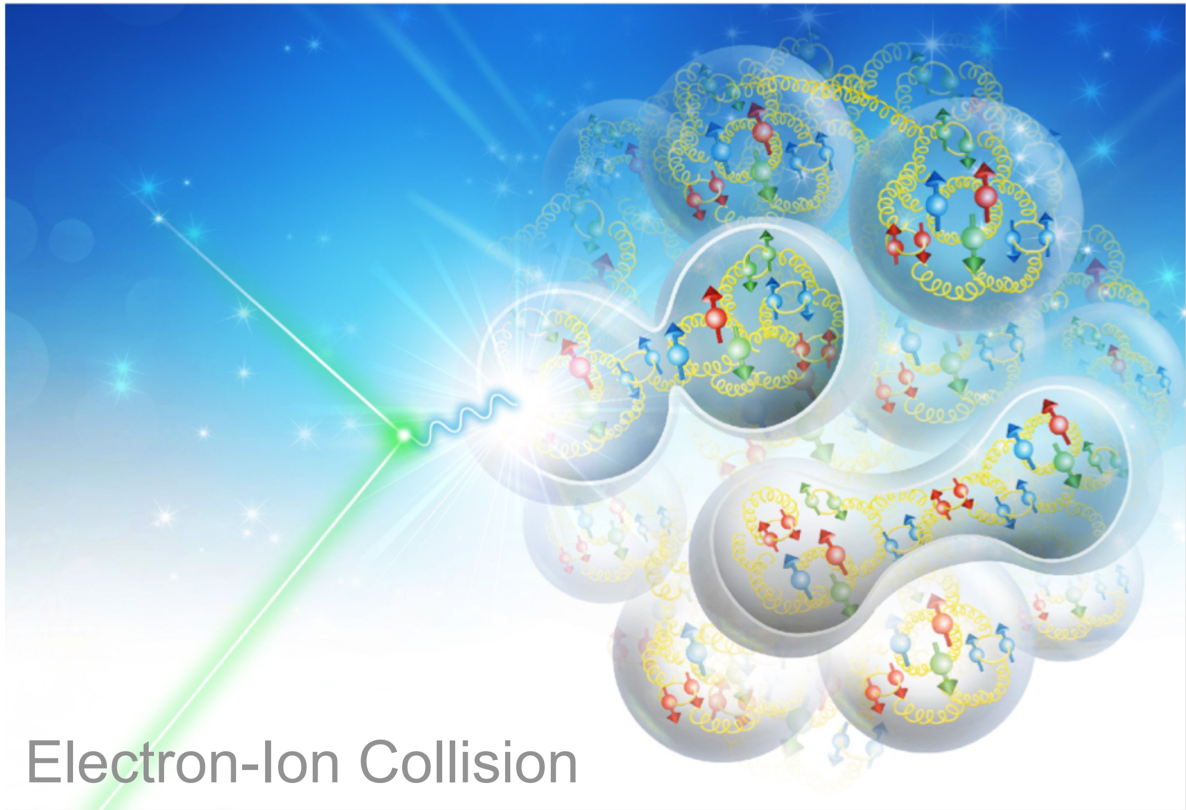
Software Initiatives for the Electron-Ion Collider

Markus Diefenthaler (EIC[2] Jefferson Lab)
PI EIC Software Consortium
Convener EICUG Software Working Group



Electron-Ion Collision

Summary from EIC Software Meeting

Why an Electron-Ion Collider?

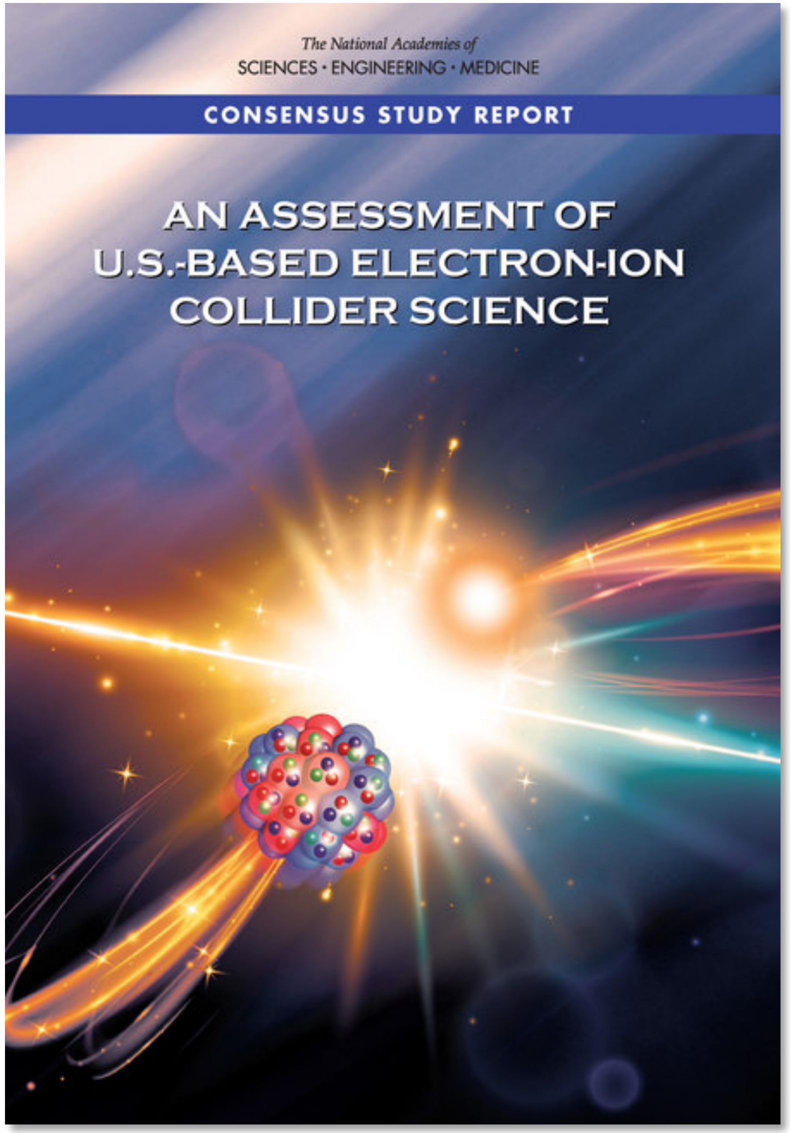
- Outstanding questions raised both by the science at HERMES/COMPASS/JLAB and RHIC/LHC, have naturally led to the science and design parameters of the **Electron-Ion Collider (EIC)**.

- EIC will enable us to embark on a **precision study of the nucleon and the nucleus at the scale of sea quarks and gluons**, over all of the kinematic range that are relevant.

- What we learn at JLAB 12 and later EIC, together with advances enabled by FRIB and LQCD studies, will open the door to:

Transformation of Nuclear Physics

… **but only if we build the right machine/detector/analysis**.



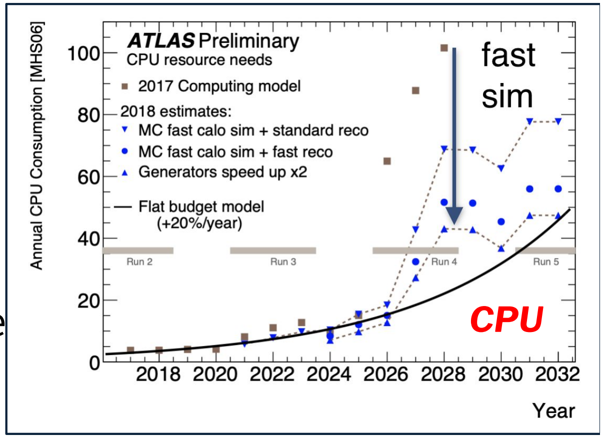
Computing Vision for the EIC

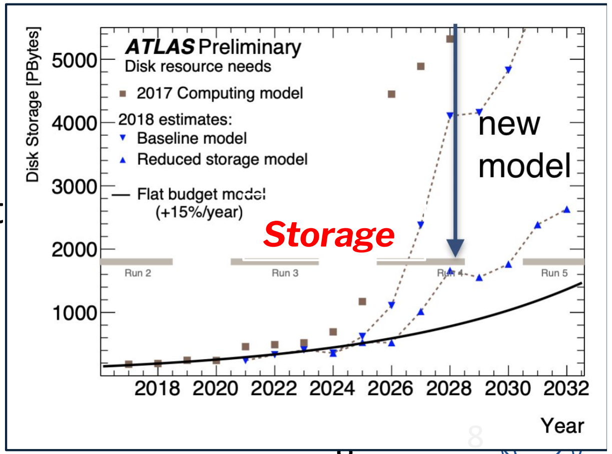
“The purpose of computing is insight, not numbers.”

Richard Hamming (1962)

S&C challenges for the HL-LHC

- Up to **10x the event rate**: much more data, and more complex events
- Pile-up of up to **200 proton-proton interactions** per beam crossing
  - Conventional tracking approaches lead to a combinatorial explosion
- Flat-budget hardware improvements fall **far short of requirements**, extrapolating present computing model
  - Must evolve our processing and data management approaches
- ATLAS is already today **compute-limited** in its science; computing must scale with data volume and complexity or we leave physics on the floor
- Heavily **data-intensive and distributed**, and will become moreso
  - Must feed our often **I/O-bound applications** with data efficiently at scale in the distributed environment
- HEP **software is too serial** for future architectures
  - HEP code is ~1 op/cycle, HPC code ~4, vector instructions up to 8
  - Major re-engineering *beyond multithreading* required to maximize benefit from modern & future CPUs (vectorization, pipelining, accelerators)
- **We cannot afford to buy our way out of the problem with hardware -- we must innovate in software**



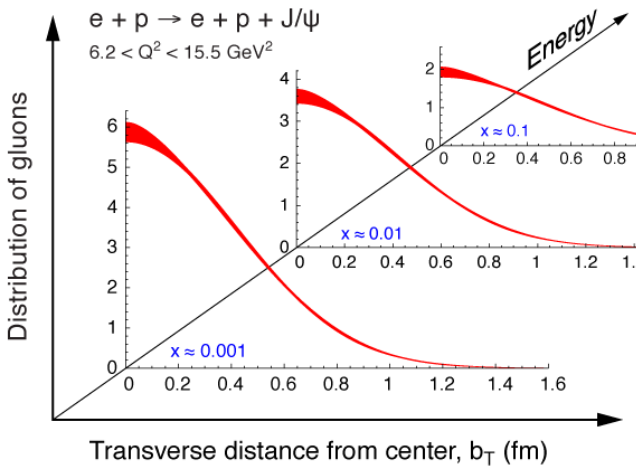


Computing Challenges in Nuclear Physics

NP experiments driven by beam intensity, polarization, exquisite control of background and systematics

multi-dimensional challenges

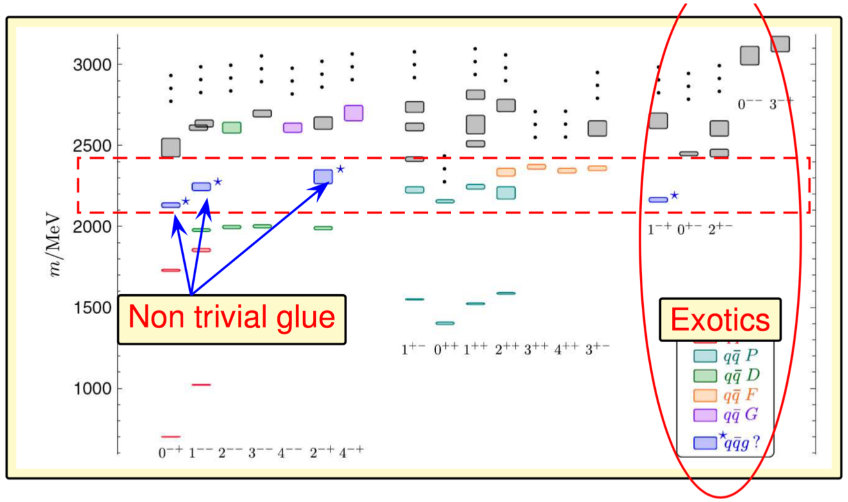
example 3D imaging of quarks and gluons



high statistics in five or more dimensions and multiple final-state particles

multiple channel challenges

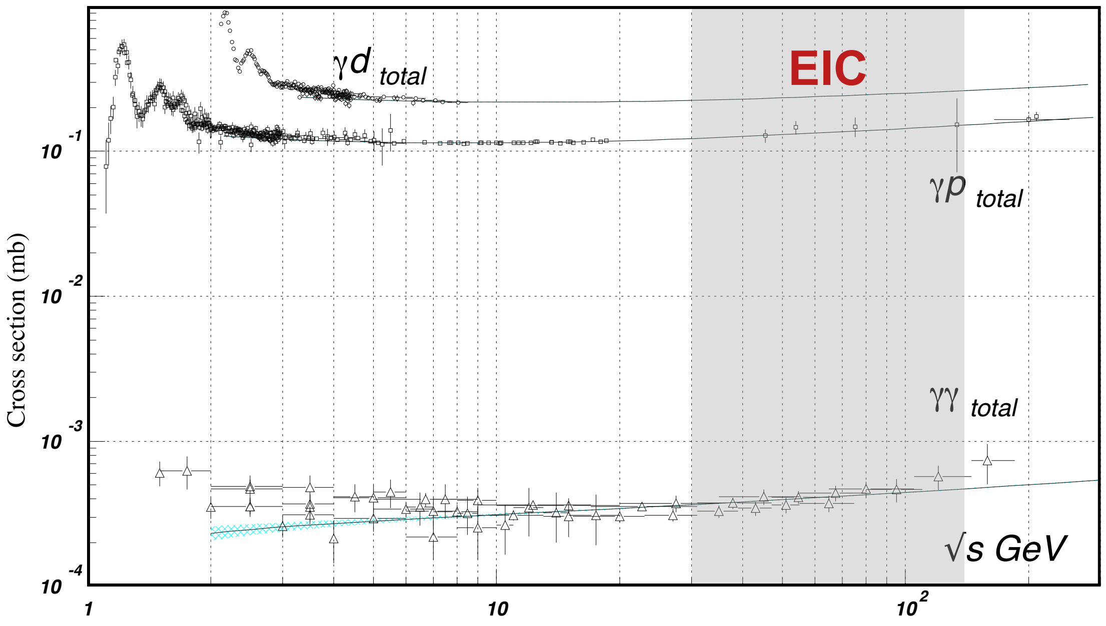
example discovery search of gluon-based exotic particles (PWA, 1000s of waves)



strongly iterative analysis for reliable, model-independent analysis

Rates at the EIC

Dominant cross-section contribution



Signal and background rates at the EIC

$\mathcal{L} = 10^{34}\ cm^{-2}\ s^{-1}$ = 10 kHz/μb

Photoproduction background estimator
cross-section ~ 100 μb
interaction rate ~ 1 MHz

ep (E_e = 10 GeV, E_p = 100 GeV) signal
cross-section ~ 45 μb
interaction rate ~ 450 kHz

LHC ~69 mb at √s=7 TeV
RHIC ~42 mb at √s=200 GeV

Data size data size / event ~ 100kb data size / s ~ 100Gbit

Towards the next generation research model in Nuclear Physics

EIC rates
- expected data rates similar to next phase LHCb
- not enormous rates creates opportunity for other initiatives

Torre Wenaus (BNL)

goal

NP research model not changed for over 30 years
Science & Industry remarkable advances in computing & microelectronics
evolve & develop **NP research model** based on these advances

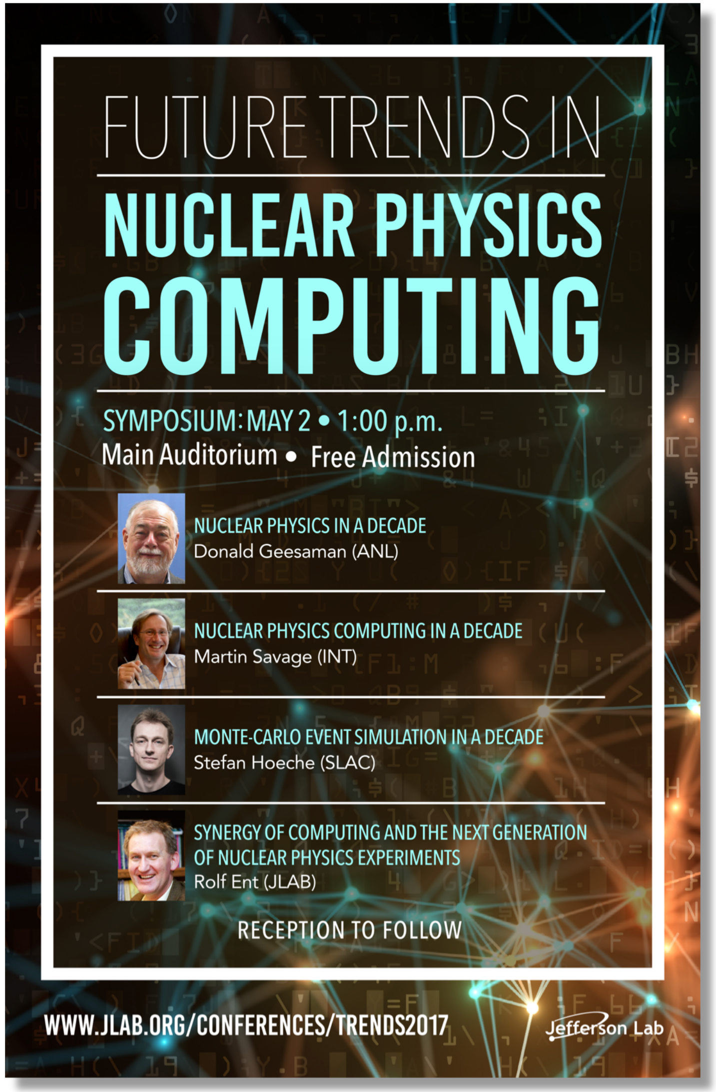
rethink

how measurements are compared to theory
- examine capabilities of event level analysis (**ELA**) taking the multi-dimensional challenges of NP fully into account

how experimental data are handled
- identify ways to speed up analysis in the context of **ELA**

how we read out detectors and assemble detector data
- investigate capabilities of streaming readout in view of **ELA**

Future Trends in Nuclear Physics Computing



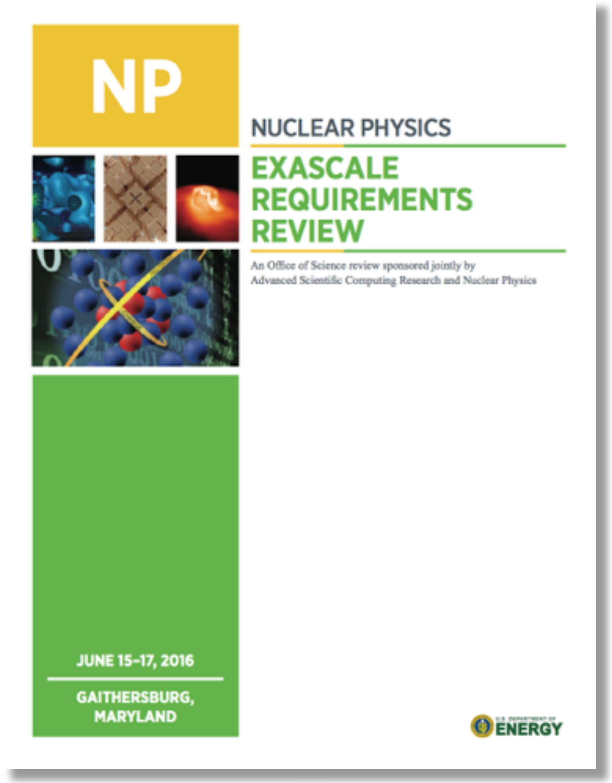
Donald Geesaman (ANL, former NSAC Chair) *"It will be **joint progress of theory and experiment** that moves us forward, not in one side alone"*

Martin Savage (INT) *"The next decade will be looked back upon as a **truly astonishing period in Nuclear Physics** and in our understanding of fundamental aspects of nature. This will be **made possible by advances in scientific computing** and in how the Nuclear Physics community organizes and collaborates, and how DOE and NSF supports this, to take full advantage of these advances."*

Implications of Exascale Computing

Past efforts in lattice QCD in collaboration with industry have driven development of new computing paradigms that benefit large scale computation. These capabilities underpin many important scientific challenges, e.g. studying climate and heat transport over the Earth.

The EIC will be the facility in the era of high precision QCD and the first Nuclear Physics facility in the **era of Exascale Computing**: This will affect the interplay of experiment, simulations, and theory profoundly and result in a new computing paradigm that can be applied to other fields of science and industry.



Petascale-capable systems at the beamline

- **unprecedented compute-detector integration**, extending work at LHCb
- requires fundamentally new and different algorithms
- computing model with AI at the DAQ and analysis level and a compute-detector integration to deliver **analysis-ready data from the DAQ system**:
  - responsive calibrations in real time
  - real-time event reconstruction and filtering
  - physics analysis in real time

A similar approach would allow **accelerator operations** to use real-time simulations and AI over operational parameters to tune the machine for performance.

Streaming readout software from GSI and ALICE

Mohammad Al-Turany (GSI)

FAIR GSI

Summary

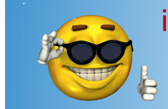
- ALFA allows developers to write their specific code in whatever language they choose as long as that language can send and receive data through message queues.
- allows non-expert to write messaged based code without going into the details of the transport or the system below
- offers a **clean** and **maintainable** and **extendable** interface to the existing different data transport (ZMQ, nanomsg, shared Memory, OFI, ..etc)

21 May 2019 M. Al-Turany, EIC Software Meeting 37

FAIR GSI

Summary, contd.

- provides utilities to deploy topologies (via DDS) to computing clusters, online clusters as well as on a laptop
- has a plugin based system for configuration and control of user processes
- Developer productivity is improved by allowing them to focus on the unique requirements of their tasks instead of spending time on application infrastructure (“plumbing”).



https://github.com/FairRootGroup

21 May 2019 M. Al-Turany, EIC Software Meeting 38

Giulio Eulisse (CERN)

DESIGNING A NEW COMPUTING ARCHITECTURE FOR ALICE IN RUN 3: ALICE O2

ALICE can cope with the challenges of Run 3 only by a radical redesign of its software and computing architecture.

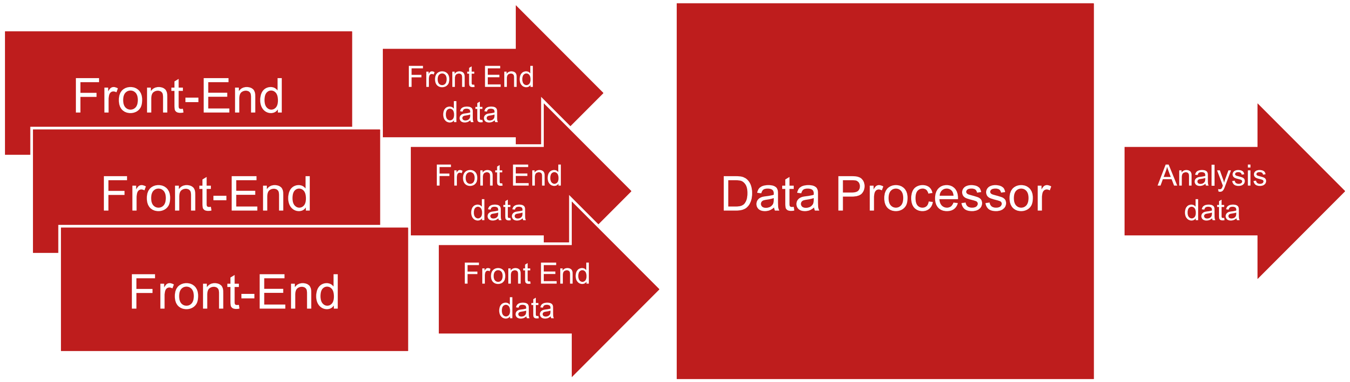
- *New architecture based on the experience accumulated in the ALICE HLT during Run 1 / Run 2.*
- *Focus on online data compression, only keeping raw data and AOD objects, **trading computational cost for storage**.*
- ***Simplified Data Model** in order to improve I/O performance.*
- ***Appropriately chosen algorithms** providing higher throughput for negligible loss of physics performance. Algorithms tuned for vectorisation / exploitation of hardware accelerators (GPUs).*
- *Ability to port software components in a **gradual manner.***
- ***Close collaboration with the physics community** in order to organise analysis efforts.*
- ***Close collaboration with GSI and FAIR** experiments on a common software stack (**ALFA**).*

ALICE O2: SOFTWARE FRAMEWORK IN ONE SLIDE

Data Processing Layer (DPL)	Abstracts away the hiccups of a distributed system, presenting the user a familiar "Data Flow" system. ➤ ***Reactive-like design*** *(push data, don't pull)* ➤ ***Declarative Domain Specific Language*** *for topology configuration (C++17 based).* ➤ ***Integration*** *with the rest of the production system, e.g. Monitoring, Logging, Control.* ➤ ***Laptop mode****, including graphical debugging tools.*
Data Layer: O2 Data Model	Message passing aware data model. Support for multiple backends: ➤ **Simplified, zero-copy** *format optimised for performance and direct GPU usage. Useful e.g. for TPC reconstruction on the GPU.* ➤ **ROOT based serialisation.** *Useful for QA and final results.* ➤ **Apache Arrow based.** *Useful as backend of the analysis ntuples and for integration with with other tools.*
Transport Layer: ALFA / FairMQ[1]	➤ **Standalone processes** *for deployment flexibility.* ➤ **Message passing** *as a parallelism paradigm.* ➤ **Shared memory** *backend for reduced memory usage and improved performance.*

20

Streaming Readout and Real-Time Processing



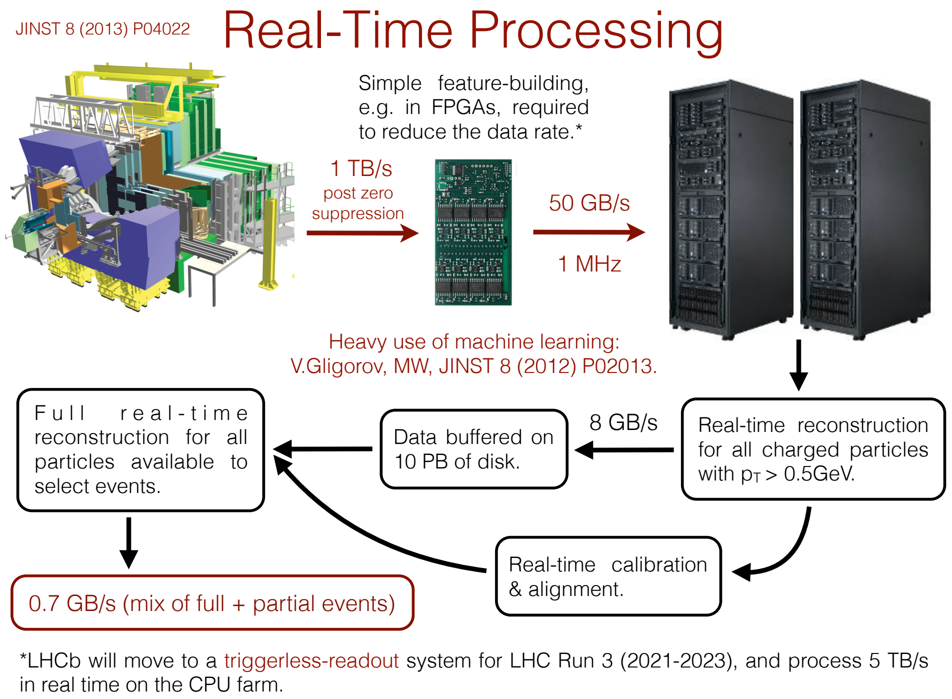
Data Processor

- assembles the data into events
- outputs data suitable for final analysis (**Analysis data**)

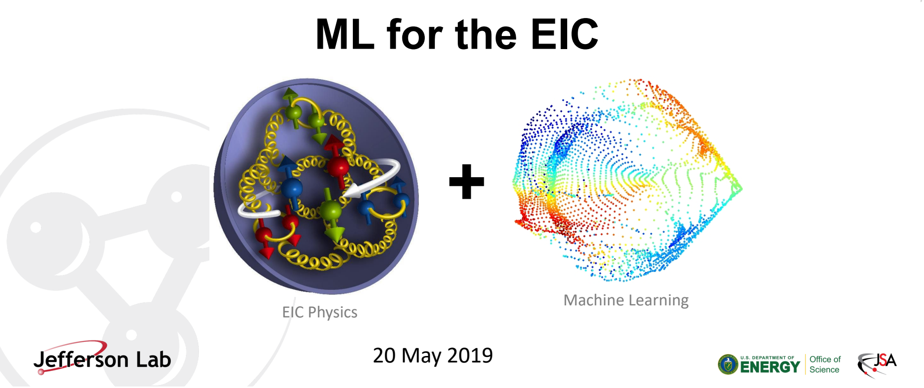
Features (among others)

- ideal for machine learning
- automated calibration and alignment
- real-time reconstruction of events
- event selection and/or labeling into analysis streams
- automated anomaly detection
- responsive detector

LHCb Example



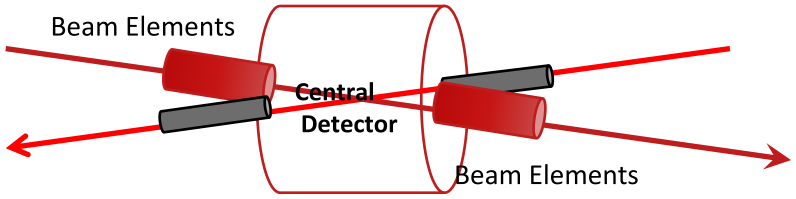
Dmitry Romanov (JLAB)



Machine-Detector Interface (MDI)

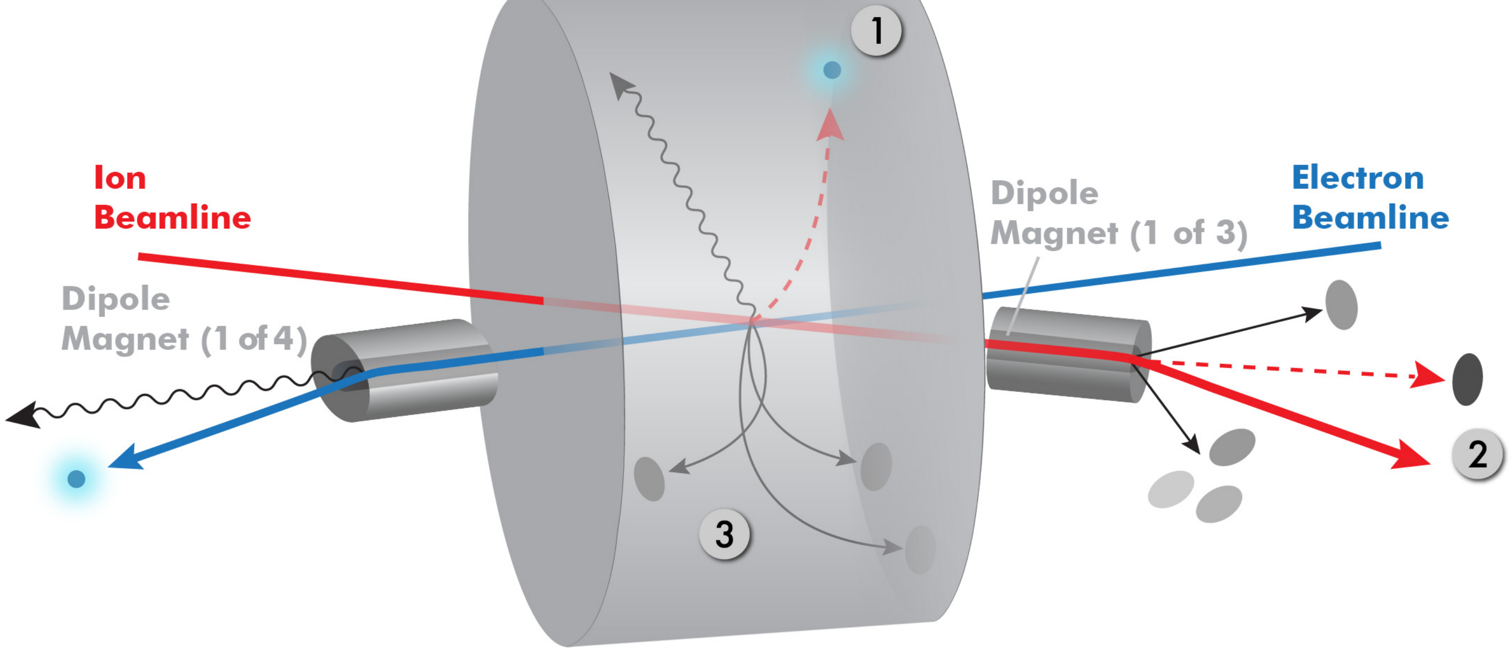
Integrated interaction region and detector design to optimize physics reach

The aim is to get **~100% acceptance** for all final state particles, and measure them with good resolution.



Experimental challenges:

- beam elements limit forward acceptance
- central Solenoid not effective for forward



Possible to get ~100% acceptance for the whole event.

Beyond MDI

Integration of DAQ, analysis and theory to optimize physics reach

Integration of DAQ, analysis and theory

- not about building the best detector
- but the best detector that fully supports streaming readout and fast algorithms for real-time analysis
- research model with seamless data processing from DAQ to data analysis using AI and strong interplay experiment - theory

Jefferson Lab

Recommendations from HEP (ATLAS) Torre Wenaus (BNL)

Torre Wenaus (BNL)

Torre Wenaus

- various software and computing roles in ATLAS, LHC Computing Grid (LCG), STAR, BaBar, SSC, LEP, (…)
- recent term as ATLAS computing coordinator
- now focused on HL-LHC computing R&D for ATLAS
- and setting up a new HEP and NP software group at BNL

Full simulations It's Geant4. **Fast simulations** Look for potential for commonality. **Geometry description** DD4HEP

C++ and python Do we have the right balance between C++ and python, particularly with the rise of the python scientific software stack and accelerators?

Code complexity Fine-grained processing among micro-services based on message passing and data on the wire is something we do outside the framework now; first thoughts of moving it inside.

Fine-grained event processing Breaking out of file-based processing to flexibility support granular processing down to event level pays off, particularly on HPCs, and particularly if you engineer it in early rather than retrofitting.

Event data model and I/O EDM simplicity, concurrent design/development of transient and persistent EDM, consider streaming and accelerator use cases; minimize file dependencies in metadata usage, don't add complexity and degrade performance for being technology agnostic

Analysis lessening *chaotic analysis* by doing analysis in managed production is a big resource payoff

Software infrastructure Be modern, it pays big dividends.

Software Initiative for the EIC

EIC Software Consortium

Existing software frameworks for the EIC

ANL TOPSiDE detector concept (ILC software variant)

BNL BeAST detector concept: EICroot (FairRoot variant)

BNL ePHENIX detector concept (fun4all)

JLAB JLEIC detector concept (GEMC)

Software Review by EIC Community in November 2017

- EicRoot, Fun4All , GEMC, and the ANL software are actively maintained.
- The analysis environments for the EIC will be chosen when the EIC experimental collaborations will form.
- Until then, we will examine the **requirements** for the EIC analysis environment and work on the **R&D** aspects of the EIC analysis environment.

EIC Software Consortium

EIC SOFTWARE CONSORTIUM

Goals and focus

- work on common interfaces among EIC simulation tools
- explore new avenues of software development (e.g., AI)
- **reach out to the EIC community**
    - communicate present status of EIC software
    - bring existing EIC software to end users
    - produce publicly available consensus-based documents on critical subjects
    - provide vision for the future

ESC members

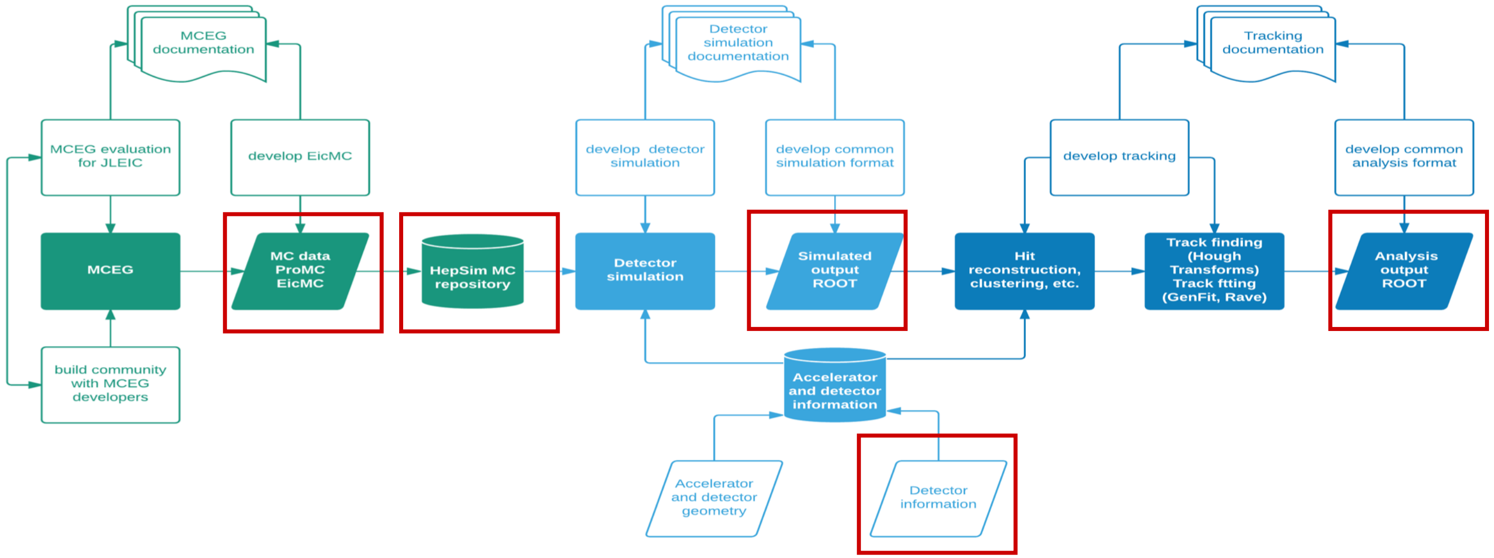
ANL, BNL, JLAB, LUND, INFN, SLAC, Trieste, W&M

Part of EIC Generic Detector R&D program

Common interfaces

Advice from ILC effort

- facilitate interoperability
- focus on exchange detector designs and data
    - get the event data model right and keep it open
    - pick a detector definition which is exchangeable



Norman Graf (SLAC)
“It’s very difficult to ~~herd cats~~ keep physicists from re-inventing the wheel and writing new software packages.“

MCEG

BeAST container (EICroot)

ePHENIX container (fun4all)

JLEIC container

TOPSiDE container

reconstruction

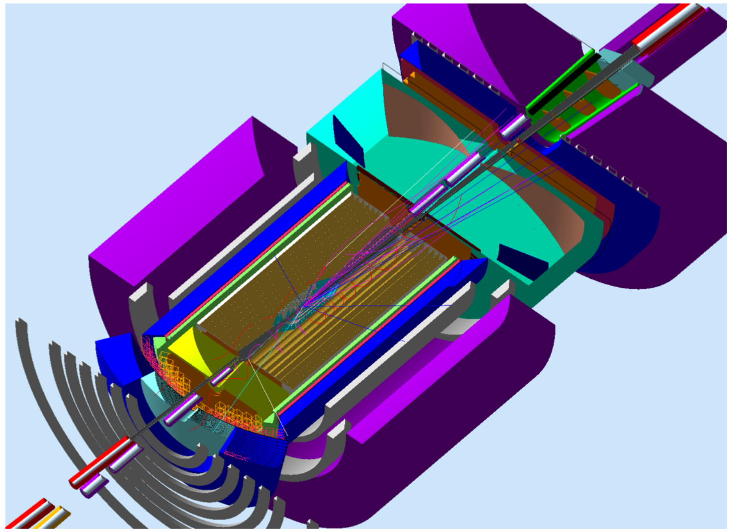
The worldwide EIC community

EIC User Group (EICUG)
Currently **867 members** from **184 institutions** from **30 countries**.

Physicists around the world are thinking about and are defining the **EIC science program**.

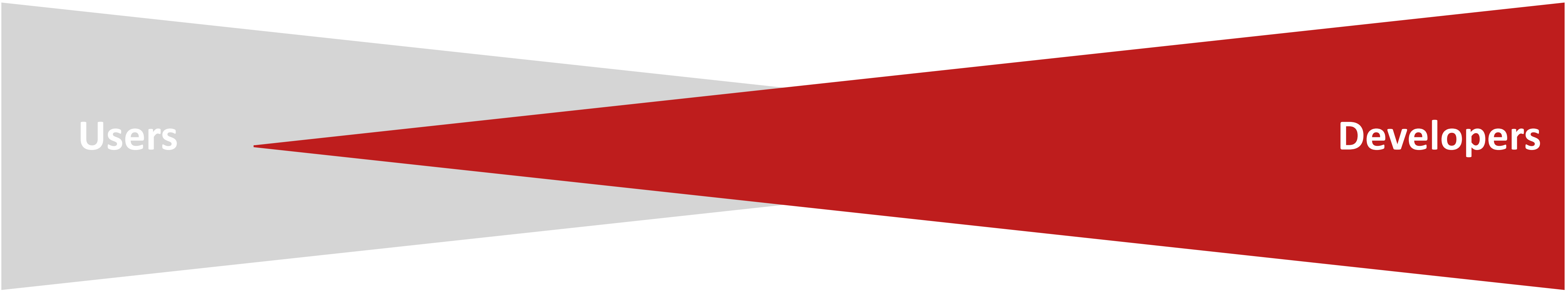
EICUG Software Working Group

- simulations of physics processes and detector response to enable quantitative assessment of measurement capabilities and their physics impact
- build on ESC effort, conveners chosen from ESC: Andrea Bressan, MD, Torre Wenaus



EIC Software Users

Novice: No experience | Level of experience | Expert: Highly experienced



User involved in

Ongoing EIC project
Software ✓
Documentation ✓
Requests none

EIC User Group
Common Software ✗
Common Documentation ✗
Requests software, documentation

Generic Detector R&D projects
Software ✓
Documentation ✗ - ✓
Requests common software

EIC Software Groups (beyond the simulation effort at the labs)

High Energy Physics

CERN ROOT
Possible collaboration

Geant4
Established collaboration

HEP Software Foundation
Possible collaboration

MCnet
Started collaboration

Nuclear Physics

EIC Software Consortium
Community Endorsement ✗
Funding ✓

Same software suite Seamless data processing from DAQ to data analysis using AI

EIC Streaming Readout Consortium
Community Endorsement ✗
Funding ✓

EICUG Software Working Group
Community Endorsement ✓
Funding ✗

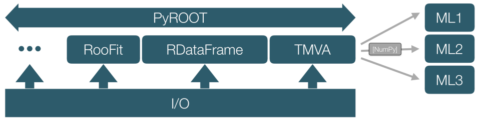
Evolution

- ROOT is a very lively project
- maintenance of old code but several new important features are being developed
- Focus on key areas important for HEP
  - parts seeing by every physicists
- Focus on performances
  - Efficient code
  - Parallelization and SIMD vectorisation whenever possible
  - Transparent usage of GPU (e.g. in deep learning)
- Improved user interfaces
  - make usage of several new features of new C++
  - maintain a dual C++/Python API (uniqueness of ROOT)

New Developments

What are we focusing on ?

- Data frame for declarative analysts
- new Web based graphics and GUI
- new I/O data interfaces
- new PyROOT
- new C++ Histograms
- modernisation of TMVA
- faster RooFit



Distributed Analysis

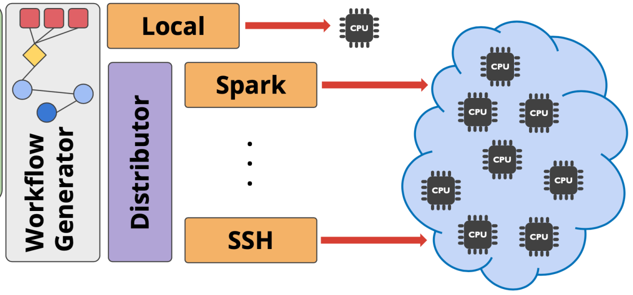
Investigate and prototype a complement to PROOF

- Parallelism on many nodes
- Transparent distribution
- Support several different backends

Re-use RDF interface: Minimal/No change in analysis code

```
d = RDataFrame (“t”, dataset)
f = d.Define(...)
     .Define(...)
     .Filter(...)

h1 = f.Histo1D(...)
h2 = f.Histo1D(...)
h3 = f.Histo1D(...)
```



JavierCVilla/PyRDF

Working prototype available!

Outlook

- Essential to maintain expertise of data analysis in HEP community
- Steer ROOT to maintain it competitive to alternative solutions
  - make use of external tools whenever useful
    - at the lower level (Clang) or for GUI (OpenUI)
  - provide interfaces to powerful external components
    - e.g Python tools for Machine Learning (e.g. Numpy, Tensorflow)
- We need to prove itself against these existing alternative
  - accepting challenge to deliver a simpler, friendly and more robust ROOT

CERN Root and EIC Software

Discussion

EIC Community

- How many physicists and where are they? How are they organized? E.g. how do they receive training?
- Where will they recruit their software experts from? Physicists? Do they need an easy migration path for physicists from analysis to detector reconstruction to software expert?

Data

- data rates, data "kind" ("event data model"), overview of the different ones relevant for storage (RAW / RECO / AOD / tuple)?
- expected overlap of data for different analyses: will they all look at the same "files", or analysis-group specific striping, skimming etc?

Computing

- Computing model: everything HPC ("exascale"), or HPC to laptop, or HEP-style distributed tier-1,-2,-3?
- How much willing to define / restrict vs leave door open for the future? Converge early for better tools versus stay flexible but risk lack of convergence in the community.
- How much can EIC provide sustainability itself, how much does it rely on external entities? In other words, what is the model for sustainable software within EIC and for external tools EIC relies on? What does the MoU say?
- Agreement on language for experiment / reconstruction frameworks?

Analysis

- Statistical analysis: Do you need to rely on advanced modelling and fitting tools such as RooFit?

Machine Learning

- What are the possible ML applications for EIC?
- How is expected to use ML models at the various stages (e.g. reconstruction, simulation and analysis) and relying on which tools?

Time line

- timeline for need of analysis package?
- timeline for reconstruction framework?

EIC Software challenges

I/O performance comparisons

abstract data model with simulated data

Speed comparisons

abstract data model with mock-up analysis

Detector Simulation

- **collaboration with Geant4 International Collbaboration**
  - liaison: Makoto Asai (SLAC)
- reach out to EIC Community
  - online catalogue of detector simulations
- **Detector Simulation R&D**
  - containers and tutorials for EIC detector simulations
  - coordinate input for Geant4 validation based on EIC physics list maintained by SLAC Geant4 group

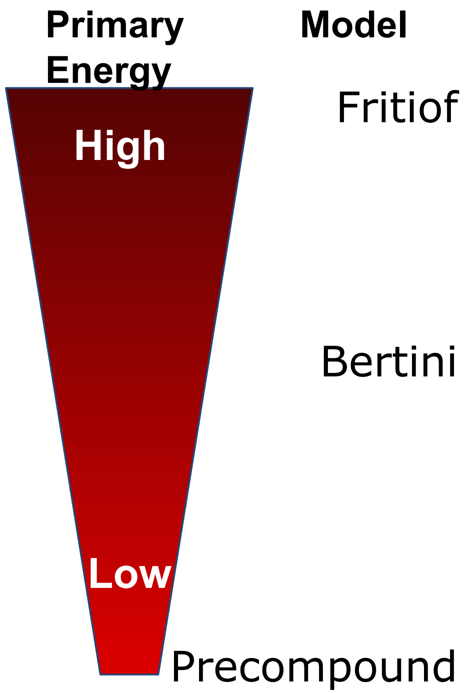
09/24 **Technical forum on EIC** as part of

09/23 – 09/27 Geant4 Collaboration Meeting (JLAB)

coordinate with EIC Detector R&D

EIC

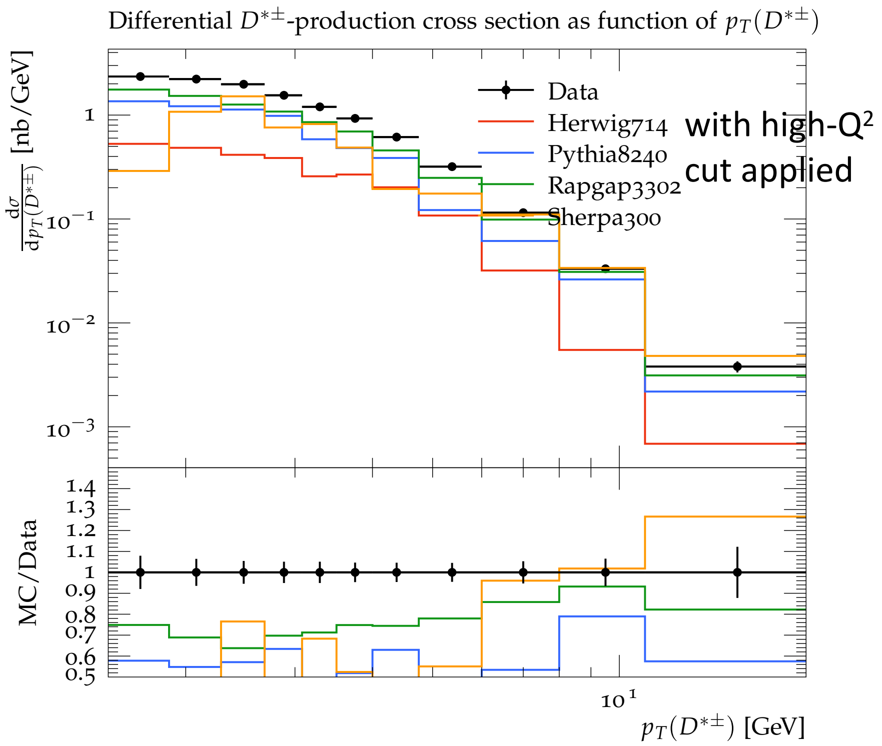
- energy range is different from LHC
- validation, tuning and extension including test beam studies



Outlook

- We have a wide ranging and ambitious physics programme in HEP and in associated disciplines
- Our experiments are highly data intensive now and require high quality software and computing
- The landscape for software is becoming ever more challenging
- Working together on common problems is not only the best use of our resources, our funding agencies will mandate it
- HSF is now well established to help HEP achieve that goal
  - Roadmap and active working groups in key areas
- From the start, working with allied sciences has been an ambition

We would be delighted to do that with the EIC software developers



Detailed comparisons between modern MCEG and HERA data

- workshop on **Rivet for ep** (Feb 18—20 2019)
- mailing list **rivet-ep-l@lists.bnl.gov**
- HERA data not (yet) included in MCEG tunes

- TMDlib, xFitter software tools available to EICUG community
- DIS framework being developed by all general purpose parton-shower MCEG (Pythia, Herwig, Sherpa)
- TMD Monte Carlo tools: Cascade, KaTie (unpolarized TMD) – to be developed for polarized TMDs; nuclear TMDs; QED corrections
- NLO + TMD framework needed (a la MC-at-NLO, Powheg, . . . : Pythia/Herwig framework not sufficient)
- EICUG – MCnet co-operation much needed for EIC physics

Example for EIC MCEG: Pythia (1978 – now)

General-purpose MCEG

- extensively used for e^+e^-, ep and pp physics, e.g. at LEP, HERA, Tevatron, and LHC
- as a building block used in heavy-ion and cosmic-ray physics
- recent pA effort in Pythia8 with Angantyr model

Pythia 6 product of over thirty years of progress

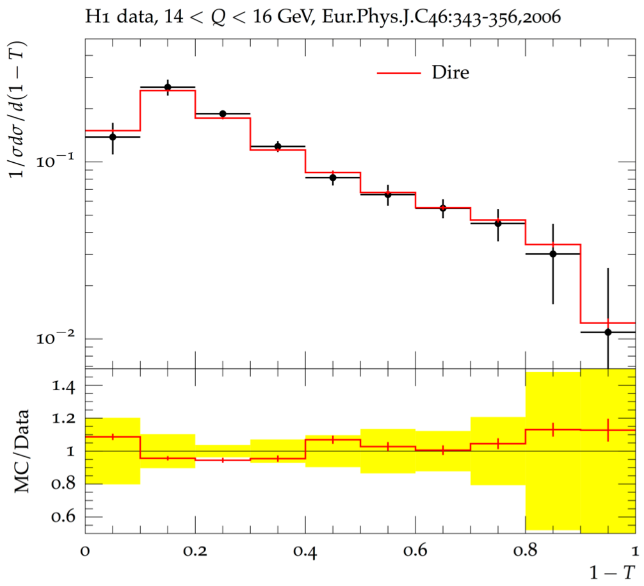
Pythia 8 successor to Pythia 6, standalone generator, but several optional hooks for links to other programs are provided

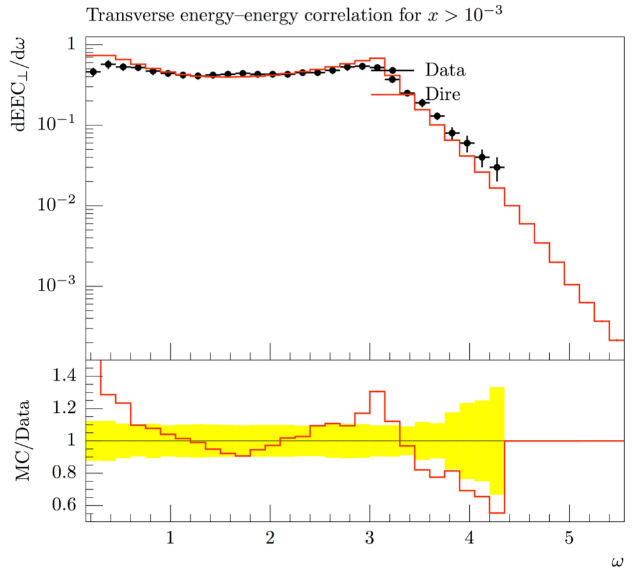
MCEG2018 and **MCEG2019**

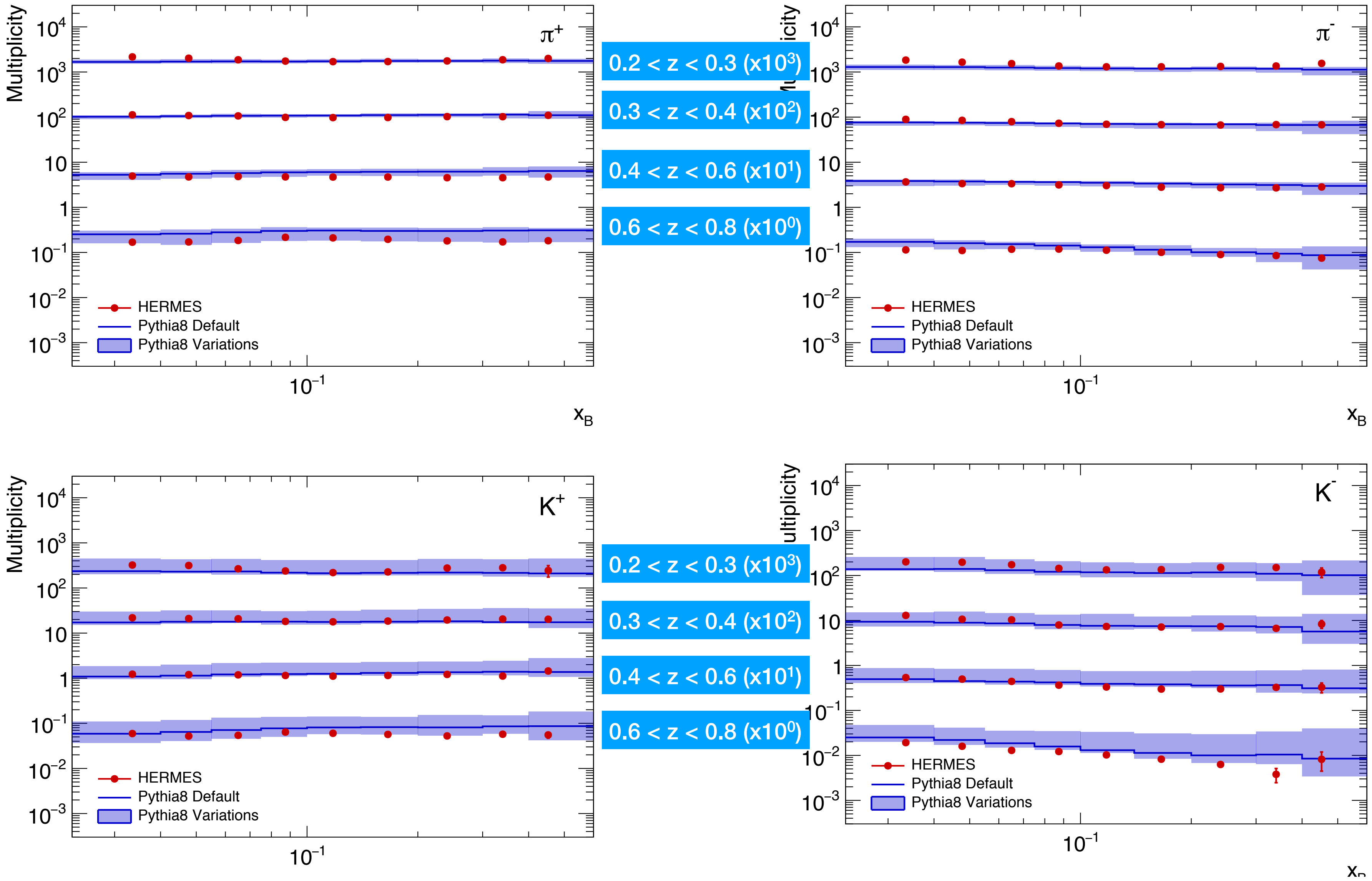
- possible to generate DIS events with the new dipole shower implementation
- higher-order corrections via Dire plugin, soon part of Pythia core
- photoproduction for hard and soft QCD processes, also hard diffraction

DIS in Pythia8

JLAB LDRD







Jefferson Lab

Getting started with Pythia8

Container for Pythia8+DIRE by Nadine Fischer (Pythia)

jupyter README ✔ 8 minutes ago Logout

File Edit View Language Plain Text

```
Welcome to the Jupyter notebooks for Pythia 8 and DIRE!


You have the choice to run the following notebooks:

pythiaPI.ipynb
  Gives a basic idea of the Pythia 8 event generator, by using the Python
  interface of Pythia 8. You can adjust a set of parameters and choose
  from different different histograms to be plotted.

pythiaRivetPI.ipynb
  Shows how to use the Pythia 8 event generator, together with Rivet,
  by using the Python interface of Pythia 8.

pythiaRivet.ipynb
  Shows how to use Pythia 8, together with Rivet, by using an already
  compiled executable called pythiaHepMC. You can adjust a set of parameters
  and a settings file is created.

pythiaRivetUS.ipynb
  As pythiaRivet.ipynb, but uses a prepared settings file, to be provided
  by the user.

direRivet.ipynb
  Shows how to use Pythia 8 with the DIRE parton shower, together with
  Rivet, by using the default DIRE executable. You can adjust a set of
  parameters and a settings file is created.

direRivetUS.ipynb
  As direRivet.ipynb, but uses a prepared settings file, to be provided
  by the user.

direEvent.ipynb
  Pythia 8 with the DIRE parton shower, graphical output of one event
  with the default DIRE exectuable.
  The process can be choosen as well as a few basic parameters.

tuning.ipynb
  Tuning with Professor, Rivet, and Pythia 8 / DIRE.
```

Jupyter notebook interface

Pythia 8 standalone

This notebook gives a basic idea of the Pythia 8 event generator, by using the Python interface of Pythia 8. You can adjust a set of parameters and choose from different different histograms to be plotted.

First, lets import all neccessary modules.

In [1]:
```
import os, sys, pythia8
from plotting import MULTIHIST
import py8settings as py8s
```

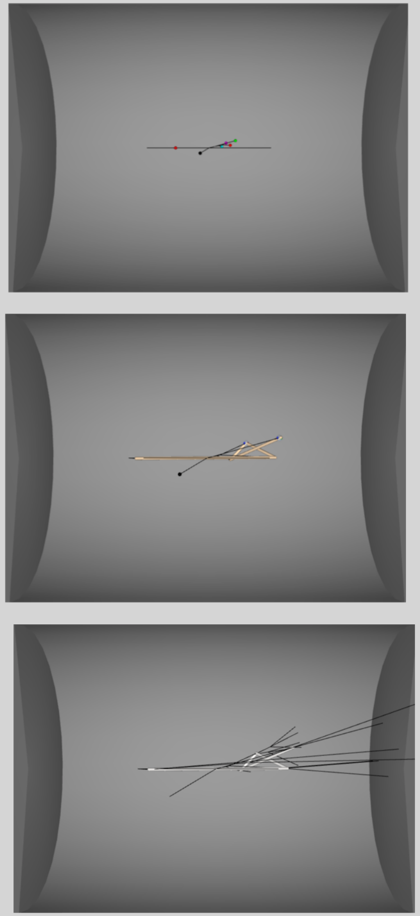
Now we create a Pythia 8 object and apply the settings to define the incoming beams. More settings can be adjusted later.

In [2]:
```
# Setup pythia, apply beam settings.
pythia = pythia8.Pythia()
py8s.beam_settings(pythia)
```

You can now set the parameters for the incoming beams:

beam A id [Beams:idA]	e-
beam B id [Beams:idB]	p
beam frame type [Beams:frameType]	2: back-to-back beams with different energies, set Beams:eA and Beams:eB
CMS energy for Beams:frameType = 1 [Beams:eCM]	65.7
beam A energy for Beams:frameType = 2 [Beams:eA]	10.8
beam B energy for Beams:frameType = 2 [Beams:eB]	100

Visualization of ep collision



Jefferson Lab

EIC Software Consortium (ESC)

Software projects for the EIC

User interface for EIC simulations for EICUG

ejana-gui container image

JupyterLab web interface (work in progress)

ejana-app container image (work in progress)

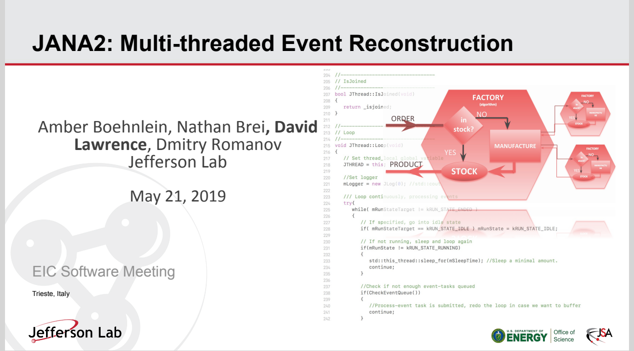
eJana (beta version)

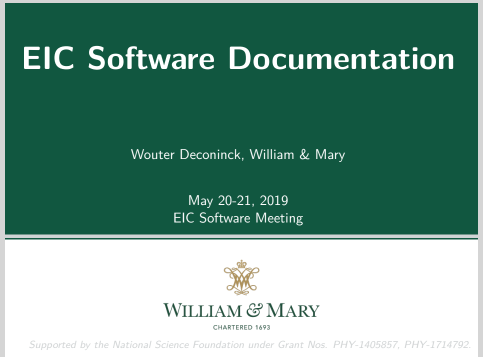
Fast simulations (work in progress)
Monte Carlo event generators
eic-smear used for EIC Whitepaper

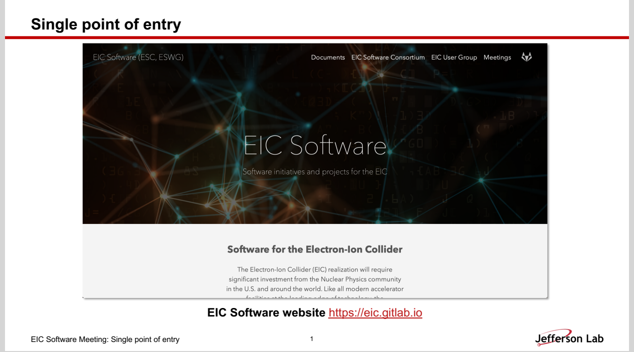
Alterative for fast simulations (tbd)
Monte Carlo event generators
Geant4 in fast mode

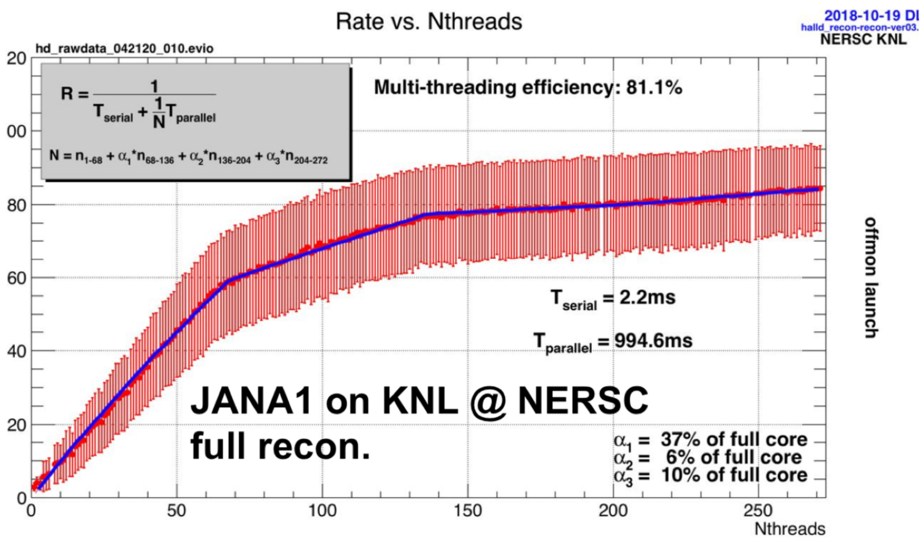
Full simulations (beta)
Monte Carlo event generators
Geant4
Event reconstruction toolkit

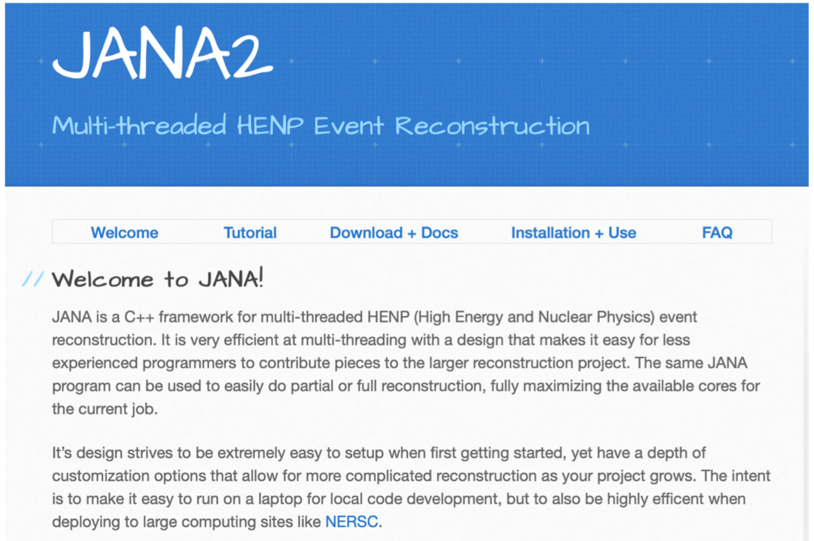
Detailed Status Reports











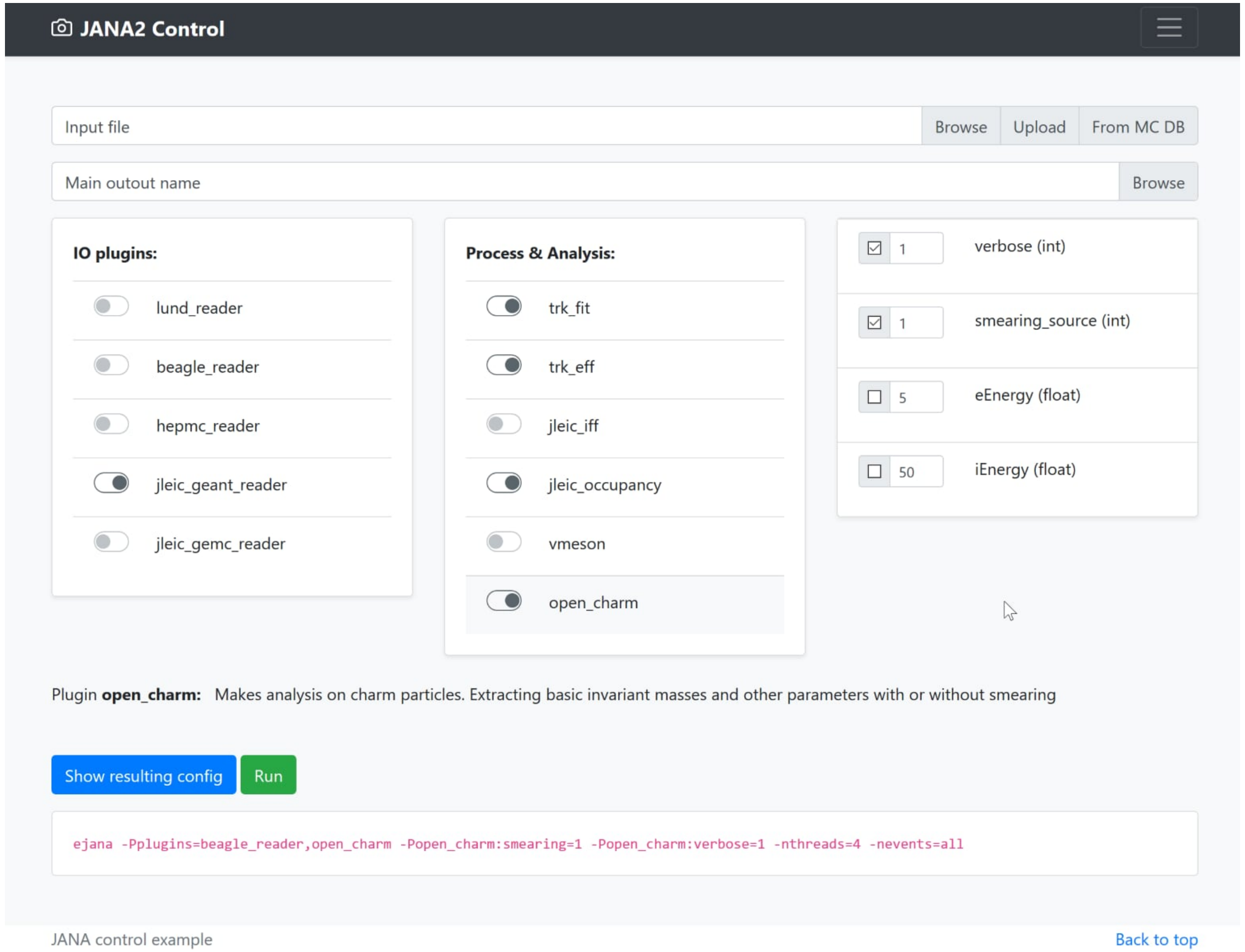
JANA C++ event processing framework

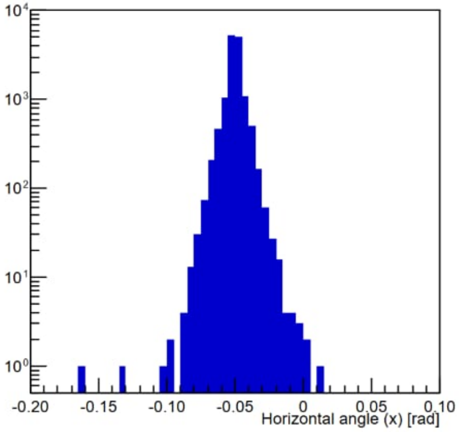
- **factory model** on demand interface, user-centered design
- multi-threaded with > 10 years experience
- **plugin support** provide mechanism for many physicists to contribute, multi-threading external to contributed code (parallelizer)

JANA2 under development (JLAB LDRD)

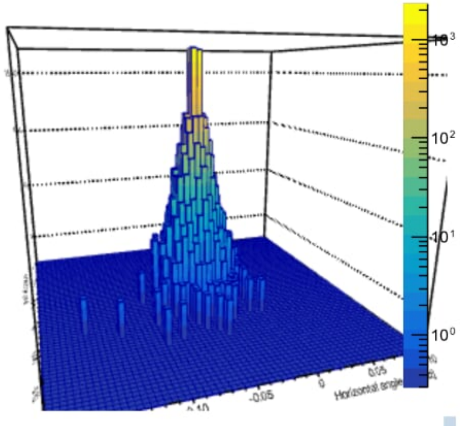
- take advantage of new C++ standards
- Python interface
- first production release by end of year
- part of Streaming Readout Grand Challenge at Jefferson Lab (C++ streamed events processing framework)

Dmitry Romanov (JLAB)

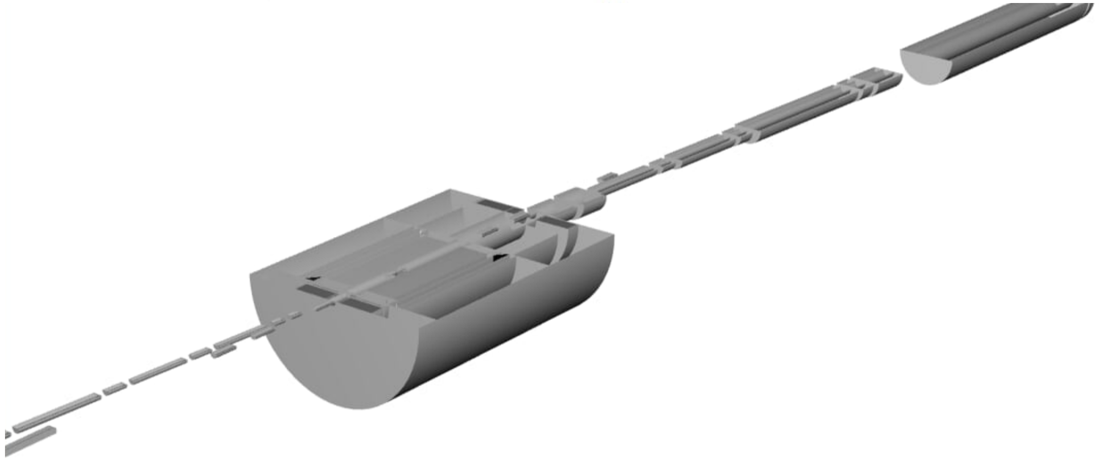




Neutrons angle distribution

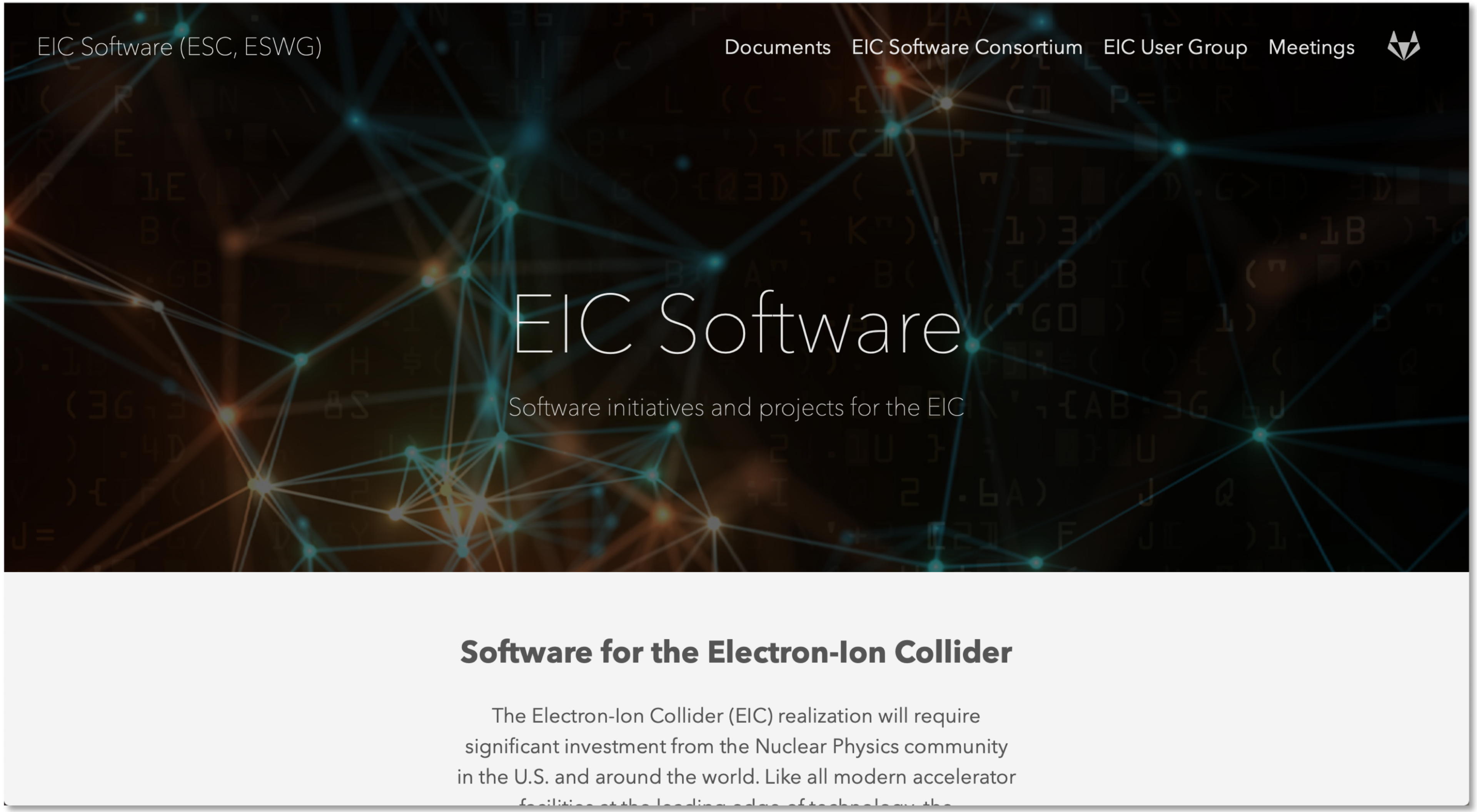


Neutrons vertical angle distribution



Jefferson Lab

Single point of entry



EIC Software website https://eic.gitlab.io

Single point of entry: Content

Color coding Available ● Not available

Introduction
- EIC Software Consortium
- EICUG Software Working Group

Documents
- overview (rates, requirements, visions)
- container guidelines (update)
- geometry exchange
- MCEG catalogue
- MCEG requirements

Meetings
- schedule including meeting links
- archive of 2015 – 2018 Meetings

References to
- repository
- documentation
- tutorials
- other EIC software resources

What else?

Single point of entry: Online catalogue for MCEGs

- **Goals** Hosted on https://eic.gitlab.io, editable for EIC group on GitLab
- **First steps** Agree on fields and then open call for input among EICUG

- **Proposed fields**
  - **Categories** ep, eA, radiative effects
  - Name
  - Contact information
  - **Brief Description** What processes are described? What is unique about the MCEG? Include version number as reference.
  - **References (links)** website, repository, documentation, container, validation plots

Example: Online catalogue for MCEGs

- **Category** ep, eA, exclusive vector meson production, general photoproduction
- **Name** eSTARlight
- **Contact Information** Spencer Klein, srklein@lbl.gov
- **Brief description** eSTARlight simulates coherent photoproduction and electroproduction of vector mesons in ep and eA collisions. It can simulate a variety of different vector mesons, and it also includes an interface to DPMJET, which allows for general simulation of photonuclear interactions. It internally simulates most simple (2-body) vector meson decays with a correct accounting for the initial photon polarization (transverse for Q^2 ~ 0, with an increasing longitudinal component with increasing Q^2) in the angular distributions of the final state. It can also interface to PYTHIA8 to simulate more complicated decays.
- **References** The code is freely available from https://estarlight.hepforge.org/ The Readme file includes a fairly comprehensive users manual. The physics behind the code is documented in M. Lomnitz and S. Klein, Phys. Rev. C99, 015203 (2019).

Meeting schedule

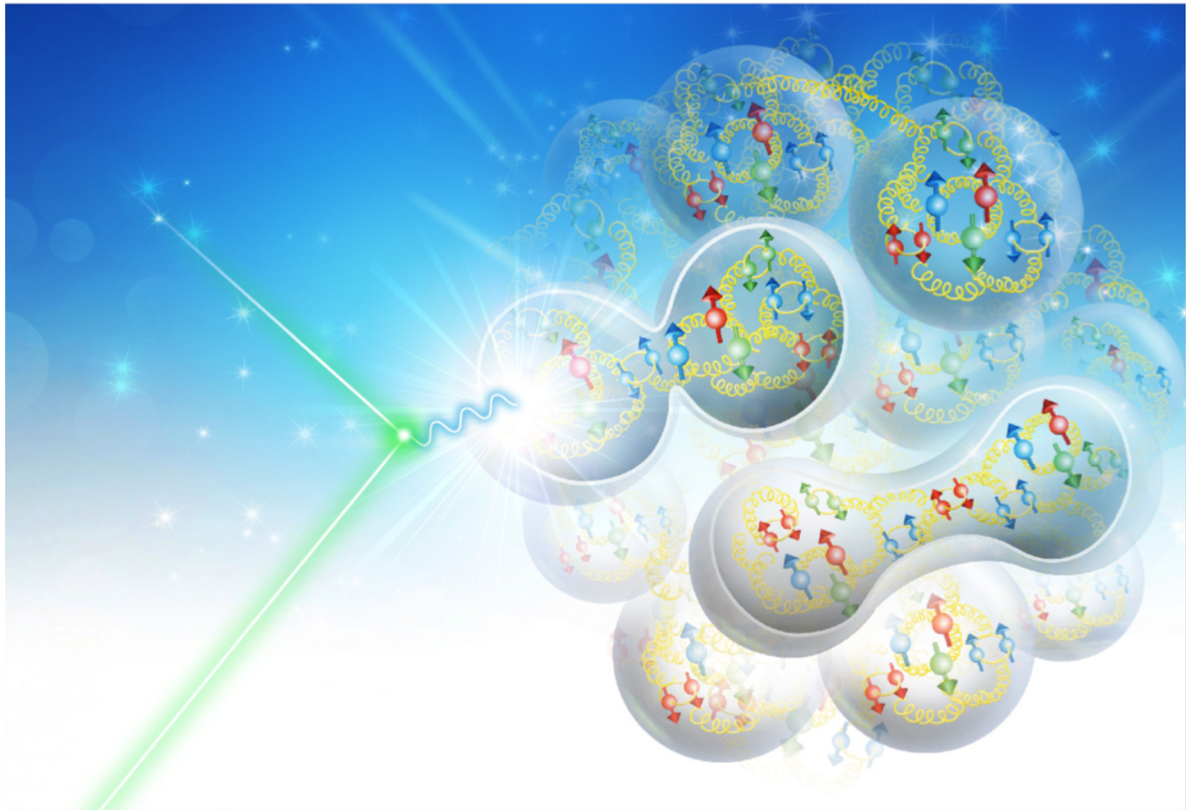
Date	Topic(s)
05/20 – 05/22	**EIC Software Meeting in Trieste, Italy**
06/13	Benchmarks and validation (only remote)
06/25	Tutorial during JLab Users Group Meeting (test run for EICUG Meeting in Paris, France)
07/10	*Full simulations and geometry exchange at BNL (tbc)*
07/11	Review of theory tools (only remote)
07/23	Tutorial during EICUG Meeting in Paris, France
08/	Summer break
09/05	FY20 planning (only remote)
09/24	Geant4 Technical Forum on EIC
	tba

Markus Diefenthaler
mdiefent@jlab.org

Summary

Electron-Ion Collider (EIC)

- precision study of the nucleon and the nucleus at the scale of sea quarks and gluons
- extremely broad science program



Computing vision for the EIC

- seamless data processing from DAQ to data analysis using artificial intelligence
- integration of DAQ, analysis and theory
- flexible, modular analysis ecosystem

Software Initiatives for the EIC

- EIC Software Consortium
- EICUG Software Working Group
- Website https://eic.gitlab.io