Test Beam Software Support

Derek Anderson (JLab) and others

ePIC Weekly S/C Meeting

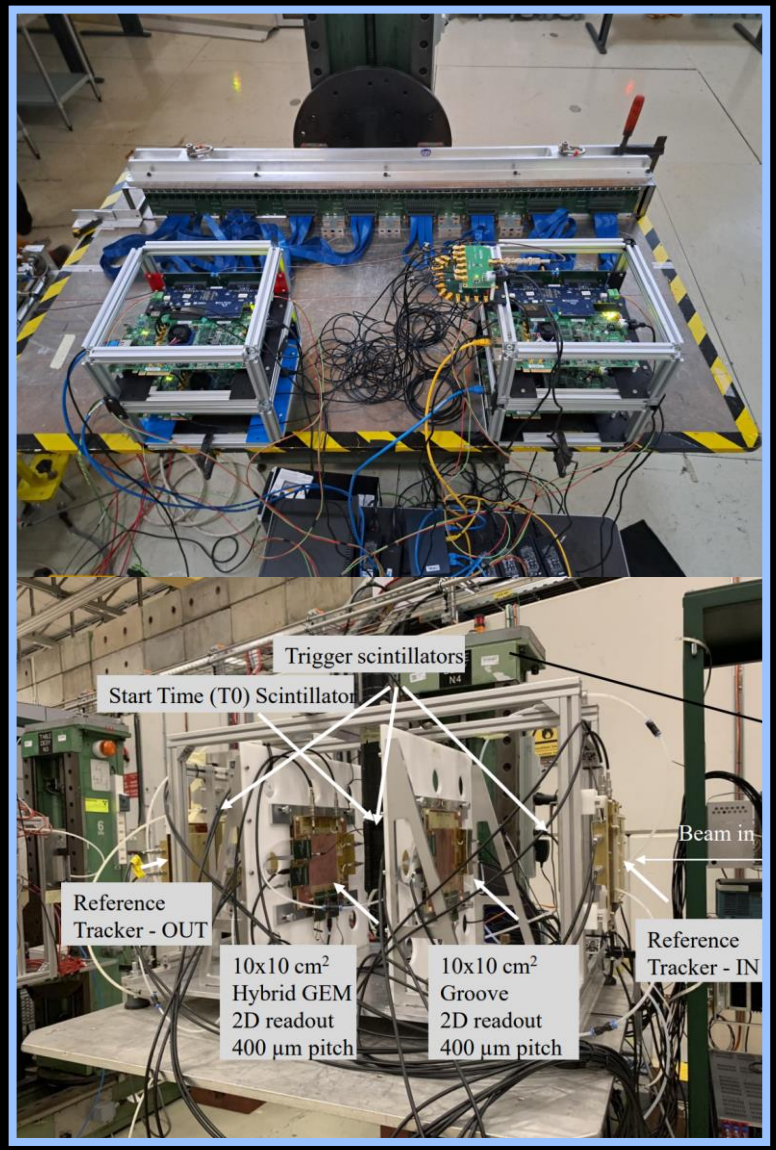
October 29th, 2025

Intro | Test-Beam Support & Data-Taking Readiness

- **Critical period for EIC:** beginning to move into construction!
  - Several test beams (TB) already happened and many being planned
    - ☞ cf. this week's TIC meeting
  - **Left:** setups from LFHCAL (upper) & MPGD (lower) 2024 TBs at CERN PS

- **From software side:** want to support test beams as much as possible
  - And want to begin preparing for real data-taking
    - ☞ TB data allow for practicing reconstructing / analyzing real data
  - The earlier we start, the better!

- Held dedicated workfest on topic at Summer 2025 EIC UGM/ePIC CM
  - ∴ **In this talk:** will summarize workfest and provide recent updates



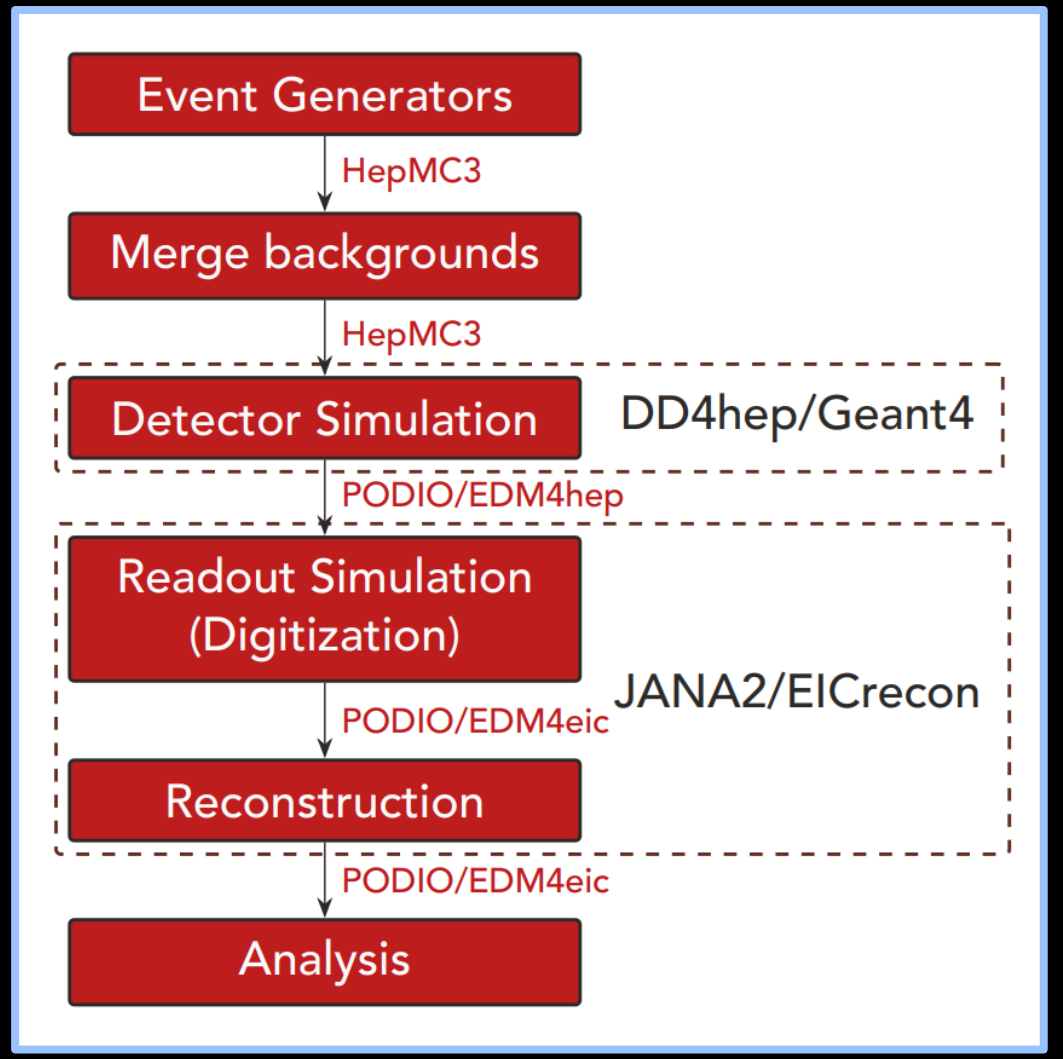
Tristan Protzman (LU), CPAD2024

Kondo Gnavo (JLab), CPAD2024

Intro | Reminder: the ePIC Software Stack

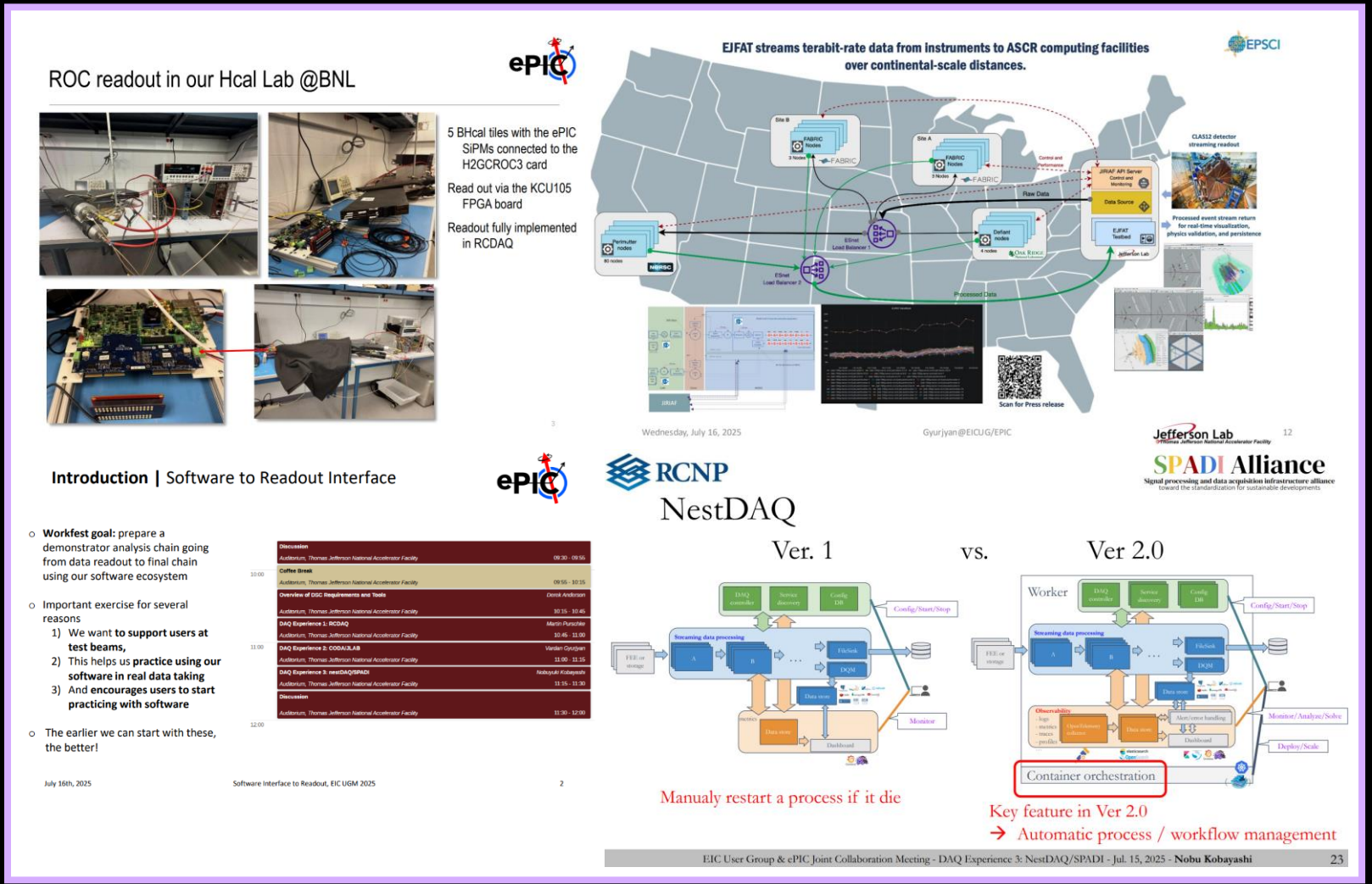
- **Right:** an overview of the ePIC software stack
  1) MCEGs simulate physics channels + backgrounds
  2) GEANT4 + DD4hep simulates detector response
  3) JANA2 + EICrecon simulates digitization and runs reconstruction
  4) PODIO structures flow of data through simulation, reconstruction to analysis
  5) Flat output format facilitates analysis in wide range of analysis frameworks

☞ **Stack is flexible enough to support simulation/reconstruction of both main detector *and* TB setups**
  - Working in stack *now* prepares us for real data taking *later*
  - And is *very* useful for reproducibility!
  - ∴ **Now working towards supporting TBs!**



Dmitry Kalinkin (BNL), 2025 RHIC/AGS AUM

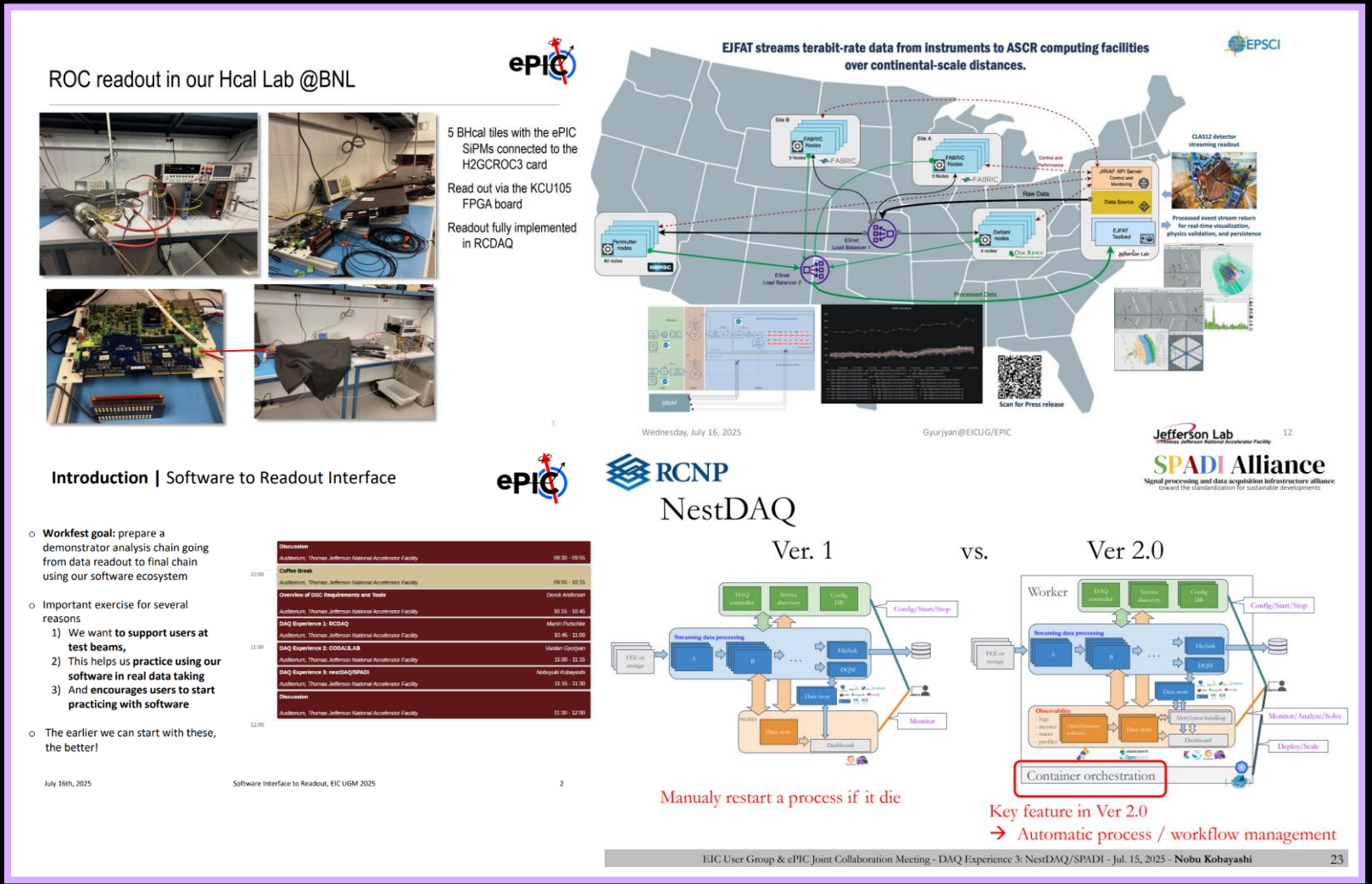
EIC UGM Workfest | Morning Talks



- 2-part workfest at recent EIC UGM aimed at interfacing software stack to TB data
  - Joint morning session w/ E&DAQ hosted talks on
    - Select DSC TB experience (BIC, LFHCAL/EEEMCal)
    - Common DAQ frameworks

- 3 DAQ Frameworks discussed:
  - **Martin Purschke (BNL)** discussed RCDAQ, highlighting its portability
  - **Vardan Gyurjyan (JLab)** discussed CODA+ERSAP, highlighting its streaming capabilities
  - **Noboyuki Kobayashi (OU-RCNP)** discussed nestDAQ, highlighting its throughput & integration with 3rd-party software

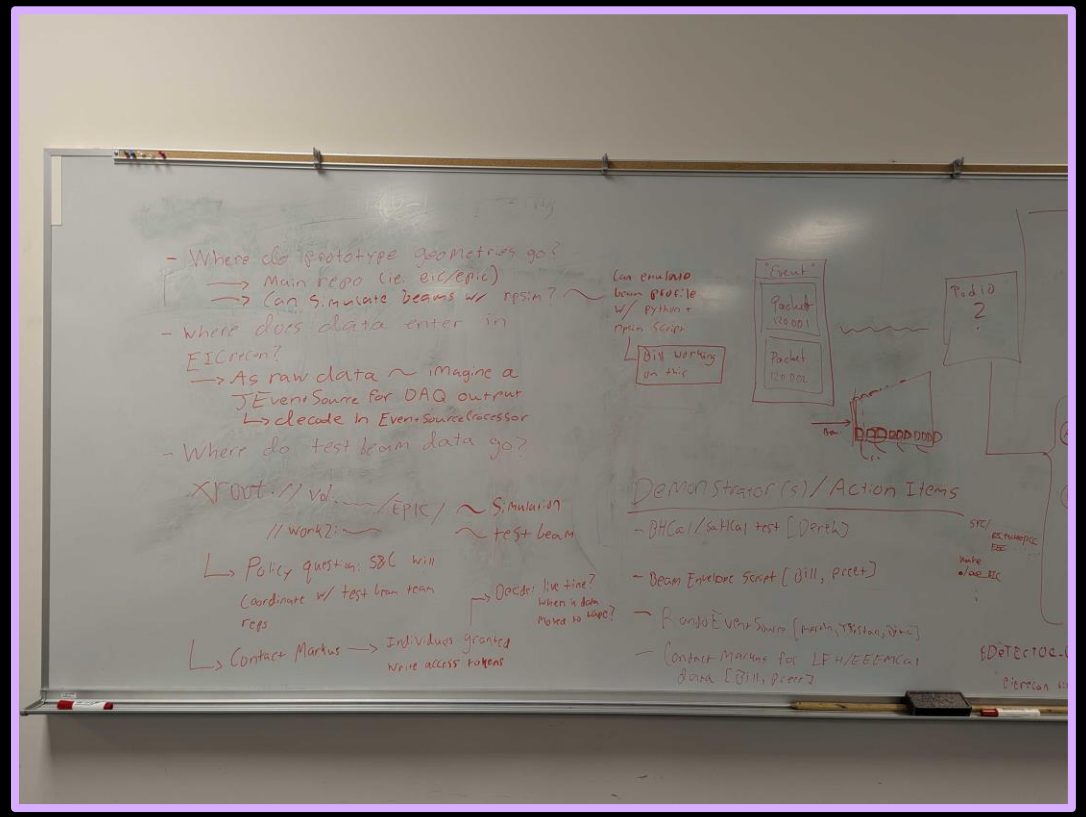
EIC UGM Workfest | Morning Talks

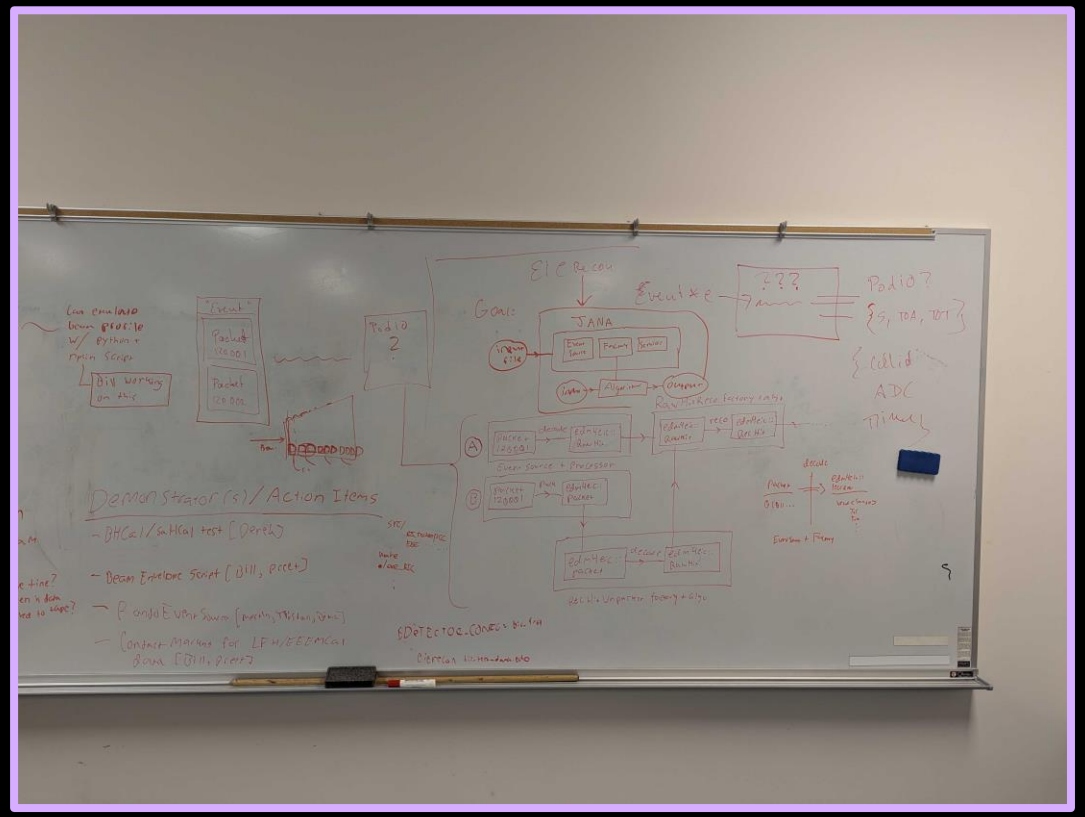


- 2-part workfest at recent EIC UGM aimed at interfacing software stack to TB data
  - Join morning session w/ E&DAQ hosted talks on
    - Select DSC TB experience (BIC, LFHCAL/EEEMCal)
    - Common DAQ frameworks

- **Updates on all 3 today!**
  - **RCDAQ:** later this talk
  - **CODA+ERSAP:** later this talk
  - **NestDAQ:** following talk by Nobuo & Taku

EIC UGM Workfest | Evening “Hackathon”





- Dedicated hackathon in evening followed morning session of talks
  - As per tradition: **no code was written…**
  - **But still had immensely productive discussion**
- **Above:** notes from discussion, which are digitized here
- Discussion helped bring folks on the same page
  - And produced **concrete actions items!**

EIC UGM Workfest | Hackathon Action Items

- **Right:** action items from workfest
  - Fall into a few categories
    a) Data availability
    b) Simulation
    c) Infrastructure
    d) Demonstrators for working with raw data

- **Note:** 2024 BIC TB good candidate for CODA example JEventSource/Processor
  - Also can build on work of 2023 streaming tests with ERSAP + JANA2
  - Will discuss more in Grand Challenge Dispatch section

Action items

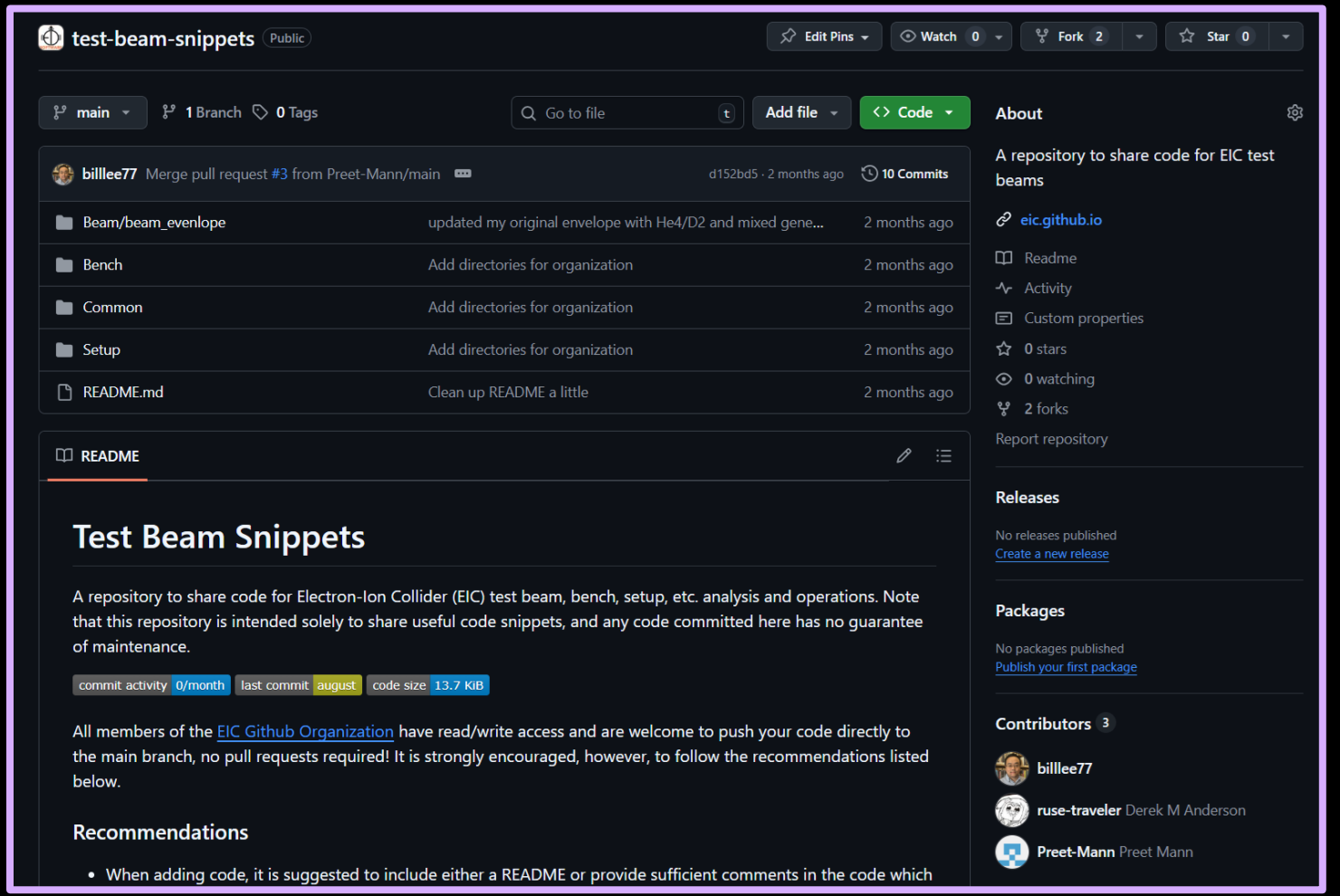
- ☑ **[Bill]** Get LFHCAL/EEEMCal TB data in XrootD/Rucio [Bill]
- ☑ **[Derek]** Create channel to share TB code
- ☑ **[Bill, Preet]** Complete beam-envelope script for npsim
- 💻 Create example JEventSource/Processors for raw data
  - **(Step 1) [Martin, Tristan]** Create test code with random hits
  - **(Step 2) [Martin, Tristan]** Then expand code to process RCDAQ data
  - **(Follow-up) [TBD]** Exercise should be repeated for CODA and NestDAQ data
- 💻 **[Derek]** Finish standalone BHCal (saHCal) custom plugin demonstrator
- ☑ **[Derek]** Finish implementing HGCROC type

☑ = Done
💻 = In progress/To-do

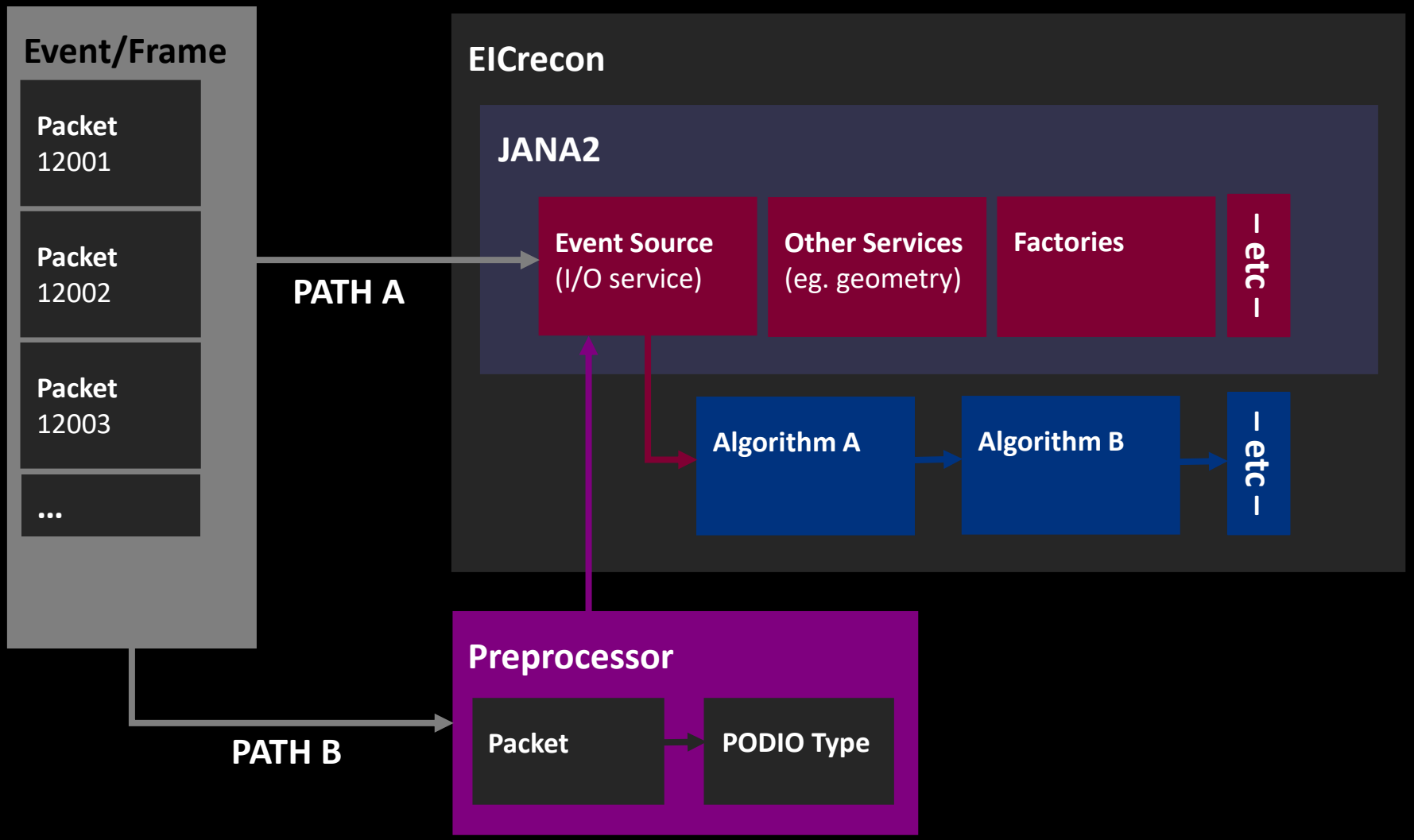
EIC UGM Workfest | Envelope Script and TB Snippets

- Bill Lee (MSU) and Preet Mann (SBU) have since completed their beam envelope script
  - Simulates beams received during tests via npsim
  - ☞ **Overview and tutorial later this meeting!**

- **To share TB-related code:** we now have a dedicated Test Beam Snippets repo!
  - Like snippets, no guarantee of code maintenance
  - But provides channel to stage and share code aimed at test beams, stands, benches, etc.



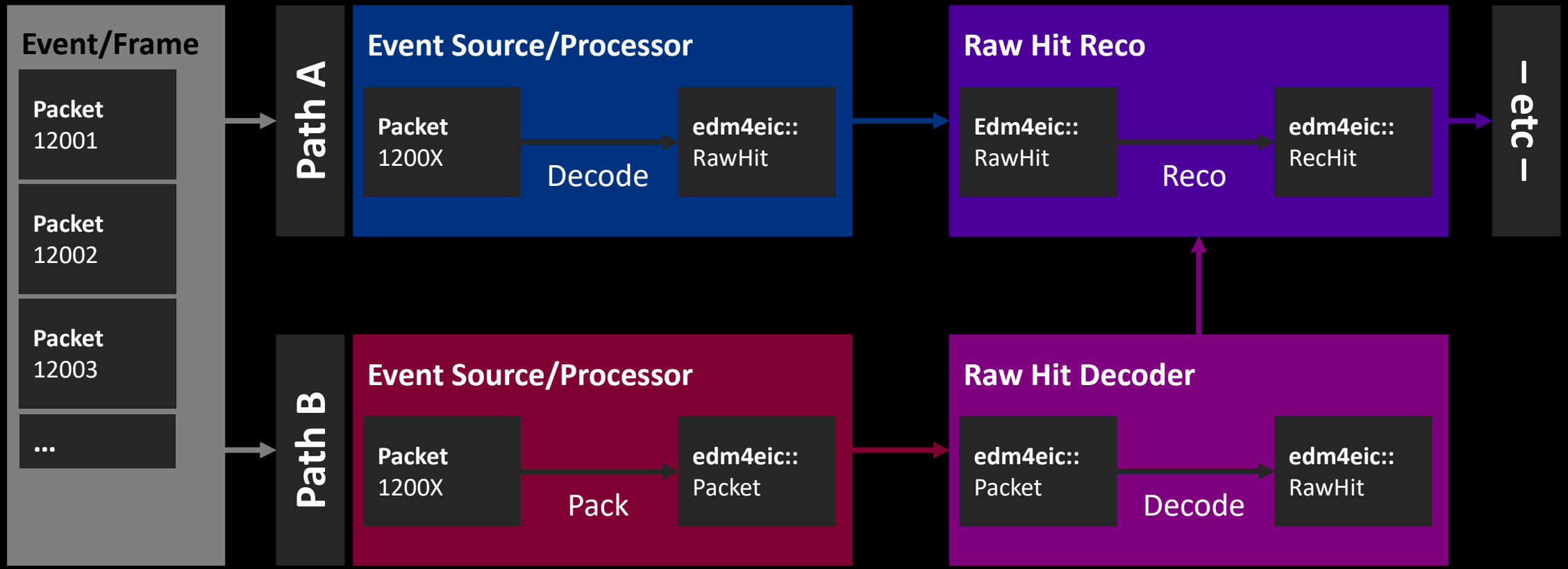
Interfacing | EICrecon Input



- **Central question at hackathon:** how do we process real data with EICrecon? Ie.
  a) How does data *arrive* at EICrecon (raw packets, PODIO types, etc.)?
  b) And if it arrives raw, how do we unpack it?

- **Left:** 2 approaches to answer question (a)
  - Consensus in room was **Path B** is fine for TBs
  - But **Path A** necessary for actual data taking

Interfacing | Potential Algorithm Flows



Interfacing | TB Example: LFHCAL

- **Thought Experiment:** what would it take to process the 2024 LFHCAL TB data in EICrecon?
  1) **Implement geometry in DD4hep**
     - In progress at epic#881
     - Crucial for simulations and **consistent geometry description**
  2) **Prepare data for EICrecon processing either**
     a) By creating a preprocessor (saHCal demonstrator), or
     b) Implementing a custom JEventSource
  3) **Create an EICrecon plugin to run desired algorithms**
     - Illustrated in saHCal demonstrator

Action items

- ☑ **[Bill]** Get LFHCAL/EEEMCal TB data in XrootD/Rucio [Bill]
- ☑ **[Derek]** Create channel to share TB code
- ☑ **[Bill, Preet]** Complete beam-envelope script for npsim
- 💻 Create example JEventSource/Processors for raw data
  - **(Step 1) [Martin, Dima, Tristan]** Create test code with random hits
  - **(Step 2) [Martin, Dima, Tristan]** Then expand code to process RCDAQ data
  - **(Follow-up) [TBD]** Exercise should be repeated for CODA and NestDAQ data
- 💻 **[Derek]** Finish standalone BHCal (saHCal) custom plugin demonstrator
- ☑ **[Derek]** Finish implementing HGCROC type

☑ = Done
💻 = In progress/To-do

HGCROC Data Model | Data Model and Changes

- **Critical for TB analyzers:** can the data model accommodate your data?
  - Eg. most calos will readout w/ variant of CMS HGCROC, **the CALOROC**
  - Test HGCROCs already used in 2024 LHFCAL/EEEMCal TBs

- But existent RawCalorimeterHit doesn't fully capture HGCROC output…
  - ∴ **Prepared 2 proposals for HGCROC implementation (right)**

```
+
+  edm4eic::RawCALOROCHit:
+    Description: "Raw hit from a CALOROC/HGCROC chip"
+    Author: "D. Anderson, S. Joosten, N. Novitzky"
+    Members:
+      - uint32_t type               // Chip type, 0 - type 1A (readout in onl
+      - uint64_t cellID             // Detector specific (geometrical) cell i
+      - int32_t  samplePhase        // Phase of samples in [# samples], for s
+      - int32_t  timeStamp          // [TDC counts]
+    VectorMembers:
+      - uint32_t amplitude          // If type == 0 - waveform amplitudes sto
lower/upper 16 bits [ADC counts]
+      - int32_t  timeOfArrival      // Calculated times of arrival, i.e. time
+      - int32_t  timeOverThreshold  // Calculated times over threshold, i.e.
counts]
+    ExtraCode:
+      declaration: "
+        bool isType1A() const {return getType() == 0;}\n
+        bool isType1B() const {return getType() == 1;}\n
+        /// If type == 1, retrieve the low gain readout (lower 16 bits) at a p
+        uint16_t getLowGainAmplitude(const size_t sample) const {\n
+          assert(sample < getAmplitude().size());\n
+          return getAmplitude(sample) & 0xFFF;\n
+        }\n
+        /// If type == 1, retrieve the high gain readout (upper 16 bits) at a
+        uint16_t getHighGainAmplitude(const size_t sample) const {\n
+          assert(sample < getAmplitude().size());\n
+          return getAmplitude(sample) >> 16;\n
+        }\n
+      "
+
```

```
+  ## An individual sample output by an HGCROC chip
+  edm4eic::HGCROCSample:
+    Members:
+      - uint16_t ADC                // [ADC Counts], amplitude of signal duri
+      - uint16_t timeOfArrival      // Time of arrival (TOA) [TDC counts], no
+      - uint16_t timeOverThreshold  // Time over threshold (TOT) [TDC counts]
+      - bool     TOTInProgress      // Flag which indicates if a TOT fired in
+      - bool     TOTComplete        // Flag which indicates if a TOT calculat
+
 datatypes:

   edm4eic::Tensor:

@@ -265,6 +274,17 @@ datatypes:

   ## ============================================================================
   ## Calorimetry
   ## ============================================================================
+
+  edm4eic::RawHGCROCHit:
+    Description: "Raw hit from an HGCROC chip"
+    Author: "D. Anderson, S. Joosten, T. Protzman, N. Novitzky, D. Kalinkin"
+    Members:
+      - uint64_t cellID             // Detector specific (geometrical) cell
+      - int32_t  samplePhase        // Phase of samples in [# samples], for
+      - int32_t  timeStamp          // [TDC counts]
+    VectorMembers:
+      - edm4eic::HGCROCSample sample  // ADC, Time of Arrival (TOA), and Time
+
```

HGCROC Data Model | Implementation and Digitization

- **EDM4eic#116 merged ahead of 2025.10.0 campaign**
  - edm4eic::RawHGCROCHit now available for use!
- This type valuable not only for TB analysis
  - ☞ **But also critical for development of more realistic digitization models**
- **Minho Kim (ANL)** now working on algorithm utilizing HGCROC type for digitization in the BIC
  - Algorithm will be penultimate stage in BIC digitization chain
  - ☞ **And will be applicable to all other calos using HGCROCs!**

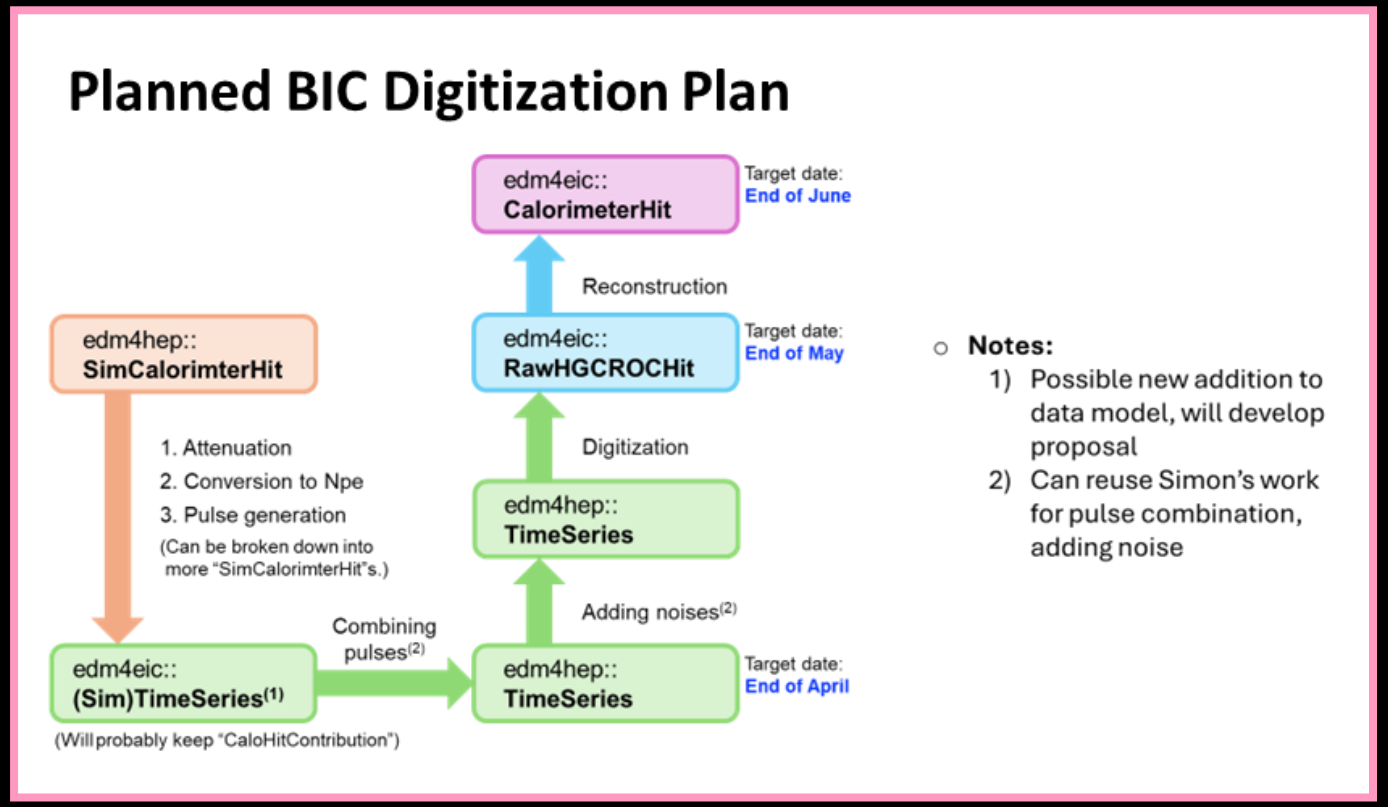


Figure: planned algorithm chain for simulation of BIC digitization. Work being driven by Minho Kim (ANL) and Simon Gardner (Glas)

JLab Streaming Grand Challenge Dispatch

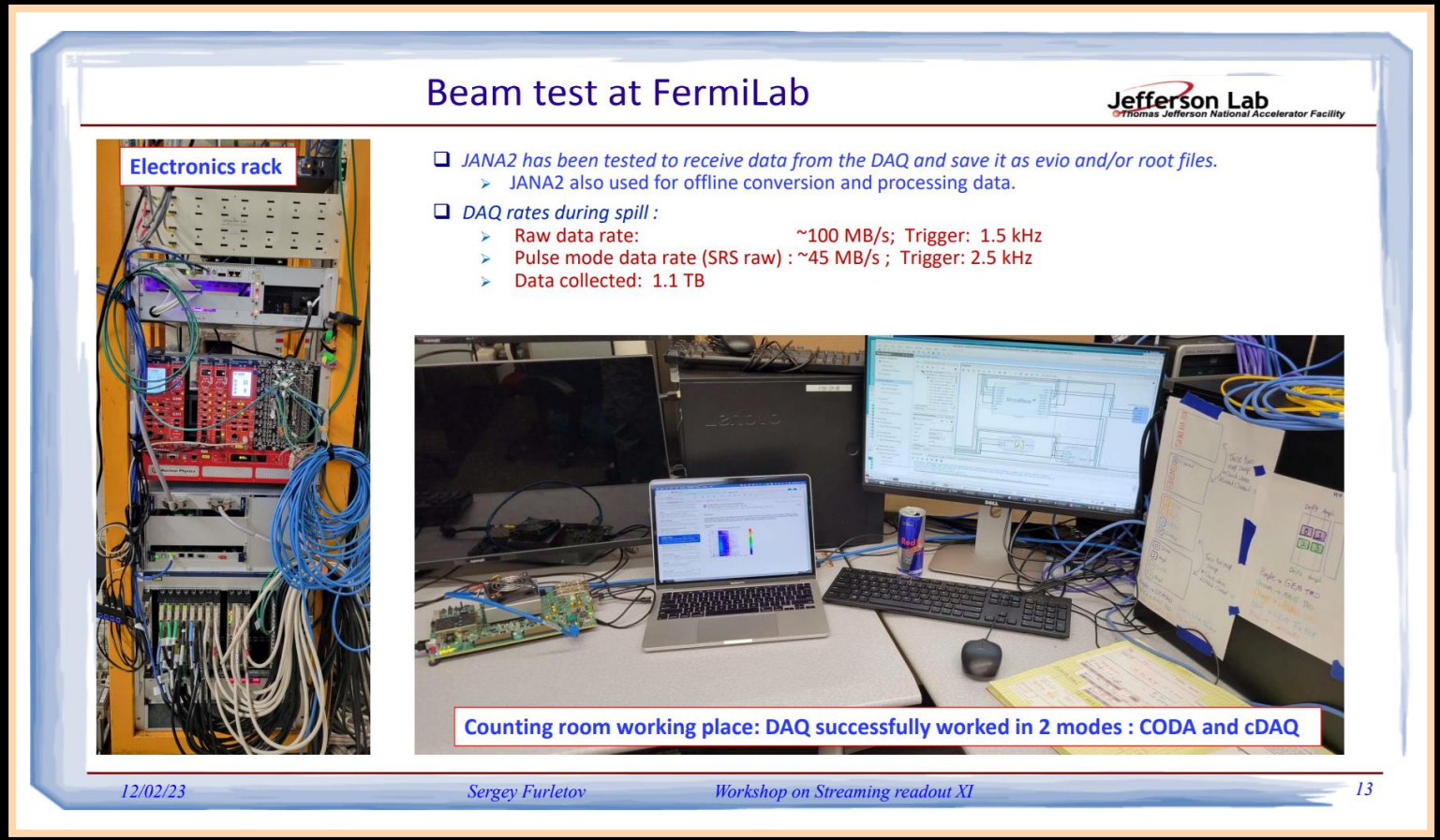
Hanjie Liu (JLab), Raiqa Rasool (JLab)
David Lawrence (JLab), Dmitry Romanov (JLab)

Grand Challenge | Dispatch (1/3)

- **SRO Grand Challenge:** JLab initiative started in 2018 to develop a whole-detector approach to SRO pipeline
- Dmitry Romanov (JLab) prepared test using CODA+JANA2
  - For 2023 ML-FPGA TB at FNAL using uRWell TB stack

☞ **Should explore extending work to EICrecon!**

Slide by Sergey Furletov (JLab) and Dmitry Romanov (JLab)

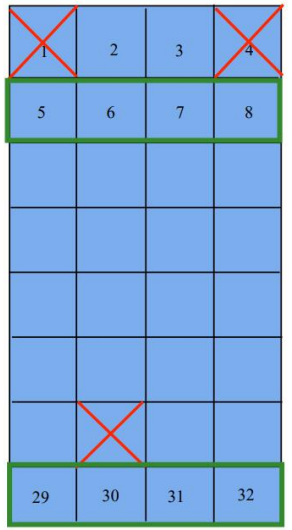


Grand Challenge | Dispatch (2/3)

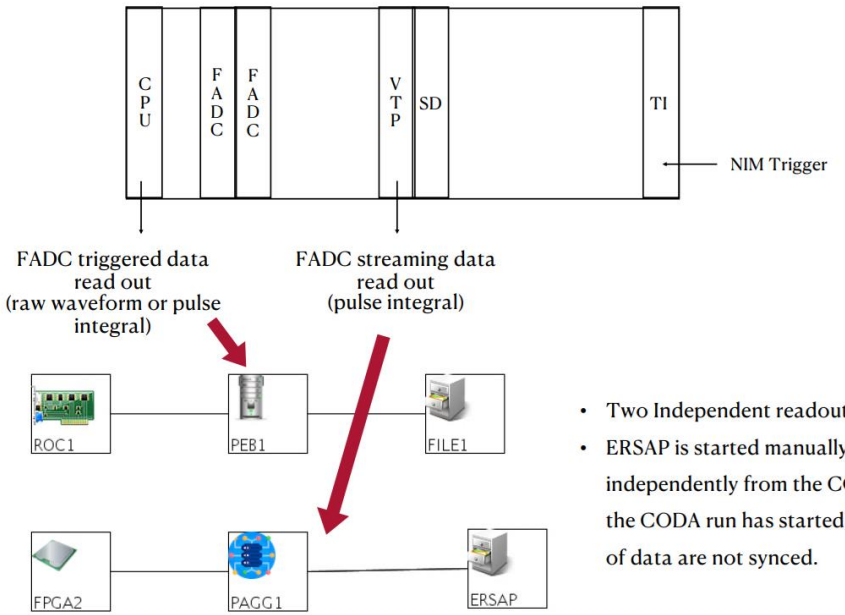
- Hanjie Liu (JLab) has set up and is analyzing a SRO test stand at JLab
  - 4x8 PMTs (3 bad)
  - Data read out by CODA, streaming w/ ERSAP
- **Data read out along 2 paths simultaneously**
  a) 1 triggered,
  b) 1 streamed w/ online software filter

Slide from Hanjie Liu (JLab)

Work summary – Oct 13 2025



NIM trigger: (5 OR 6 OR 7 OR 8) AND (29 OR 30 OR 31 OR 32)



- Two Independent readout path per run
- ERSAP is started manually and independently from the CODA (after the CODA run has started) → two sets of data are not synced.

Grand Challenge | Dispatch (3/3)

- Raiqa Rasool (JLab) profiling use of TTrees vs. RNTuples in JAPAN-MOLLER software
  - ☞ **RNTuples show improved CPU utilization, processing speeds**
- While not *directly* relevant to TBs, **this work *is* critical for future data-taking**

Slide from Raiqa Rasool (JLab)

TTree vs RNTuple

Metric name	TTree	RNTuple
Total time	85 sec	30 sec
Time/event	4.25 ms	1.5 ms
Rate	235 Hz	667 Hz
CPU Utilization	90.9%	99.2 %
Total samples	85k	30K

Note: Data collected using 'perf record -F 999', i.e., 999 samples/second

Update on RCDAQ Integrations

Weibin Zhang (UCR), Xilin Liang (UCR),
Martin Purschke (BNL), Norbert Novitzky (ORNL),
Eric Mannel (BNL)

RCDAQ | Integrations (1/3)

Slides from Martin (BNL)

ePIC ZDC test beam at the BNL NSRL

Weibin Zhang, Xilin Liang, Martin, Norbert, Eric

The ePIC ZDC is a close sibling to the forward HCal insert (similar hardware)

23 Layers of absorber with 5x5 scintillator/SiPM readout

The NSRL is a paid-for by NASA with <= 2.5GeV p beams, and ions

Whenever there is beam, we can ”play auxiliary beam dump” – completely opportunistic (and free). We actually received a warm welcome there!

We use the current H2GROC3, the ZCU105, and RCDAQ to read out as many channels as we can (5 layers = 125 of 144 max)

We have the setup in place but not seen actual beam yet – the target area is open this week for us to install

We can show you some cosmics…

1

RCDAQ | Integrations (2/3)

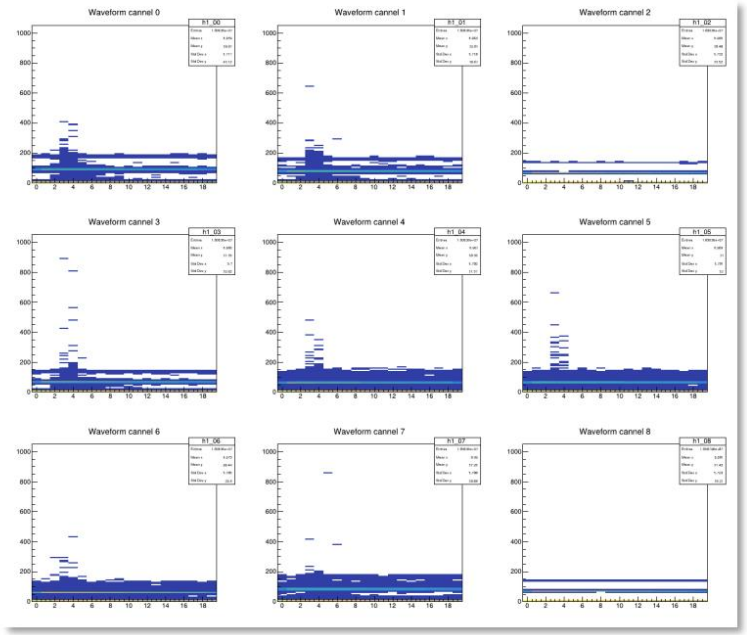
Slides from Martin (BNL)

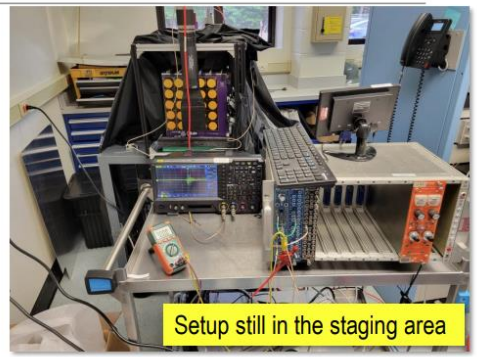
The setup

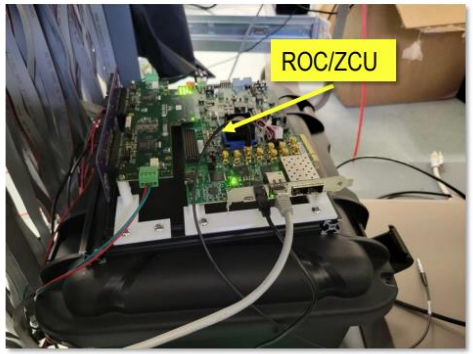
Pretty standard: 2 scintillator paddles to make a trigger, detector behind it

Same DAQ setup, online monitoring etc etc as from the BHCal

As promised, some cosmics (more once we see beam)







2

RCDAQ | Integrations (3/3)

Slides from Martin (BNL)

Related work, and other test setups

1) Software integration

- Dmitry and Martin are working on the full test beam data -> EICrecon integration, interrupted by actual tests…

2) We have 2 additional active test [not necessarily beam] setups that we support:

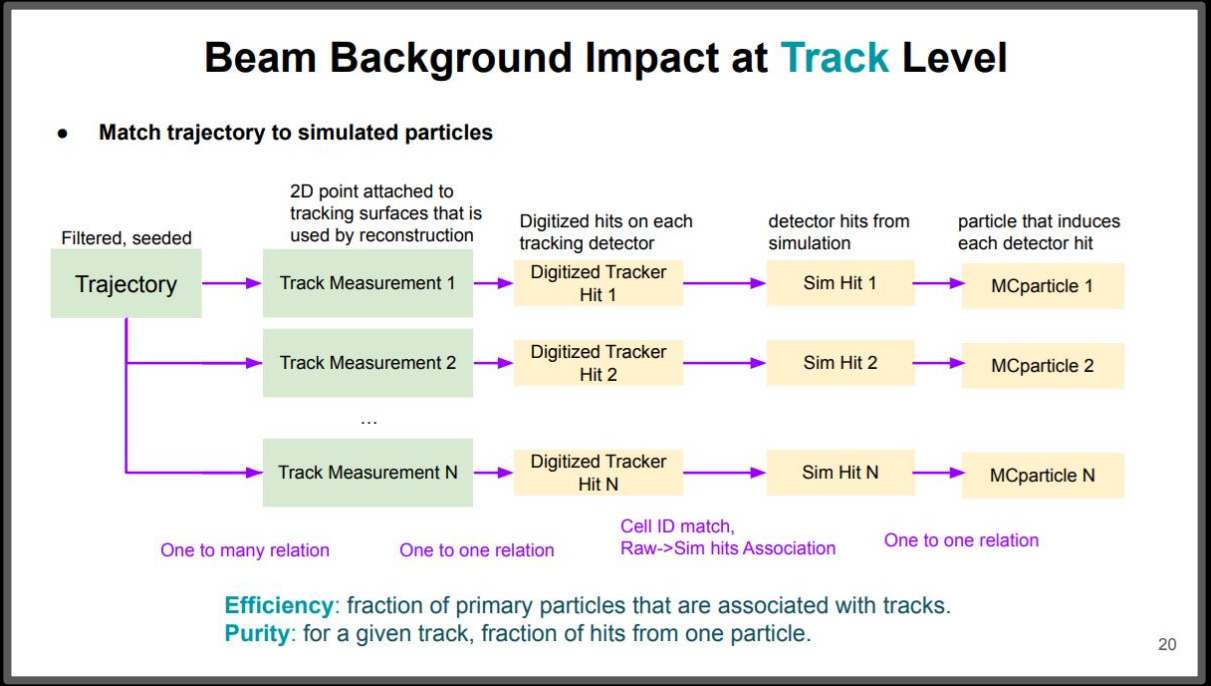
- pfRICH (Alexander's) HRPPD magnetic field test (not a test *beam* per se, but close in spirit)
- and the continuing tests we do with the BHCal prototype.

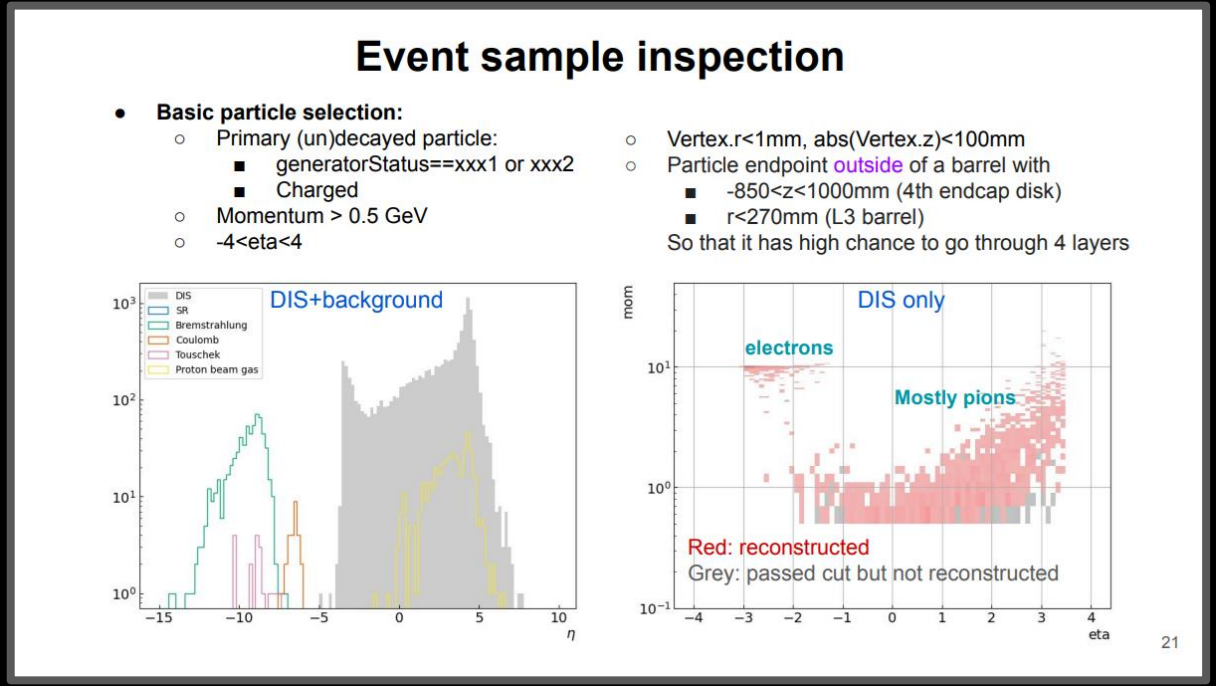
More about the EICrecon status next time.

3

Thanks!

Backup | Studying Background Impact

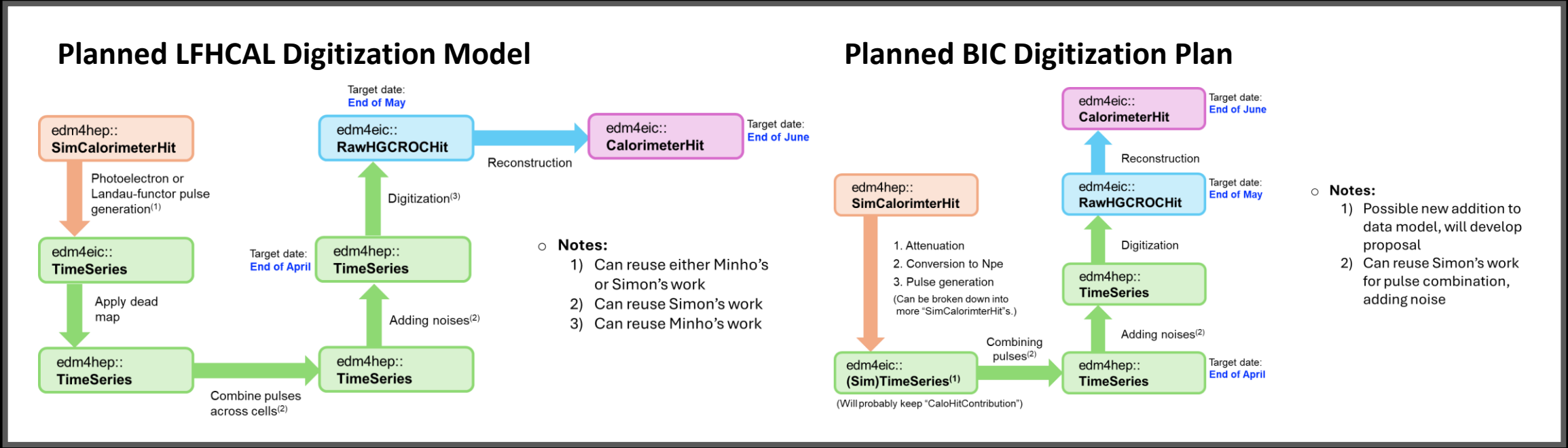




- **Recent push:** moving towards reconstruction + analysis in more realistic environments, **incl. backgrounds**
  - Studies of impact on tracking being led by **Shujie Li (LBNL)**, **Mito Funatsu (UCB)**, and more

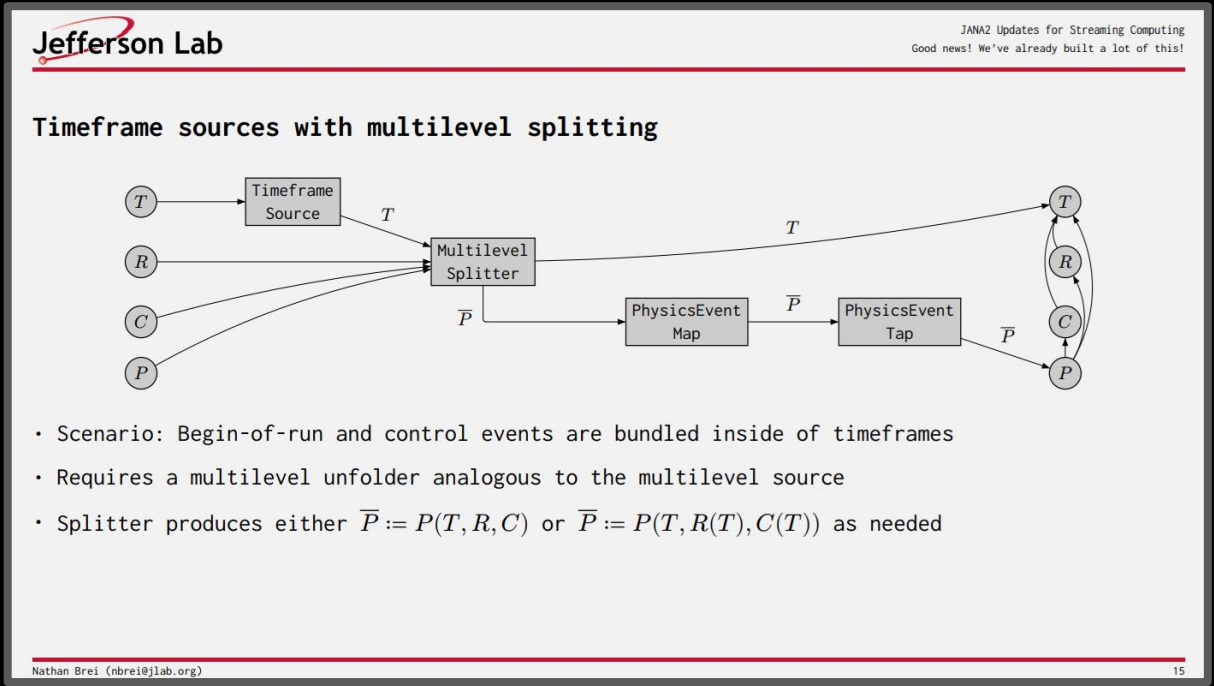
- **Above:** slides from Shujie's EIC UGM talk on tracking with backgrounds
  - **Left:** a slide illustrating relation of types in track reconstruction
  - **Right:** eta, momentum distributions of signal and background particles

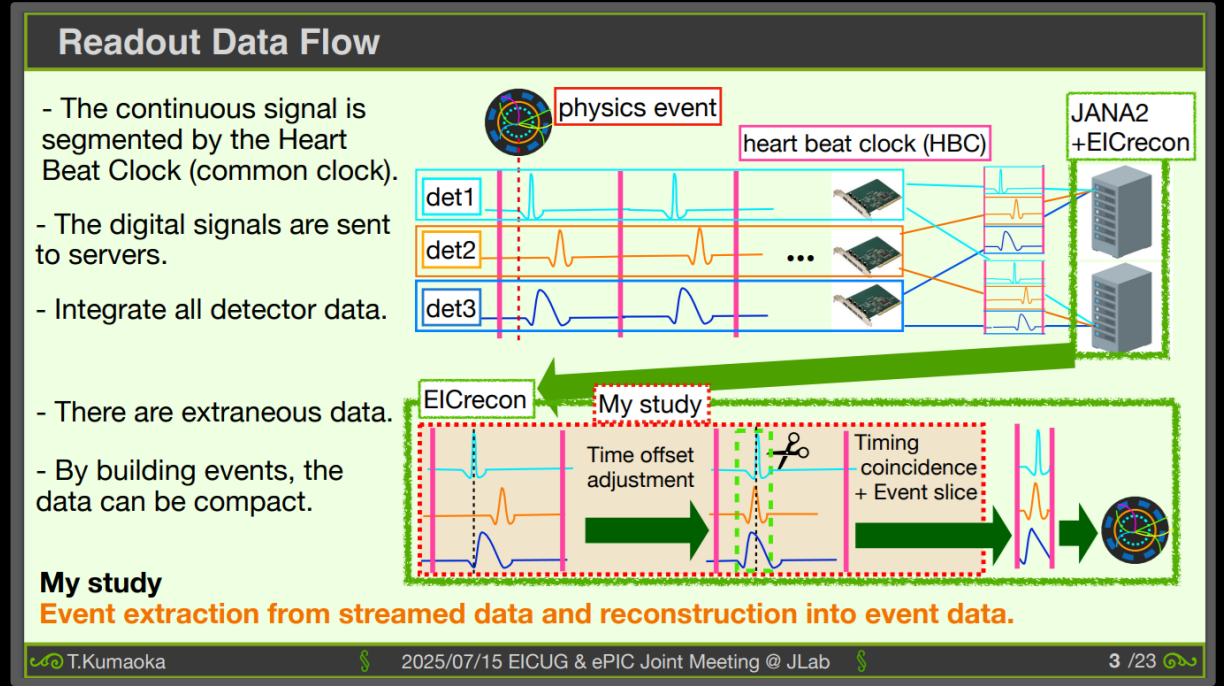
Backup | Towards Realistic Digitization



- Ongoing work to increase realism in simulation of digitization
  - **Above:** planned sequence of types/algorithms for digitization simulation in LFHCAL (left) and BIC (right)
  - LFHCAL sequence also applies to most other calos using HGCROCs
- Development being driven by **Minho Kim (ANL)** and **Simon Gardner (Glas)**

Backup | Towards SRO





- Significant development on EICrecon + JANA2 needed to process real & simulated streaming data
  - **Nathan Brei (JLab)** talked at EIC UGM about recent updates to JANA2 addressing use-cases encountered in SRO & data-taking
  - **Left:** schematic of EICrecon processing input timeframe stream with both begin-of-run and ”control” events bundled in frames
- Critical algorithm for processing streamed data: **the event builder**
  - **Takuya Kumaoka (UT-QNSI)** gave update at EIC UGM on his work towards implementing an event builder
  - **Right:** illustration of the event builder

Backup | Data Model Notes

- **The Event Data Model (EDM):** how we represent our data (broadly speaking) in software
  - This Wikipedia article provides a nice summary

- **In ePIC software:** defined and managed by the PODIO (Plain-Old-Data I/O) toolkit
  - A single YAML file defines flat(ish) data types – eg. tracks, clusters, particles, etc. – and how they relate to each other
    - **Right:** the edm4eic::track as an example
  - PODIO auto-generates necessary c++ code to implement types

- **Critical piece of our software stack!**
  - **D**efines both the types themselves and the *flow* of data
  - Ie. what is the sequence of algorithms we run during reconstruction

```
453     edm4eic::Track:
454       Description: "Track information at the vertex"
455       Author: "S. Joosten, J. Osborn"
456       Members:
457         - int32_t           type                        // Flag that defines the type of track
458         - edm4hep::Vector3f  position                    // Track 3-position at the vertex
459         - edm4hep::Vector3f  momentum                    // Track 3-momentum at the vertex [GeV]
460         - edm4eic::Cov6f     positionMomentumCovariance  // Covariance matrix in basis [x,y,z,px,py,pz]
461         - float              time                        // Track time at the vertex [ns]
462         - float              timeError                   // Error on the track vertex time
463         - float              charge                      // Particle charge
464         - float              chi2                        // Total chi2
465         - uint32_t           ndf                         // Number of degrees of freedom
466         - int32_t            pdg                         // PDG particle ID hypothesis
467       OneToOneRelations:
468         - edm4eic::Trajectory                  trajectory     // Trajectory of this track
469       OneToManyRelations:
470         - edm4eic::Measurement2D measurements     // Measurements that were used for this track
471         - edm4eic::Track      tracks           // Tracks (segments) that have been combined to create this track
472
```