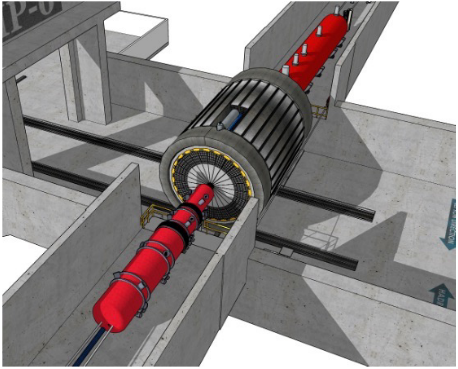
ePIC Computing Status Update

Dmitry Kalinkin (University of Kentucky)
for the ePIC collaboration

RHIC/AGS Annual Users' Meeting
May 20-23 2024

Computing for the ePIC detector at EIC



Enabling science at EIC

- Computing is an integral part of operation of a streaming readout detector, and needs to be considered as such during the design phase
- Support full set of ePIC detectors with **holistic reconstruction** to eliminate the trigger selection bias
- Rapid turnaround of 2-3 weeks between data-taking, alignment/calibrations, validation and physics analyses

Software and Computing Effort in ePIC

Software and Computing Coordinator – Markus Diefenthaler (JLab)
Deputy Coordinator for **Operations** – Wouter Deconinck (Manitoba)
Deputy Coordinator for **Development** – Dmitry Kalinkin (UKY)
Deputy Coordinator for **Infrastructure** – Torre Wenaus (BNL)

Operations WGs:

- Production
  Thomas Britton (JLab), Sakib Rahman (BNL)
- User Learning
  Stephen Kay (York), Holly Szumila-Vance (FIU)
- Validation

Development WGs:

- Physics and Detector Simulation
  Chao Peng (ANL), Sakib Rahman (BNL)
- Reconstruction
  Derek Anderson (ISU), Shujie Li (LBNL)
- Analysis Tools

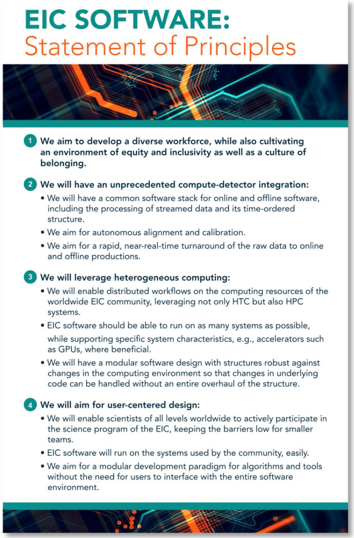
Infrastructure WGs:

- Streaming Computing Model
  Marco Battaglieri (INFN Genoa), Taku Gunji (Tokyo), Jeff Landgraf (BNL)
- Multi-Architecture Computing
- Distributed Computing

Cross-cutting WGs:

- Data and Analysis Preservation

EIC Software: Statement of principles



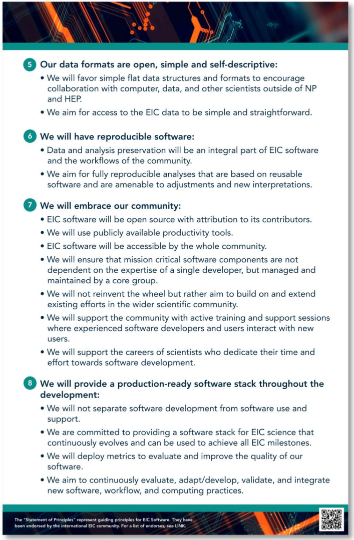
EIC SOFTWARE:
Statement of Principles

1. **We aim to develop a diverse workforce, while also cultivating an environment of equity and inclusivity as well as a culture of belonging.**

2. **We will have an unprecedented compute-detector integration:**
   - We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
   - We aim for autonomous alignment and calibration.
   - We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.

3. **We will leverage heterogeneous computing:**
   - We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.
   - EIC software should be able to run on as many systems as possible, while supporting specific system characteristics, e.g., accelerators such as GPUs, where beneficial.
   - We will have a modular software design with structures robust against changes in the computing environment so that changes in underlying code can be handled without an entire overhaul of the structure.

4. **We will aim for user-centered design:**
   - We will enable scientists of all levels worldwide to actively participate in the science program of the EIC, keeping the barriers low for smaller teams.
   - EIC software will run on the systems used by the community, easily.
   - We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment.



5. **Our data formats are open, simple and self-descriptive:**
   - We will favor simple flat data structures and formats to encourage collaboration with computer, data, and other scientists outside of NP and HEP.
   - We aim for access to the EIC data to be simple and straightforward.

6. **We will have reproducible software:**
   - Data and analysis preservation will be an integral part of EIC software and the workflows of the community.
   - We aim for fully reproducible analyses that are based on reusable software and are amenable to adjustments and new interpretations.

7. **We will embrace our community:**
   - EIC software will be open source with attribution to its contributors.
   - We will use publicly available productivity tools.
   - EIC software will be accessible by the whole community.
   - We will ensure that mission critical software components are not dependent on the expertise of a single developer, but managed and maintained by a core group.
   - We will not reinvent the wheel but rather aim to build on and extend existing efforts in the wider scientific community.
   - We will support the community with active training and support sessions where experienced software developers and users interact with new users.
   - We will support the careers of scientists who dedicate their time and effort towards software development.

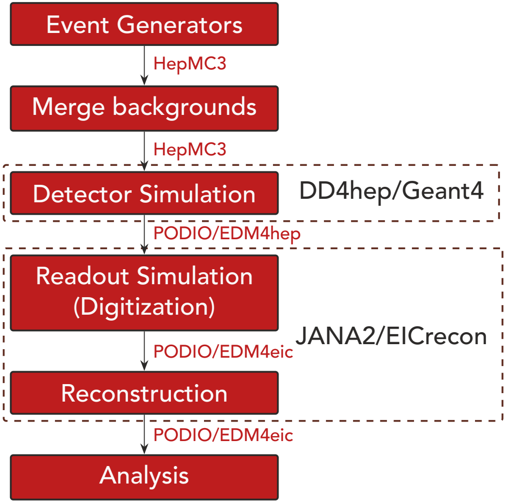
8. **We will provide a production-ready software stack throughout the development:**
   - We will not separate software development from software use and support.
   - We are committed to providing a software stack for EIC science that continuously evolves and can be used to achieve all EIC milestones.
   - We will deploy metrics to evaluate and improve the quality of our software.
   - We aim to continuously evaluate, adapt/develop, validate, and integrate new software, workflow, and computing practices.

The "Statement of Principles" represent guiding principles for EIC Software. They have been endorsed by the international EIC community. For a list of endorses, see LINK.

Endorsed at
https://eic.github.io/activities/principles.html

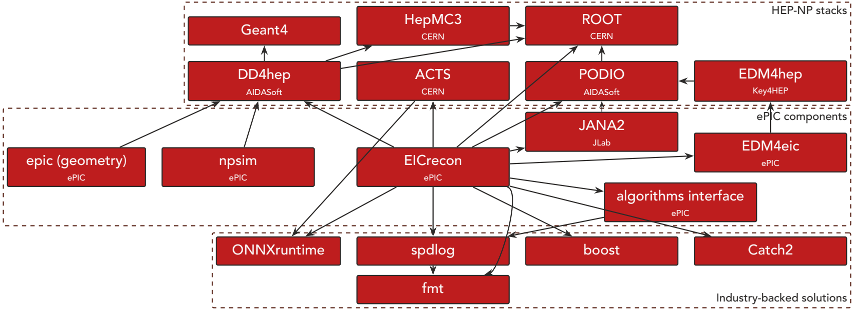
- agile development
- production-ready software stack
- meeting near-term needs of ePIC
- timeline-based prioritization
- user-centered design

Simulation framework



- Common simulation and reconstruction geometry is defined with DD4hep
- DDG4 component: interface to Geant4 with first-class PODIO/EDM4hep support
- EICrecon implements ePIC detector-specific response and digitization simulation steps

Reconstruction framework

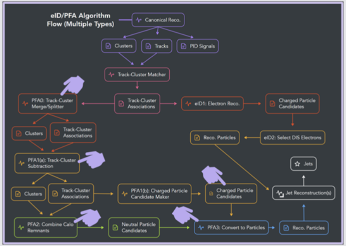


Data-processing algorithms organized

- EICrecon framework is based on JANA2 (streaming, multithreading, heterogeneous hardware)
- Modular algorithms for tracking, vertexing, calorimettry, jet reconstruciton, PID
- EDM4eic defines PODIO structures for our data, exchanged immutably between algorithms

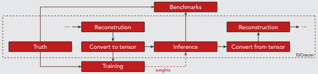
Reconstruction development

Calorimetry



- Ongoing development *e*-Finder and Particle Flow
- Track-based cluster splitting

AI/ML workflows for reconstruction and simulation

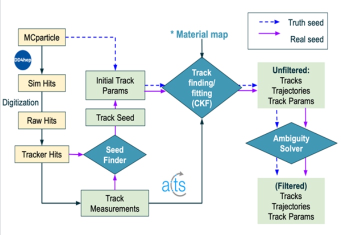


- Example for electron finder in EEEMCal using ONNX
- ePIC Hackathon/Tutorial during Frascati collaboration meeting with two problems:
  - Low-Q^2 tagger momentum calibration
  - PID in DIRC

Reconstruction development

Acts tracking and vertexing

- Full chain of track reconstruction (seeding, fitting, ambiguity resolution) in EICrecon
- Primary vertexing ready
- Secondary vertexing is under development
  - Acts vs KFParticle is evaluated
  - ACTS::AdaptiveMultiVertexFinder has initial results

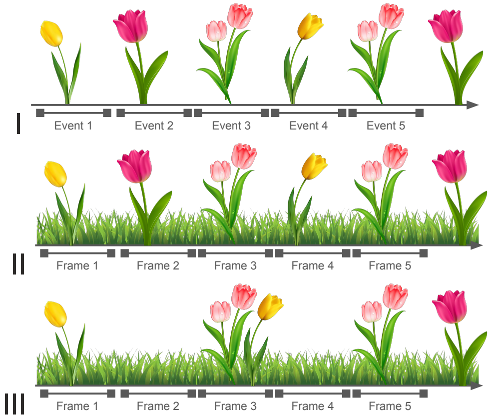


Reconstruction for RICH detectors

- IRT (Inverse Ray Tracing) 1.0 library used in EICrecon only works for single particles
- Development of IRT 2.0 integration is ongoing in a separate development branch
- Goal of providing unified plugin for DRICH and pfRICH

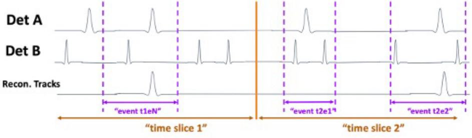
Streaming simulation and reconstruction

Simulation types to address different needs

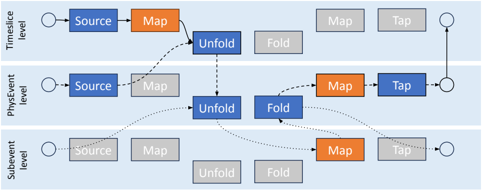


(Schematic by Wouter Deconinck)

Event reconstruction has to reassemble frames back into events



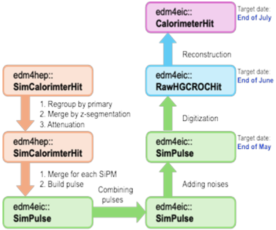
Multi-level event processing topology implemented in JANA2



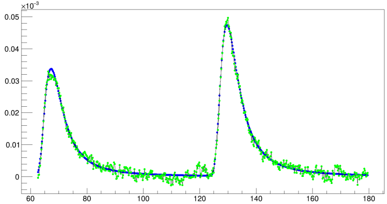
(Diagram by Nathan Brei)

Digitization and noise in simulation

Digitization is key to understanding detector response for timing and resolution



Data flow for HGCROC digitization by Derek Anderson



Example simulated pulse shapes with added noise by Simon Gardner

Rapid advancements in digitization for various subsystems

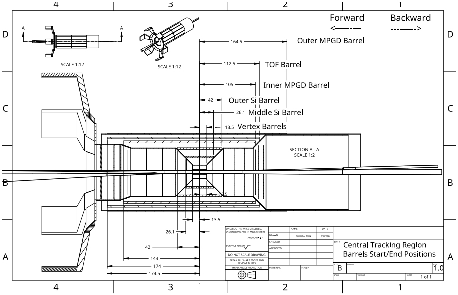
- **BTOF** for AC-LGAD
- **Low-Q^2** tagger for Timepix4 (potential application of Allpix^2 package)
- **LFHCAL & BIC** CALOROC/HGCROC for (potential application of SimSiPM package)

Strong emphasis on finding commonalities to improve review and reuse of the work.
Supporting analysis and comparisons on test beam measurements is key to validation.

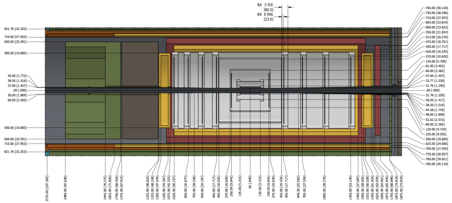
Simulation geometry validation

A major task of resolving discrepancies between engineering and simulation designs (in works since past year)

- Setup a conversion routine from DD4hep geometry to STEP CAD files.
- Working together with EIC Project to identify discrepancies. Meeting with engineers was highly productive.
- Working with DSCs to evaluate issues and implement changes as needed.



ePIC Simulation geometry converted to CAD



Project detector CAD design

ePIC Streaming Computing: Computing Model

https://doi.org/10.5281/zenodo.14675920

The ePIC Streaming Computing Model
Version 2, Fall 2024

Marco Battaglieri[1], Wouter Deconinck[2], Markus Diefenthaler[3], Jin Huang[4], Sylvester Joosten[5], Dmitry Kalinkin[6], Jeffery Landgraf[4], David Lawrence[3] and Torre Wenaus[4]
for the ePIC Collaboration

[1]Istituto Nazionale di Fisica Nucleare - Sezione di Genova, Genova, Liguria, Italy.
[2]University of Manitoba, Winnipeg, Manitoba, Canada.
[3]Jefferson Lab, Newport News, VA, USA.
[4]Brookhaven National Laboratory, Upton, NY, USA.
[5]Argonne National Laboratory, Lemont, IL, USA.
[6]University of Kentucky, Lexington, KY, USA.

Version 2 prepared for ECSAC review (November 2024)

Use Case	Echelon 0	Echelon 1	Echelon 2	Echelon 3
Streaming Data Storage and Monitoring	✓	✓		
Alignment and Calibration		✓	✓	
Prompt Reconstruction		✓		
First Full Reconstruction		✓	✓	
Reprocessing		✓	✓	
Simulation		✓	✓	
Physics Analysis		✓	✓	✓
AI Modeling and Digital Twin		✓	✓	

- Integrated system between **Streaming Computing** (Echelons 1-3) and **Streaming DAQ** (Echelon 0)
- Echelon 1 sites at host facilities
  - Symmetric access to raw data, full archiving
  - Prompt reconstruction for monitoring
- Echelon 2 sites accelerate use cases other than low-latency streaming workflows

ePIC Streaming Computing: Computing resource estimates

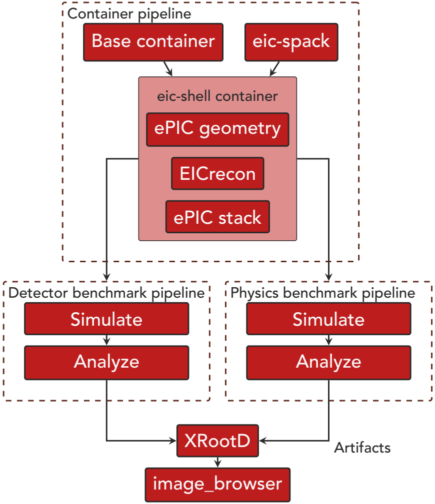
Processing by Use Case [cores]	Echelon 1	Echelon 2
Streaming Data Storage and Monitoring	-	-
Alignment and Calibration	6,004	6,004
Prompt Reconstruction	60,037	-
First Full Reconstruction	72,045	48,030
Reprocessing	144,089	216,134
Simulation	123,326	369,979
Total estimate processing	**405,501**	**640,147**

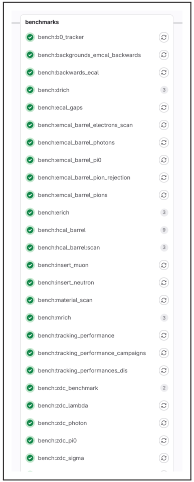
Storage Estimates by Use Case [PB]	Echelon 1	Echelon 2
Streaming Data Storage and Monitoring	71	35
Alignment and Calibration	1.8	1.8
Prompt Reconstruction	4.4	-
First Full Reconstruction	8.9	3.0
Reprocessing	9	9
Simulation	107	107
Total estimate storage	**201**	**156**

Initial estimate show that ePIC is a compute-intensive experiment.

- $\mathcal{O}(1M)$ core-years to process a year of data
  - Motivates attention to leveraging distributed and opportunistic resources from the beginning
- 350 PB storage for a year of data

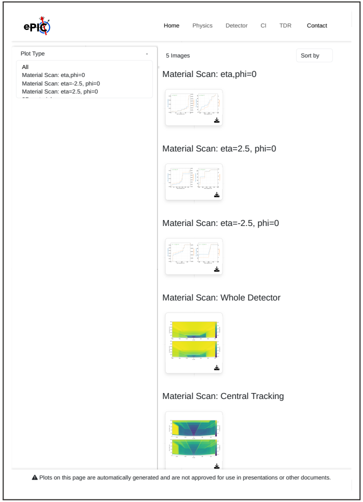
Validation workflow



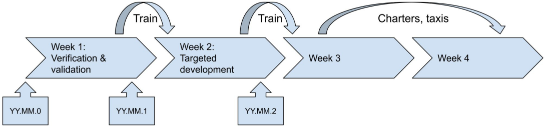


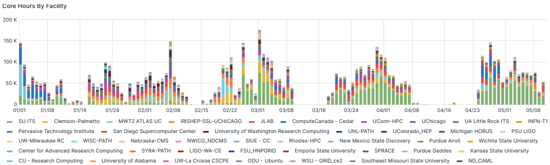
Running in a dedicated GitLab CI
(AMD EPYC 7H12 - 256 threads, 512 GB)

image_browser



Simulation Campaigns





- Delivered monthly campaigns since October 2022
- Intuitive path design provides ease of access

  Path: `/EPIC/RECO/24.09.0/epic_craterlake/DIS/NC/10x100/minQ2=1000/`

  File name: `pythia8NCDIS_10x100_minQ2=1000_beamEffects_xAngle=-0.025_hiDiv_2.1409.eicrecon.tree.edm4eic.root`

- Recently rolled out `Rucio` data management system
- Working with Analysis Coordinators and PWGs to deliver complete set of simulations covering all (pre-)TDR and Early Science studies
- Training production liasons to reduce bottlenecking
- Mixed background simulations upcoming

Growing workforce across the timezones

Weekly software news (Indico)

WG News

Physics and Detector Simulations WG

- Start evaluation of simulation vs. engineering designs
  - Creating CAD models converted from ePIC DD4hep geometry.
  - Started with tracking this week (and then PID, calorimetry).
  - Converted models (and views) will be distributed to WGs.
- Progress in CAD to DD4Hep conversion (see slides from Sam Henry and Tuna Tasali)

Recent tutorials:

- Understanding the Simulation Output – Shujie Li
- Analysis and Working with the Simulation Output – Stephen Kay
- Getting Started with a Physics Analysis – Alex Jentsch
- Inclusive Kinematics Reconstruction – Stephen Maple
- ePIC Event Display (TBA) - Dmitry Romanov

https://github.com/eic – **200+** members are the "ePIC Devs" Team

eic

Overview | Repositories 158 | Discussions | Projects 3 | Packages | Teams 14

Electron-Ion Collider (EIC) Software

Electron-Ion Collider (EIC) software, documentation and resources

Follow

SOFTWARE

138 followers · https://eic.github.io · eic-software-l-owner@lists.bnl.gov

Online documentation

Landing Page

Get started	ePIC Tutorials
HEP Software Training Center	FAQ

Welcome to the **ePIC Landing Page**!

Our mailing list: eic-projdet-compsw-l@lists.bnl.gov

Subscribe here: https://lists.bnl.gov/mailman/listinfo/eic-projdet-compsw-l

Developing Benchmarks

Benchmarks are scripts that run detector simulations and analyze the resulting data to extract quantities related to the detector performance.

Prerequisites

The following tutorial assumes basic knowledge of shell. It may be practical to use eic-shell for benchmark development, as it is the environment that will closely match the one during benchmark execution. It also assumes that you are a member of the EIC organization on GitHub and belong to the "epic-devs" team. You also need to have your local ssh key added to GitHub so that you can push.

Schedule

	Setup	Download files required for the lesson
00:00	1. Exercise 1: Analysis Scripts and Snakemake	How does one set up data analysis workflows?
00:20	2. Exercise 2: Setting up your first benchmark with pipelines	How do we create a new pipeline with GitLab CI?

~**Helpdesk** channel on Mattermost

Ping 3:17 PM

Is there option to turn off multiple scattering in npsim?

✅ 1

1 reply · Following

Building our ePIC Software & Computing community

Regular in-person meetings

- ePIC Software & Computing Meeting at CERN
  April 22-26, 2024, Indico, 103 registrants
- HSF-India/ePIC Workshop Indian Institute of Technology, Bombay, India
  May 12-17, 2025, Indico, 48 registrants

Organizing internal efforts on all fronts: Development, Simulation, Streaming/DAQ, Validation and User Learing.

Strengthen collaboration with HEP, specifically Acts, CERN EP-SFT, HSF, Key4HEP, Rucio, and with developers of MCEGs.

Summary and Outlook

- ePIC Software effort embraces open development model with aim at sustainability
- Building on top of NP-HEP community's past experience, we are working together with it on improving common set of state-of-art tools
- Software development is supported with focus on priorities identified by the collaboration
- Today, ePIC Software is well-prepared to deliver crucial results needed for finalizing detector design and validating its fitness for the purposes of the EIC science program
- Design of the Distributed Computing Model for streaming data is progressing well