ILC Software Framework for SBelle

Z. Drásal[*], K. Prothmann[+]

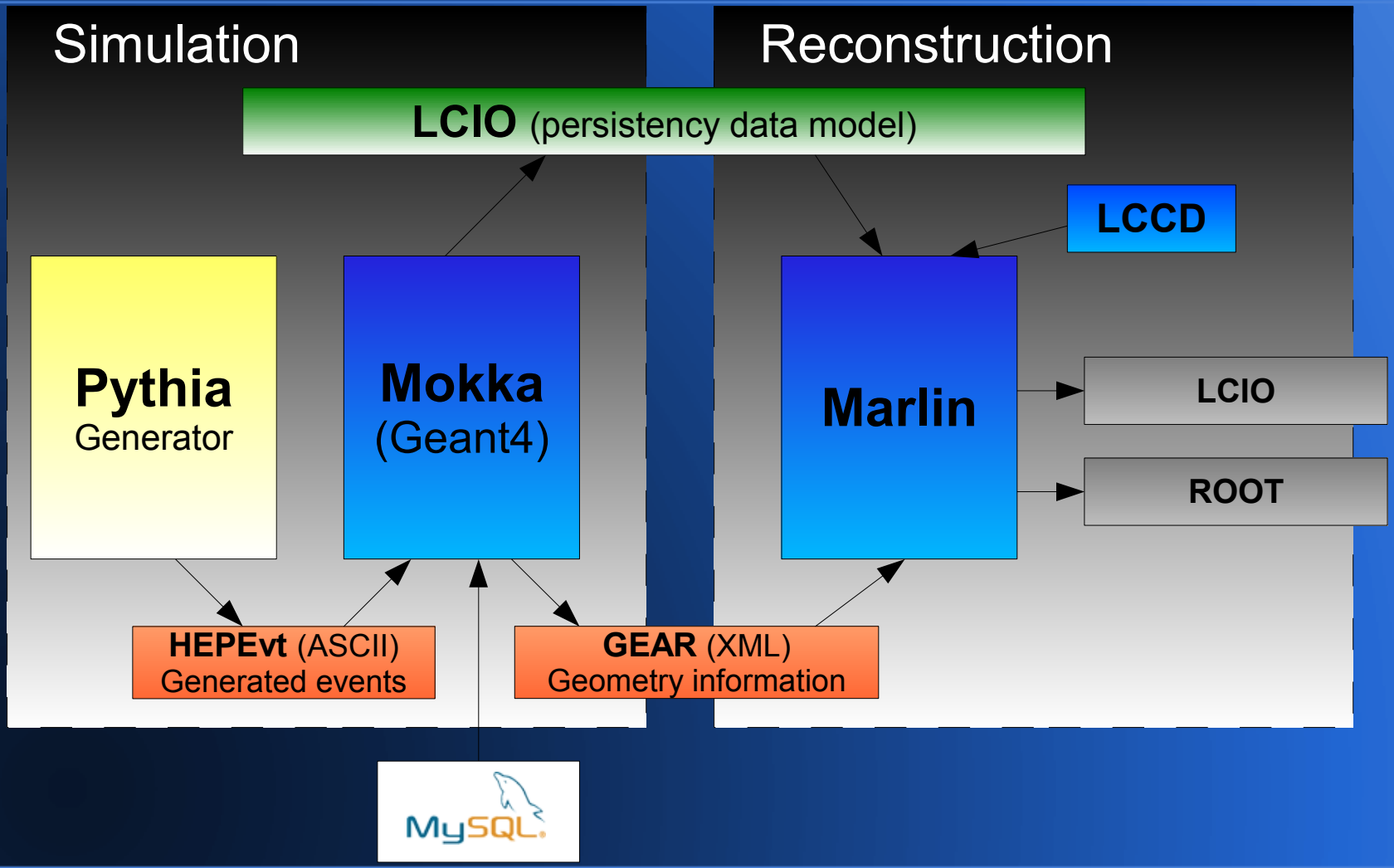
with special thanks to: C. Kiesling[+], A. Raspereza[+]

[*] Charles University Prague
[+] MPI Munich

ILC Software Framework – Summary

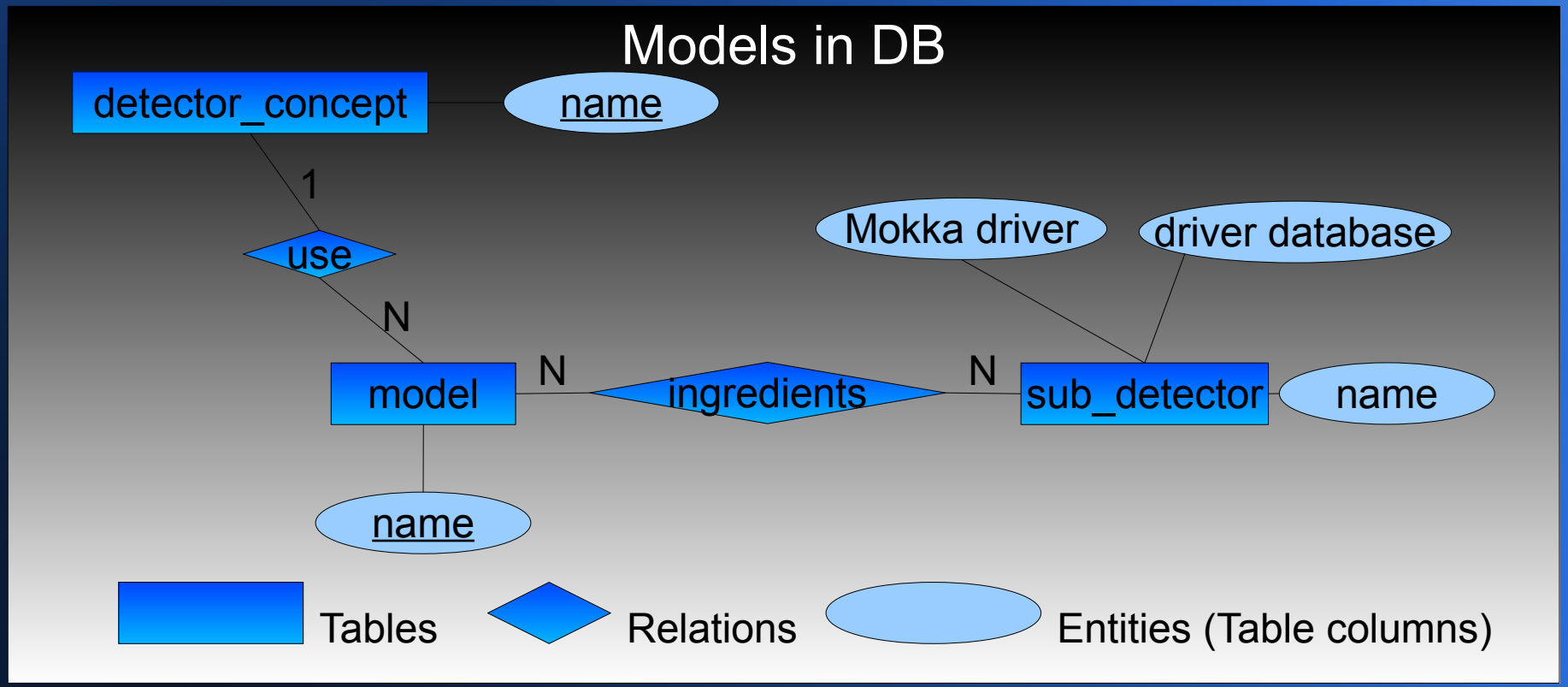
- ***Mokka***: Geant 4 based, full simulation tool using a realistic detector geometry (accessible via a MySQL database) → output in ***ascii*** or ***lcio*** format
- ***LCCD***: Conditions data framework for the ILC
- ***LCIO***: Linear Collider I/O persistency framework, which defines a data model for ILC; concrete data format implementation – Serial Input/Output (**SIO**) → output in ****.slcio***
  - **API:** in Fortran77, C++ and Java
- ***GEAR***: Geometry description toolkit for ILC analysis and reconstruction software → output in ****.xml*** file
- ***Marlin***: ILC Modular C++ Analysis & Reconstruction tool that enables modular approach (using so-called processors) to development of analysis and reconstruction code based on LCIO
- ***Marlin Reco***: Marlin based toolkit providing reconstruction algorithms for data analysis

ILC Software Framework – Scheme



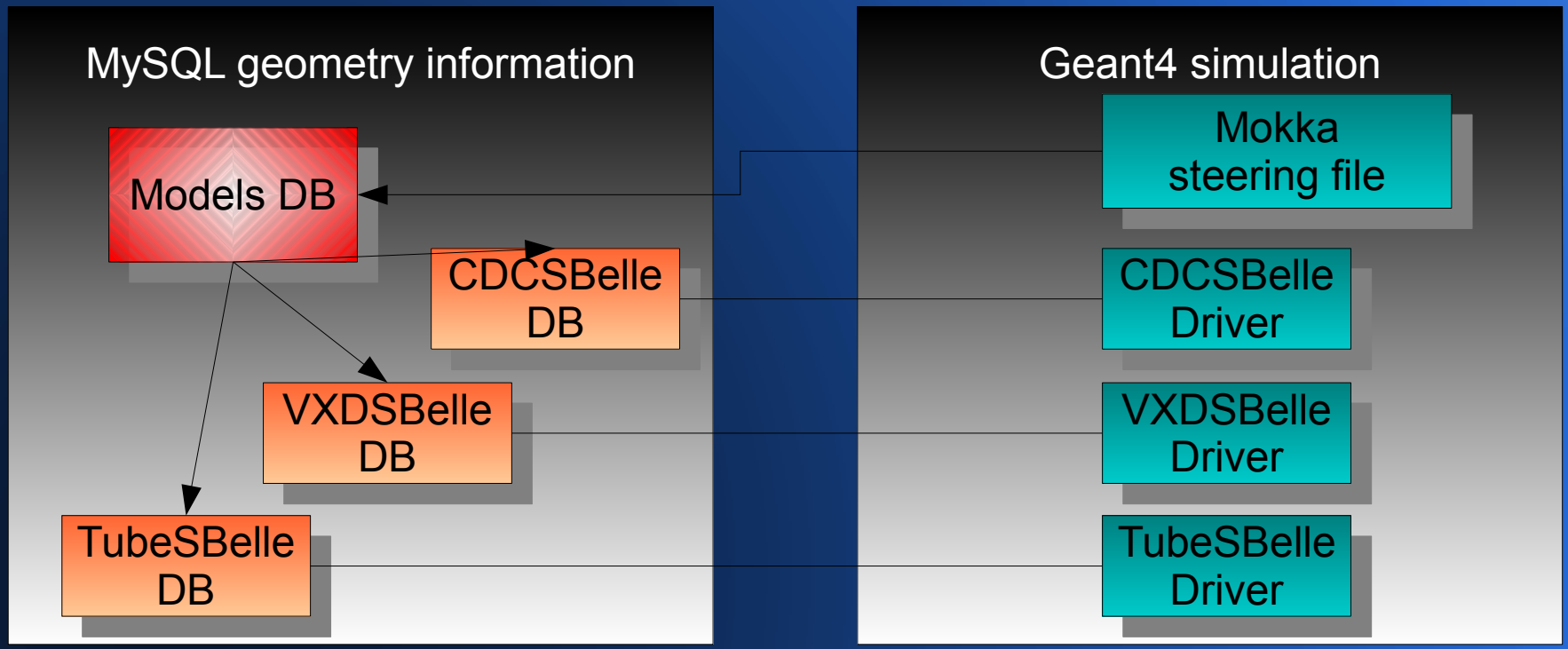
Mokka – Database Layout

- ***Mokka DB***: All geometry information is held in a central database (or its local copy)
  - For each detector concept (BelleExp) different models are defined: Belle, SuperBelle, Super Belle Upgr; each model consists of individual ingredients: PXL, SVD, CDC ...



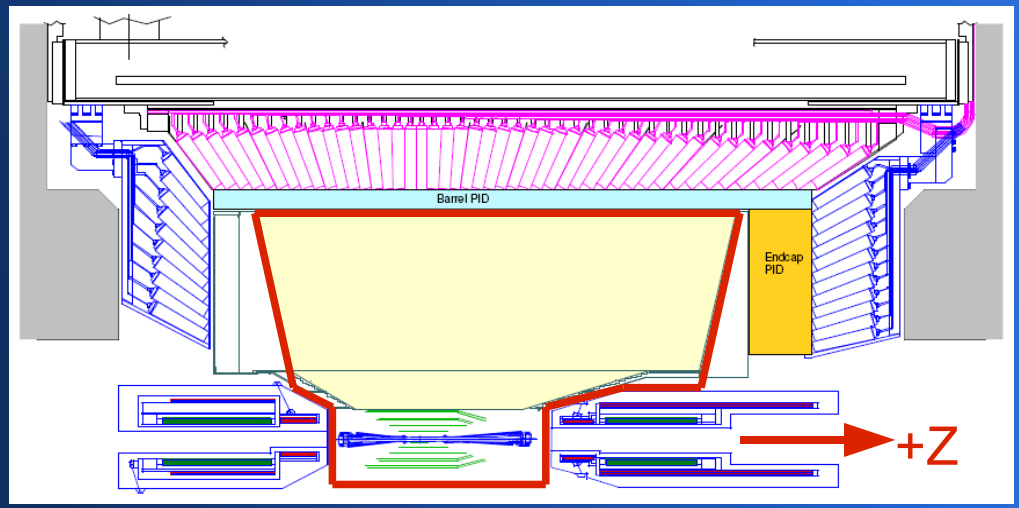
Mokka – Database & Drivers

- ***Mokka geometry***:
  - Geometry drivers (Geant4 DetectorConstruction class) read the information from corresponding MySQL databases
  - Simulation controlled by Mokka steering file; output LCIO file + geometry *.xml file



Mokka – BelleExp Geometry

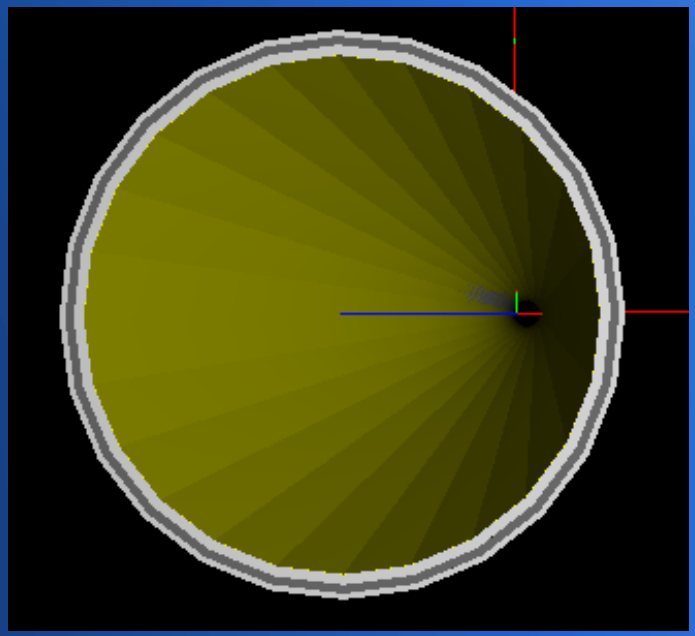
- ***BelleExp geometry***: currently, only tracker implemented (will be officially released withing Mokka framework)
  - ***TubeBelle, TubeSBelle*** – geometry driver of a beam pipe
  - ***VXDBelle, VXDSBelle*** – geometry driver of a vertex detector (pixel and strip detectors together)
  - ***CDCBelle, CDCSBelle*** – geometry driver of a central drift chamber
  - ***Sensitives:***
    - *TRKSD00* (Belle), *VXDSens* (SBelle)
    - *TPCSD02* (both)
  - ***Hits:***
    - *TRKHit* (Belle), *VXDHit* (SBelle)
  - ***Mag. field :***
    - *Field00* (both) – 1.5 T in z



Mokka – Beam Pipe Geometry

- ***TubeSBelle:*** geometry driver that describes a beam pipe for SBelle & SBelle upgrade
  - Cylindrical, onion-like structure , with option to be rotated around Y axis (by 22 mrad):
    - "vacuum"
    - inner gold layer (shielding against soft SR): 10 μm
    - inner beryllium wall: 0.6 mm
    - cooling gap (filled with paraffin): 0.5 mm
    - outer beryllium wall: 0.35

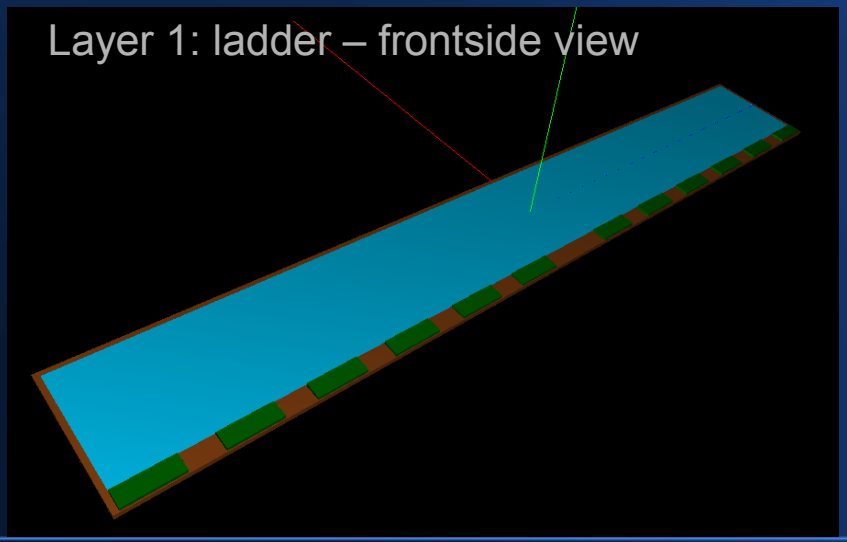
	R_{min} [mm]	R_{max} [mm]
SuperBelle	14.99	16.45
SuperBelleUpgr	8.54	10.00

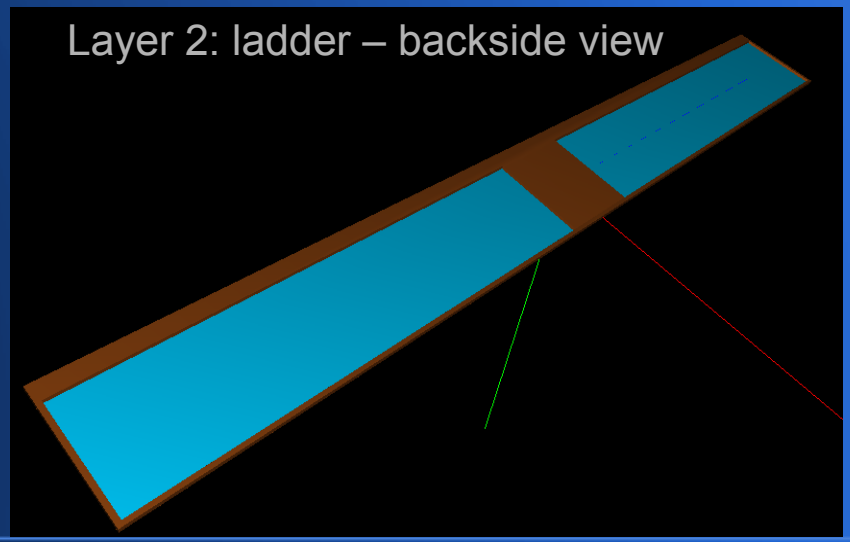


Mokka – VXD Geometry – Pixel Ladders

- ***VXDSBelle:*** realistic geometry driver for SBelle & SBelle upgrade – VXD
  - Description: 2 layers (3 layers for upgrade) → ladders → Si sensors (50 μm) + rims (450 μm) + support (400 μm) + 12 switchers

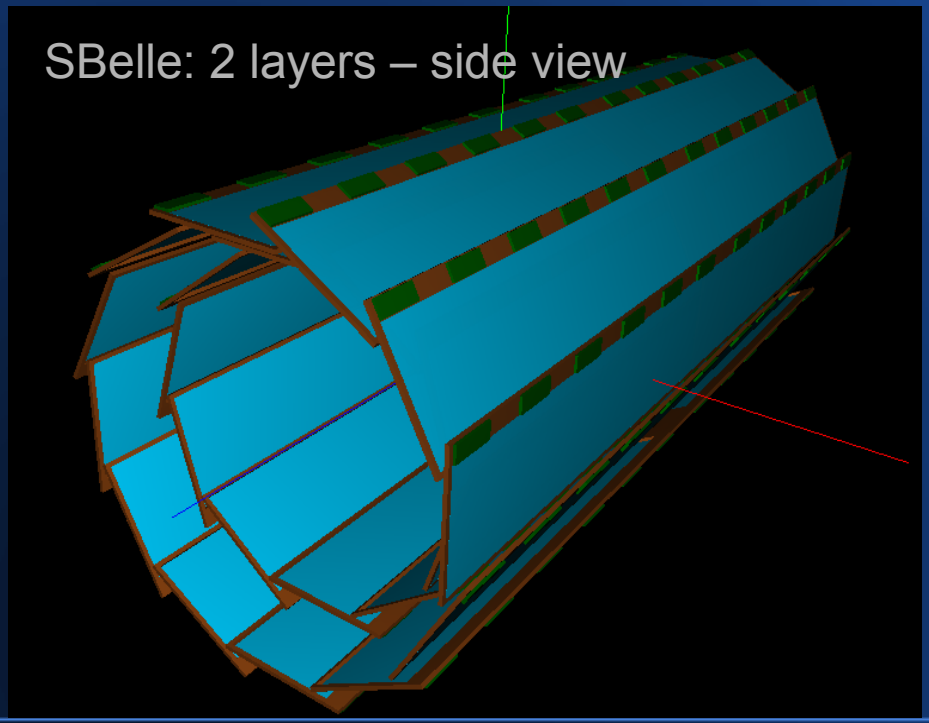
	R [mm]	***# ladders***	***support***
Pxl layer 0	13.00	8	no
Pxl layer 1	18.00	10	no
Pxl layer 2	22.00	12	yes

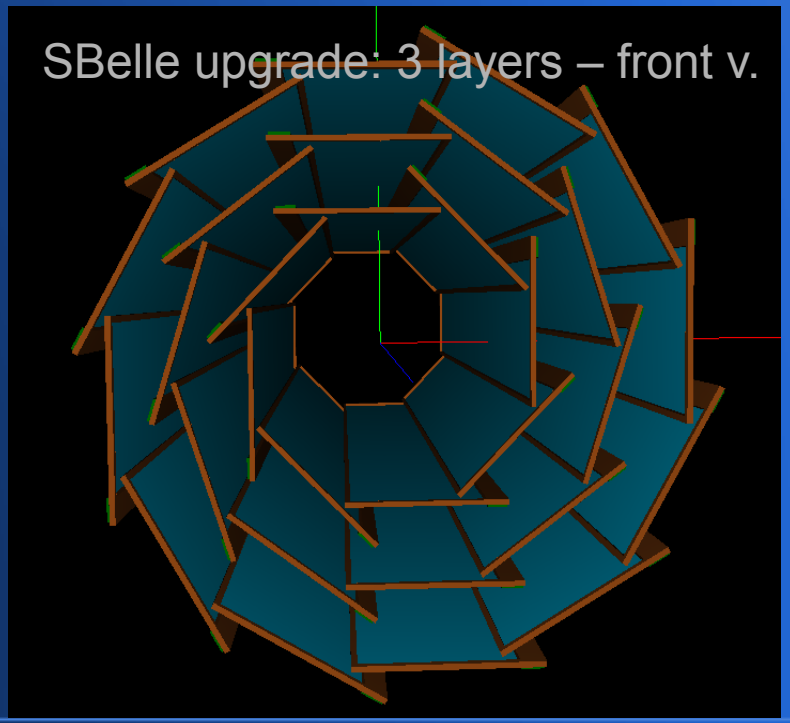




Mokka – VXD Geometry – Pixel Layers

- ***VXDSBelle:*** detail of pixel part for SBelle & SBelle upgrade
  - Layers with wind-mill structure
  - Option: rotate pixel layers together with beam pipe (by 22 mrad)

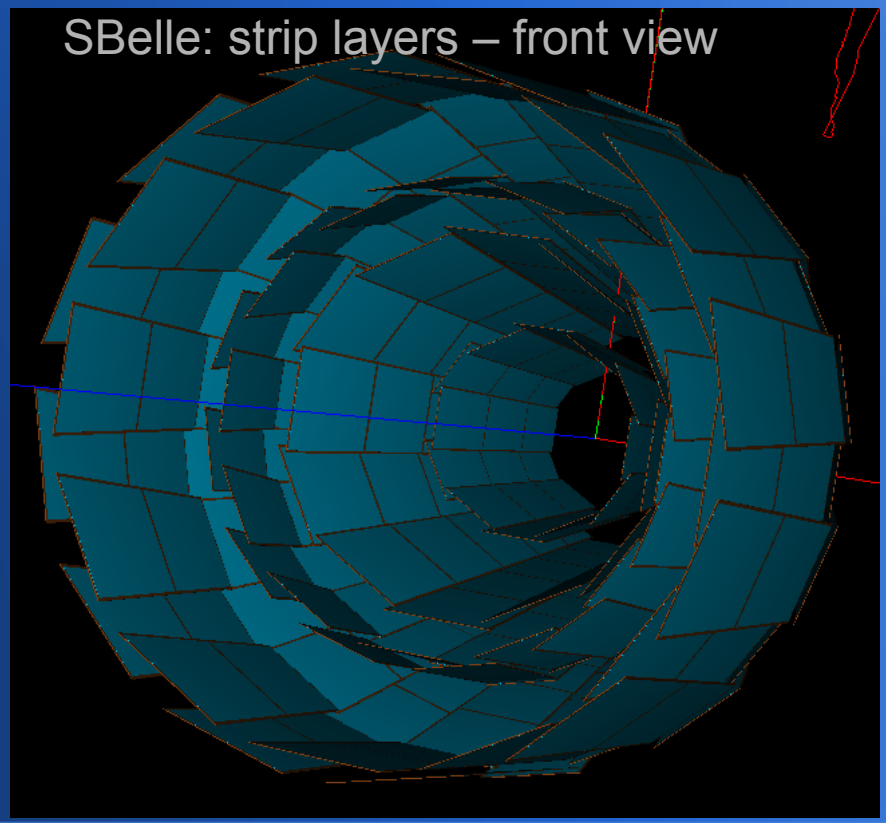




Mokka – VXD Geometry – Strip Layers

- ***VXDSBelle:*** driver used for strip part of VXD of SBelle & SBelle upgrade as well
  - Description: 4 layers in barrel part + 2 layers in "forward" region - (stagger-like structure) → ladders → Si sensors (active part - 300μm) + Si rims (pasive part - 300μm)

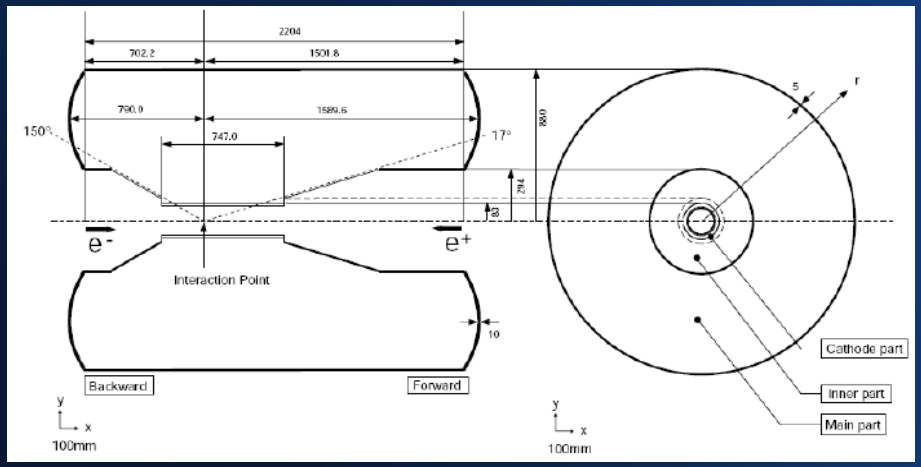
	R [mm]	# *ladders*	# *DSSDs*
Strip layer 31 – barrel	45.15	6	3
Strip layer 32 – barrel	48.15	6	3
Strip layer 41 – barrel	70.15	6	5
Strip layer 42 – barrel	73.15	6	5
Strip layer 51 – barrel	100.15	12	5
Strip layer 52 – barrel-slanted	89.15	12	1
Strip layer 53 – barrel	103.15	12	5
Strip layer 54 – barrel-slanted	92.15	12	1
Strip layer 61 – barrel	137.15	12	6
Strip layer 62 – barrel-slanted	117.65	12	2
Strip layer 63 – barrel	140.15	12	6
Strip layer 64 – barrel-slanted	120.65	12	2

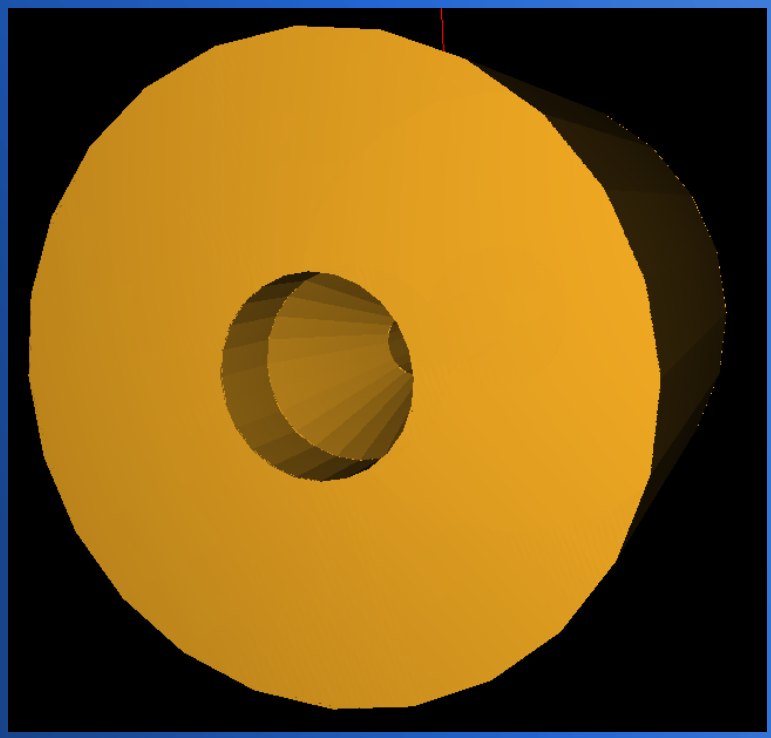


Mokka – CDC Geometry

- ***CDCSBelle:*** realistic geometry driver (by P. Vanhoefer) describing central drift chamber
  - Aluminium cylinder with cone-shaped inner parts filled with gas He/C_2H_6 (50:50)

Radius – inner boundary	361 mm
Radius – inner-middle boundary	150 mm
Radius – outer boundary	1150 mm
Radius – innermost sens. wire	172 mm
Radius – outermost sens. wire	1120 mm
Number of sensitive layers	58

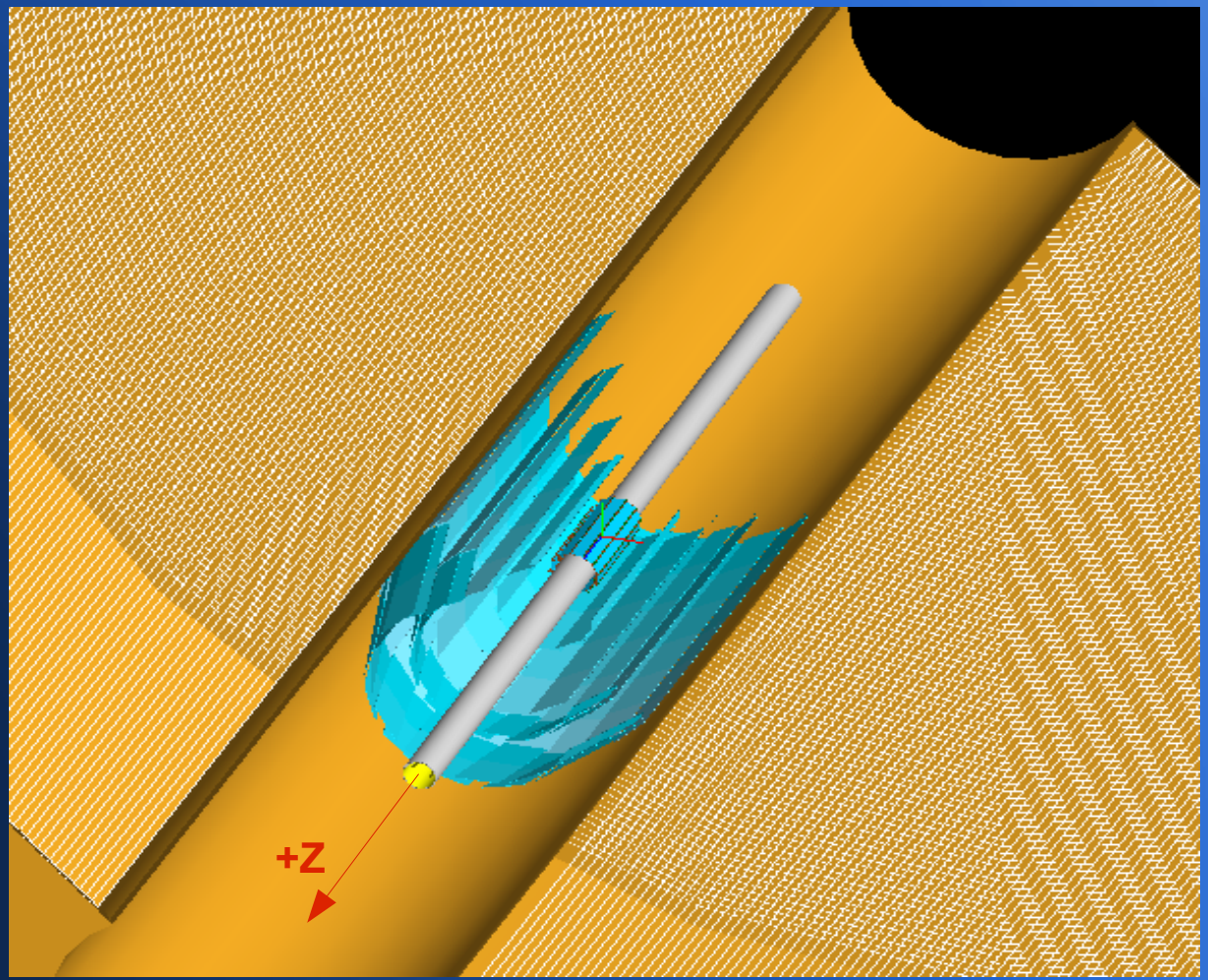




SBelle: CDC – front view

Mokka – Tracker Geometry

- ***Implemented tracker:***
  - Pixel layers (VXD)
  - Strip layers (VXD)
  - CDC



Gear – Geometry Description

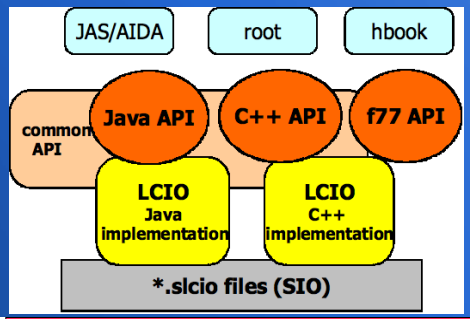
- ***Gear:*** Geometry API for reconstruction and analysis → input/output in xml file
  - ***GearMgr*** – abstract manager class which returns (set/get) all subdetector parameter classes:
    - *BField*
    - *Gear parameters (general)*
    - *VXD parameters*
    - *CDC parameters*
    - ...

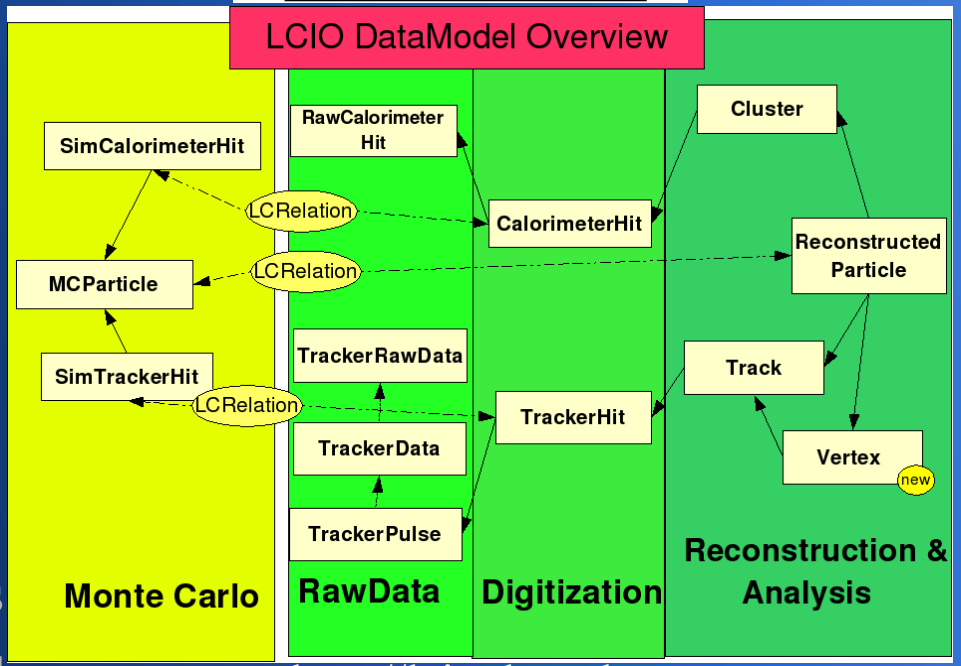
```
<gear>
  <global detectorName="TrkSBelle_CPS1000_SUP10"/>
  <!--
    Gear XML file automatically created with GearXML::createXMLFile ....
  -->
  <BField type="ConstantBField" x="0.000000000e+00" y="0.000000000e+00" z="1.500000000e+00"/>
  <detectors>
    <detector name="TPC" geartype="TPCParameters">
      <driftVelocity value="0.000000000e+00"/>
      <maxDriftLength value="1.102000000e+03"/>
      <readoutFrequency value="0.000000000e+00"/>
      <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin="1.720000000e+02" rMax="1.120000000e+03" padHeight="1.634482759e+01"
      padWidth="4.903448276e+00" maxRow="58" padGap="0.000000000e+00"/>
      <parameter name="CDCGasProperties_RadLen" type="double" value="8.841062425e+05"/>
      <parameter name="CDCGasProperties_dEdx" type="double" value="1.465886091e-07"/>
      <parameter name="CDCWallProperties_RadLen" type="double" value="8.896325058e+01"/>
      <parameter name="CDCWallProperties_dEdx" type="double" value="4.328945801e-04"/>
      <parameter name="cdcInnerRadius" type="double" value="1.500000000e+02"/>
      <parameter name="cdcInnerWallThickness" type="double" value="5.000000000e+00"/>
      <parameter name="cdcIonPotential" type="double" value="4.800000000e-08"/>
      <parameter name="cdcOuterRadius" type="double" value="1.150000000e+03"/>
      <parameter name="cdcOuterWallThickness" type="double" value="5.000000000e+00"/>
    </detector>
    <detector name="VXD" geartype="VXDParameters">
      <vxdType technology="CMOS"/>
      <shell halfLength="0.000000000e+00" gap="0.000000000e+00" innerRadius="0.000000000e+00" outerRadius="0.000000000e+00"
      radLength="0.000000000e+00"/>
      <layers>
        <layer nLadders="10" phi0="9.000000000e+01">
          <ladder distance="1.800000000e+01" thickness="5.000000000e-02" width="1.250000000e+01" length="9.800000000e+01"
          offset="-2.200000000e+00" radLength="9.366078130e+01"/>
          <sensitive distance="1.800000000e+01" thickness="5.000000000e-02" width="1.250000000e+01" length="9.800000000e+01"
          offset="-2.200000000e+00" radLength="9.366078130e+01"/>
        </layer>
        <layer nLadders="12" phi0="9.000000000e+01">
          <ladder distance="2.200000000e+01" thickness="5.000000000e-02" width="1.250000000e+01" length="1.174000000e+02"
          offset="-2.530000000e+00" radLength="9.366078130e+01"/>
          <sensitive distance="2.200000000e+01" thickness="5.000000000e-02" width="1.250000000e+01" length="1.174000000e+02"
          offset="-2.530000000e+00" radLength="9.366078130e+01"/>
        </layer>
```

http://ilcsoft.desy.de/portal/software_packages/gear

LCIO – Data Format

- Bidirectional relations between objects (LCRelation)
  - one to one; one to many, many to many (with weigths)
- API bindings to various languages
  - C++ & Java & Fortran (interface to C++ implement.)
- LCEvent – container for all the data related to this event; LCEvent holds:
  - named collections (LCCollection)
  - data objects (LCObject)
- LCRunHeader – run related information
- SIO – first concrete implementation of LCIO
  - serial data format (machine independent)
  - allows to store, retrieve pointers or references within one record; data compression using zlib



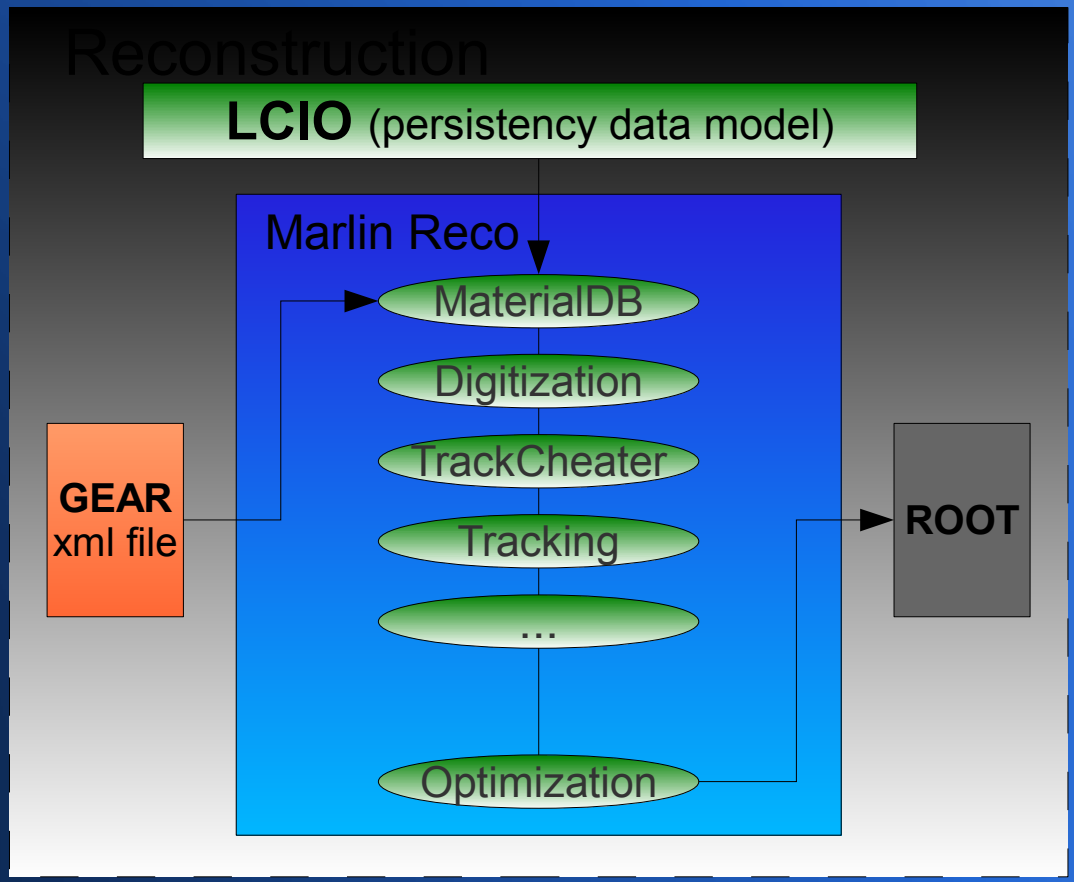


http://lcio.desy.de

Marlin & MarlinReco – Scheme

- ***Marlin&MarlinReco:*** modular C++ software framework (adapted to SBelle experiment)
  - Controlled by an xml steering file

```
<!--
  ##########################################################################
  #                                                                        #
  #         Optimisation studies - Marlin package steering file            #
  #                                                                        #
  ##########################################################################
-->
<marlin>
  <!--
     ***********************************
        Execute following processors
     ***********************************
  -->
  <execute>
    <!--processor name="MyMaterialDB"/-->
    <processor name="MyVTXDigitizer"/>
    <!--processor name="MyCDCDigiProcessor"/-->
    <!--processor name="MyCurlKillerProcessor"/-->
    <!--processor name="MyTrackCheater"/-->
    <!--processor name="MyLEPTrackingProcessor"/-->
    <!--processor name="MySiliconTracking"/-->
    <!--processor name="MyFullLDCTracking"/-->
    <!--processor name="MyVertexViewer"/-->
    <!--processor name="MyFullTrackPerformance"/-->
  </execute>
  <global>
    <!--
       ********************************
          LCIO input files
       ********************************
    -->
    <parameter name="LCIOInputFiles"> ../data3/slcio
    /Output_CPS1000_20deg.slcio </parameter>
    <!--
       ********************************
          GEAR input files
       ********************************
    -->
    <parameter name="GearXMLFile"> ../data3/geometry
    /Geometry_CPS1000_20deg.xml </parameter>
    <!--
```



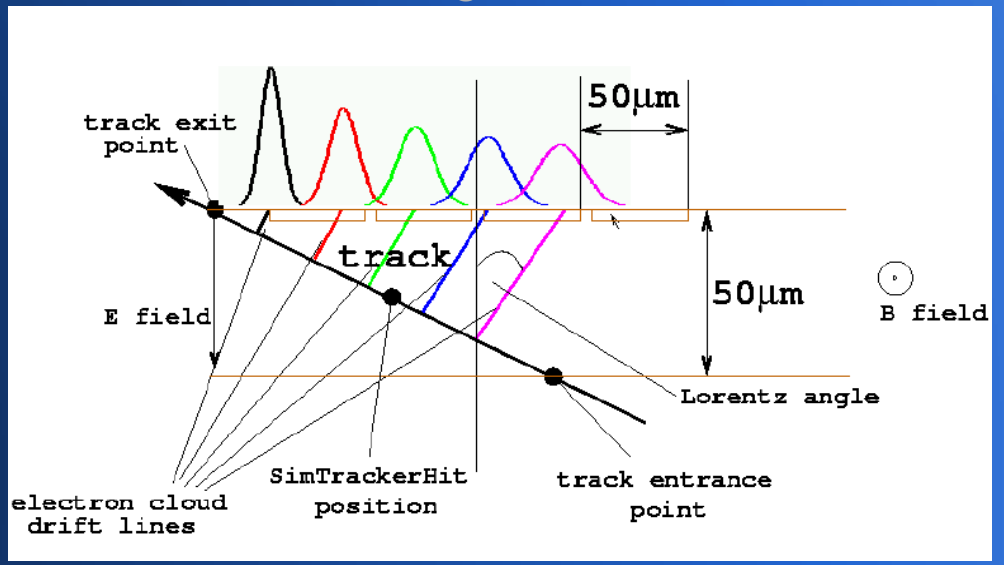
http://ilcsoft.desy.de/portal/software_packages/marlin

Marlin – Material DB

- ***MaterialDB:*** material database processor
  - based on info saved in Gear xml file provides necessary information to Kalman filter in Brahms tracking code (written in Fortran 77) – used to correct tracks to MS effects and particle energy losses
  - currently – tracker geometry implemented (beam pipe, SVD, CDC)

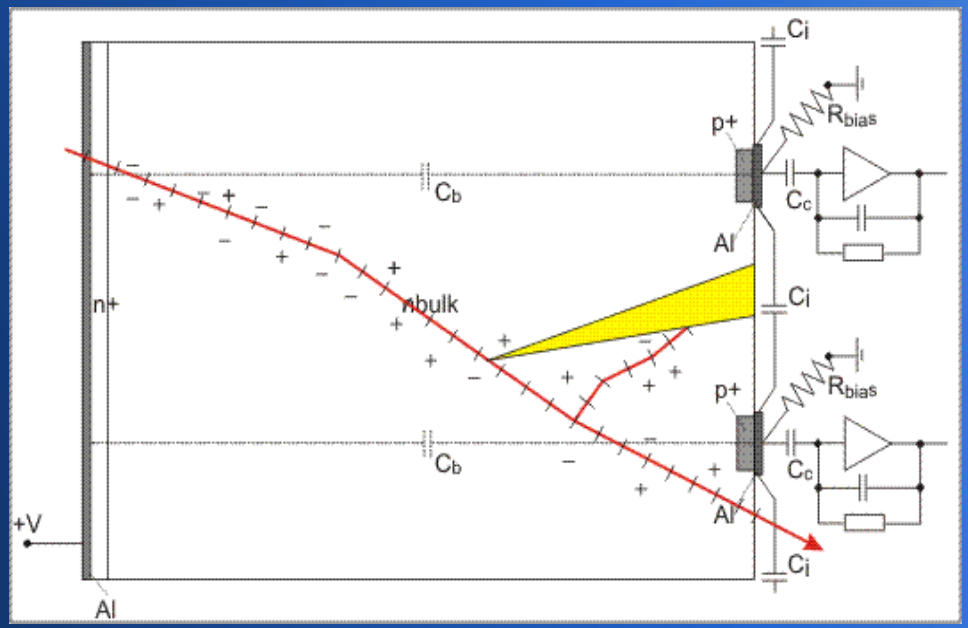
Marlin – PXD Digitizer

- ***VTXDigitizer:*** MarlinReco pixel digitizer – adapted A. Raspereza's VTXDigitizer
  - Input: LCIO SimTrackerHits → Output: LCIO TrackerHits
  - Processes:
    - Global to local ref. system transformation
    - Ionization points generated: energy loss fluctuation added → e-h pairs along the path created
    - Signal points generated: e^- drift performed → e^- Lorentz shift in mag. field of 1.5 T calculated → e^- diffusion calculated
    - Digits produced: pixels with signal bigger than threshold (2 x noise) found
      - noise for pixels set = 100 e
      - noise for strips set = 1200 e
    - Local to global ref. system transformation
    - Hits produced + resolution calculated
    - Background generated



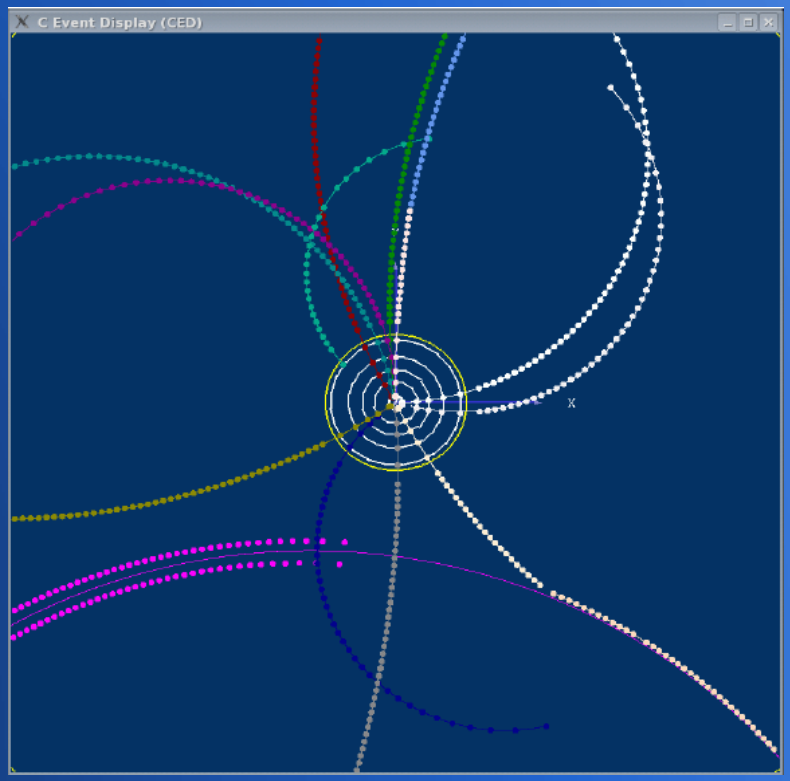
Marlin – SVD Digitizer

- ***SiStripDigi:*** MarlinReco strip digitizer – almost finished
  - Input: LCIO SimTrackerHits → Output: LCIO TrackerHits
  - Geometry: Mokka hits transformation – from global to local reference system
  - Physical processes:
    - Generation of e-h pairs (E_{eh}=3.65 eV)
    - Drift of e-h pairs in electric field
    - Diffusion of e-h due to multiple collisions
    - Lorentz shift of e-h pairs in magnetic field
    - Mutual microstrip cross talks (wrt. AC or DC)
    - Noise: sensor, electronics ...
  - Clustering: (based on COG algorithm)
    - Cluster finding (seed strips + their neighbours)
    - Cluster transformation back to global ref. s.



Marlin – Tracking & Pattern Recognition

- ***Chain of Tracking processors*** *(for more details see http://ilcsoft.desy.de/portal)*:
  - LEPTracking → SiliconTracking → FullLDCTracking
  - TrackCheater – tracking based on MC information
- Pattern recognition in CDC performed: inward search for continuous hit patterns compatible with helix hypothesis (DELPHI code)
- Fit CDC tracks with Kalman filter
- Perform separate pattern recognition in SVD
- Combine SVD tracks and CDC tracks
- Extrapolate track back to the PXD area
- Assign hits on backward helical road
- Refit track after inclusion of new hits using Kalman filter



CEDViewer – SBelle: A.Raspereza's simulation of $e^-e^+ \rightarrow B^-B^+$ event

Summary

- Mokka
  - Geometry drivers for beam pipe, PXD, SVD, CDC - i.e. the whole tracker written
  - MySQL database tables prepared
- Gear
  - A lot of data saved using generic vectors (int, double ...); better solution to redefine particular parameter classes for SBelle purposes
- LCIO – perfectly ok
- Marlin&MarlinReco
  - MarlinDB: adapted
  - Digitizers: for PXD - adapted, for SVD - under development, for CDC – standard used
  - Tracking code: adapted & functional (see presentation in PXL session)
  - Other packages (vertexing, ...): should be compatible with reconstructed data