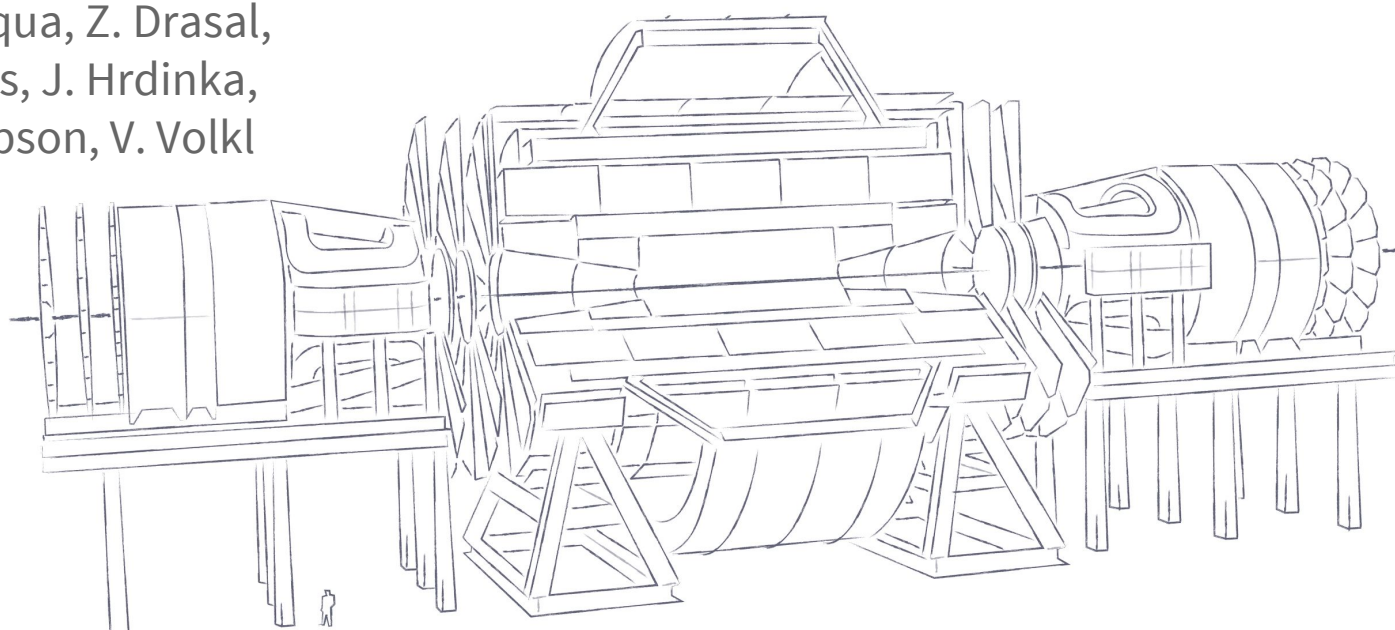
The FCC software
how to keep SW experiment independent

A. Zaborowska
C. Bernet, A. Dell'Acqua, Z. Drasal,
B. Hegner, C. Helsens, J. Hrdinka,
J. Lingemann, A. Robson, V. Volkl



The FCC design study

A conceptual design for accelerators and detectors

- Different requirements ee vs pp
- Support physics and detector studies
  - Detector concepts: Moving targets
  - Both fast and full simulation essential
- Challenging backgrounds (computing and physics-wise):
  - Pile-up for pp: 1000 events
  - Beam-background for ee

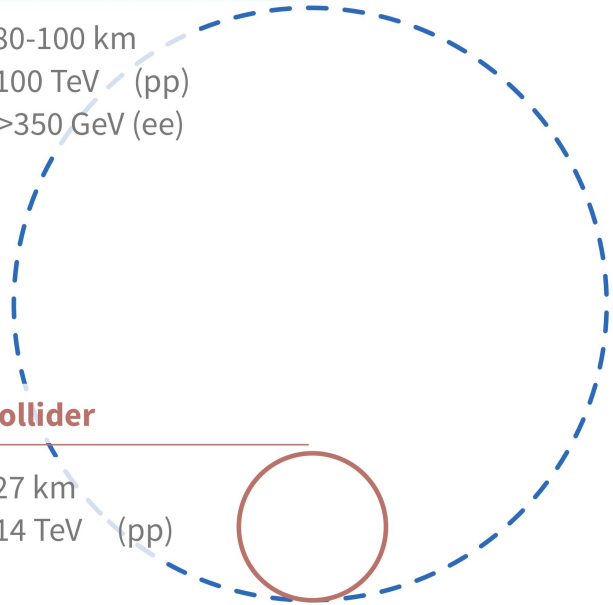
One software stack to support them all

Future Circular Collider

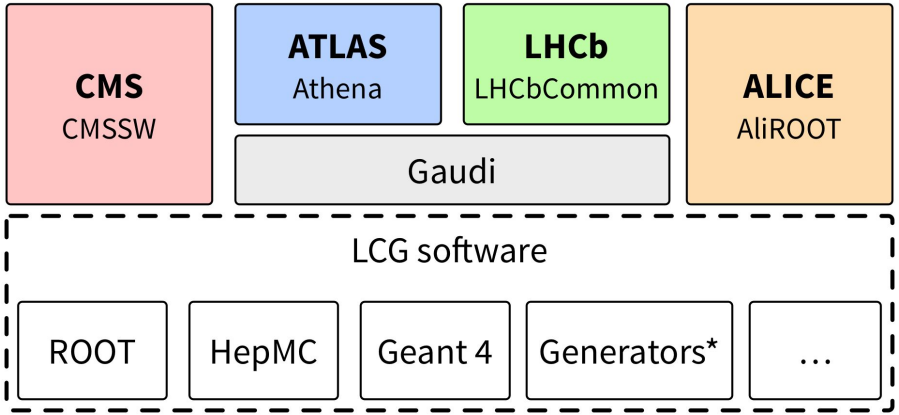
Circumference: 80-100 km
Energy: 100 TeV (pp)
>350 GeV (ee)

Large Hadron Collider

Circumference: 27 km
Energy: 14 TeV (pp)



Contrasting with the LHC experiment approach



Each experiment has dedicated software:

- Effort duplication
- LHCb and ATLAS started to collaborate
  - Gaudi as underlying framework
  - Also used in FCC software

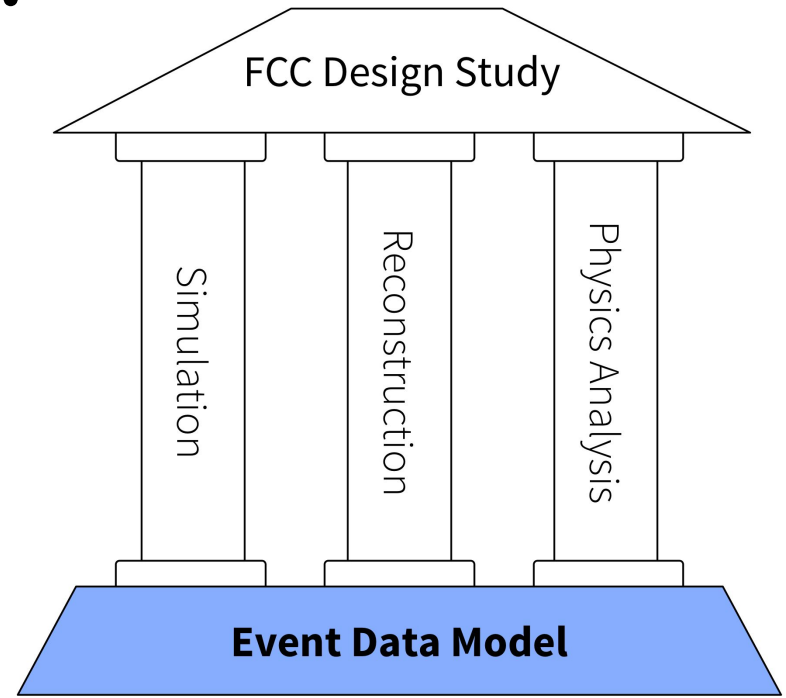
Is there a fundamental reason not to collaborate and share?

We decided to try!

*too many to list them all

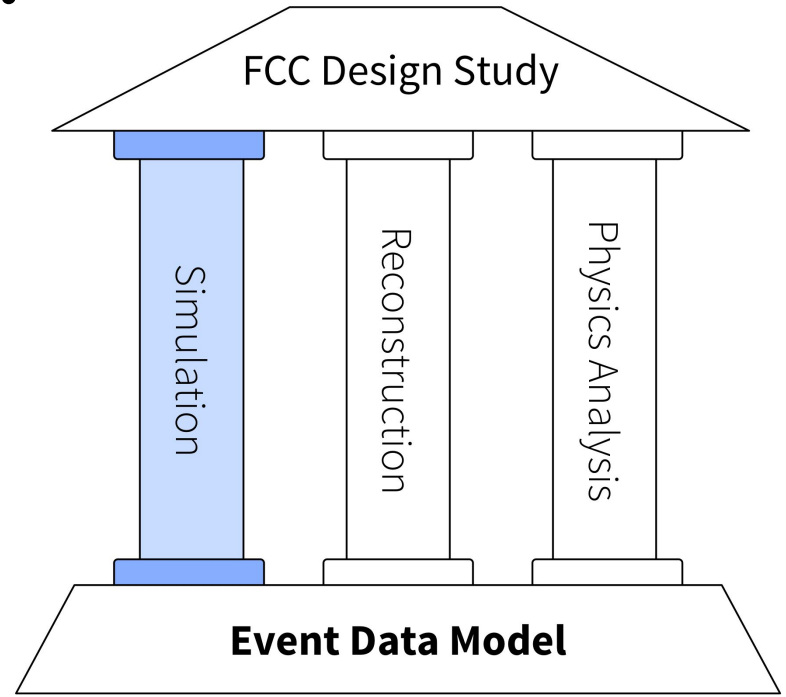
What are the ingredients?

- Flexible event data model
- Flexible detector description
  - Detector concepts are moving targets



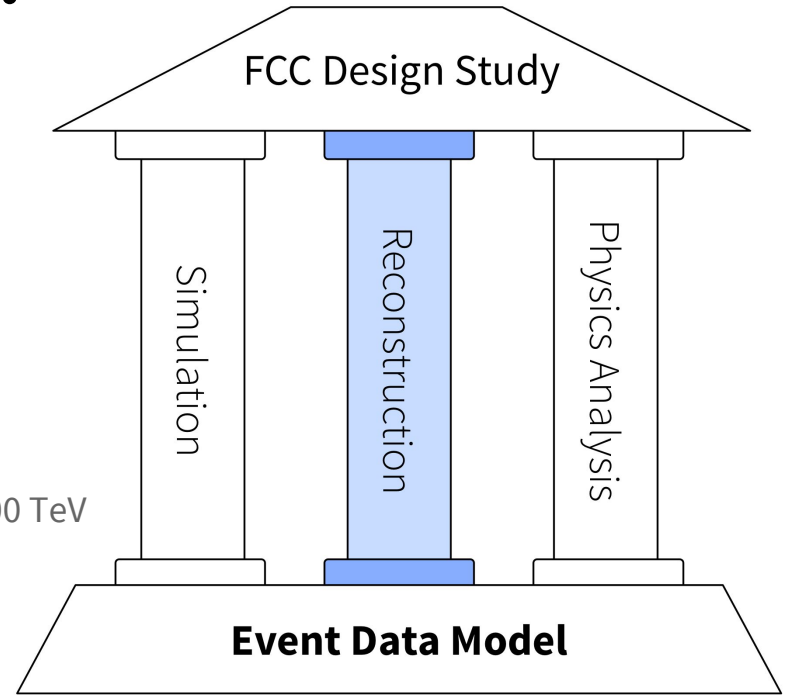
What are the ingredients?

- Flexible event data model
- Flexible detector description
  - Detector concepts are moving targets
- Simulation
  - Full simulation for detector studies
  - Fast simulation for physics studies



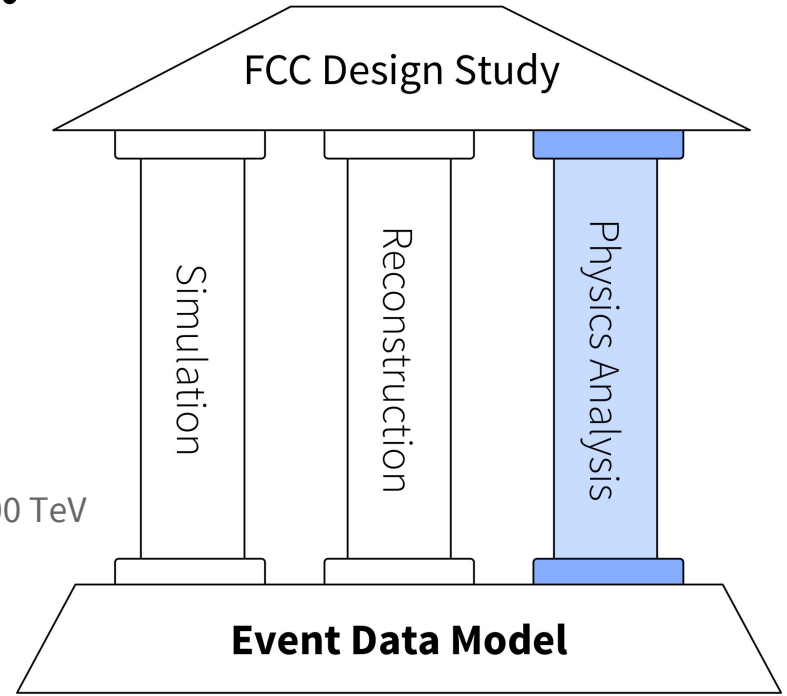
What are the ingredients?

- Flexible event data model
- Flexible detector description
  - Detector concepts are moving targets
- Simulation
  - Full simulation for detector studies
  - Fast simulation for physics studies
- Reconstruction:
  - FCC-hh: Extreme pile-up, extrapolation to 100 TeV
  - FCC-ee: Achieve the best possible precision



What are the ingredients?

- Flexible event data model
- Flexible detector description
  - Detector concepts are moving targets
- Simulation
  - Full simulation for detector studies
  - Fast simulation for physics studies
- Reconstruction:
  - FCC-hh: Extreme pile-up, extrapolation to 100 TeV
  - FCC-ee: Achieve the best possible precision
- Physics analysis
  - Allow use outside of framework
  - Python flexibility & C++ performance



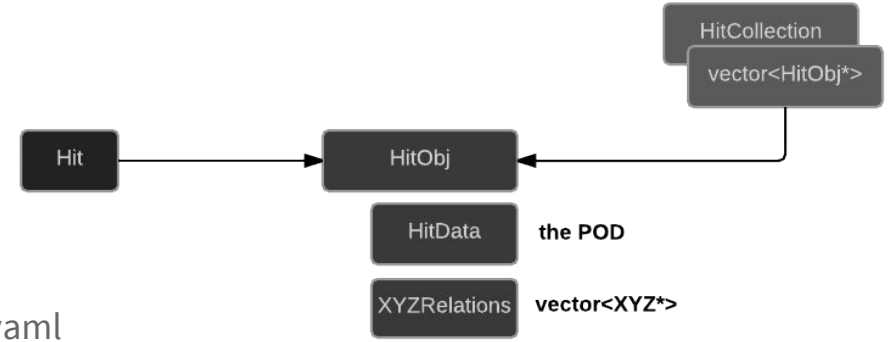
Event data model

Reviewing existing experiment solutions:

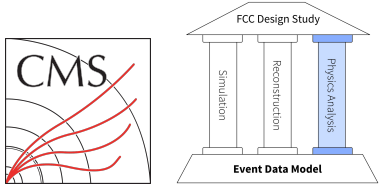
- Most: Complex inheritance and polymorphism
  - Problematic: data layout becomes more important
- Users: Not comfortable using them
  - Final analysis: Uses ntuples (and python)

Decided to invest here:
Plain **O**ld **D**ata **I**nput **O**utput (PODIO)

- Focus on reusability
  - All code generated: Data types described in yaml
  - Interest expressed by LHCb and ILC
- Designed with parallelism in mind
- Python and C++ supporting EDM on equal footing

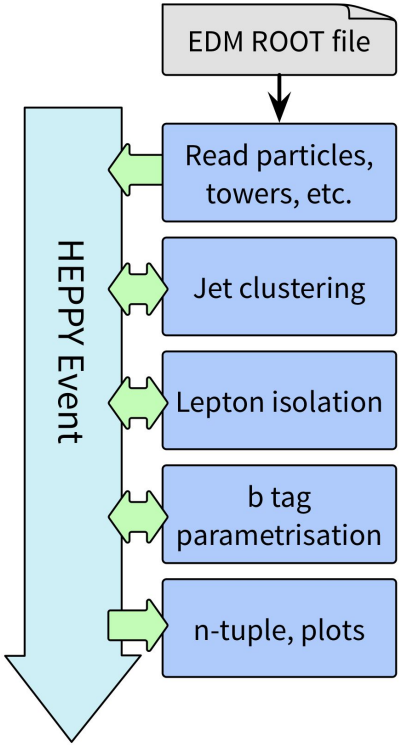


The Analysis Front-End



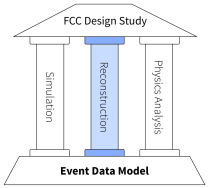
Python-based analysis: **HEPPY**

- Highly configurable, encourages code reuse
- Includes PAPAS simulation
  - Parametric simulation + particle flow reco (used in FCC-ee)
- Python: Testing ideas and prototyping algorithms
- Gradual port of code from Python to C++
  - Ported functionality still available via ROOT-based bindings



Balancing development agility vs. code performance

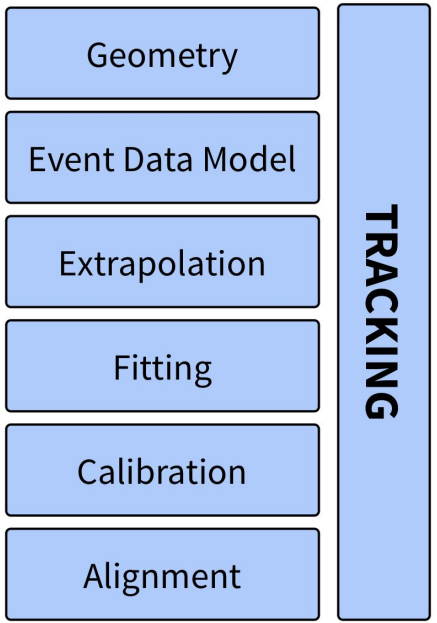
A Common Tracking Software



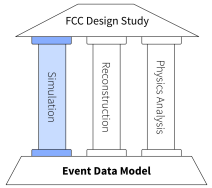
Extracting ATLAS tracking software into a standalone toolkit:

- Simplified geometry description optimised for tracking
  - Plugins available for ATLAS, DD4hep and TGeo
- Internally use simple EDM
- All necessary tools included
- Addon: Alternative fast simulation of inner detector
- Completely independent of experiment software

Solution that offers all functionality that is needed.



Generation and Simulation

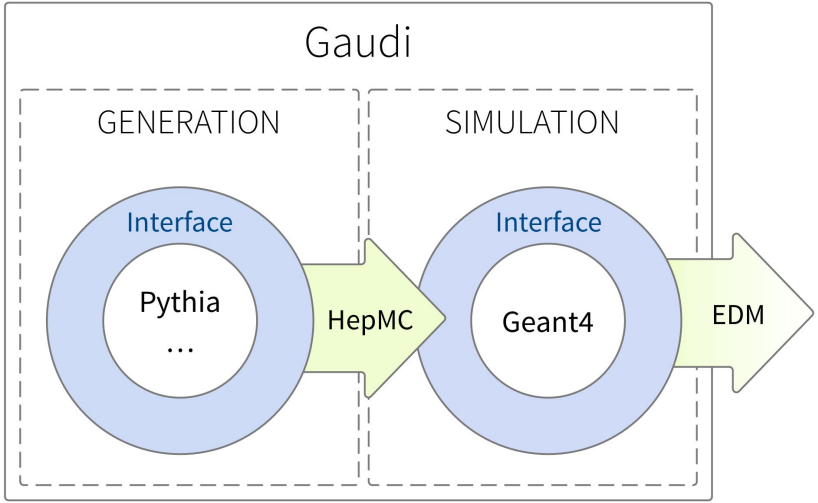


Based on LHCb's Gauss package:

- Event generation and simulation
- Based on Gaudi for event processing
- Independent of EDM: Uses HepMC

Extract the core: **Gaussino**

- Generation infrastructure: ~taken as is
- Simulation: decided to re-design
  - Take the opportunity to simplify (also needed for multi-threading)
  - Integrated fast and full simulation with Geant 4



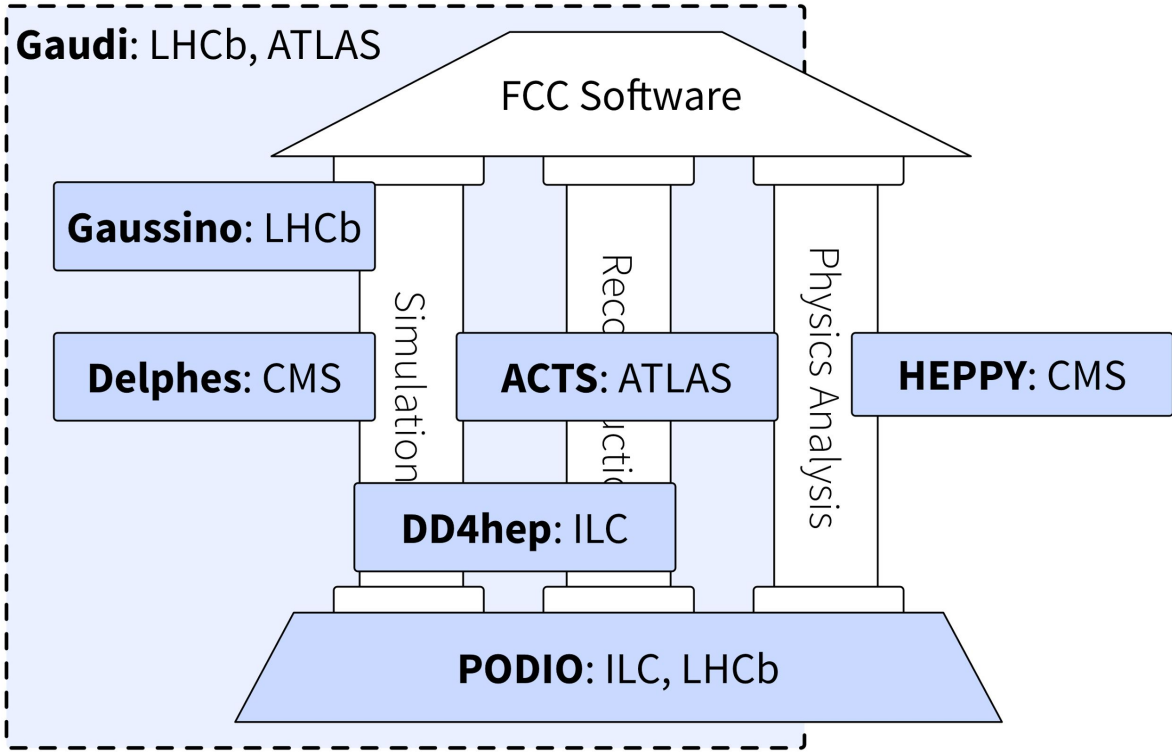
The Solutions

Collaborating where possible

- LHC experiments
- Linear collider community

Synergies

- Share the effort
- **Don’t re-invent the wheel**
- **Improve** existing solutions



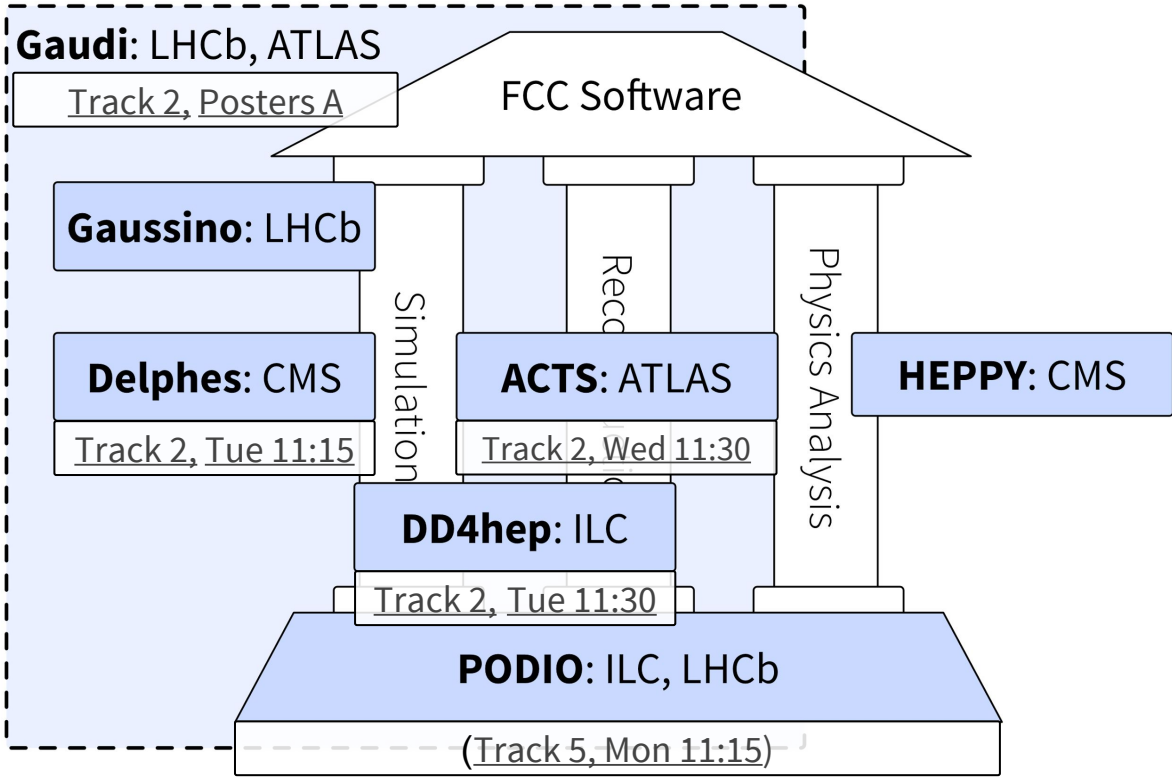
The Solutions

Collaborating where possible

- LHC experiments
- Linear collider community

Synergies

- Share the effort
- **Don’t re-invent the wheel**
- **Improve** existing solutions



Conclusion

Is it possible to keep software experiment independent? **Yes**

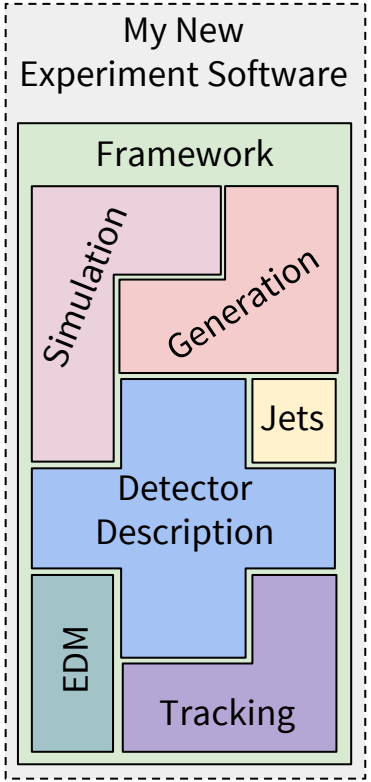
- Met with enthusiasm from the community
  - Developers are happy if their code is reused
- Only a small additional step needed to achieve it

Many good solutions out there

- Cherry pick solutions that fit

Identify areas without obvious candidates

- Invest and give back: Event data model, HEPPY



Thank You

Anna Zaborowska for
C. Bernet, A. Dell'Acqua, Z. Drasal,
B. Hegner, C. Helsens, J. Hrdinka,
J. Lingemann, A. Robson, V. Volkl