CalDigi Introduction and Status

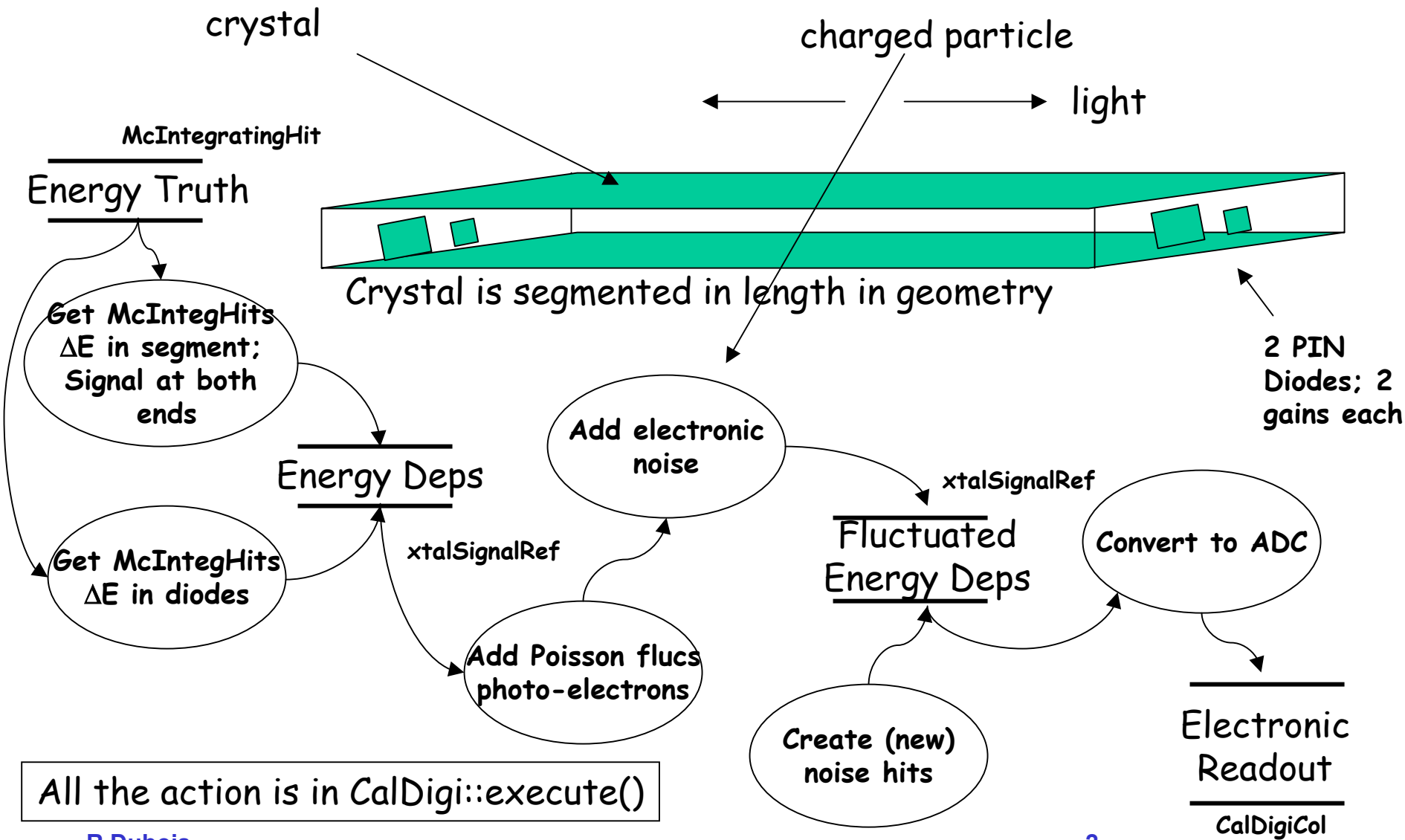
Executive summary: Take MC Truth info on energy deposit (MeV) from simulator and convert to electronic readout format (ADC).

Input is McIntegratingHit, output is CalDigiCol/CalDigi on TDS

In detail:

- for deposit in a crystal segment, take into account light propagation to the two ends and apply light taper based on position along the length.
- keep track of direct deposit in the diode.
- add noise to the diode
- combine (with appropriate scale factor) with crystal deposits
- add noise to 'unhit' crystals; save those above threshold.
- convert to ADC units and pick the appropriate readout range for hits above threshold.

Data Transformation



Technical Details

- xtalSignalRef is an std map, keyed on CalXtalId
  - Store 2 signals (one per end)
  - Segment contributions are summed per CalXtalId
  - 4 diode energies
  - Diode energies are overwritten when noise & electron stats added (bad?)
- Two modes for storing readout
  - 'normal'
    - Best ADC range selected
    - Use highest range that is not saturated
  - 'Calibration'
    - All 4 read out
- Realism (or lack of)
  - "ideal" digitization so far
    - Linear taper
    - 1 pedestal, 1 gain, ie no real calibration constants applied
- Constants storage
  - Almost all constants obtained from detModel in CalDigiAlg::initialize()
  - Noise - e^-/MeV in diode is hard-wired ☹
    - Put in detModel or create own xml file?
- CalDigi is a simple class
  - Crystal ID
  - Vector of readouts (energy range; only one if BEST)
    - CalXtalReadout is nested in CalDigi
    - ADC, range # for both ends

Status

- Ideal digi is essentially done
- Will add realism for the October release
  - 'proper' calibrations
  - Non-linearities
- Should look at parameters handling and remove hard-wired constant
- Maybe look at breaking up execute() into smaller modules