

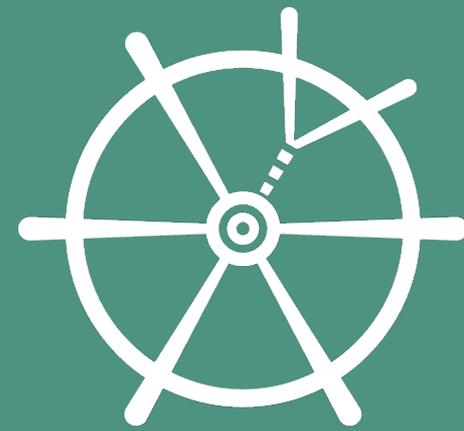
SHiP

*Search for Hidden Particles*

# Plans for SND detector layout optimisation

Jesús P. Márquez Hernández





SHiP

*Search for Hidden Particles*

FairShip





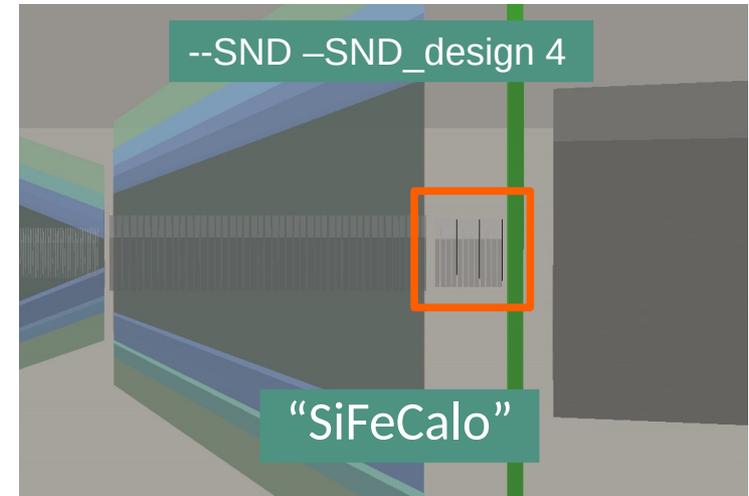
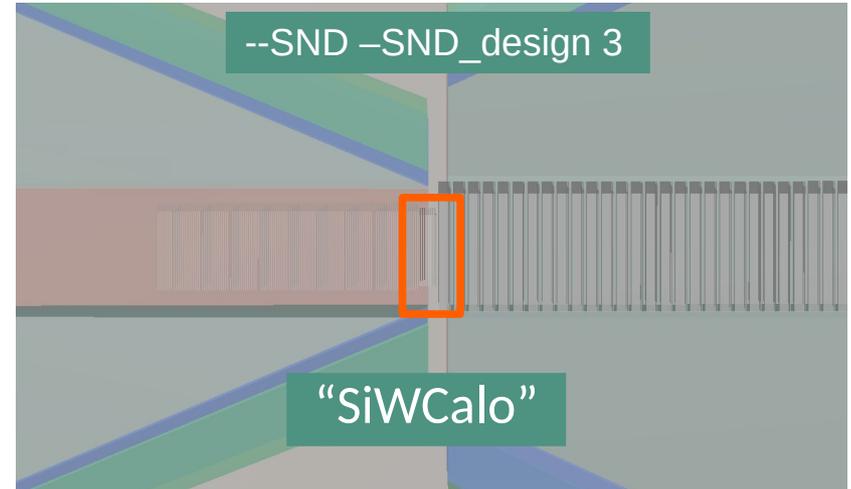
# Adding our calorimeter (2 models)

## ► In between SiW strip detector and MTC:

- ▷ 5 simplified layers (5.6mm W + **0.65mm Si**)
- ▷ Pixel size of **~5.5mm**
- ▷ 36x36 cm<sup>2</sup> layers (2x2 ASUs)

## ► After MTC:

- ▷ 12 simplified layers (50mm Fe + **0.65mm Si**)
- ▷ Pixel size of **~5.5mm**
- ▷ 54x54 cm<sup>2</sup> layers (3x3 ASUs)

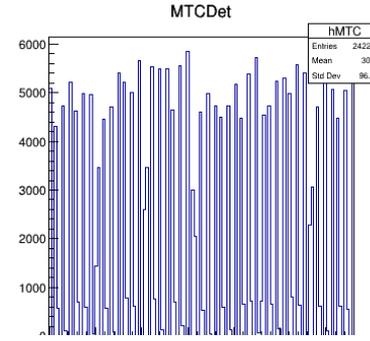
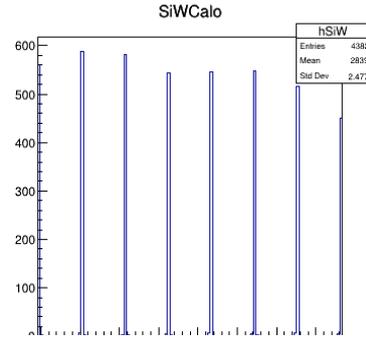
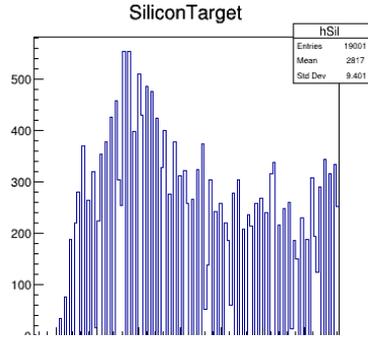


Technical detail: We need "gGeoManager → SetVisLevel(5);" for it to appear in TGeoViewer

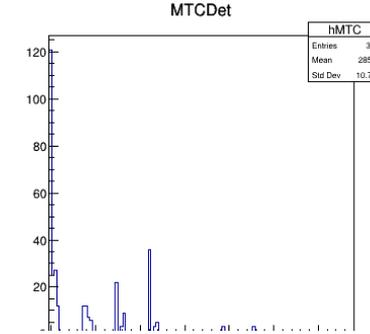
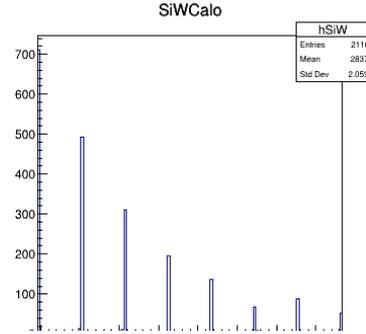
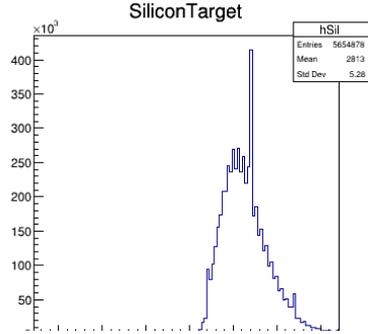
# Sanity checks (SiWCalo): PG 60 GeV $\mu$ , $\gamma$ & $\pi$ (N=300)



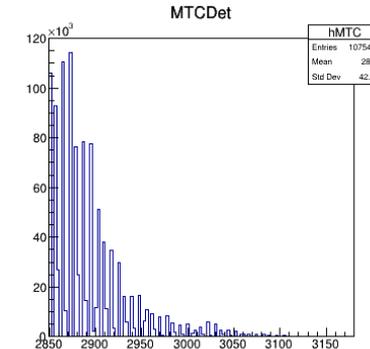
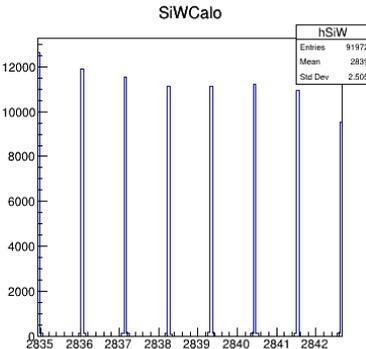
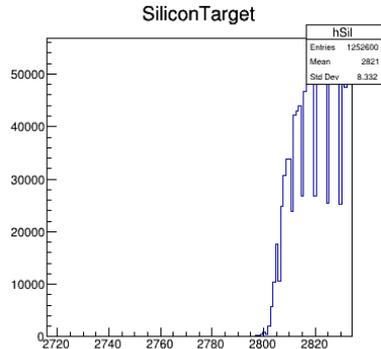
$\mu$



$\gamma$



$\pi$



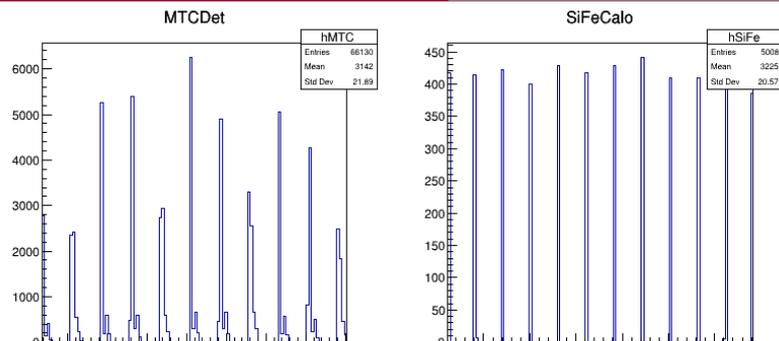
PG(z) = 2800

Caveat: These are just raw sim hits

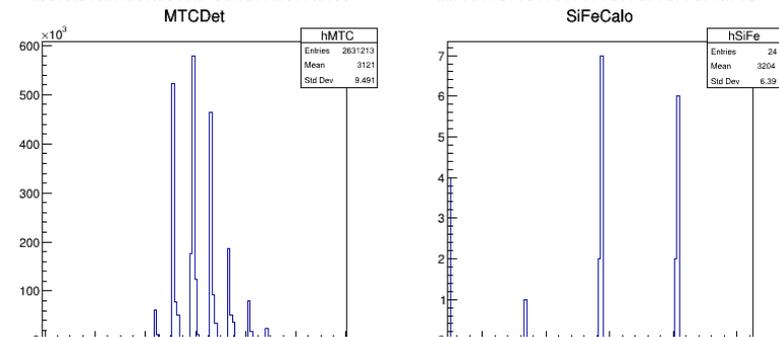
# Sanity checks (SiFeCalo): PG 60 GeV $\mu$ , $\gamma$ & $\pi$ (N=300)



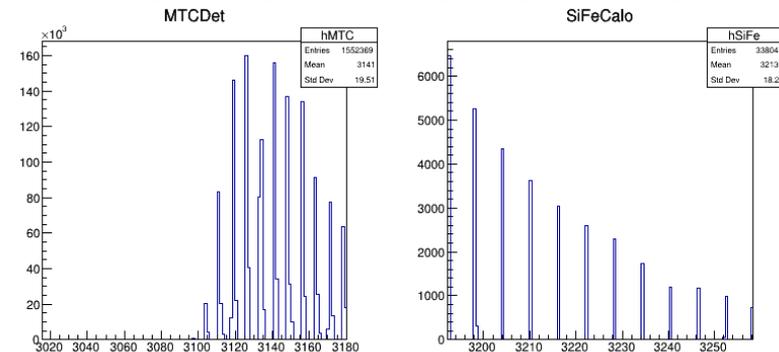
$\mu$



$\gamma$



$\pi$



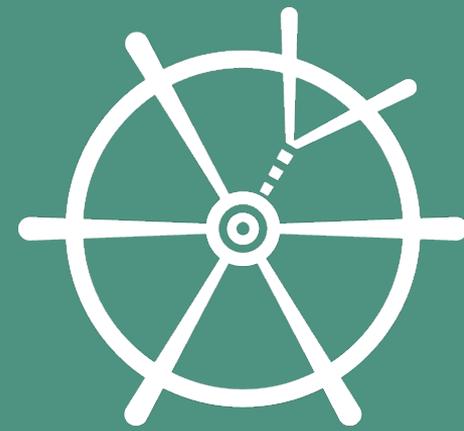
PG(z) = 3200

Caveat: These are just raw sim hits



- ▶ Implementing the SiWCalo and SiFeCalo models into the reconstruction chain
  - ▷ + a modified version substituting some SiTarget's layers for ours (TBD?)
- ▶ Adding minimal digitization in our sensors
  - ▷ GeV2MIP calibration factor and cutting out hits with  $E_{\text{hit}} < 0.5 \text{ MIP}$  (TBD?)
    - It seems like they don't do anything like this, maybe they do it in the analysis step (?)
    - What they call "digitization" is merging the Geant4 depositions into a simcalorimeterhit
- ▶ Have a look at neutrino events (TBD?)
- ▶ I tried to re-run what Eduard and me have but in the newer releases and everything breaks, a full reimplementation needs to be done. (worth it?)

Important: This software is meant to be dropped before summer.



SHiP

*Search for Hidden Particles*

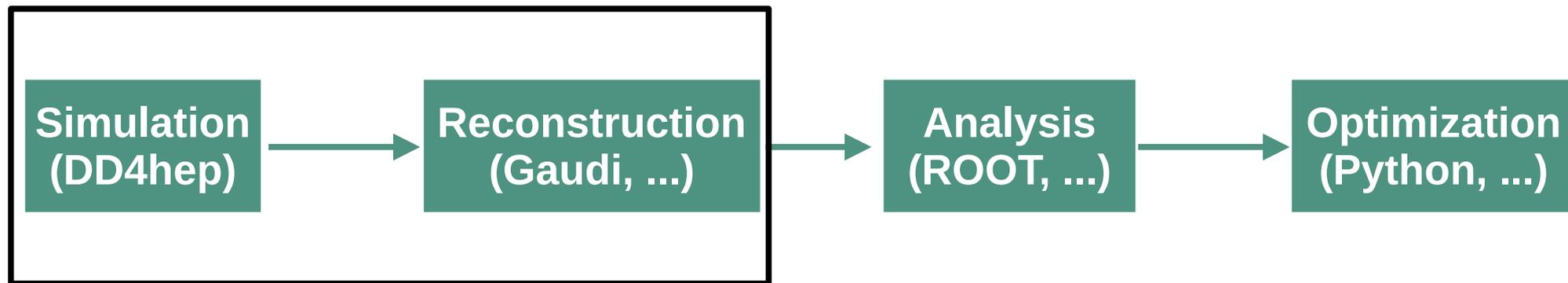
# Future options (and discussions)



# Calorimeter studies in key4hep



► Standard key4hep workflow:



Last Friday I presented this proof-of-concept in the SHiP sw meeting

[https://github.com/marherje/key4ship\\_PoC](https://github.com/marherje/key4ship_PoC)



# Proof-of-concept (key4ship)

```

  ▾ key4ship_PoC
    ▾ geometry
      🔥 elements.xml
      🔥 materials.xml
      🔥 SiPixelDetector.xml
      🔥 SND_compact.xml
    ▾ plugins
      🚀 SiPixelDetector.cpp
      🚀 SiTargetDetector.cpp
    ▾ reco
      🚀 BasicDigitizer.cpp
      🚀 GeV2MIPConversion.cpp
  ▾ run_scripts
    📄 ddsim_steering.py
    📄 digitize.py
    📄 .gitignore
    💵 clean.sh
    📄 CMakeLists.txt
    💵 init_key4hep.sh
    💵 init_key4ship.sh
    📄 LICENSE
    📄 README.md

```

## ► Geometry:

- ▷ General definitions
- ▷ Simulation steering files

## ► Plugins:

- ▷ The C++ detector building plugins

## ► Reco:

- ▷ The C++ Gaudi algorithms

## ► Run\_scripts:

- ▷ Python steering files for sim (ddsim) and reco (Gaudi)

## Languages



Simply source key4hep setup.sh, **with a fixed date**

Same + configuring runtime paths for C++ plugins and Python modules



## SND\_sim proof-of-concept of full key4hep sim+reco chain: dd4hep + ddsim + gaudi

SND PoC with STarget and SiPixel — DD4hep v01-35 / key4hep 2026-02-01

### Target stack (key4hep latest release in Feb 2026)

- DD4hep: v01-35
- ddsim: integrated in DD4hep v01-35
- Geant4: 11.x
- Gaudi: v4x series
- ROOT: 6.36+
- Platform: lxplus.cern.ch, AlmaLinux 9
- Source: `/cvmfs/sw.hsf.org/key4hep/setup.sh -r 2026-02-01`

### Build and run

This is as simple as it gets, if you have 40 subdetectors and 200 Gaudi algorithms the process is **the same**

### Build and run

```
# 1. Source key4hep
source /cvmfs/sw.hsf.org/key4hep/setup.sh

# 2. Build plugin
mkdir build && cd build
cmake -DCMAKE_PREFIX_PATH="$CMAKE_PREFIX_PATH" \
      -DCMAKE_INSTALL_PREFIX=./install \
      -DCMAKE_BUILD_TYPE=RelWithDebInfo ..
make install -j4
cd ..

# 3. Expose plugins to DD4hep & Gaudi algorithms to python
export LD_LIBRARY_PATH=$PWD/install/lib64:$PWD/install/lib:$LD_LIBRARY_PATH
export PYTHONPATH=$PWD/install/lib64:$PWD/install/lib:$PWD/install/python:$PYTHONPATH

# 4. Run ddsim
cd run_scripts
ddsim --steeringFile ddsim_steering.py

# 5. Run algorithm/s
k4run digitize.py
```

Loading the project, once compiled, is just this:

```
1  #!/usr/bin/env bash
2
3  # To use only if the project is already compile and you want to play with the steering file, for example.
4  # If you want to compile the project, please follow the instructions in the README.md file.
5
6  source /cvmfs/sw.hsf.org/key4hep/setup.sh -r 2026-02-01
7  export LD_LIBRARY_PATH=$PWD/install/lib64:$PWD/install/lib:$LD_LIBRARY_PATH
8  export PYTHONPATH=$PWD/install/lib64:$PWD/install/lib:$PWD/install/python:$PYTHONPATH
```

Now I'll go part-by-part over the next slides



- ▶ Easy to use
- ▶ Highly flexible (Very desirable for prototyping)
  - ▷ This flexibility comes from how you prepare the plugin/s
- ▶ Compact steering file:
  - ▷ Constants for world and detector plugins
  - ▷ Constants for visualization (colors, transparency, etc.)
  - ▷ Readouts for all the subdetectors
  - ▷ Magnetic fields
  - ▷ Subdetectors
    - Can be defined here directly
    - Can be imported (better option in the long run)



# My subdetectors' steerings

SiPixel defined on his own file and build layer-by-layer

```
<!-- Layers 2-4: W(1.40x) + W(4.20x) + CF + Cu + glue + Si(650um) + glue + Cu + PCB + chip + gap -->
<layer repeat="10" vis="EcalVis">
  <slice material="Air"          thickness="1.40*Ecal_WThickness"    vis="AirVis"/>
  <slice material="TungstenDens1910" thickness="4.20*Ecal_WThickness"    vis="TungstenVis"/>
  <slice material="CarbonFiber"  thickness="Ecal_CFTickness"      vis="CFVis"/>
  <slice material="Copper"       thickness="Ecal_KaptonThickness"  vis="CuVis"/>
  <slice material="Air"          thickness="Ecal_GlueThickness_kap" vis="AirVis"/>
  <slice material="Silicon"      thickness="Ecal_WaferThickness650" vis="SiVis" sensitive="yes"/>
  <slice material="Air"          thickness="Ecal_GlueThickness_pcb"  vis="AirVis"/>
  <slice material="Copper"       thickness="Ecal_KaptonThickness"  vis="CuVis"/>
  <slice material="PCB"          thickness="Ecal_PcbThickness"     vis="PCBVis"/>
  <slice material="Air"          thickness="Ecal_ChipThickness"     vis="AirVis"/>
  <slice material="Air"          thickness="Ecal_w_slab_gap650"     vis="AirVis"/>
</layer>
```

For SiTarget only general parameters are plugged in

```
<detectors>
  <include ref="SiPixelDetector.xml"/>

  <detector name="SiTarget" type="SiTargetDetector" id="5" readout="SiTargetHits">
    <parameters n_layers="SiTarget_n_layers"
                target_thickness="SiTarget_absorber_thick"
                target_spacing="SiTarget_layer_pitch"
                module_offset="SiTarget_module_offset"
                sensor_width="SiTarget_sensor_width"
                sensor_length="SiTarget_sensor_length"
                sensor_thickness="SiTarget_sensor_thickness"
                env_width="SiTarget_env_width"
                env_height="SiTarget_env_height"
                z_position="SiTarget_z_position"
                absorber_material="TungstenDens1910"
                sensor_material="Silicon"/>
  </detector>
</detectors>
```

This is in the compact file



- ▶ This IS the hard part, the only hard part!
  - ▷ Not harder than the current FairShip geometry at all
- ▶ You defined here how to construct your detector:
  - ▷ Input general parameters: N layers, slices of materials, type of detector
  - ▷ Sensitive part/s description/s: Type (pixels, strips), information for encoding (system, layer, slice, plane)
- ▶ I have 2 plugins
  - ▷ SiPixelDetector: Very flexible, recycled from our previous SiWECAL TB simulations
    - Built detector *layer-by-layer*, allowing customization per slice (different W, Si, glue, PCB, whatever). All layers with similar digitization (pixels)
  - ▷ SiTargetDetector: Inspire from the FairShip Tgeo implementation
    - Get general parameters and built similar N Layers. All layers with 2 strip planes (X-Y).



```
StatusCode execute(const EventContext&) const override {
    const auto* input = m_inputHandle->get();
    auto* output = m_outputHandle->createAndPut();

    for (const auto& hit : *input) {
        auto nh = output->create();
        nh.setCellID(hit.getCellID());
        nh.setEnergy(static_cast<float>(hit.getEnergy() * (1/m_MIPValue)));
        nh.setPosition(hit.getPosition());
    }

    info() << "GeV2MIPConversion: " << input->size() << " hits processed" << endmsg;
    return StatusCode::SUCCESS;
}

StatusCode finalize() override {
    m_inputHandle.reset();
    m_outputHandle.reset();
    return Gaudi::Algorithm::finalize();
}

private:
    // Python-configurable properties
    Gaudi::Property<std::string> m_inputName{
        this, "InputCollection", "SiTargetHits", "Input SimCalorimeterHit collection"};
    Gaudi::Property<std::string> m_outputName{
        this, "OutputCollection", "SiTargetHitsMIP", "Output SimCalorimeterHit collection"};
    Gaudi::Property<double> m_MIPValue{
        this, "MIPValue", 0.000009, "MIP value for conversion (in GeV)"}; // Default is 9 keV, for 0.3mm silicon

    // DataHandles constructed in initialize() from the properties above
    mutable std::unique_ptr<k4FWCore::DataHandle<edm4hep::SimCalorimeterHitCollection>> m_inputHandle;
    mutable std::unique_ptr<k4FWCore::DataHandle<edm4hep::SimCalorimeterHitCollection>> m_outputHandle;
```

► Every algorithm inherits from Gaudi

► You always fit the template

► Building your own functions is not more difficult than ROOT

► Everything is read/written from/into EDM4hep

► You always stick to the format

**It protect us from ourselves (& each other)**





```
import os
from DDSim.DD4hepSimulation import DD4hepSimulation
from g4units import mm, GeV, MeV

SIM = DD4hepSimulation()

SIM.runType          = "batch"
SIM.numberOfEvents  = 1000
SIM.skipNEvents     = 0

SIM.compactFile = "../geometry/SND_compact.xml"
SIM.outputFile  = "output.edm4hep.root"
print("COMPACT FILE =", SIM.compactFile)

SIM.enableGun      = True
SIM.gun.particle   = "mu-"
SIM.gun.energy     = 50 * GeV
SIM.gun.position   = (0, 0, -500 * mm)
SIM.gun.direction  = (0, 0, 1)

SIM.physicsList    = "QGSP_BERT"
```

- ▶ Very simple python script
- ▶ Easy for generalization/template
- ▶ This is a simple PG, example for PoC
  - ▷ In principle ddsim can also interface Genie with Geant4 into the dd4hep geometry



# run\_script: Gaudi steering file

- ▶ I made 2 Gaudi algorithms for the PoC:
  - ▷ GeV to MIP conversion using a constant value
  - ▷ Basic “digitization”, just a MIP cut
- ▶ Again, very simple python script & easy for generalization/template work
- ▶ The workflow is trivial to follow
  - ▷ As well as adding future algorithms in the chain

AppMgr: Controls what events to process, what algorithms to run, and in what order. What happens if you do [mip\_1, dig\_1, mip\_2, dig\_2]? And if you do [mip\_1, dig\_2, mip\_2, dig\_1]?

```
from k4FWCore import ApplicationMgr, IOSvc
from Configurables import GeV2MIPConversion, BasicDigitizer
# Initializer
iosvc = IOSvc()
iosvc.Input = "output.edm4hep.root"
iosvc.Output = "snd_digi.edm4hep.root"

# Loading the digitization algorithms
mip_1 = GeV2MIPConversion("GeV2MIP_SiTarget")
mip_1.InputCollection = "SiTargetHits"
mip_1.OutputCollection = "SiTargetHitsMIP"
mip_1.MIPValue = 0.00009

mip_2 = GeV2MIPConversion("GeV2MIP_SiPixel")
mip_2.InputCollection = "SiPixelHits"
mip_2.OutputCollection = "SiPixelHitsMIP"
mip_2.MIPValue = 0.0002

dig_1 = BasicDigitizer("BasicDigitizer_SiTarget")
dig_1.InputCollection = "SiTargetHitsMIP"
dig_1.OutputCollection = "SiTargetHitsDigi"
dig_1.Threshold = 0.5

dig_2 = BasicDigitizer("BasicDigitizer_SiPixel")
dig_2.InputCollection = "SiPixelHitsMIP"
dig_2.OutputCollection = "SiPixelHitsDigi"
dig_2.Threshold = 0.5

# Application manager
ApplicationMgr(
    EvtSel = "NONE",
    EvtMax = -1,
    TopAlg = [mip_1, mip_2, dig_1, dig_2],
    ExtSvc = [iosvc]
)
```

Jesus Pedro Marquez Hernandez, 1 hour ago • First con

# Testing geoDisplay and materialScan (in my VM)



► I showed some dd4hep in-house tools

Geometry viewer — Mozilla Firefox

GL drawing

Search

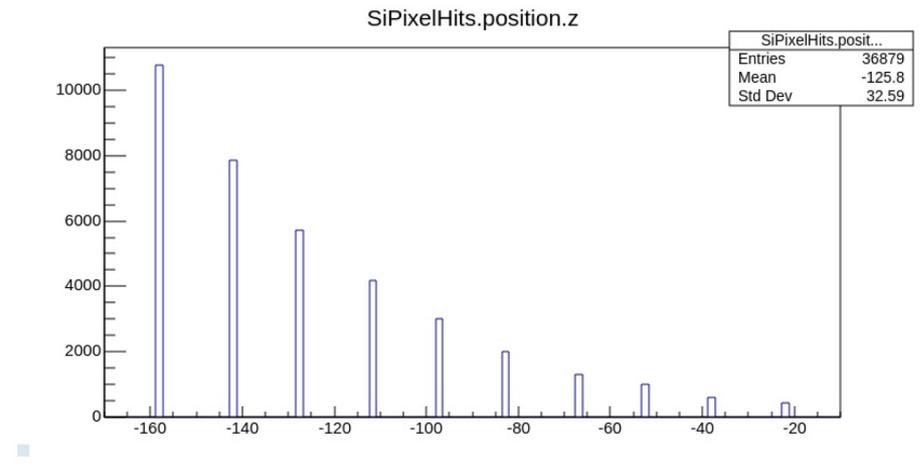
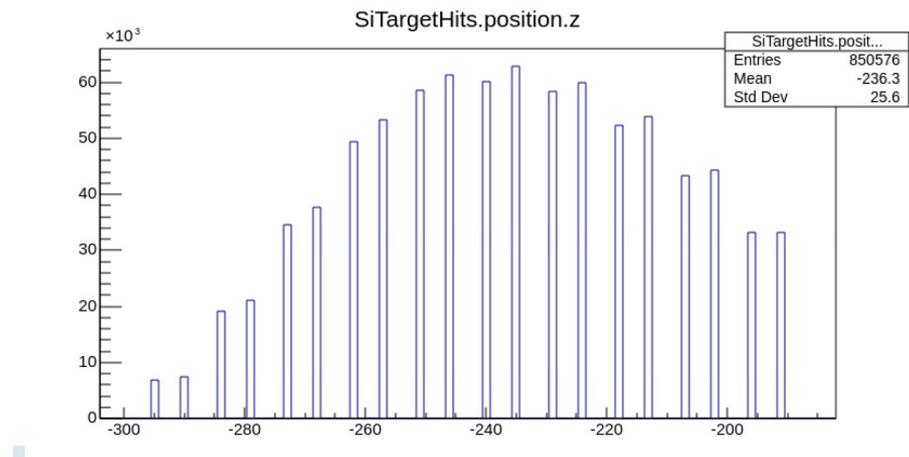
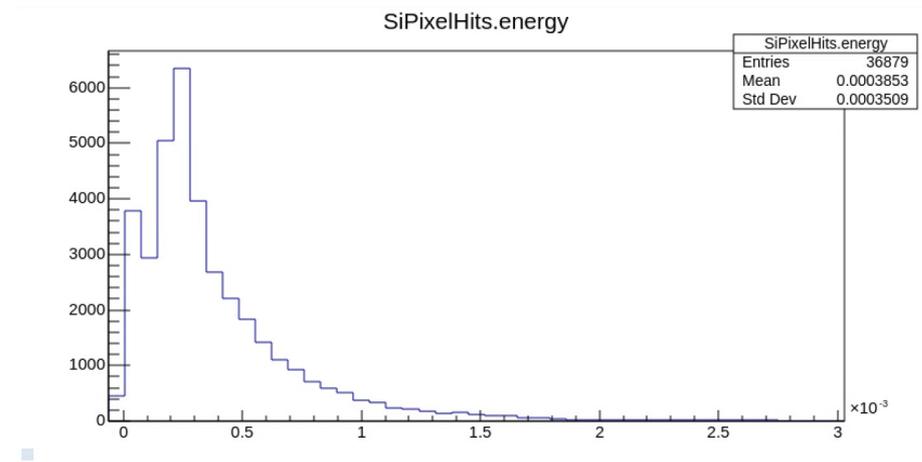
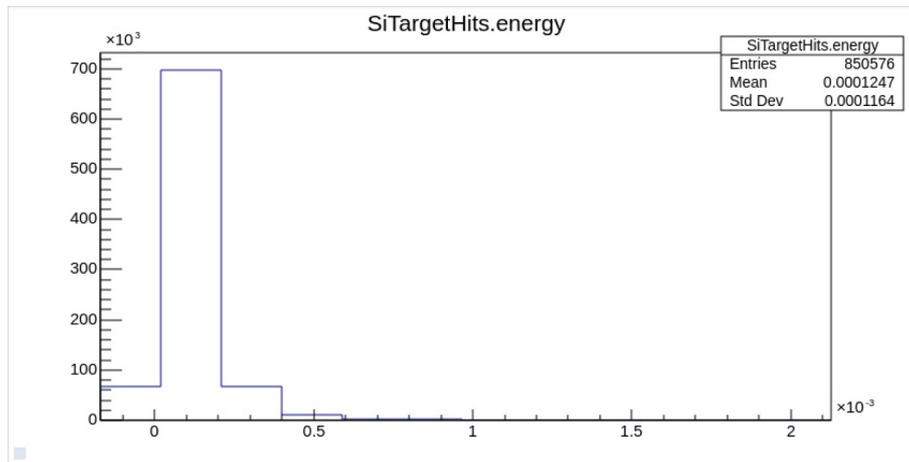
Description	Visibility	Co...	M...
world_volume_1	<input type="checkbox"/>	<input type="checkbox"/>	Air
SiPixel_envelope_C	<input type="checkbox"/>	<input type="checkbox"/>	Air
SiTarget_env_1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Air
SiTarget_W_0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_W_3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_W_6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_W_9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_W_12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_13	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_W_15	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_16	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_17	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_W_18	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_19	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_W_21	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tu...
SiTarget_si_22	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...
SiTarget_si_23	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sili...

I placed one SiPixel ASU instead of 2x2, will change it later

+ Material scan between: x\_0 = ( 0.00, 0.00,-100.00) [cm] and x\_1 = ( 0.00, 0.00, 100.00) [cm] :

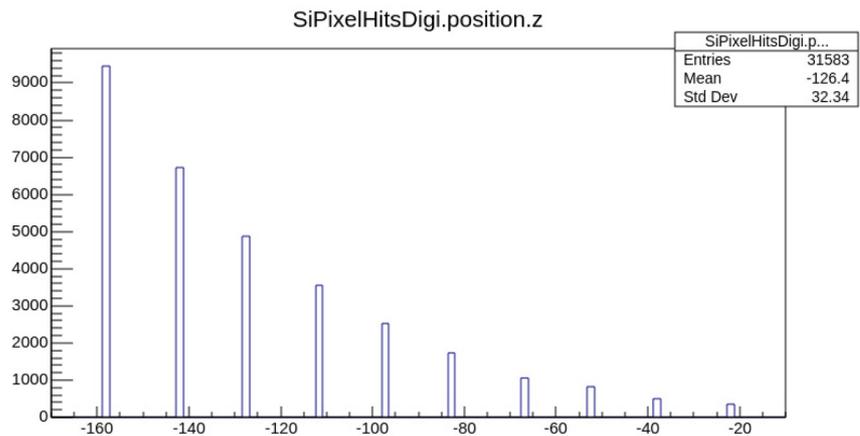
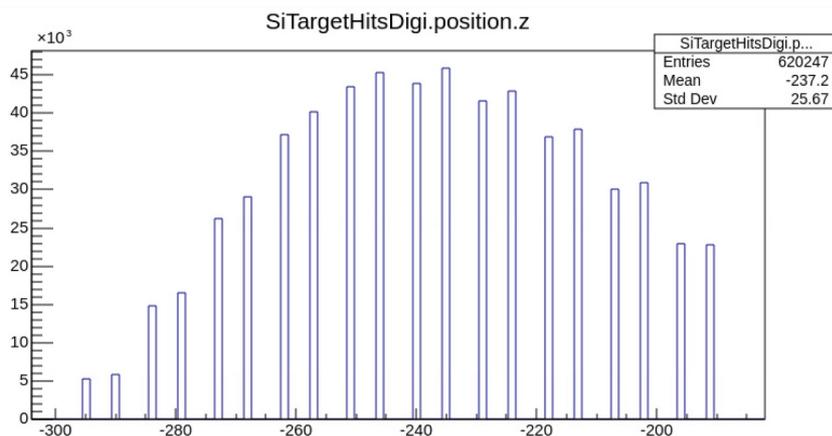
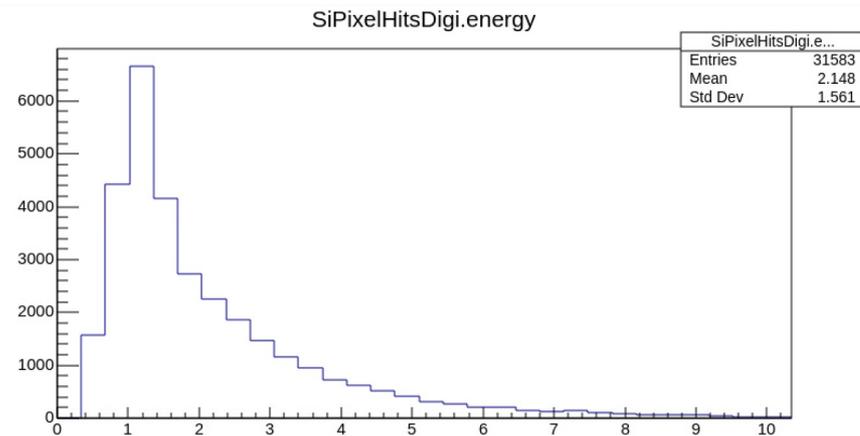
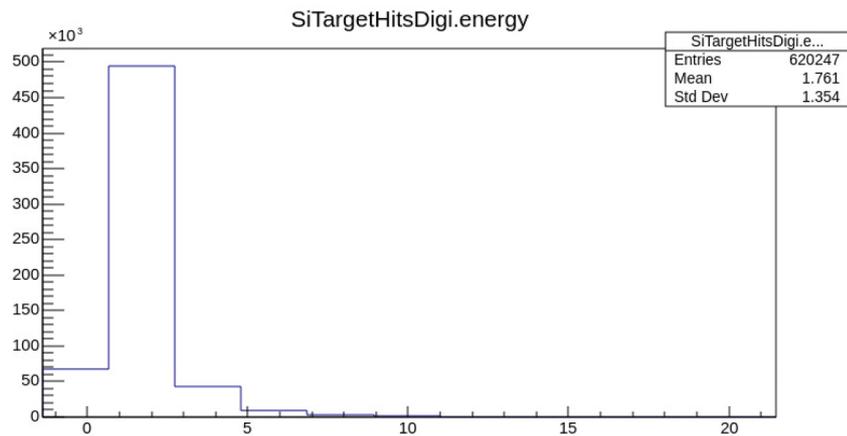
Num. Layer	Material Name	Atomic Number/Z	Mass/A [g/mole]	Density [g/cm3]	Radiation Length [cm]	Interaction Length [cm]	Thickness [cm]	Path Length [cm]	Integrated X0 [cm]	Integrated Lambda [cm]	Material Endpoint/Startpoint (cm, cm, cm)
(start)	Air	7	14.801	0.0012	30392.1308	71716.4382	0.000	0.00	0.000000	0.000000	( 0.00, 0.00,-100.00)
1	Air -> Air	7	14.801	0.0012	30392.1308	71716.4382	69.900	69.90	0.002300	0.000975	( 0.00, 0.00,-30.10)
2	Air -> TungstenDens1910	74	183.842	19.1000	0.3541	10.4515	0.100	70.00	0.002303	0.000976	( 0.00, 0.00,-30.00)
3	TungstenDens1910 -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.350	70.35	0.990758	0.034464	( 0.00, 0.00,-29.65)
4	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.100	70.45	0.990761	0.034465	( 0.00, 0.00,-29.55)
5	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	70.48	0.993964	0.035121	( 0.00, 0.00,-29.52)
6	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.550	71.03	0.993982	0.035129	( 0.00, 0.00,-28.87)
7	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	71.06	0.997185	0.035785	( 0.00, 0.00,-28.84)
8	Air -> TungstenDens1910	74	183.842	19.1000	0.3541	10.4515	0.040	71.10	0.997187	0.035785	( 0.00, 0.00,-28.80)
9	TungstenDens1910 -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.350	71.45	1.985641	0.069273	( 0.00, 0.00,-28.55)
10	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.100	71.55	1.985644	0.069274	( 0.00, 0.00,-28.45)
11	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	71.58	1.988847	0.069930	( 0.00, 0.00,-28.42)
12	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.550	72.13	1.988865	0.069938	( 0.00, 0.00,-27.87)
13	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	72.16	1.992068	0.070594	( 0.00, 0.00,-27.84)
14	Air -> TungstenDens1910	74	183.842	19.1000	0.3541	10.4515	0.040	72.20	1.992070	0.070594	( 0.00, 0.00,-27.80)
15	TungstenDens1910 -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.350	72.55	2.980524	0.104082	( 0.00, 0.00,-27.45)
16	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.100	72.65	2.980528	0.104084	( 0.00, 0.00,-27.35)
17	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	72.68	2.983731	0.104739	( 0.00, 0.00,-27.32)
18	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.550	73.23	2.983749	0.104747	( 0.00, 0.00,-26.77)
19	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	73.26	2.986952	0.105403	( 0.00, 0.00,-26.74)
20	Air -> TungstenDens1910	74	183.842	19.1000	0.3541	10.4515	0.040	73.30	2.986953	0.105403	( 0.00, 0.00,-26.70)
21	TungstenDens1910 -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.350	73.65	3.975408	0.138891	( 0.00, 0.00,-26.35)
22	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.100	73.75	3.975411	0.138893	( 0.00, 0.00,-26.25)
51	TungstenDens1910 -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.350	79.15	8.949824	0.312936	( 0.00, 0.00,-20.85)
52	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.100	79.25	8.949827	0.312938	( 0.00, 0.00,-20.75)
53	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	79.28	8.953030	0.313593	( 0.00, 0.00,-20.72)
54	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.590	79.83	8.953048	0.313601	( 0.00, 0.00,-20.17)
55	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	79.86	8.956251	0.314257	( 0.00, 0.00,-20.14)
56	Air -> TungstenDens1910	74	183.842	19.1000	0.3541	10.4515	0.040	79.90	8.956253	0.314257	( 0.00, 0.00,-20.10)
57	TungstenDens1910 -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.350	80.25	9.944707	0.347745	( 0.00, 0.00,-19.75)
58	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.100	80.35	9.944711	0.347747	( 0.00, 0.00,-19.65)
59	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	80.38	9.947914	0.348402	( 0.00, 0.00,-19.62)
60	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.550	80.93	9.947932	0.348410	( 0.00, 0.00,-19.07)
61	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.030	80.96	9.951135	0.349066	( 0.00, 0.00,-19.04)
62	Air -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.100	81.06	9.951138	0.349067	( 0.00, 0.00,-18.94)
63	Air -> Air	7	14.801	0.0012	30392.1308	71716.4382	2.440	83.50	9.951218	0.349101	( 0.00, 0.00,-16.50)
64	Air -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.000	83.50	9.951218	0.349101	( 0.00, 0.00,-16.50)
65	Air -> TungstenDens1910	74	183.842	19.1000	0.3541	10.4515	0.140	83.64	9.951223	0.349103	( 0.00, 0.00,-16.36)
66	TungstenDens1910 -> CarbonFiber	6	11.956	1.4667	28.6083	51.8139	0.420	84.06	11.137368	0.389289	( 0.00, 0.00,-15.94)
67	CarbonFiber -> Copper	29	63.546	8.9600	1.4356	15.6778	0.150	84.21	11.142612	0.392184	( 0.00, 0.00,-15.79)
68	Copper -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.010	84.22	11.149577	0.392821	( 0.00, 0.00,-15.78)
69	Air -> Silicon	14	28.085	2.3300	9.3661	45.7531	0.010	84.23	11.149578	0.392822	( 0.00, 0.00,-15.77)
70	Silicon -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.065	84.30	11.156518	0.394242	( 0.00, 0.00,-15.70)
71	Air -> Copper	29	63.546	8.9600	1.4356	15.6778	0.010	84.31	11.156518	0.394242	( 0.00, 0.00,-15.69)
72	Copper -> PCB	10	20.338	1.7000	17.5049	49.2238	0.010	84.32	11.163484	0.394880	( 0.00, 0.00,-15.68)
73	PCB -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.170	84.49	11.173195	0.398334	( 0.00, 0.00,-15.51)
74	Air -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.120	84.61	11.173199	0.398336	( 0.00, 0.00,-15.39)
75	Air -> Air	7	14.801	0.0012	30392.1308	71716.4382	0.395	85.00	11.173212	0.398341	( 0.00, 0.00,-15.00)
76	Air -> TungstenDens1910	74	183.842	19.1000	0.3541	10.4515	0.140	85.14	11.173217	0.398343	( 0.00, 0.00,-14.86)

# Examples of PoC (5 GeV e<sup>-</sup> shower - Sim)





# Examples of PoC (5 GeV e<sup>-</sup> shower - Digi)





# Other's people contributions

- ▶ Matei has done some geometries+G4 simulation in GeoModel. Feedback:
  - ▷ Geometry building is very easy
  - ▷ G4 binding is fine
  - ▷ Simulation output and edm4 is not well defined yet
- ▶ Eduard did MTC in GeoModel
  - ▷ No sim or anything else
- ▶ Gilhermo said he prefers GeoModel for geometry because it knows it
  - ▷ But not a strong opinion or work presented

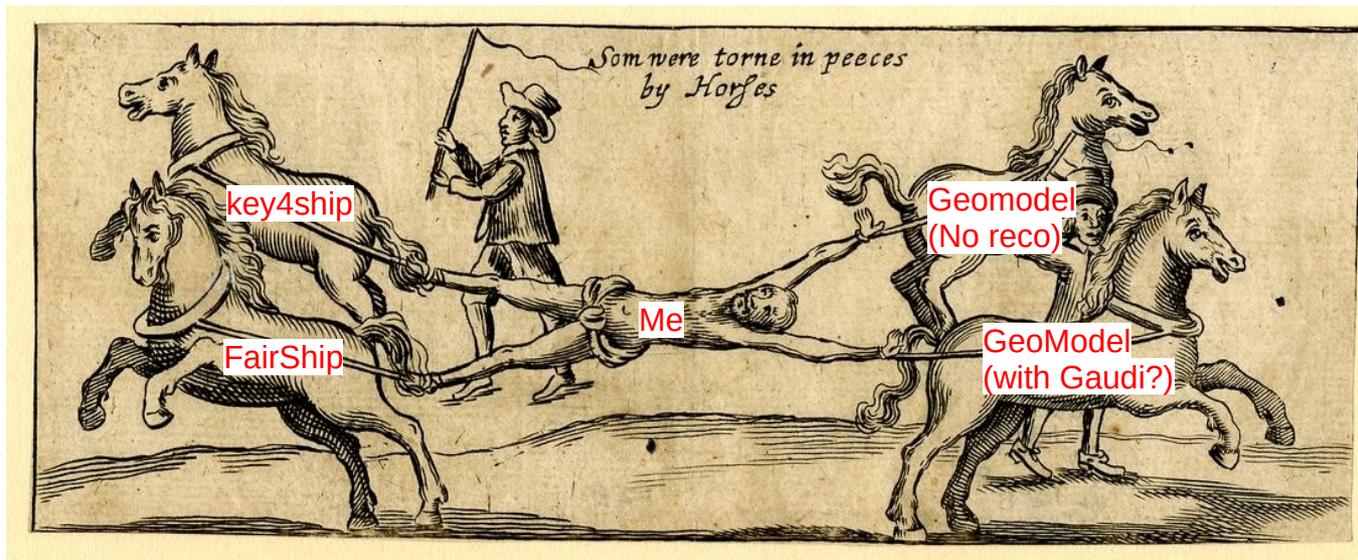
Basically everyone wants GeoModel because it's easy for building the geometry but there's no define pipeline after that

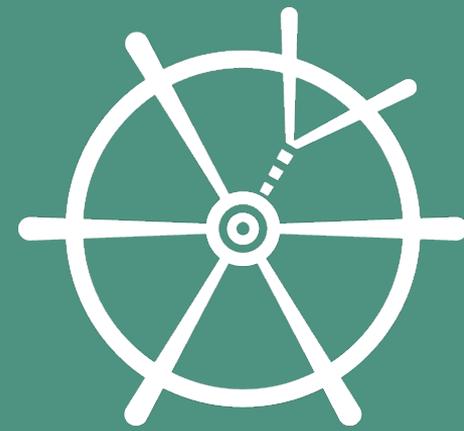


# Current status

- ▶ Can be reduced to a single word: uncertainty
- ▶ I don't know what to do
- ▶ I don't want to waste resources
- ▶ Right now there are too many options, mostly in opposite directions

Collaboration meeting is in 2 weeks, what should I present there?





SHiP

*Search for Hidden Particles*

Back-up

