EDM4HEP: Common software stack for Future Collider projects

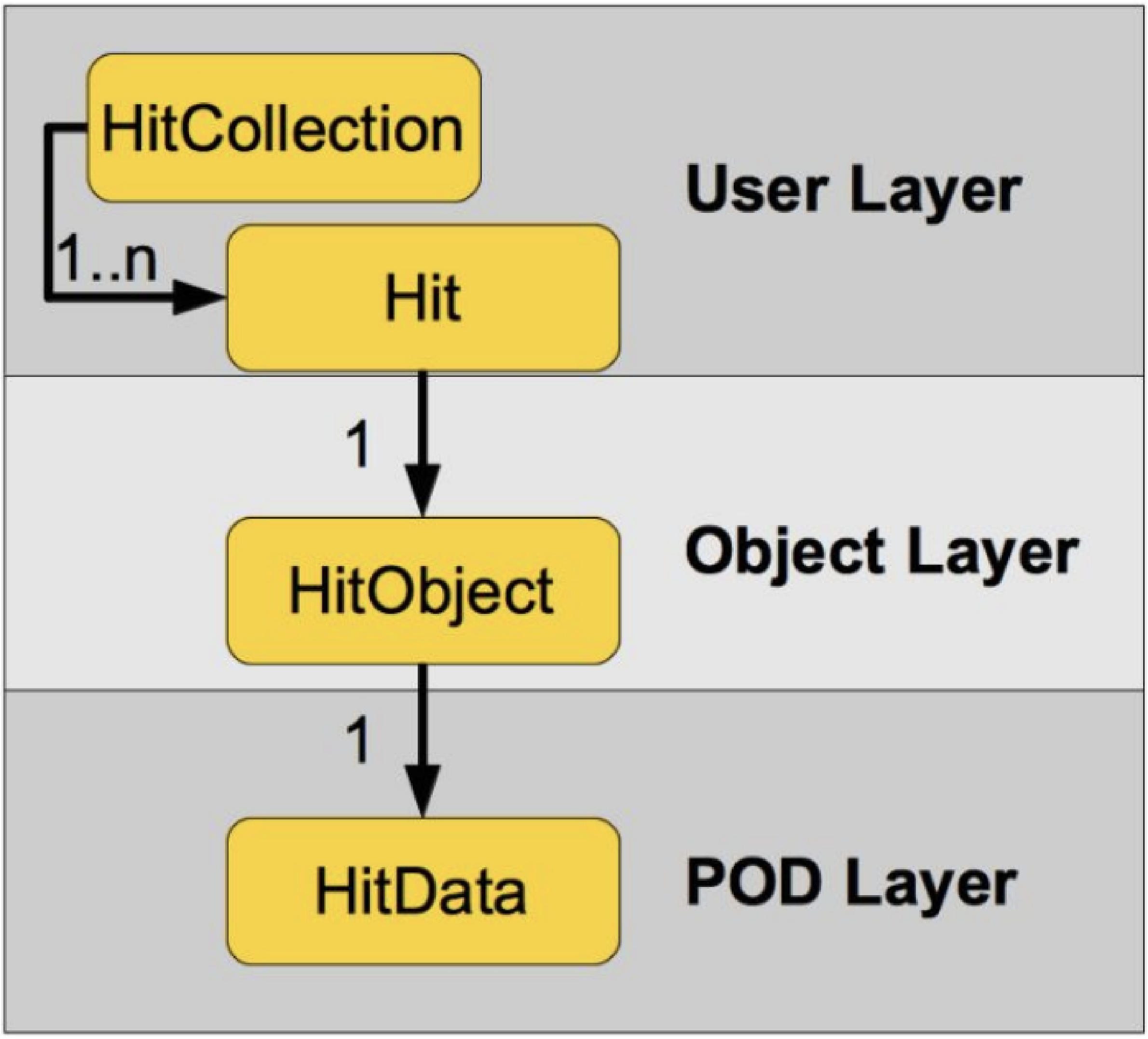
Link to the actual talk: https://indico.fnal.gov/e/55542
Github: https://github.com/key4hep/EDM4hep

Motivation:

- Provide common software stack for future collider experiments
  - ILC, CLIC, FCC, CEPC, EIC…
- Support different use cases of the communities.
- Support multi threading and designed with heterogeneous resources (GPU/Accelerators+CPU) in mind.

Design

- Podio (Plain Old Data IO) to generate high level code (github link).
  - Divergence from C++ object oriented EDM codes in HEP
  - Favors Composition approach over Inheritance one.
- Podio Features
  - Components can be reused.
  - Supports ROOT, RDataFrame and python (ROOT assisted)
  - Code generation from a yaml based markup syntax.
  - Support inter relations between data objects
    - One to one relation
    - One to many relation



Design layout of the Podio data format. The data itself is written in the POD format. There is an object layer (hit object) to access the data. Each hit object corresponds to individual hit data that is written in POD. This is an example of one to one relation between two components. The user layer is the code that accesses the hit object. A hit collection corresponds to many hits. This is an example of one to many relations of the layers.

- POD buffers are stored as TBranches in the TTree or SIO backend.
- No plans on adding an HDF5 backend currently.
- Data models are generated with YAML File when building the EDM based project.
  - YAML files are not committed in the repository.
  - Example of a yaml file (link)
  - YAML basically has the extension information of PODIO data model (which is at the base level)

    - Define Hit, Hit Collection (based on picture above)
    - Relation with various objects defined in the yaml file (one to one, one to many)
    - Internal and External referencing between objects
- Memory Ownership
    - Does a data object act like a smart pointer or weak pointer?
        - Not clear what happens when the object created and attached to the data goes out of scope.

Container of the Event Data

- Aggregation of relevant data defined by the user.
- Store data and related metadata
- Only Immutable read access is allowed
- Frame can be a collection of sub frames (no hierarchies allowed i.e. frame cannot own subframes)