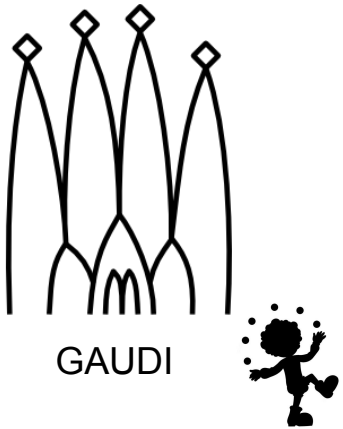
Thoughts about frameworks for the EIC



or

Sylvester Joosten
CompSW Meeting, July 6, 2022

Considerations	Gaudi	Jana2
Major requirements	✓ Meets requirements	✓ Meets requirements
Short term work needed	✓ Working solution today	⭕ Needs major upfront investment to meet October deadline
Training prospects	✓ Training of user base could start immediately	⭕ Training will have to wait for core infrastructure to be available
Documentation	⭕ Gaudi documentation 10+ years outdated ⭕ Critical documents behind CERN login	⭕ Documentation needs EIC-specific work too (but overcomable).
Project specs	✓ Large development team ⭕ Development requires CERN account ⭕ No strong interest from developers in EIC (unresponsive to emails)	⭕ Smaller project (only few devs, all at a single lab) ✓ Could shift to open-source model? ✓ Core developers invested in EIC
Support	⭕ Volunteer-only support from a small development team ○ Best pathway for support through Key4hep	✓ Dedicated (formal?) support from JLab

Jana2 can work but requires *significant* work by proponents to meet October deadline. Meeting this deadline is important for buy-in from the collaboration in the entire modern software stack.

Possible path forward with Jana2

- Need strict timeline for Jana2 short-term development.
  - Data model integration and DD4hep integration need to happen ASAP.
- Needs/wants:
  - style-guide for reconstruction algorithms and services for the framework, focus on composable algorithms.
  - Interoperability with Key4hep?
  - Need path to enable e.g. SYCL support (C++17/CLANG)
    - CI test pipelines with different compilers?
  - Dedicated EIC documentation
- Needs formal engagement from JLab group
  - FTEs for initial implementation of algorithms
  - Commitment to become open source project (open development model)
  - EIC will be biggest customer of Jana2!
  - Support all system (don’t assume JLab computing model, but also e.g. HPC nodes on exascale machines, …)

But, we need to do things today (or yesterday!)

- New developments for DD4hep+Gaudi stack still needed (cannot wait for October).
  - E.g. to enable background studies, which are required ASAP
  - Should be algorithmic in nature (easy to translate), and minor.
  - Larger endeavors should only be undertaking if the pathway to the Jana2 environment is clearly defined.
- Impact on users:
  - End-users should be fine, as the output format (driven by Podio) will be identical (no retraining users on analysis).
  - Need to deal with lowered incentive to write new algorithms while “waiting” for Jana2.
  - Fallback in case we don’t meet the timelines is to continue using Gaudi.
  - Good to start training ASAP, as soon as critical services are in place.
- Expect Jana2 to reach feature parity with Gaudi by October

Possible timeline

Today: ● simulations either in fun4all or DD4hep + Gaudi (Juggler)
Tomorrow: ● Short-term simulation needs can still be served by DD4hep + Gaudi (Juggler). ● Start implementation of Podio and DD4hep services in Jana2, then move to algos ● Stop major new developments in Juggler unless there is a clear path to move to Jana2
August: ● first training sessions on the DD4hep + Jana2 stack?
October: ● full switch to Jana2 (fallback keep using Gaudi)

We are taking on risk by switching from Gaudi to Jana2,
but I think it can be done if there is strong desire.