Start the docker

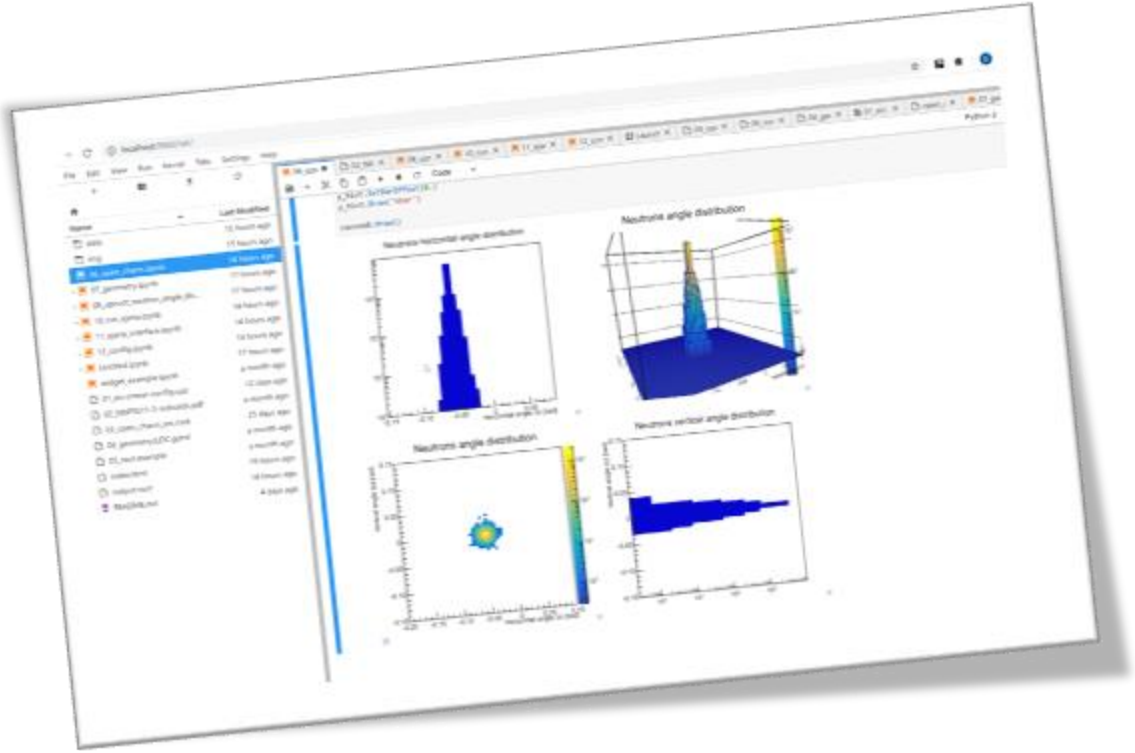
docker pull electronioncollider/escalate

How to install docker:

eic.gitlab.io / documents / quickstart

Entry point for EIC software

Interactive tutorial
Fast and full simulations
In ESCalate



Dmitry Romanov

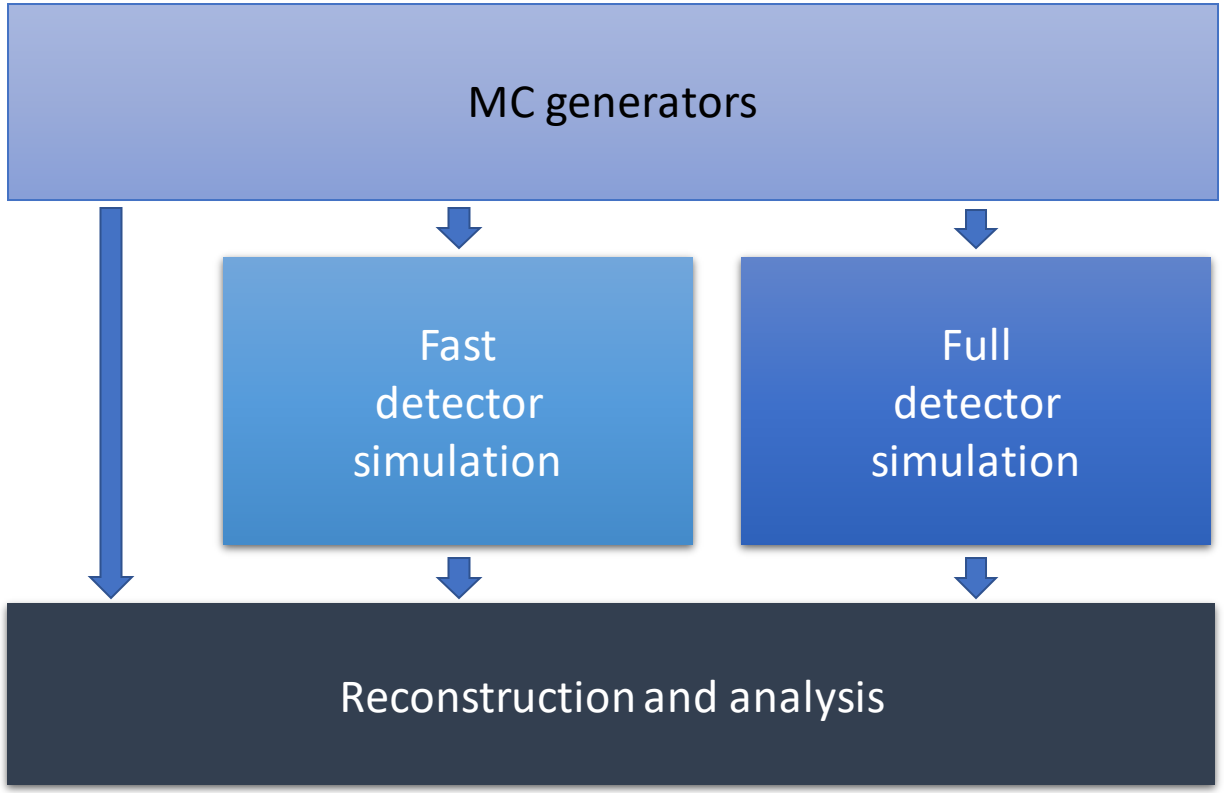
Start the docker

docker pull electronioncollider/epic-gui

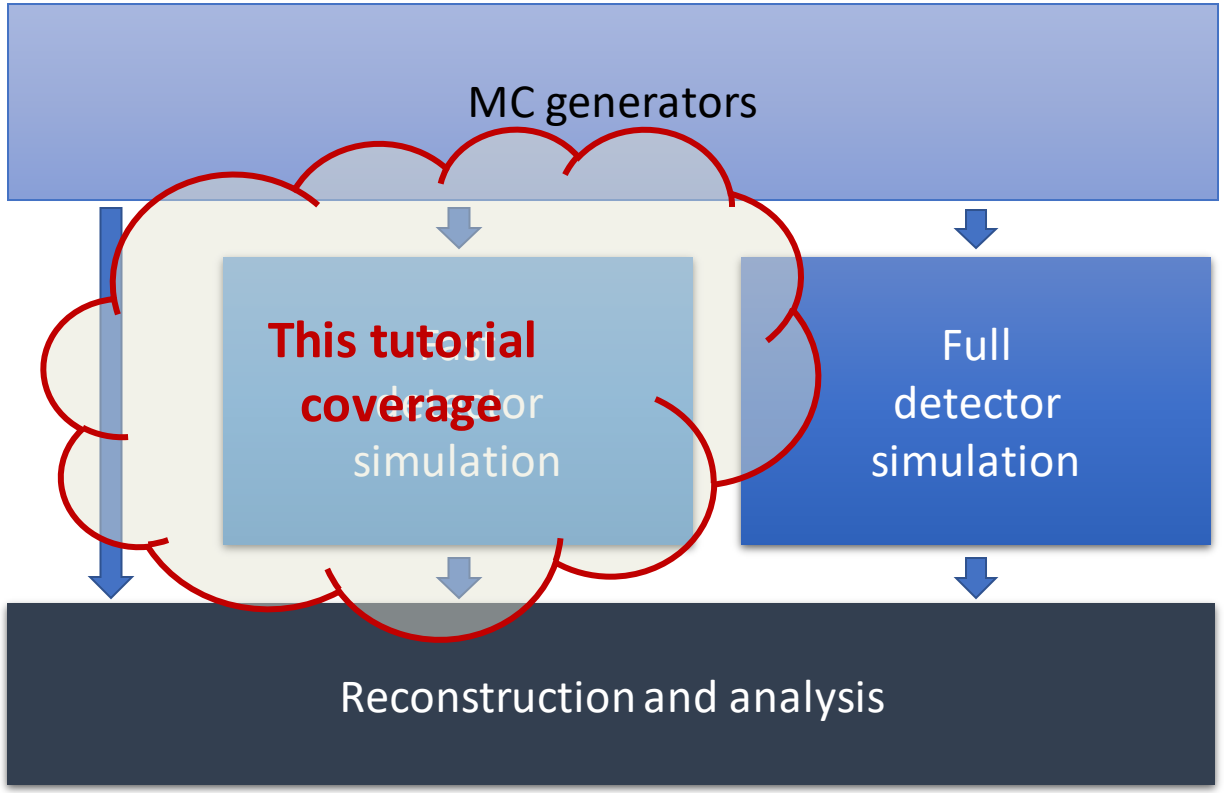
Plan now:

- Short introduction to our software
- **1. Tutorial – running fast simulation**
- **2. Tutorial – running in different environments**
- **2. Tutorial – running full simulation**
- 3 PM - Answering questions (presentation)
- Adjourn!

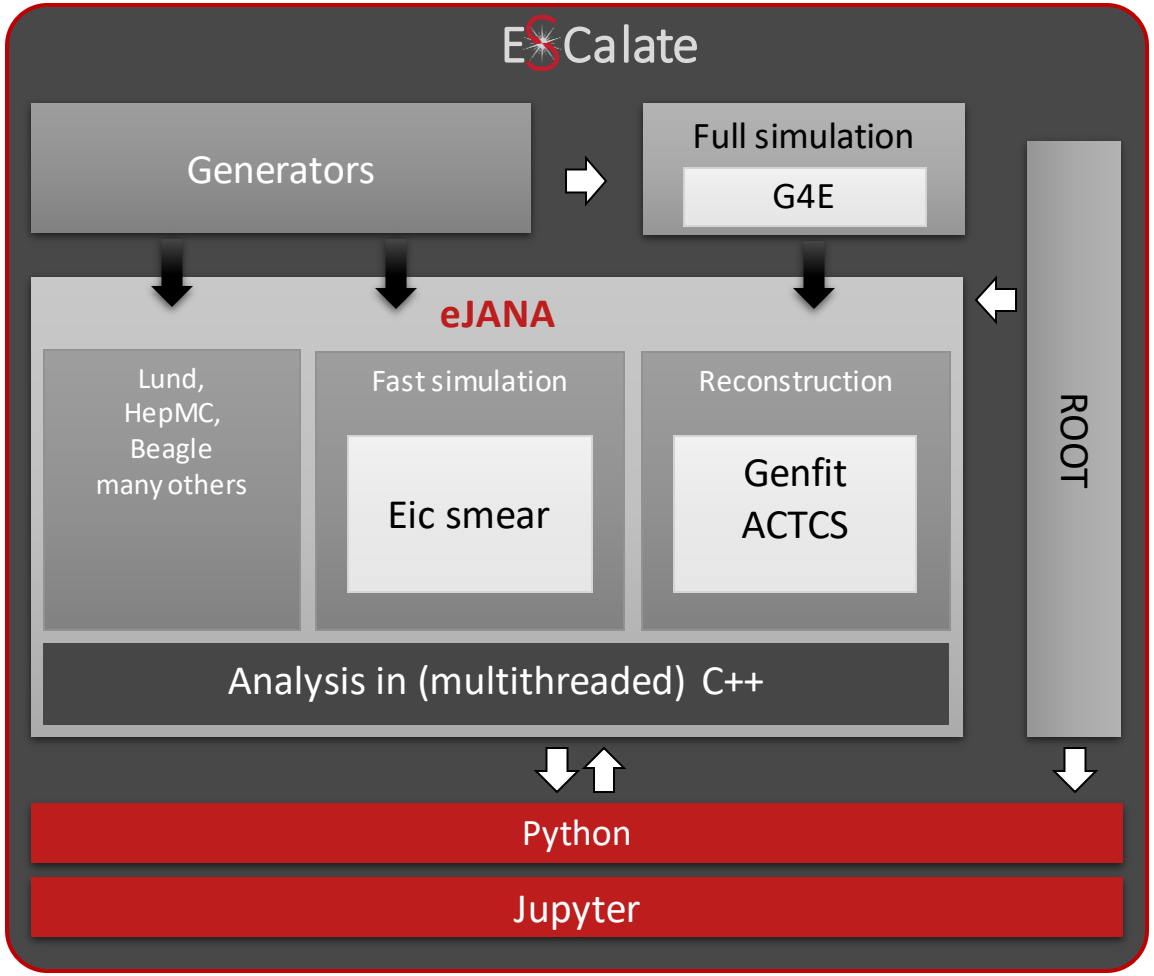
Software chain



Software chain



Software stack for EIC simulations



ESC – EIC Software and Computing group

Software version table

electronioncollider/escalate v 1.0.1

(Changed packet versions are bold)

Core tools		HENP		MCEG		EIC	
Packet	**Version**	**Packet**	**Version**	**Packet**	**Version**	**Packet**	**Version**
gcc	9.2.1	eigen3	3.3.7	LHAPDF6	6.2.3	ejpm	**0.3.12**
CMake	3.17.0	clhep	2.3.2.2	pythia8	8.244	eic-smear	**1.0.4f1**
python	3.7.5	hepmc	2.6.9	DIRE	2.004	jana	**2.0.2**
ROOT	6-20-04	**hepmc3**	**3.2.1**	Cernlib	2006-12-20	ejana	**1.2.2**
Geant4	10.6.1	vgm	4.5	lhapdf5	5.9.1-6	g4e	**1.3.4**
		genfit	2020.1	PYTHIA6	RAD-CORR		
		acts	0.22.00				
		delphes	3.4.2				
		fastjet	3.3.3				

Update the docker

docker pull electronioncollider/escalate

eic.gitlab.io / documents / quickstart /

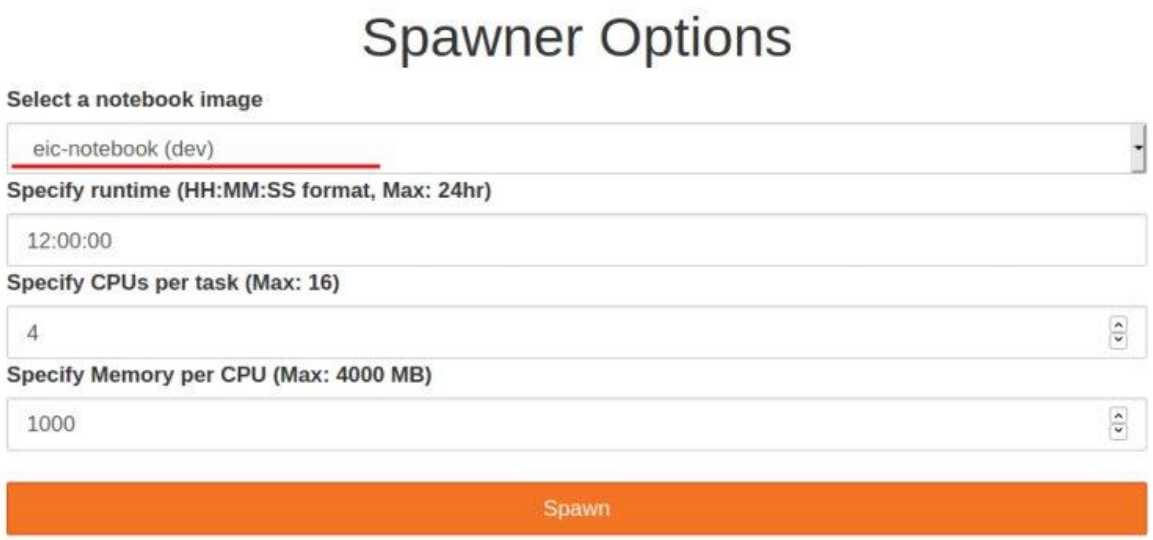
Escalate at JeffersonLab JupyterHub

- jupyterhub.jlab.org

  To download all examples:

  ```
  git clone https://gitlab.com/eic/escalate/workspace
  ```

- Full documentation on JupyterHub



launch binder

Singularity on farms

- `module load singularity`
- `singularity shell --cleanenv /cvmfs/eic.opensciencegrid.org/singularity/escalate:latest`
- `source /etc/profile`

- Available both for Jlab and BNL farms
- Full ESCalate singularity docuementation

Smear tool

- Smearing should be as easy as:

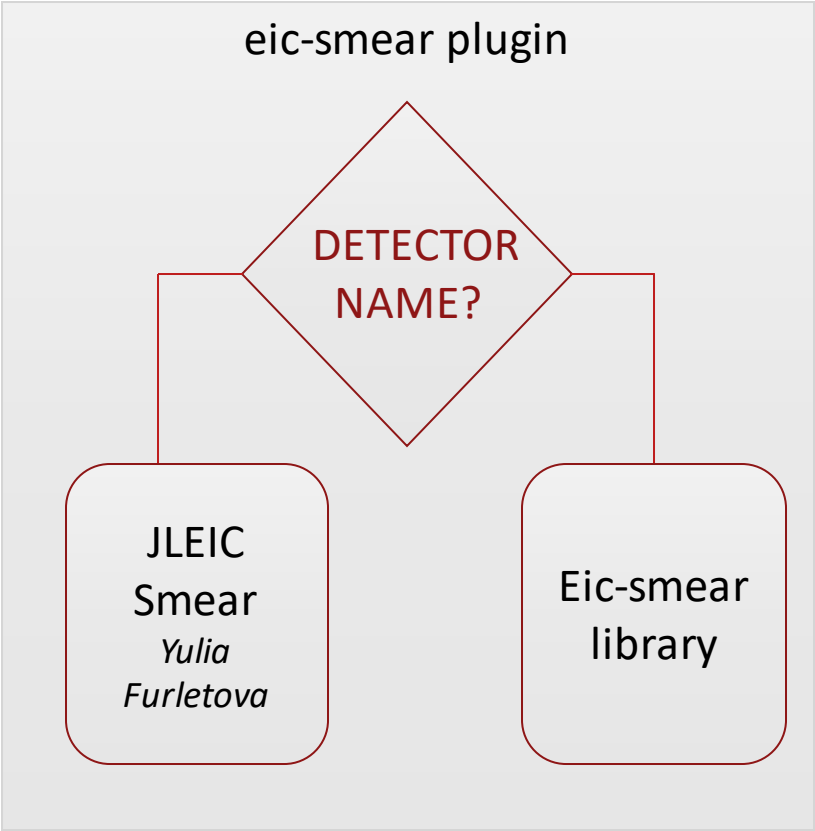
```
smear my_file.txt
```

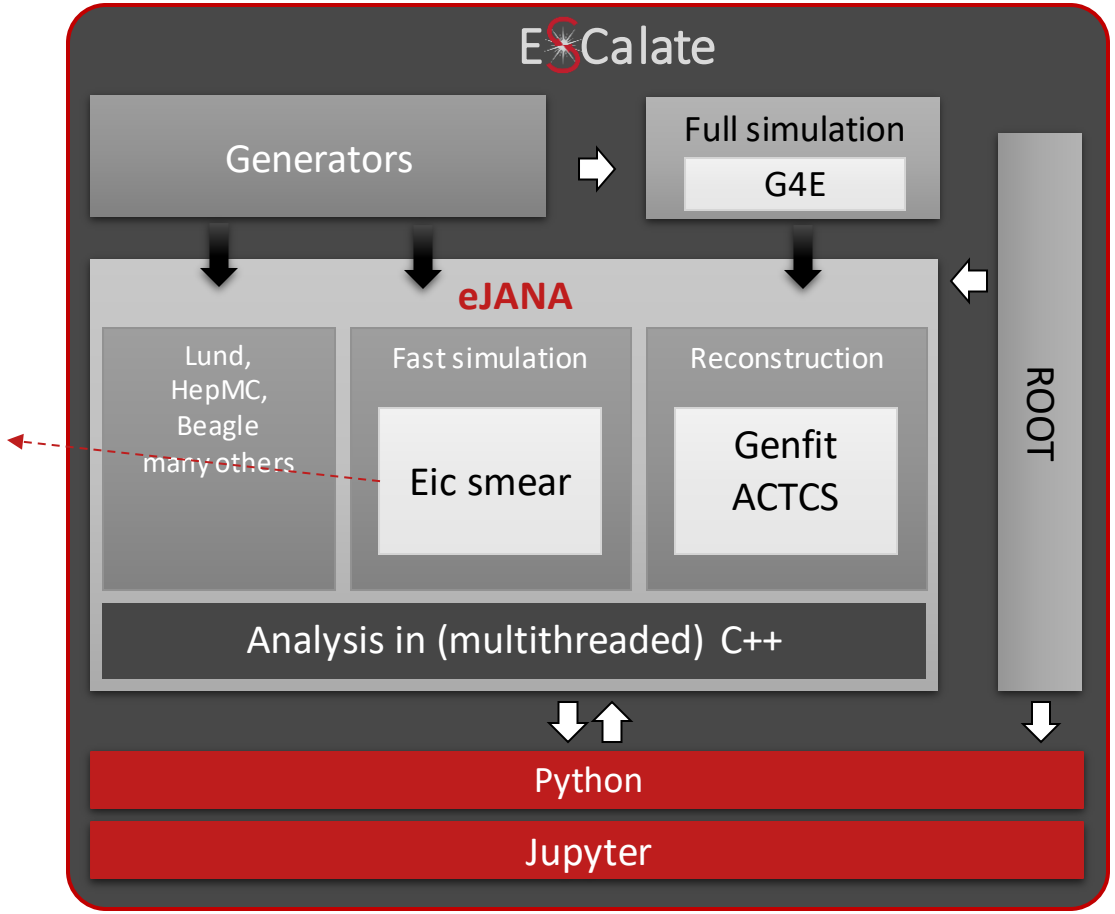
- Select handbook detector and process only 1000 events:

```
smear -d handbook -n 1000 my_file.txt
```

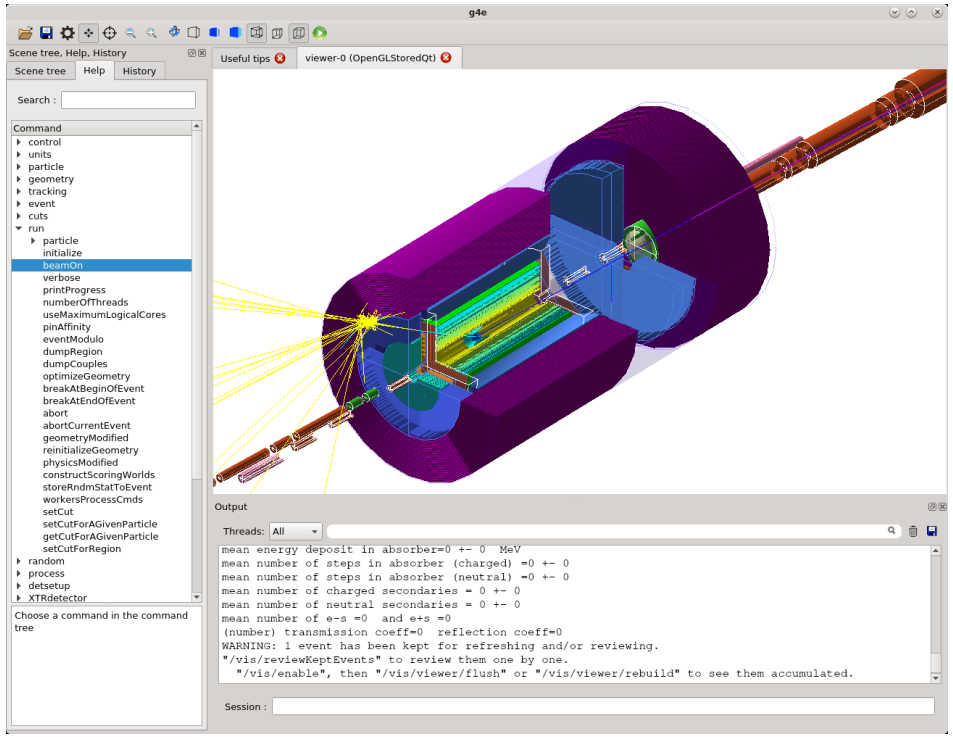
- Full documentation of the smear tool

Software stack for EIC simulations





G4E – Geant 4 EIC



https://gitlab.com/jlab-eic/g4e

- Standalone C++
- Multithreaded
- Geant4 application
- Various EIC MC file formats: Beagle, Pythia6, HEPMC - Pythia8, Herwig and others
- Integration with accelerator elements
- Infrastructure to import Geant4 detector geometry and simulation code

Grow with user input

Develop

Support

Workflow environment for EICUG

- **to use** (tools, documentation, support) **and**
- **to grow with user input** (direction, documentation, tools)

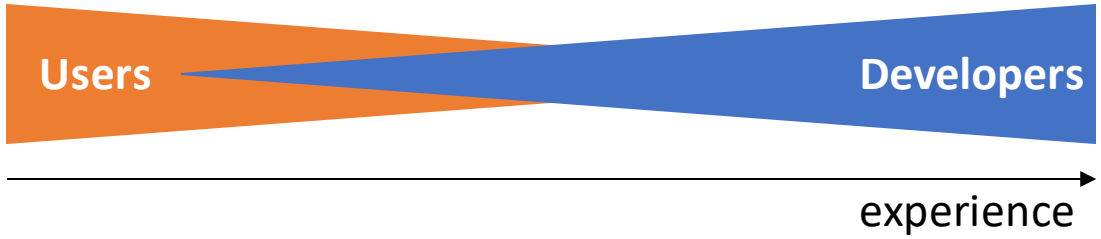
Involvement from EICUG

Support

Thank you!

The next tutorials will be on Monte Carlo Event Generators. The dates will be slightly adjusted due to the new dates for the EIC User Group Meeting (July 15-17).

eic.github.io



software-support@eicug.org

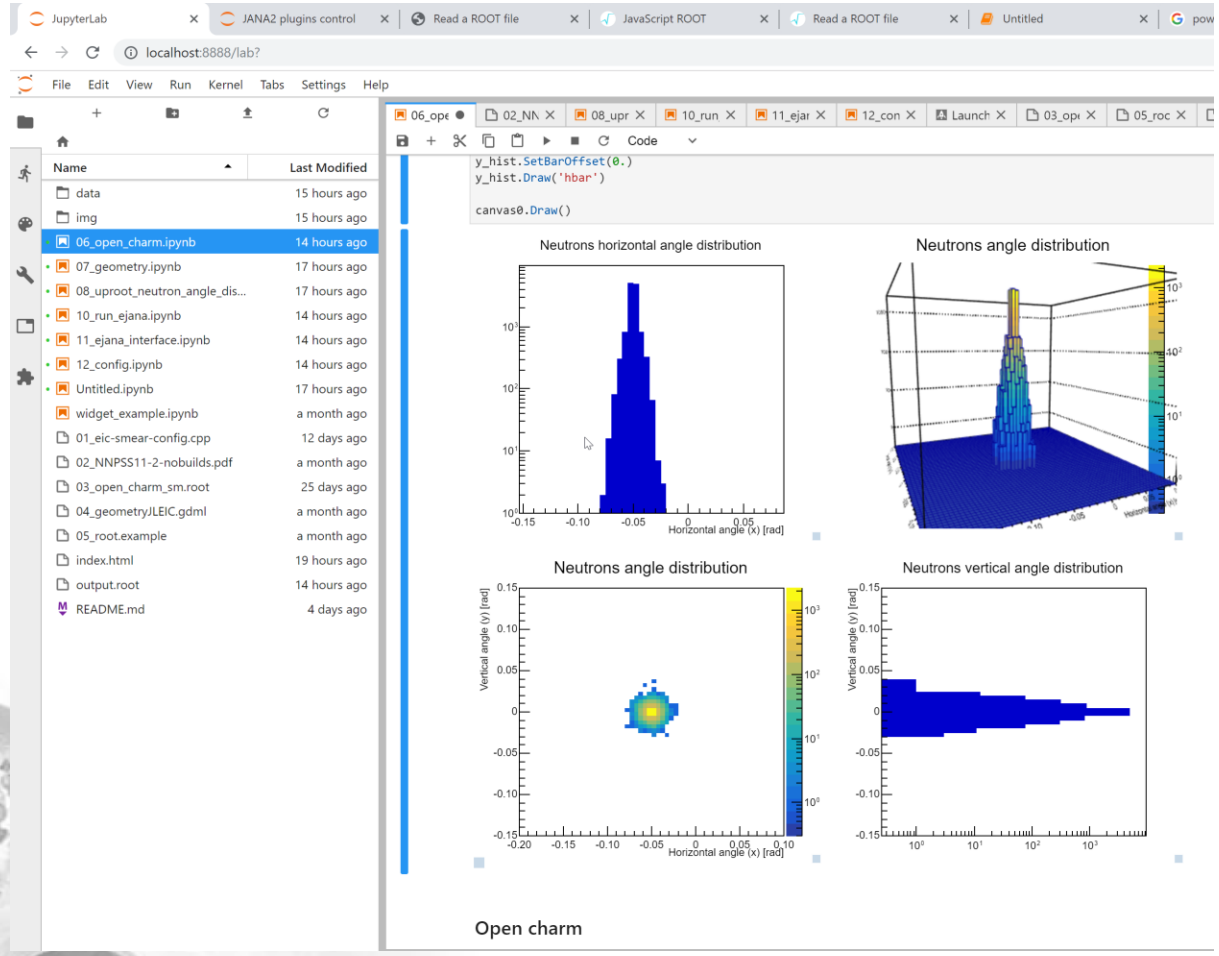
Mailing list (anyone can contact)
Google forum (for archive of support requests and start of knowledge base)

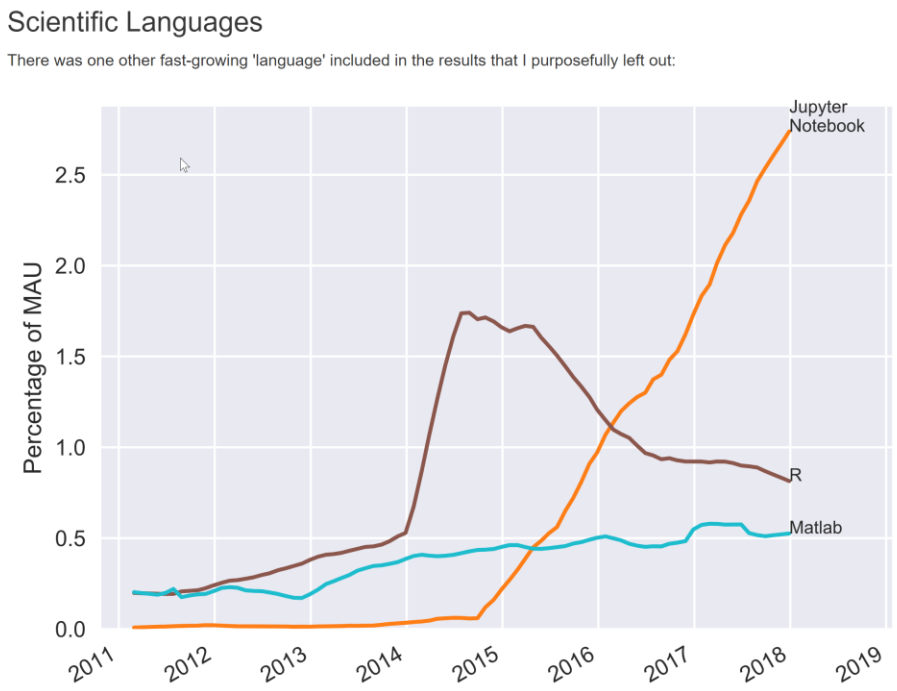
http://eicug.slack.com/

EICUG Slack workspace with software-support channel

BACKUP SLIDES

Jupyterlab new interface to Jupyter





Do we hide complexity?

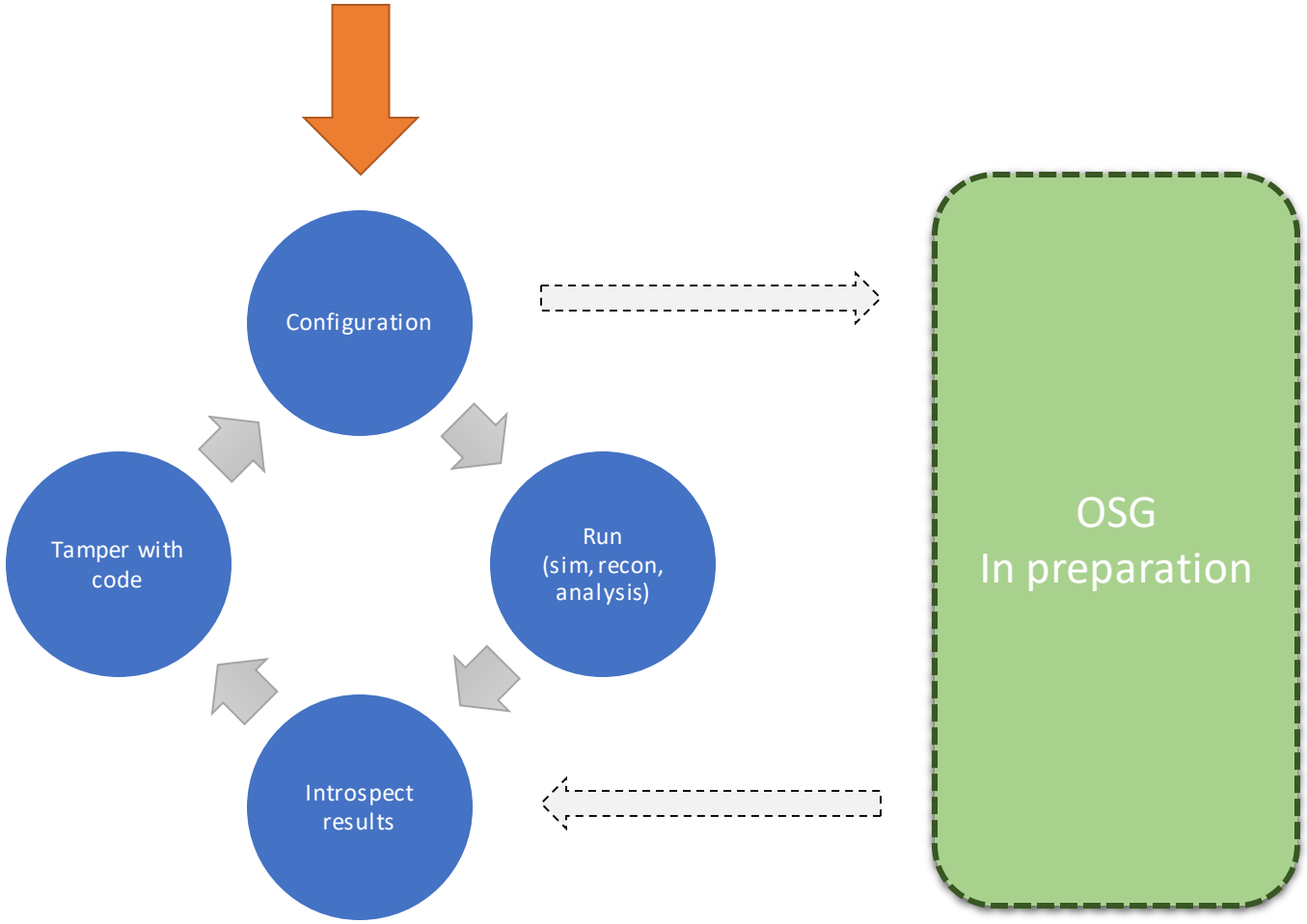
Jupyter lab, GUI,

Python, scripts, analysis

C++, eJANA, plugins

JANA, eic-smear, ROOT, Geant4

Run on Open Science Grid



Ways to interact G4E with the docker

1. JupyterLab (in browser)
2. noVNC (in browser)
3. Any VNC viewer
4. X11 (directly or through ssh)
5. Remote debugging

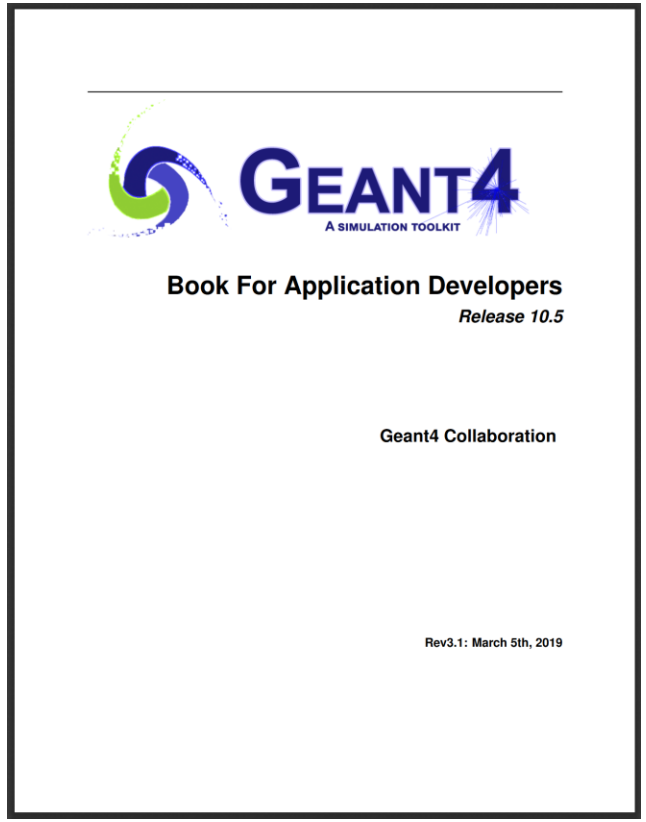
… or just install everything on your machine

https://gitlab.com/eic/ejpm

Geant4Eic (g4e)

https://gitlab.com/jlab-eic/g4e

- Keep close to raw Geant 4 (10.6)
- Small code base and fast compilation is good - KISS
- Users coding in “Geant4” paradigms
- Coding is ~~OK~~ GOOD
- Few interfaces are well defined and documented. E.g.
  - How to move in a detector
  - What is the output root file structure
  - etc.
- It is up to users what to do, but there are Recommendations to commit code back



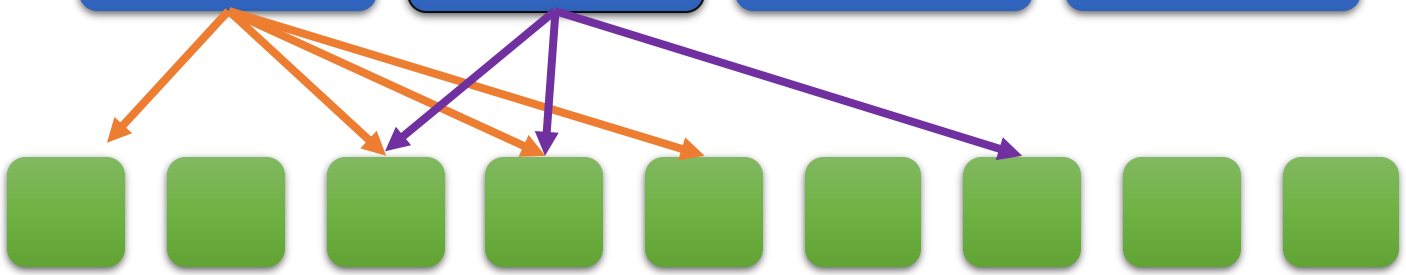
Main detectors – sub detectors

Beamlines

ERHIC | JLEIC | Future updates?

Master detectors

BEAST | JLEIC | ePhenix | Etc…



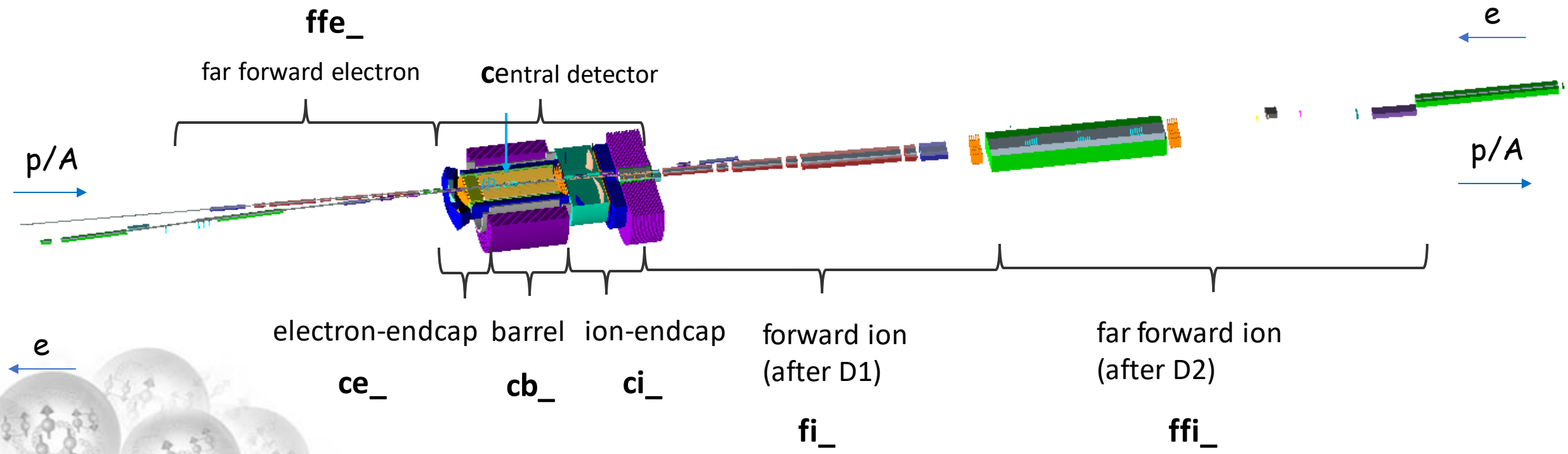
Subdetectors

Now 24 subdetectors

Naming convention

Names - <region>_<DETECTOR>_sub…

Examples:
cb_VTX = central barrel Vertex detector
ffe_LUMI = far forward electron Luminosity monitor



Naming sum up

1. Central Detector (c):
   - Barrel (cb) == Central Barrel
     - *Solenoid (cb_Solenoid)*
   - Electron endcap (ce) == Central Electron endcap
     - *GEM tracking (ce_GEM)*
   - Ion endcap (ci) == Central detector Ion endcap
     - GEM tracking (ci_GEM)

2. Forward ion (fi) direction area near D1 magnet:
   - Tracker detector1 (fi_TRKD1)

3. FarForward ion (ffi) direction area (near D2, D3 magnets)
   - ZeroDegree Calorimeter (ffi_ZDC)
   - Roman Pots (ffi_RPOTS)

4. Far forward electron (ffe) direction area
   - Low*Q2 tagger (ffe_LQ2)
   - Electron Polarimeter (ffe_CPOL)
   - Luminosity monitor (**ffe_LUMI**)

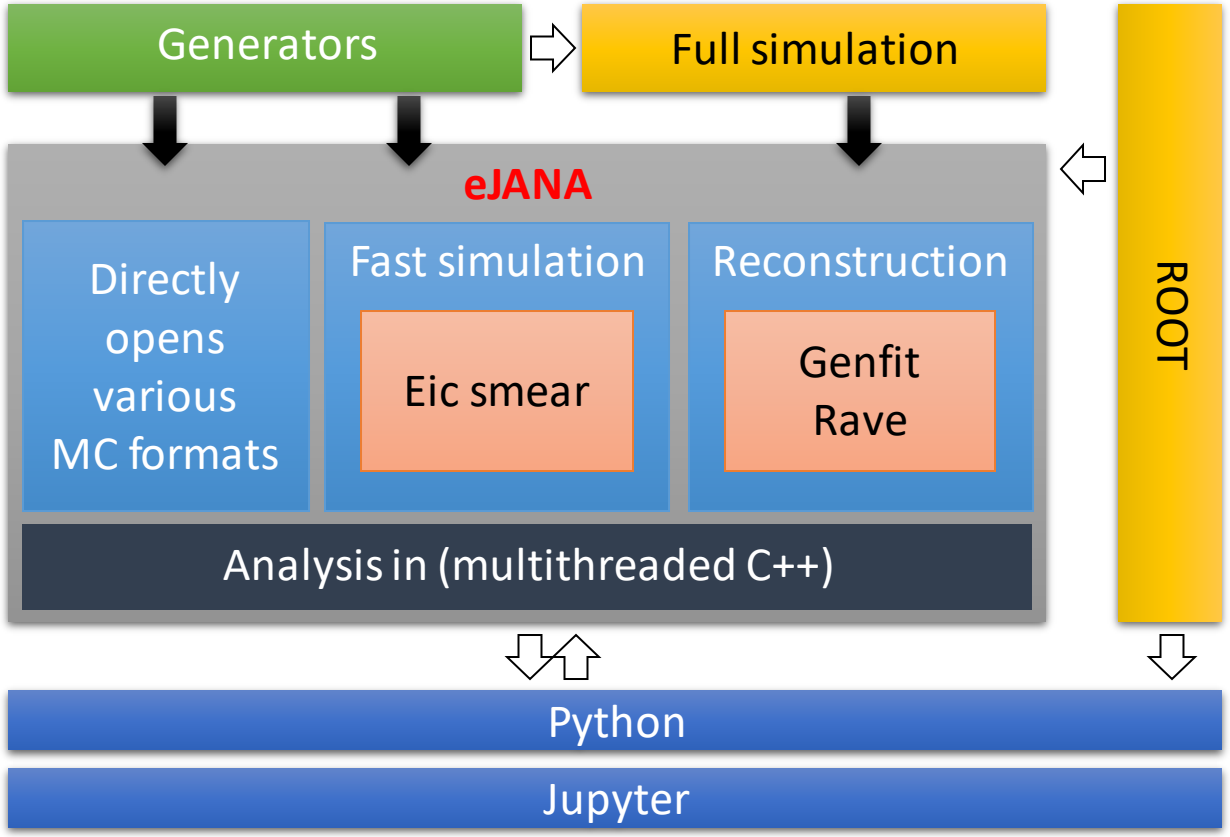
e^{JANA}

e^{JANA} **is JANA + plugins** for EIC data reconstruction and analysis

But also:

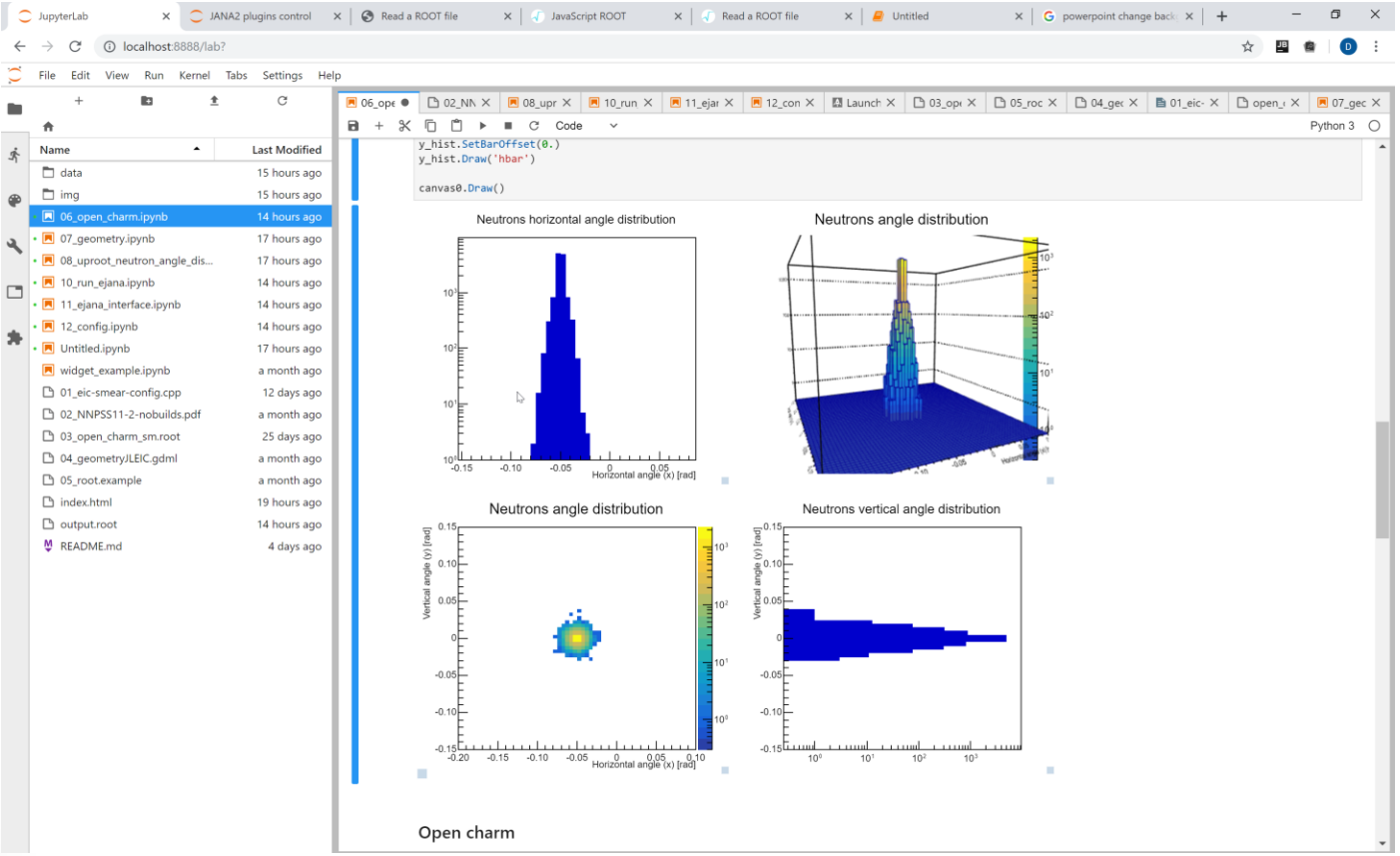
- tools to manage dependencies and run eJANA in different environments
- Integration with python and extensions to Jupyter Lab

  (ejpm, edock, pyjano, and others..)



e^{JANA} stands for EIC JANA

Jupyter lab, Jupyter notebooks, EPW, epic…



Transparency between layers

- JupyterLab -> Python/ROOT C++. Python -> Command line…

../data/beagle_eD.txt

[3]:
```
jana.run()
```

Total events processed: 10001 (~ 10.0 kevt)

▸ **Full log**

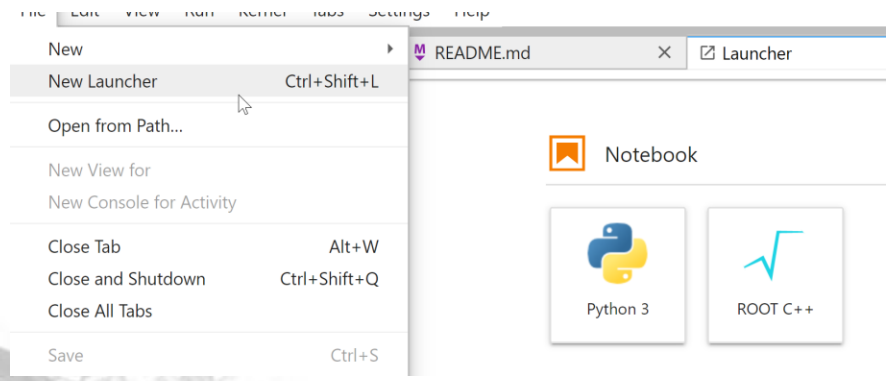
▾ **Run command**

```
ejana
-Pplugins=beagle_reader,vmeson,event_writer
-Pnthreads=1
-Pnevents=10000
-Poutput=beagle.root
../data/beagle_eD.txt
-Pjana:debug_plugin_loading=1
```

Working with ROOT and CLI

- Run docker with bash
  - docker run -it -p 8888:8888 eicdev/epic:latest **bash**
  - To run jupyter lab environment > **jlab**

- **Run ROOT + C++ in notebooks:**




```
[1]: auto file = TFile::Open("beagle.root");

[2]: auto hst = file->Get<TDirectory>("vmeson")->Get<TH1>("h2_XQ2_true_log");

[3]: auto c = new TCanvas("myCanvasName","The Canvas Title",800,600);
     hst->Draw("colz");
     c->Draw();
```