Software Effort for JLEIC

Markus Diefenthaler (mdiefent@jlab.org)

presenting

Jefferson Lab EIC software effort

EIC Software Consortium (ESC)

Overview

Lessons learned

What do we need to improve?

Urgent requirements

- develop EIC science case
- design the EIC detectors

Vision for EIC Computing


PAST

Lessons learned

Anecdote from HEP typical LHC student or post-doc spends up to 50% of her/his time dealing with computing issues

similar in NP

User centered design

- understand the user requirements first and foremost
- engage wider community of physicists in design whose primary interest is not computing
- make design decisions solely based on user requirements


PRESENT
DUB POLICE

Discussion of requirements

Use case 1

Requirements for studying physics processes at EIC:

- interface to MCEG
- open access to accelerator specifications
- open access to detector information and simulation
- documentation

Use case 2

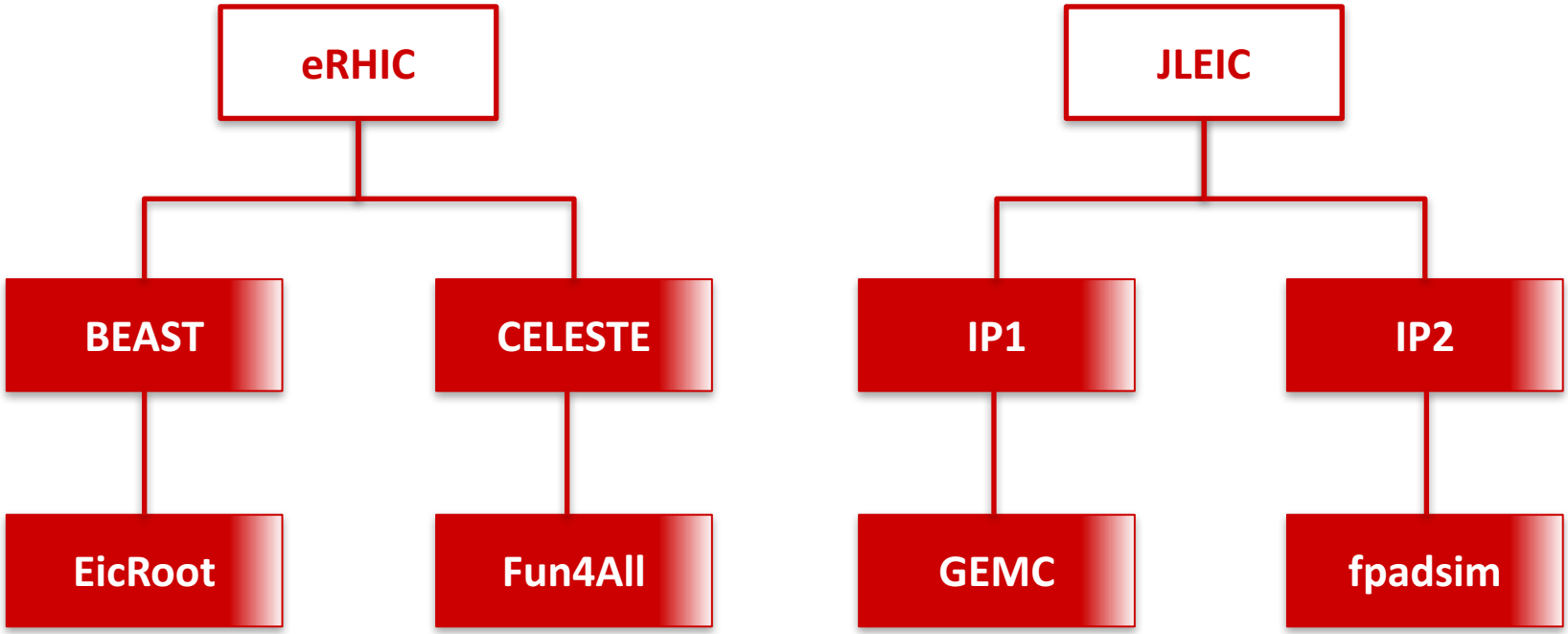
Requirements for studying detectors at EIC:

- open access to physics simulations or interface to MCEG
- open access to accelerator specifications
- open access to detector information and simulation
- documentation

Use cases 1 and 2 might involve comparison of eRHIC and JLEIC:

- eRHIC setup might be used in JLEIC software
- JLEIC setup might be used in eRHIC software

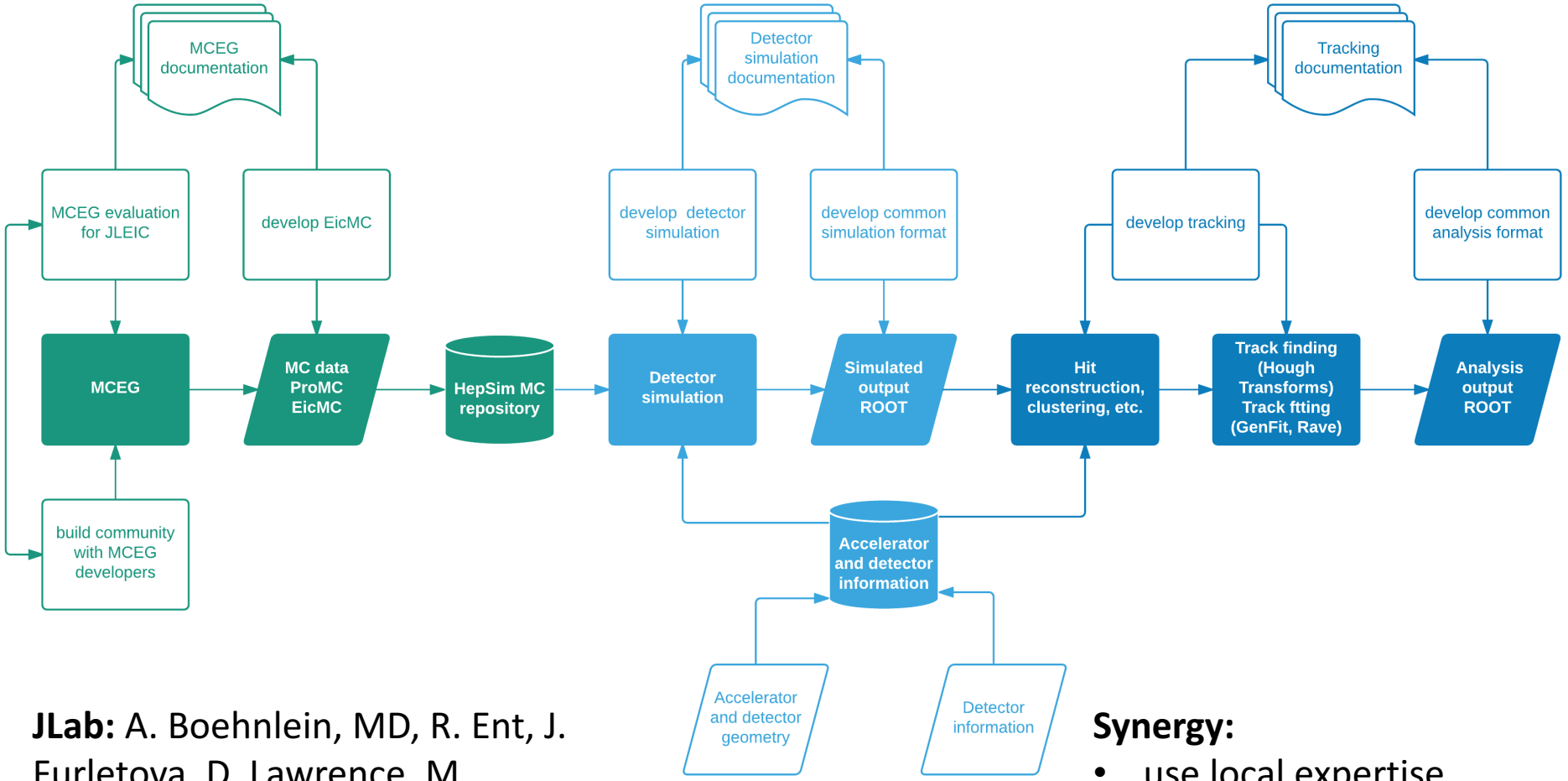
Existing software frameworks for the EIC



ESC Review

- all actively maintained
- examine the requirements for the EIC analysis environment
- work on the R&D aspects of the EIC analysis environment
- **initial focus on fast simulations**

JLEIC software planning



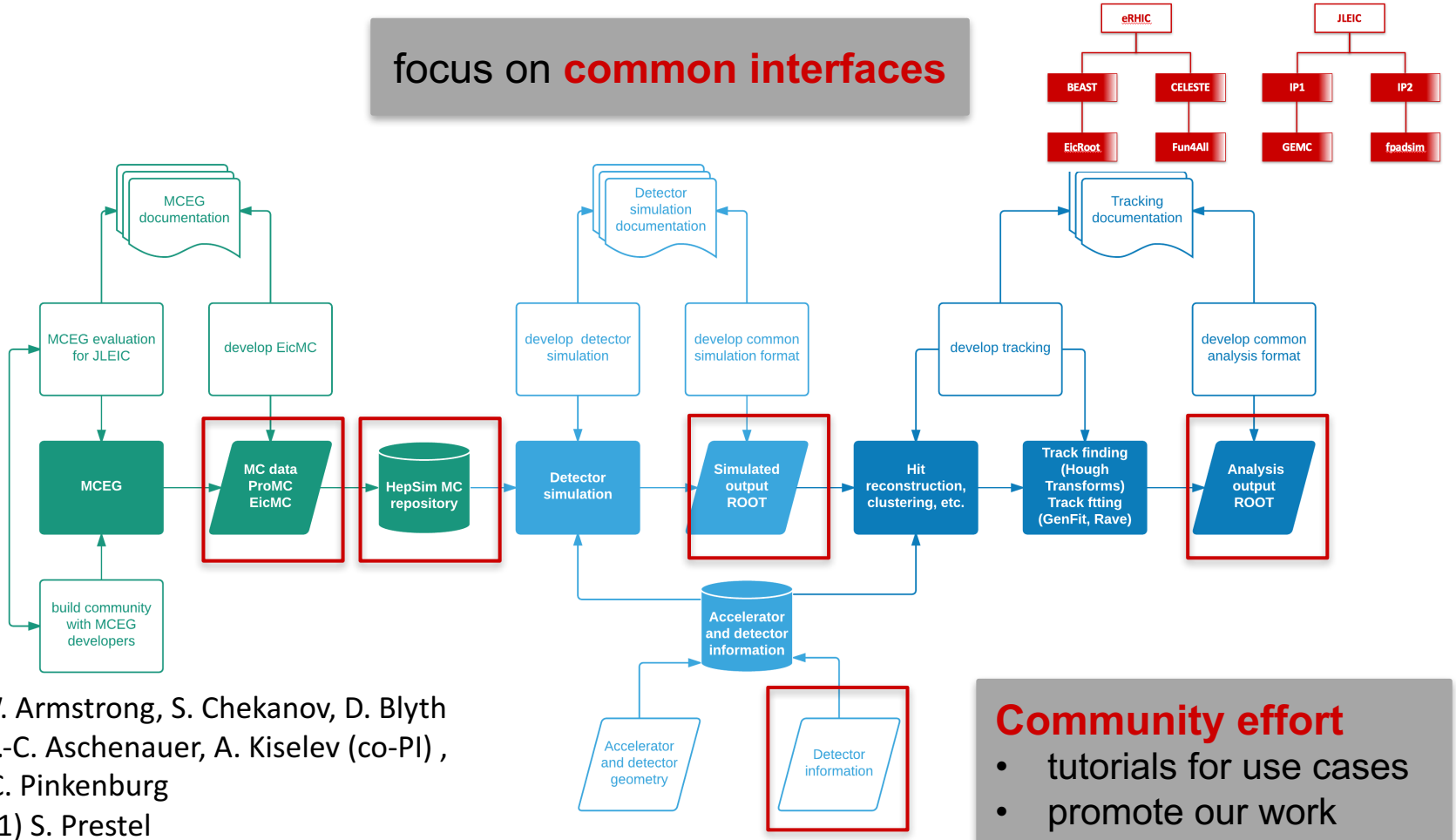
JLab: A. Boehnlein, MD, R. Ent, J. Furletova, D. Lawrence, M. Ungaro, R. Yoshida

Synergy:

- use local expertise
- work closely with Computing Center

ESC contributions

focus on **common interfaces**

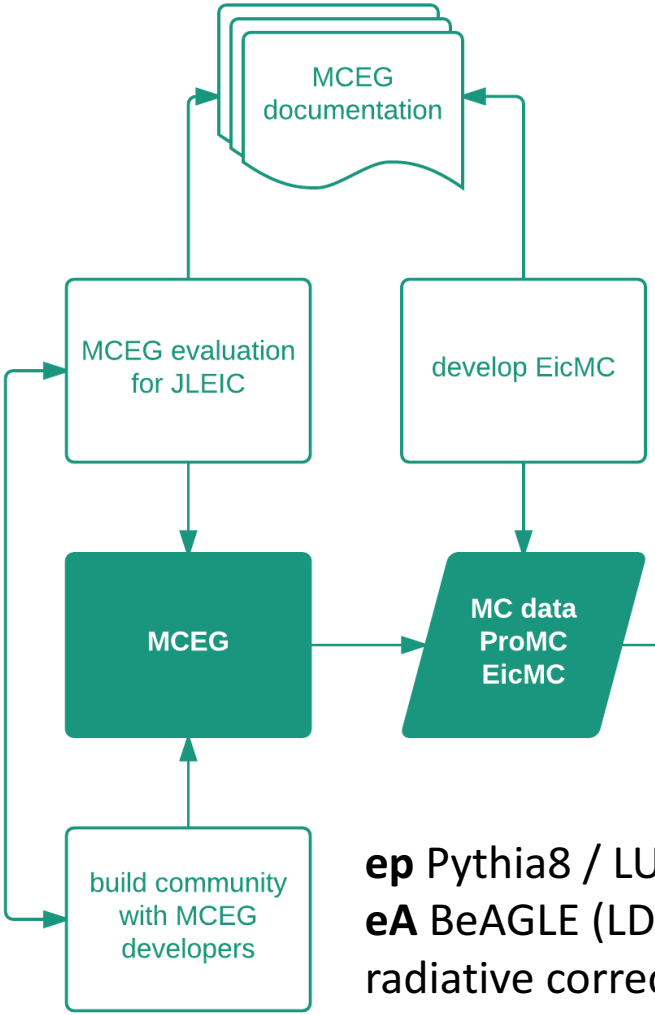


ANL (3) W. Armstrong, S. Chekanov, D. Blyth
BNL (4) E.-C. Aschenauer, A. Kiselev (co-PI) , J. Lauret, C. Pinkenburg
Fermilab (1) S. Prestel
INFN Trieste (1) A. Bressan
Jefferson Lab (4) MD (co-PI) , D. Lawrence, D. Romanov, M. Ungaro
SLAC (2) M. Asai, A. Dotti
William & Mary (1) W. Deconinck

Community effort

- tutorials for use cases
- promote our work towards interfaces and integrations and the tutorials within EIC UG

Monte Carlo simulations



ESC MC format driven by

- self-descriptive data format
- HPC compatible

Repository eicsim01.jlab.org
Client hepsim.jlab.org (tba)
Sergei Chekanov (ANL)

ep Pythia8 / LUND group (LDRD)
eA BeAGLE (LDRD M. Vasiliy)
radiative correction (ESC with H. Spiesberger (HERA))

EicMC – self-descriptive MC files

by-product of ProMC import incorporation into EicRoot framework

➔ compact and portable C++ code (~3k lines) developed by Alexander Kiselev (BNL)

Main objectives

- unified format for all MCEG for EIC (and beyond)
- avoid using either ROOT or ASCII file formats in the future MC repository
- overcome few intrinsic ProMC format limitations

EicMC features

in final stage of validation

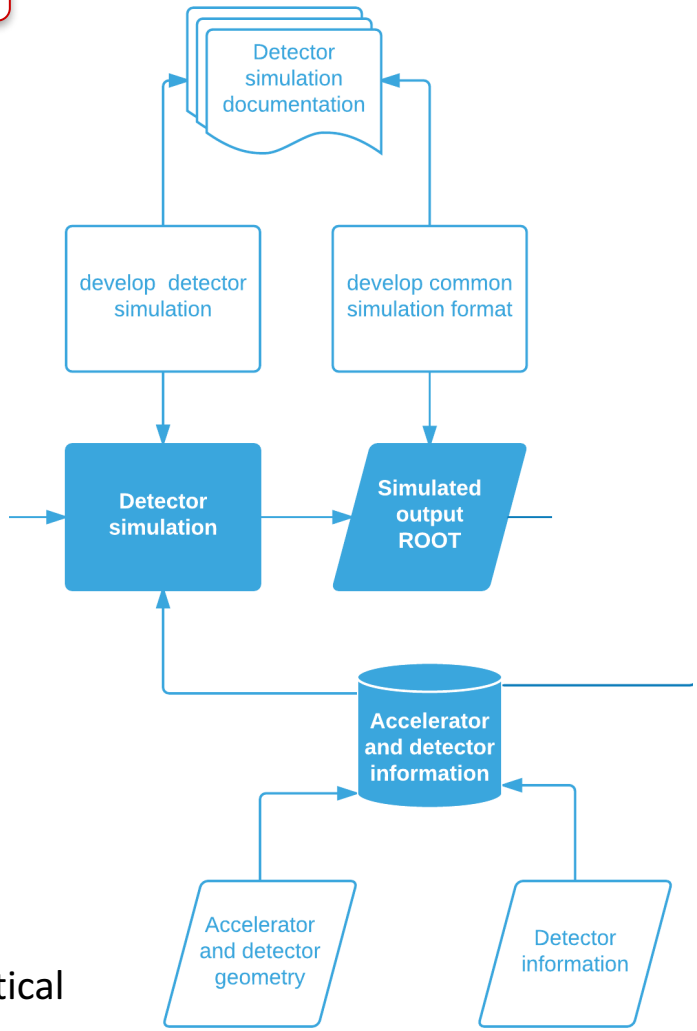
- unified extendable binary MCEG format (all generators from eic-smear are supported)
- true automatic self-description built into the format core .proto file and the user library
- 64-bit implementation ➔ no 16-/32-bit limitations on file size, record count
- flexible set of compression schemes (for file size, input speed, floating point precision)
- both sequential and direct access to event records (with scalable multi-level catalogues)
- no external dependencies on the user (input) side other than Google Protobuf libraries
- convenient user interface
- performance (file size, speed) similar to or better than ProMC and ROOT equivalents

Detector simulations

GEMC (M. Ungaro)

ideal for detector concepts

- application for detector simulations based on Geant4
- reducing the learning curve to use Geant4
  - macro language for detector design
  - various geometry definitions (GEMC, gdml, CAD)
  - data card (XML) to steer application, all Geant4 macro commands supported by design
  - GUI for interactive sessions
  - excellent documentation
- full Geant4 support: adding Geant4 features relatively simple
- transparent in-house development



GEMC for JLEIC (Z. Zhao)

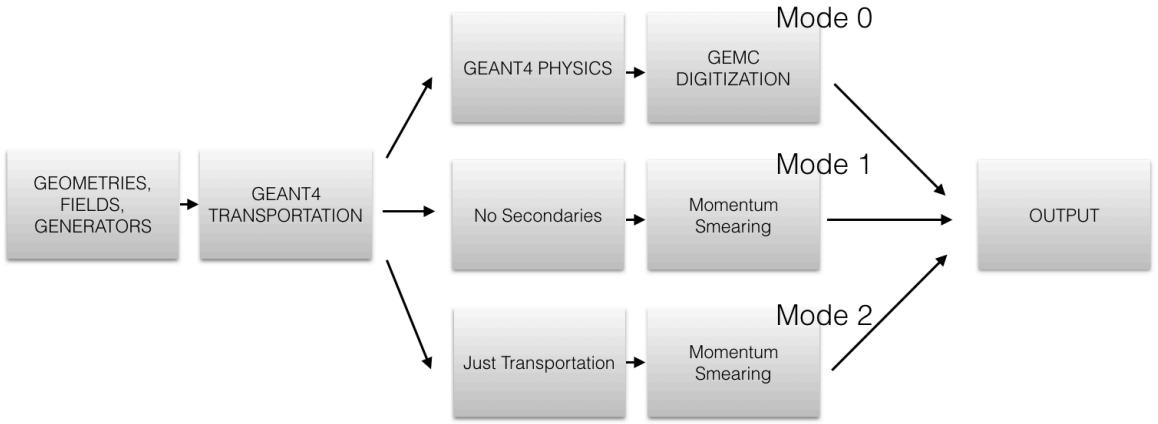
Simulations level

same application for fast and full detector simulations fully adjustable simulation levels, e.g.,

- only material transport
- using Geant4 for geometry and physics only in some critical areas and ad-hoc non-Geant4 models in other regions

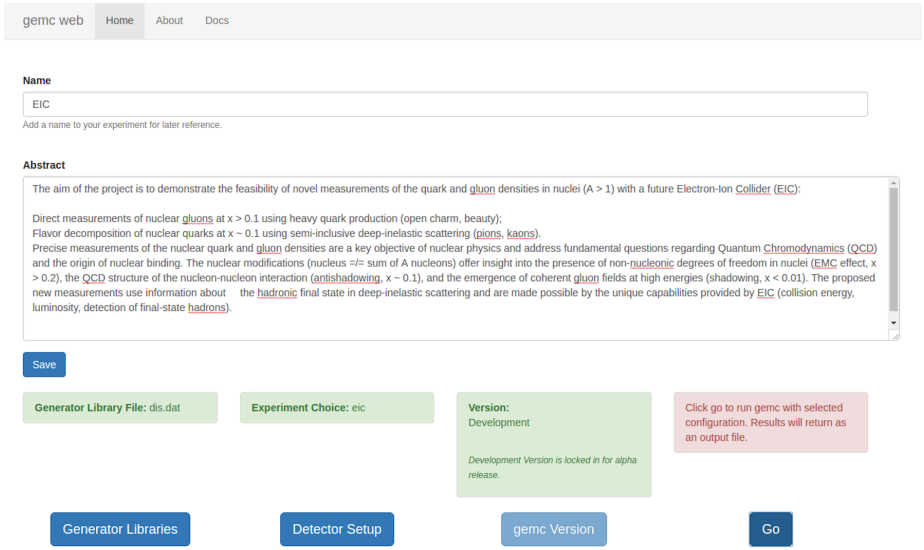
GEMC development for JLEIC

Adjustable simulation level



Web interface

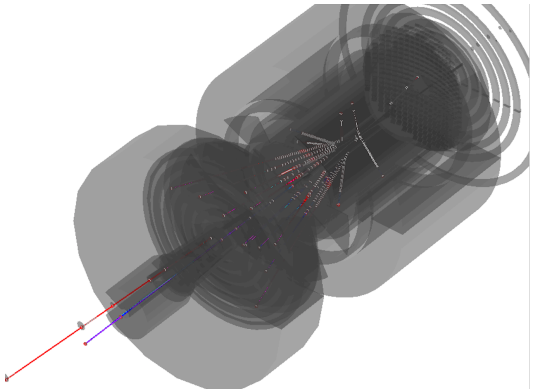
Sam Markelon (UConn)

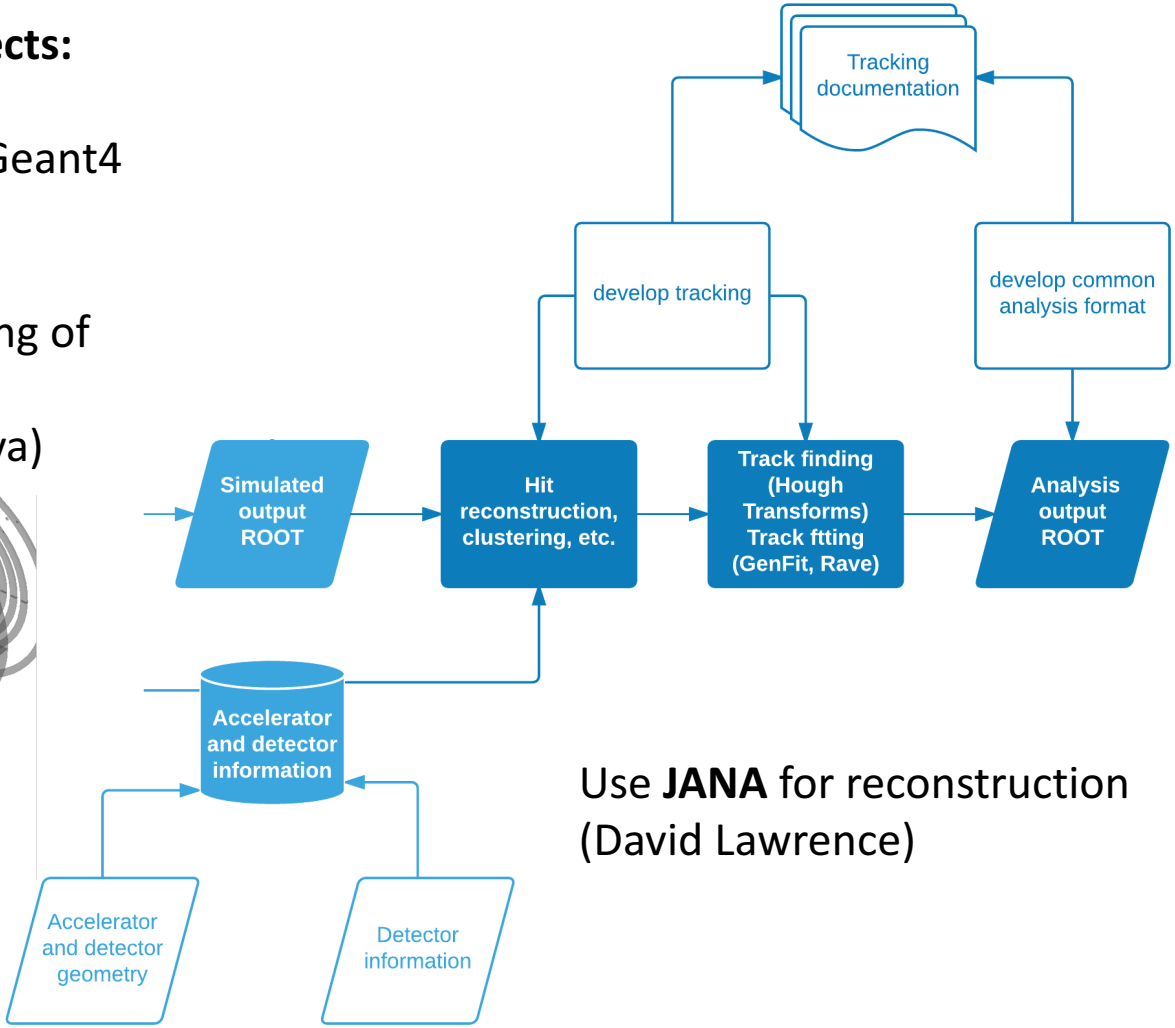


Reconstruction

Initial reconstruction projects:

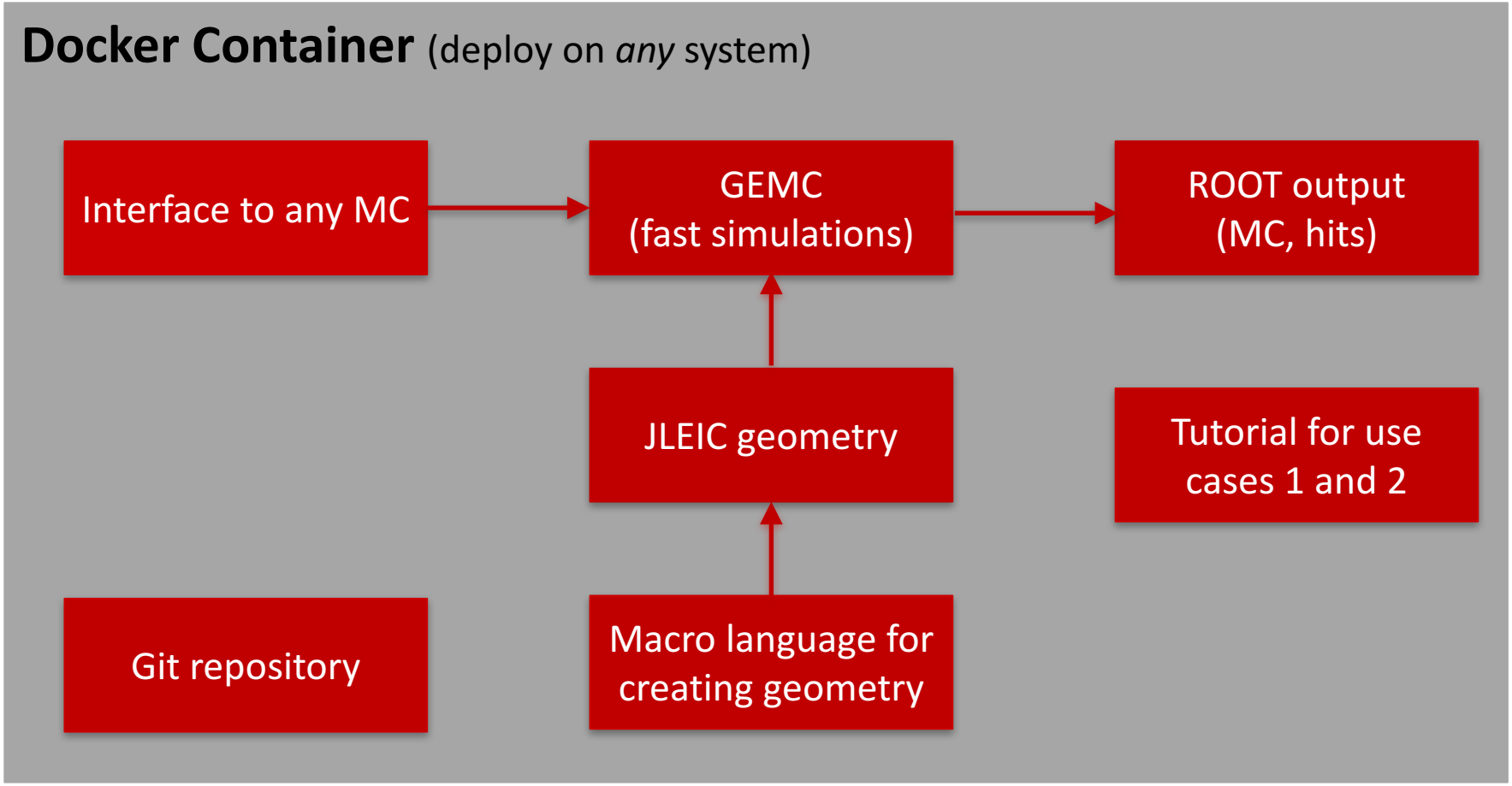
- Track fitting of generic Geant4 geometry (Whitney Armstrong, MD)
- Track fitting and vertexing of JLEIC Geant4 / GEMC geometry (Julia Furletova)





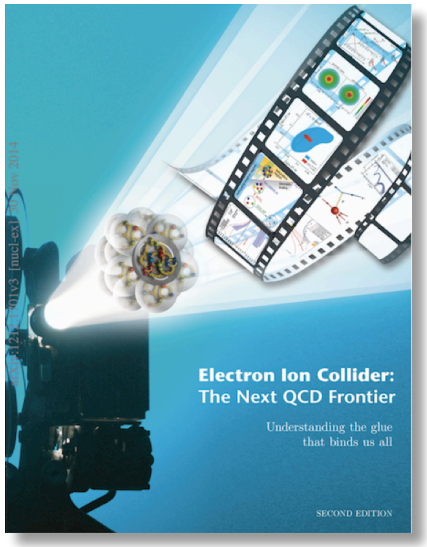
Use **JANA** for reconstruction (David Lawrence)

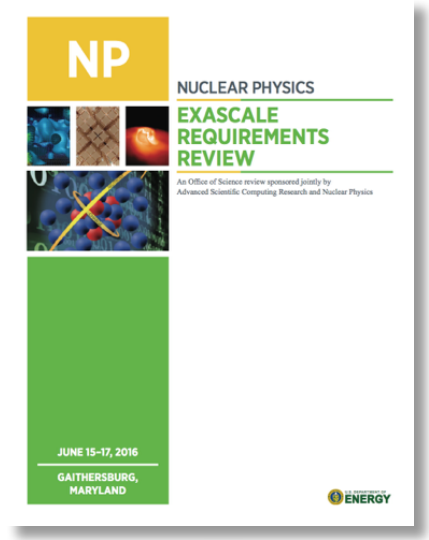
Summer deliverable



FUTURE

Towards a computing vision for the EIC





EIC

- extremely broad science program
- strong **interplay theory – experiment**
- **EIC Software Consortium** (EIC Generic R&D Program)

Computing requirements

- flexible, modular analysis ecosystem
- interplay of data analysis, MC calculations, and Lattice QCD effort

Lessons learned from LHC

- computing central to success of scientific goals
- complexity of analysis ecosystem limits time on physics analysis
- strong role of machine learning

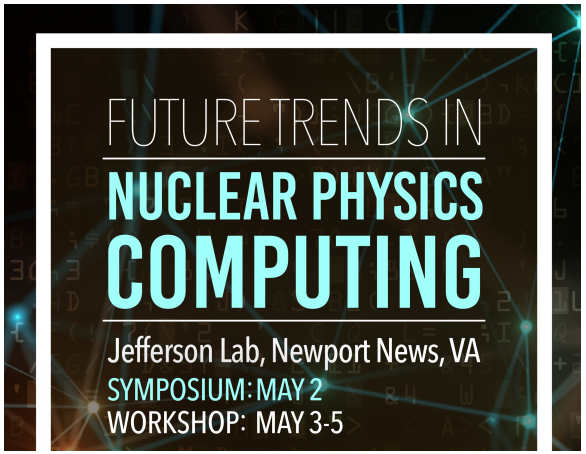
Era of Exascale Computing

- changing the paradigm for I/O, storage and compute
- high-precision non-perturbative QCD measurements (MC, Lattice QCD)

Summary

Improve analysis environments with user-centered design

JLEIC software planning based on user requirements



Vision for EIC Computing based on community effort