RNTUPLE API REVIEW

FROM JAKOB: SCOPE

- The review should include the RNTuple public interface in the ROOT sources
- The ROOT team would in particular appreciate input on the following aspects:
  - Completeness: is the RNTuple API sufficiently powerful to support experiment I/O workflows?
  - Adherence to modern C++ best practices (e.g., core guidelines)
  - Error handling compatible with experiment frameworks
  - Future-proofing / evolvability: will we be able to evolve the interface in the future in a backwards-compatible way?
  - Usability for end users using C++ and/or Python
  - **Compatibility: ease of migration from the TTree interfaces**

ARCHITECTURE

- Description: https://github.com/root-project/root/blob/master/tree/ntuple/v7/doc/architecture.md

- Top level: TTree ~ RNTupleReader/RNTupleWriter
  - Event iteration for reading/writing

- Any fundamental downside?

RNTUPLEWRITER

- ROOT: ROOT::Experimental::RNTupleWriter Class Reference (cern.ch)
- From ROOT Architecture:
  - *The RNTupleWriter is the primary interface to create an RNTuple.*
  - *The writer takes ownership of a given model.*
  - *The writer can either add an RNTuple to an existing ROOT file (RNTupleWriter::Append()) or create a new ROOT file with an RNTuple (RNTupleWriter::Recreate()).*
  - *Once created, entries are added to an RNTuple either serially (RNTupleWriter::Fill()) or in concurrently in multiple threads with the RNTupleParallelWriter.*
  - *Once committed (e.g. by releasing the RNTupleWriter), the RNTuple is immutable and cannot be amended. An RNTuple that is currently being written cannot be read.*

RNTUPLEREADER

- ROOT: ROOT::Experimental::RNTupleReader Class Reference (cern.ch)
- From ROOT Architecture:
  - *The RNTupleReader is the primary interface to read and inspect an RNTuple.*
  - *An RNTupleReader owns a model:*
    - *either a model created from the on-disk information or an imposed, user-provided model.*
    - *The user-provided model can be limited to a subset of fields.*
    - *Data is populated to an explicit REntry or the model's default entry through RNTupleReader::LoadEntry().*
  - *The reader can create RNTupleView objects for the independent reading of individual fields.*
  - *The reader can create RBulk objects for bulk reading of individual fields.*
  - *Additionally, the reader provides access to a cached copy of the descriptor.*
  - *It can display individual entries (RNTupleReader::Show()) and summary information (RNTupleReader::PrintInfo()).*