# eRD20: Developing Simulation and Analysis Tools for the EIC

EIC SOFTWARE CONSORTIUM

Jefferson Lab

Markus Diefenthaler

U.S. DEPARTMENT OF ENERGY | Office of Science

JSA

# Software Initiative for the EIC

## EIC Software Consortium

# 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

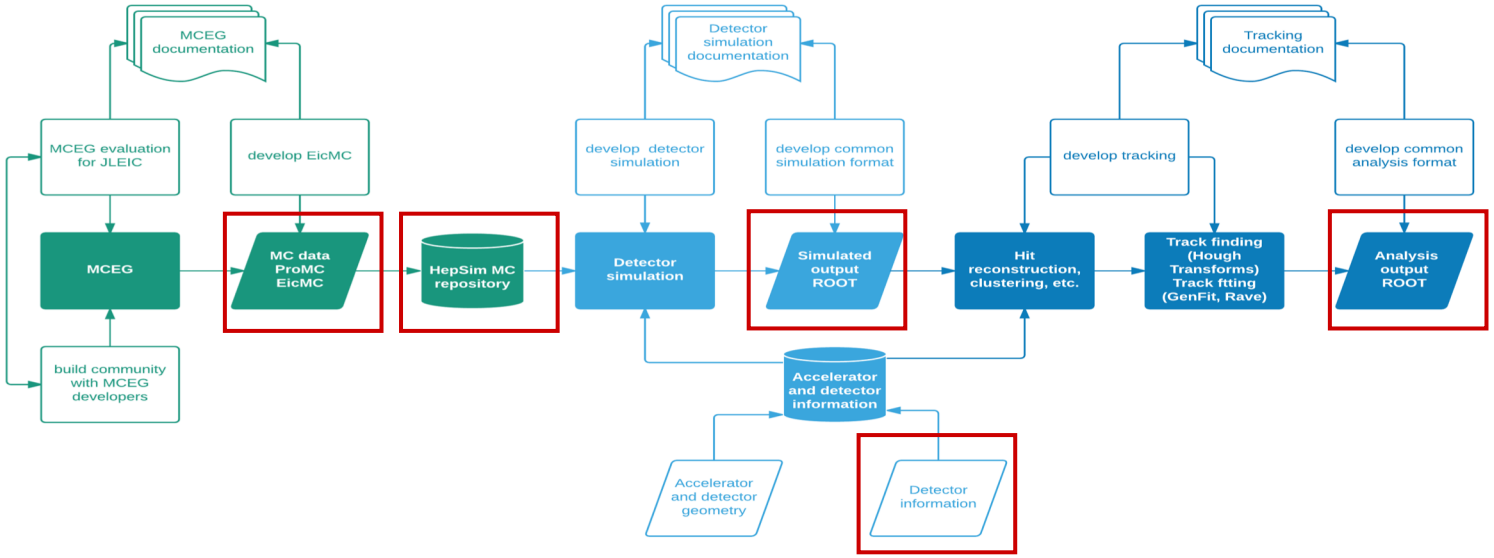
ANL, BNL, JLAB, LUND, INFN, SLAC, Trieste, W&M (→ Manitoba)

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 (eJANA)

TOPSiDE container

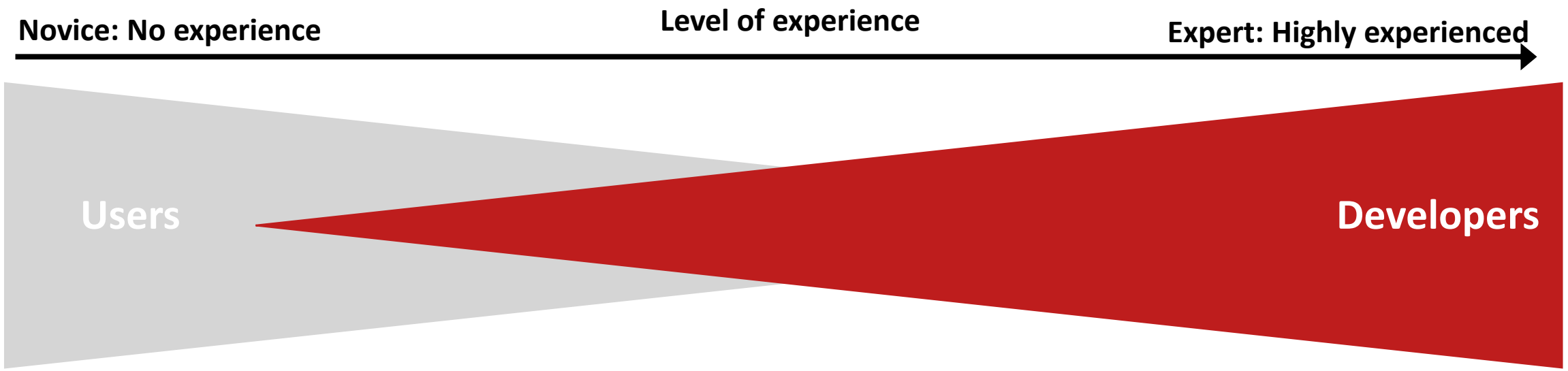
reconstruction

# ESC Initiatives

# Engaging with the wider community

## EICUG Software Working Group

# EIC Software Users




**User involved in**

**Ongoing EIC project**
Software ✓
Documentation ✓
**Requests** none

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

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

# User requests

**Ongoing EIC project**
Software ✓
Documentation ✓
**Requests** none

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

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

**Focus on** (pre)**CDRs and site selection as part of CD1**

**Focus on preparation of EIC collaborations**

- further develop EIC Science
- examine detector requirements
- work on detector designs
- work on detector concepts

requires simulations of physics processes and detector response

**Request from Thomas Ullrich, manager of the R&D program**:

- in most cases only GEANT simulations are needed:
  - no need for sophisticated framework
  - no need for elaborate tracking
- a simple *lite setup* with a well defined geometry description standard might get them a long way as long if it is EIC wide and easy to use

# Charge for EICUG Software Working Group

The EICUG Software Working Group's **initial focus** will be on **simulations of physics processes and detector response to enable quantitative assessment of measurement capabilities and their physics impact**. This will be pursued in a manner that is **accessible**, **consistent**, and **reproducible** to the EICUG as a whole.

It will embody simulations of all processes that make up the EIC science case as articulated in the white paper, eventually integrating new processes under request and with the help of interested communities within the EICUG. The Software working group is to engage with new major initiatives that aim to further develop the EIC science case, including for example the upcoming INT program(s), and is anticipated to play key roles also in the preparations for the EIC project(s) and its critical decisions. The working group will build on the considerable progress made within the EIC Software Consortium (eRD20) and other efforts. The evaluation or development of experiment-specific technologies, e.g. mass storage, clusters or other, are outside the initial scope of this working group until the actual experiment collaborations are formed.

The working group will be open to all members of the EICUG to work on EICUG related software tasks. It will communicate via a new mailing list and organize regular online and in-person meetings that enable broad and active participation from within the EICUG as a whole.

# EIC Generic Detector R&D projects

**Recommendation from January 2019**

*"The feedback from the user community indicates that there is a keen desire to have access to reliable and easy simulation and reconstruction frameworks. It seems that there remains a threshold for users to quickly engage in Monte Carlo simulations to carry out an end-to-end evaluation of various detector designs and study the EIC physics performance for different detector configurations.[...] The effort on providing common interfaces is strongly supported"*

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

## High Energy Physics

**CERN ROOT**
Possible collaboration

**HEP Software Foundation**
Started collaboration

**MCnet**
Started collaboration

**SLAC Geant4**
Established 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 ✗

## EIC Software initiative for community

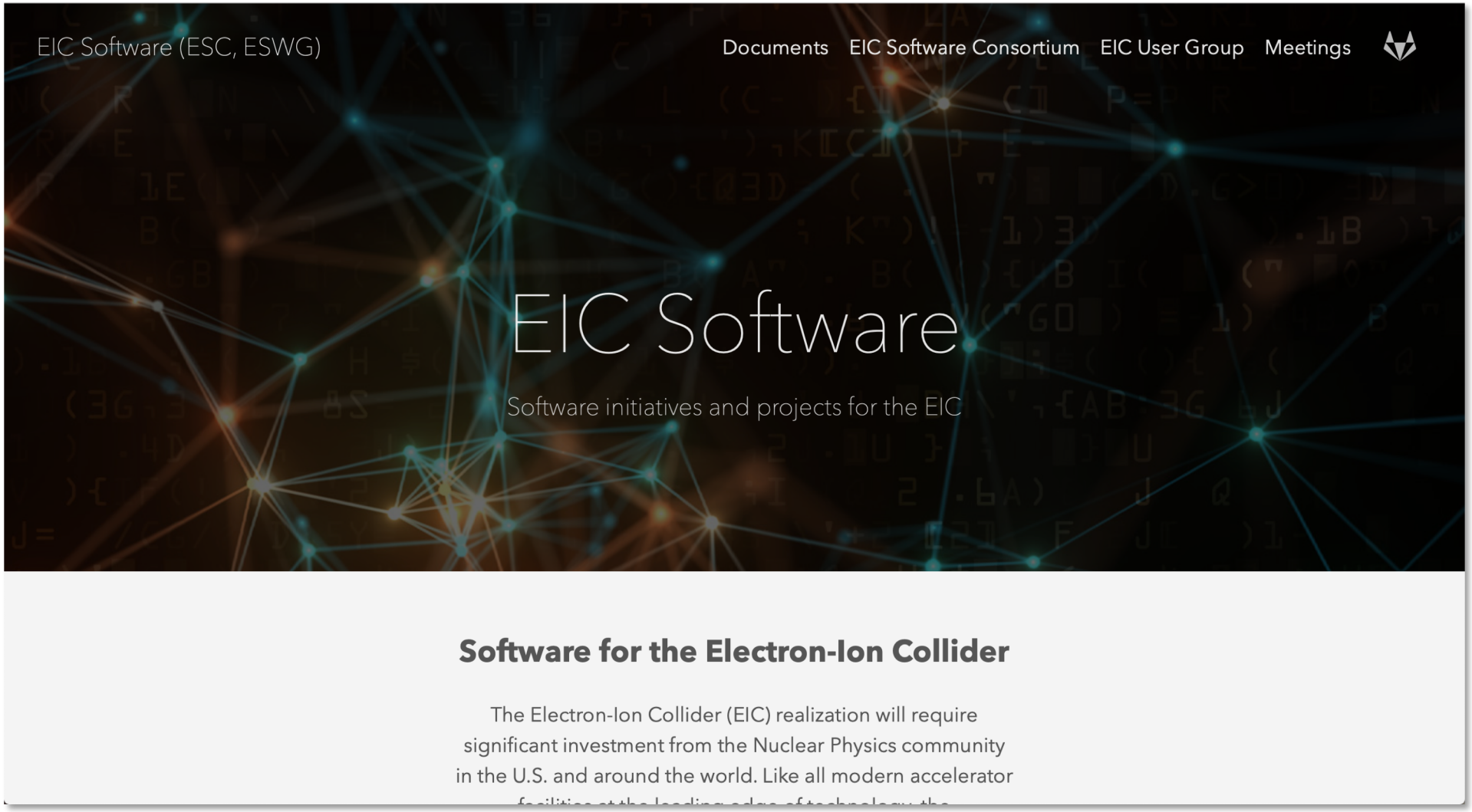
**Workflow environment for EICUG**

**to use** (tools, documentation, support) **and**

**to grow with user input** (direction, documentation, tools)

# EIC Software Meeting

# Single point of entry




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

# User interface for EIC simulations for EICUG

**ejana-gui container image**

**JupyterLab web interface**

**ejana-app container image**

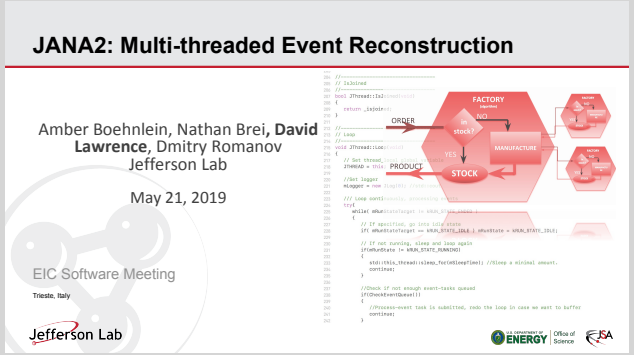
**eJana**

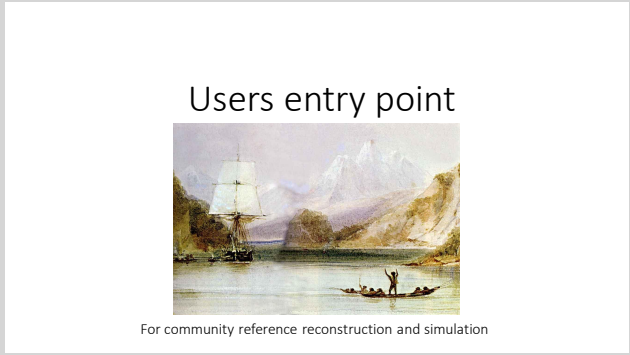
**Fast simulations (current focus)**
Monte Carlo event generators
**eic-smear** used for EIC Whitepaper

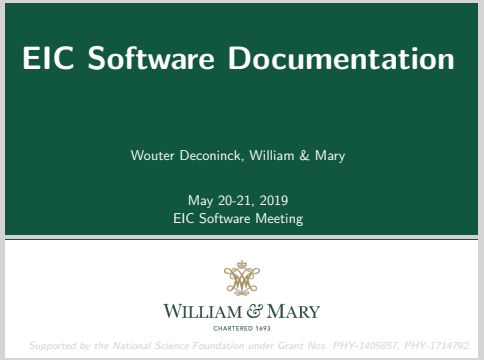
**TBD** Alterative for fast simulations
Monte Carlo event generators
Geant4 in fast mode

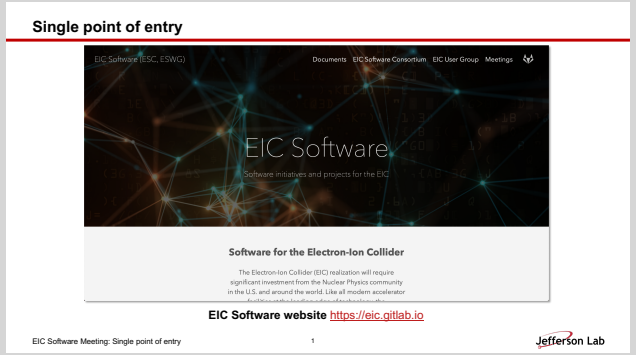
**Full simulations**
Monte Carlo event generators
Geant4 (to-be-unified)
Event reconstruction toolkit

## Detailed Status Reports



# Software design

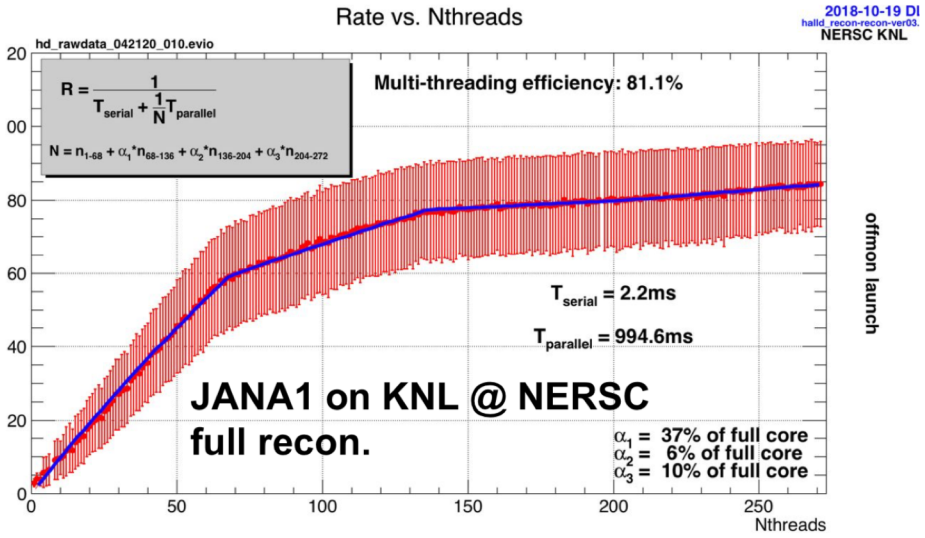
Dmitry Romanov (JLAB)

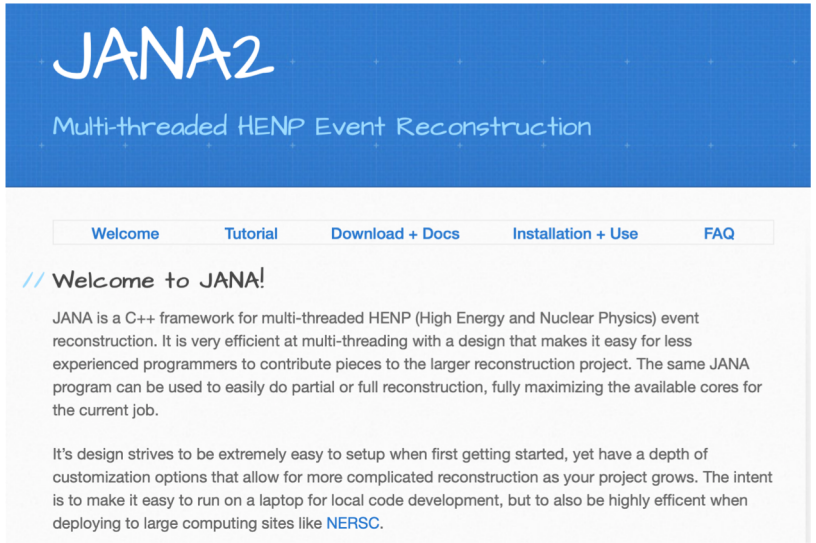
| | |
|---|---|
| **simple** | JupyterLab web interface |
| **moderate** | analysis scripts, python |
| **complex** | eJANA, plugins, C++ |
| **expert** | JANA, eic-smear, ROOT, Geant4 |

**Escaping complexity scaling trap**

- provide interfaces to internal layers
- interaction between layers must be clear

**Modularity** each layer must be replaceable




## 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
- part of Streaming Readout Grand Challenge at Jefferson Lab (C++ streamed events processing framework)

# $e^{JANA}$ - Community reference reconstruction

Dmitry Romanov (JLAB)

## $e^{JANA}$ - stands for EIC JANA

- Basic reconstruction
- Physics analysis
- **Users detector codebase integration**

Reconstruction

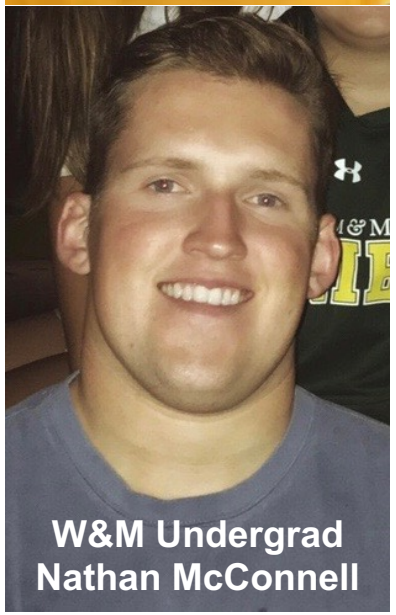
- **Tracking** - Genfit
- **Vertex finding** – Rave
- **Physical analysis**:
  - ROOT C++ or
  - Python data science tools (Jupyter, Seaborn, Pandas, etc.)

Any existing C++ (or even others) code can be:

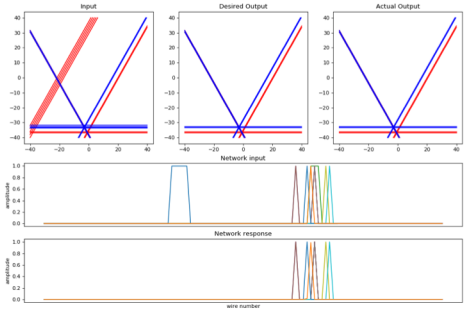
- compiled as JANA plugin
- run parallelized in eJANA
- accessed by other plugins

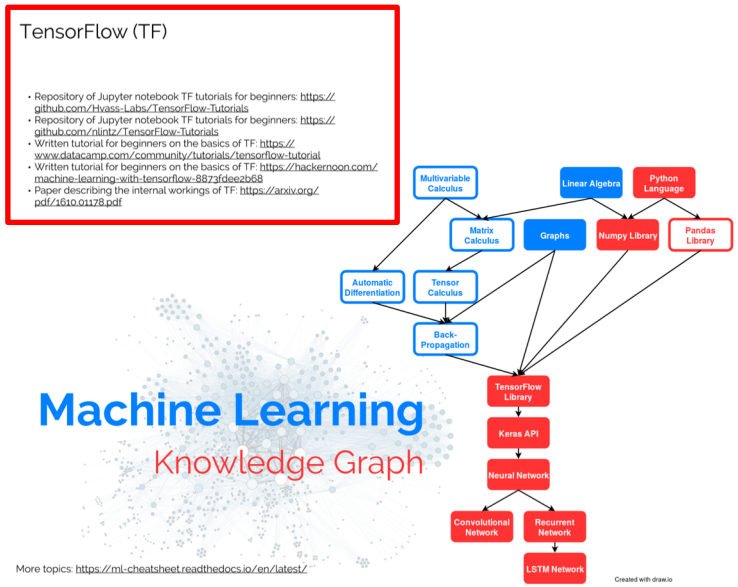
# ESC summer students

# Wouter Deconinck (W&M, Manitoba)



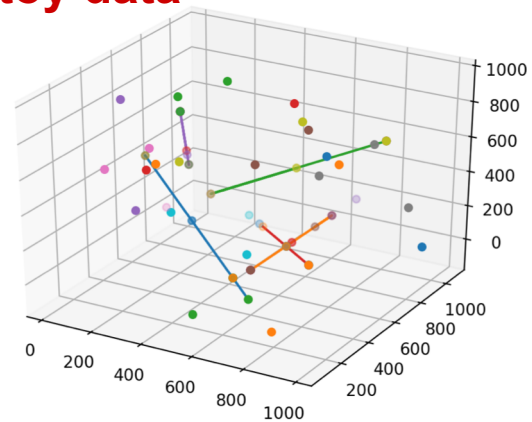

## Track finding in Jlab 12 data

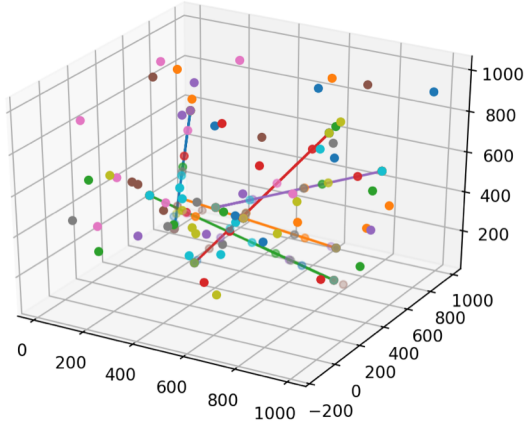


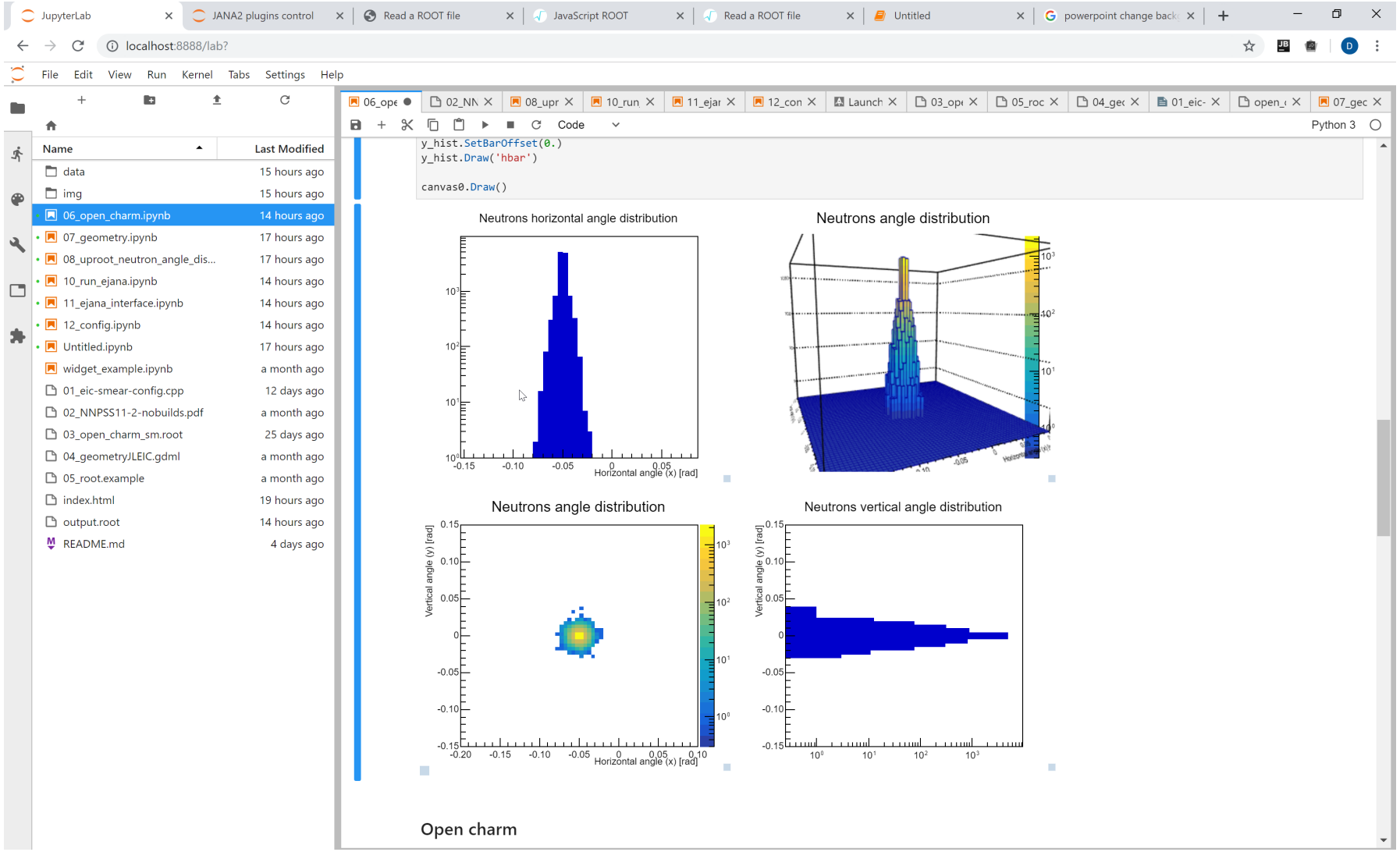

## Track finding in toy data

using JLab JupyterHub at https://jupyter.jlab.org/



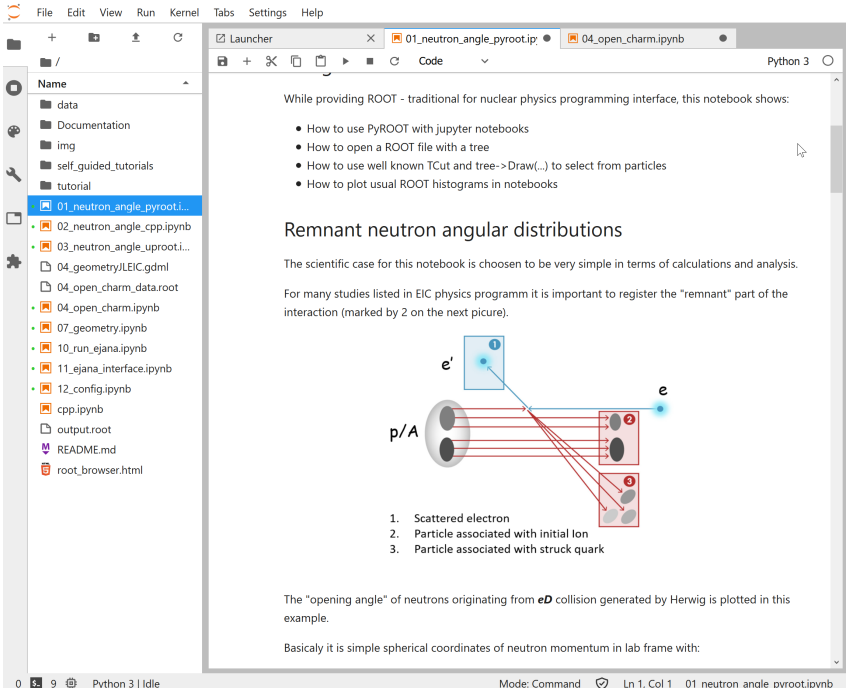
# Workflow oriented interactive environment based on JupyterLab

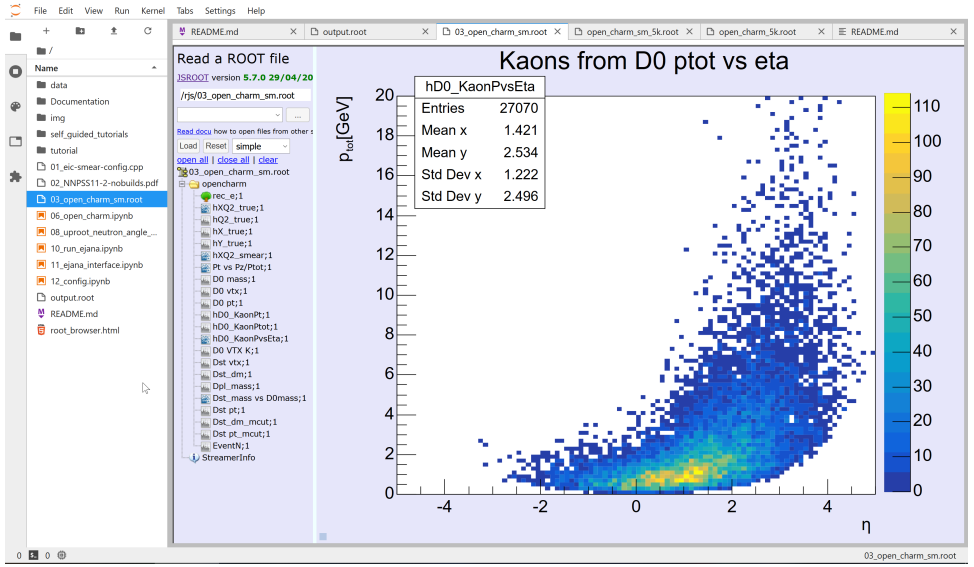


# Motivation for JupyterLab

## Project Jupyter

- bridge between modern Data Science and traditional Nuclear Physics methods
- self documenting

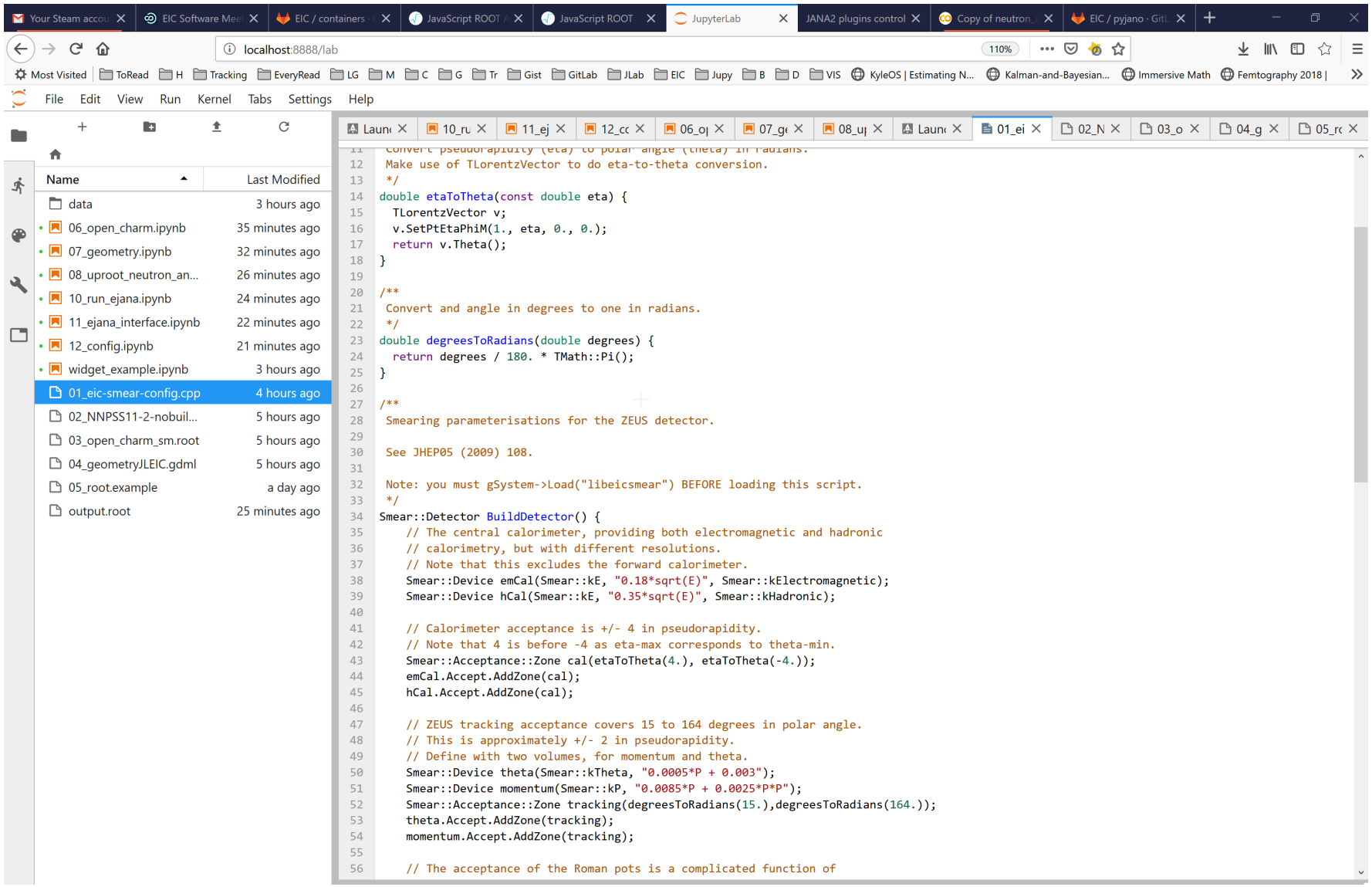


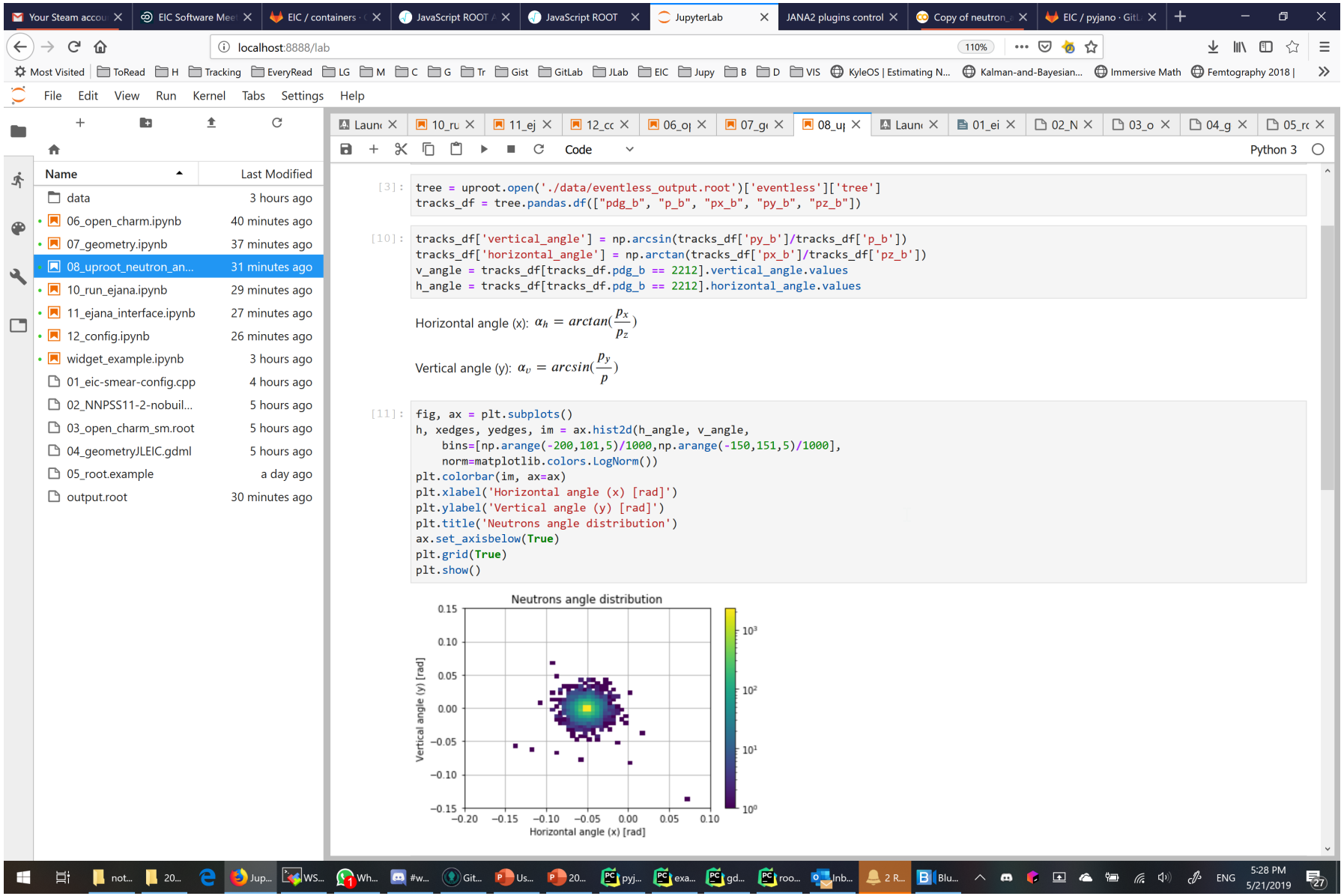

## Ready to be

- cloud-based collaborative workspace
- medium for analysis, studies, reports (helping with reproducibility problem)

# Complex layer: Interact with code



# Moderate layer: Jupyter notebooks



# Simple layer: Interactive vidgets

```
from pyjano import Jana
jana = Jana()
```

```
jana.plugins_gui()
```

**IO plugins:**

- lund_reader
- beagle_reader
- hepmc_reader
- jleic_geant_reader
- jleic_gemc_reader

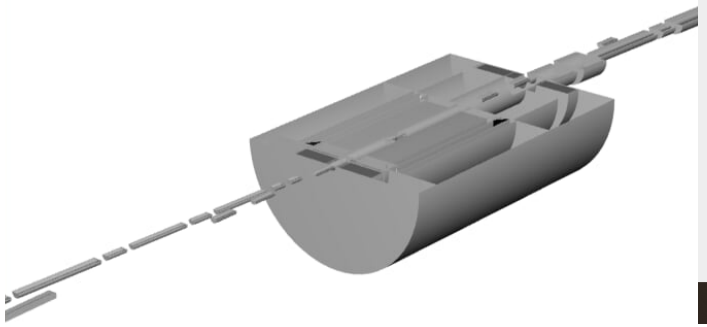
**Process & Analysis:**

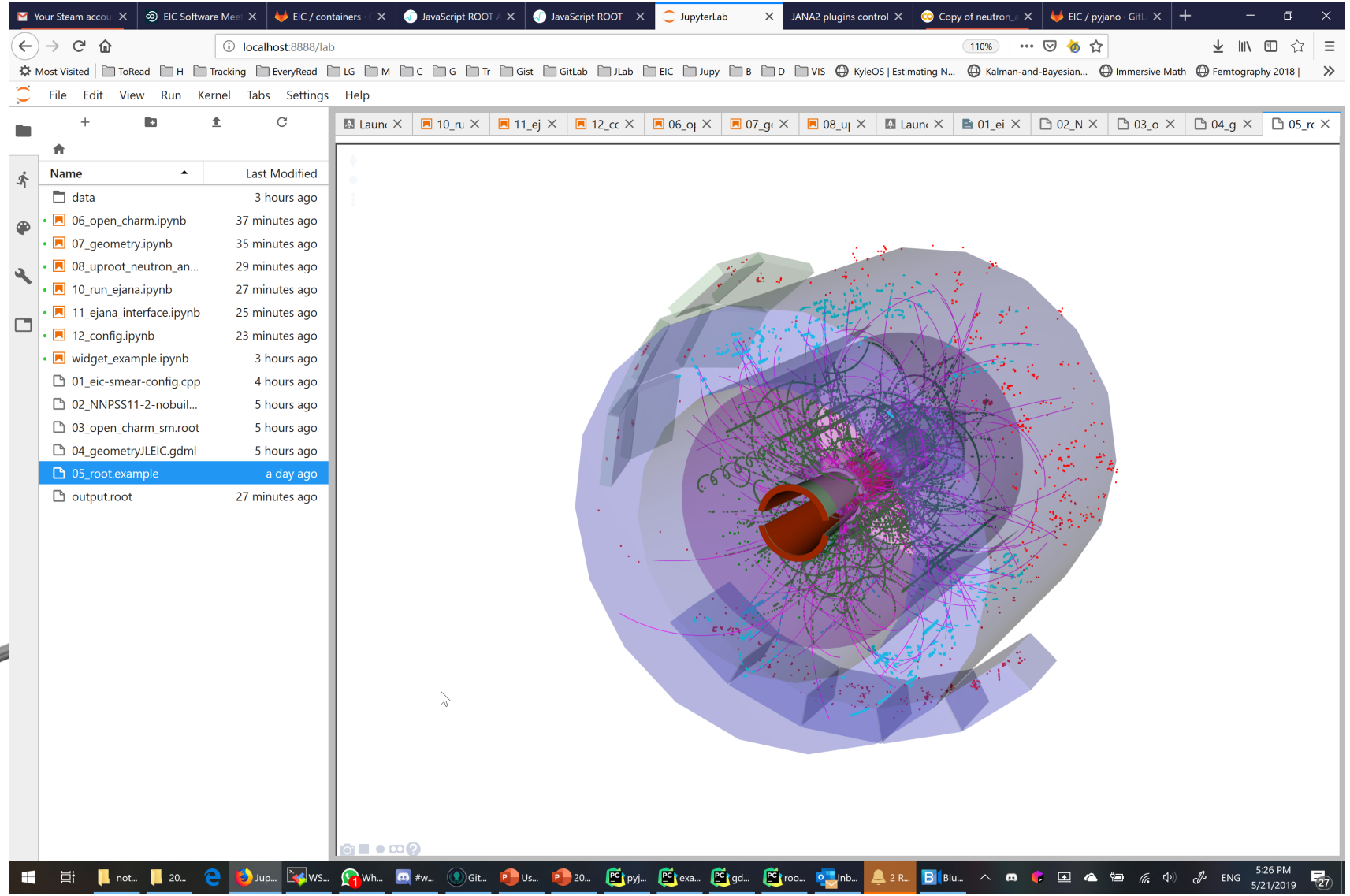
- trk_fit
- trk_eff
- jleic_iff
- jleic_occupancy
- vmeson
- open_charm

2 verbose (int)

Plugin **beagle_reader:** Opens files from BeAGLE event generator as a data source
**BeAGLE** - **Be**nchmark **eA** **G**enerator for **LE**ptoproduction
Documentation

# Eve7 root7 event display





# EIC Software Meeting on Detector and Physics Simulations

## EIC Software Meeting on Detector and Physics Simulations

Wednesday 10 Jul 2019, 09:00 → 15:00 US/Eastern

2-160 (BNL Physics)

Andrea Bressan (Trieste), Markus Diefenthaler (Jefferson Lab), Torre Wenaus (BNL)

**35 participants**

**Description**

The Electron-Ion Collider User Group (EICUG) is organizing an EIC Ad-hoc Meeting on detector and physics simulations:

We will work out the requirements for simulation tools including geometry exchange between the eRHIC and JLEIC concepts and plan next steps for common tools for the EICUG. The work will be also relevant for TOPSiDE and other detector concepts for the EIC.

The goal of the meeting is to clarify the strategy for EIC detector and physics simulations and to meet on the time scale of FY19/FY20 the requirements for common tools and documentation in the EICUG. We would like to include all EIC Software initiatives in our discussion.

The meeting will be held at BNL on July 10, prior to the EIC Generic Detector R&D meeting on July 11-12, and will include presentations from HEP and NP experts as well as discussions with EIC Software developers.

The meeting has a hard finish at 3pm (we lose the room).

If you wish to attend in person and do not have a current BNL badge, please contact Torre Wenaus (wenaus@gmail.com, 631-681-7892) as soon as possible. Also contact Torre with any BNL related issues in general.

BlueJeans: https://bluejeans.com/920347364

# Plans and actions

- agree on detector naming conventions
- agree on G4 C++ as the basis of detector description for now (access to all G4 volumes including new recommended ones like tessellated solids)
  - 3 of the 4 approaches (not EicRoot) can work with C++ G4 detector descriptions
  - make the (inevitable) framework-specific adaptation layer needed around the C++ as thin as possible
- agree on how structs for parameters are defined and managed
  - ‘numbers in the code’ is not necessarily a bad thing if they are managed carefully. Coming to commonality on a more long term solution like DB is a bigger issue (and not on Thomas’ ‘now’ list)
- agree on API/class design for sensitive detector stepping action/digi
- agree on hits output structure
  - hits structure - can we agree on that? depends on detector and how reco is handled afterwards
  - even without this we’ve agreed on enough to satisfy Thomas
- common repository we point people to when they come along with a detector model to plug in
- deliverable: the simple document and template(s) we give to the user on how to implement and integrate their new subdetector in the detector concepts via our common Geant4 infrastructure
  - time frame?
    - by end of the fiscal year (end Sep), have made the agreements concrete in a document
    - for implementation…?
- deliverable: an updated/agreed document of the requirements
- need follow-up meeting on participation, who will do what by when
  - what resources do we have and how do we use them

# Meeting schedule

| Date | Topic(s) |
|---|---|
| **05/20 – 05/22** | **EIC Software Meeting in Trieste, Italy** |
| **06/27** | Benchmarks and validation (only remote) |
| **07/10** | EIC Ad-hoc Meeting on detector and physics simulations |
| **07/23** | Tutorial during EICUG Meeting in Paris, France |
| **08/** | Summer break |
| **09/24** | Geant4 Technical Forum on EIC |
| | **tba** |

# Geant4 Technical Forum at Jefferson Lab

Makoto Asai (SLAC)

- Geant4 Technical Forum will be held at Jefferson Lab on Tuesday, September 24 2019, as a part of the Geant4 Collaboration Meeting.
  - 16:00-18:00 US Eastern time (22:00-24:00 Central European time)
- This Technical Forum is highlighting EIC:
  - Geant4 developers will update the EIC community of the recent and ongoing developments that are relevant for EIC,
  - Requirements and concerns from the EIC community should be presented,
  - and discussed.
- Please keep Makoto and me informed on
  - What you want to hear from Geant4 developers, and
  - What you want to tell / argue to Geant4 developers

Markus Diefenthaler

mdiefent@jlab.org

# FY20 budget request

**Funds for hiring a postdoctoral researcher of USD 120,000**

- works solely on the community tools for physics and detector simulations including geometry exchange between eRHIC and JLEIC and BeAST, ePHENIX, JLEIC, TOPSiDE detector concepts
- service for the EIC and in particular Detector R&D communities: support for using the tools and for integrating existing detector designs and reconstruction algorithms into the community tools

**Travel funds of USD 20,000**

- in-person meetings to work together on key tasks
- invite scientists that are essential to the R&D effort

**Budget scenarios**

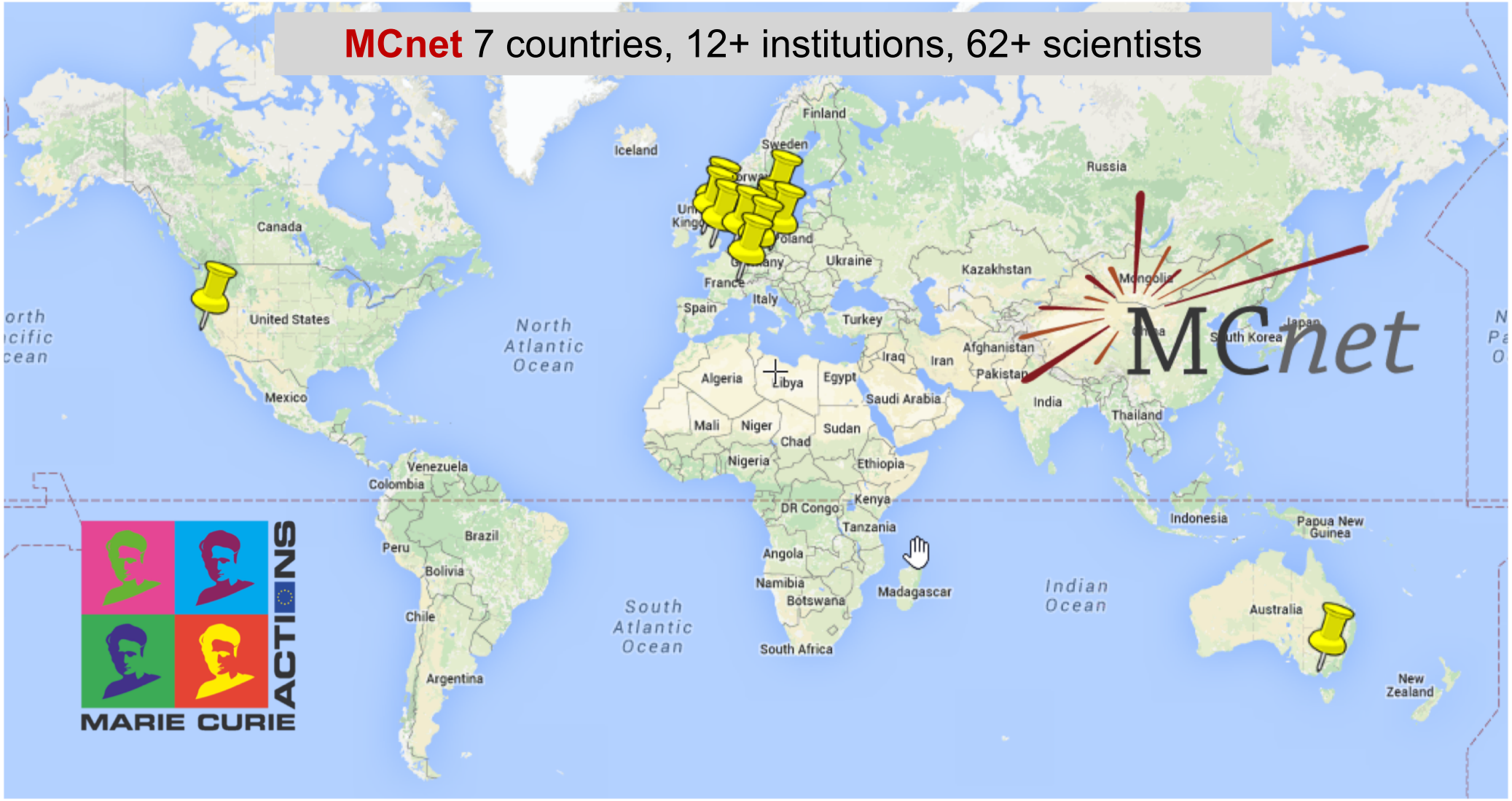
- Nominal budget (100%): USD 140,000
- Nominal budget (-80%): USD 20,000 (only travel)

EIC SOFTWARE CONSORTIUM

# Additional slides

## MCEG Initiative

# MCEG Developers



# Workshops: MCEGs for future ep and eA facilities

**MCEG2018** 19–23 March 2018

- Started as satellite workshop during POETIC-8

POETIC 8
8th International Conference on Physics Opportunities at an ElecTron-Ion-Collider
19-23 March 2018, University of Regensburg

- Collaboration EICUG-MCnet

**Goal of workshop series**

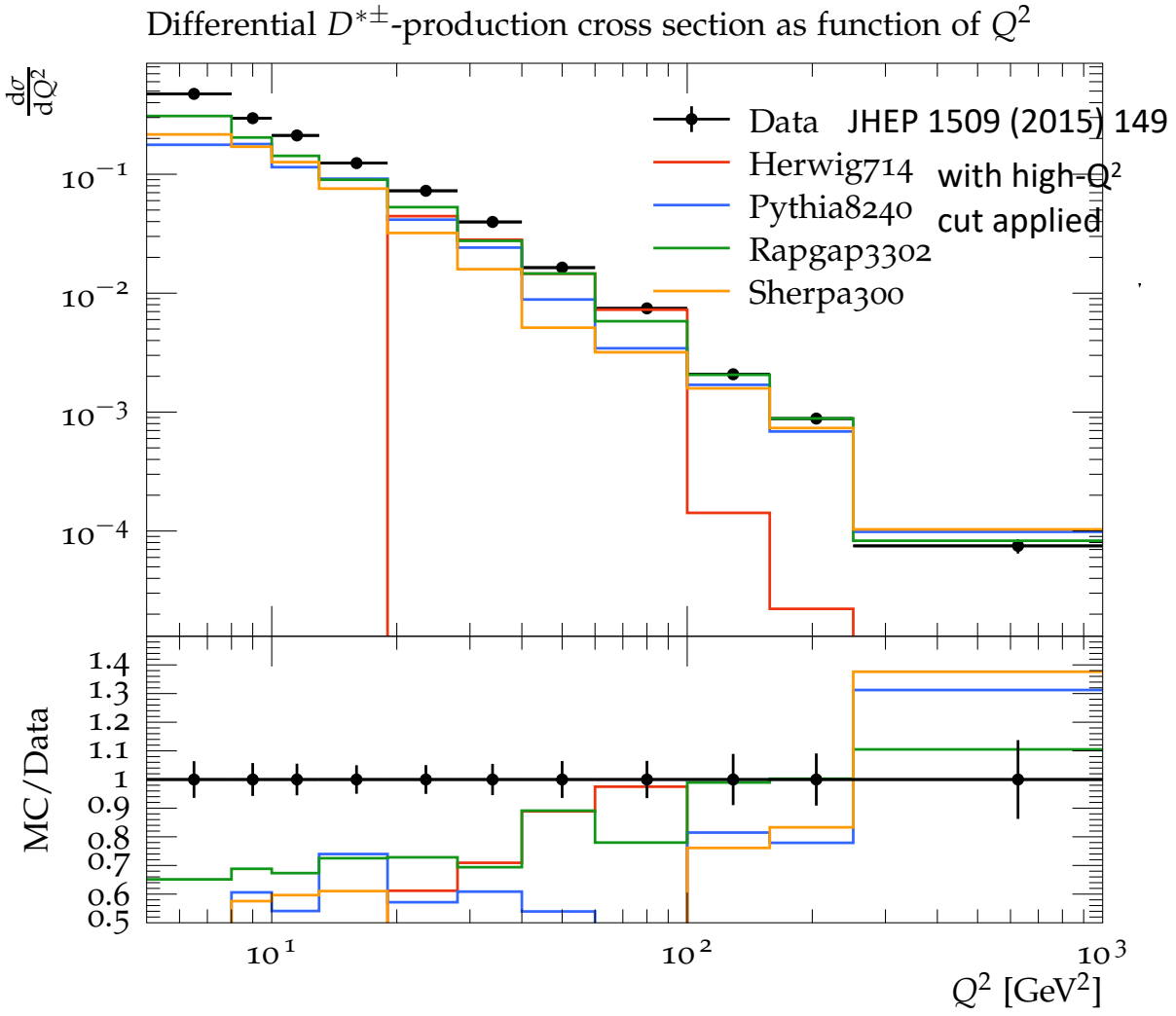
- Requirements for MCEGs for ep and eA
- R&D for MCEGs for ep and eA

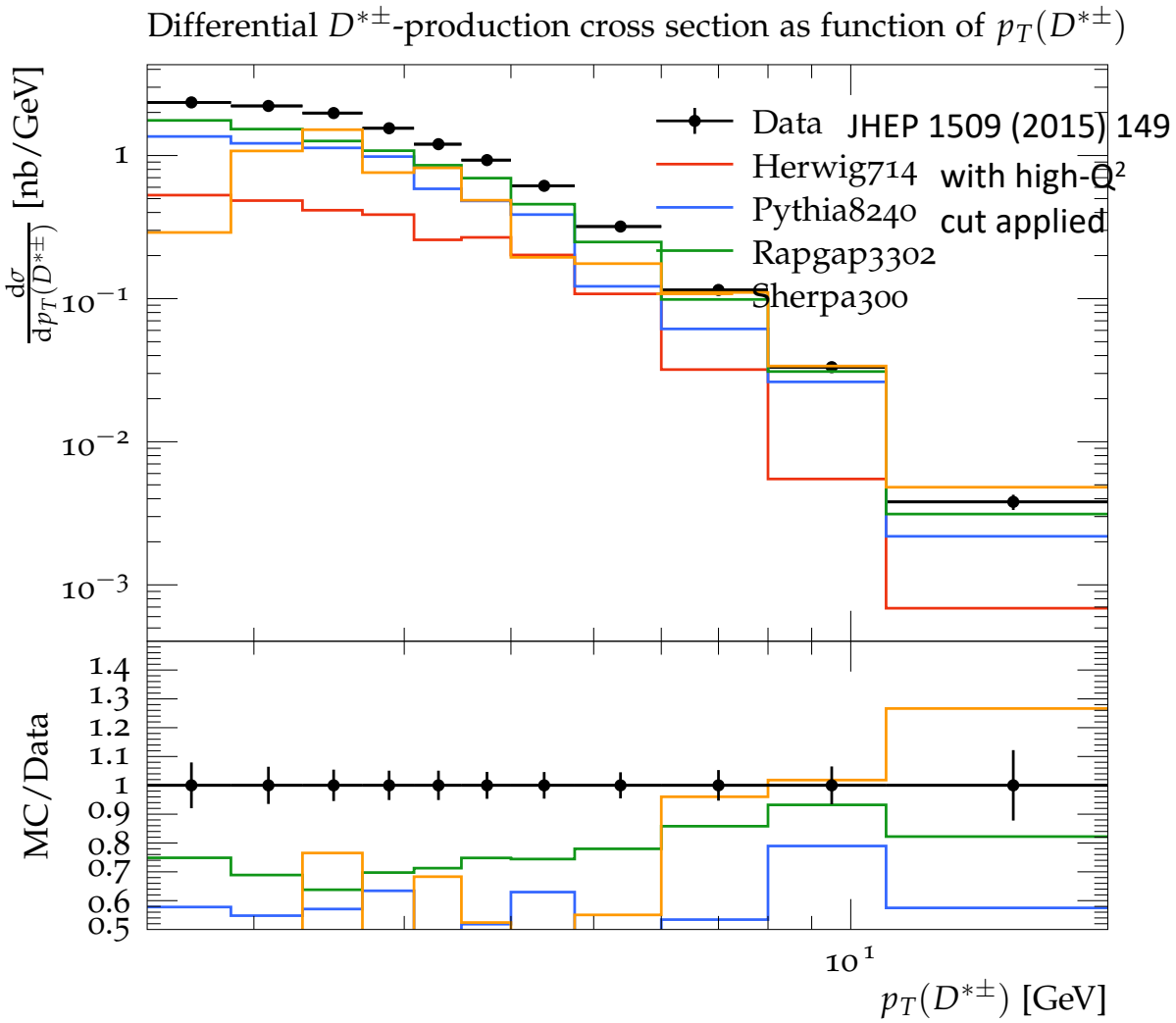
**MCEG2019** 20–22 February 2019

- Status of ep and eA in general-purpose MCEG
- Status of NLO simulations for ep
- TMDs and GPDs and MCEGs
- Merging QED and QCD effects

# Comparisons to combined H1 and ZEUS analysis

A. Verbytskyi (MPI Munich)



# Getting started with Pythia8

JLAB LDRD

## 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 MULTHIST
import py8settings as py8s
```

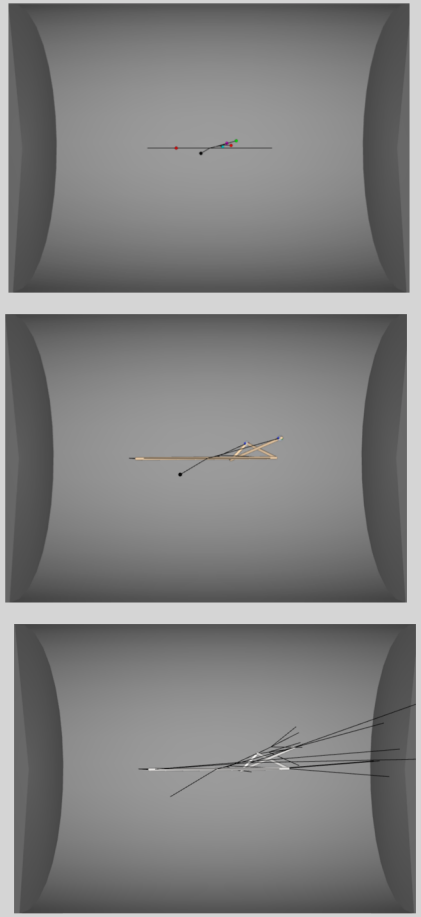
Now we create a Pythia 8 object and apply the settings to define the incoming beams. More settings can be adjusted later.

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



# MCEG–HERA comparisons and MCEG validation for ep

## Rivet example

SIDIS analysis at HERMES

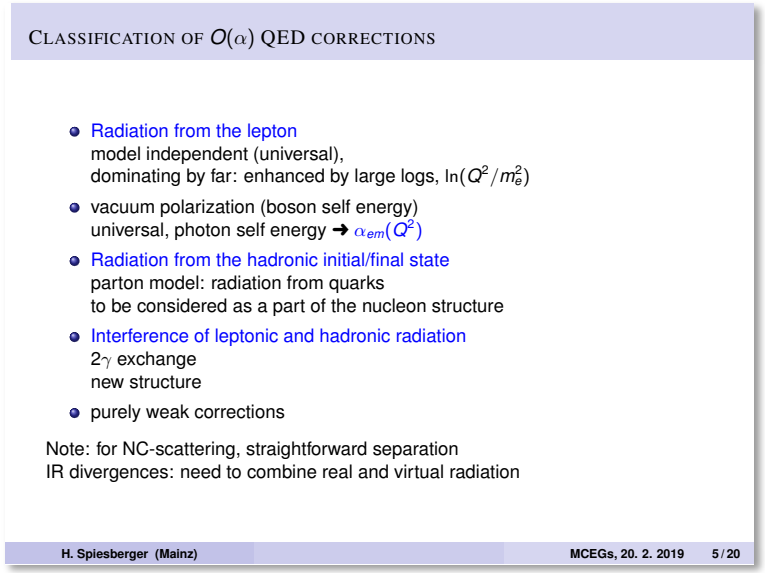
```
// Extract the particles other than the lepton
const FinalState& fs = apply<FinalState>(event, "FS");
Particles particles;
particles.reserve(fs.particles().size());
const GenParticle* dislepGP = dl.out().genParticle();
foreach (const Particle& p, fs.particles()) {
  const GenParticle* loopGP = p.genParticle();
  if (loopGP == dislepGP)
    continue;
  particles.push_back(p);
}

// Apply HERMES cuts.
bool validx = (x > 0.023 && x < 0.6);
if (q2 < 1. || w2 < 10. || y < 0.1 || y > 0.85 || !validx)
  vetoEvent;

// good inclusive event, let's do bookkeeping before we look at the hadrons
dis_tot += weight;
dis_x->fill(x, weight);
dis_Q2->fill(q2, weight);

for (size_t ip1 = 0; ip1 < particles.size(); ++ip1) {
  const Particle& p = particles[ip1];

  // get the particle index, check if it is a particle of interest
  const int part_idx = get_index(p.genParticle()->pdg_id());
  if (part_idx < 0) {
    continue;
  }

  // we have a particle of interest, let's calculate the kinematics
  // z
  const double z = (p.momentum() * pProton) / (pProton * q);
  // pt
  const double pth = sqrt(p.momentum().pT2());

  // get our z index, if negative, we have a particle outside of [.2, .8]
  const int z_idx = calc_zslice(z);
  if (z_idx < 0) {
    continue;
  }

  // store the events and make cuts where necessary
  //
  // pt cut for variables not binned in pt
  if (pth > 0 && pth < 1.2) {
    mult_z[part_idx]->fill(z, weight);
    mult_zx[part_idx][z_idx]->fill(x, weight);
    mult_zQ2[part_idx][z_idx]->fill(q2, weight);
  }
  mult_zpt[part_idx][z_idx]->fill(pth, weight);
}
```

- **MCEG R&D** requires *easy* access to *data*
- data := analysis description + data points

- **HEP** existing workflow for MCEG R&D using tools such as Rivet and Professor

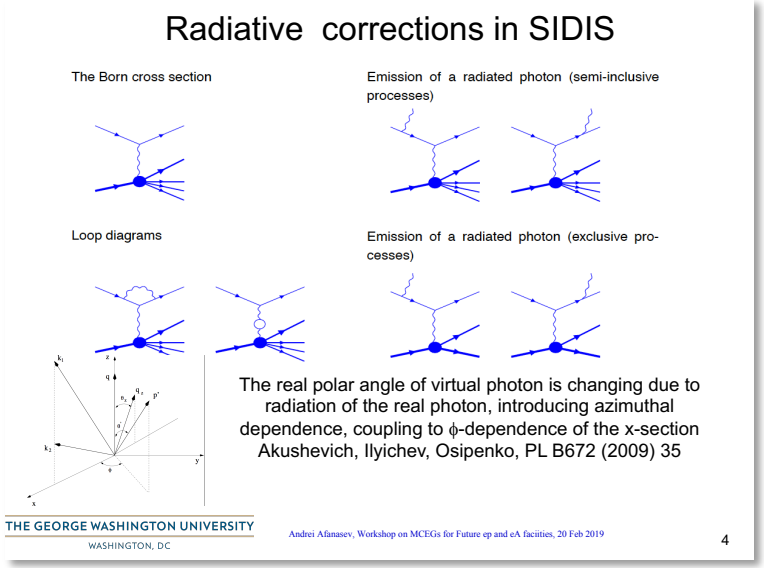
- **Detailed comparisons between modern MCEG and HERA data**
  - workshop on **Rivet for ep** (Feb 18—20 2019)
  - mailing list **rivet-ep-l@lists.bnl.gov**
  - HERA data not (yet) included in MCEG tunes

# Merging QED and QCD effects




**Hubert Spiesberger (Mainz): QED corrections for electron scattering**

- High-precision measurements need careful treatment of radiative corrections.
- Closely related to experimental conditions need full Monte Carlo treatment (Unfolding) including simulation of hadronic final states.
- The basics are known and available …
- … but improvements are needed.



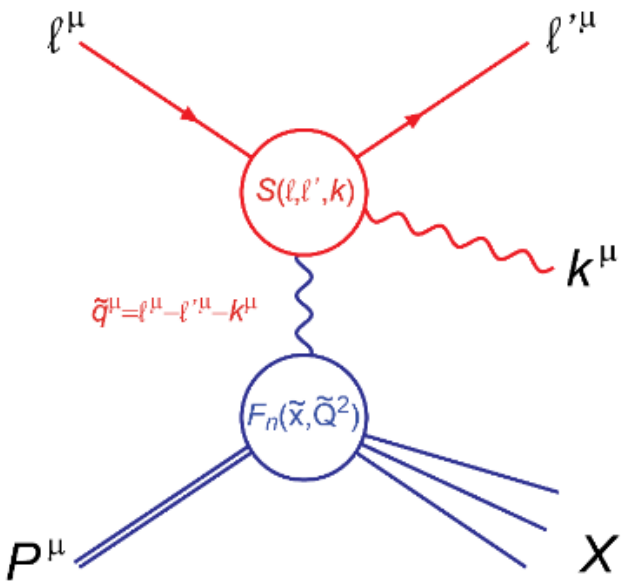

**Andrei Afanasev (GWU): Semi-analytic vs. Monte-Carlo Approaches for QED Corrections to SIDIS**

- Consistent approach to address RC for SSA in polarized SIDIS
- SSA due to two-photon exchange need to be included in analysis of SSA from strong interaction, of same size at JLAB experiments
- More detailed calculation of the two-photon exchange at quark level required: elastic scattering, inclusive, semi-inclusive, and exclusive DIS

# EIC Software Consortium: Radiative Effects and MCEG

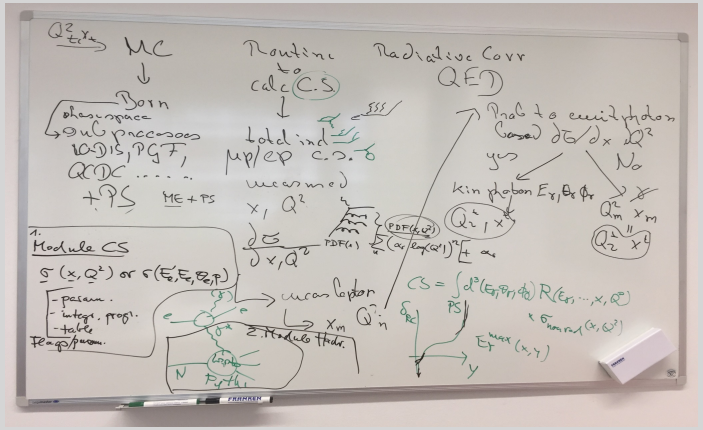
## Radiative effects

- change kinematics on an event by event basis:
  - smearing of kinematic distributions
- change of virtual-photon direction:
  - false asymmetries in the azimuthal distribution of hadrons
- correction:
  - unfolding procedure, requires MCEG including radiative corrections / effects




## ESC Radiative effects library

- Elke-Caroline Aschenauer, Andrea Bressan
- essential for high-precision measurements at the EIC
- collaboration with Hubert Spiesberger:
  - start back from HERACLES part of Djangoh
  - work on interface to PYTHIA6/8



# MCEG2019: Next steps

- **General-purpose MCEG**s, HERWIG, PYTHIA, and SHERPA, will be significantly improved w.r.t. MCEGs at HERA time:
  - MCEG-data comparisons in Rivet will be critical to tune the MCEGs to DIS data and theory predictions.
  - The existing general-purpose MCEG should soon be able to simulate NC and CC unpolarized observables also for eA. A precise treatment of the nucleus and its breakup is needed.
  - First parton showers and hadronization models for ep with spin effects, but far more work needed for polarized ep / eA simulations.
  - Need to clarify the details about merging QED+QCD effects (in particular for eA).

- **TMD physics**
  - Vibrant community working on various computational tools for TMDs.
  - CASCADE: MCEG for unpolarized TMDs at high energy.
  - Need more verification of MCEG models with TMD theory / phenomenology.

- **GPD physics**
  - No modern MCEG available.
  - There is a path from PARTONS to a GPD MCEG, similar there is a project to extend MCEG for exclusive processes from JLAB12 to EIC.