Introduction to the Muon Collider Software Tutorial

P. Andreetto[(a)], N. Bartosik[(b)], M. Casarsa[(c)], A. Gianelle[(a)], S. Jindariani[(d)],
K. Krizka[(e)], D. Lucchesi[(a,f)], F. Meloni[(g)], S. Pagan Griso[(h)], L. Sestini[(a)]

[(a)]*INFN-Padua, Italy* [(b)]*INFN-Turin, Italy* [(c)]*INFN-Trieste, Italy* [(d)]*FNAL, USA*
[(e)]*University of Birmingham, UK* [(f)]*University of Padua, Italy* [(g)]*DESY, Germany* [(h)]*LBNL, USA*

Muon Collider Physics and Detector Workshop
FNAL, December 14-16, 2022

Introductory remarks

- The software used for Muon Collider studies is based on ILCSoft, a common simulation/reconstruction framework originally developed by F. Gaede (DESY) for ILC and CLIC:
  - ILCSoft repository: `https://github.com/iLCSoft`;
  - documentation: `https://github.com/iLCSoft/ilcsoftDoc`.
- Our approach was to start with CLIC's ILCSoft, a complete, GRID-ready, well supported and documented framework, and adjust it to the different experimental challenges of a Muon Collider:
  - MuonColliderSoft repository: `https://github.com/MuonColliderSoft`.

ILCSoft main components

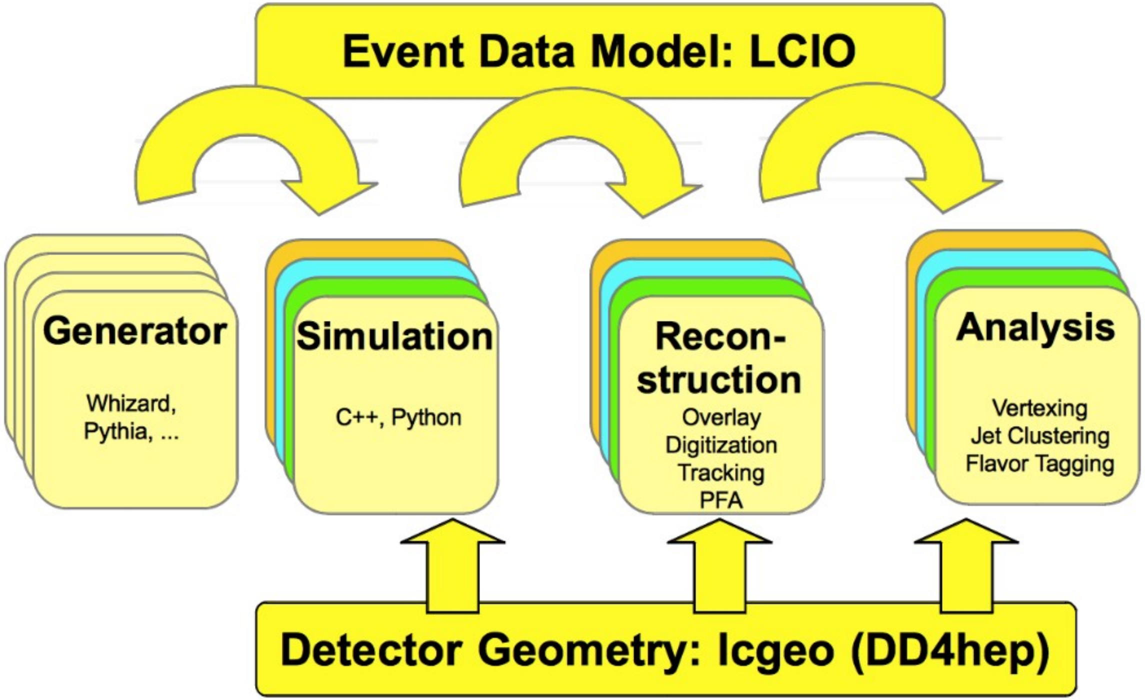
- LCIO (Linear Collider I/O):
  provides the event data model and the persistency framework.
  - github.com/iLCSoft/LCIO.

- DD4hep (Detector Description for High Energy Physics):
  detector geometry description for both the full simulation and the reconstruction step and interface to GEANT4.
  - dd4hep.web.cern.ch/dd4hep.

- MARLIN (Modular Analysis & Reconstruction for the LINear collider):
  is the application framework, based on *processors* dedicated to specific tasks.
  - github.com/iLCSoft/Marlin.

ILCSoft workflow

F. Gaede



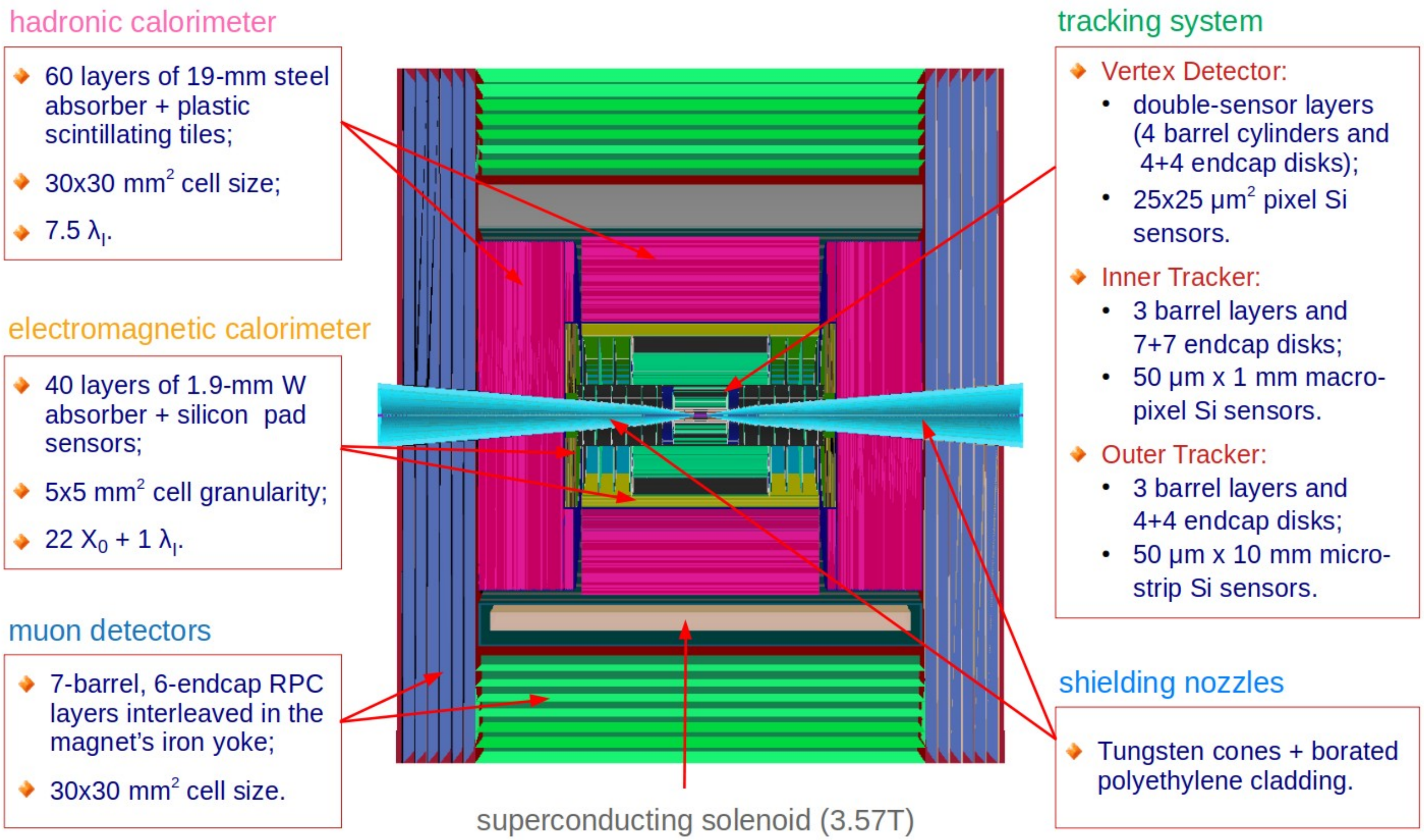
- The ILCSoft workflow consist of four main steps:
  - event generation;
  - detector response simulation;
  - event digitization + reconstruction;
  - event analysis.

Generation step

- Not part of the ILCSoft framework, we have to run our favorite event generator by ourselves.

- And better to get advice from our theorist colleagues before: generating $\mu\mu \rightarrow X$ processes in the multi-TeV regime might be in general tricky.

Simulation step



- Includes a detector model.
- The detector response simulation is based on GEANT4.

Digitization and reconstruction step

- Digitization:
  - tracker: Gaussian smearing of SIM hits' positions and times, readout time window;
  - calorimeters: simple digitization with an energy calibration constant, an energy threshold and a selection time window;
  - muon detectors: simple digitization with an energy calibration constant, an energy threshold and energy saturation.
- Reconstruction:
  - tracks: pattern recognition and track finding based on a Combinatorial Kalman Filter implemented in `ACTS`;
  - calorimeter clusters: `Pandora Particle Flow Algorithm` to recognize different patterns of hits released by different particle types in the high granularity calorimeters;
  - jets: PF-objects clusterization implemented in `FastJet` with a K_t algorithm with R=0.5.

Analysis step

- Many options available:
  - production of ROOT ntuples from the SLCIO files and analysis with C++ macros or PyROOT;
  - histogramming with a dedicated Marlin processor running on the SLCIO files;
  - access and analysis of the SLCIO files with Python API.

Today's program

Muon Collider Software Tutorial

- Computing Setup
- Event Generation
- Simulation
- Digitization and Reconstruction
  - Basics
  - Closer look into the configuration
  - Useful tools
  - Realistic Beam-Induced-Background (BIB)
- Study of Object Performance
  - Histogramming in Marlin (LCIO input files)
  - LCTuple (plain ROOT ntuples)
  - Python Analysis of SLCIO Files
- Algorithm Development
  - Adding a Marlin Processor
  - Tracking Custom Packages
- Advanced Topics
  - Modifying the Detector Geometry
  - Event Displays
  - BIB overlay optimisation
  - Running on batch systems
  - Tweaking particle lifetimes
  - Developing with VSCode
  - Jet reconstruction

Contacts

- For any questions, issues, doubts:

muon_collider_software@lists.infn.it