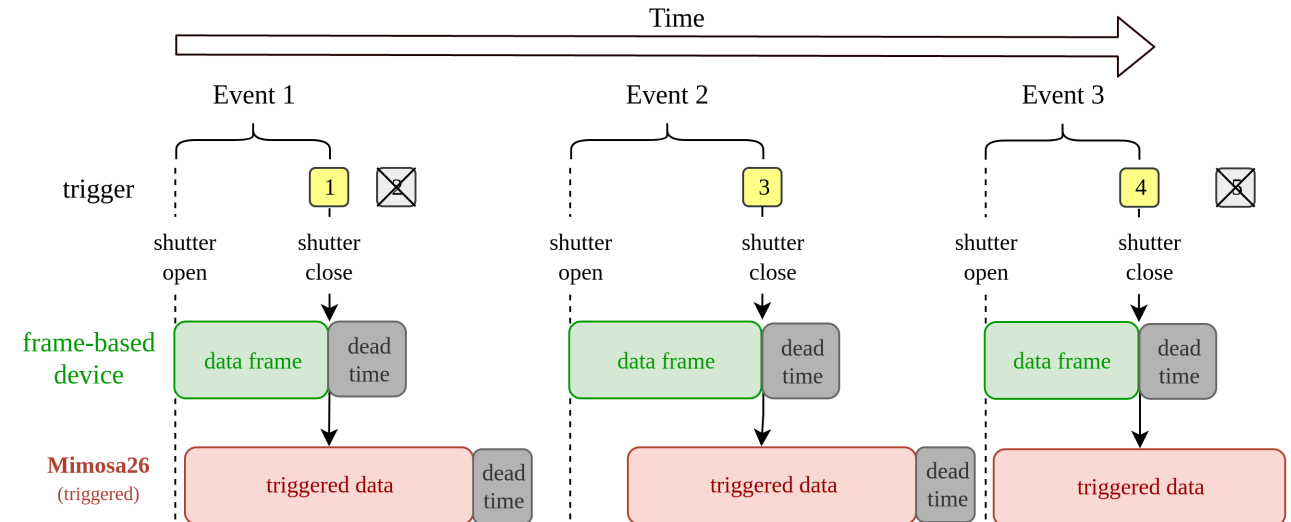
- *example:*
AIDA TLU + CLICTD + Mimosa26 telescope

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

The 1st module defines the event!

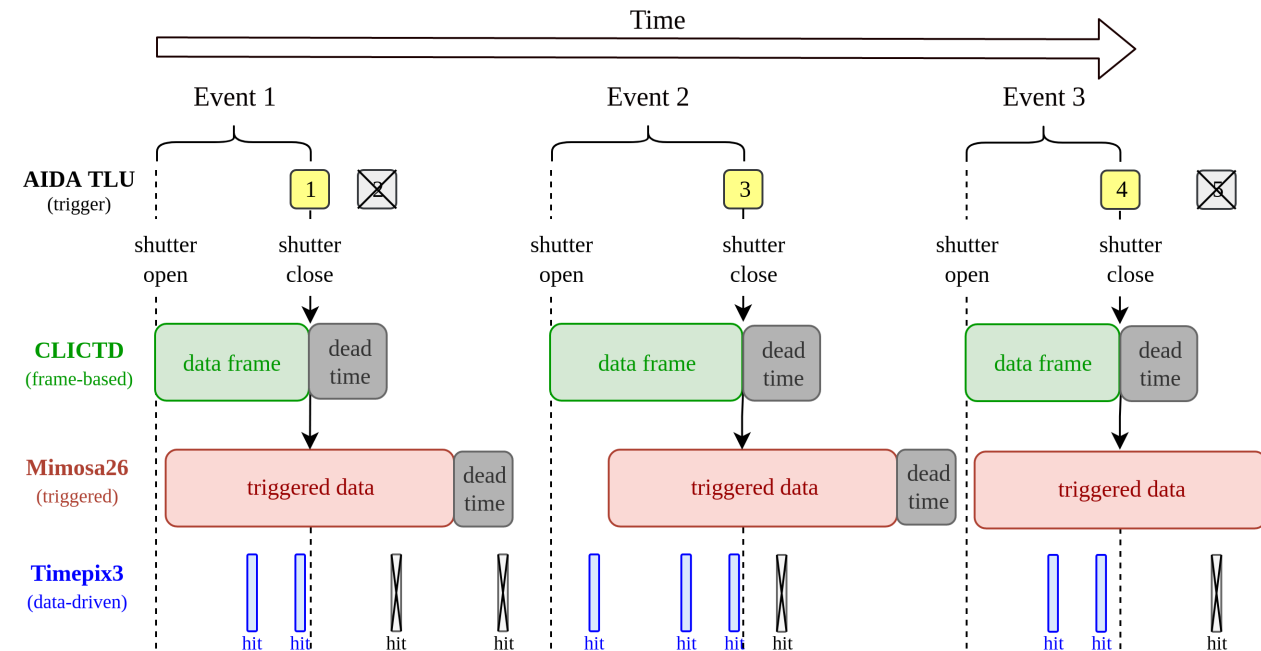
```
[EventLoaderEUDAQ2]
name = "CLICTD_0"

[EventLoaderEUDAQ2]
name = "TLU_0"

[EventLoaderEUDAQ2]
type = "MIMOSA26"

[EventLoaderEUDAQ2]
name = "Timepix3_0"
```

- *example:*
AIDA TLU + CLICTD
+ Mimosa26 telescope + Timepix3

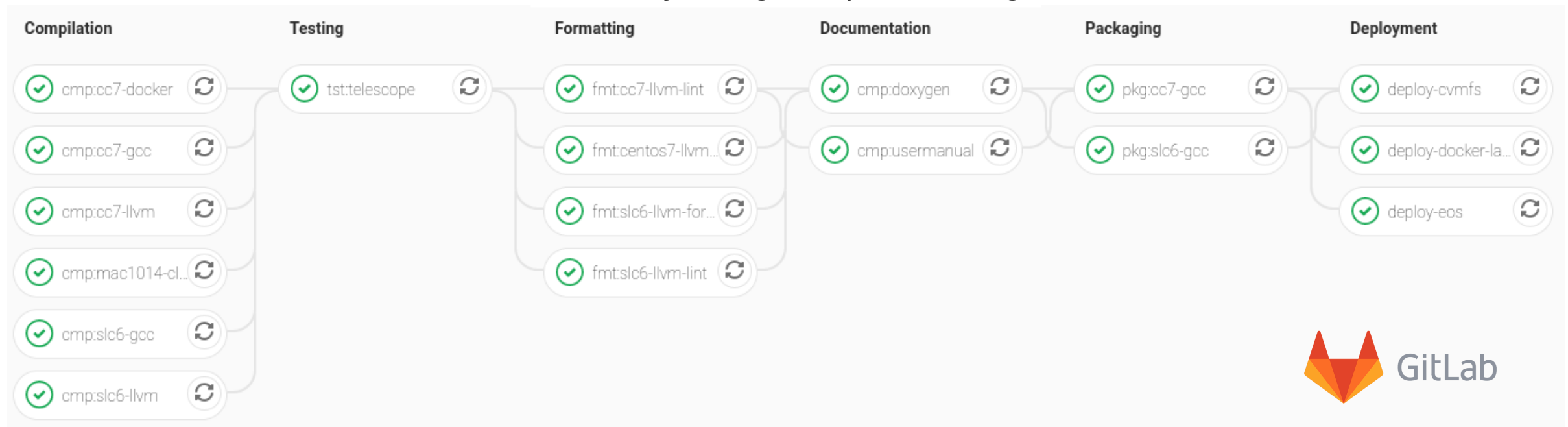


GitLab Continuous Integration

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

for every commit

for every merge request or tag



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

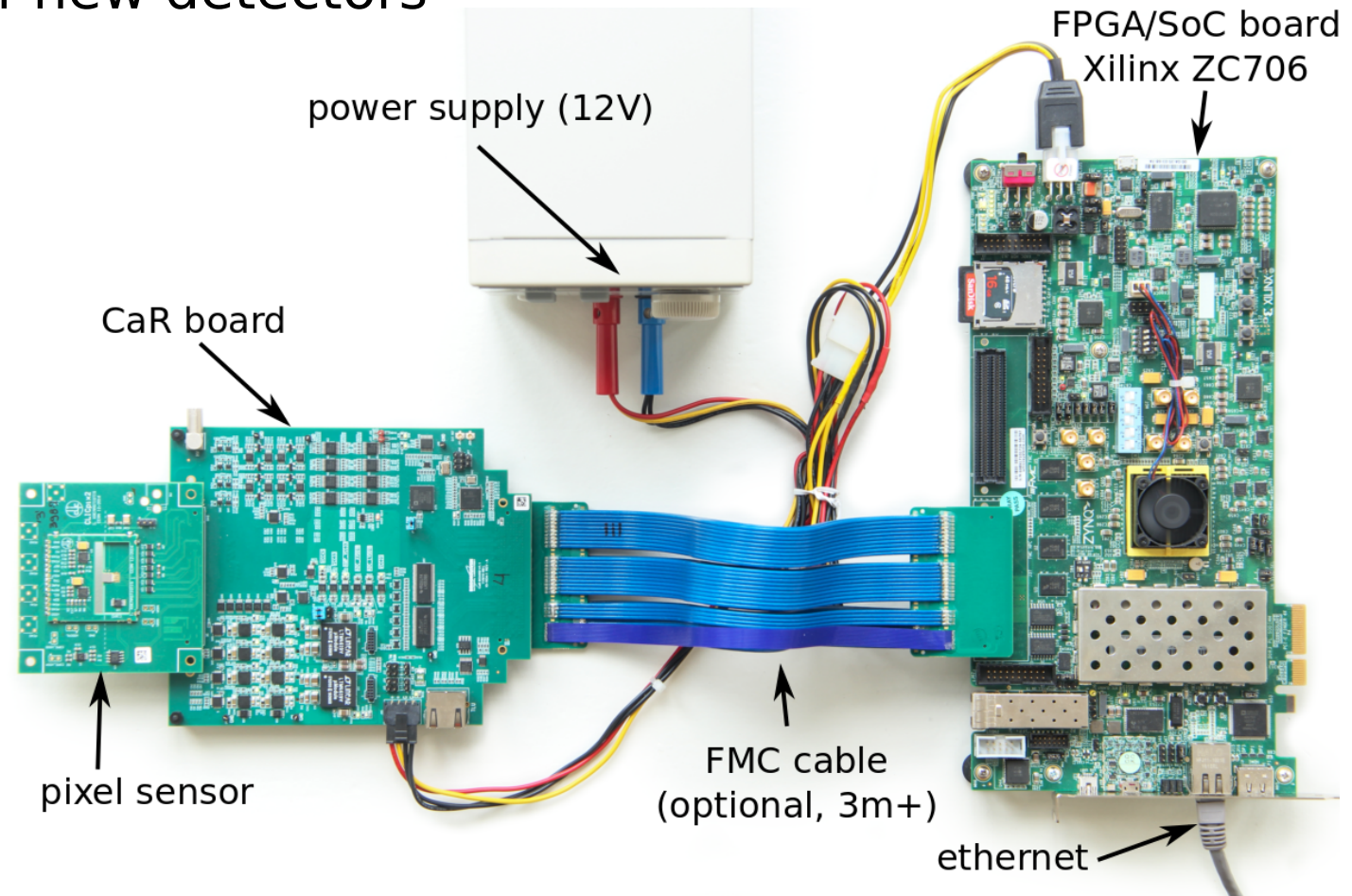
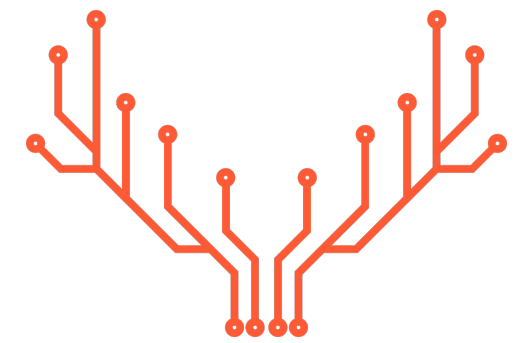
- **Deployment**

- 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

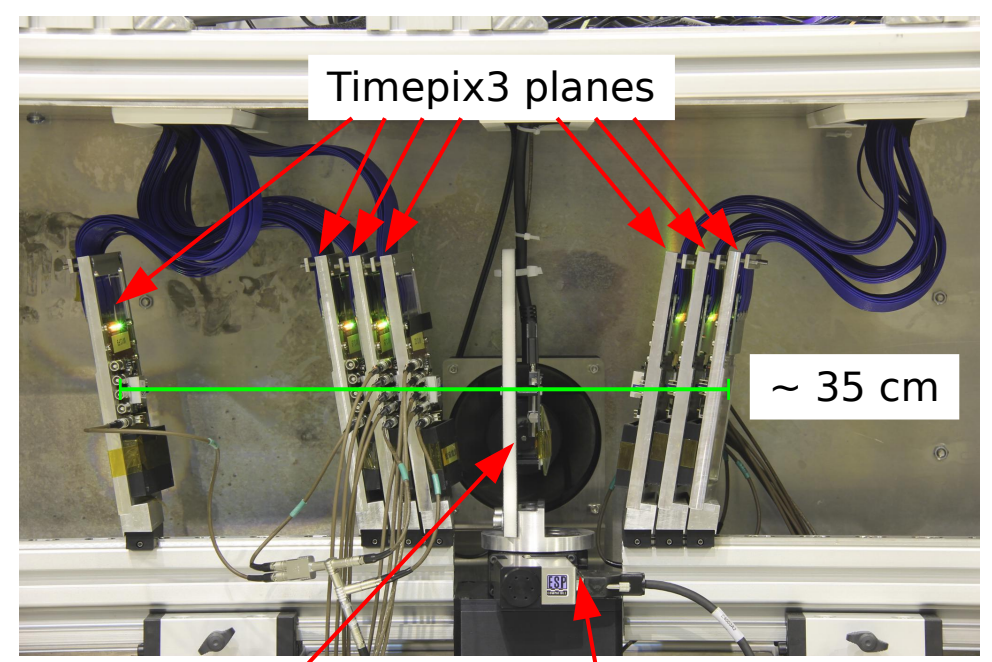
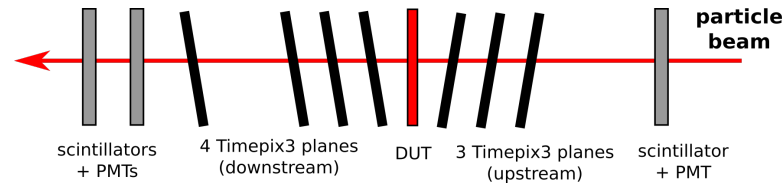
- versatile, open-source, linux-based
- fast & simple implementation of new detectors
→ “fast prototyping”
- **universal:**
 - FPGA board
 - Control & Readout (CaR) board
 - “most of the” firmware/software
- **chip-specific:**
 - chip board
 - “some” firmware/software blocks



SPS vs. DESY II

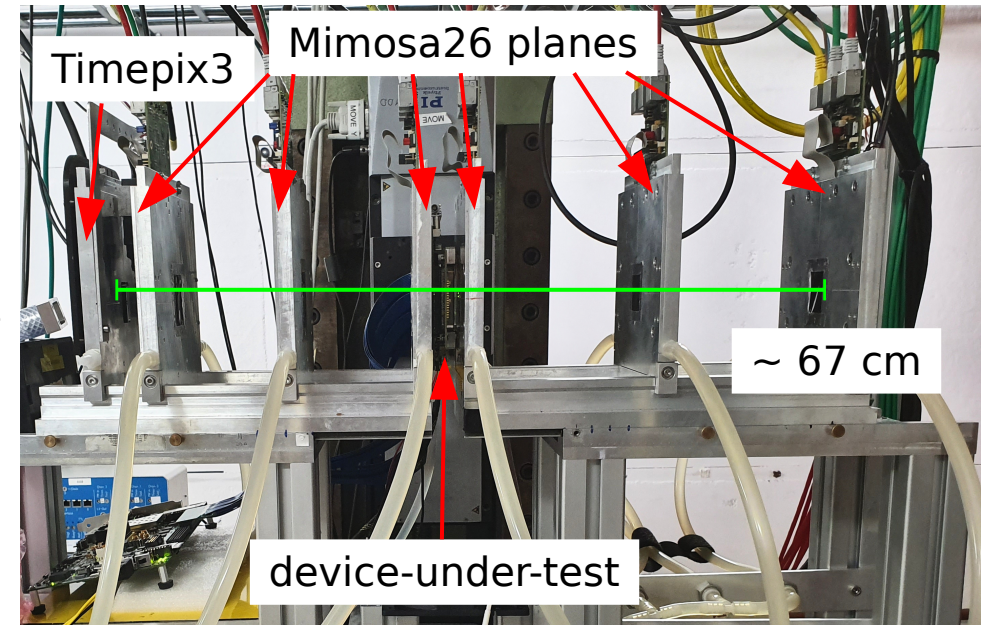
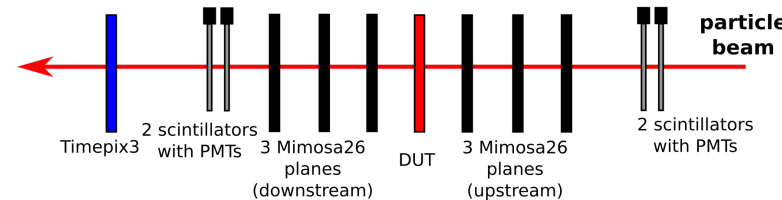
SPS:

- 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



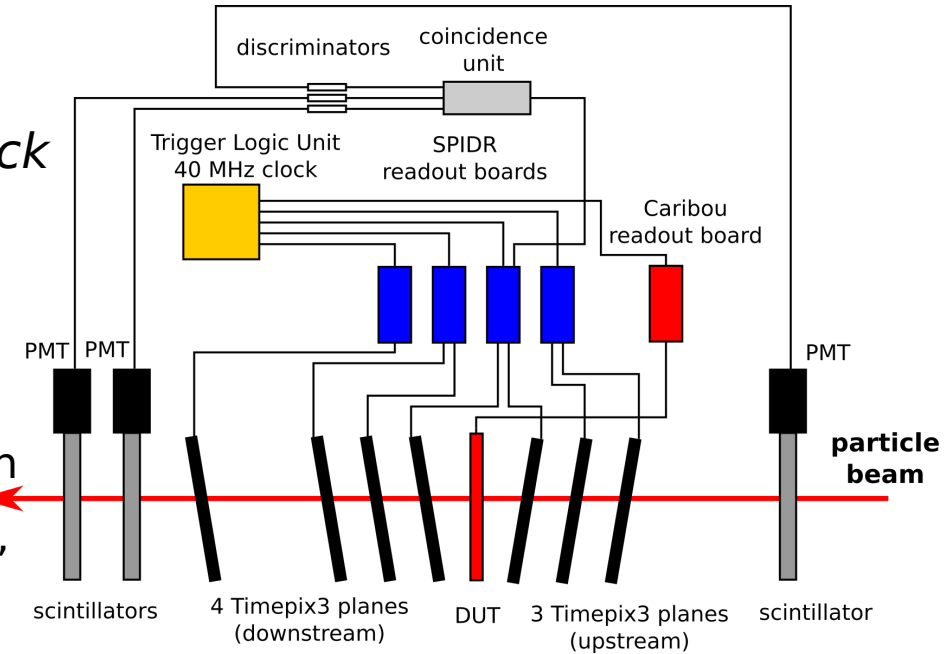
→ much lower rate & energy

SPS vs. DESY II - Readout

continuous readout, timestamps synchronous with global clock

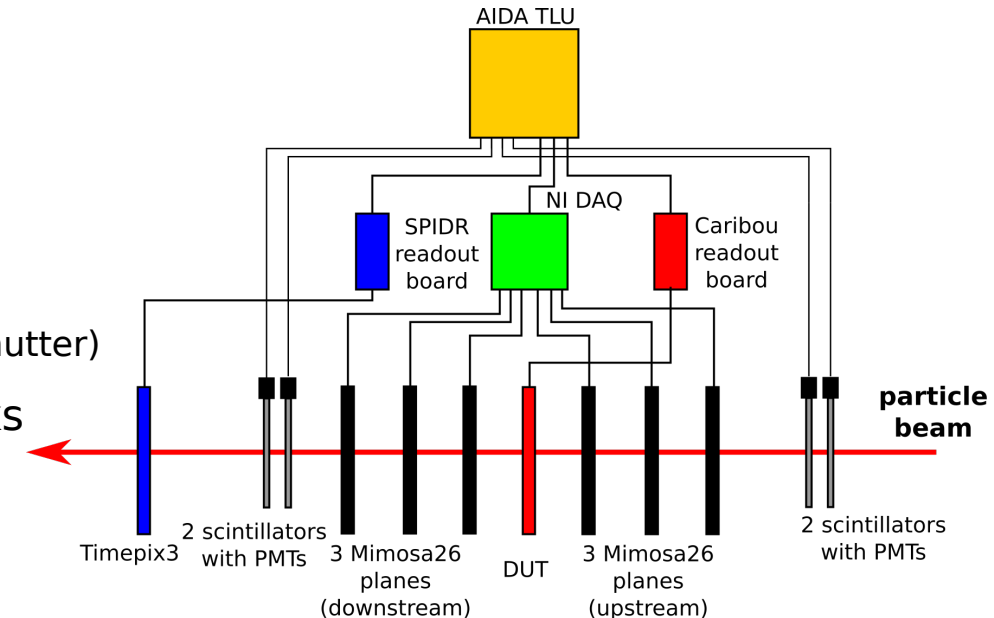
SPS:

- 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**

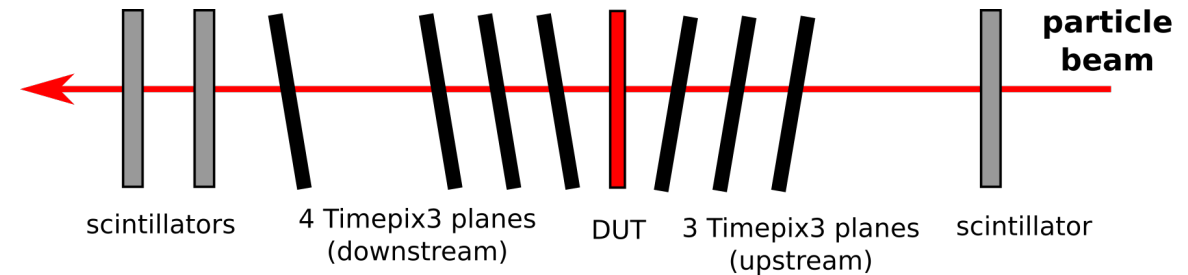
SPS vs. DESY II - Differences in the Analysis

Tracking:

- **SPS:**
 - 7 Timepix3 hits with precise timestamp
 - track timestamp = average TPX3 timestamp
- **DESY:**
 - Mimosa26 hits (3x 115 μ s) 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

