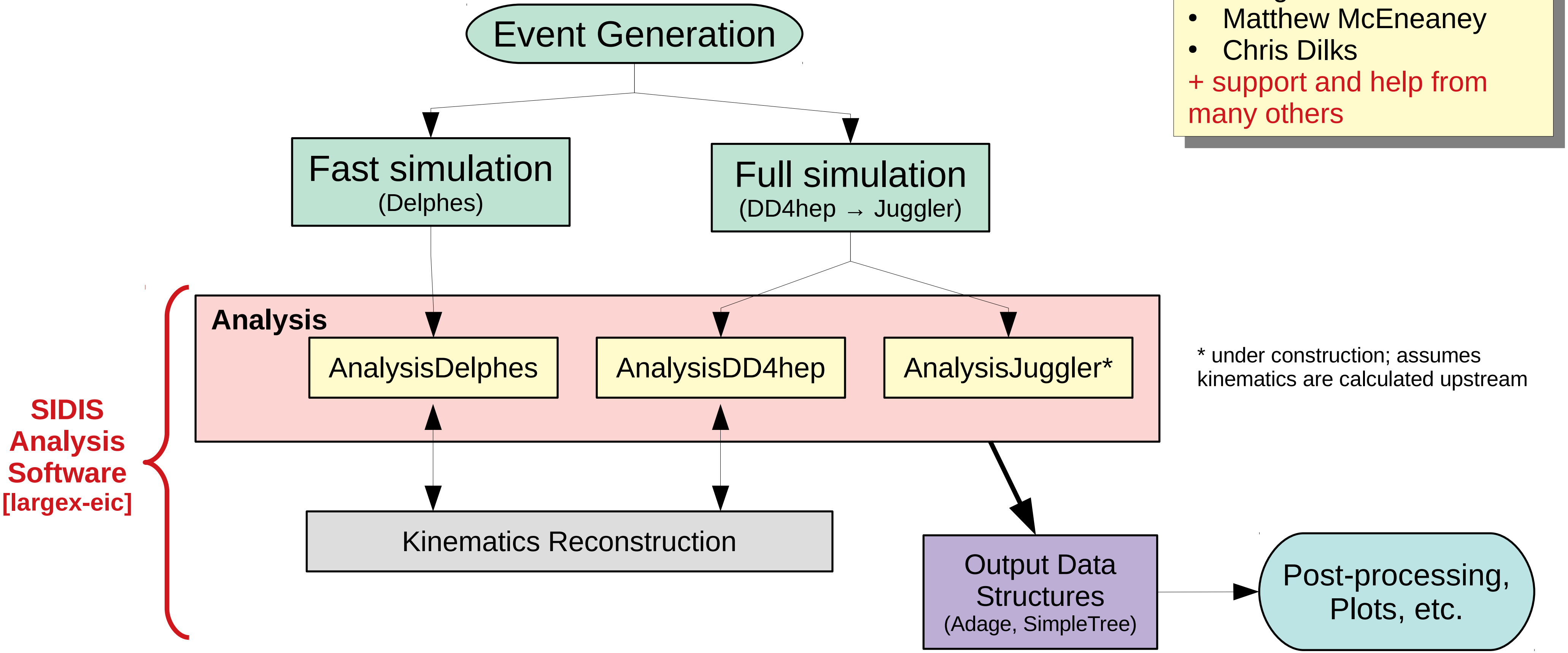
ATHENA SIDIS Analysis Software

https://github.com/c-dilks/largex-eic

Developers

- Duane Byer
- Connor Pecar
- Sanghwa Park
- Matthew McEneaney
- Chris Dilks

+ support and help from many others



Event Generation

Fast simulation
(Delphes)

Full simulation
(DD4hep → Juggler)

Analysis

AnalysisDelphes

AnalysisDD4hep

AnalysisJuggler*

Kinematics Reconstruction

Output Data Structures
(Adage, SimpleTree)

Post-processing, Plots, etc.

SIDIS Analysis Software [largex-eic]

* under construction; assumes kinematics are calculated upstream

Software Design Principles adopted from ATHENA Software Group

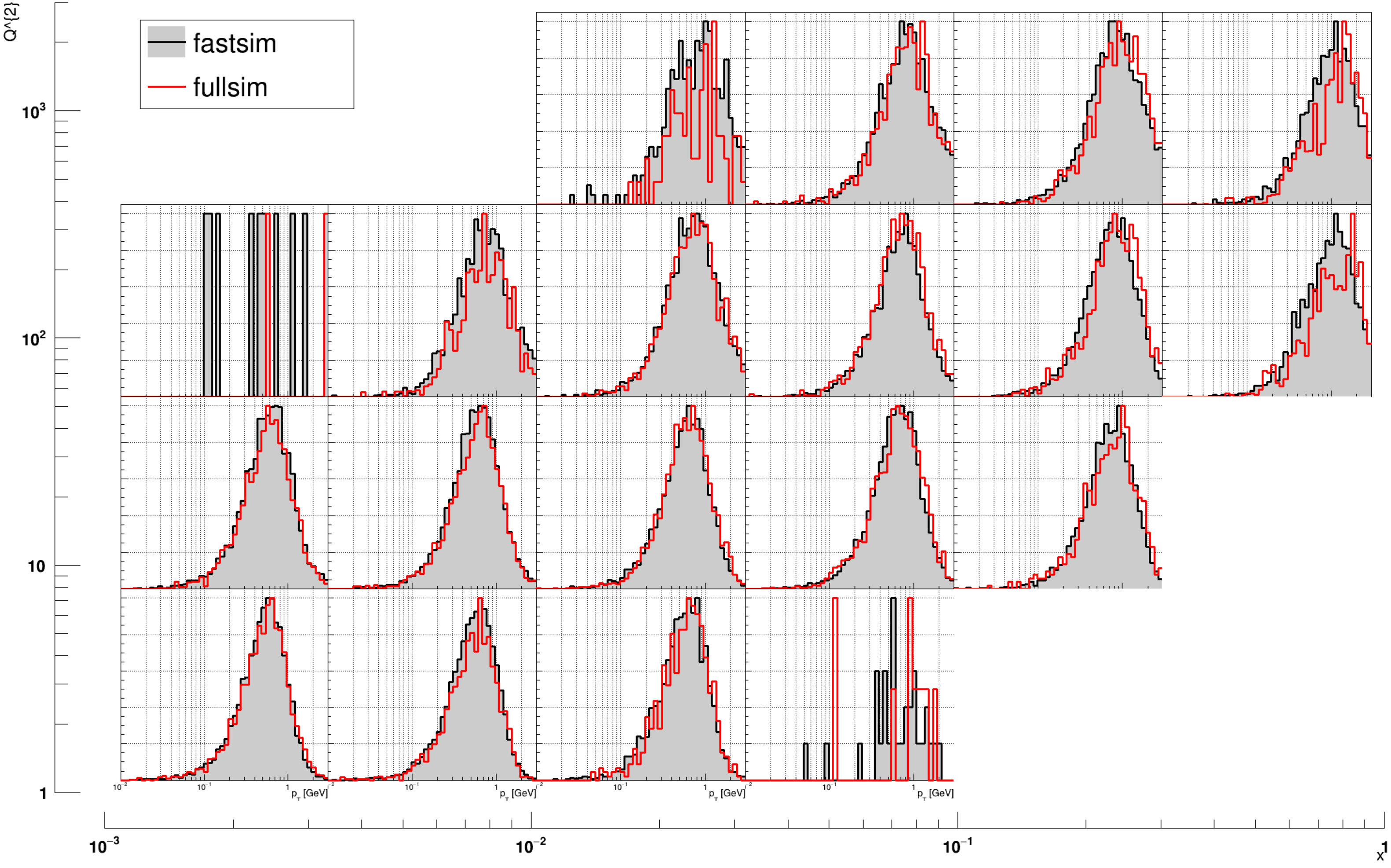
- **Modularity**
  - One “task” = one “module”
  - SIDIS SW itself is a module, reading output from fast/fullsim
  - Adaptable to upstream data structure changes → Analysis sub-classes
  - Adaptable to downstream needs → Edit existing or add new data structures

- **Continuous Integration**
  - Support development / testing
  - Automate generation of benchmark plots
  - Track evolution of any plot as development proceeds

- **Containerization**
  - Singularity / Docker image available, including dependencies such as Delphes and ROOT
  - Entry point for new contributors
  - Support CI

- **Version control**
  - Trunk-based development → pull requests and code reviews

Example benchmark plot: pion p_T for fast and full simulations, in (x,Q^2) bins



Issues / Lessons / Ideas

- **Fast simulation storage**
  - Could not use S3 to store Delphes files (performance limits + S3 is primarily for full simulations)
  - Workarounds:
    - For high statistics: JLab storage – limited accessibility
    - Run Delphes on S3 hepmc files – adequate for low stats
    - Considered other short-term cloud storage options, but JLab storage was sufficient for the proposal

- **Automated access to S3**
  - MinIO Client and TFile::Open provides access, but a wrapper would be useful
  - We developed a set of wrapper scripts for SIDIS SW, where the user supplies:
    - Full simulation version (e.g., DeathValley-1.0), or which Delphes card to use
    - Beam energy (e.g., 18x275)
    - How much data (e.g., number of files per Q^2 minimum)
    - Decide whether to stream from S3 or download to local disk
  - Such a wrapper would be dependent on Working Group needs → beyond scope of SW group
    - Similarities in data organization between Working Groups? → generalized wrapper could be developed
    - Maybe all we need is better documentation of what’s available and how it was produced

Support for the Future

- **Upstream Integration:** migrate to EICweb (gitlab)
  1) **Connect to upstream CI pipelines**
     - Example Scenario:
       - A change in detector design is being considered
       - Proposed change triggers ATHENA detector CI pipelines and benchmarks
       - SIDIS analysis SW pipeline could also be triggered, providing immediate feedback of the effect of the proposed design change
  2) **More modularization**
     - Adage “backend” data structure is general purpose → should be a separate module (repository)
     - A new backend could make more use of POD structures, or add POD support to Adage
  3) **Generalization**
     - We don’t have to limit ourselves to SIDIS
     - Already we have (some) support for jets
     - Support broader needs of the collaboration
     - Name change, since “largex-eic” is historical

Contributions are welcome!

backup

Data Structures in SIDIS analysis SW

- **<u>Simple Tree</u>** – flat TTree, useful for quick tests etc.

- **<u>Adage</u>** – Analysis in a Directed Acyclic Graph Environment
  - Store data with arbitrary multi-dimensional binning and cuts
  - Store lambdas, executable by graph traversal high-order functions
    → no need for nested for loops!
  - Prototype design, could use more testing and development

- ***It’s fairly straightforward to add your own data structure***
  - Existing data structures may not suit our future needs
  - Future development idea: data structure “plugins”