CET
Computing Enabling Technologies

Jim Kowalkowski

Who we are

- As members of the Fermilab Scientific Computing Sector, we provide core infrastructure software to experiments and projects, including development and integration.
- Our main focus is data acquisition, experimental data processing and filtering in real-time environments, and C++ programming.
- Our goals this year include bring HPC practices and tools into the systems that we build.
- Part of the service we provide is to help improve software through better problem analysis, design, and implementation techniques and we provide improved tools to help scientists perform their work.

CET People

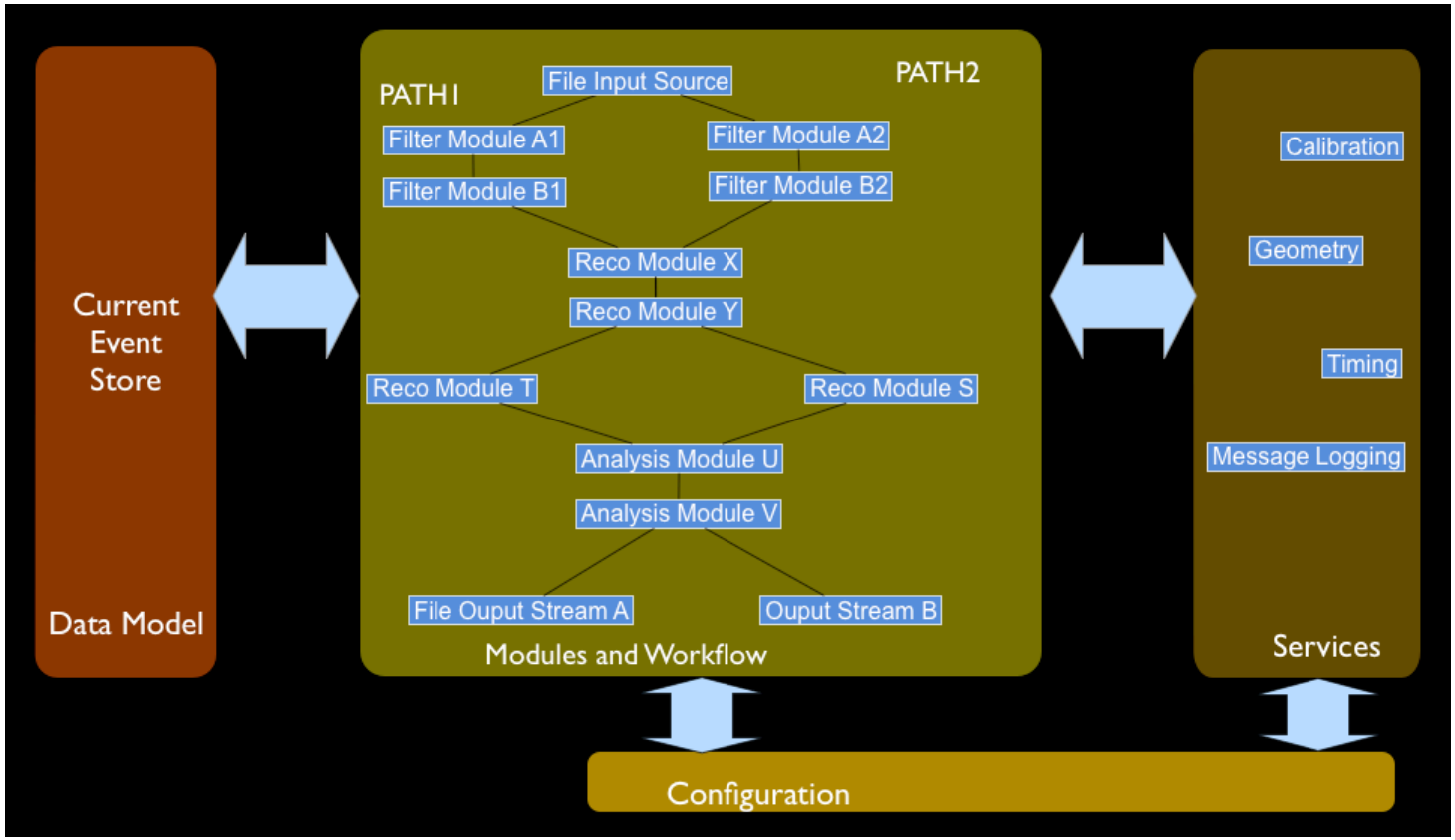
- Kurt Biery
- Walter Brown
- Steve Foulkes
- Chris Green
- Qiming Lu
- Gennadiy Lukhanin
- Marc Paterno
- Ron Rechenmacher

What we do

- We support the needs of small and large collaborations (100's to 1000's of people)
  - efficient processing of large data volumes (including reprocessing)
  - provide ways for everyone to contribute to software development
  - allow collaborations to share software tools and libraries
- We've delivered systems that are relatively easy for scientists to use:
  - to configure and run,
  - to know what has been processed and how,
  - to extend with new algorithms and services,
  - to diagnose problems
- We build software that can be deployed in large cluster environments
- We provide an end-to-end solution: from raw data to final analysis

What we provide

- **Event Processing Framework**: Software that coordinates the processing of *collision events* by pluggable reconstruction, filtering, and analysis modules. Events and modules are independent. Modules add data to and retrieve data from one event at a time.



See http://oink.fnal.gov/new_tut/tutorial.html

FNAL Science Data Processing: Software Frameworks

- Engineer and implement software frameworks for HEP collaborations (CMS, CDF, D0, Intensity Frontier)
- ***Key aspect***: couple software people with the experiment (attend experiment meeting, do software analysis, design, work directly with physics groups on software)
- Frameworks automatically collect metadata (data product provenance, processing history, machine attributes)
- Frameworks define a data model (way to describe "data products" and use them
- Multiple aspects
  - Define general data models for sharing physics objects ("data products")
  - Define protocols for algorithms to work together,
  - Define several I/O interfaces to storage systems, files, dataset catalogs
  - Used for dataset generation, processing, and analysis (auto-generating metadata and status, running applications allowing dataset-level queries

Example system