Towards a generalized pfRICH/dRICH implementation

- C++ and XML geometry files are of course (Q,PF,D)RICH detector specific
  - Example: https://github.com/eic/epic/blob/irt-2.0/src/PFRICH_geo.cpp

- A unified RICH-IRT plugin [(Q,PF,D)RICH agnostic]
  - https://github.com/eic/EICrecon/blob/irt-2.0/src/detectors/RICH-IRT/RICH-IRT.cc
  - Looks for EICrecon command-line keys *-PQ(PF,D)RICH:config=<JSON config file>*
    - Be aware that plugin name should not coincide with a detector name for this to work

- A partly unified (and detector specific) JSON configuration file
  - QRICH example: https://github.com/eic/EICrecon/blob/irt-2.0/sandbox/qrich-reco.json
  - Mandatory: IRT optics input file, radiator command line style options, photosensor QE
  - Whatever else may need to be turned on/off and / or configured in IRT 2.0 pass
  - Optional and / or detector specific stuff: TBD [HRPPD vs SiPM digitization?]

Towards a generalized pfRICH/dRICH implementation

- A unified IRT 2.0 algorithm / factory interface [(Q,PF,D)RICH agnostic]
  - https://github.com/eic/EICrecon/blob/irt-2.0/src/algorithms/pid/IrtDebugging.h
  - https://github.com/eic/EICrecon/blob/irt-2.0/src/algorithms/pid/IrtDebugging.cc
  - https://github.com/eic/EICrecon/blob/irt-2.0/src/global/pid/IrtDebugging_factory.h

- Neither IrtGeo nor ActsGeo geometry services or their equivalent used [for now]
  - IRT optics is imported as a ROOT C++ class instance image
  - RICH-IRT.cc builds a projection plane request for ACTS engine on its own

- A sparse README.md: https://github.com/eic/EICrecon/tree/irt-2.0/sandbox

Next steps [reordering the focus]

- Implement a single spherical mirror in QRICH
  - To confirm that IRT 2.0 (no sampling along the trajectory) works for dRICH gas radiator
- Confirm that using physical volumes (dd4hep implementation) rather than {logical volume & parent physical volume copies} in the geometry tree (pfRICH standalone code implementation) for sensitive volume elements (cell IDs) does not pose an issue for more complicated geometries (like dRICH sectors) with IRT algorithm

- Back to the “Next steps” items shown 2025/04/03 slides
  - pfRICH conical & pyramid mirrors implementation
  - Calibration pass (and calibration data import in EICrecon)
  - Standalone post-processing scripts -> integrate into IrtDebugging.cc plugin
  - Represent IRT 2.0 digitization as an intermediate plugin
  - Start thinking of a data model