Center for Frontiers in Nuclear Science

sPHENIX

Data Reconstruction for the sPHENIX experiment

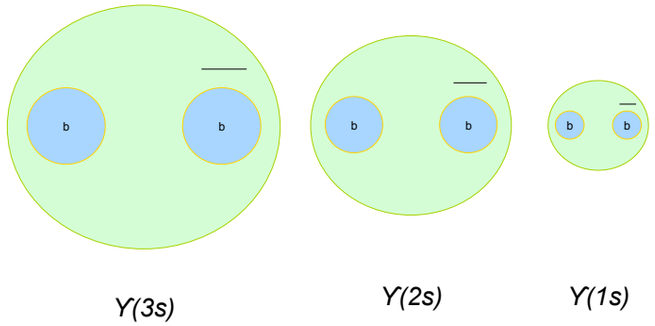
Ross Corliss
for the sPHENIX Collaboration

Stony Brook University

sPHENIX at a Glance

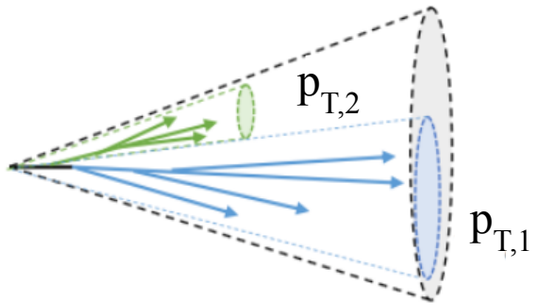
Quarkonium Spectroscopy

vary size of probe



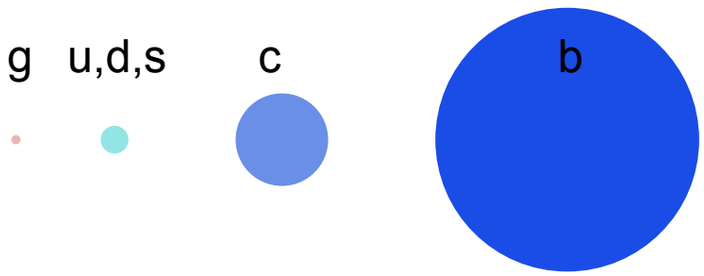
Jet Structure

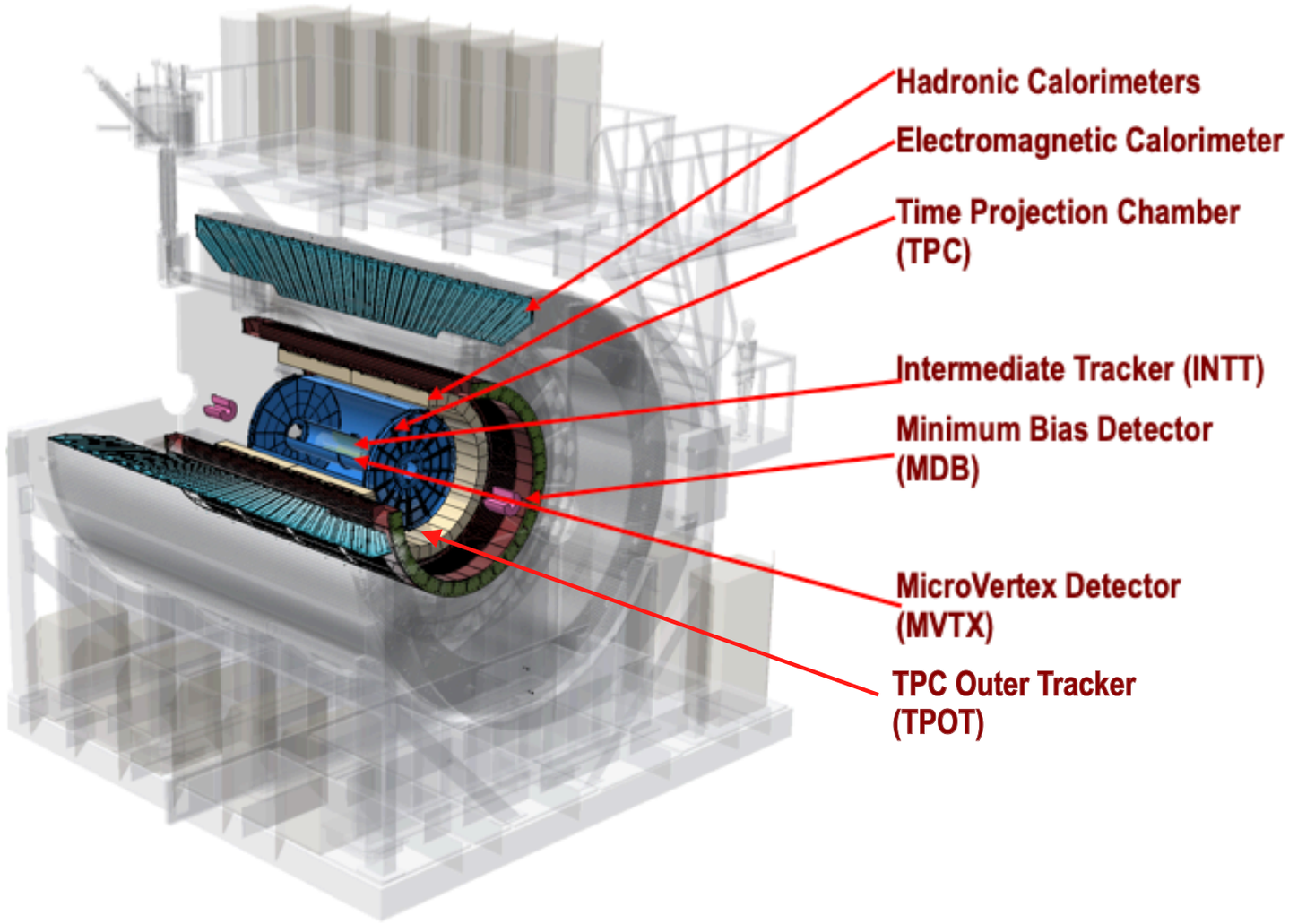
vary momentum/angular scale of probe



Parton Energy Loss

vary mass/momentum of probe





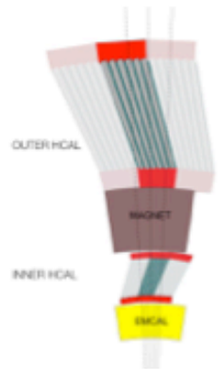
- 3-year run plan:
  - 2023: Commissioning and Au+Au physics
  - 2024: p+p and p+Au reference data
  - 2025: Au+Au high statistics production

See Murad Sarsour's slides (July 7) for details

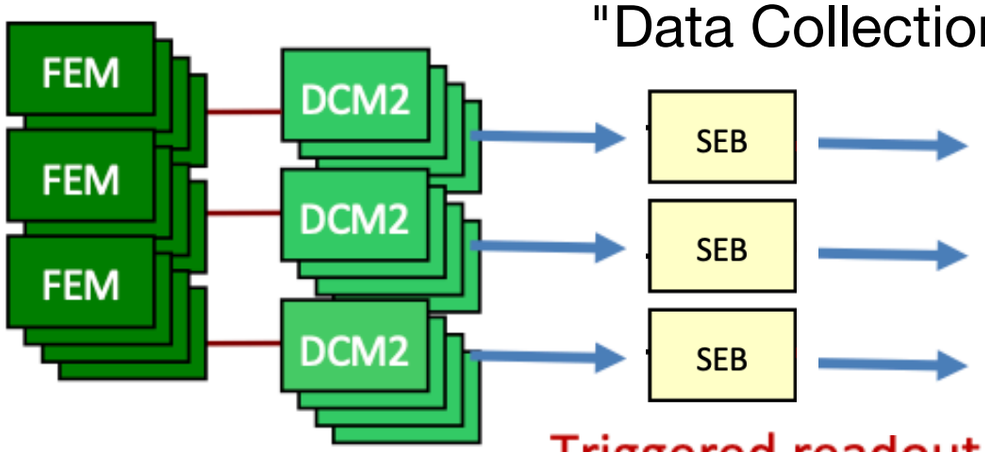
sPHENIX Data Environment

- Designed for **unbiased sampling** of Au+Au to avoid trigger effects in jets and collect untriggerable observables
- Maximize data-taking before EIC construction begins:
  - 15kHz readout, tracking detectors streaming
  - Near-realtime offline event reconstruction
  - Computing resources in each year will be needed for that run -- limited opportunity to revisit/reprocess raw data from earlier runs
- RHIC is mature: Delivered luminosity will be high on day one.

DAQ Overview

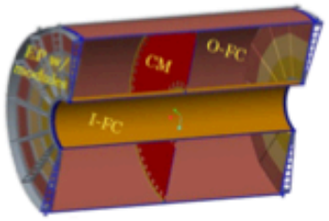


HCALs, ECAL, MBD

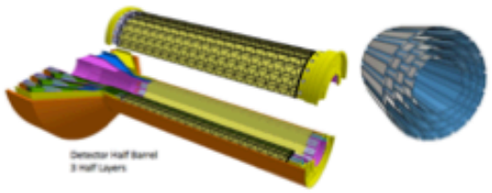


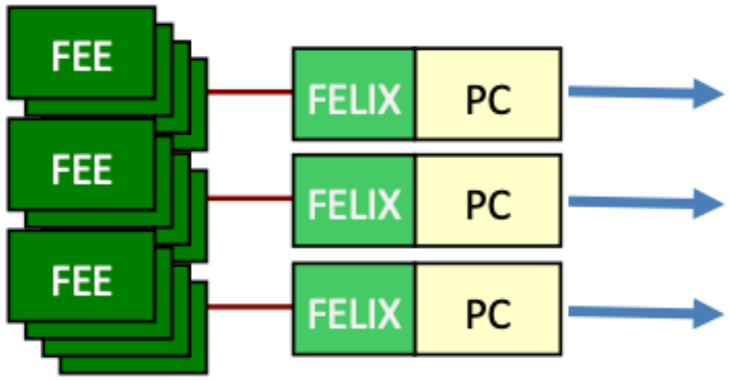
Triggered readout

The calorimeters and the MBD re-use the PHENIX "Data Collection Modules" (v2)



TPC, TPOT, MVTX, INTT





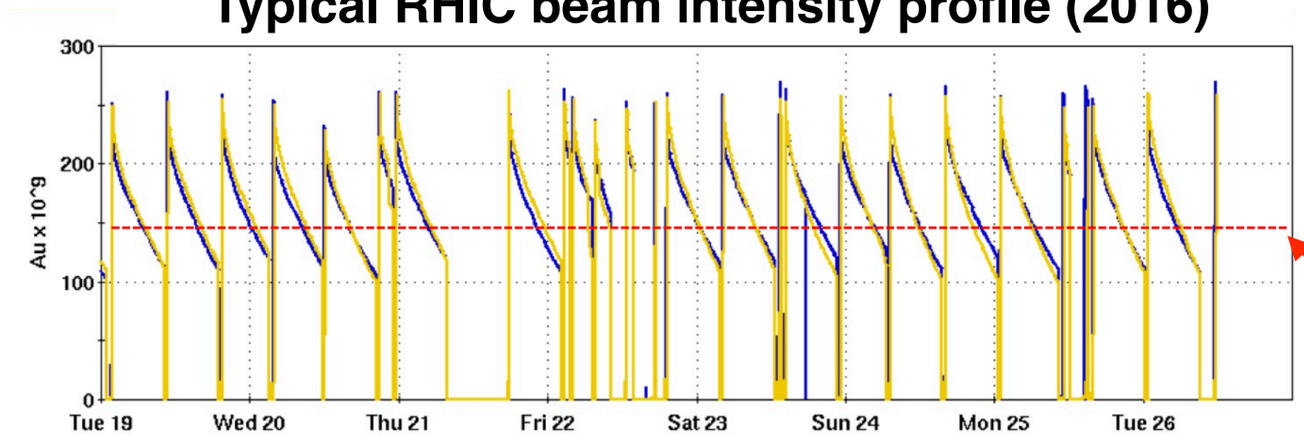
Streaming readout

The tracking detectors are read out through the ATLAS "FELIX" card directly into a standard PC

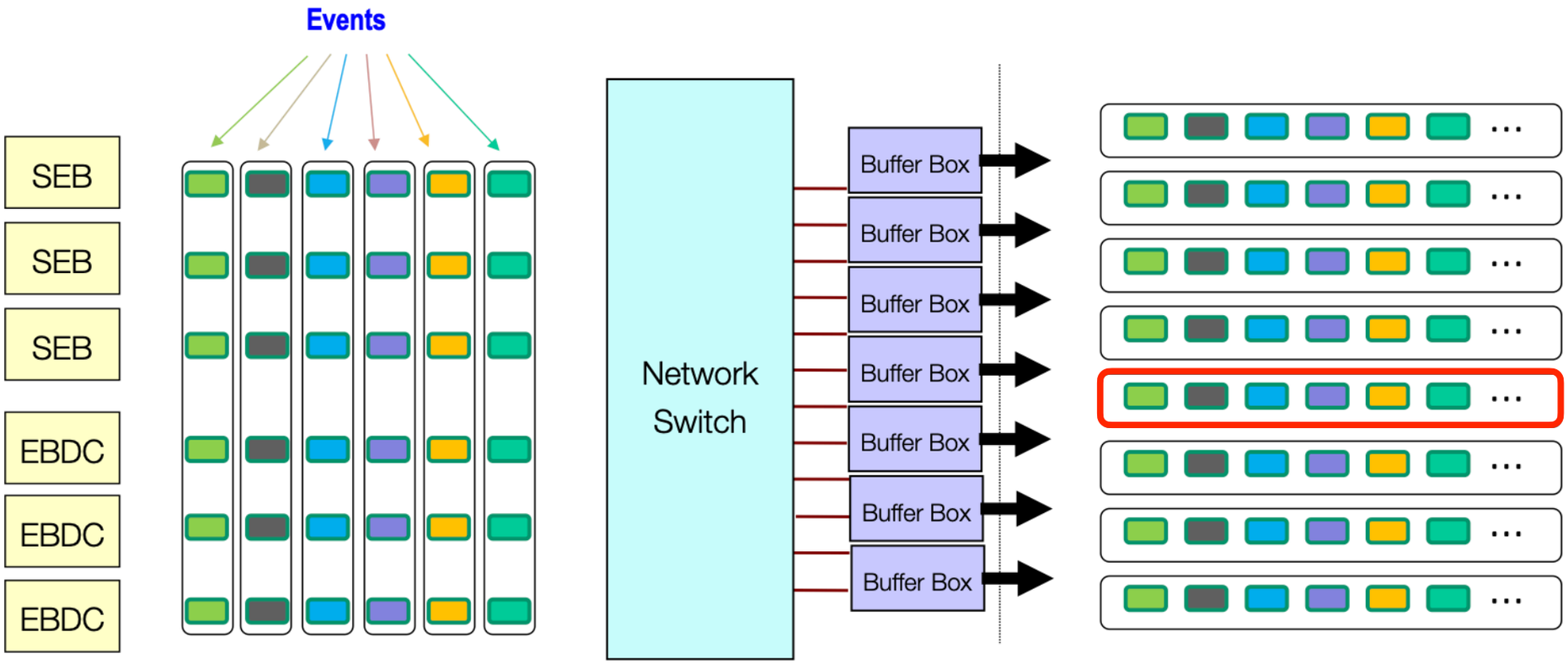
ATLAS FELIX Card

Installed in a Server

Writing Data



average

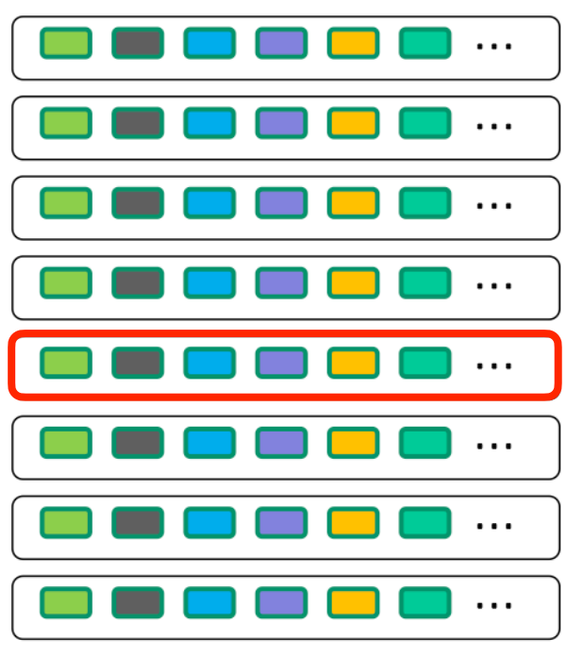


each of 61 files contains *portions* of multiple events

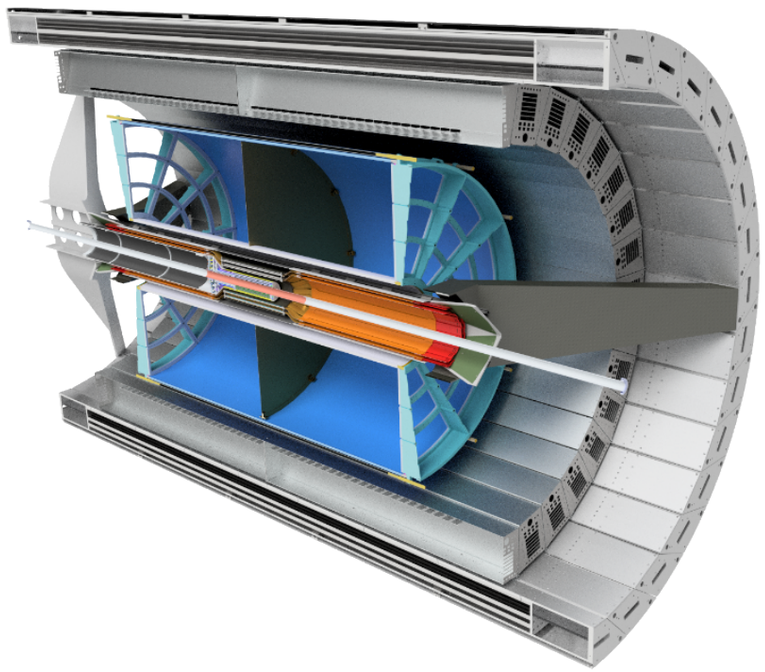
- Buffer boxes have twin file systems that can hold several stores
- Downstream links only need to handle *average* load (including breaks, accesses, accelerator studies)

Writing Data

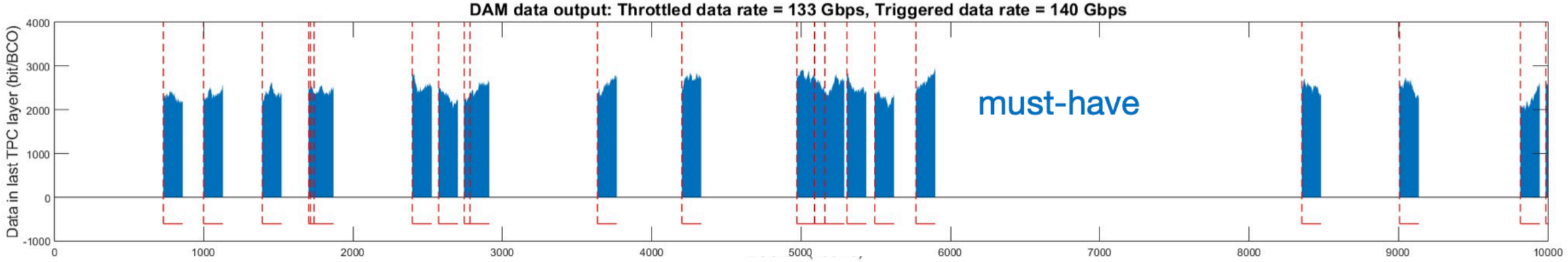
- **sPHENIX has no online event building** -- we write data from the buffers directly to 61 files by subsystem
- Less complexity in the online system, less real-time risk
- Many advantages in the offline system too:
  - **Event building occurs on the fly** during event reconstruction -- only needs to keep up with average rate
  - **Events are time-ordered within files** -- calibrations apply to contiguous blocks
  - **Reco steps only load the data they will use** -- tracking jobs only read tracking detector data, etc.
- But there's a price:
  - Reprocessing directly from tape is much harder -- many files have to be staged together in order to reconstruct events.
  - Remote processing would require lots of additional logistics

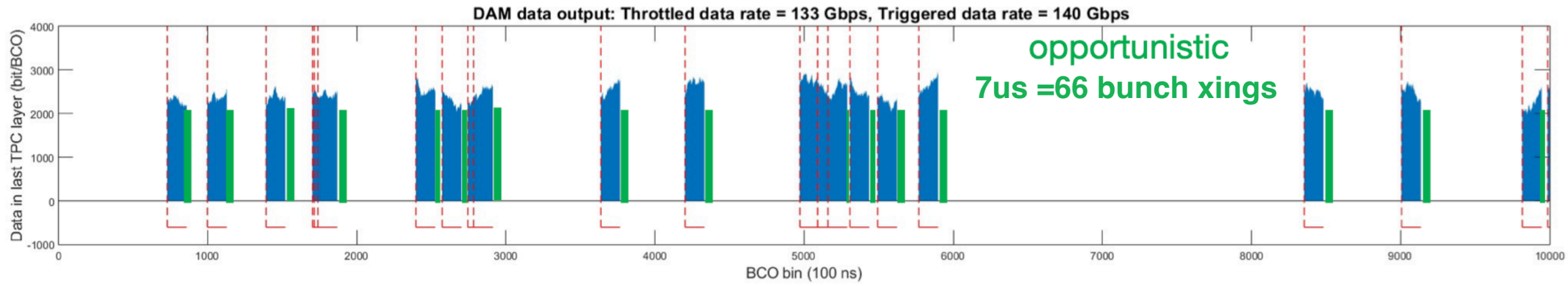


Maximizing Minbias Data in pp

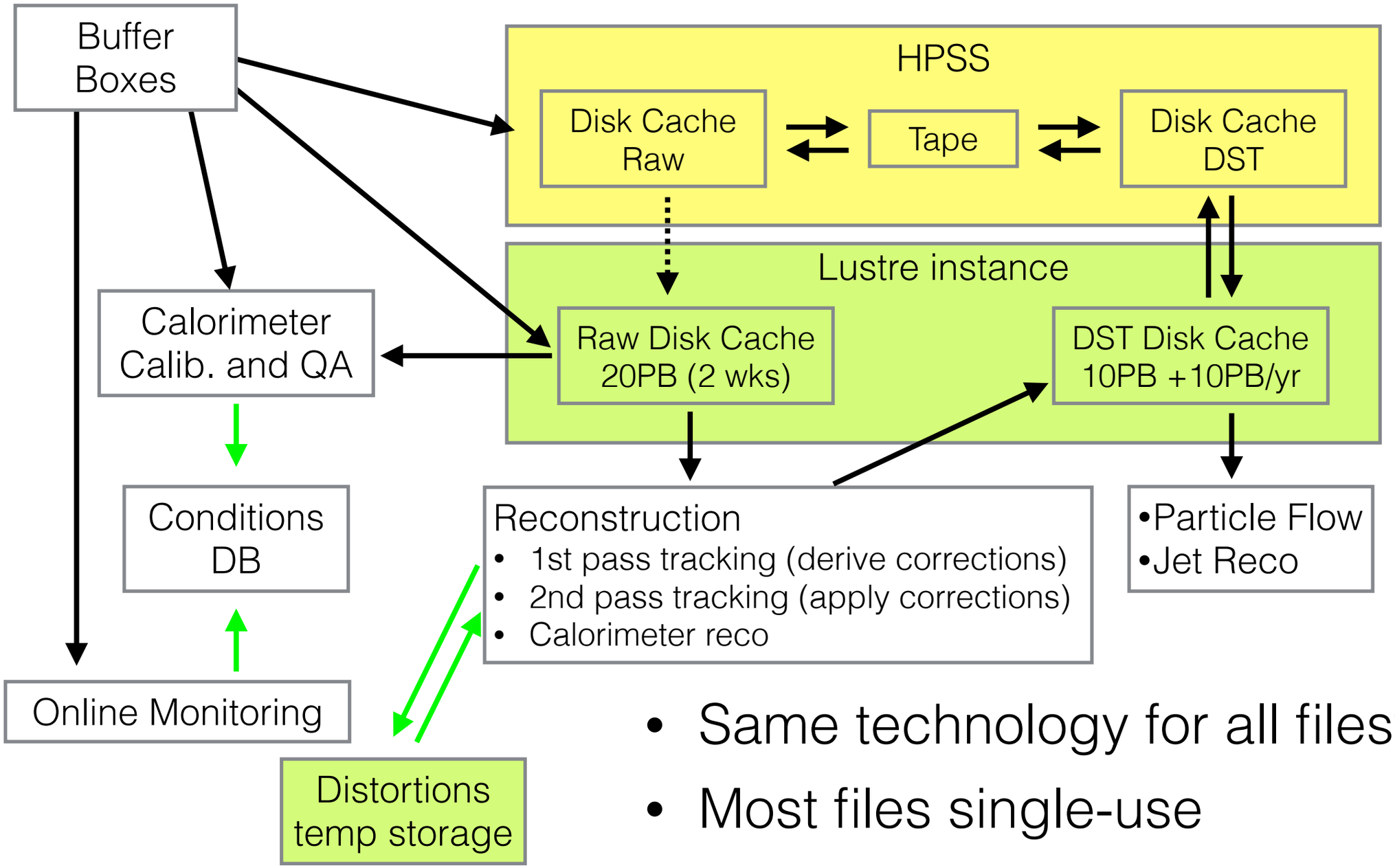


- TPC drift time means ~13us of data streaming for a full event
- ~118 collisions happen during this time, and are partially captured in the 13us window
- More minbias by recording more timebins: Extending window by 110ns completes another collision for nearly free





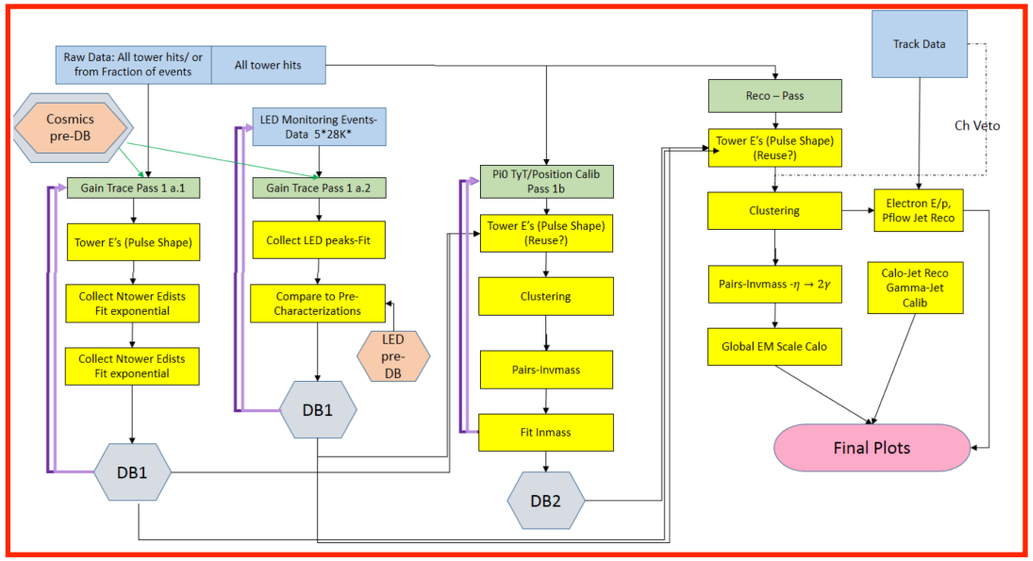
Event Reconstruction



- Same technology for all filesystems
- Most files single-use
- Cached files which have been processed can be deleted

Calorimeter Calibration Scheme

- Multiple calibration steps in full workflow:
- Initial: flatten gains across detector
- Steady-state: compare each new time period to detect and correct gain drift



LED Monitoring System

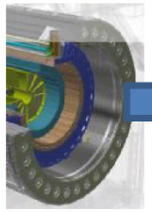
LED and Pedestal Scan (Cosmics?)

LED and Pedestal Scan (Cosmics?)

LED/Voltage/Temp Continuous

LED/Voltage/Temp Continuous

LED/Voltage/Temp Continuous



Raw Data - Fill

Run Run Run

Raw Data - Fill

Run Run Run

Raw Data - Fill

Run Run Run

...

Beam Data-Driven Methods

All: Tower E Slope

All: Tower E Slope

All: Tower E Slope

EMC: E < 1.2 π^0 Tower by Tower

EMC: E < 1.2 π^0 Tower by Tower

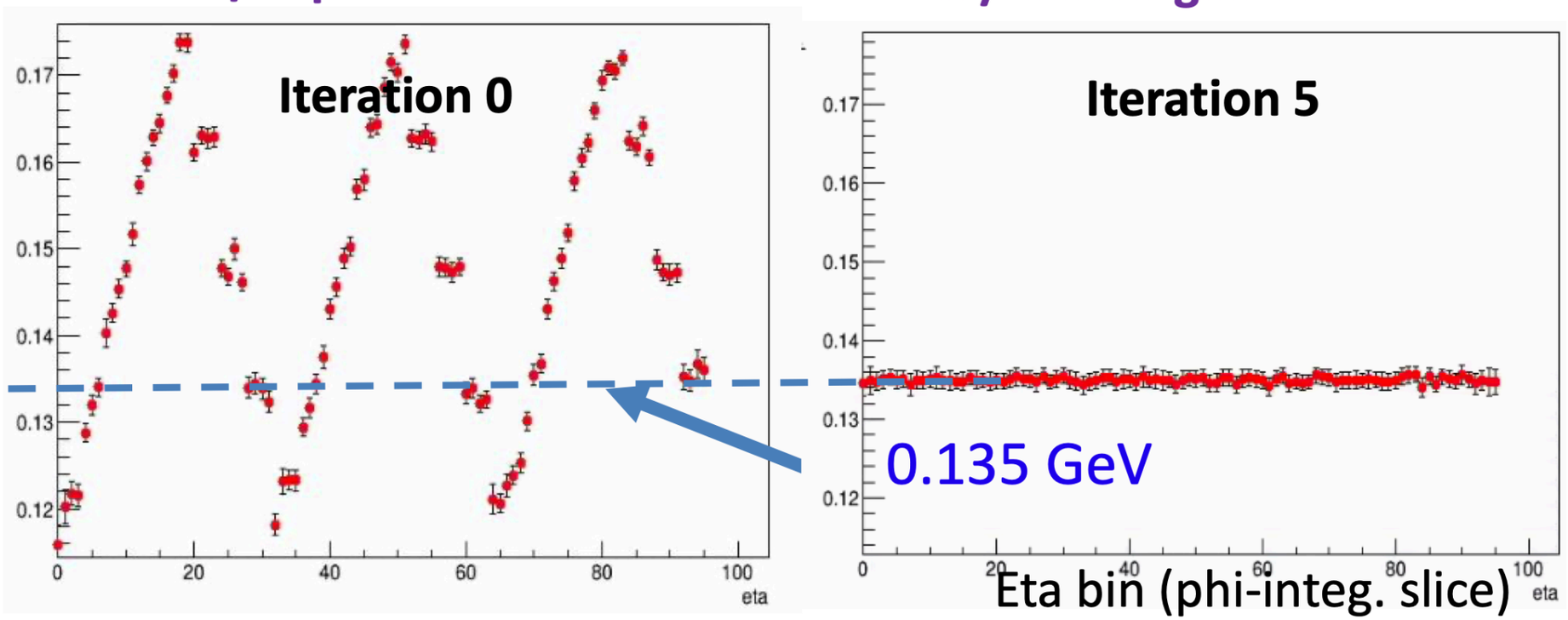
EMC: π^0 , $\eta \rightarrow 2\gamma$ & $e^{+/-}$ E/p

HCal: MC Based Jet Calibrations

Iterative Calibration Performance

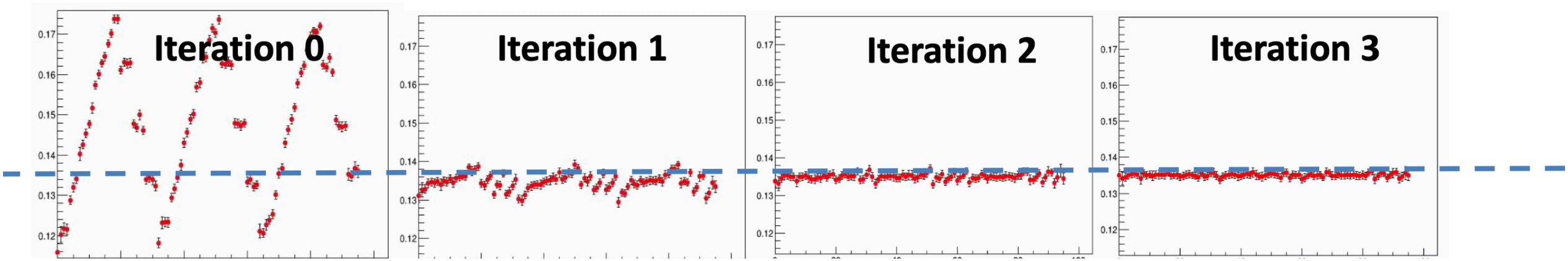
- 'Tower-by-Tower' π^0 peak finding and calibration in mock data
- Clusters include multiple cal. blocks -- sort by center tower

Iteration 0 w/ Input Decalibration Shifts **W/ Converged Corrections**



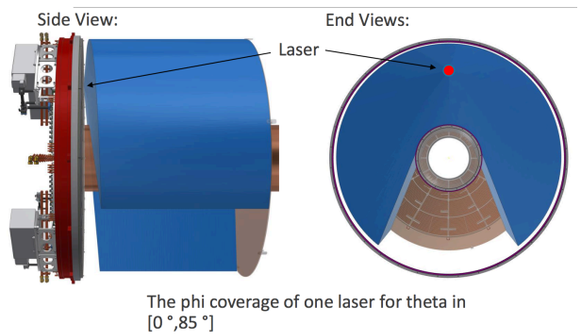
Input zig-zag shift pattern

Convergence faster for "1D" η only energy sharing

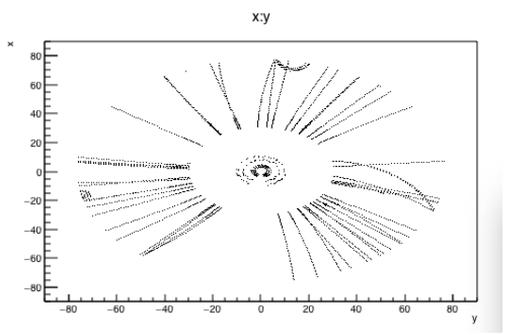


TPC Calibration

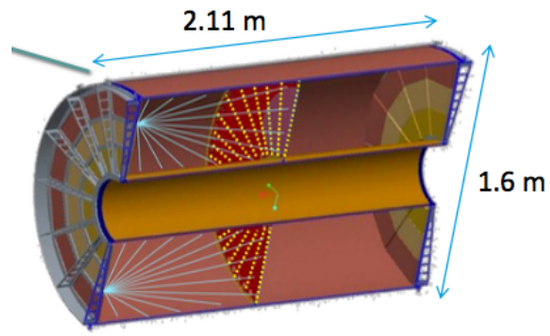
- Easier: Correct the readout signals for position-dependent gain
  - measure gain of all GEM modules before assembly
  - monitor gain from MIP spectra
- Harder: Determine where in the TPC the electrons came from:



Static Distortions mapped by **line laser**. O(**Hz**) rep rate, not used during data-taking
Monitors full 3D volume

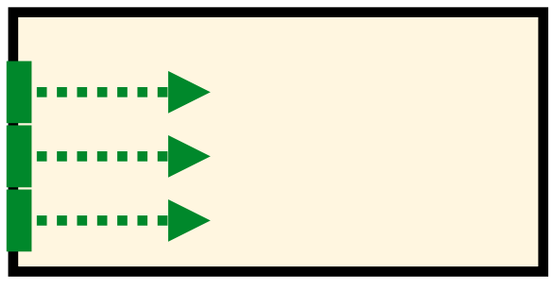


Average distortions monitored by **tracks** O(**10min**) to accumulate statistics
Monitors full 3D volume



Distortion **fluctuations** monitored by **CM pattern/diffuse laser** O(**kHz**) rep rate, interleaved with triggered events
Monitors only at z=0

Digital current infers ion backflow from electrons at readout. Provides indirect measure of SC distortion



Distortion Correction Scheme

- Very short time window where derived corrections will apply
- Derive corrections on the fly, preserve only for QA sampling

Job 0: Assemble Clusters

- Sort and tag tracker clusters and hits

Job A: Assemble Tracks

- Apply <u>static</u> correction
- Find all tracks
- Write track residuals

Job B: Calculate <u>Average</u> Correction

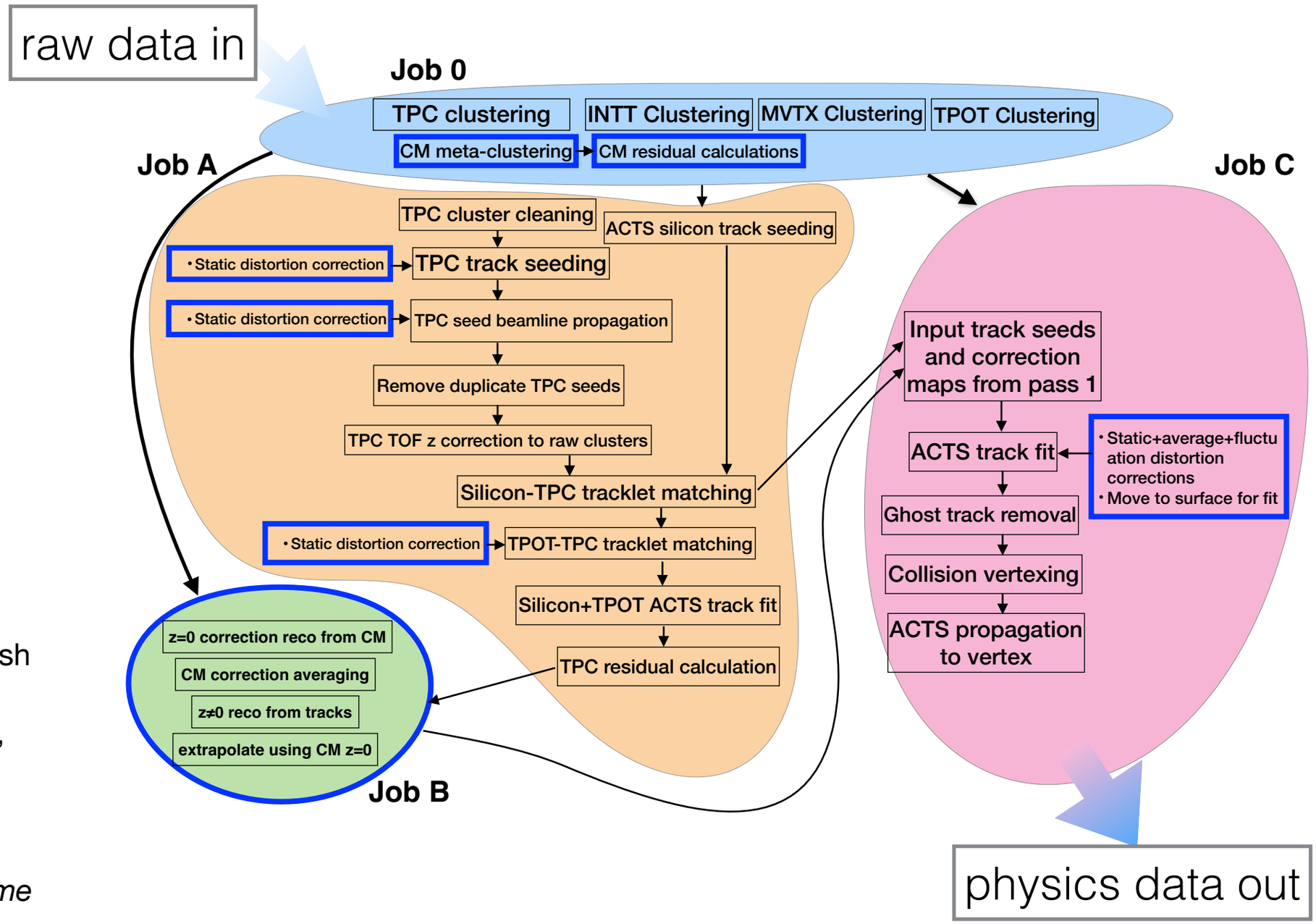
- Once per ~30min <u>aggregated</u> Job A data

Job C: Correct and Fit Tracks

- Apply <u>static</u> and <u>average</u> corrections
- Derive new <u>fluctuation</u> correction each time a CM flash is found
- Apply full correction to tracks, then run ACTS fit.

Job D: Derive distortions from Digital Currents

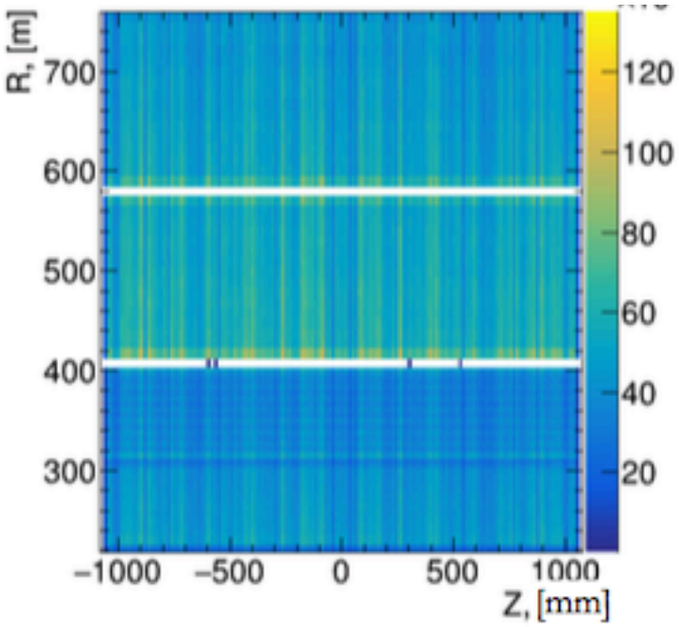
- *Once per ~10ms wall clock time*

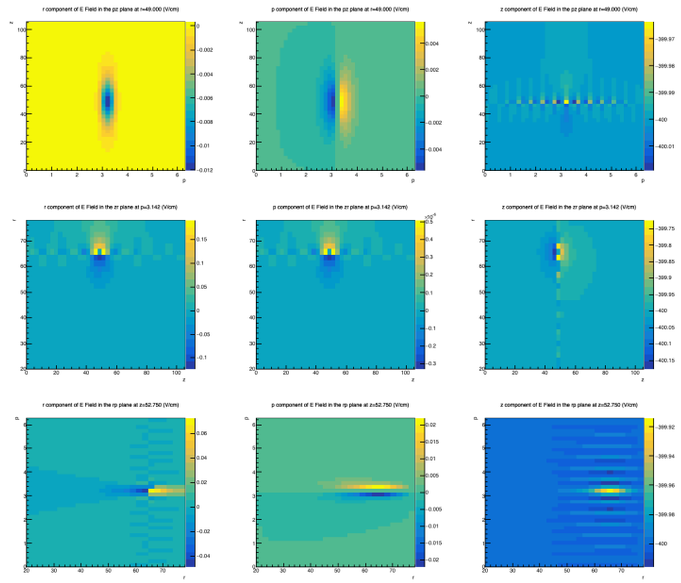


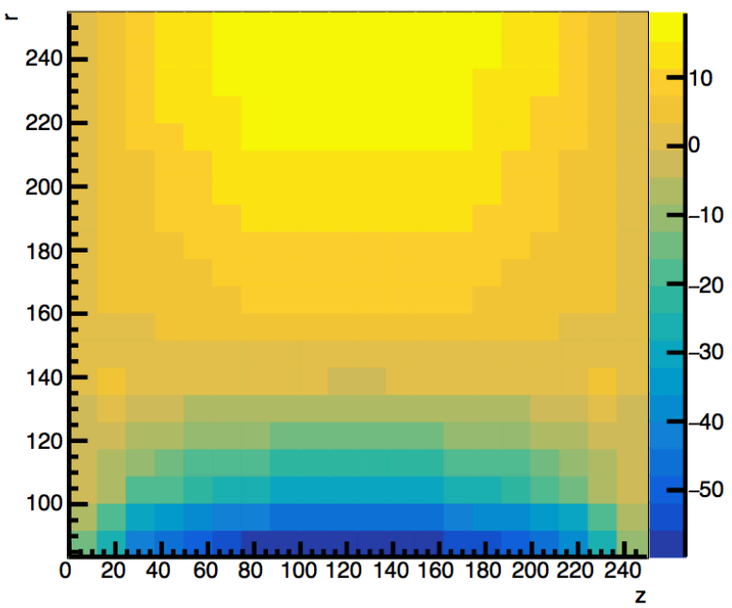
Digital Current

- Integrate e- arriving at logical blocks of readout pads over O(10ms), scale by IBF gain to get "digital current" density
- Calculate fields from charge
- Swim test particles through modeled fields to derive corrections
- Also studying ML approaches

IBF + Primary Density (3D, Scalar) x **Rossegger Greens functions (6D, Vector)** = **3D Electric Field Map (3D, Vector)**



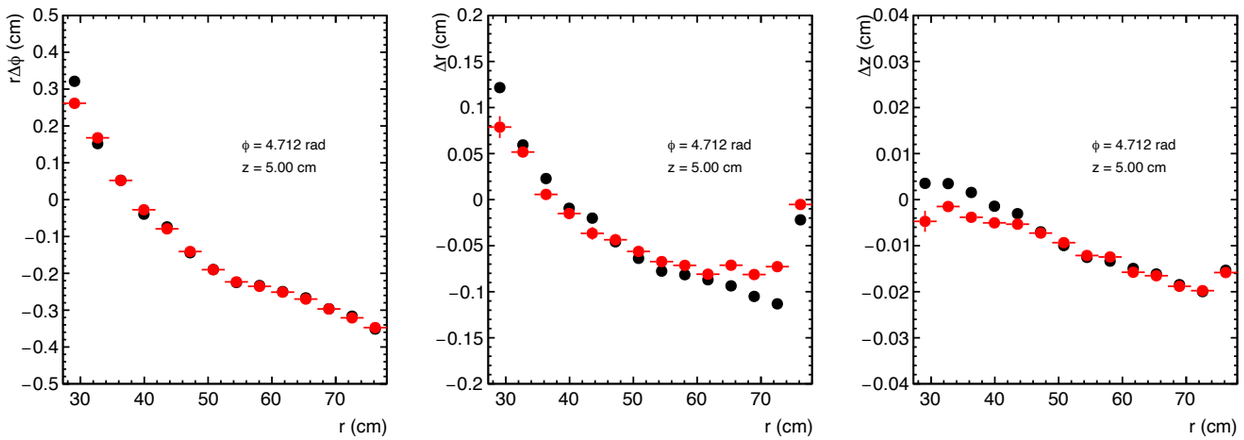




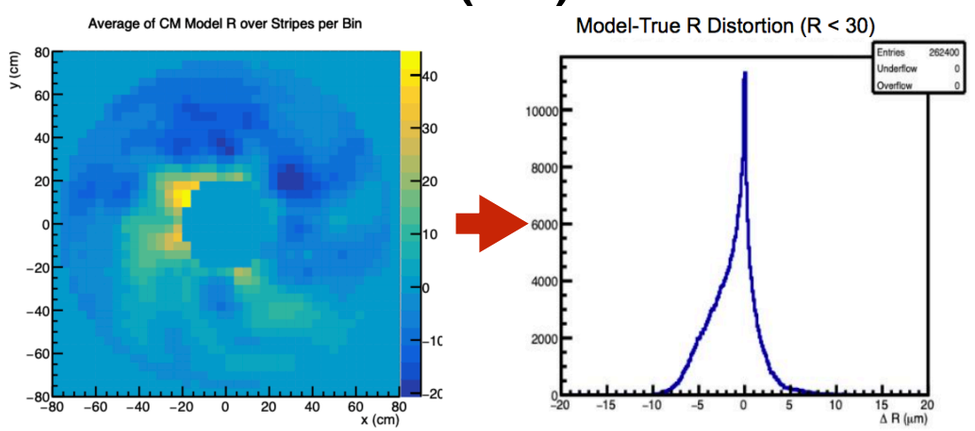
Distortion and Tracking Performance

- Distortion unit tests:

Average (track based) residuals <100um



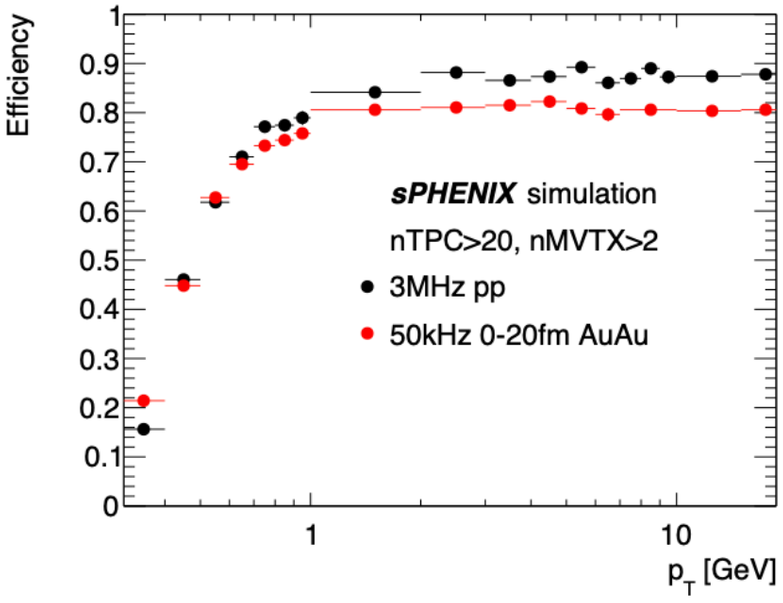
Fluctuation (CM) ~ 2x smaller

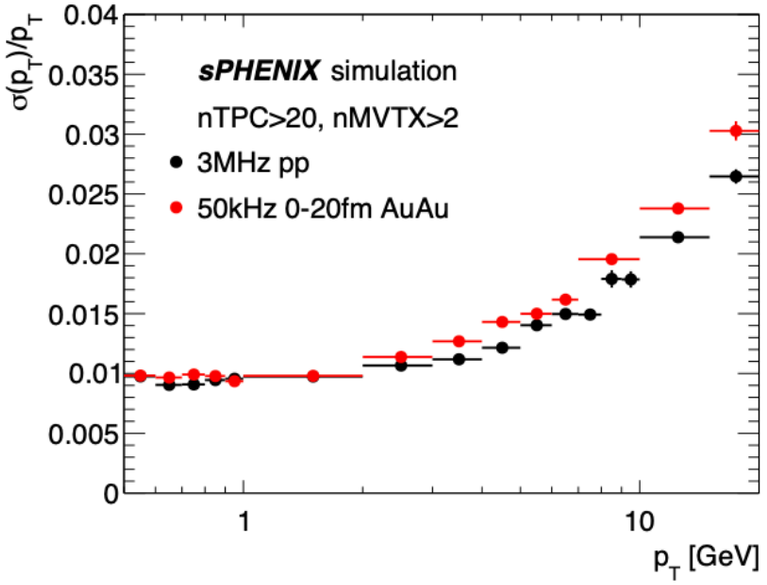


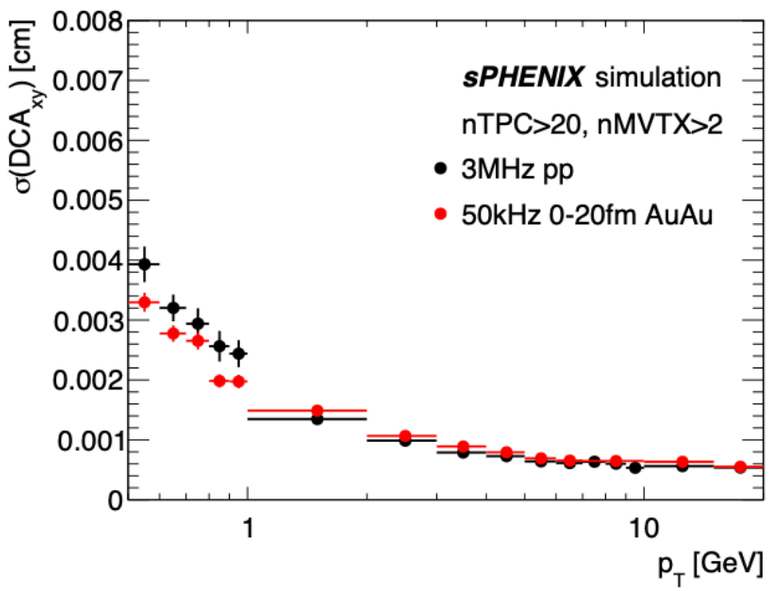
*shown from stand-alone study without full clustering chain.

*shown with truth tracking. Expect similar performance from ACTS tracks

- ACTS Tracking in mock data (w/ pile-up)





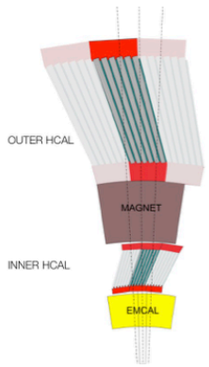


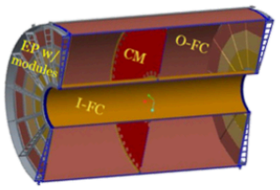
Summary

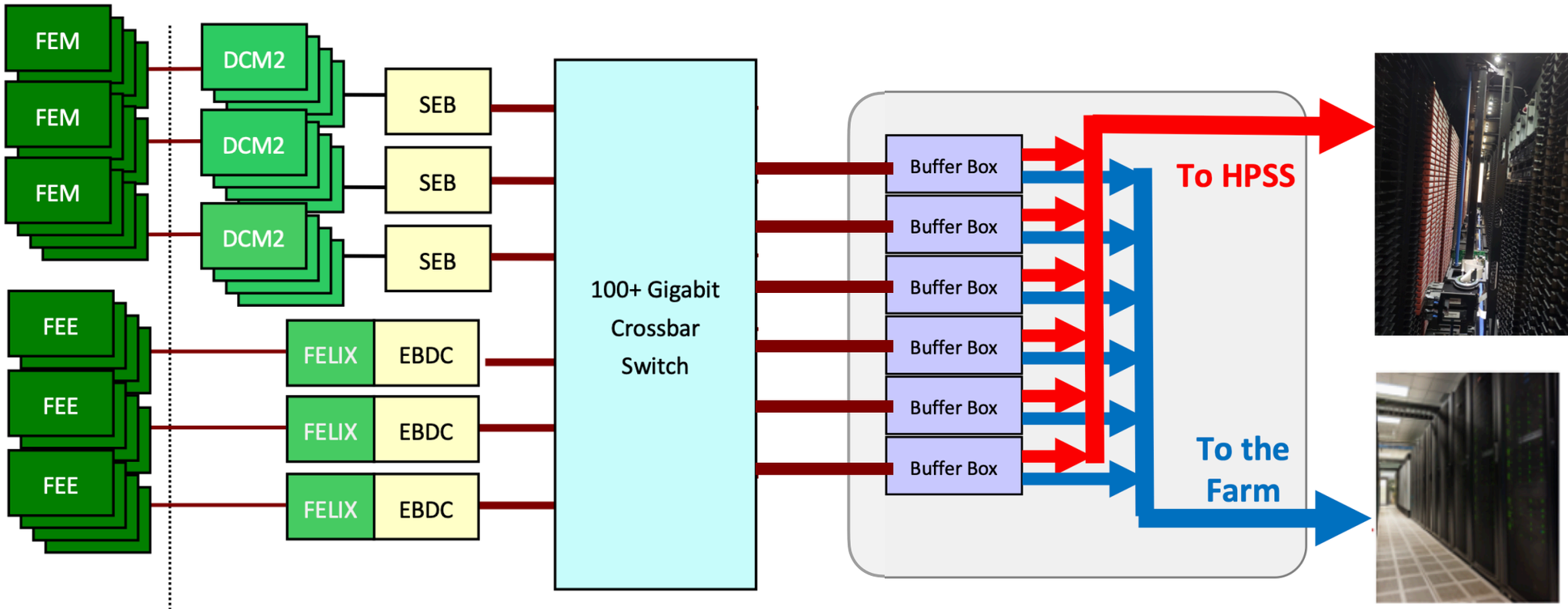
- Compressed sPHENIX run schedule: need to shorten the learning curve that usually starts when beam arrives
- RHIC is mature: high luminosity will be available from the start.
- Mixed triggered+streaming readout
- Data rate and run schedule are saturated: reconstruction must run with ~fixed latency

- Busy time for us:
  - Continuing to develop reconstruction software
  - Building and installation as well -- beam scheduled early 2023!

DAQ Diagram







FEM/FEE: Front-End Module / Electronics

DCM2 - Data collection Module (v2)

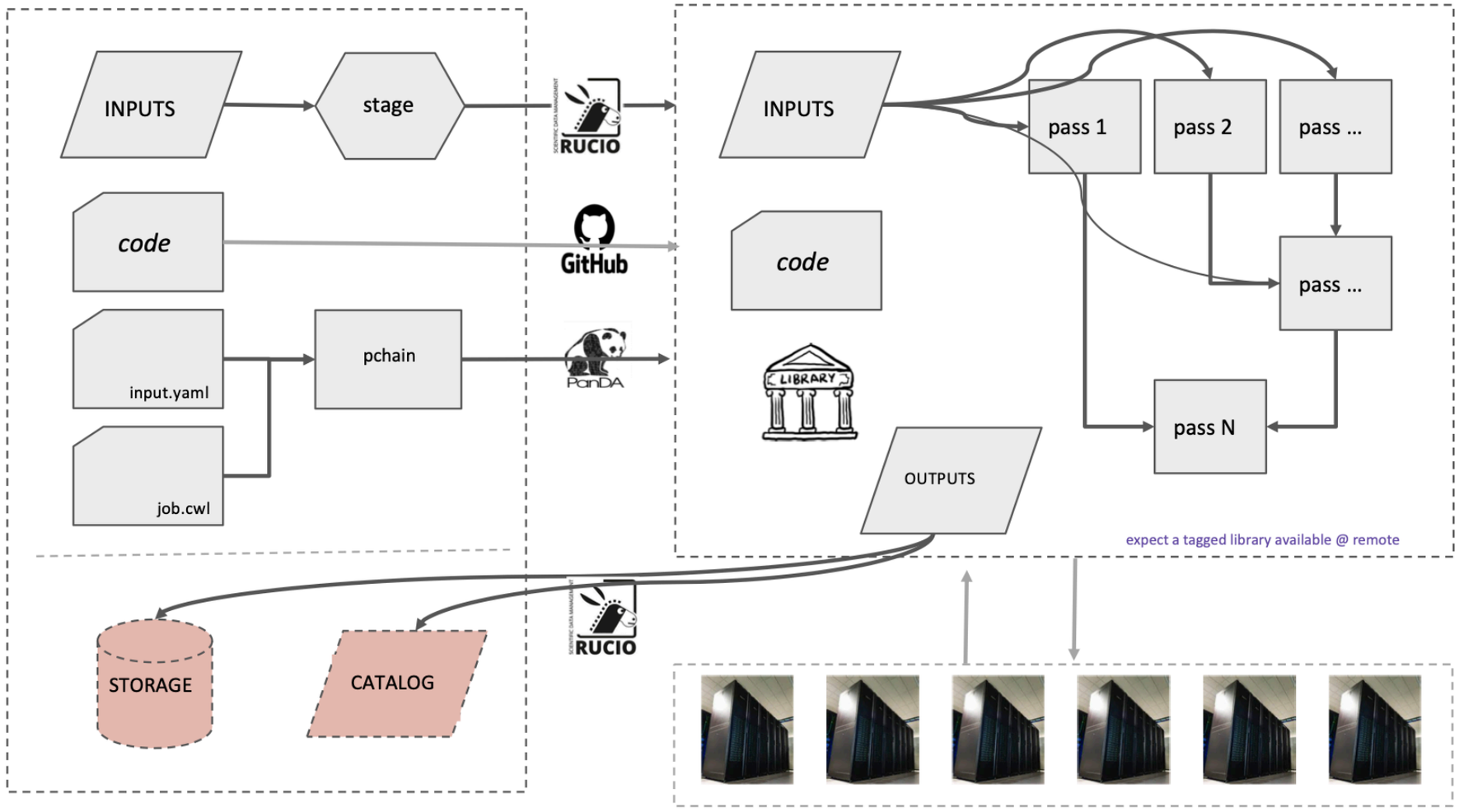
FELIX - ATLAS-developed readout card

SEB - SubEvent Buffer

EBDC - Event Buffer and Data Compressor

Courtesy Martin Purschke

Production System



PanDA

Workflow Management System

- Submits production jobs on local and remote compute resources
- Steers execution of inter-dependent job pipelines
- Returns job outputs through Rucio

SCIENTIFIC DATA MANAGEMENT RUCIO

Data Management System

- Stages input data to compute resources
- Manages intermediate job products
- Catalogs the final job outputs

GitHub

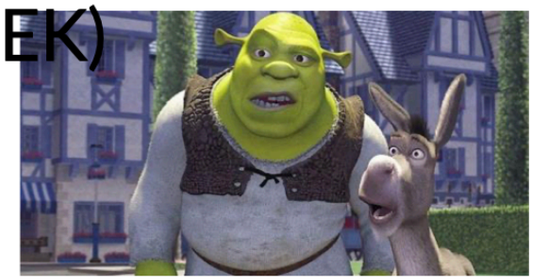
Version Control

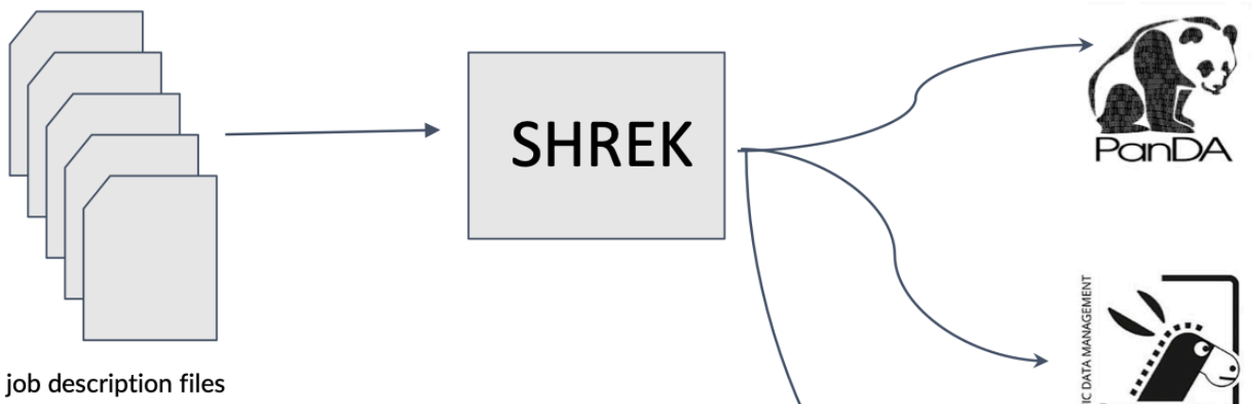
- Tags and archives production job definitions
- Revision control for the scripts and macros which execute the jobs

SHREK

sPHENIX Handy Remote Execution Koordinator (SHREK)

SHREK provides a coordination layer, simplifying the task of defining jobs, staging data sets and documenting production campaigns while providing a single source description for the complete production.





- Individual jobs declared in a job description file
- SHREK analyzes dependencies and maps jobs onto PanDA
- Required inputs are staged via Rucio if not present
- Production files are tagged and archived to github

Workflow Management System
- Submits production jobs on local and remote compute resources
- Returns job outputs through Rucio

Data Management System
- Stages input data to compute resources
- Manages intermediate job products
- Catalogs the final job outputs

Version Control
- Tags and archives production job definitions
- Revision control for the scripts and macros which execute the jobs

Calorimeter Flow Chart