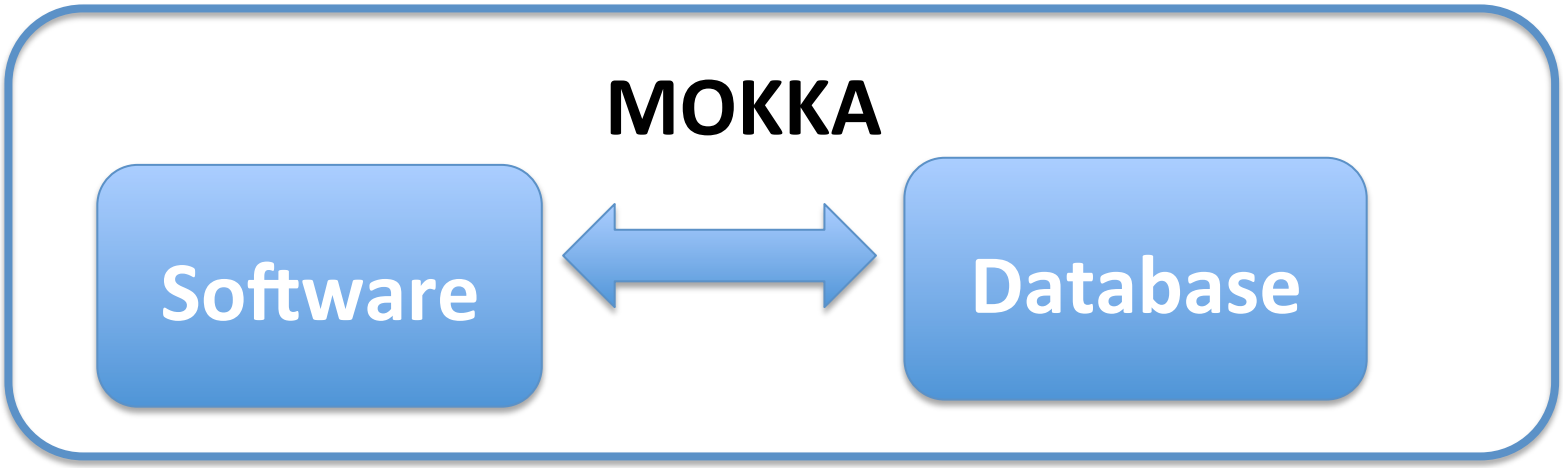


I. General presentation of Mokka software and database architecture, overview relationships between software and database

II. Modifying detector geometry using steering commands

III. Creating new geometry with new drivers

Get better understanding of internal structure of the code, learn how to create a driver template

IV. Creating new geometry model in the database

Working with Mokka

Modify the geometry within Mokka parameters

Emilia BECHEVA
Laboratoire Leprince-Ringuet – Ecole Polytechnique, CNRS

Motivations

- Working with a given model (a set of sub-detectors with fixed geometry dimensions)

Create with Mokka

- Modify the model
- Modify model sub-detectors

=> focus on VTX, ECAL and TPC

Only via steering file

Be confident and free with Mokka

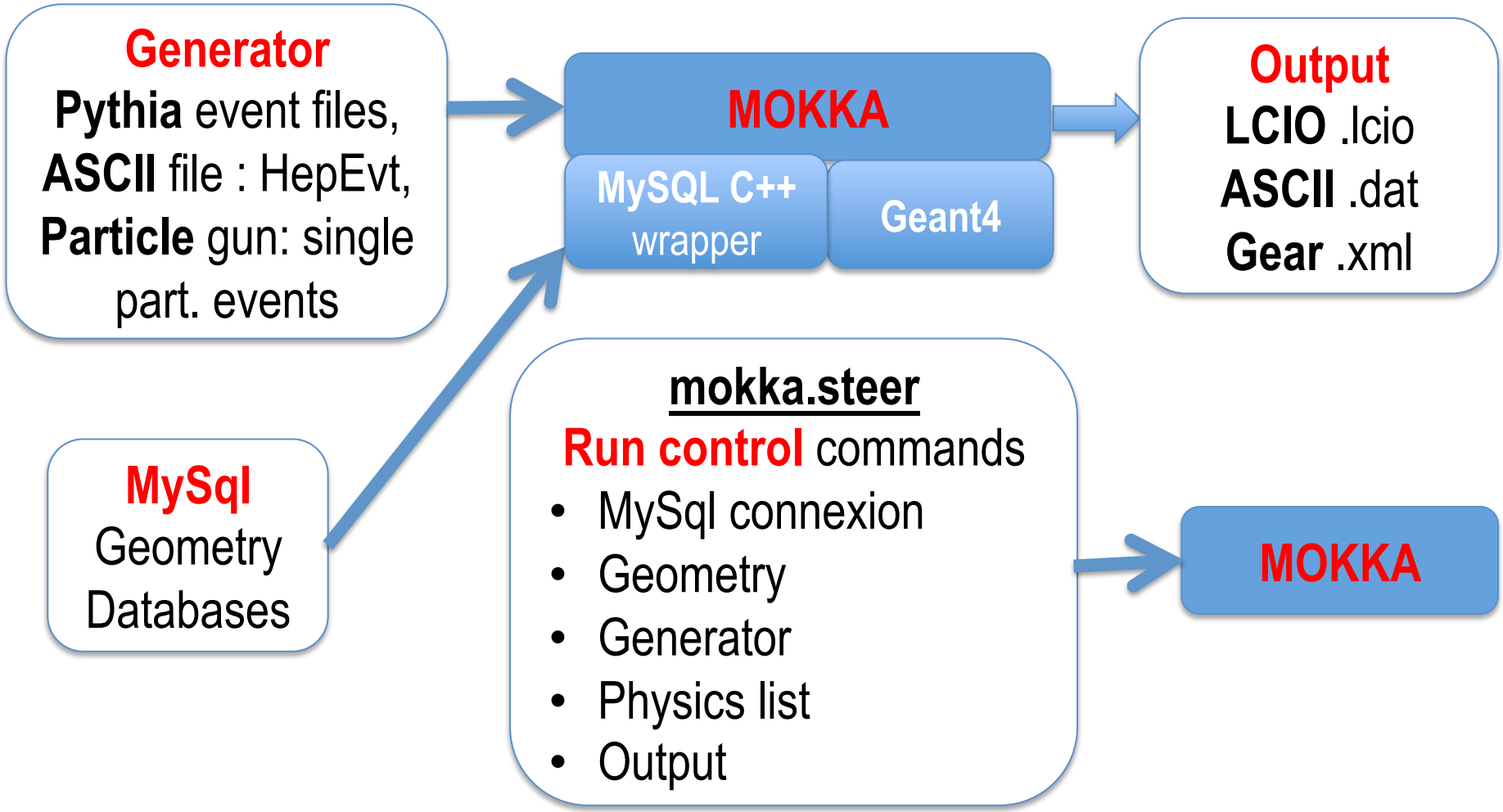
- Check modifications with G4/Mokka tools
- Where to find informations and help

Outline

- Brief reminder of Mokka SW-relationships
- Presentation of a basic steering file
  - Connect to the DB
  - Define a geometrical setup

  $\Rightarrow$ compose a steering file for launching Mokka
- Modifying the model geometry at launch time
  - remove detector
  - replace a sub-detector with another one
  - two ways for running only selected subdetector(s)
- Changing detectors parameters (general presentation)

Mokka SW-Relationships



Reminder

- Run Mokka => **Mokka [option] mokka.streer**
- As Mokka is a Geant4 based application in order to use it you have to:

1) **Define a geometrical setup => choose a model**
2) Define physics list
3) Generate primary event
4) **Save physical information from sub-detectors**
5) Visualization of the geometry

Specific Mokka tasks

mokka.steer

Run control commands

- MySql connexion
- Geometry
- Generator
- Physics list
- Output

Presentation of a basic steering file

mokka.steer

Run control commands

- MySql connexion
- Geometry
- Generator
- Physics list
- Output

Connect to the Mokka database

- Connect to the database : all the information about the model/geometry is in the Mokka databases

/Mokka/init/dbHost pollin1.in2p3.fr ()

/Mokka/init/user consult

/Mokka/init/dbPasswd consult

pollin1.in2p3.fr is an alias to the physical mysql server llrmokka.in2p3.fr

These are default hardcoded values

Connect to the Mokka database

- Connect to a particular port on the MySQL server host

host:port

/Mokka/init/dbHost pollin1.in2p3.fr

/Mokka/init/dbHost pollin4.in5p6.fr:3306 *(default port)*

/Mokka/init/dbHost pollin7.in8p9.fr:0xFCE2

Choosing a different port can be useful if you have multiple MySQL servers (differing in the assigned ports) running on the same machine.

- Connect to the local host via a Unix socket file instead of TCP/IP.

localhost:socket

/Mokka/init/dbHost localhost

/Mokka/init/dbHost localhost:/tmp/mysql-5.0.21.sock

/Mokka/init/dbHost localhost:/afs/cern.ch/user/j/jdoe/mokka/mysql.sock

Define a geometrical setup

mokka.steer

Run control commands

- MySql connexion
- **Geometry**
- Generator
- Physics list
- Output

- Choose a model

A **model** is a set of sub-detectors with fixed geometry dimensions

/Mokka/init/detectorModel model_name

or

$ Mokka -M <model name>

Models (1)

- How to find informations about models

The Mokka Detector Model Database Browser by Adrian Vogels

http://www-flc.desy.de/ldcoptimization/tools/mokkamodels.php

Models (2)

- Please pay attention

In some models and for some sub-detectors

sub_detector name ≠ C++ driver name

Example:

ILD_o3_v06

Sub-det name	vxd07	SEcal05
C++ driver	SVxd04	SEcal05

Example of a model ILD_o2_v06

```
mysql> select model, sub_detector, build_order from ingredients
where model="ILD_o2_v06";
| model         | sub_detector            | build_order |
+---------------+-------------------------+-------------+
| ILD_o2_v06 | LHcal01                    |         120 |
| ILD_o2_v06 | tpc10_01                   |         200 |
| ILD_o2_v06 | ftd_simple_stagg           |         220 |
| ILD_o2_v06 | SEcal05                    |          90 |
| ILD_o2_v06 | SHcalRpc01                 |         110 |
| ILD_o2_v06 | SCoil03                    |         400 |
| ILD_o2_v06 | yoke05                     |         500 |
| ILD_o2_v06 | LumiCalV                   |         100 |
| ILD_o2_v06 | tubeX06                    |         150 |
| ILD_o2_v06 | sit_simple_planar_.        |         210 |
| ILD_o2_v06 | SField01                   |        1000 |
| ILD_o2_v06 | vxd07                      |          20 |
| ILD_o2_v06 | set_simple_planar_.|       230 |
……..
```

Generate primary event

mokka.steer

Run control commands

- MySql connexion
- Geometry
- **Generator**
- Physics list
- Output

- Particle gun (Geant4)

```
/generator/generator particleGun
/gun/particle gamma
/gun/position 0 0 0 mm
/gun/direction 0.0 1.0 0.0
/gun/momentum 10 GeV

/gun/phiSmearing 25 deg
/gun/directionSmearingMode uniform
/gun/thetaSmearing 25 deg

/run/beamOn 1
```

Define physics list

mokka.steer

Run control commands

- MySql connexion
- Geometry
- Generator
- **Physics list**
- Output

- **/Mokka/init/physicsListName LHEP**

Available are all default physics lists provided with geant4, e.g. LHEP, QGSP,...

Default hardcoded physics list is **QGSP_BERT**

Save physical information from sub-detectors

mokka.steer

Run control commands

- MySql connexion
- Geometry
- Generator
- Physics list
- **Output**

- **Native ASCII format**
  - A « Run » corresponds to a directory
  - Every sub-detector creates one or more hit files per event

- **LCIO**
  - Access to data via a high-level interface
  - API for C, C++, Java and f77
  - Automatically integrated to MARLIN

  For more information => **http://lcio.desy.de/**

LCIO output

/Mokka/init/lcioFilename outLCIO_barrel_ref.slcio

/Mokka/init/lcioWriteMode WRITE_NEW

(or WRITE_APPEND)

/Mokka/init/lcioStoreCalHitPosition true

/Mokka/init/lcioDetailedShowerMode true

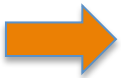
Mokka ASCII native output mode

Save the primaries trajectories only if the user didn't set up the -P option

- Set up the directory name

/Mokka/init/outDirName directory_name

One file/event



Mokka.log
Run.control
ecal000000.hits
event000000.kin
event000000.steps

- Save primaries => *eventxxxxxx.kin output file*

For each track to be saved saves the track ID,
the PDG code, the start position,
the momentum and the particle charge,
the initial energy, the parent ID and the end position.

/Mokka/init/savingPrimaries true

Default values are set to « true »
so if directory name is set
the primaries and the trajectories
are automatically saved

- Save primary trajectories => *eventxxxxxx.steps output file*

For each trajectory, saves only the steps attached to the primaries and at least 100 mm far each one.

/Mokka/init/savingTrajectories true

MokkaGear output

- By default the data is automatically written into GearOutput.xml
- The destination can be changed to newGearFile in the steering file by using the init command:

/Mokka/init/MokkaGearFileName newGearFile

If the file already exists it will be overwritten. Not all parameters can be obtained during construction.

- Also, users might want to change parameters. Therefore two gear xml files can be merged. Using the steering file command

/Mokka/init/MokkaGearMergeSource mergedFile

will merge the automatically generated file with mergedFile. The result will be in the file specified in /Mokka/init/MokkaGearFileName.

Currently for the model **ILD_o2_v06** all subdetectors (**TPC, Ecal, Hcal, Yoke, Lcal, Lhcal, BeamCal, VXD, FTD, SIT, SET, Beam Pipe, Coil)** are supported.

Visualize the geometry using Geant4 commands

- **Draw all geometry**

/vis/scene/create

/vis/open OGLIX

/vis/viewer/flush

/vis/viewer/set/viewpointThetaPhi 70 20

/vis/viewer/zoom 1.5

- **Visualize particular volume**

Add following command to the commands above

/vis/drawVolume **BarrelEcalModule** (This is physical volume name in Geant4 sens)

Visualize the geometry using Mokka commands (based on Geant4)

Several commands to help developers to debug new sub detector drivers, built in the new command: **/Mokka/Visu/Detector/**

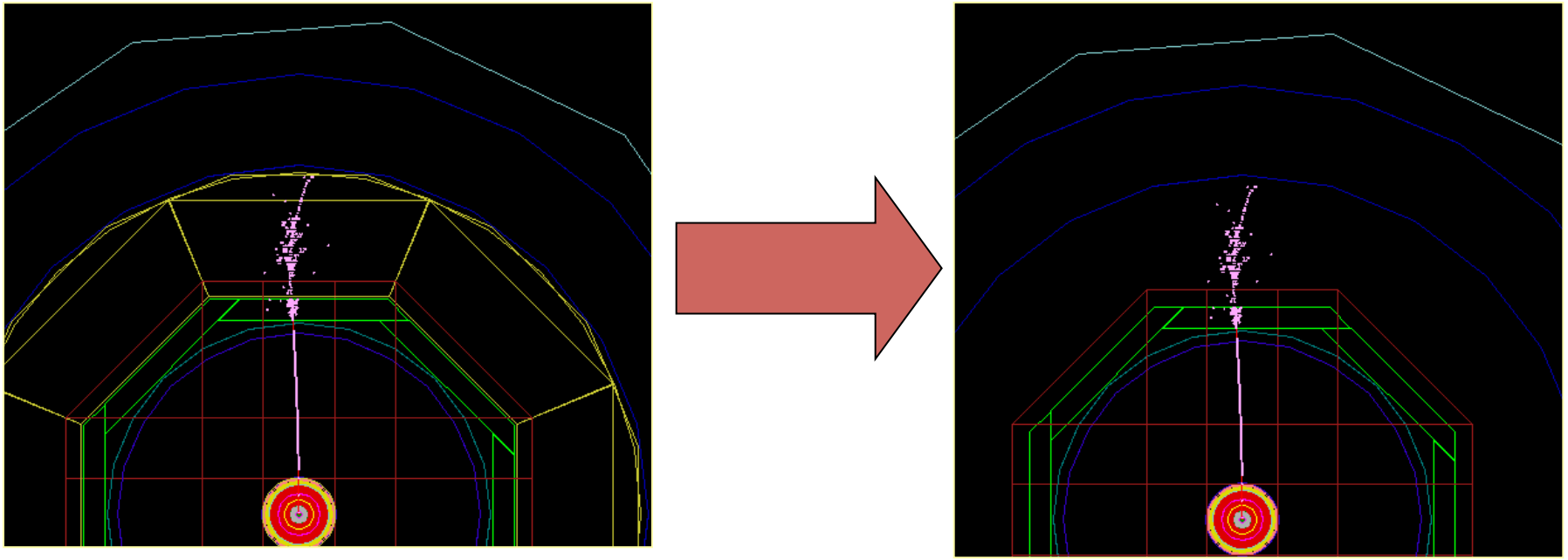
1) Model * Set the rendering mode for a given sub detector and deep
2) Colour * Set the rendering color for a given sub detector deep
3) Daughters * Set the daughter's visibility for a given sub detector and deep
4) **Visibility** * Set the visibility for a given sub detector
5) **ListGeometryTree** * Prints the sub detector names, visibility and sub detector trees
6) ImmediateMode * Automatical refresh of the viewer after each command
7) Reset * Reset the vis attributes to the model default

The user can select the volume to have new visualisation attributes giving a sub detector name (ecal, vxd, hcal, etc.), a deep level in the geometry three and/or a logical volume name. For more information, please, type help and follow the command path.

Visionning models

- The user is able to interactively modify the model rendering, ex :

  Idle> /Mokka/Visu/Detector/Visibility hcal false



Basic example of steering file

- https://llrforge.in2p3.fr/viewvc/Mokka/tags/mokka-08-03/

⇒ mokka.steer

mokka.steer

```
/Mokka/init/detectorModel model_name
/Mokka/init/dbHost pollin1.in2p3.fr
/Mokka/init/user consult
/Mokka/init/dbPasswd consult

/Mokka/init/physicsListName QGSP_BERT
/Mokka/init/initialMacroFile initFile

/Mokka/init/lcioWriteMode WRITE_NEW
/Mokka/init/lcioStoreCalHitPosition true
```

initFile

```
/vis/open OGLIX
/vis/viewer/flush
/vis/viewer/set/viewpointThetaPhi 70 20
/vis/viewer/zoom 1

/generator/generator particleGun
/gun/particle gamma
/gun/position 0 0 0 mm
/gun/direction 0.0 1.0 0.0
/gun/momentum 10 GeV

/run/beamOn 1
```

Motivations

- Working with a given model (a set of sub-detectors with fixed geometry dimensions)

Create with Mokka

- Modify the model
- Modify model sub-detectors

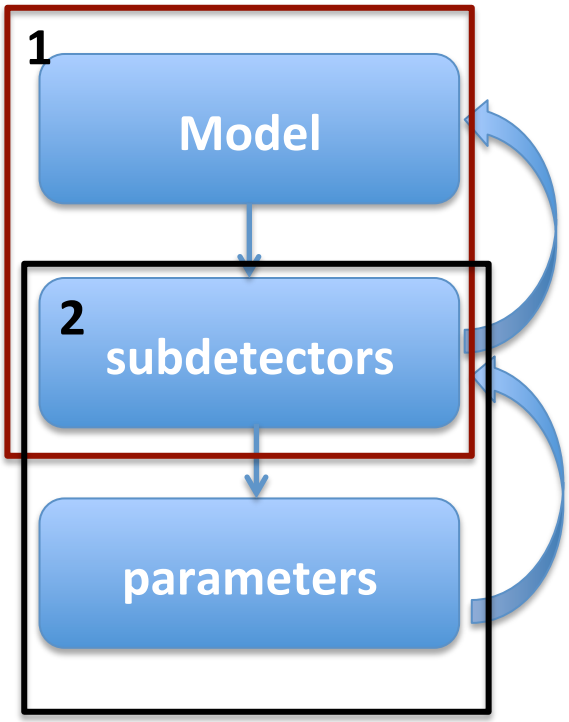
=> focus on VTX, ECAL and TPC

Only via steering file

Be confident and free with Mokka

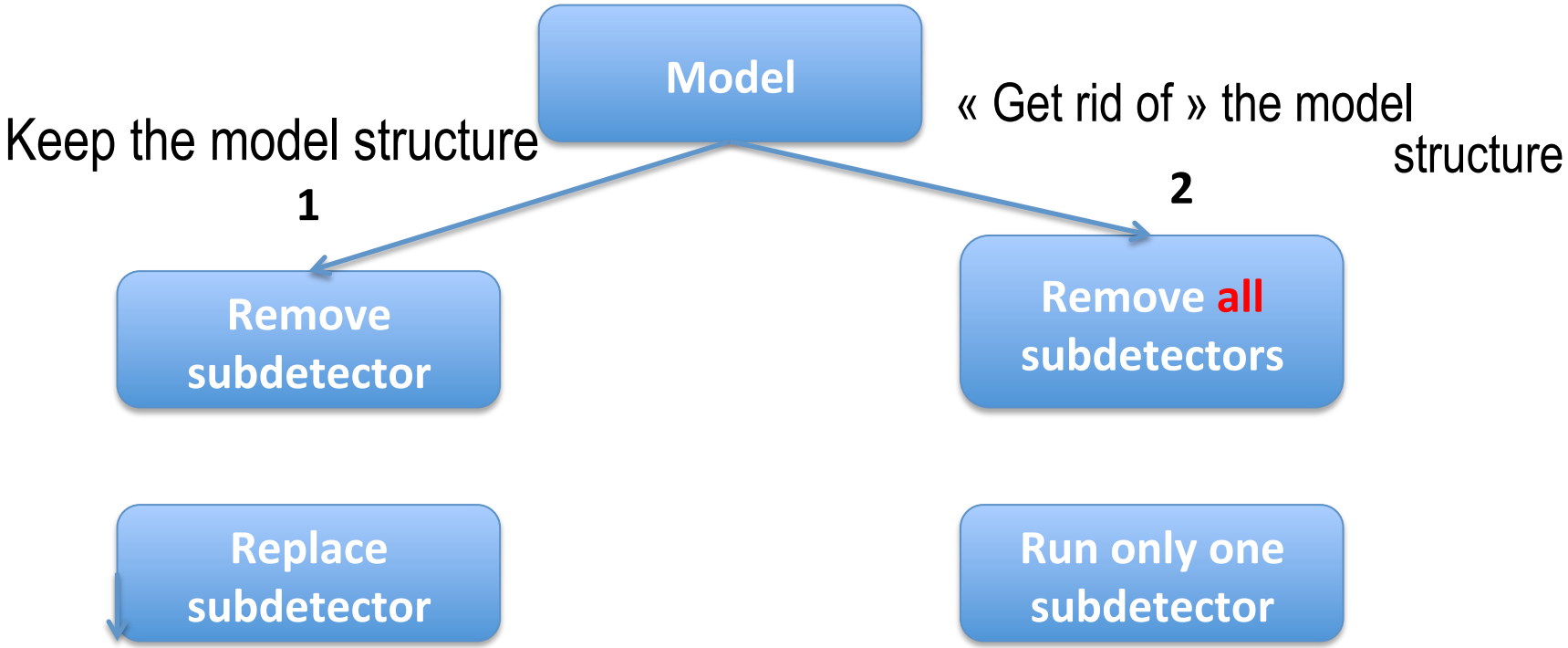
- Check modifications with G4/Mokka tools
- Where to find informations and help

Start creating with Mokka



- Modify the model at lunch time
- Modify model sub-detectors by modifying parameters
  => examples for VTX, ECAL and TPC

Modify the model



Modifying the model geometry at launch time
Remove detector

/Mokka/init/detectorModel **model_name**

/Mokka/init/EditGeometry/rmSubDetector **sub_det1**

mysql> select * from ingredients where model="ILD_o2_v06";

```
+-------+---------------+--------------------+---------------+
| id    | model         | sub_detector       | build_order |
+-------+---------------+--------------------+---------------+
| 1633 | ILD_o2_v06 | LHcal01            |       120     |
| 1634 | ILD_o2_v06 | tpc10_01           |       200     |
.....................................................................|
| 1636 | ILD_o2_v06 | SEcal05            |       90      |
| 1637 | ILD_o2_v06 | SHcalRpc01         |       110     |
.....................................................................
```

Remove a subdetector from an existing detector model

/Mokka/init/detectorModel ILD_o2_v06
/Mokka/init/EditGeometry/rmSubDetector SEcal05

Information in the Mokka log file

Order for constructing detectors

Original model (ILD_o2_v06) recipe in database:

Subdetector/build order
Vxd07 / 20
SEcal05 / 90
LumiCalV / 100
SHcalRpc01 / 110
LHcal01 / 120
tubeX06 / 150
maskX03 / 160
tpc10_01 / 200
etc

Final model (based on ILD_o2_v06)

Subdetector/build order
Vxd07 / 20
LumiCalV / 100
SHcalRpc01 / 110
LHcal01 / 120
tubeX06 / 150
maskX03 / 160
tpc10_01 / 200
etc

Replace a subdetector

/Mokka/init/detectorModel **model_name**

/Mokka/init/EditGeometry/rmSubDetector **sub_det1**

/Mokka/init/EditGeometry/addSubDetector **sub_det2** **build_order**

Default value is set to 0

mysql> select * from ingredients where model="ILD_o2_v05";

id	model	sub_detector	build_order
1550	ILD_o2_v05	LHcal01	120
1551	ILD_o2_v05	tpc10_01	200
1552	ILD_o2_v05	ftd_simple_staggered_02	220
1553	ILD_o2_v05	SEcal03p01	90
1554	ILD_o2_v05	SHcalRpc01	110

Replace a sub-detector with another one in existing model

/Mokka/init/detectorModel ILD_o2_v05
/Mokka/init/EditGeometry/rmSubDetector SEcal03p01
/Mokka/init/EditGeometry/addSubDetector SEcal05 90 (ILD_.._v06)

Order for constructing detectors

Original model (ILD_o2_v05) recipe in database:

Subdetector/bulid order

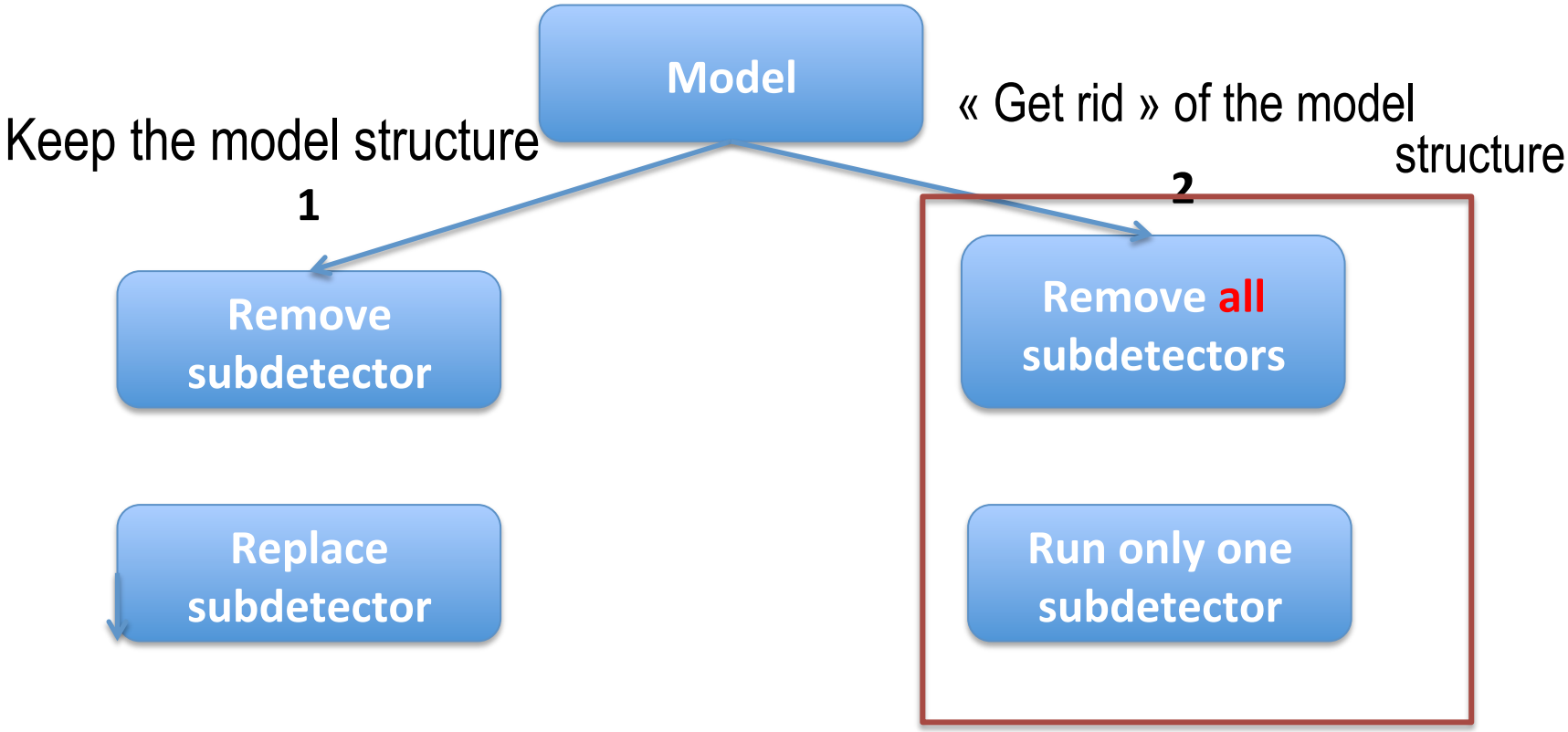
Vxd07 / 20
SEcal03p01 / 90
LumiCalV / 100
SHcalRpc01 / 110
LHcal01 / 120
tubeX06 / 150
maskX03 / 160
tpc10_01 /200
etc

Final model

Subdetector/bulid order

Vxd07 / 20
SEcal05 / 90
LumiCalV /100
SHcalRpc01 / 110
LHcal01 / 120
tubeX06 / 150
maskX03 / 160
tpc10_01 / 200
etc

Modify the model



Run only selected detector

- Run only selected detector

- Get a model structure and drop all the sub-detectors

/Mokka/init/detectorModel ILD_o2_v05

/Mokka/init/EditGeometry/rmSubDetector all

- Add particular sub-detector(s)

/Mokka/init/EditGeometry/addSubDetector SEcal05 90

Run only selected detector

Information in the Mokka log file

Connecting to the database "models03 »
Asking for the model ILD_o2_v05: found.
Cooking the geometry, original model recipe in database:
(subdetector/database/driver/sub_driver/build_order)
vxd07 / vxd07 / SVxd04 / vxd04 / 20
SEcal03p01 / VOID / SEcal04 / / 90
……

Edition commands
(0=add, 1=rm / name / build_order)
1 / all / 0
0 / SEcal05 / 90
Final model recipe after cooking it:
(subdetector/database/driver/sub_driver/build_order)
SEcal05 / VOID / SEcal05 / / 90
Building sub_detector SEcal05, geometry db VOID, driver SEcal05:

Run more than one selected detectors

/Mokka/init/detectorModel ILD_o2_v05

/Mokka/init/EditGeometry/rmSubDetector all

/Mokka/init/EditGeometry/addSubDetector SEcal05 90

/Mokka/init/EditGeometry/addSubDetector tpc10_01 200

Cooking the geometry, original model recipe in database:
(subdet/db/driver/sub_driver/build_order)
vxd07 / vxd07 / SVxd04 / vxd04 / 20
SEcal03p01 / VOID / SEcal04 / / 90
……..
tpc10_01 / tpc10_01 / tpc10 / / 200

Edition commands
(0=add, 1=rm / name / build_order)
1 / all / 0
0 / SEcal05 / 90
0 / tpc10_01 / 200
Final model recipe after cooking it:
(subdet/db/driver/sub_driver/build_order)
SEcal05 / VOID / SEcal05 / / 90
tpc10_01 / tpc10_01 / tpc10 / / 200

Run only selected detector

In steering file:

/Mokka/init/detectorModel ILD_o2_v06

/Mokka/init/subDetector SEcal05

On line when running Mokka

Mokka –S <subdet name> mokka.steer

!!! Works also with the model ILD_o2_v05 even the subdetector SEcal05 is not included in this model.

Run only selected detector(s)

Differences between two methods

1 **/Mokka/init/EditGeometry/rmSubDetector all**
/Mokka/init/EditGeometry/addSubDetector SEcal05 90
/Mokka/init/EditGeometry/addSubDetector tpc10_01 200
and
2 **/Mokka/init/subDetector SEcal05**

1 – can add more than one subdetectors even they are not included in the model

2- only one subdetector, even not be included in the model, if two subdetectors are given, the second one is only constructed!

Changing detectors parameters

- Change the geometry easy

/Mokka/init/globalModelParameter parameter_name value

- The steering file can contains several

/Mokka/init/globalModelParameter commands, so the user can setup several parameters before building the detector geometry.

- The values found in the steering file overwrite all the default values

found in the parameters, model_parameters and setup (if a setup name was given) tables.

- All the geometry will be scaled accordingly new values
- See the parameters to change into db (easy with the Mokka web browser)

!!! Please pay attention to give a valid **parameter name**, the code does not check if it is correct ; if it is not correct the job will not crash, the default value is used.

Changing detectors parameters

/Mokka/init/globalModelParameter

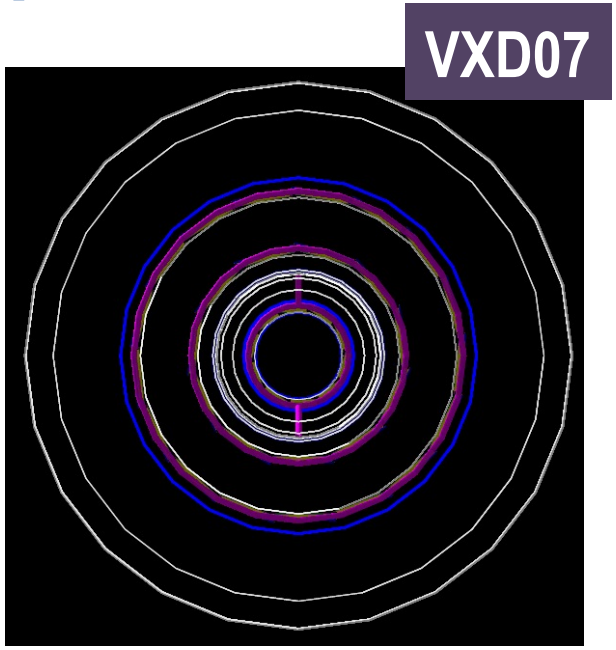
Ecal_barrel_number_of_tower 3

Correct parameter's name Ecal_barrel_number_of_tower**s**

Global model parameter "Ecal_barrel_number_of_tower" set to "3 »
Connecting to the database "models03 »
….
Building sub_detector SEcal05, geometry db VOID, driver SEcal05:
A scalable ILD Ecal mixing Si and/or scintillator
Current parameters for the SEcal05 detector :
Ecal_barrel_number_of_towers = **5**
……

The calculations are performed using the default value 5

Changing Vertex detector parameters



Example of steering commands :

/Mokka/init/globalModelParameter VXD_inner_radius 15

/Mokka/init/globalModelParameter VXD_outer_radius 80

/Mokka/init/globalModelParameter VXD_radius_r1 15
/Mokka/init/globalModelParameter VXD_radius_r2 26
/Mokka/init/globalModelParameter VXD_radius_r3 37
/Mokka/init/globalModelParameter VXD_radius_r4 48
/Mokka/init/globalModelParameter VXD_radius_r5 60
/Mokka/init/globalModelParameter VXD_length_r1 50

/Mokka/init/globalModelParameter VXD_active_silicon_thickness 0.05

/Mokka/init/globalModelParameter VXD_side_band_electronics_width 1

/Mokka/init/globalModelParameter TUBE_central_thickness 0.2

/Mokka/init/globalModelParameter VXD_support_ladder_material "graphite"

/Mokka/init/globalModelParameter VXD_support_ladder_thickness 0.05

/Mokka/init/globalModelParameter VXD_end_electronics_thickness 0.05

/Mokka/init/globalModelParameter VXD_cryostat_option 0

Changing TPC parameters

TPC_Ecal_Hcal_barrel_halfZ

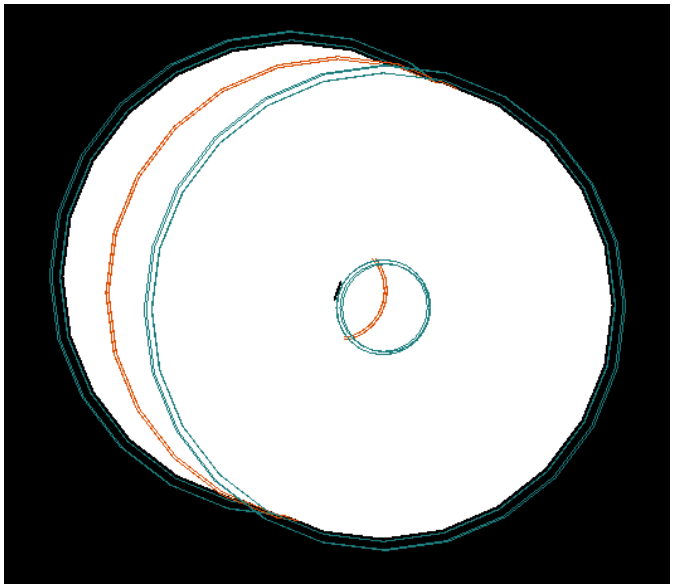
TPC_inner_radius

TPC_outer_radius

TPC_pad_height

TPC_pad_width

TPC_max_step_length

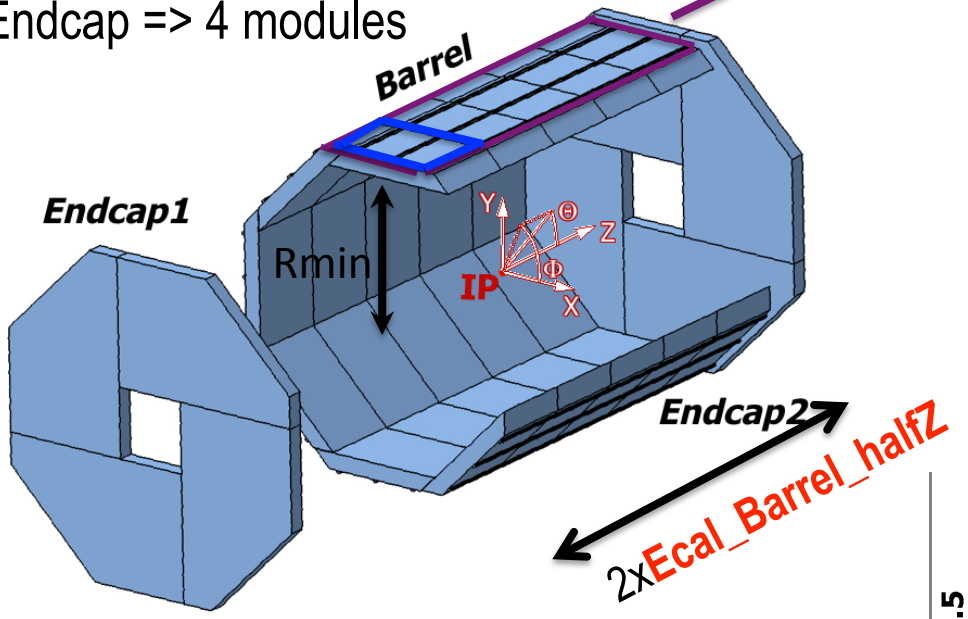


Changing ECAL geometry parameters

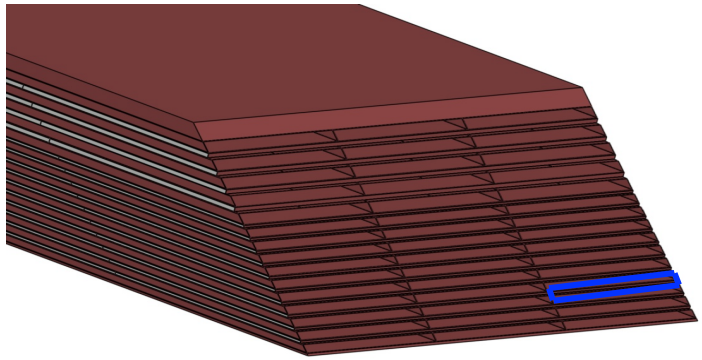
ECAL geometry in the code

2 Endcaps, at z and at -z
Endcap => 4 modules

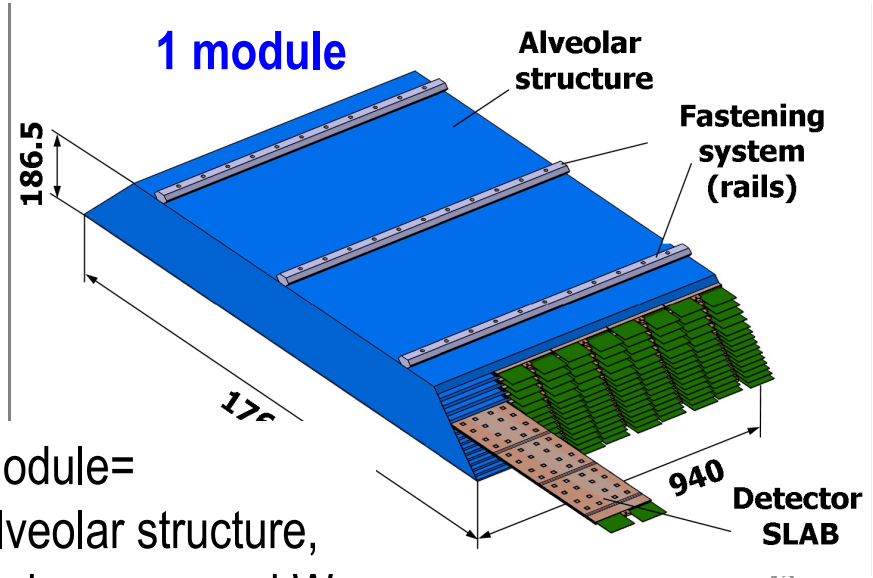
Stave = 5 modules



Barrel
8 staves
1 stave = 5 modules
1 module = **5 towers**
1 tower=1slab/alveolus



1 module



Module=
Alveolar structure,
carbon wrapped W

Ecal geometry: the slab

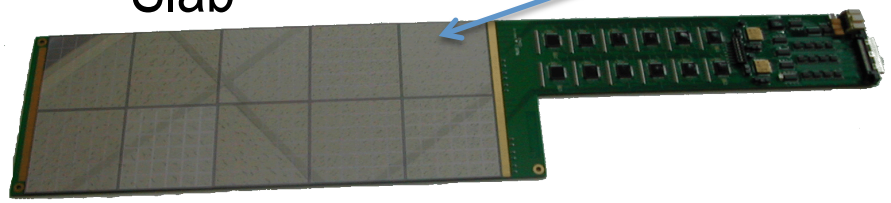
1 slab:
H structure=2 active materials (Sillicon or scintillator)

- Absorber material (Pb or W)

Hamamatsu wafer

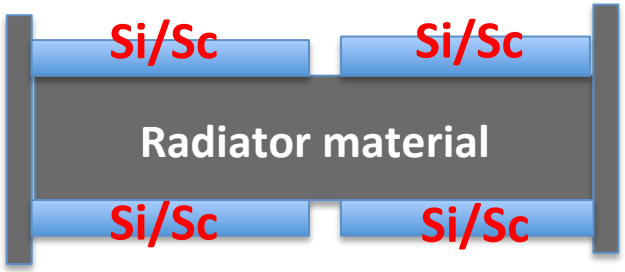
Cells size

Slab



Si/Sc thickness

Radiator thickness
Radiator material



Main parameters that could be changed

Ecal_Barrel_halfZ

Ecal_barrel_number_of_towers

Ecal_cells_size

Ecal_radiator_layers_set1_thickness

Ecal_radiator_layers_set2_thickness

Ecal_radiator_layers_set3_thickness

Ecal_radiator_material

Ecal_nlayers1, Ecal_nlayers2, Ecal_nlayers3 (of the active material)

Ecal_Si_thickness/Ecal_Sc_thickness

- Hard-coded parameters in a Ecal Barrel:
  - Number of staves, Number of modules in a stave
- Calculated parameters: Ecal_inner_radius, Ecal_outer_radius

Main parameters that could be changed

For optimization study of wafer dimension take into account:

Ecal_Barrel_halfZ

Ecal_barrel_number_of_towers

Ecal_cells_size

2 wafers in z direction

Wafer_dimension

Example of ECAL part in steering file

/Mokka/init/detectorModel ILD_o2_v06

/Mokka/init/globalModelParameter **Ecal_Sc_Si_mix** 000000000000000

/Mokka/init/globalModelParameter Ecal_nlayers1 20

/Mokka/init/globalModelParameter Ecal_nlayers2 9

/Mokka/init/globalModelParameter Ecal_nlayers3 0

/Mokka/init/globalModelParameter Ecal_radiator_layers_set1_thickness 2.1

/Mokka/init/globalModelParameter Ecal_radiator_layers_set2_thickness 4.2

/Mokka/init/globalModelParameter Ecal_radiator_layers_set3_thickness 0

/Mokka/init/globalModelParameter Ecal_radiator_material tungsten

/Mokka/init/globalModelParameter **Ecal_Si_thickness** = 0.5

Switch from Si to Scintillator

The global model parameter **"Ecal_Sc_Si_mix »** allows to choose the sensitive detecttors. It is a set of numbers - one for every layer pair - that can each have the following values

- **0** - both layers are made of silicon « all sillicon »
- **1** - both layers have scintillator strips oriented longitudinally in the slab (along the larger dimension of the slab)
- **2** - both layers have scintillator strips oriented transverse to the slab (along the smaller dimension of the slab)
- **3** - the first layer has scintillator strips oriented longitudinally in the slab (along the larger dimension of the slab) - the second layer has scintillator strips oriented transverse to the slab (along the smaller dimension of the slab)
- **4** - the first layer has scintillator strips oriented transverse to the slab (along the smaller dimension of the slab) - the second layer has scintillator strips oriented longitudinally in the slab (along the larger dimension of the slab)

Switch from Si to Scintillator

Since SEcal05 driver, thanks to Daniel Jeans implementation in the code, it is possible to mix silicon and scintillator within the same alveolus. The corresponding configurations are:

- **5**: sillicon – Scintillator strip allong X
- **6**: sillicon – Scintillator strip allong Z
- **7**: Scintillator strip allong X - sillicon
- **8**: Scintillator strip allong Z - sillicon

Both the Barrel and EndCaps use the same 'Ecal_Sc_Si_mix' parameter, so they will have exactly the same combination of sensitive layers.

Examples:

1) Ecal_nlayers1 20, Ecal_nlayers2 9, Ecal_nlayers3 0 => **((20+9)+1)/2=15 pairs**

'All sillicon' 15 layer pairs - « 000000000000000 » (this is a default value)

'All scintillator' 15 layer pairs - « 333333333333333 »

Si and Si – « 030303030303030 »

2) Ecal_nlayers1 18, Ecal_nlayers2 7, Ecal_nlayers3 0 => **((18+7)+1)/2=13 pairs**

'All sillicon' 13 layer pairs - « 0000000000000 »

ECAL « all scintillator »

- The scintillator strips version uses the same geometry as the 'all silicon' version: the same shapes and dimensions of barrel and endcap modules, towers. Only the 'H structures' are filled with different sensitive materials and connected ingredients (the scintillator layers have no 'ground' and glue layers, for example).

- Each 'H structure' has two slabs that are the same (either both silicon or both scintillator - even if the scintillator layers can be oriented differently), as shown above.

- The scintillator strips are made of polystyrene and are coated by a reflector film made of G4_MYLAR. The multi-pixel photon counters are also implemented, and are made of polystyrene.

ECAL « all scintillator »

- The parameters connected to the scintillator version and their default values are:

Ecal_Sc_thickness
Ecal_Sc_reflector_thickness
Ecal_Slab_Sc_PCB_thickness
Ecal_Sc_MPPC_breadth
Ecal_MPPC_size
Ecal_Sc_N_strips_across_module
Ecal_Sc_number_of_virtual_cells

- Separate hit collections are created for silicon and scintillator. For scintillator there are also two hit collections: one for longitudinal strips and one for transverse strips, since the meaning of the 'cell' indices is different and the treatment - corrections - depend on the direction - along strips or transverse to the strips.

Confident and free when creating with Mokka

Check modifications with G4/Mokka tools

Where to find informations and help

Check modifications with G4/Mokka tools

- Check if Mokka took into account what we asked
- Check if what Mokka did is OK

See the information in the Mokka log information

- model name
- parameters values
- detectors build order

Check the geometry

Methods for checking the geometry

- Geant4 Tools for geometry overlapping
  - Idle>/geometry/test/grid_test
  - Option pSurfChk=true in G4PVPlacement and in G4PVParameterised
  - Visual Overlap Checking
  - Propagate a geantino trough volumes and check using tracking verbose tools
- Check the geometry architecture using ASCII Tree
- Check the geometry using the hits

Conclusion

- The steering file allows to change easelly the detector geometry
- We presented basics elements for a **steering file**

- We learned how by using the commands in the steering file
  - to modify the model by dropping or replacing some sub-detectors
  - to modify model sub-detectors by changing their parameters
  => focus on VTX, ECAL and TPC

Exercices

Try exercices that we reviwed during the presentation
1. Working with models
commands are in exercices_WorkingWithCommands.txt
2. Explore the parameters for different subdetectors in the web db browser
http://www-flc.desy.de/ldcoptimization/tools/mokkamodels.php
3. Run Mokka for vxd, tpc, ecal detectors

Credits

- Page 2 : photo

http://www.linkedin.com/today/post/article/20131202064053-175081329-how-to-become-a-master-of-creativity


ÉCOLE
POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY


cnrs
advancing the frontiers

MOKKA


UNIVERSITY OF
CAMBRIDGE


東京大学
THE UNIVERSITY OF TOKYO


DESY

Backup slides

Hard coded parameters default values

See **Control** class in **Kernel** svn directory

Some examples:

G4String Control::DBHOST=**"pollin1.in2p3.fr"**;

G4String Control::USER=**"consult"**;

G4String Control::DBPASSWD=**"consult"**;

G4String Control::PhysicsListName = **"QGSP_BERT"** ;

G4double Control::RadiatorRangeCut = 0.005 * mm;

G4double Control::PCBRangeCut = 0.005 * mm;

G4double Control::ActiveRangeCut = 0.005 * mm;

G4double Control::RangeCut = 0.005 * mm;

G4double Control::TPCCut = 10 * MeV;

Geometry visuallisation with GDML

- Run Mokka in interactive mode (do not use BatchMode), type commands on line
- Get the list of logical volumes namesfor all geometry

/Mokka/Visu/Detector/ListGeometryTree

/Mokka/Visu/Detector/ListGeometryTree detector_class

Example: /Mokka/Visu/Detector/ListGeometryTree ecal

- To get the logical volume SlabLogical

/Mokka/Visu/Detector/DumpGDML detector-class logical-volume-name

Example: /Mokka/Visu/Detector/DumpGDML ecal SlabLogical

Obtain SlabLogical.gdml

- To get all the geometry

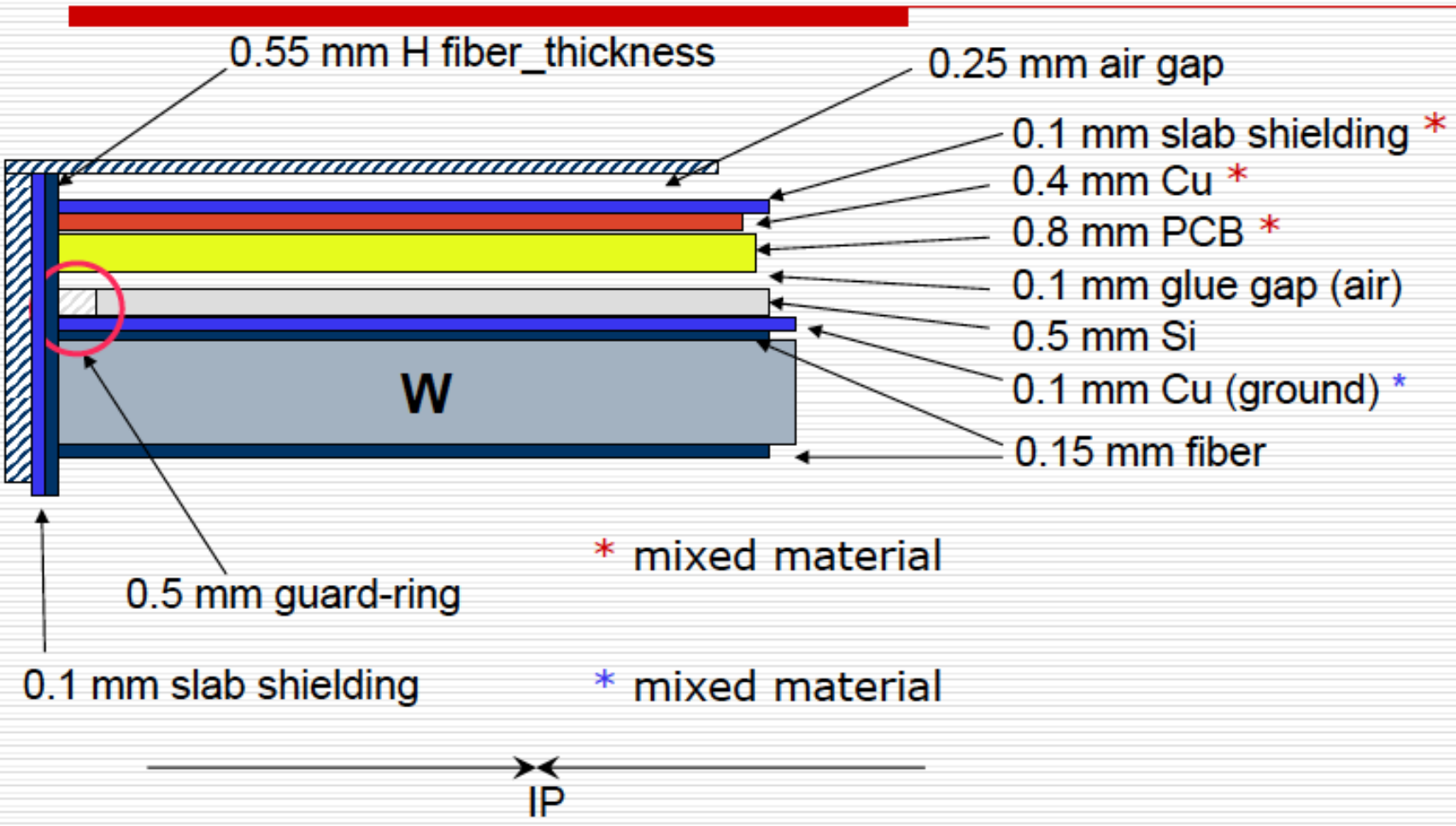
/Mokka/Visu/Detector/DumpGDML

- To visualize the geometry from *.gdml with ROOT:

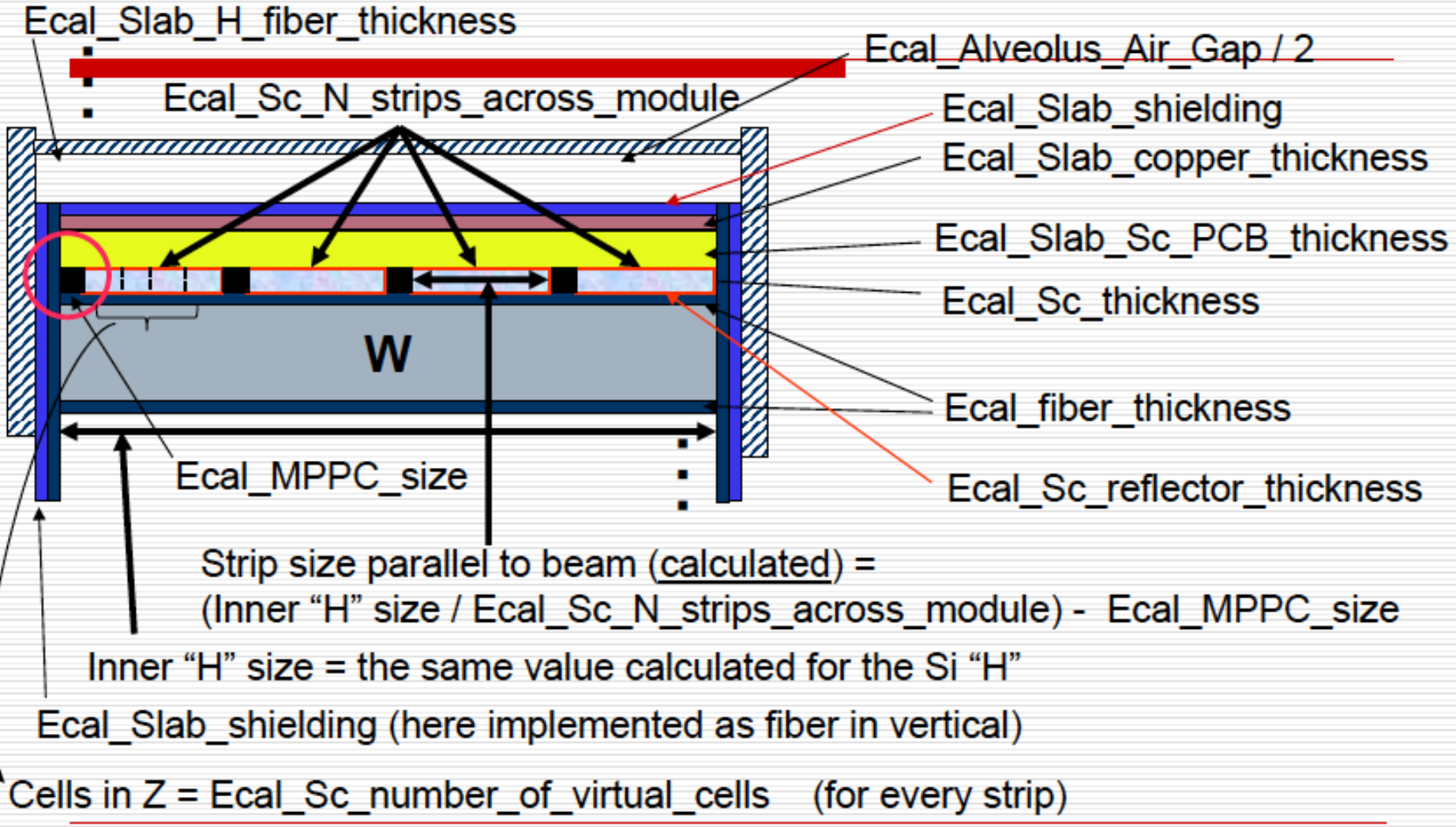
root[]TGeoManager::Import("logical-volume-name.gdml")

root[]gGeoManager->GetTopVolume()->Draw("ogl")

Si layers: Alveoli & “H” slab structure

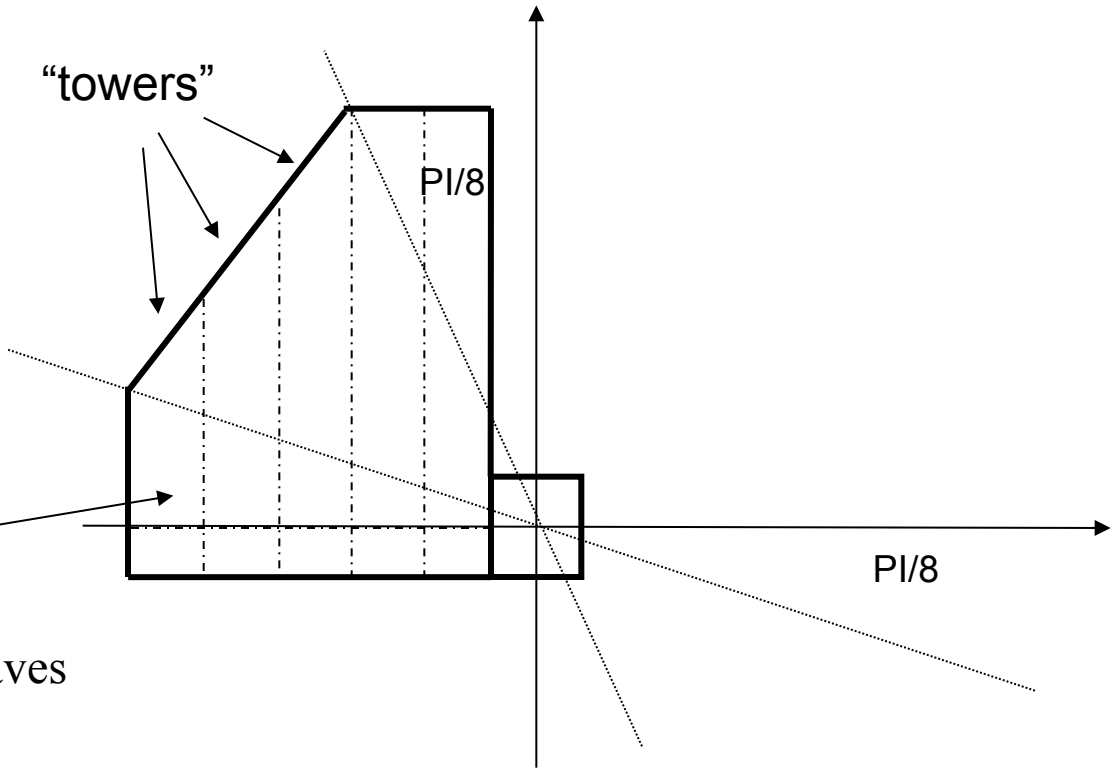


Parameters for Alveoli & "H" slab structure for Sc strips, Z direction (parallel to beam)



Ecal EndCap Module

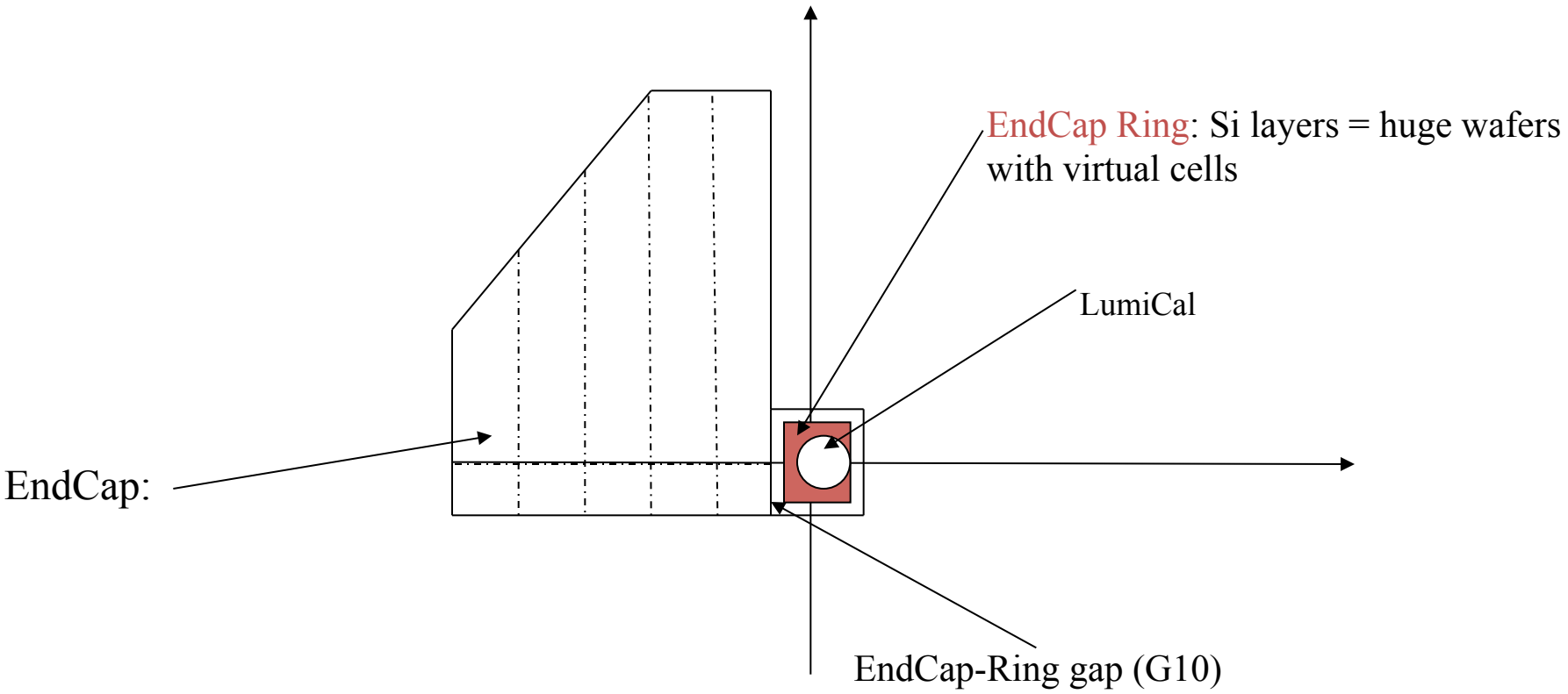
Paulo Mora de Freitas, Gabriel Musat



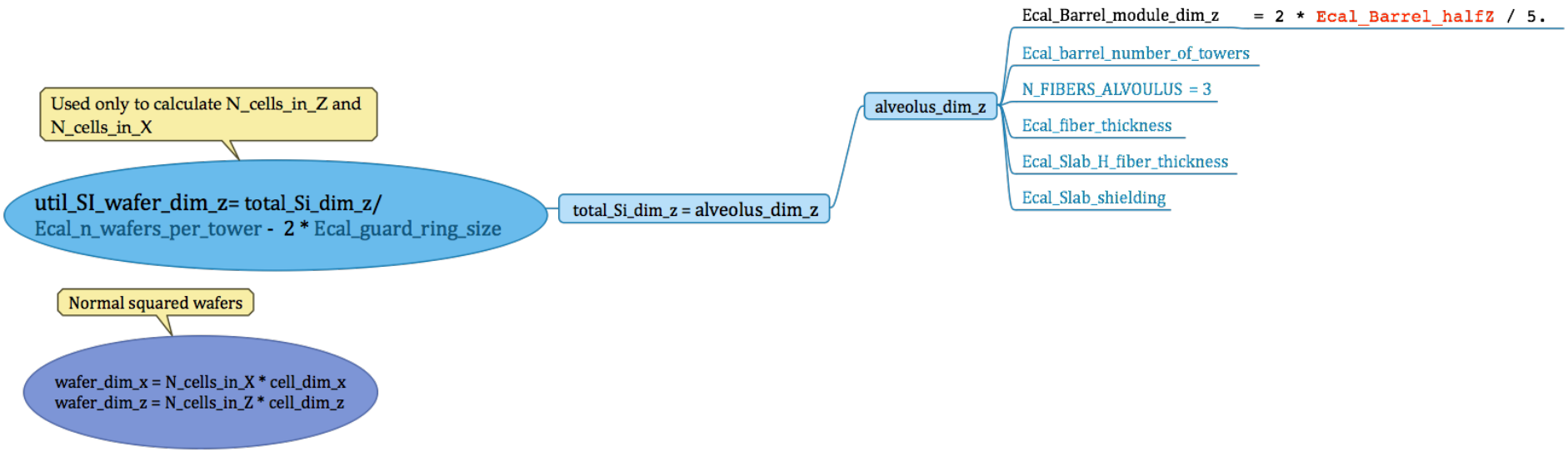
EndCap:
4 staves / module
no gaps between staves

Ecal EndCap Ring

Paulo Mora de Freitas, Gabriel Musat



Wafer dimension calculation



Change the Tracker

SiD detector models

model	sub_detector
LDC00	SEcal01
LDC00	SHcal01
LDC00	SCoil01
LDC00	SYoke01
LDC00	SField01
LDC00	STpc01
LDC00	vxd00
LDC00	ftd01
LDC00	tube01
LDC00	mask04
LDC00	sit00
LDC00	LumiCalS

Model SiD01

model	sub_detector
SiD01	SEcal01
SiD01	SHcal01
SiD01	SCoil01
SiD01	SYoke01
SiD01	SField01
SiD01	**SiDFwd00**
SiD01	**SiDBar00**
SiD01	vxdSiD00
SiD01	ftdSiD00
SiD01	tubeSiD01

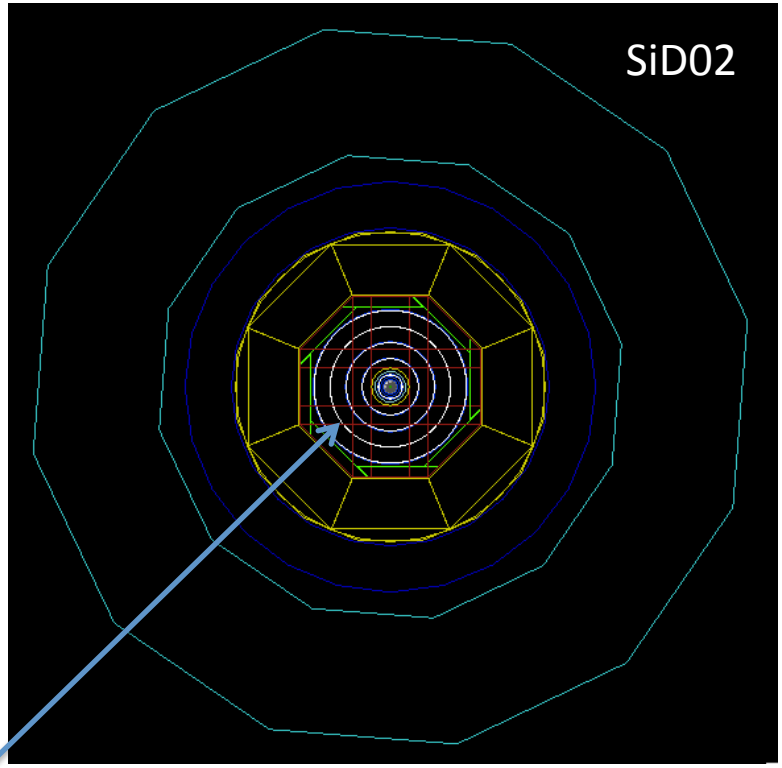
Model SiD02

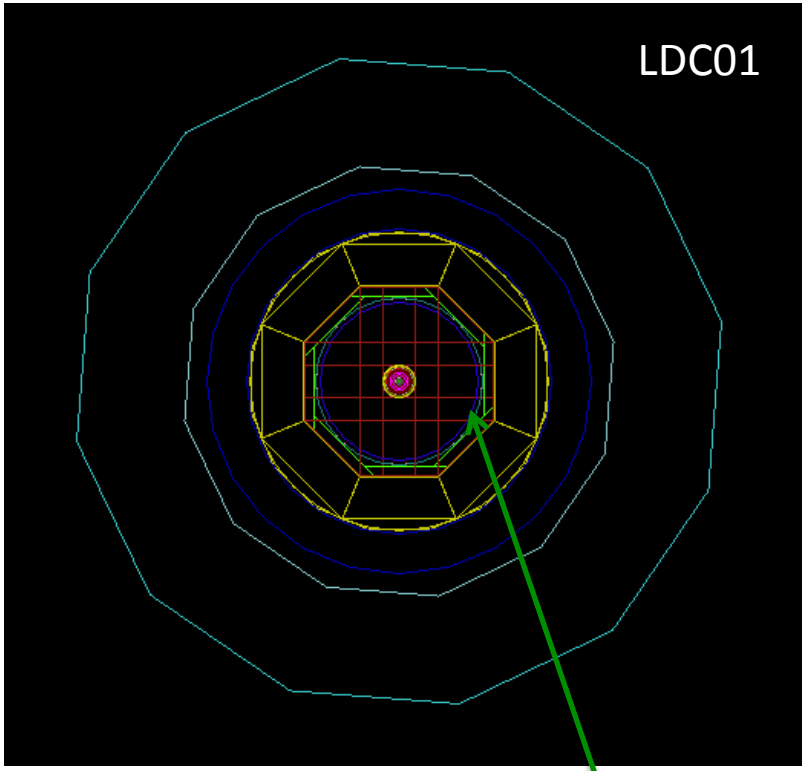
model	sub_detector
SiD02	SEcal01
SiD02	SHcal01
SiD02	SCoil01
SiD02	SYoke01
SiD02	SField01
SiD02	**SiDFwd00**
SiD02	**SiDBar00**
SiD02	vxdSiD00
SiD02	ftdSiD00
SiD02	tubeSiD01

- **SiDBar00** implements the Si Tracker barrel and **SiDFwd00** the Si Tracker end caps
- tubeSiD01, vxdSiD00 and ftdSiD00 are LDC devices adapted for the SiD dimensions
- ftdSiD00 works as the Si vxd disks for these models

SiD detector models

- Two models SiD01 and SiD02 **/Mokka/init/detectorModel SiD02**





The SiD tracker is composed of five barrels with five endplates.
The z extent of the barrels increases with radius.

- SiDBar00 and SiDFwd00 classes are in Mokka/trunk/source/Geometry/SiD

Special model for testing SiD Tracker

model	sub_detector
SiD02	SEcal01
SiD02	SHcal01
SiD02	SCoil01
SiD02	SYoke01
SiD02	SField01
SiD02	**SiDFwd00**
SiD02	**SiDBar00**
SiD02	vxdSiD00
SiD02	ftdSiD00
SiD02	tubeSiD01

Model SiDTracker

model	sub_detector
SiDTracker	SiDFwd00
SiDTracker	SiDBar00
SiDTracker	vxdSiD00
SiDTracker	ftdSiD00
SiDTracker	tubeSiD01

SiD models

To get more information about SiD models in Mokka:

- mokka.in2p3.fr: Go to **Detector Models -> SiD detector models**
- https://confluence.slac.stanford.edu/display/ilc/sidmay05