



{Docker, JupiterLab, Jana} software environment trial with EicRoot as a guinea pig

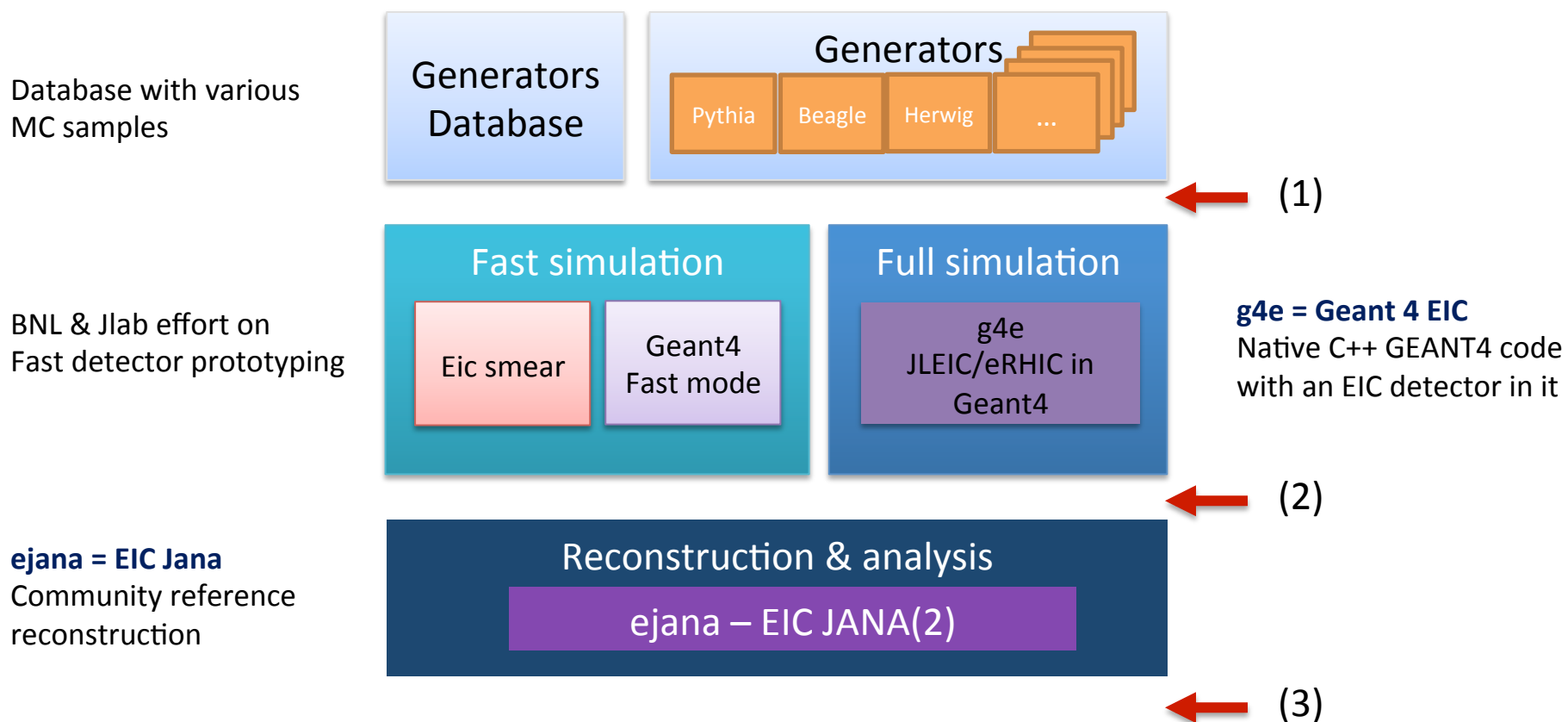
Alexander Kiselev

NPPS Group Meeting September, 13 2019

Outline

- Recap of the 08/23/2019 presentation
- Jana2: first contact
- EicRoot as a Jana plugin
- Interactive JupyterLab session
- What comes next?

“EIC container” functionality overview



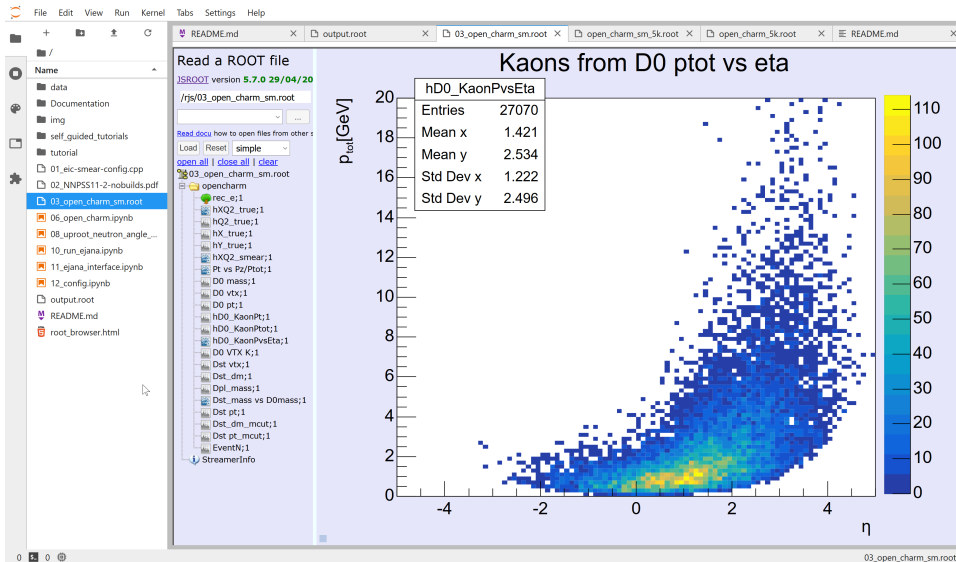
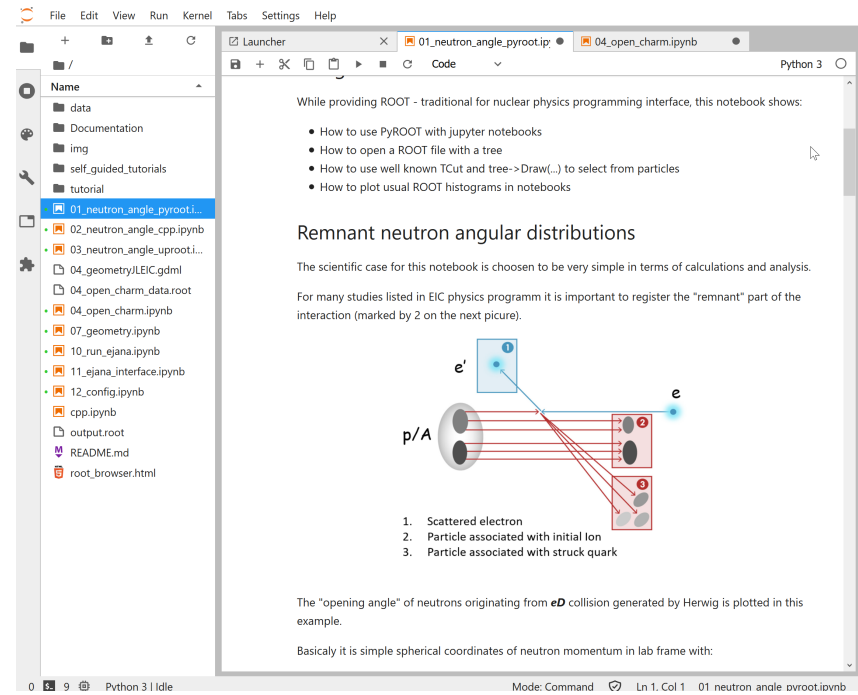
- (1) MC events
- (2) Digitized hits + magnetic field + material distribution
- (3) Reconstructed events

-> user access (with graphics) either directly or through SSH or Web interface

[See D.Romanov: talk at the EIC software meeting 07/10/2019](#)

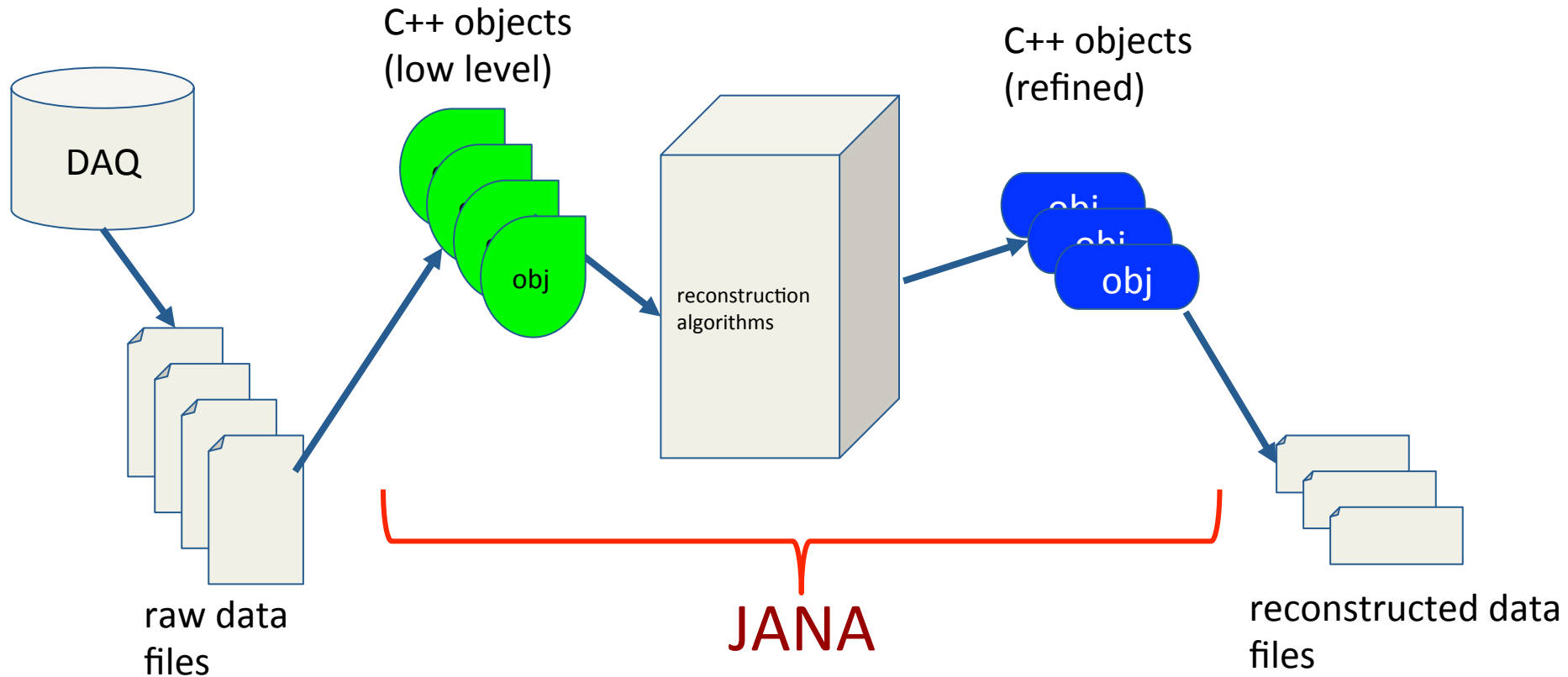
JupyterLab Web interface to all this

Wiki: *Jupyter Notebook is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format.*



- Cloud based collaborative workspace
- The medium for studies, reports, analysis
- The bridge between modern Data Science and traditional Nuclear Physics methods

Jana(2) software framework



- Provide mechanism for many physicists to contribute reconstruction codes to the “shared pool”
- Implement multi-threading efficiently & external to the contributed codes
- Provide common mechanisms for accessing job configuration, calibrations, etc

Jana(2) software framework

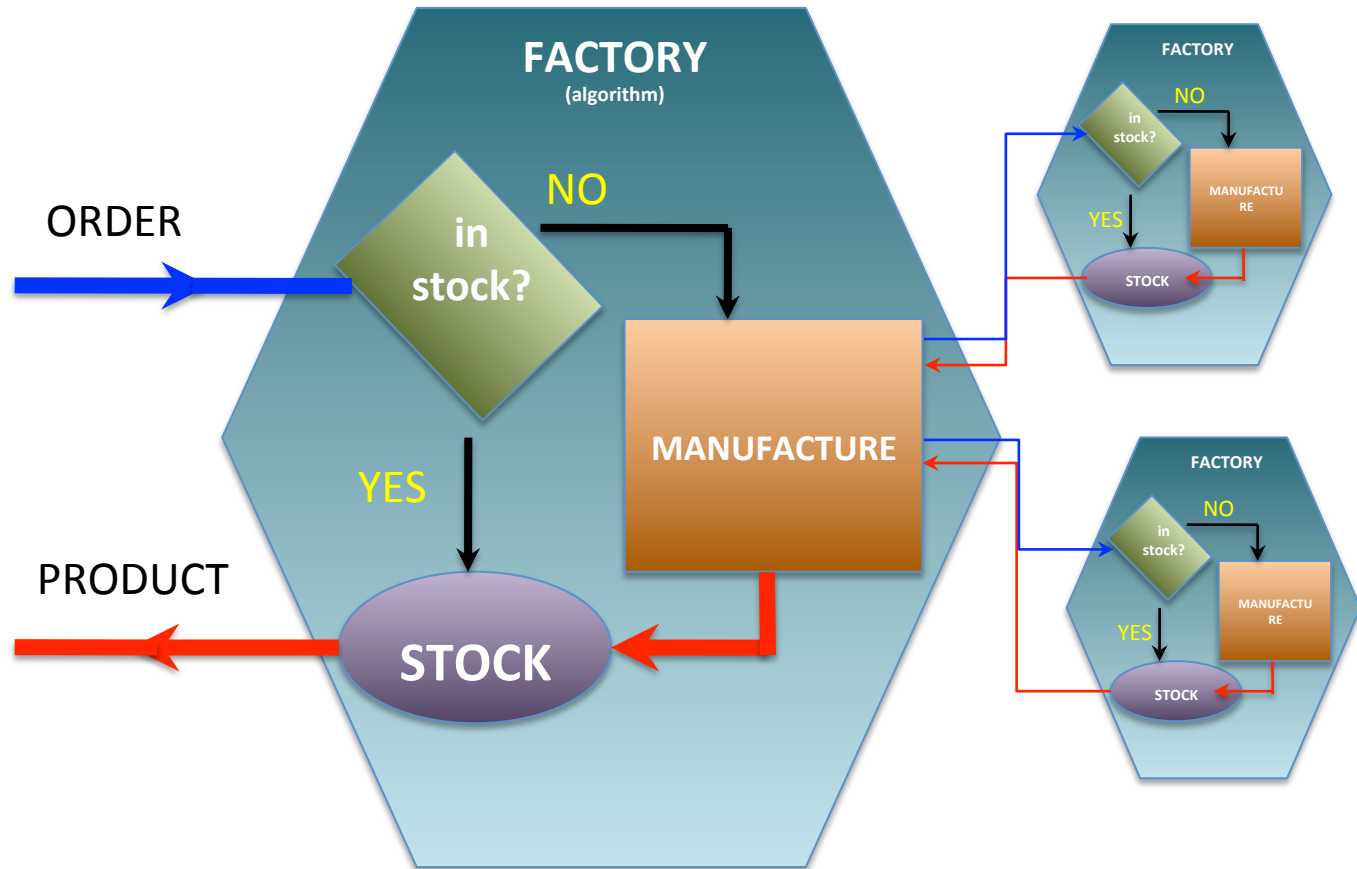
Built-in features	Implementation
Multithreading	Yes
ROOT interface	No
GEANT interface	No
“Checkpointing”	No
I/O formats	Not specified
Steering	Configurable Jana executable
User codes	Plugins (shared libraries)
Event processing scheme	In parallel, top-down synchronous
In-memory event components	Tagged class instances (or vectors of those)

■ General impressions:

- modern, (much) higher than average quality C++ code
- pretty functional within its specs; in active development; has room for improvements

See e.g. a simple ROOT-for-Jana study: <https://gitlab.com/ESC/jreader>

Jana(2): factory model



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

Stock = Don't do it twice

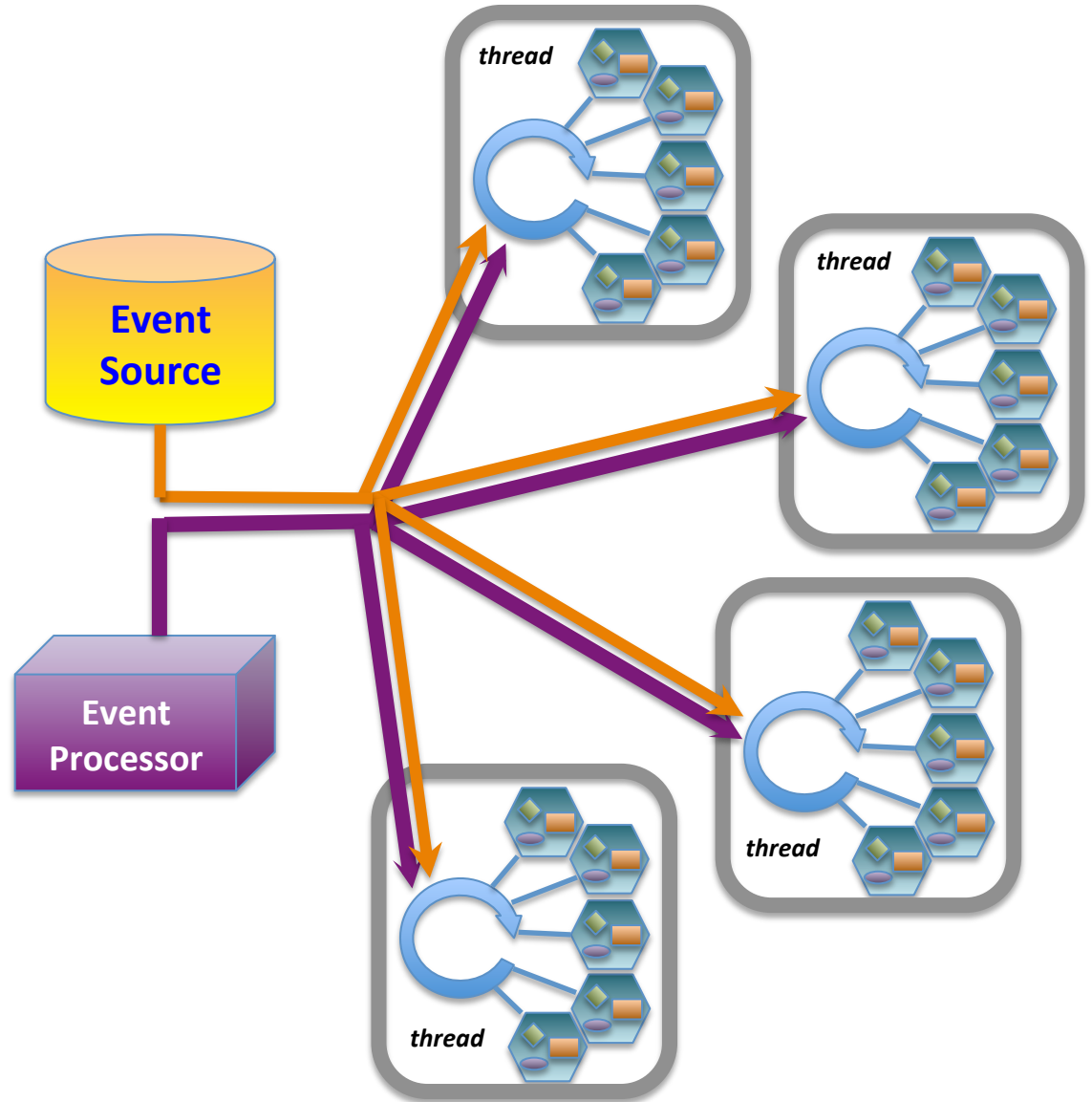
**Conservation
of CPU cycles!**

Jana(2): multi-threading

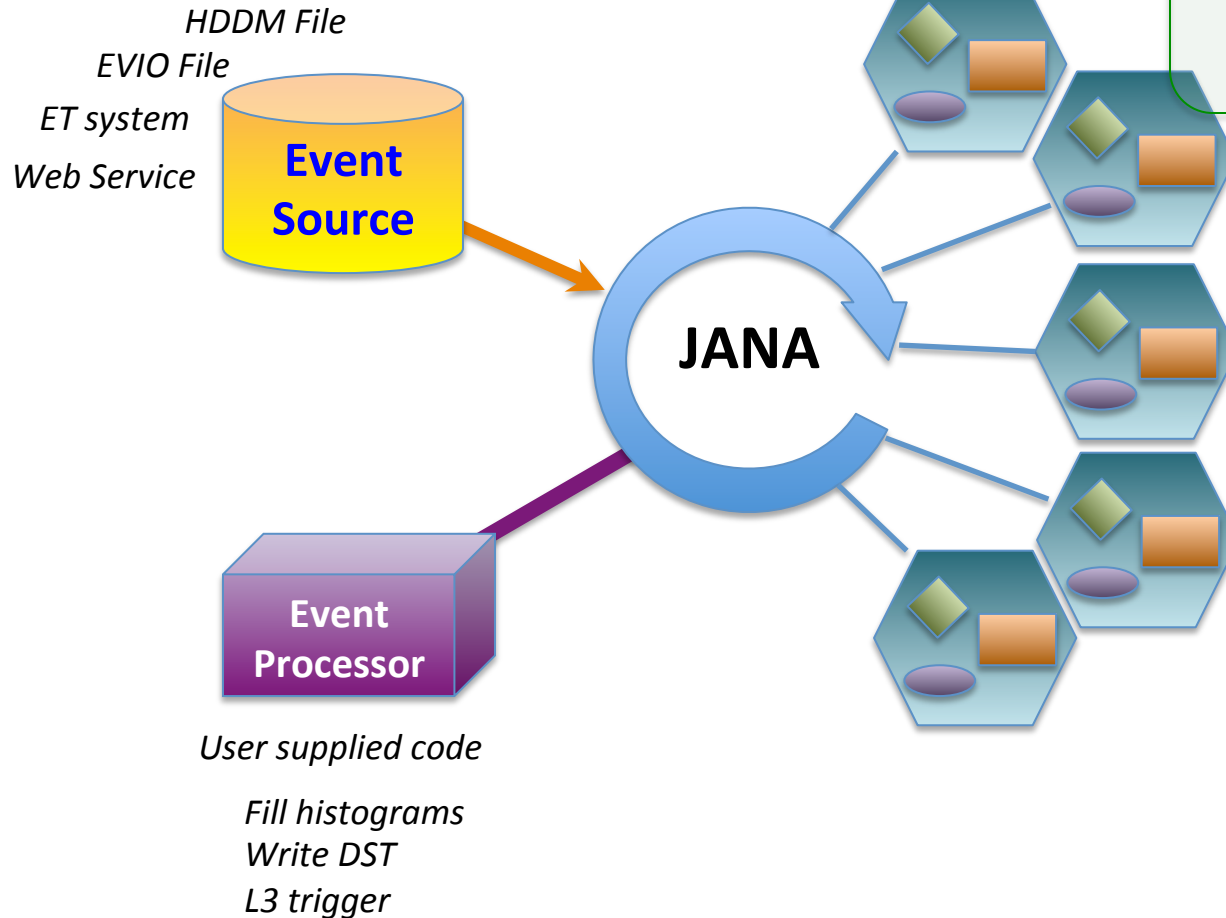
○ Each thread has a complete set of factories making it capable of completely reconstructing a single event

○ Factories only work with other factories in the same thread eliminating the need for expensive mutex locking within the factories

○ All events are seen by all Event Processors (multiple processors can exist in a program)



Jana(2): event reconstruction scheme



Framework has a layer that directs object requests to the factory that completes it

Multiple algorithms (factories) may exist in the same program that produce the same type of data objects

This allows the framework to easily redirect requests to alternate algorithms specified by the user at run time

EicRoot data processing flow

- No executable (steering through ROOT macro scripts)



- ROOT at a core of the framework ...
- ... used also for I/O and data exchange (TClonesArray)
- Built-in interface to GEANT(s)
- Each module is a collection of shared libraries

[See AK: talk at the EIC software meeting 07/10/2019](#)

Outline of the integration exercise

- Use JLAB Jana+Jupyter Docker container base image
- Add other missing stuff like *sshd*, *editors*, etc
- Rework EicRoot installation scripts (no fairsoft bundle, etc)
- Integrate Jana into CMakeLists.txt where appropriate
- Eventually build EicRoot as a Jana plug-in
- Play around with the JupyterLab interface

-> interactive presentation follows