cern.ch/allpix-squared

Allpix Squared

Generic Pixel Detector Simulation Framework

v1.1

Simon Spannagel

6th BTTB Workshop

Zurich, 18/01/2018

Why?

- Wanted: simulation software, that was flexible and

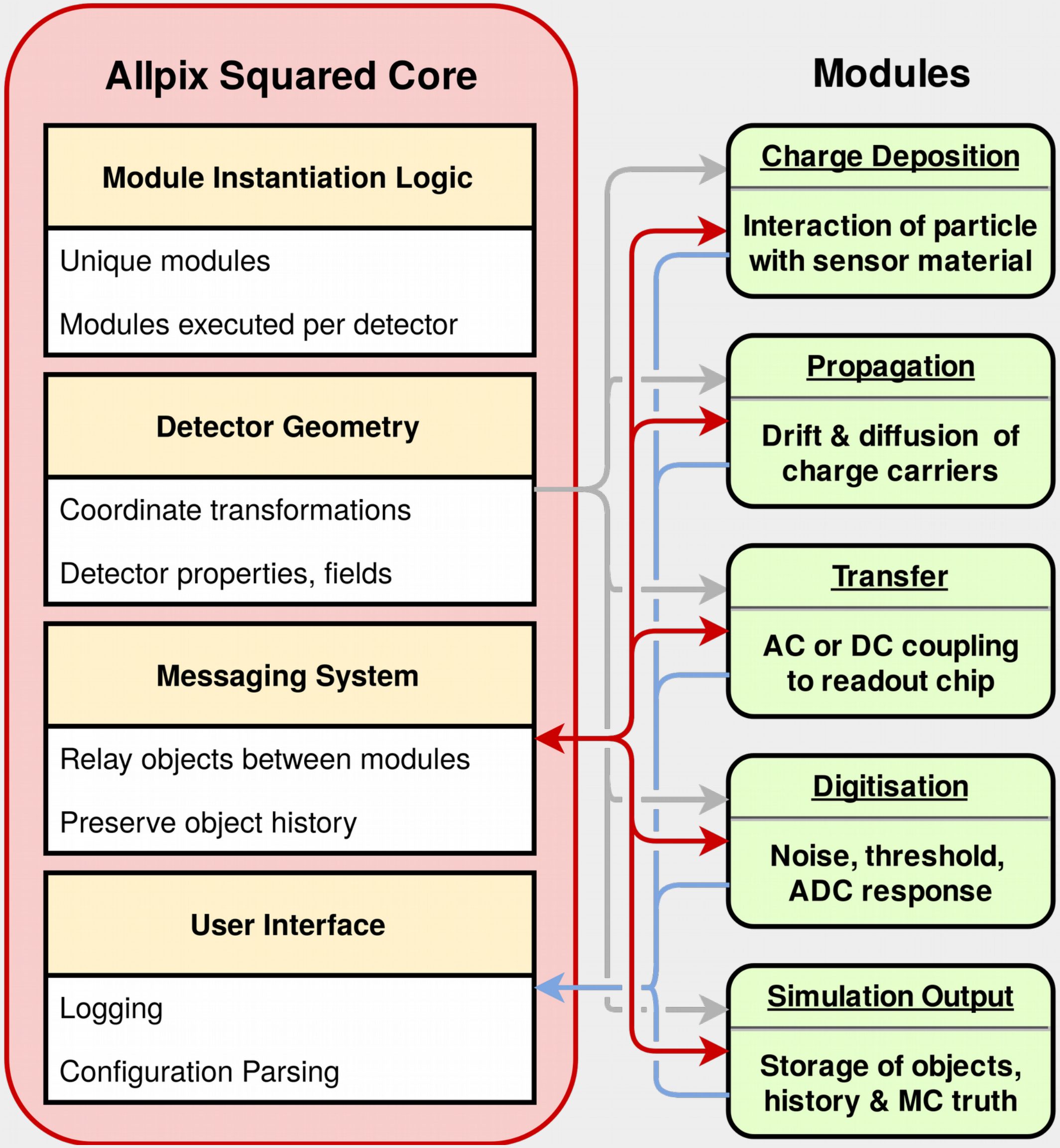
  …allowed us to **test different** simulation **models**

  …wouldn't require huge efforts to support new detectors

  …would facilitate usage of **precise electric fields**
- Many existing frameworks which cover parts of this
- Developed new software within **CLICdp collaboration**
  - Understand prototypes, optimize designs for future sensors
  - Simulate **SOI, HVCMOS, capacitively coupled hybrids, …**
- Decided to start from scratch, based on established ideas:
  - AllPix – Geant4 for deposition, parametrized detector models
  - PixelAV – Charge propagation with Runge-Kutta-Fehlberg

The Allpix Squared Framework

- Written in modern C++
- Prov. central components
  - Convenient user interface
  - Logging, configuration
  - Geometry and transformations
- Implement physics in independent modules
  - Plug & play concept
  - IO using ROOT TTrees
- Loading lib, parallelization…



```
[AllPix]
log_level = "INFO"
number_of_events = 500000
detectors_file = "telescope.conf"

[GeometryBuilderGeant4]
world_material = "air"

[DepositionGeant4]
physics_list = FTFP_BERT_LIV
particle_type = "Pi+"
number_of_particles = 1
beam_energy = 120GeV
# ...

[ElectricFieldReader]
model="linear"
bias_voltage=150V
depletion_voltage=50V

[GenericPropagation]
temperature = 293K
charge_per_step = 10
spatial_precision = 0.0025um
timestep_max = 0.5ns

[SimpleTransfer]

[DefaultDigitizer]
adc_resolution = 4
adc_slope = 1000

[DetectorHistogrammer]
name = "telescope1"
file_name = "telescope1_histograms.root"

[ROOTObjectWriter]
file_name = "simulation_data.root"
```

Configuration

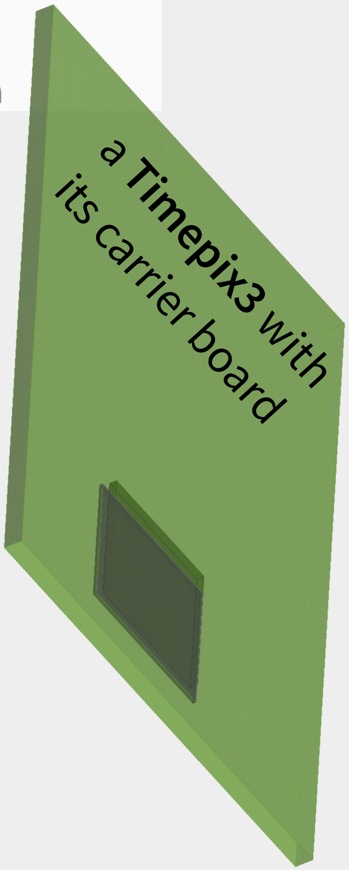
- Framework configured by one file
  - All desired modules listed in order of execution
  - Key-value pairs in TOML-style
    - Human readable
    - No overhead (as e.g. XML has)
    - Parsers for many languages
- Support for physical units
  - Never wonder again, what units are used – type them out!
  - Automatic conversion to internal units

Detector Models

- Different detector types available
  - Monolithic detectors
  - Hybrid detectors w/ bump bonds
- Easy configuration through model files
  - Give it a name, decide on the type
  - Set detector parameters
- Some model files already shipped with the framework, at the moment:

  ATLAS FE-I3, FE-I4, CMS PSI46/dig, Medipix3, Timepix3, CLICpix, CLICpix2, Mimosa23, Mimosa26

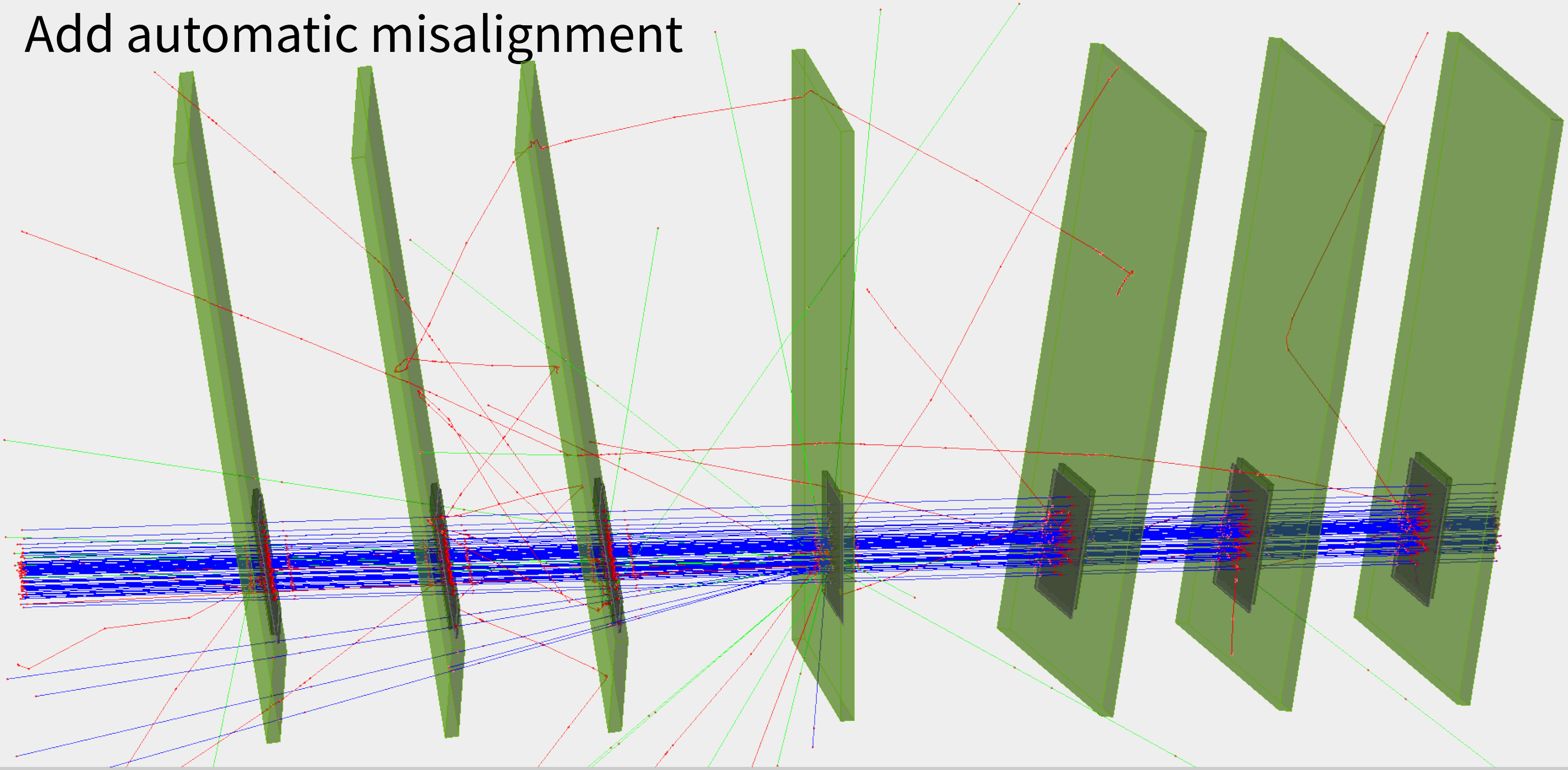
```
type = "hybrid"

number_of_pixels = 256 256
pixel_size = 55um 55um

sensor_thickness = 300um
chip_thickness = 700um

# ...

[support]
thickness = 1.76mm
```



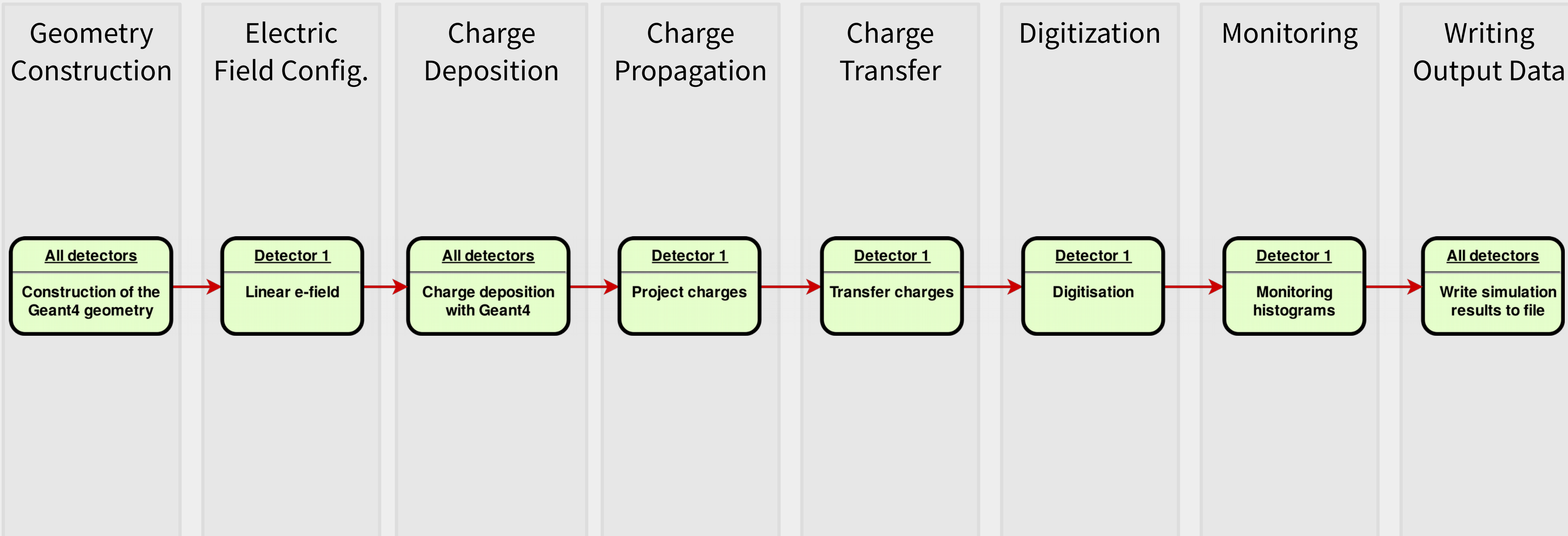
Experimental Setup

- Set of detectors in separate file
  - Using pre-defined detector models
  - Set position & orientation
- Possibility to overwrite model parameters
- Add automatic misalignment

```
[telescope0]
type = "timepix"
position = 0mm 0mm 10mm
orientation = -15deg 12deg 8deg

[telescope1]
type = "timepix"
position = 0mm 0mm 150mm
orientation = 9deg 180deg 0deg
alignment_precision_position = 50um 50um 1mm
```

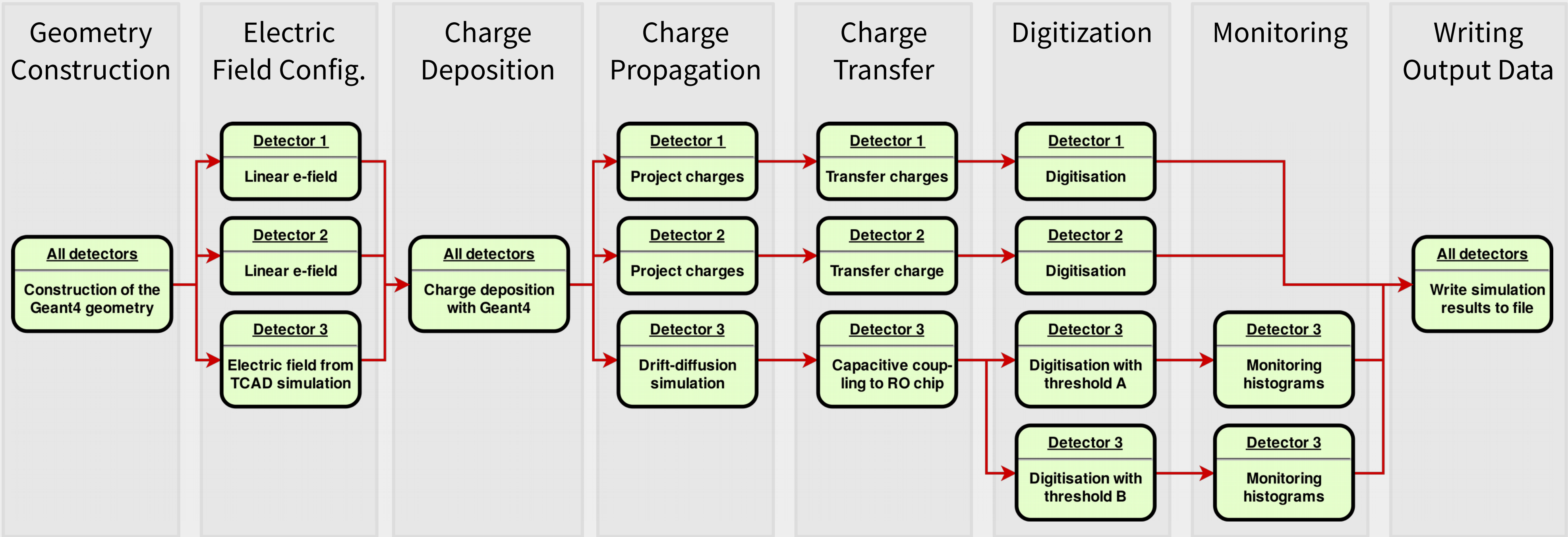


Modular Approach



- Modules are placed in linear order in the configuration file
- Select suitable deposition, propagation, ... modules
- Approach allows to quickly plug together simple simulations (start from examples shipped in the repository)

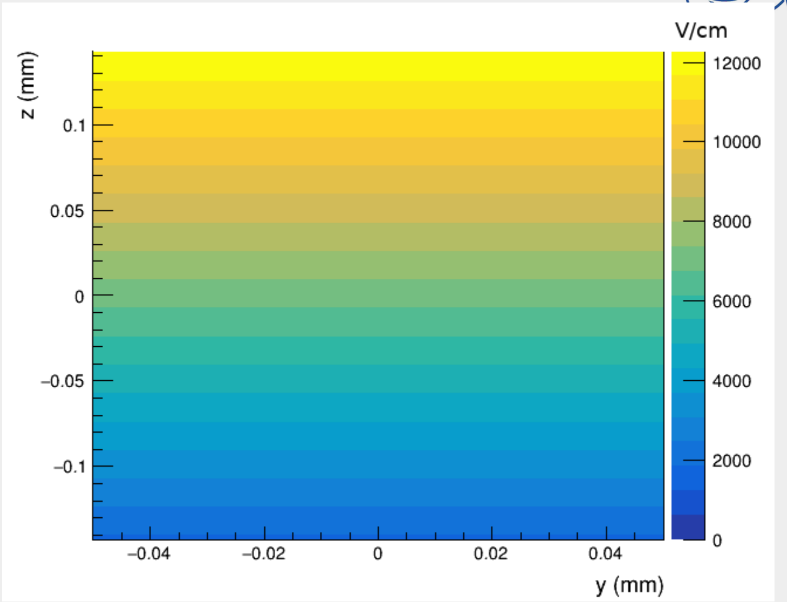
Modular Approach (II)



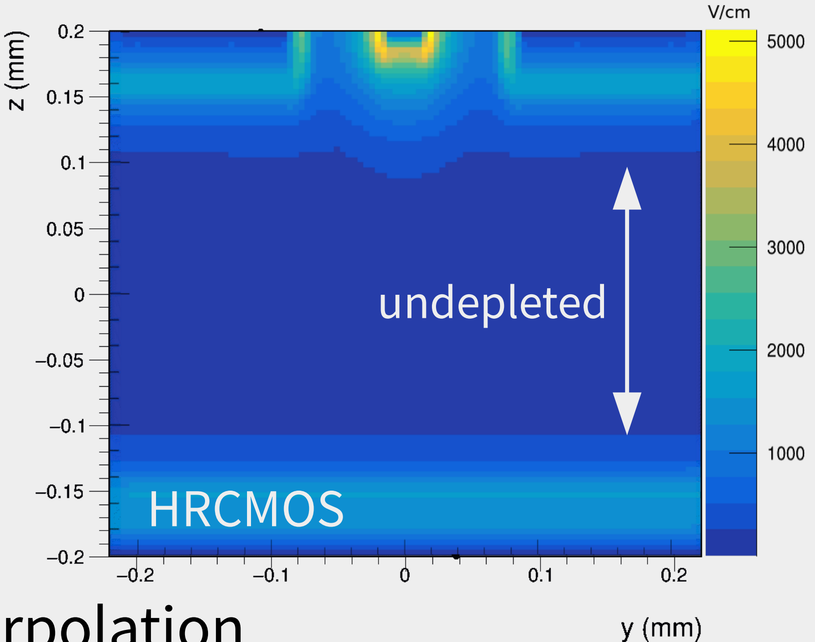
- ...as well as more involved setups
- E.g. have detailed simulation for **device under test** and quicker, less detailed simulations for **telescope planes**
- Apply different settings in the same simulation run

Electric Fields

- **Option 1: Linear electric field**
  - Good approximation for thick, planar sensors
  - Defined by setting
    - Depletion + bias voltage

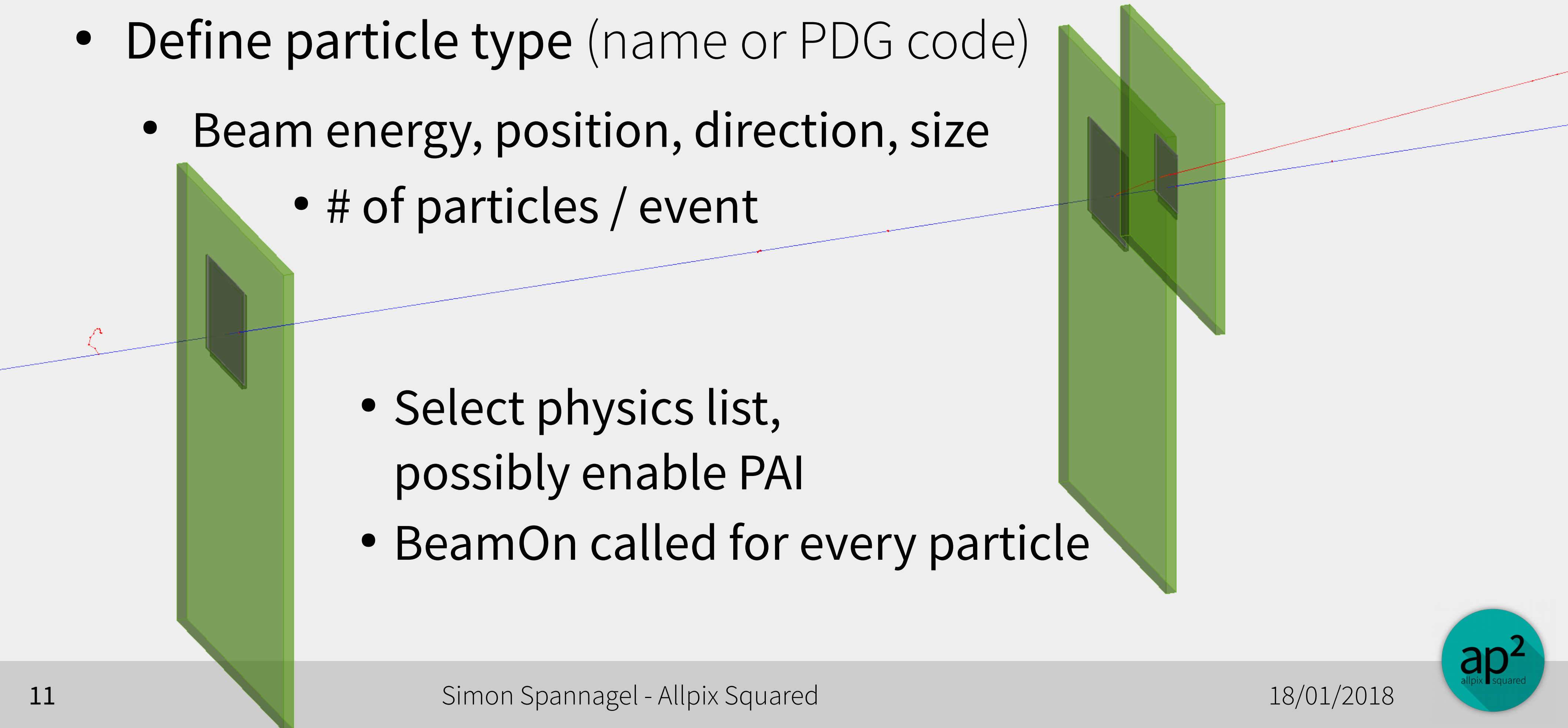


- **Option 2: Electric field from TCAD**
  - Detailed e-static simulation, drift along field lines
  - **TCAD DF-ISE Mesh Converter**
    - Converts adaptive to regular
    - 3D, 2D quantities, barycentric interpolation



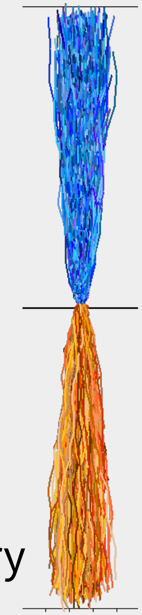
Charge Deposition: Geant4

- Dependency on Geant4 only on module level
  - Framework can be built without Geant4
  - Geant4 interaction abstracted by module
- Define particle type (name or PDG code)
  - Beam energy, position, direction, size
    - # of particles / event
  - Select physics list, possibly enable PAI
  - BeamOn called for every particle

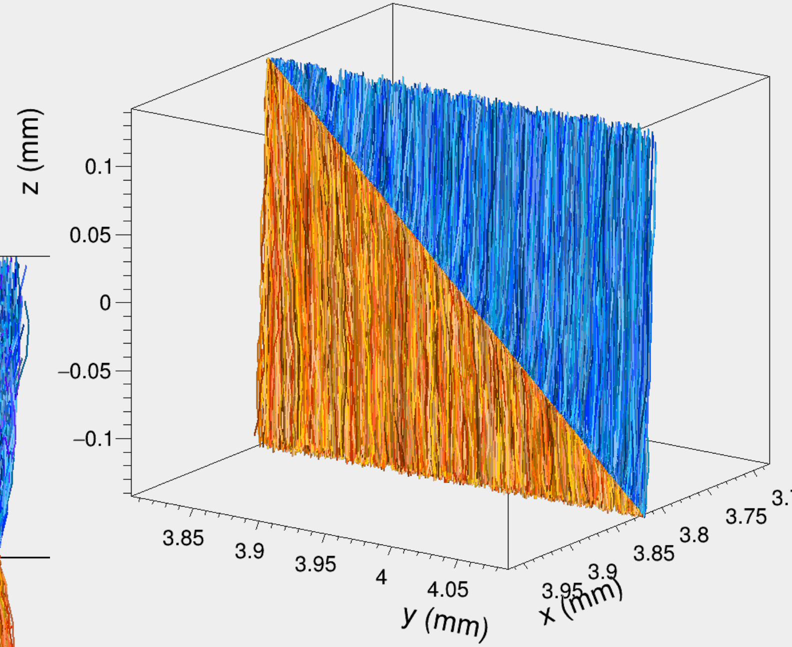


Charge Propagation

- Pick up deposited charge carriers, transport to implants
- **Option 1: Projection**
  - Project charges on implant side, adding diffusion
  - Only for linear fields, integrating
- **Option 2: Drift-Diffusion model**
  - Drift along field lines using 4th order Runge-Kutta-Fehlberg method
  - Transient information available
  - Allows to create drift animation
    - See website



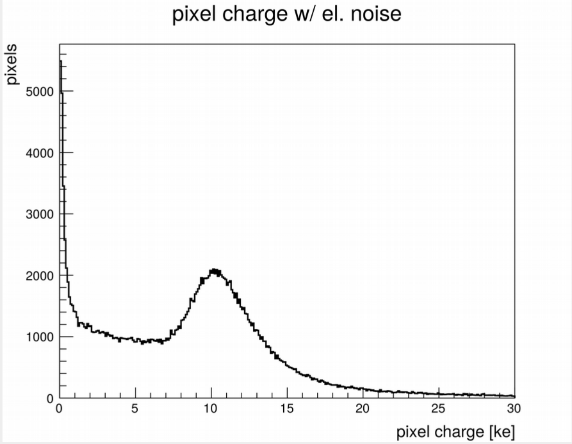
projection along particle trajectory

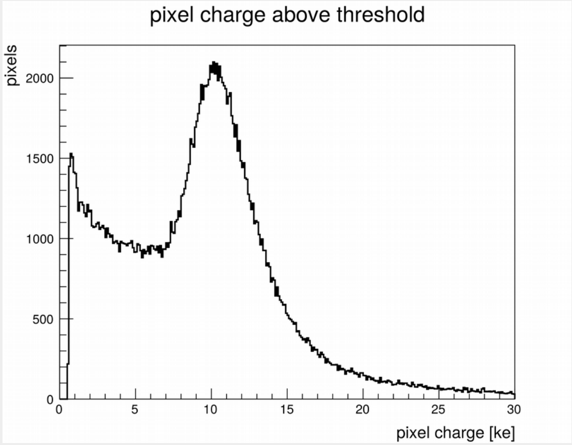


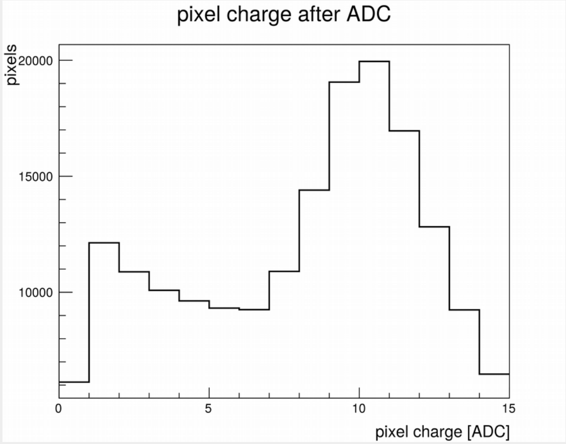
particle crossing sensor at 45° angle
drift paths of electrons & holes

Digitization

- Simulation of readout chip circuitry
- Different (optional) stages:
  - Electronics noise
  - Amplifier gain & gain smearing
  - Threshold & smearing
  - ADC resolution & smearing
  - ADC calibration (currently: linear)
- Monitoring plots for each step produced

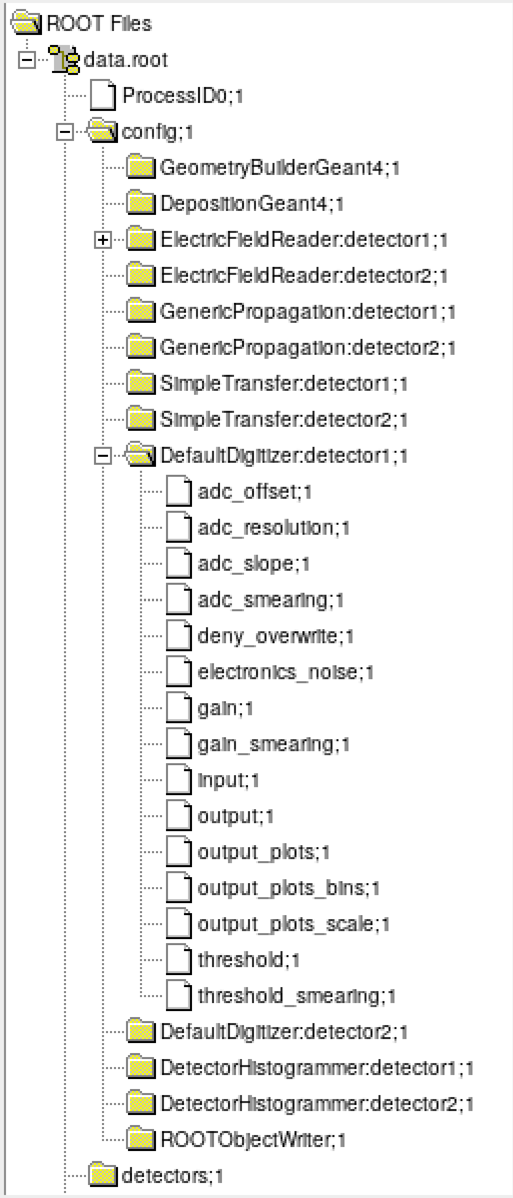






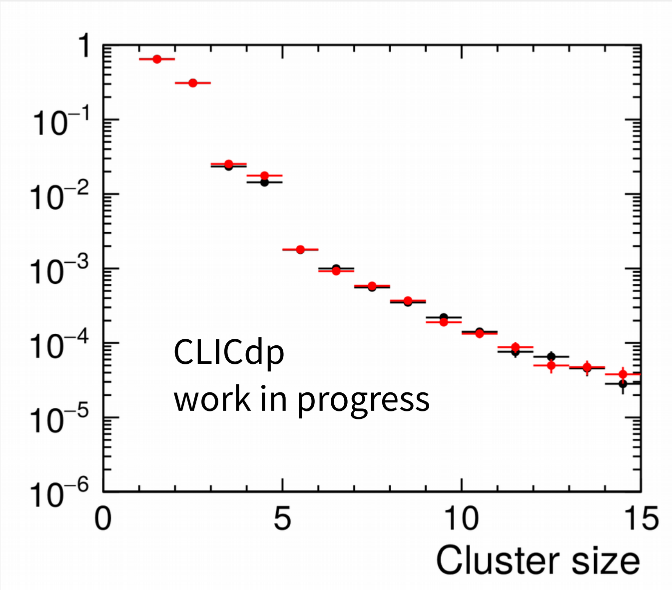
Data Storage

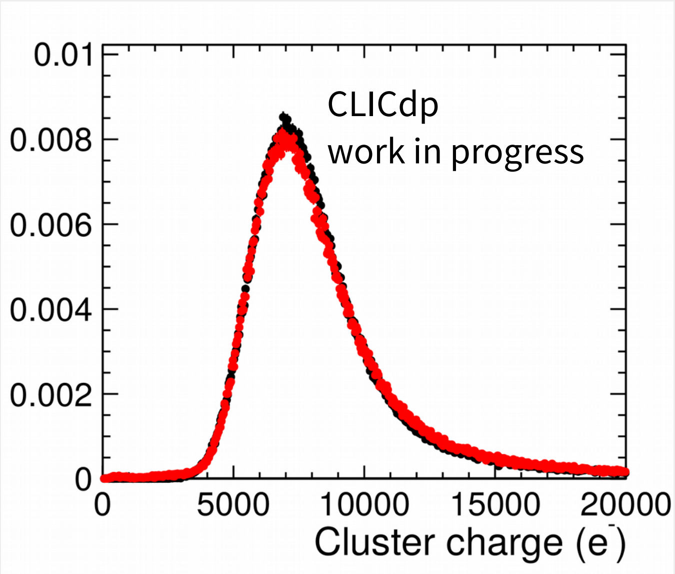
- Different output writers available:
  - LCIO (EUTelescope), RCE (Proteus), ...
- Native format: ROOT files with all objects
  - Also contains detectors & sim. parameters
  - Full framework **configuration** can be **reconstructed from** single **data file**
- ROOTObject reader replays data from file
  - Simulate deposition & propagation once
  - Read data from file and **quickly repeat** final digitization **step** with different parameters

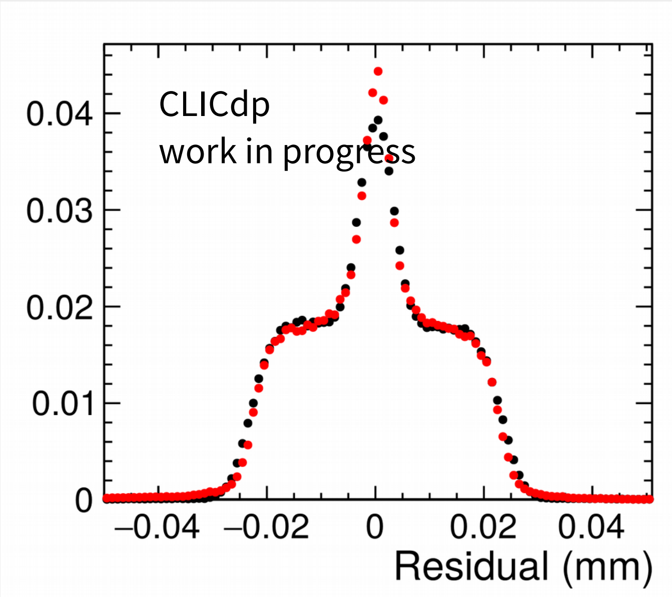


Comparison – Work in Progress

- **CLICdp** Timepix3 Telescope + DUT: **100µm planar sensor**
- "simply" simulated device with parameters taken from data
  - Bias/depletion voltages, temperature, threshold
- Reconstruction with same cuts & corrections (e.g. eta-corr)



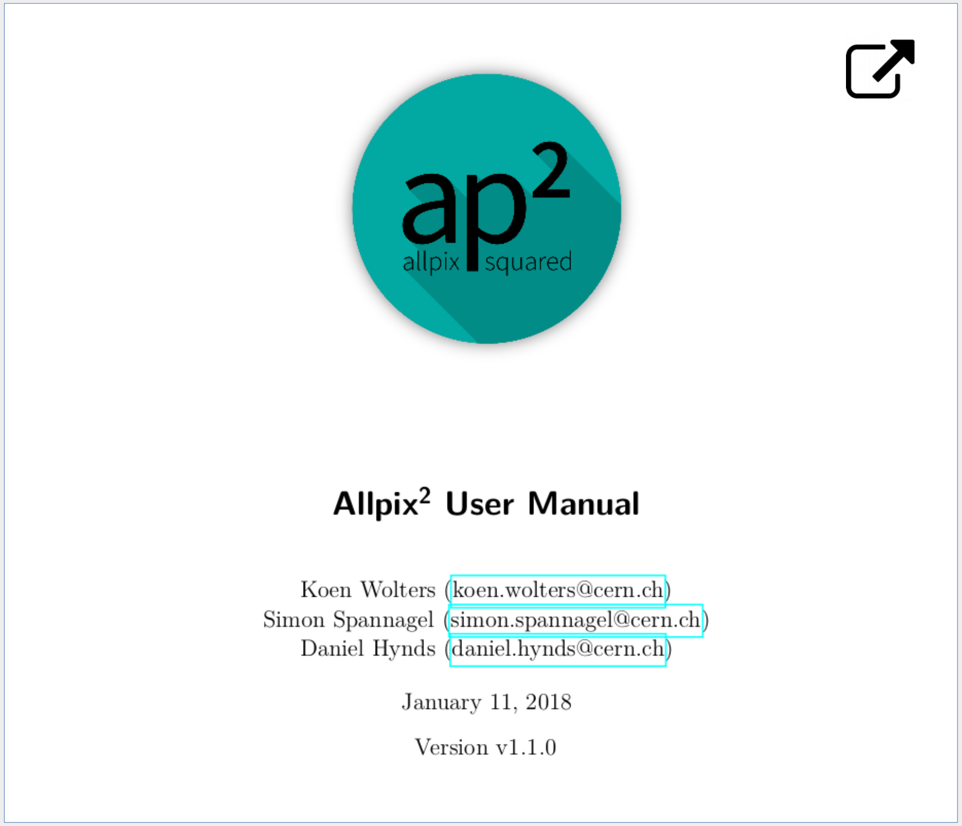




Data / **Allpix²**

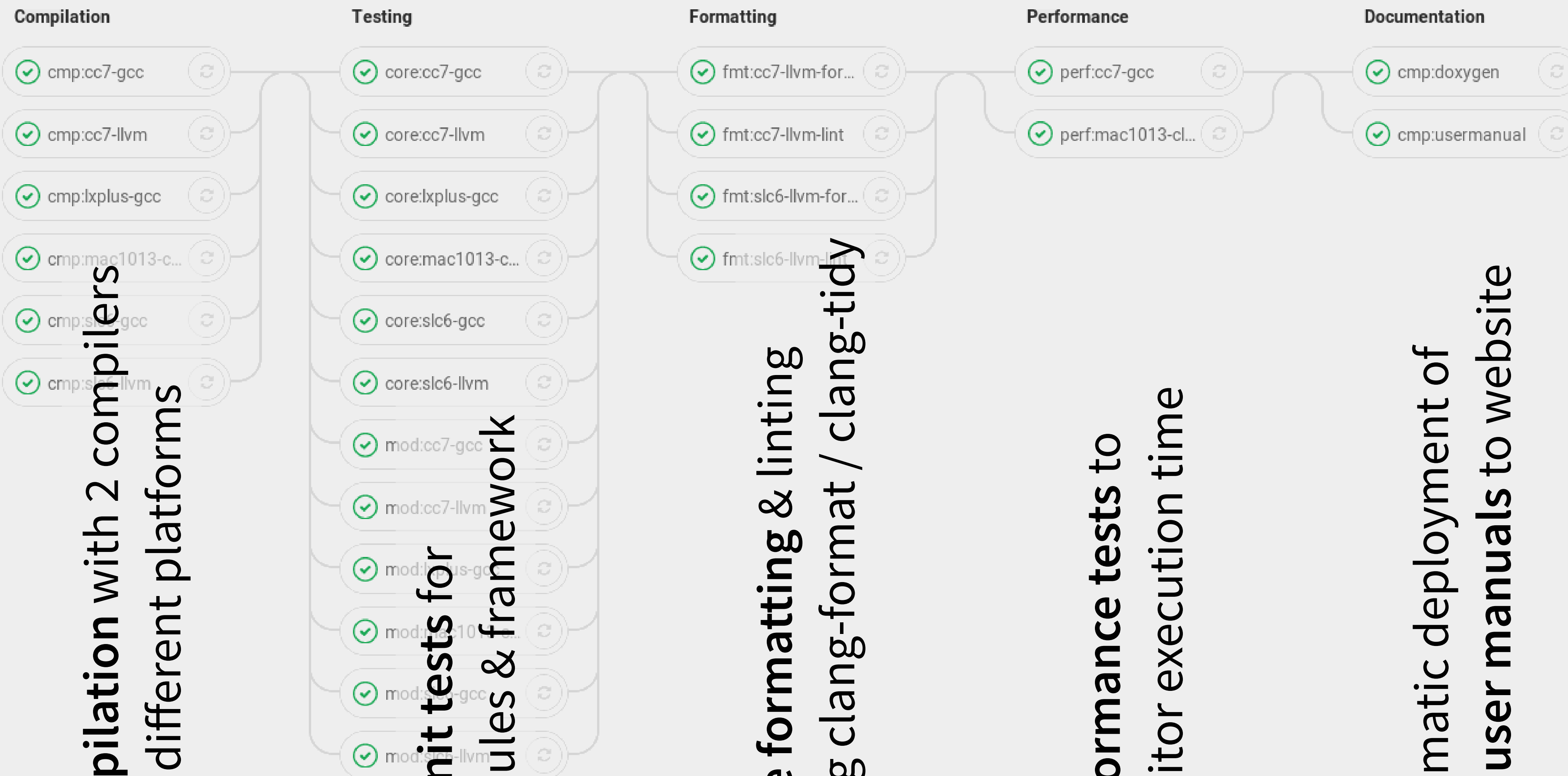
Documentation & Manuals

- Extensive User Manual ~115 pages **(PDF/TeX)**



- Well-documented code **(Doxygen)**
- Module documentation **(Markdown)**

Continuous Integration & Testing



compilation with 2 compilers on 3 different platforms

32 **unit tests** for modules & framework

code formatting & linting using clang-format / clang-tidy

performance tests to monitor execution time

automatic deployment of new **user manuals** to website

Summary & Outlook

- Generic, modular framework to simulate pixel detectors
  - Reads TCAD e-fields, implements drift-diffusion model
  - Modern, documented code & extensive user manual
- First comparison with test beam data
- Version 1.1 released last week
  - New module, many improvements to framework
  - Check release notes for details
    https://cern.ch/allpix-squared/post/2018-01-12-allpix-squared-1.1-released/
- Welcoming **contributions from the community**
- **Tutorial** this afternoon

Resources

- Website
  https://cern.ch/allpix-squared
  - Release Notes, Information, User Manual, Code Reference
- Repository
  https://gitlab.cern.ch/simonspa/allpix-squared
  - Source Code, Issue Tracker
- Mailing Lists:
  - allpix-squared-users https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10262858
  - allpix-squared-developers https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10273730
- User Manual:
  https://cern.ch/allpix-squared/usermanual/allpix-manual.pdf

CERN

Automatic Detector Misalignment

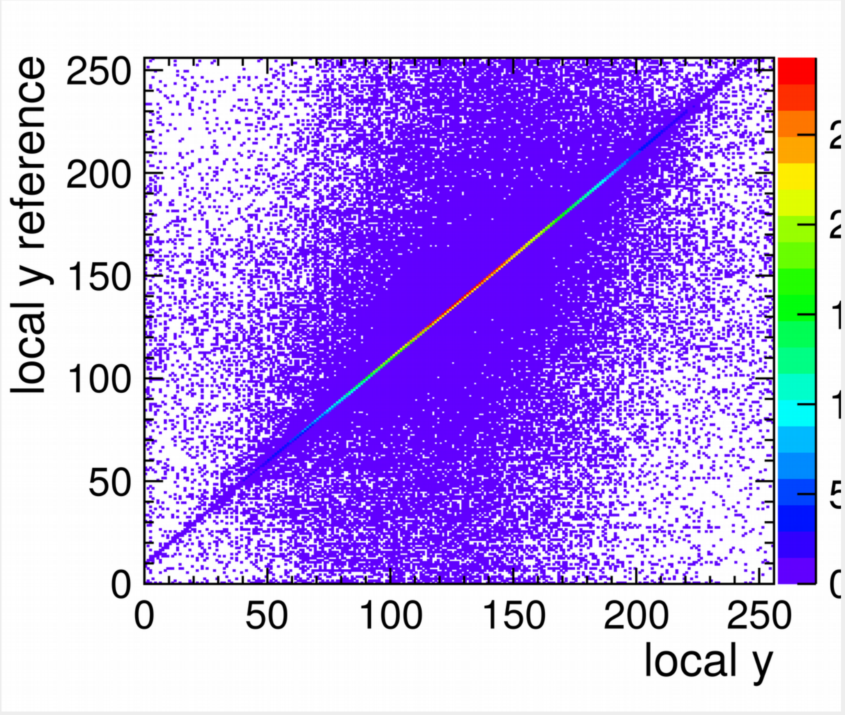
- Artifacts in reconstructed simulation residuals
  - From pixel-perfect alignment, cured by misaligning detectors
- New parameters for detectors:

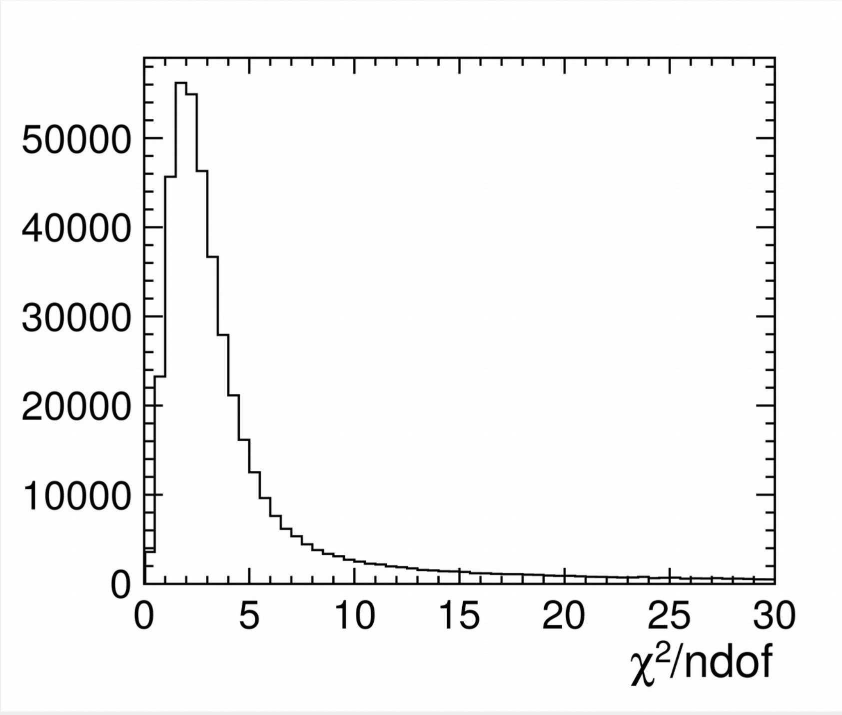
```

alignment_precision_position = 0.1mm 0.1mm 1mm
alignment_precision_orientation = 0.1deg 1.3deg 2deg

```

  - To be added in detectors file for each detector individually
  - Parameters define Gaussian width to draw shifts from
- Reproducible misalignments:
  - Set **`random_seed_core`** to known value

Reconstruction

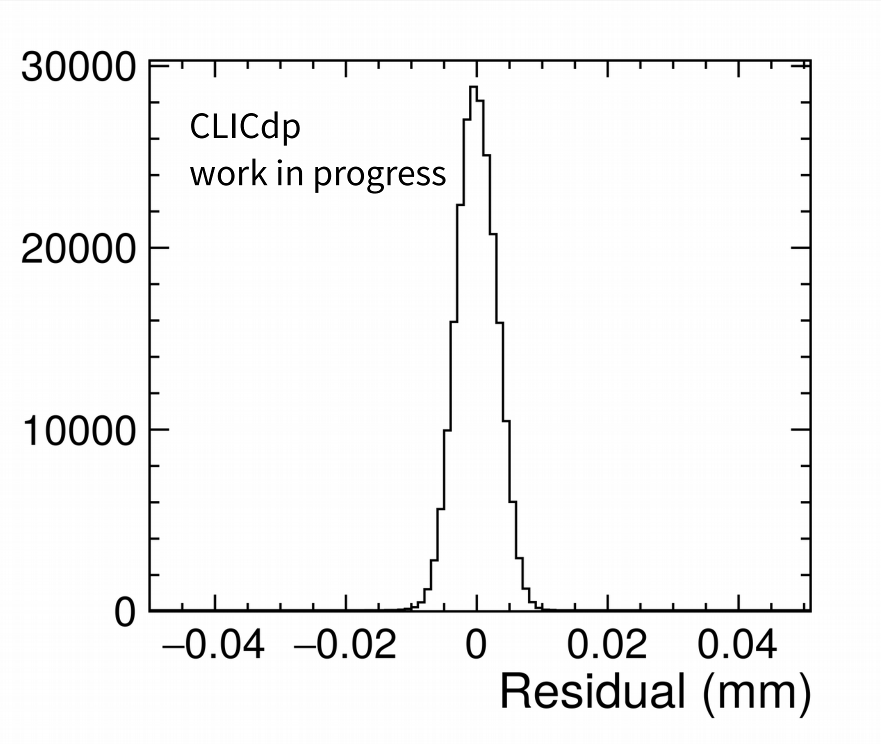
- Framework allows to simulate full setup, not only DUT
  - Beam telescopes, source measurements...
- Possible to use different methods
  - Precise simulation in DUT
  - More coarse simulation for telescope planes

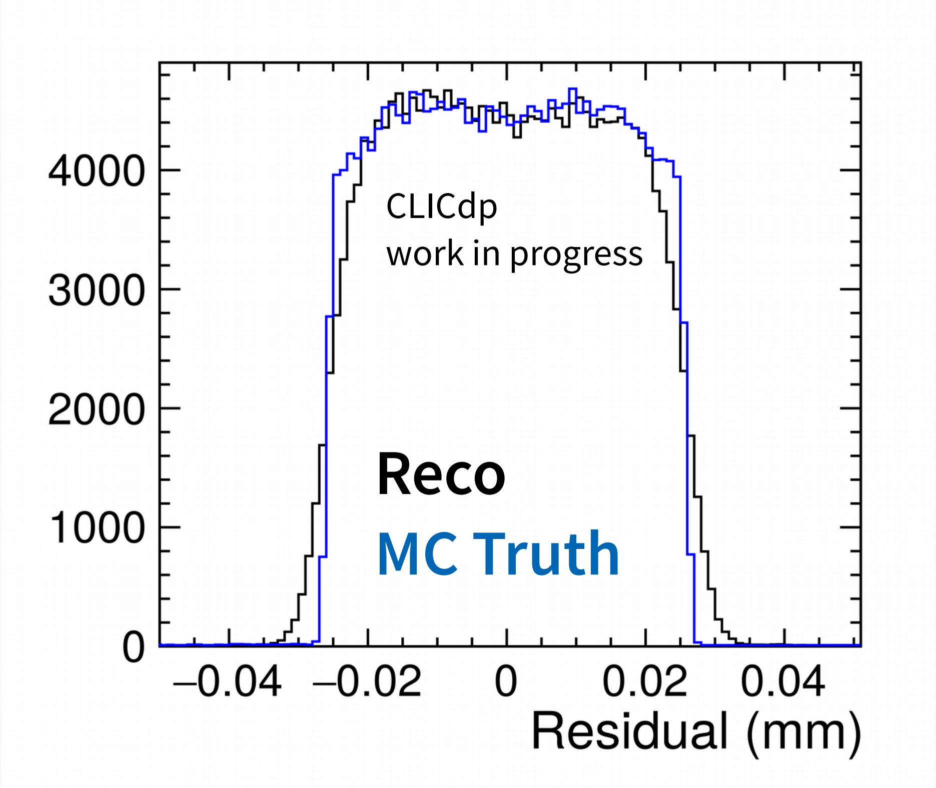




Monte Carlo Truth Information

- Monte Carlo Truth information available
  - Entry & exit point of primary and secondary particles
- Compare tracks reconstructed from telescope simulation with MC truth





Module Tests

```
Test project /builds/simonspa/allpix-squared/build
      Start 15: test_modules/test_08-1_writer_root.conf
      Start  3: test_modules/test_01_geobuilder.conf
      Start  4: test_modules/test_02-1_electricfield_linear.conf
      Start  5: test_modules/test_02-2_electricfield_init.conf
 1/18 Test  #4: test_modules/test_02-1_electricfield_linear.conf .....   Passed    1.10 sec
      Start  6: test_modules/test_03-1_deposition.conf
 2/18 Test  #3: test_modules/test_01_geobuilder.conf ................   Passed    1.30 sec
 3/18 Test  #5: test_modules/test_02-2_electricfield_init.conf .......   Passed    1.40 sec
      Start  7: test_modules/test_03-2_deposition_mc.conf
      Start  8: test_modules/test_04-1_propagation_project.conf
 4/18 Test #15: test_modules/test_08-1_writer_root.conf .............   Passed    4.67 sec
      Start  9: test_modules/test_04-2_propagation_generic.conf
 5/18 Test  #6: test_modules/test_03-1_deposition.conf ..............   Passed    4.27 sec
 6/18 Test  #8: test_modules/test_04-1_propagation_project.conf ......   Passed    4.07 sec
 7/18 Test  #7: test_modules/test_03-2_deposition_mc.conf ...........   Passed    4.11 sec
      Start 10: test_modules/test_05_transfer_simple.conf
      Start 11: test_modules/test_06-1_digitization_charge.conf
      Start 12: test_modules/test_06-2_digitization_adc.conf
 8/18 Test #12: test_modules/test_06-2_digitization_adc.conf .........   Passed    3.65 sec
 9/18 Test  #9: test_modules/test_04-2_propagation_generic.conf ......   Passed    4.20 sec
10/18 Test #10: test_modules/test_05_transfer_simple.conf ...........   Passed    3.68 sec
11/18 Test #11: test_modules/test_06-1_digitization_charge.conf ......   Passed    3.68 sec
      Start 13: test_modules/test_06-3_digitization_gain.conf
      Start 14: test_modules/test_07_histogramming.conf
      Start 16: test_modules/test_08-2_writer_rce.conf
      Start 17: test_modules/test_08-3_writer_lcio.conf
12/18 Test #13: test_modules/test_06-3_digitization_gain.conf ........   Passed    3.77 sec
13/18 Test #17: test_modules/test_08-3_writer_lcio.conf .............   Passed    3.77 sec
14/18 Test #14: test_modules/test_07_histogramming.conf .............   Passed    3.77 sec
      Start 18: test_modules/test_08-4_writer_corryvreckan.conf
      Start 19: test_modules/test_08-5_writer_corryvreckan_mc.conf
      Start 20: test_modules/test_09_reader_root.conf
15/18 Test #16: test_modules/test_08-2_writer_rce.conf ..............   Passed    4.18 sec
16/18 Test #20: test_modules/test_09_reader_root.conf ...............   Passed    0.92 sec
17/18 Test #18: test_modules/test_08-4_writer_corryvreckan.conf ......   Passed    3.69 sec
18/18 Test #19: test_modules/test_08-5_writer_corryvreckan_mc.conf ...   Passed    3.69 sec

100% tests passed, 0 tests failed out of 18
```

- Tests for every module
- Fixed random seed
  - Reproduces same output
- Matching regular expressions
- Single change (1e difference) fails the CI
  - Invaluable for monitoring framework
  - Catching issues before merging code

Adding New Unit Tests

- Very simple: **only requires configuration file**!
- Regular expression defines test condition
- More elaborate description to be found in the user manual

```
[Allpix]
detectors_file = "detector_rotate_misaligned.conf"
log_level = "TRACE"
number_of_events = 0
random_seed = 0
random_seed_core = 0

[GeometryBuilderGeant4]

#PASS (DEBUG)  misaligned: (8.72466deg,171.099deg,178.504deg)
#PASSOSX (DEBUG)  misaligned: (9.04007deg,170.886deg,178.731deg)
```