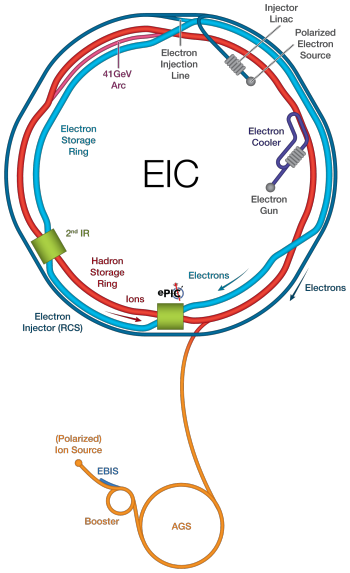
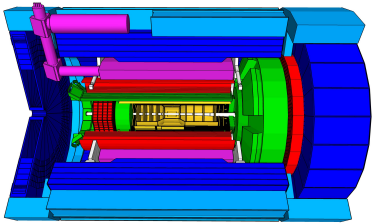
Collaborative software and maintainability for ePIC experiment at EIC

Dmitry Kalinkin (University of Kentucky)
for the ePIC collaboration

Conference on Computing in High Energy and Nuclear Physics
October 19–25, 2024

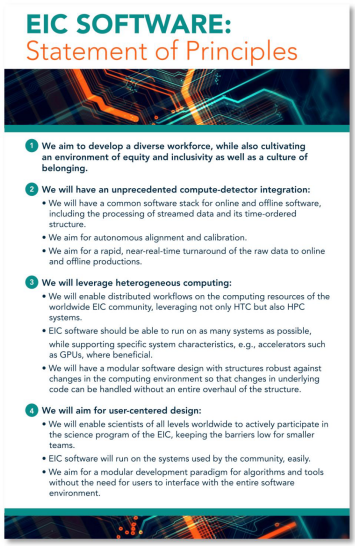
ePIC detector at EIC





- **EIC** to occupy RHIC tunnel at BNL, co-hosted with JLab
- **ePIC** at IP6, addresses EIC science program per EIC white paper and NAS report
- Strict performance requirements imposed by physics goals
- Hermetic central detector, optimized for asymmetric collisions at EIC
- Advanced Software and Computing effort with support for streaming readout, processing at multiple tiers of facilities, reconstruction with online calibration

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.

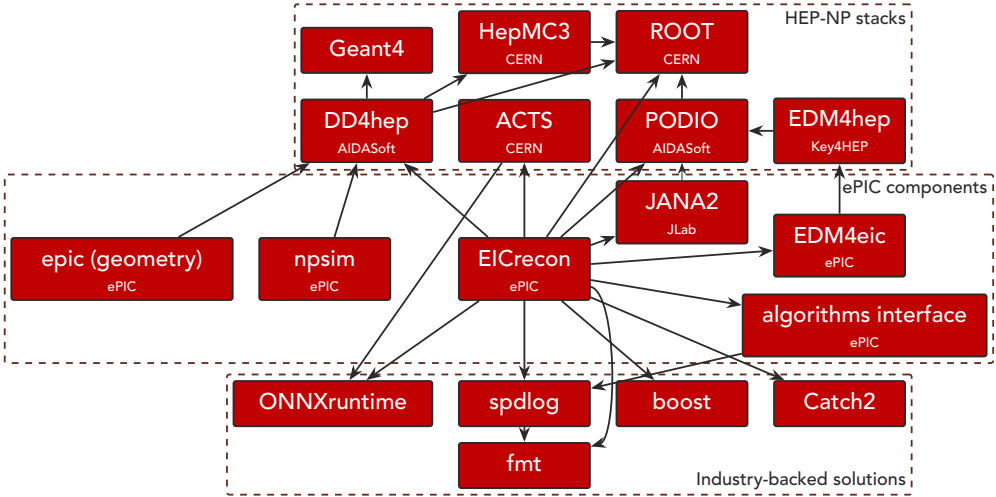
8 We will provide a production-ready software stack throughout the development:
- We will not separate software development from software use and support.
- We are committed to providing a software stack for EIC science that continuously evolves and can be used to achieve all EIC milestones.
- We will 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 endorsers, 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

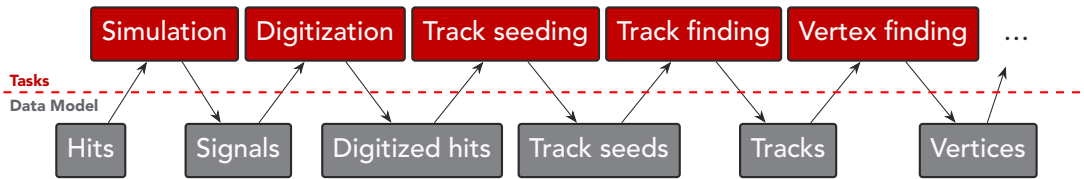
ePIC Software Stack



- Modular Simulation, Reconstruction and Analysis Software Stack powered by the NP-HEP community tools

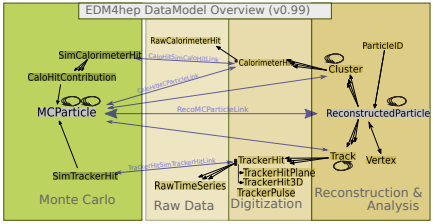
Data model in data-processing framework

- "If you've chosen the right data structures and organized things well, the algorithms will almost always be self-evident." – Rob Pike, 1989
- We spend a lot of time on educating our collaborators on understanding and using of our Event Data Model
- Event model additions and modifications have to be discussed in weekly ePIC Software & Computing meeting and reviewed closely on GitHub.



PODIO

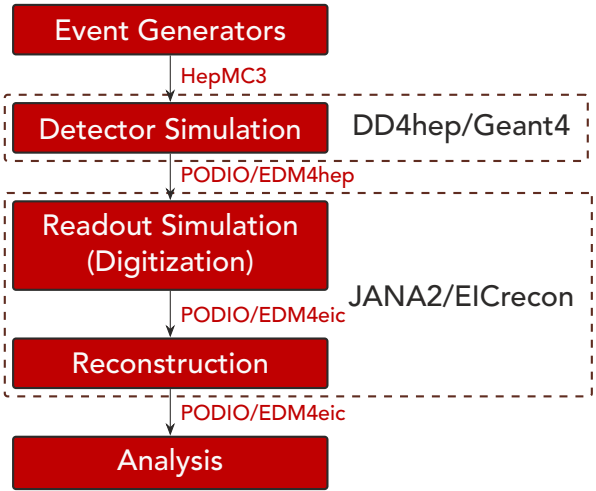
- … is a YAML-based format for defining data structures
- … has a C++/Python API
- … restricts nesting of data structures (hence, Plain Old Data)
  in favor of the relational model
- … does not impose row-wise/columnar ordering for in-memory data
- … implements on-disk representation in ROOT (TTree or RNTuple) and SIO formats, but also allows other implementations



EDM4hep / edm4hep.yaml

```
394     edm4hep::CalorimeterHit:
395       Description: "Calorimeter hit"
396       Author: "EDM4hep authors"
397       Members:
398         - uint64_t cellID  // detector specific (geometrical) cell id
399         - float energy [GeV]              // energy of the hit
400         - float energyError [GeV]         // error of the hit energy
401         - float time [ns]                 // time of the hit
402         - edm4hep::Vector3f position [mm] // position of the hit in world coordinates
403         - int32_t type                    // type of hit
```

Simulation framework



- Common simulation and reconstruction geometry is defined with DD4hep
- DDG4 component: interface to Geant4 with first-class PODIO/EDM4hep support
- Embedding of backgrounds (beam-gas interaction, synchrotron radiation) is available as a HepMC preprocessor
- EICrecon implements ePIC detector-specific response and digitization simulation steps

Reconstruction framework

Data-processing algorithms organized

- JANA2 framework is a rewrite of JANA multithreaded framework with focus on Streaming DAQ and heterogeneous hardware support
- EDM4eic defines PODIO structures for data that can be passed from algorithm to algorithm
- EICrecon uses JANA2 as backbone for passing immutable data within processed events and timeframes
- Modular algorithms for tracking, vertexing, calorimettry, jet reconstruciton, PID

Near-term goals

- External algorithm wiring configurability for JANA2
- Metadata handling (extend support in JANA2)
- Consistent conditions DB interface
  (evaluate `nopayloaddb` – HSF's design reference implementation)

Deployment and Workflows

Software-defined infrastructure

- Publically developed workflows:
  - GitHub Actions
  - GitLab CI
  - Snakemake
  - Bash script

  workflows for running tests, container building, simulations, data analysis
- Arifacts are publically accessible (Downloadable Artifacts and Web Pages, XRootD)

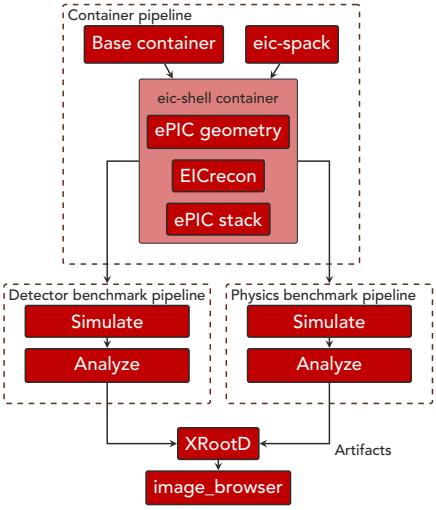
Continuous Integration

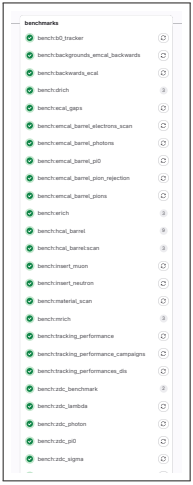
- Coding suggestions (formatting, static and dynamic analyses)
- Quick tests, distribution comparison plots

Continuous Delivery

- Release and nightly Docker/Singularity containers
- Export geometry to CAD (.step) format
- Export algorithm graph (Graphviz)
- Material maps, Calibrations, Machine Learning weights
- Sub-detector performance plots
- ePIC Detector Physics performance plots (towards Analysis-as-a-service)

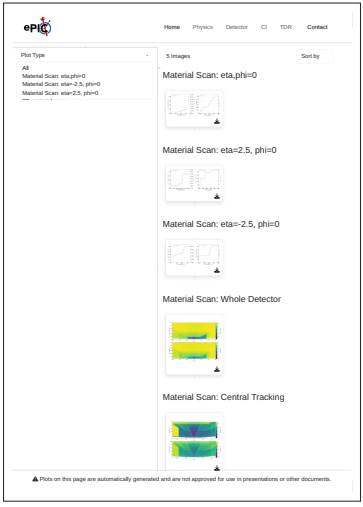
Validation workflow



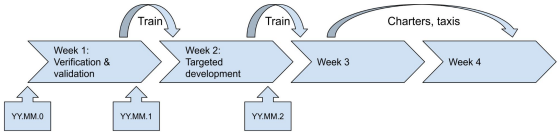


Running in a dedicated GitLab CI (AMD EPYC 7H12 - 256 threads, 512 GB)

image_browser



Simulation Campaigns



- Regular schedule for software releases and accompanying simulation campaigns
- Delivered **29** campaigns since October 2022
- Intuitive path design provides ease of access

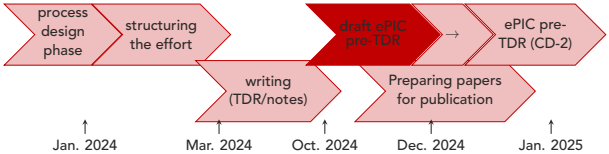
Path: `/EPIC/RECO/24.09.0/epic_craterlake/DIS/NC/10x100/minQ2=1000/`

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

  - Container version
  - Geometry configuration
  - Physics process
  - Event generator
  - Electron and proton beam energies
  - Minimal Q^2

- HepMC3 files provided and validated by Physics WGs + setups available with single particles for detector studies

Software and Simulation Readiness for Techinal Design Report



- All hands effort was organized at a Collaboration meeting in **January 2024**:
- 4 groups were given identical **``Charge'' with 6 questions** like:
  - "…What features are missing in the detector simulations for ePIC that are required for detector and physics studies for the TDR? …"
  - "…What validation of the detector simulations is required for the TDR other than the consistency check with engineering design via the Detector Geometry Matrix?"
  - "…How would you describe the minimum viable product for the reconstruction for the TDR?"
- Individual responses of each group were **discussed together** after and summarized at a **plenary close-out session**
- An **internal document** was formed based on out outcome of the discussions

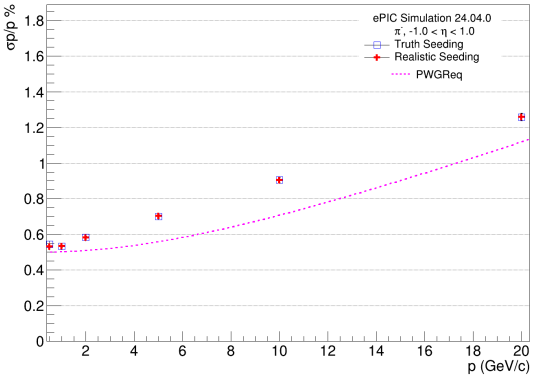
⇒ drill into details in follow-up meetings

⇒ initiate/delegate, inquire, track progress

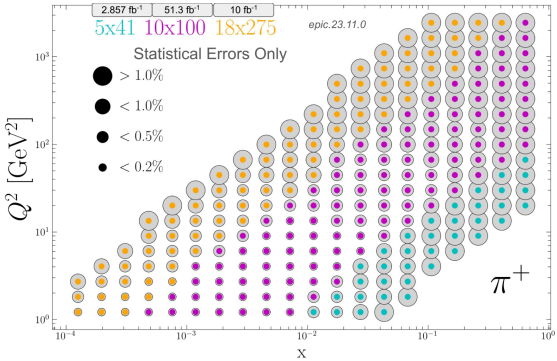
⇒ a success!

ePIC Software supporting studies for (Pre-)TDR

Select studies



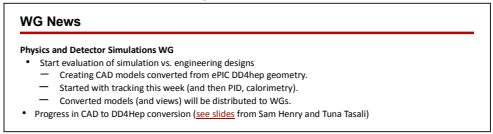
Detector benchmark, candidate for a plot in TDR: Momentum resolution for the tracker
plot by Shyam Kumar



Plot for **physics** section of the TDR:
Expected statistical uncertainty of Unpolarized TMD PDFs
plot by Gregory Matousek

Growing workforce across the timezones

Weekly software news
(slides also posted on Indico)



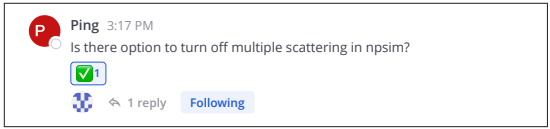
WG News

Physics and Detector Simulations WG

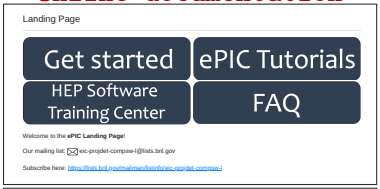
- Start evaluation of simulation vs. engineering designs
  - Creating CAD models converted from ePIC DD4hep geometry.
  - Started with tracking this week (and then PID, calorimetry).
  - Converted models (and views) will be distributed to WGs.
- Progress in CAD to DD4Hep conversion (see slides from Sam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

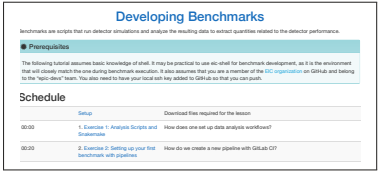
https://github.com/eic – **200+** members are the "ePIC Devs" Team

~Helpdesk channel on Mattermost



Online documentation





Building our ePIC Software & Computing community

Regular in-person meetings

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.

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
- 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