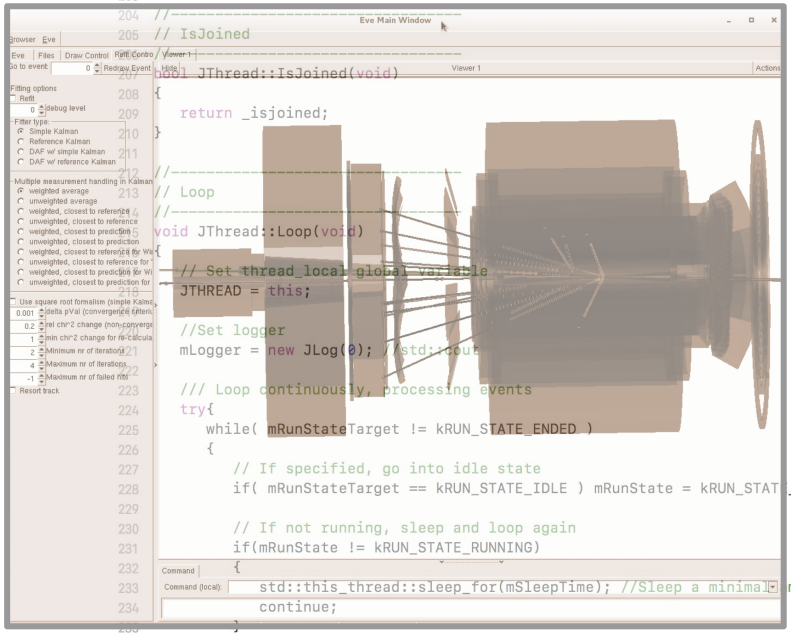
JLEIC Software Status

Status of Simulation and Reconstruction Software at JLEIC

Wouter Deconinck**, Markus Diefenthaler*, Yulia Furletova*, **David Lawrence***, Maurizio Ungaro*, Zhiwen Zhao***
(... and others ...)

*Jefferson Lab
**William and Mary
***Duke



Simulation and Reconstruction

Near Term Goals/Purpose:

1. Study ability to measure Physics Processes
   a. Interface to MC Event Generators
   b. Provide detector responses (Fast MC for acceptance/resolution, ab initio for backgrounds)

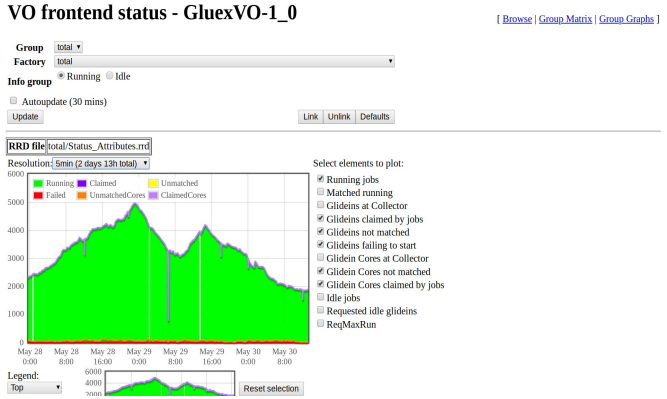
2. Study/Refine Detector Design
   a. Interface to MC Event Generators
   b. Provide detector responses (Fast MC for acceptance/resolution, ab initio for backgrounds)

(JLab) Experimental Computing Performance Plan

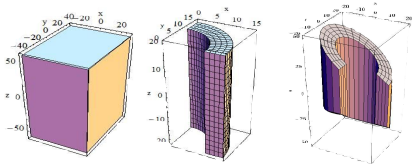
- **Insure adequate computing resources with $850K investment in FY18**
  - Use local farm for reconstruction, calibration and analysis
  - Use distributed resources for MC
  - Storage and associated bandwidth scaled to support all resources
- **Open Science Grid**
  - GlueX -6 institutions contribute resources
  - In a recent 2 week period ~1M core-hours
  - Expect yearly 35M-50M core-hours
  - Investigating options for CLAS12
- **GlueX reconstruction code at NERSC**
  - Scale test in July
  - Anticipate 70M core-hours/year
- **Cloud Computing available for bursts**

	Current	FY19	FY20
CPU (M-core-hours/year)	37	70	90
Scratch Disk & Cache Disk (PB)	0.65	1.1	2
Tape (GB/s)	3	5	7
WAN bandwidth (Gbps)	10	10	10

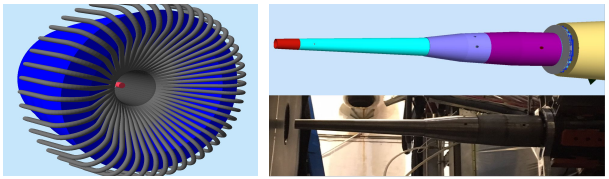
Current and Projected Capacity



GEMC Framework



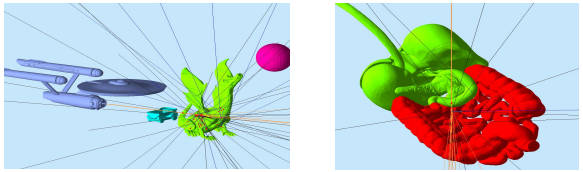
Native Geant4

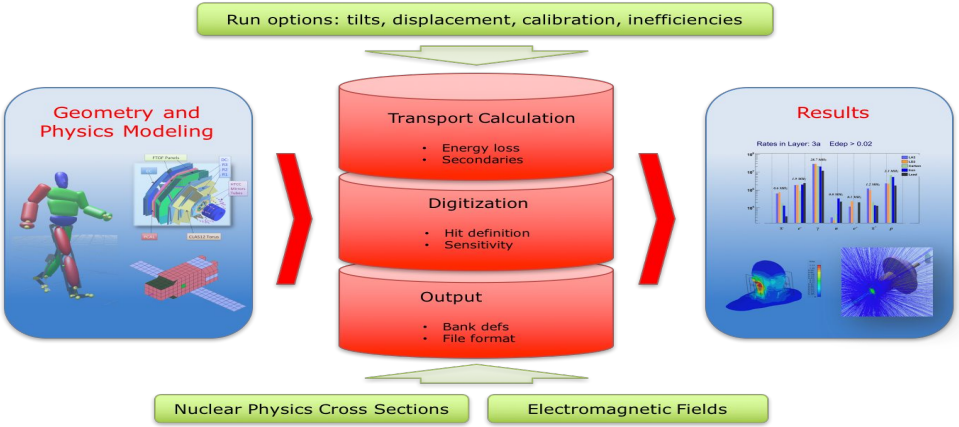


Many CLAS12 detectors use actual engineering models (STL)

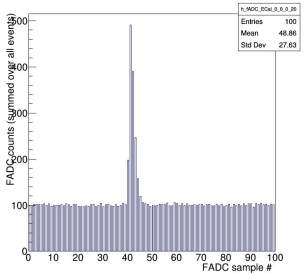
CAD

GDML

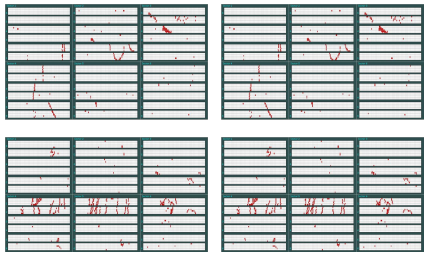




- Input: Native, CAD, GDML, can be mixed and matched.
- FADC Mode 1 (crate, slot channel)
- Background Merging.
- FAST MC Mode.
- Digitization uses actual CCDB calibration constants.



GEMC EC FADC Mode 1



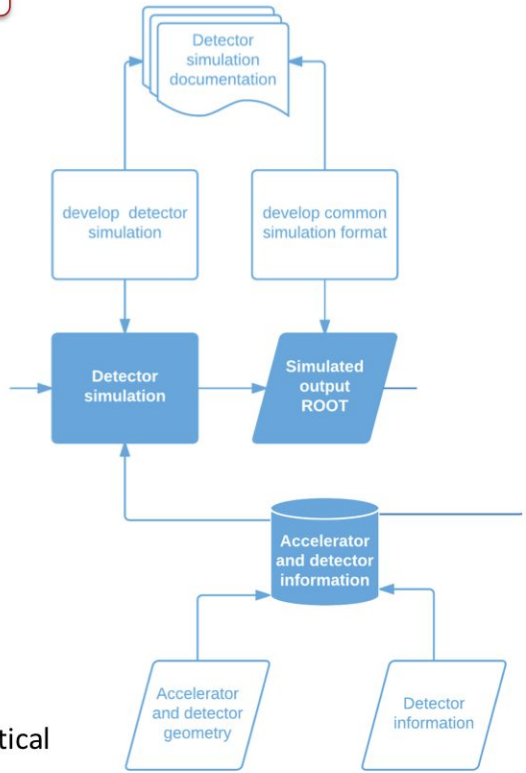
Merging of random trigger data with GEMC

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



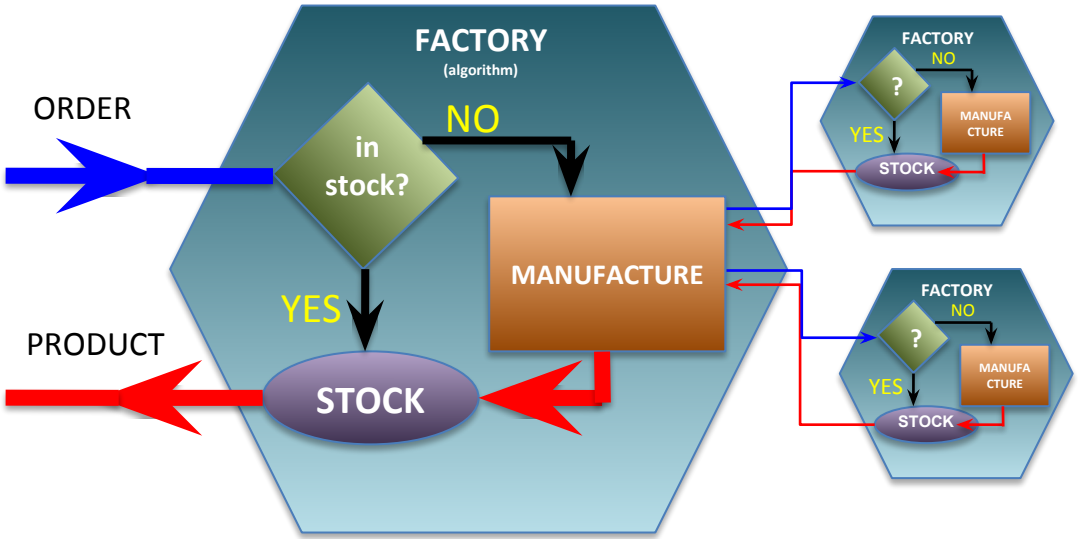
GEMC for JLEIC (Z. Zhao)

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

JANA Factory Model




```
auto tracks = jevent->Get<DTrack>();

for(auto t : tracks){
  // ... do something with a track
}
```

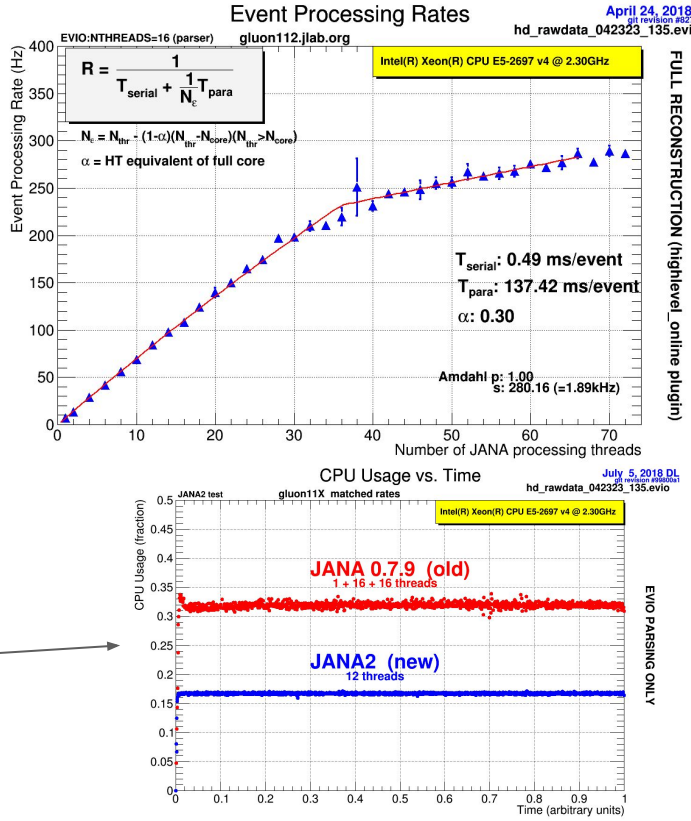
Data on demand = Don't do it unless you need it
const Stock = Don't do it twice

Conservation of CPU cycles!

JANA: C++ Software framework for Reconstruction Workflow

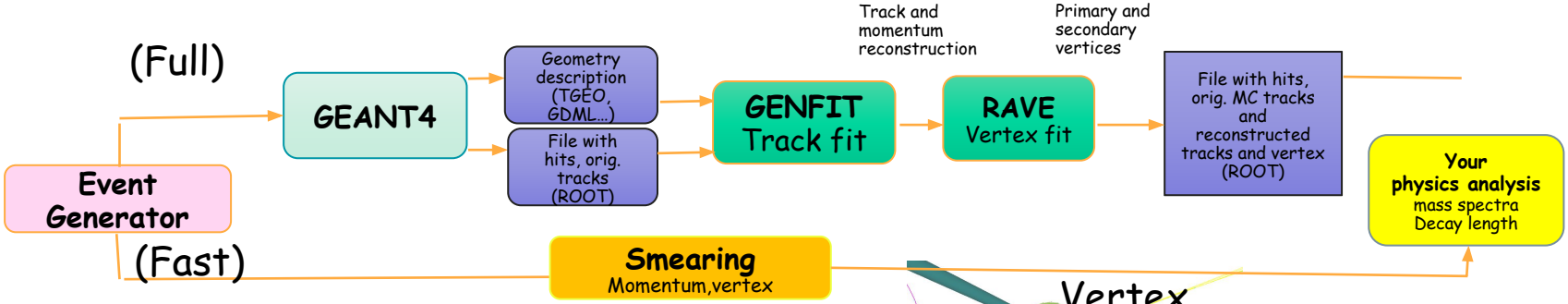
- Multi-threaded
- Modular, user-focused design
- Developed over the past 13 years specifically for 12GeV era of high rate experiments at JLab
- Used for GlueX online DQM, offline reconstruction and L3 trigger system*
- LDRD project for development of JANA2 started in FY18

JANA rate scaling for GlueX Data Reconstruction

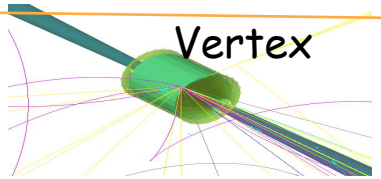


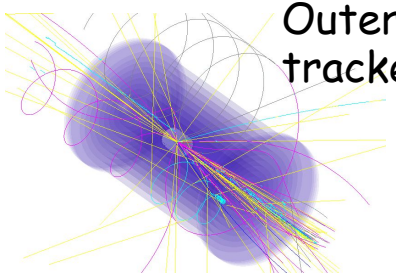
Reconstruction Chain (for LDRD)

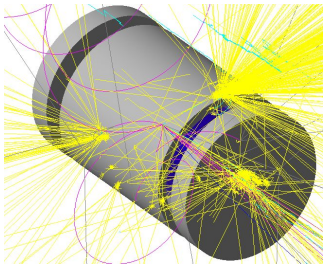
This chain has been developed to validate tracking and vertex parameters and was used for JLAB LDRD- 1601/1701 project ("Nuclear gluons with charm at EIC") to estimate a detector effect on a charm reconstruction. (Many thanks to Whitney Armstrong, Alexander Kiselev and "software consortium" for ideas and discussions)



- Geometry created/described **inside GEANT4**, and then distributed via **Root TGEO**
- All parts are connected via intermediate **Root files** (ntuples)
- Event generators (Pythia6, Herwig 6 and 7) HEPMC, Lund format.
- Original (MC) tracks are traced down to analysis
- No pattern recognition ! Only track fit and vtx!



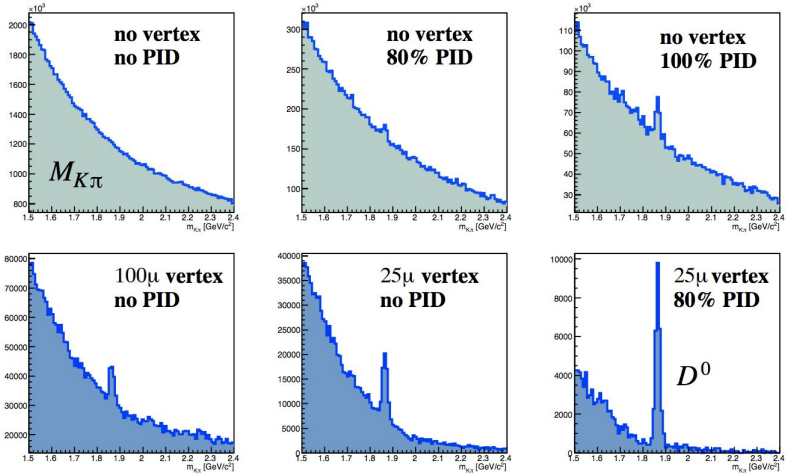




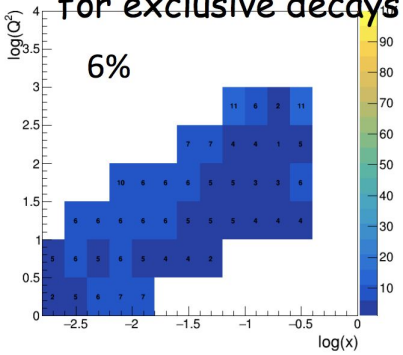
Analysis

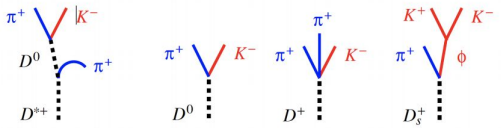
- ✓ Process charm (BGF)-only events
- ✓ Process and add all "background" events (all other non –BGF DIS events)
- ✓ Estimate efficiency and set a requirements for detector (PID, vertex, etc)

D⁰ on top of DIS background



Charm efficiency
for exclusive decays



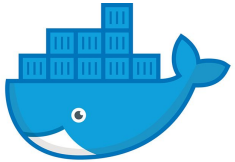


Containers

1. Install Docker or Singularity

2. Run container

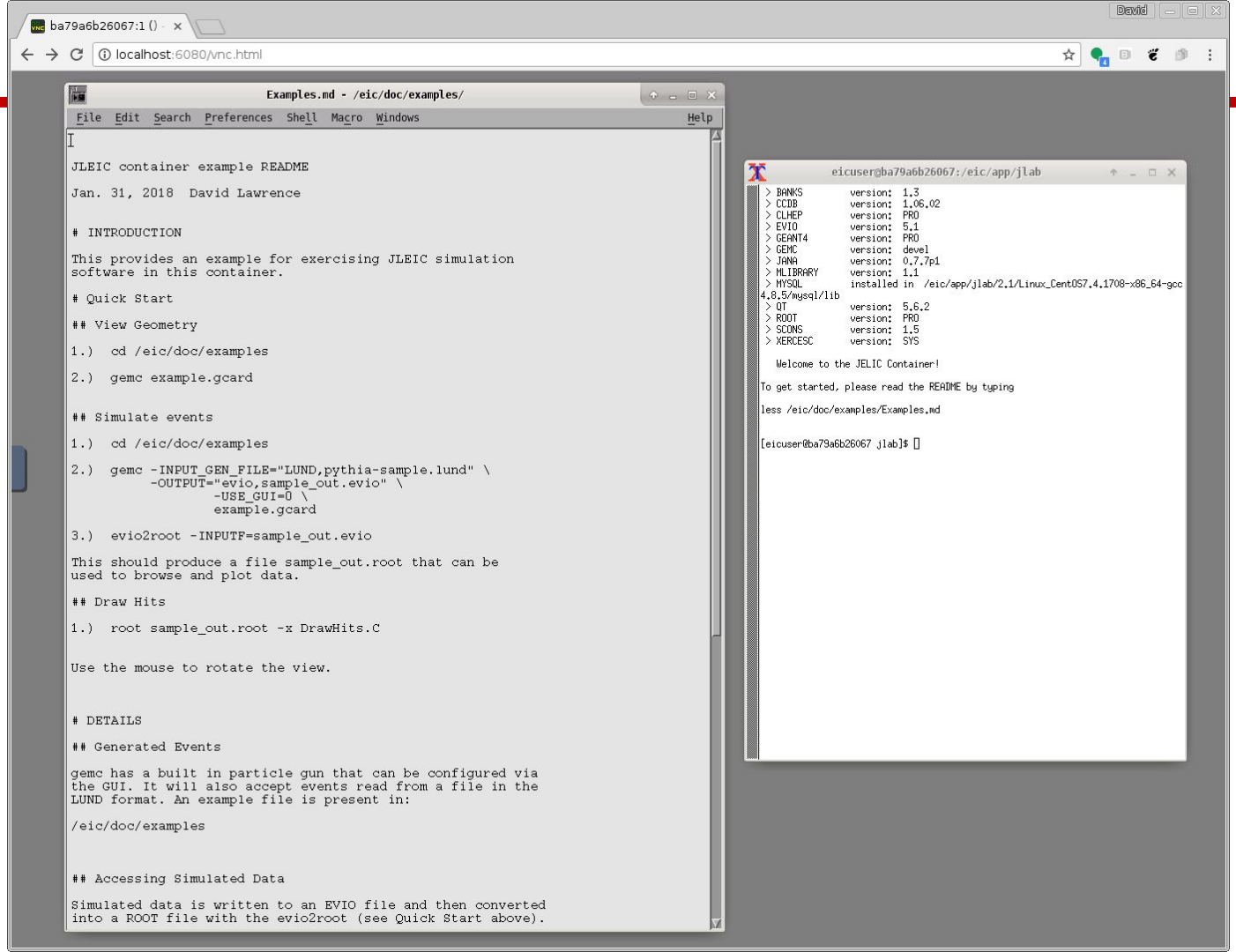
   docker run -p 6080:6080 -v /my/data/dir:/data -it --rm electronioncollider/jleic:1.0.4
   or
   singularity shell shub://electronioncollider/jleic:1.0.4 /container/utilities/xstart.csh

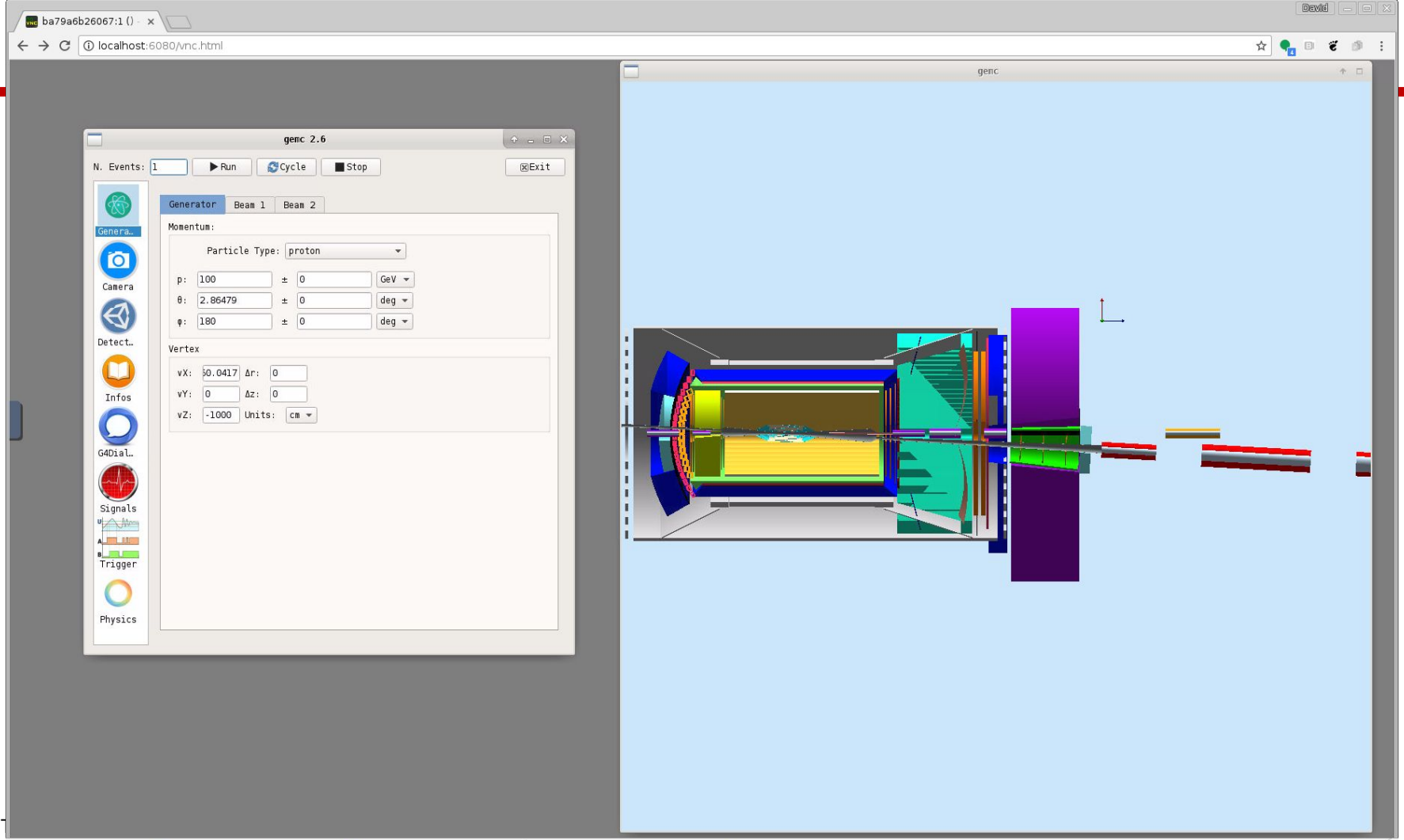


3. Point browser to:

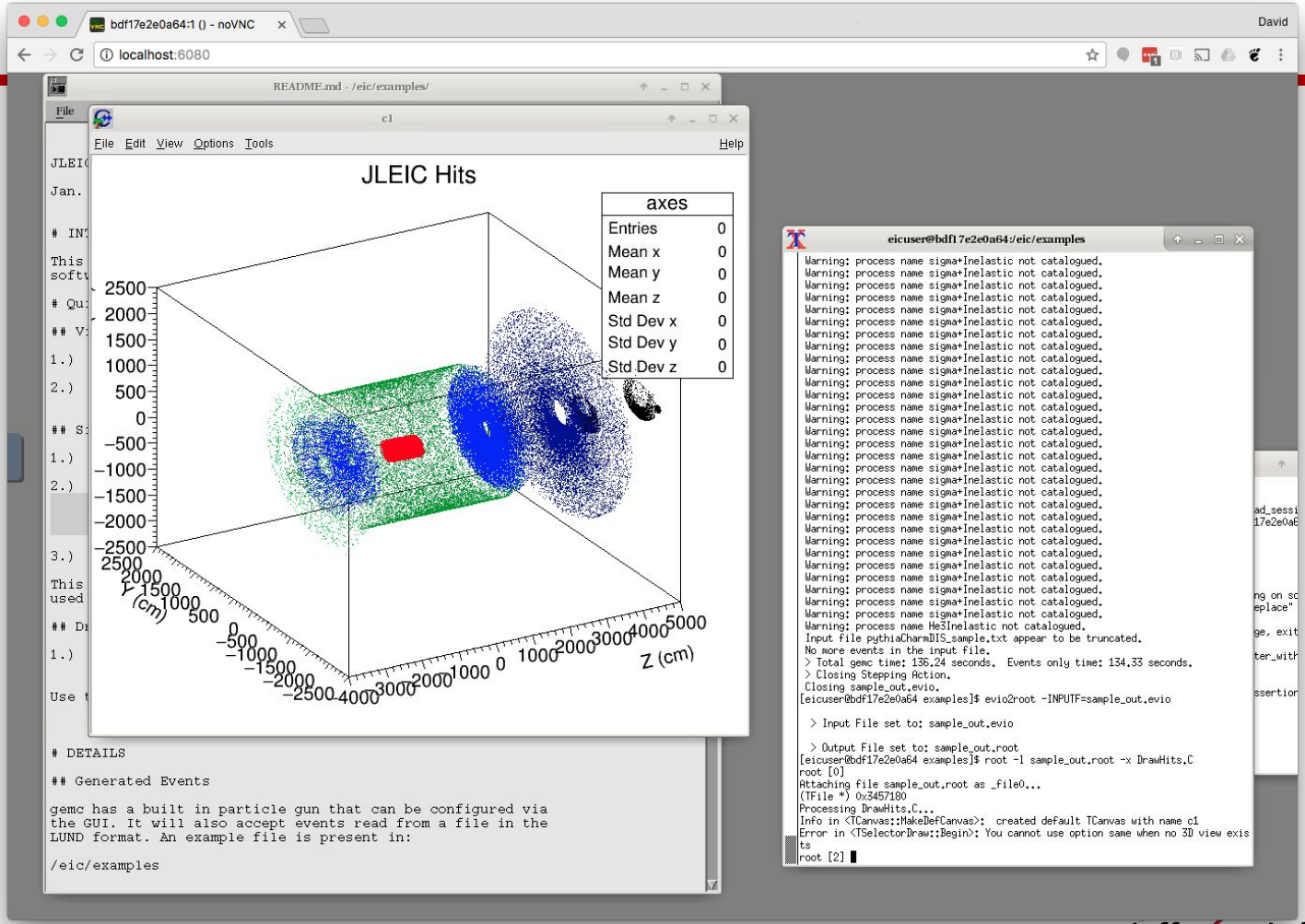
   http://localhost:6080

JLEIC Desktop Environment via web browser on host




ba79a6b26067:1 ()
localhost:6080/vnc.html
gemc 2.6
N. Events: 1
Run
Cycle
Stop
Exit
Genera...
Camera
Detect...
Infos
G4Dial...
Signals
Trigger
Physics
Generator
Beam 1
Beam 2
Momentum:
Particle Type: proton
p: 100 ± 0 GeV
θ: 2.86479 ± 0 deg
φ: 180 ± 0 deg
Vertex
vX: 50.0417 Δr: 0
vY: 0 Δz: 0
vZ: -1000 Units: cm
gemc

JLEIC Desktop Environment via web browser on host

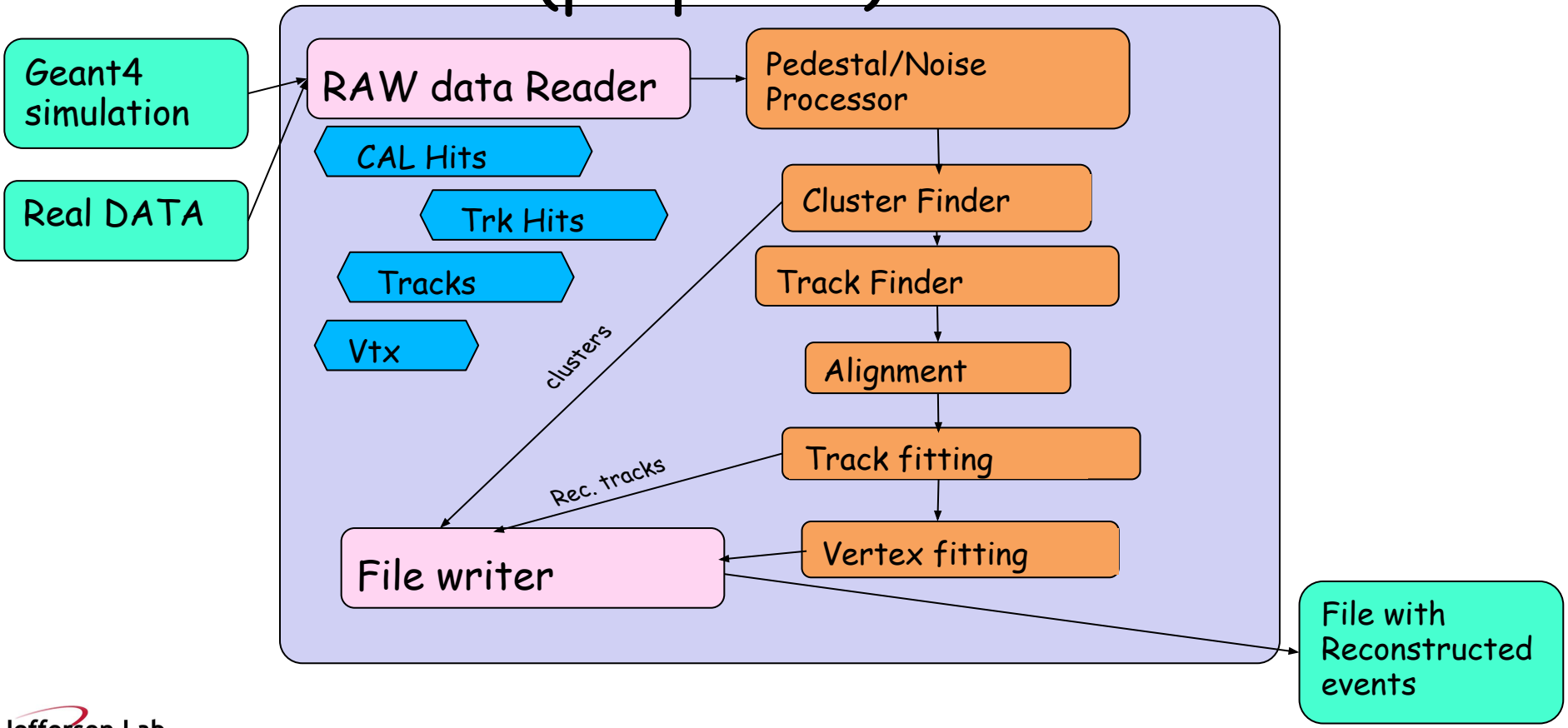


Summary and Future

- Simulation
  - GEMC (Geant4) used for simulation
  - Refining and merging geometries
- Reconstruction
  - Some work completed on tracking/vertexing using GenFit and RAVE
  - Actively integrating reconstruction components into single project using JANA framework
- Containers
  - jleic containers published on Docker hub and Singularity hub
  - targeting interactive desktop/laptop use vs. batch

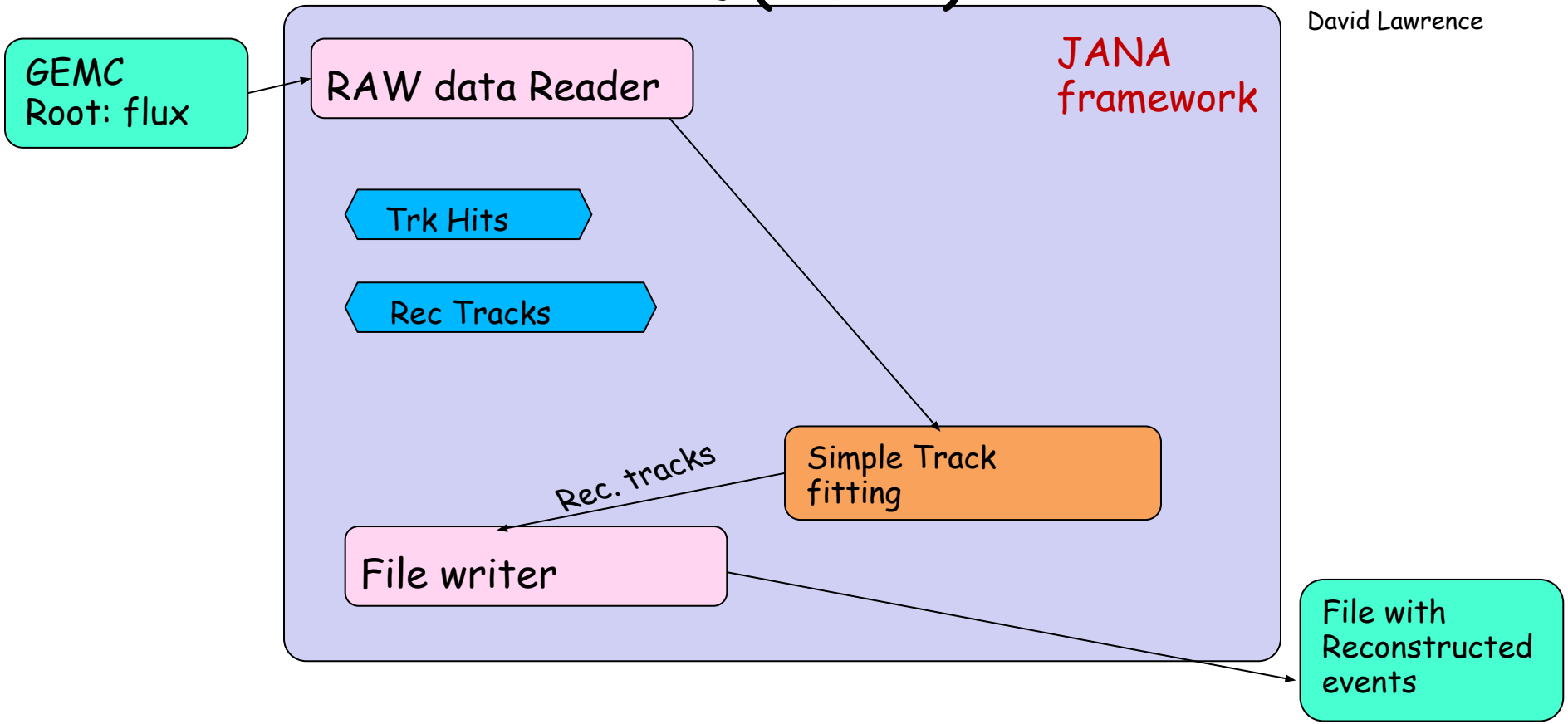
Backups

Reconstruction (proposed)



Reconstruction with J(LEIC)ANA

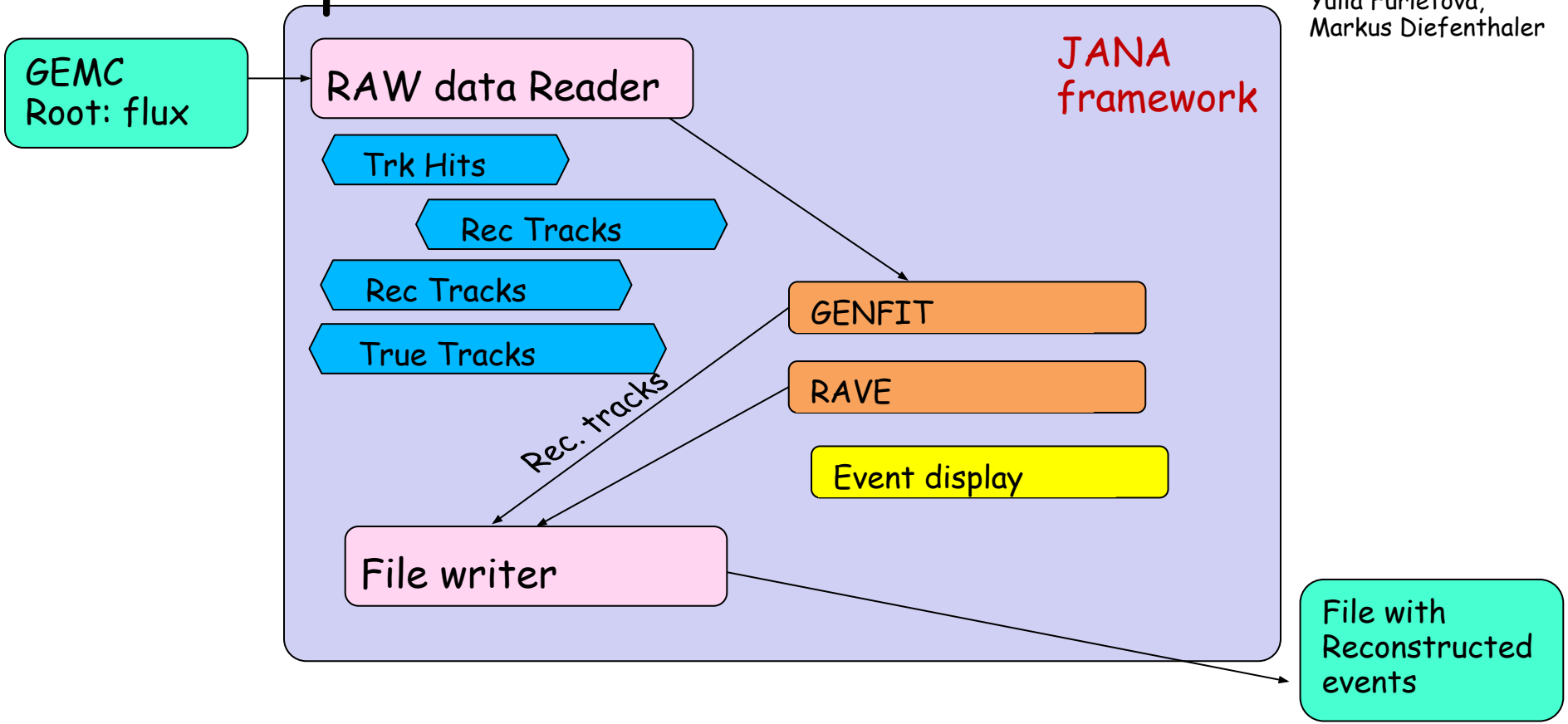
David Lawrence



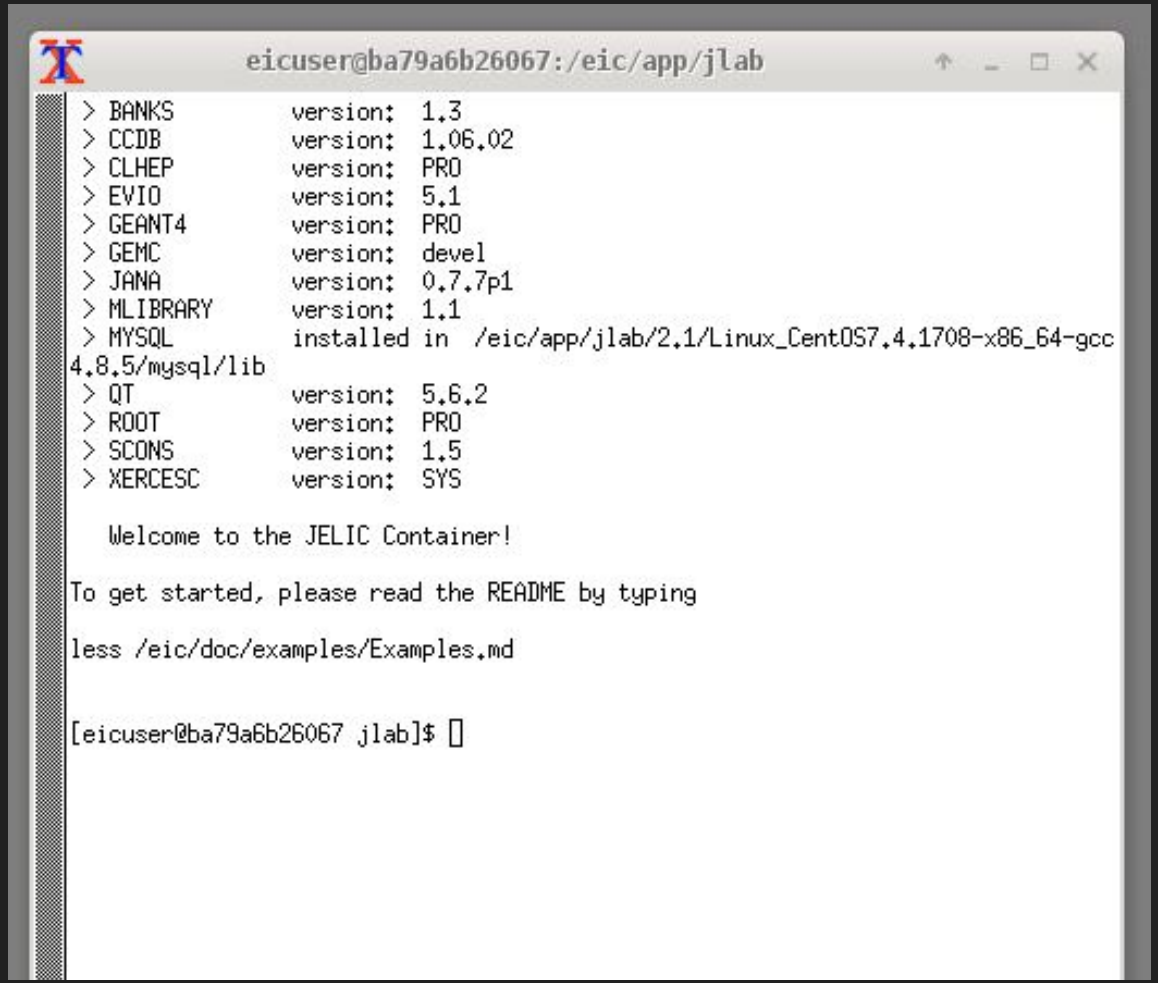
Yulia Furletova

Next steps..

David Lawrence,
Dmitri Romanov,
Yulia Furletova,
Markus Diefenthaler



JLEIC Desktop Environment via web browser on host

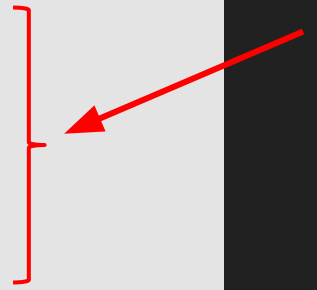


JLEIC Desktop Environment via web browser on host

```
JLEIC container example README

Jan. 31, 2018  David Lawrence


# INTRODUCTION

This provides an example for exercising JLEIC simulation
software in this container.

# Quick Start

## View Geometry

1.)  cd /eic/doc/examples

2.)  gemc example.gcard


## Simulate events

1.)  cd /eic/doc/examples

2.)  gemc -INPUT_GEN_FILE="LUND,pythia-sample.lund" \
          -OUTPUT="evio,sample_out.evio" \
                  -USE_GUI=0 \
                  example.gcard

3.)  evio2root -INPUTF=sample_out.evio

This should produce a file sample_out.root that can be
used to browse and plot data.
```

JLEIC Desktop Environment via web browser on host

```
JLEIC container example README

Jan. 31, 2018  David Lawrence


# INTRODUCTION

This provides an example for exercising JLEIC simulation
software in this container.

# Quick Start

## View Geometry

1.)  cd /eic/doc/examples

2.)  gemc example.gcard


## Simulate events

1.)  cd /eic/doc/examples

2.)  gemc -INPUT_GEN_FILE="LUND,pythia-sample.lund" \
          -OUTPUT="evio,sample_out.evio" \
                  -USE_GUI=0 \
                  example.gcard

3.)  evio2root -INPUTF=sample_out.evio

This should produce a file sample_out.root that can be
used to browse and plot data.
```



Secure https://jeffersonlab.github.io/swcarpentry-jlab-singularity/02-jlabce-container/index.html

Home Code of Conduct Setup Episodes Extras License Improve this page Search...

Using Singularity at Jefferson Lab

Running the JLab CE container

Overview

Teaching: 5 min
Exercises: 10 min

Questions
- How can we replicate the Jefferson Lab Common Environment on other systems?

Objectives
- Understand how tags are used to version containers.
- Load and use the Jefferson Lab Common Environment on the interactive farm nodes.

Tags: versioning of containers

In the previous episode we downloaded the lolcow container, at `shub://GodloveD/lolcow` and the container was stored with the filename `GodloveD-lolcow-master-latest.simg`. Let's analyze that URL and filename.

- `shub` indicates that the URL points to a Singularity Hub location.
- `GodloveD` is the user who provided this container (David Godlove, if you must know, see for example this GitHub page).
- `lolcow` is the name of the repository that was used to build this container.
- `master` is the branch from which the container was built.
- `latest` is the tag of the container, with latest for the most recent build.

The tag is commonly used for versioning of containers. By specifying the URL as `shub://GodloveD/lolcow:latest` we can explicitly ask for the latest version of the lolcow container.

Since there is not a lot of versioning one can do on this container, we will first introduce a container where versions ARE important.

Retrieving the Jefferson Lab Common Environment container from Singularity Hub

Now that we have the basics of containers behind us, we can use our first 'useful' physics container: the Jefferson Lab Common Environment container. This container replicates the scientific software suite that is installed on the interactive farm nodes, but packages it up in a nice container.

Of course, there is no real practical need to load the Jefferson Lab Common Environment on a Jefferson Lab interactive farm node, but bear with me for now.

Here is how we download the container:

JLab Software Carpentry Workshop:

EIC container effort feeding back into production operations at JLab