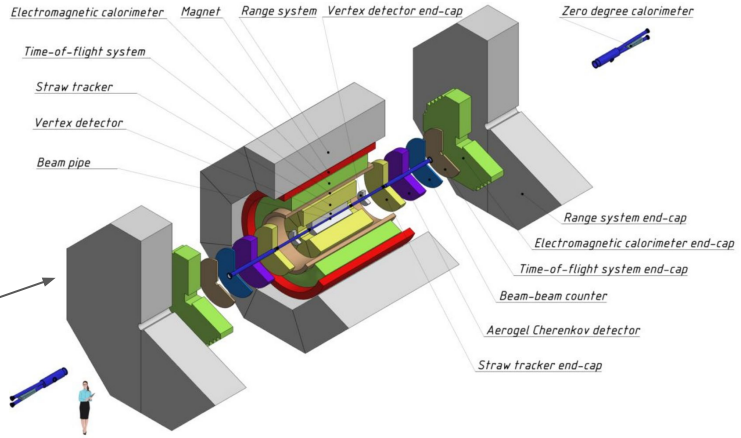
Sampo: software platform for SPD data processing

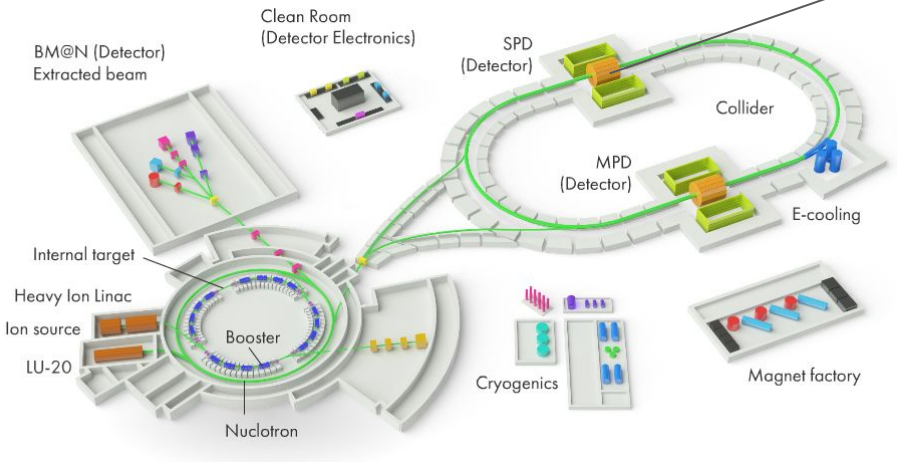
Simbiriatin Lev
lev.simbiriatin@jinr.ru

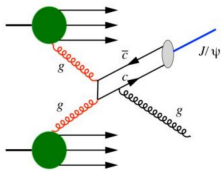
GRID
2025

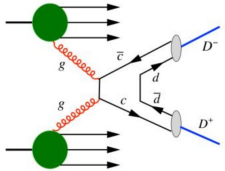
SPD experiment

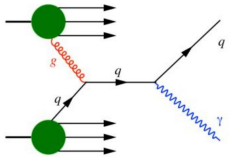
The main purpose of the SPD experiment is to extract information about gluon transverse momentum dependent parton distribution functions (EMD Pds) in polarized pp ($\sqrt{s}$ = 27 GeV) and dd ($\sqrt{s}$ = 13.5 GeV) collisions. The expected luminosity is 10^{32} cm^{-2} s^{-1}.



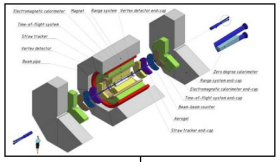




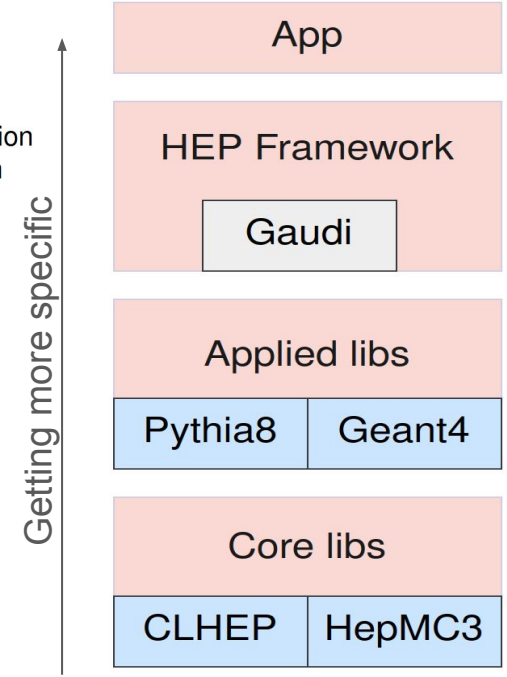




Software in HEP



SPD data processing pipeline



Typical HEP software stack

Pythia8 - primary particles generator

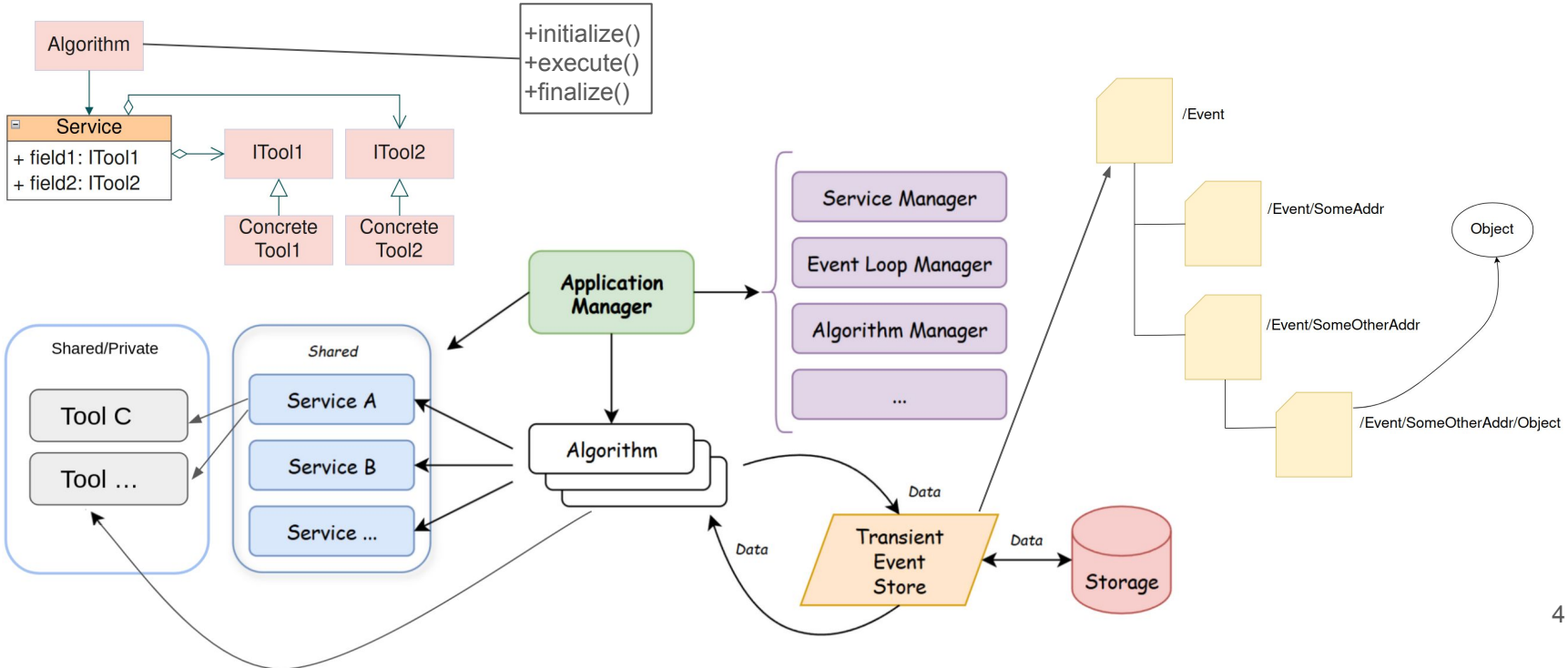
HepMC3 - generated event format

GeoModel - geometry description

Geant4 - simulation of particles traversing through detector

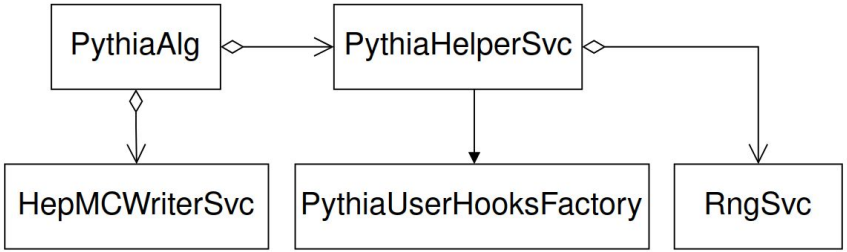
Gaudi architecture

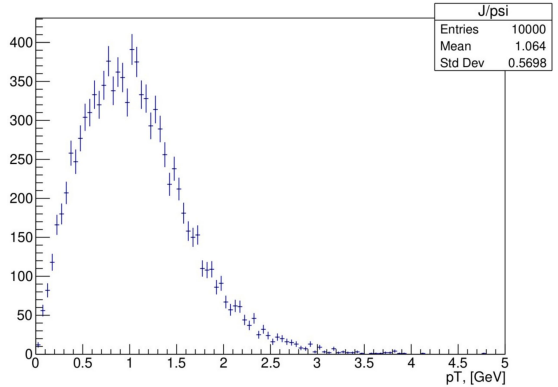
Concept: Data objects manipulated by Algorithms that are launched on per-event basis. Algorithms, Services and Tools are dynamically configurable via python script.



Pythia8

Concept: PythiaHelperSvc creates Pythia8 generator instances, equips them with random number generators and applies all requested user hooks. Generation is performed in PythiaAlg.



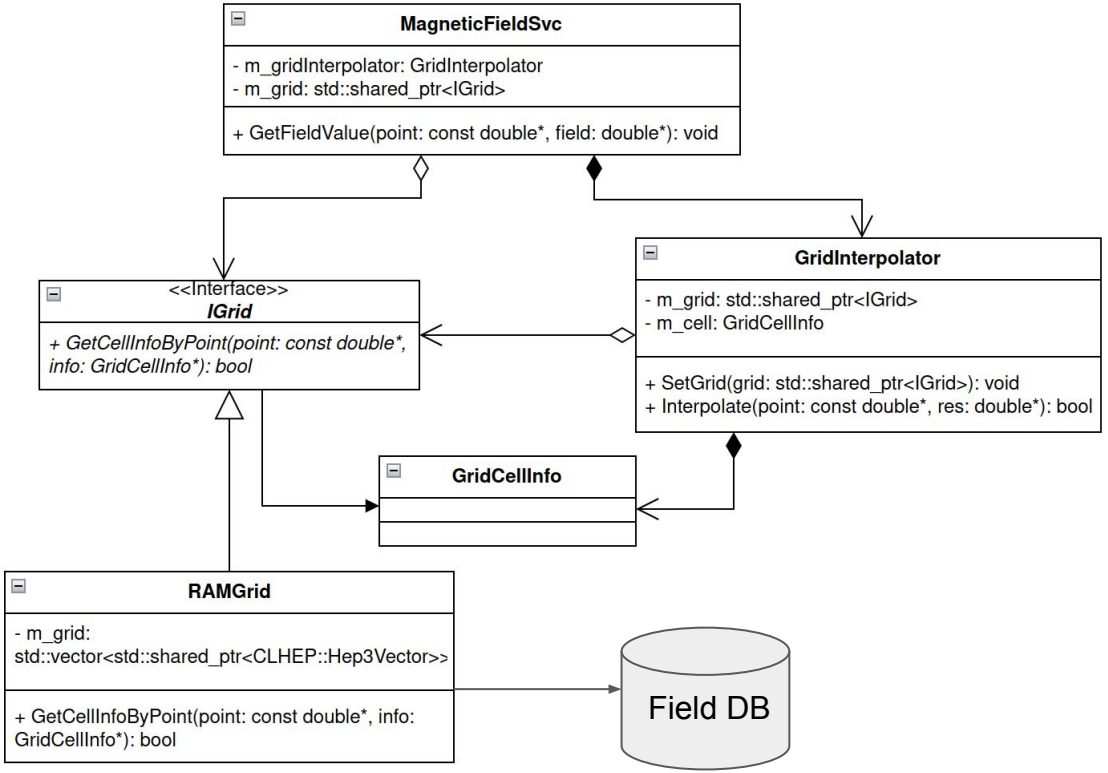


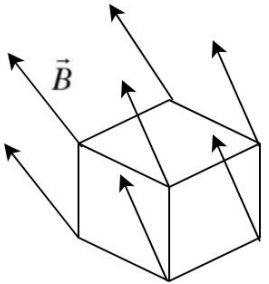
pp, √s = 27 GeV

```python
from Configurables import PythiaAlg, PythiaHelperSvc, HepMCWriterSvc

from Gaudi.Configuration import ApplicationMgr

evt_max = 10000
evt_sel = "NONE"

writer = HepMCWriterSvc()
writer.outputFormat="ROOTTree"
writer.outputPath="output.root"

my_algo = PythiaAlg("Pythia")
helper = PythiaHelperSvc()
helper.fileName = \
    "/workspaces/sampo/Generators/Pythia8/options/testconf.cmnd"
helper.printBanner = False

# create ApplicationMgr and start Gaudi app (C++)
ApplicationMgr(
    EvtMax=evt_max,
    EvtSel=evt_sel,
    TopAlg=[my_algo]
)
```

Magnetic field

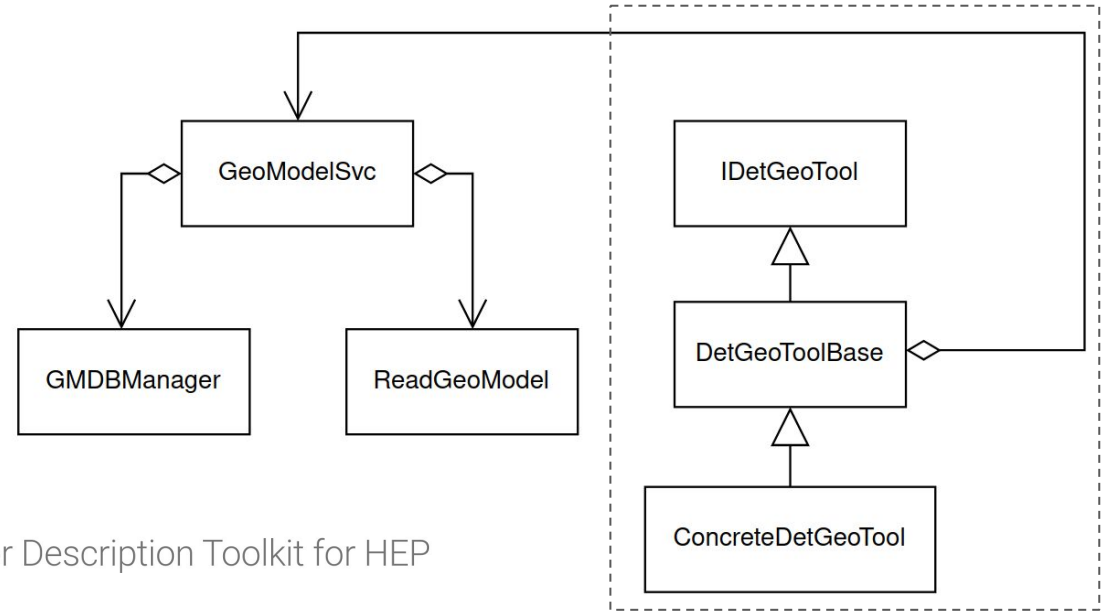
Concept: Field map is stored in DB, one gets field value via MagneticFieldSvc.



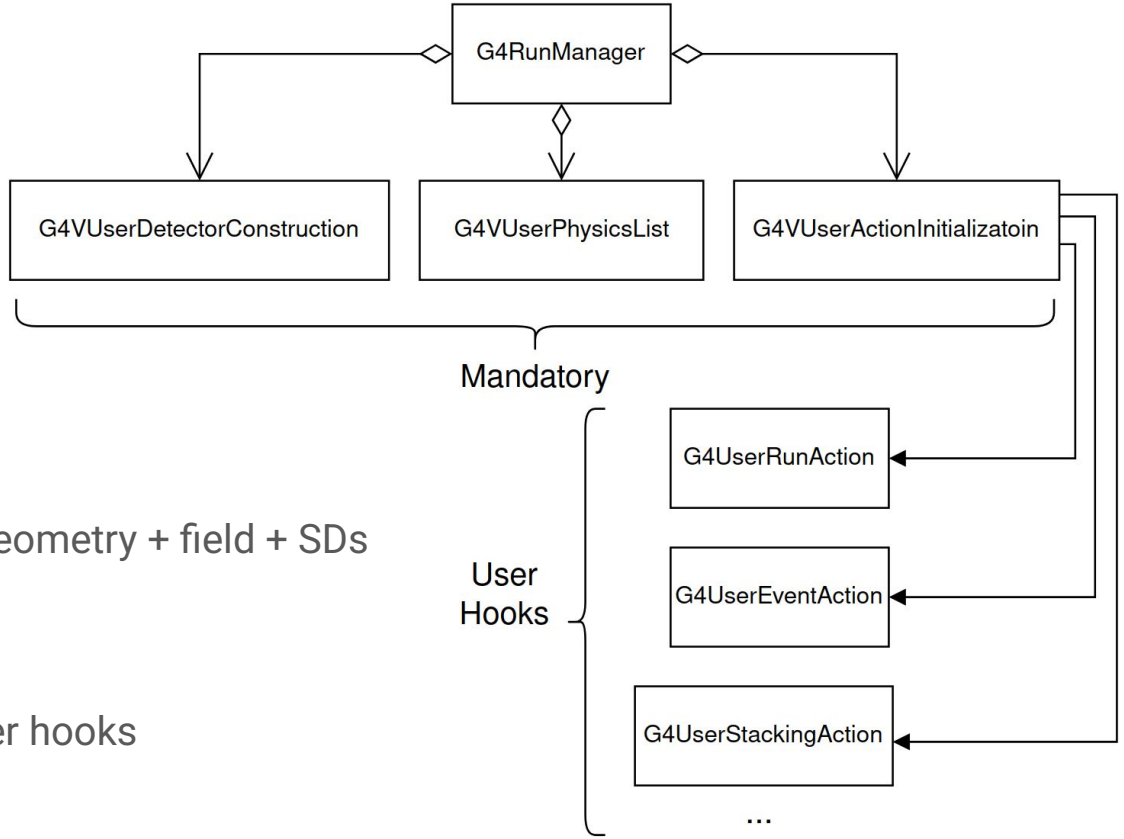


GeoModel

Concept: GeoModelSvc has access to geometry DB via GMDBManager. ReadGeoModel for geometry tree building. DetGeoTools keep link to GeoModelSvc and serve to provide information on detector subsystems.



Geant4



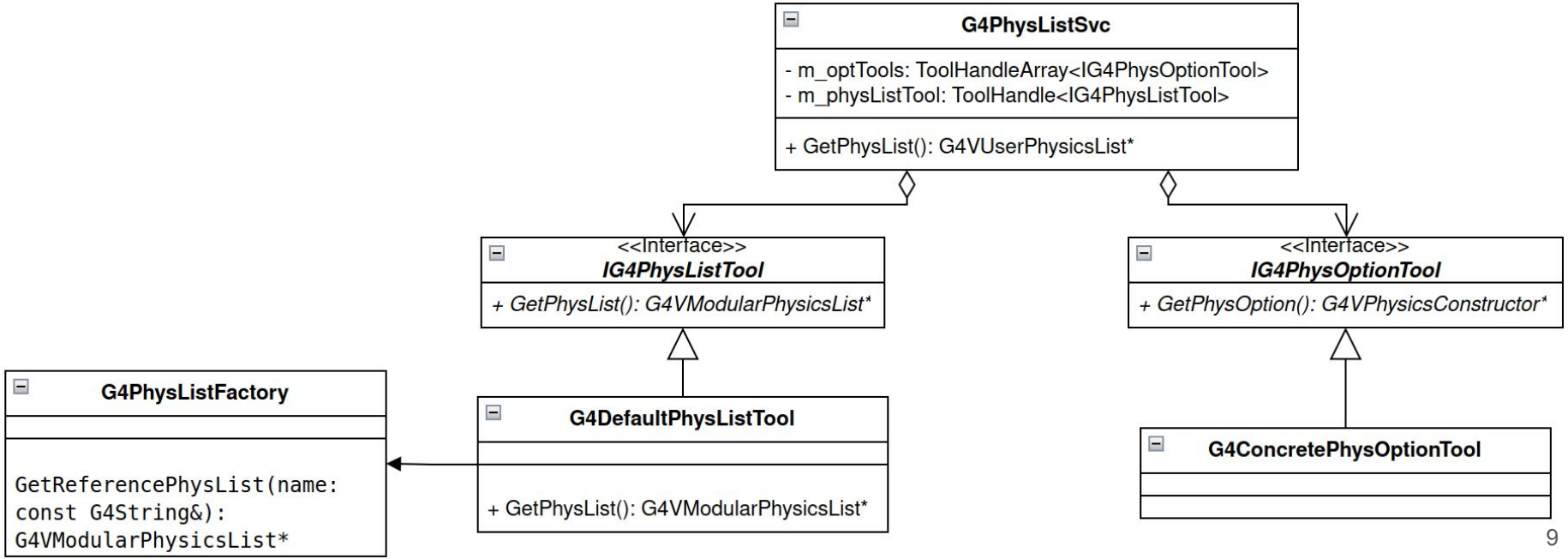
G4VUserDetectorConstruction -> geometry + field + SDs

G4VUserPhysicsList -> physics

G4VUserActionInitioalization -> user hooks

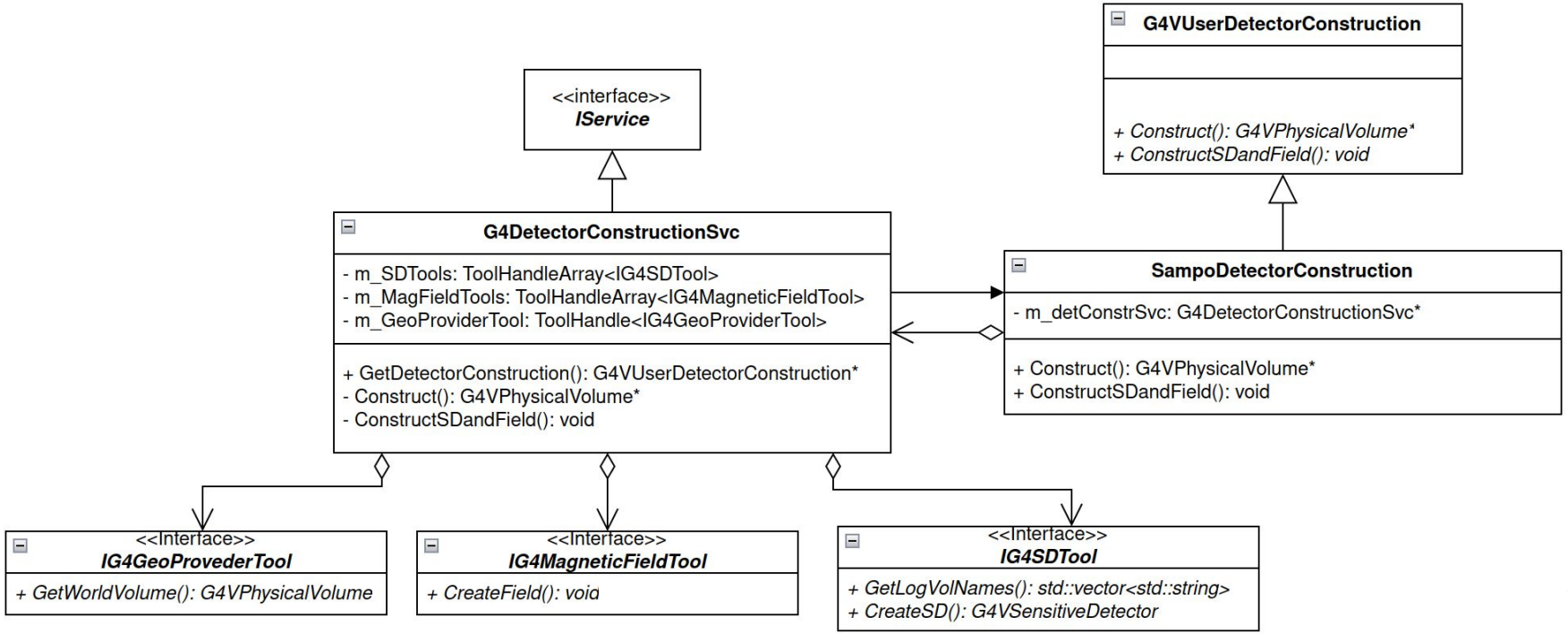
G4PhysListSvc

Concept: G4PhysListSvc is used to construct G4VUserPhysicsList. G4ConcretePhysListTool constructs PhysList itself and then hooks constructed by G4ConcretePhysOptionTools are applied. Two types of options: Register and Replace.



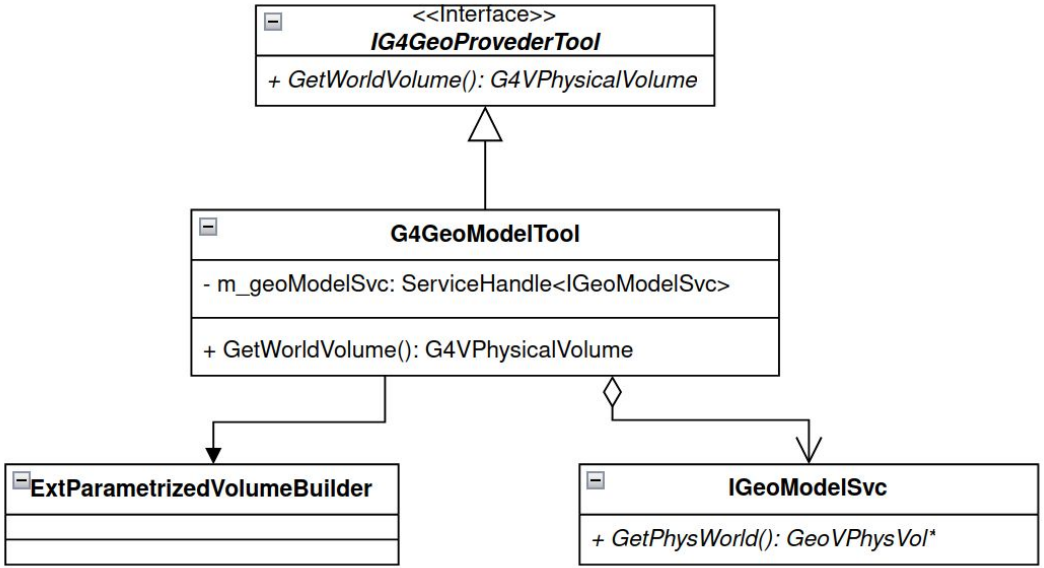
G4DetectorConstructionSvc

Concept: G4DetectorConstructionSvc wraps itself into adapter class SampoDetectorConstruction. Main work of detector and field creation is delegated to contained tools.



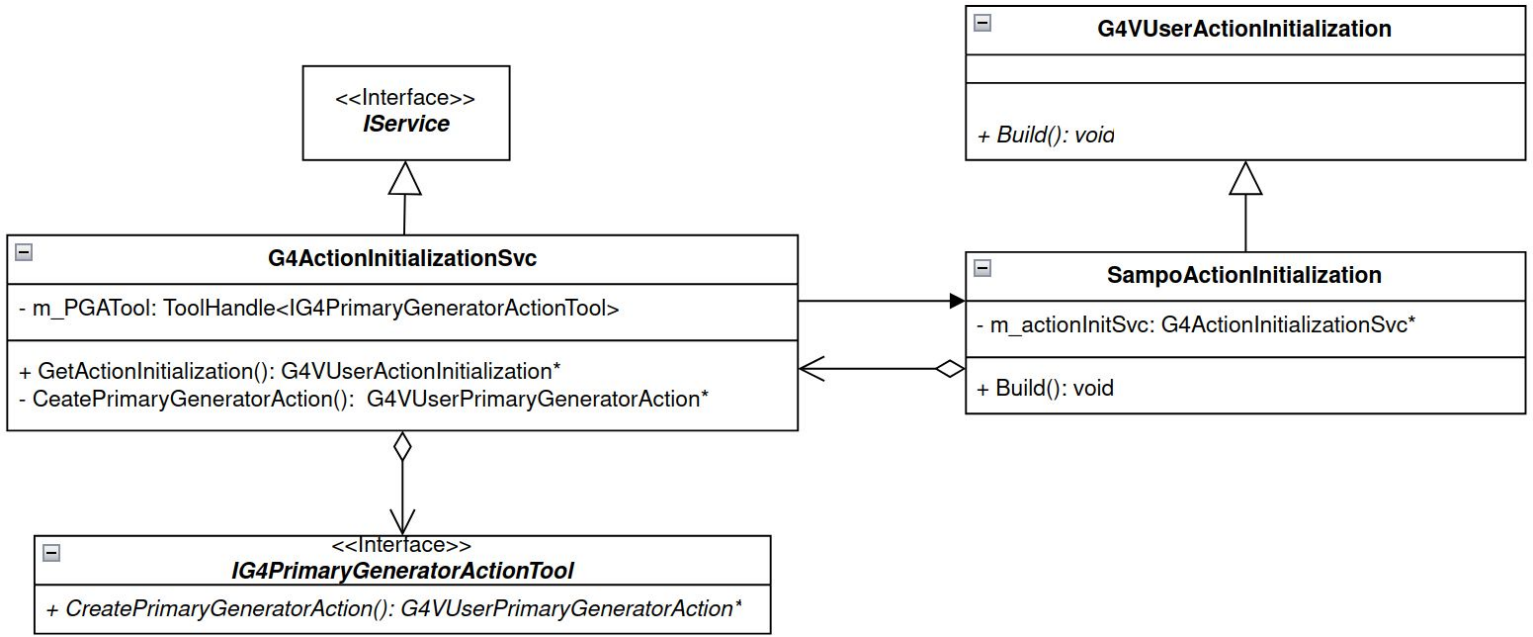
G4GeoModelTool

Concept: GeoModelTool has a link to GeoModelSvc and knows how to convert GeoModel volumes to Geant4 volumes.



G4ActionInitializationSvc

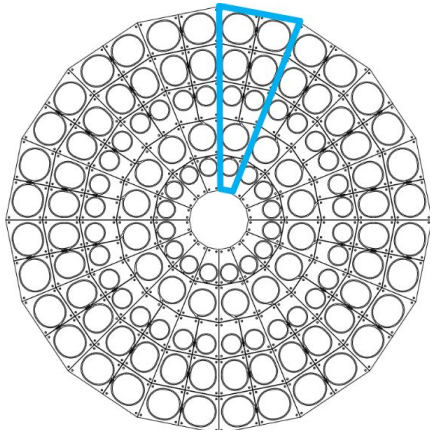
Concept: G4ActionInitializationSvc wraps itself into adapter class SampoActionInitialization. Main work of user hooks creation is delegated to contained tools.

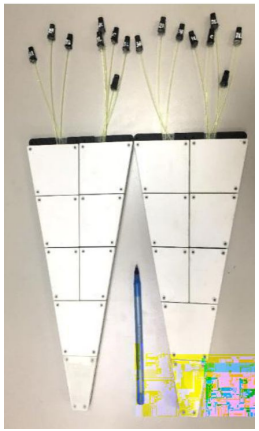


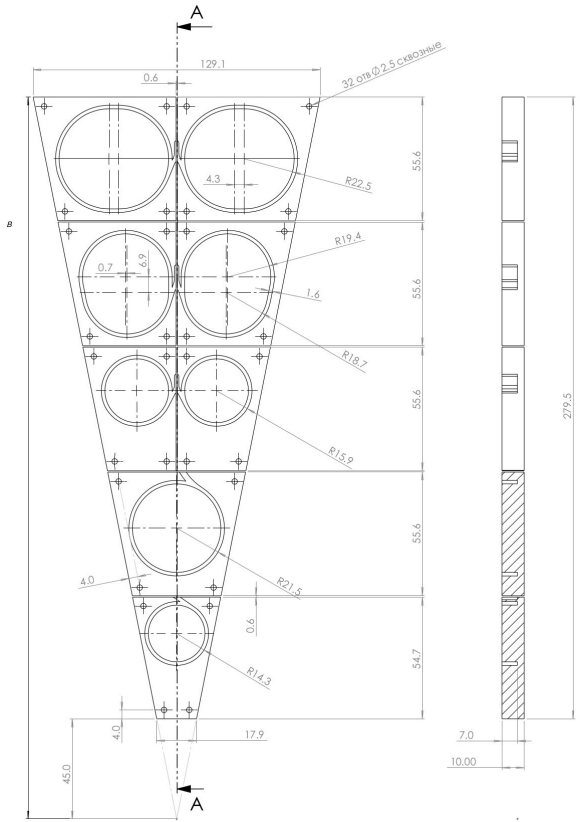
Beam-Beam Counter (BBC) Example

Primary goal - measurement of azimuthal asymmetries in inclusive charged particles production to determine the polarization of pp beams.

During Ion program - centrality measurements.

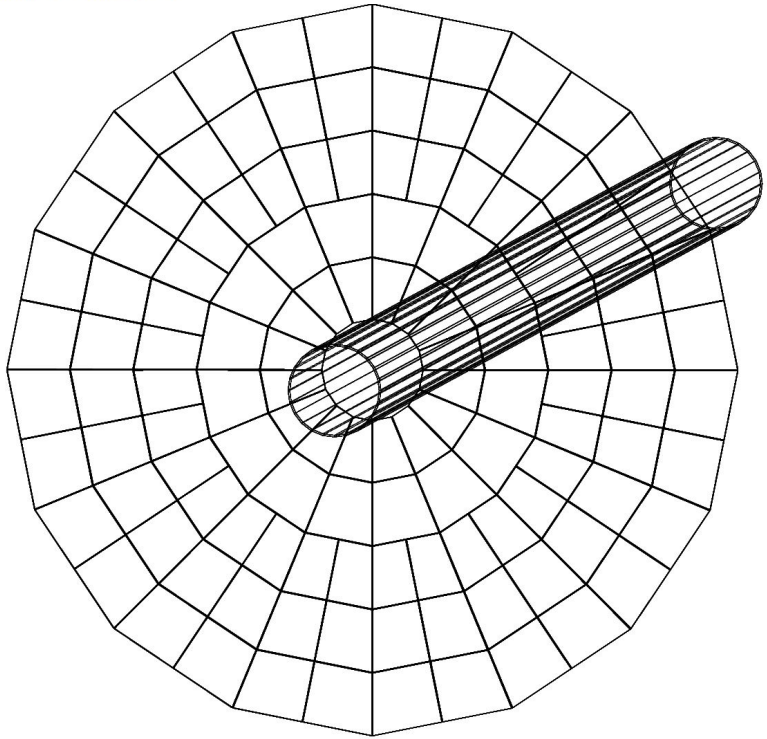






Python JobOptions: BBC simulation

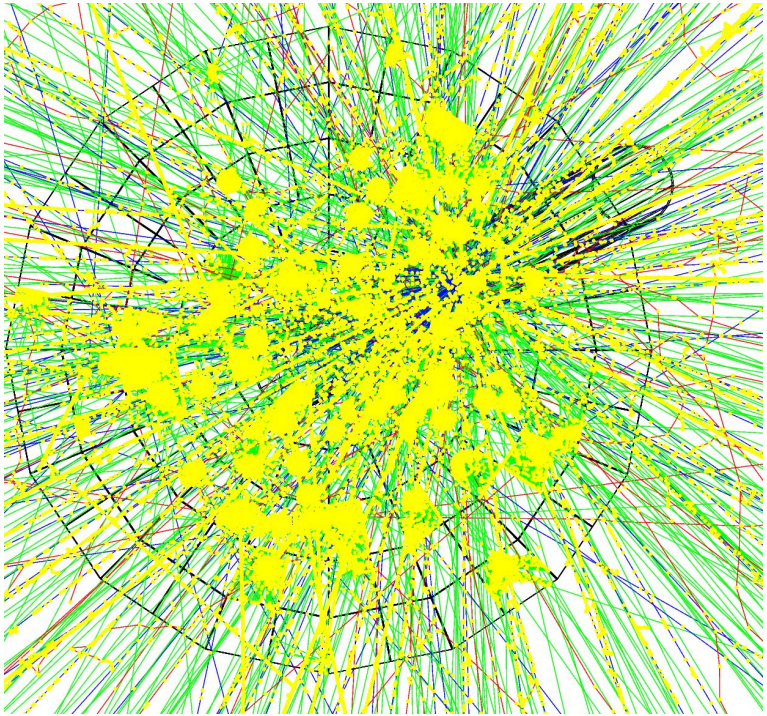
```
1  from Configurables import (HepMCReaderSvc, G4SampoAlg, G4DetectorConstructionSvc,
2                             G4PhysListSvc, G4ActionInitializationSvc, BBCGeoProviderTool,
3                             BBCGeoInserterTool, PipeGeoInserterTool,
4                             G4HepMCPrimaryGeneratorActionTool,
5                             BBCPhysListTool, BBCSDTool, GaudiPersistency)
6
7  from Configurables import Gaudi__RootCnvSvc as RootCnvSvc
8  from Gaudi.Configuration import *
9
10 # DetectorConstruction setup
11 detConstrSvc = G4DetectorConstructionSvc()
12
13 # BBC detector
14 bbsGeoInserter = BBCGeoInserterTool()
15 bbsGeoInserter.thickness = 10
16 bbsGeoInserter.distance = 3000
17 bbsGeoInserter.radius_0 = 45
18 # also other BBC parameters here
19
20 # NICA Pipe
21 pipeGeoInserter = PipeGeoInserterTool()
22 pipeGeoInserter.thickness = 1.5
23 pipeGeoInserter.distance = 7500
24 pipeGeoInserter.width = 84
25
26 bbcGeoProvider = BBCGeoProviderTool()
27 bbcGeoProvider.GeoInserters = [bbsGeoInserter, pipeGeoInserter]
28 detConstrSvc.GeoProviderTool = bbcGeoProvider
29
```



Python JobOptions: BBC simulation

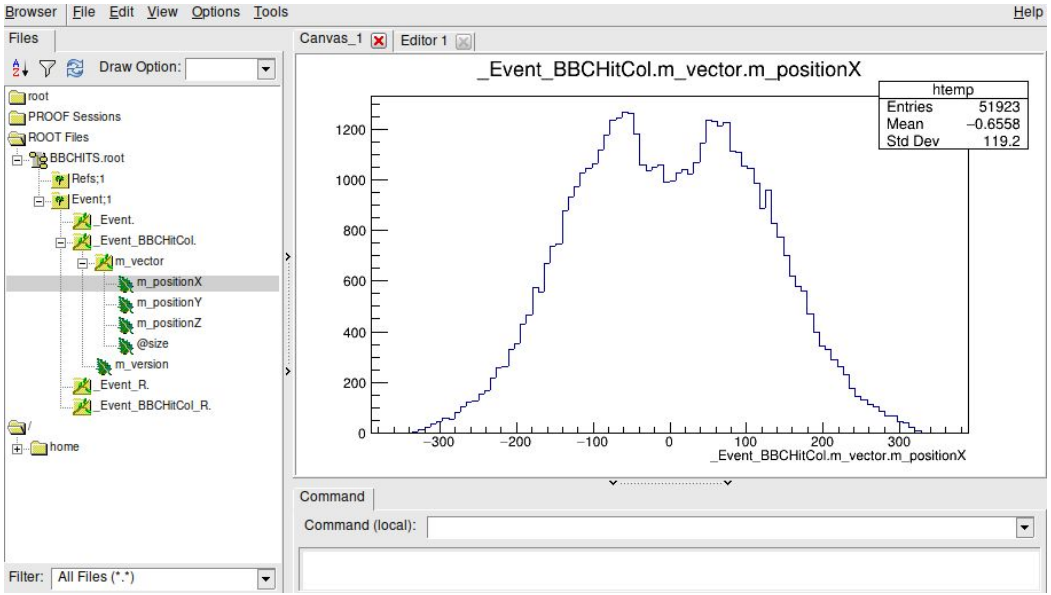
```
30  # PhysList setup
31  physListSvc = G4PhysListSvc()
32  physListSvc.PhysListTool = BBCPhysListTool()
33
34  #HepMC input
35  readerSvc = HepMCReaderSvc()
36  readerSvc.inputFormat = "Ascii"
37  readerSvc.inputFiles = \
38    ["/workspaces/sampo/BBC/data/urqmd.dat"]
39
40  # ActionInitialization setup
41  ActionInitSvc = G4ActionInitializationSvc()
42  ActionInitSvc.PrimaryGeneratorActionTool = \
43    G4HepMCPrimaryGeneratorActionTool()
44
```

collisions: Xe (3 GeV/N) with W target
generator: URQMD



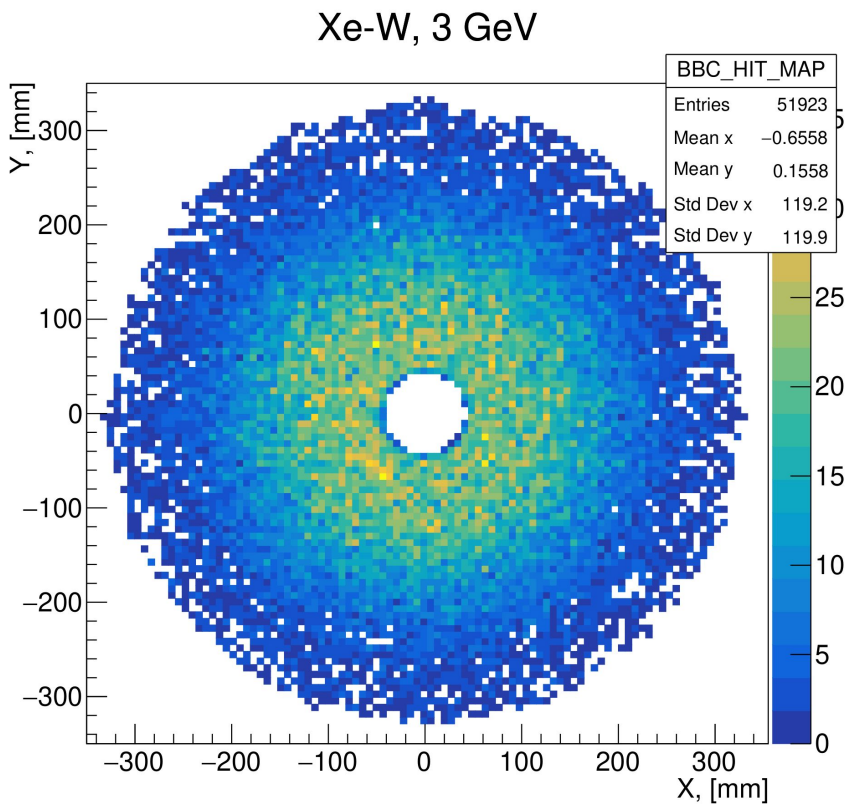
Python JobOptions: BBC simulation

```
45  # Also add SD to G4DetectorConstructionSvc
46  sdTool = BBCSDTool()
47  sdTool.LogVolNames = ["BBC"]
48  detConstrSvc.SDTools = [sdTool]
49
50  g4algo = G4SampoAlg("G4Alg")
51
52  # Setup writing to ROOT file
53  dst = OutputStream("RootDst")
54  dst.ItemList = ["/Event#999"]
55  dst.Output = ("DATAFILE='PFN:BBCHITS.root' "
56                "SVC='Gaudi::RootCnvSvc' "
57                "OPT='RECREATE'")
58  RootCnvSvc(OutputLevel=INFO)
59  GaudiPersistency()
60
61  evt_max = 1000
62  evt_sel = "NONE"
63
64  # create ApplicationMgr and start Gaudi app (C++)
65  ApplicationMgr(
66    EvtMax=evt_max,
67    EvtSel=evt_sel,
68    TopAlg=[g4algo],
69    OutStream=[dst]
70  )
```



Python JobOptions: BBC simulation

```
1  from Configurables import Gaudi__RootCnvSvc as RootCnvSvc
2  from Configurables import GaudiPersistency, BBCHitMapAlg
3  from Gaudi.Configuration import *
4
5  esel = EventSelector(PrintFreq=50, FirstEvent=1)
6  esel.Input = [
7    ("DATAFILE='PFN:BBCHITS.root' "
8    "SVC='Gaudi::RootEvtSelector' "
9    "OPT='READ'")
10 ]
11
12 # Output Levels
13 RootCnvSvc(OutputLevel=INFO)
14
15 GaudiPersistency()
16
17 hitMapAlg = BBCHitMapAlg("BBCHitMapAlg")
18
19 # Application setup
20 app = ApplicationMgr()
21
22 # - Algorithms
23 app.TopAlg = [hitMapAlg]
24 app.EvtMax = -1
25 app.HistogramPersistency = "NONE"
```



Conclusion

- Flexible design
- Pythia8 as event generator
- HepMC3 as event format
- GeoModel for detector geometry description
- Tools for magnetic field description
- Geant4 for detector simulation