Geometry Management System in OSCAR

He Li
on behalf of STCF Software Group

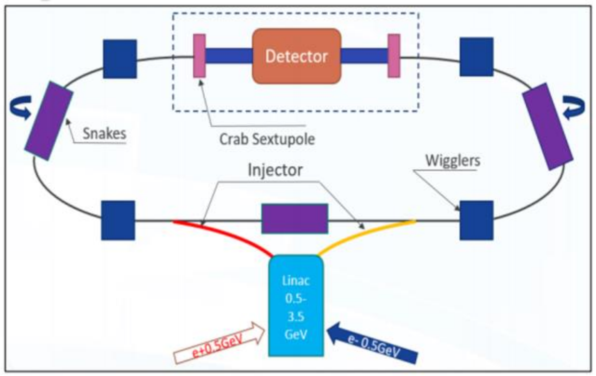
Qingdao Shandong
August,18,2021

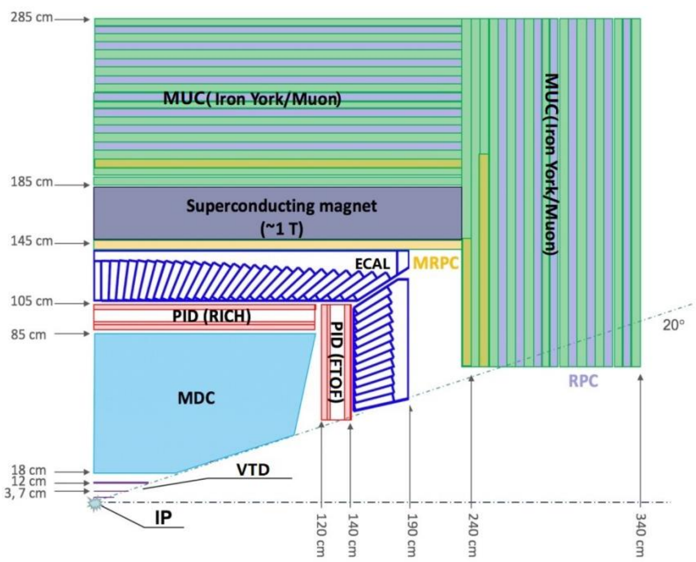
Outline

- STCF detectors
- Geometry management requirements
- Design of Geometry Management System(GMS)
- Test and performance
- Summary and outlook

STCF Detectors

Super Tau-Charm Facility





High Intensity Electron Positron Accelerator

- E_{cm}=2-7 GeV
- Lum = $(0.5\sim1.0)\times10^{35}cm^{-2}s^{-1}$
- Research for physics goals tau-charm energy region.

New challenge for the detectors:

- Large geometrical acceptance of 4π
- Hige detection efficiency for low p_t tracks
- High momentum resolution and PID capabilities for high p_t tracks
- High efficiency and resolution for photons
- Stability and reliability

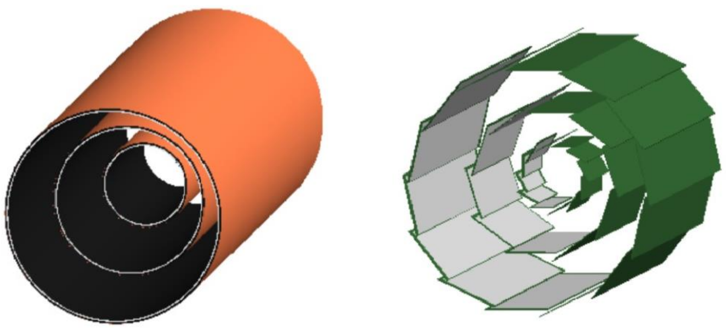
General purpose detector:

- Vertex Detector（VTD）
- Main Drift Chamber（MDC）
- Particle IDentification(RICH&DTOF)
- EM Calorimeter（EMC）
- Muon Detector（MUD）

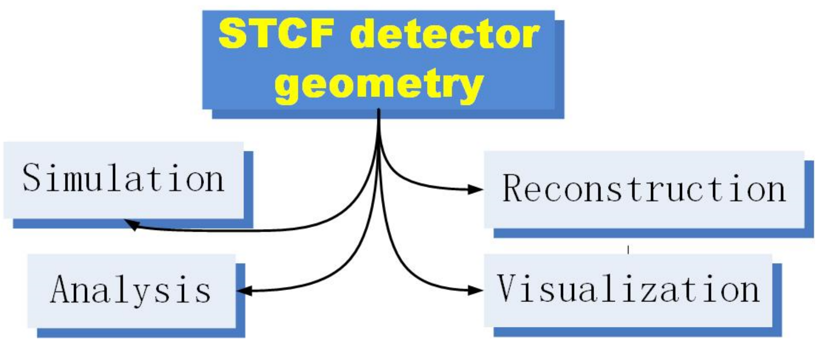
Geometry Management requirements

Application scenario in detector simulation:

- Complicated geometry descriptions
- Designers from different institutions
- Ambiguous detector designs and versions
- Detector geometry is needed by simulation, reconstruction, analysis, visualization…



(a) μRWell design option. (b) ISD design option.



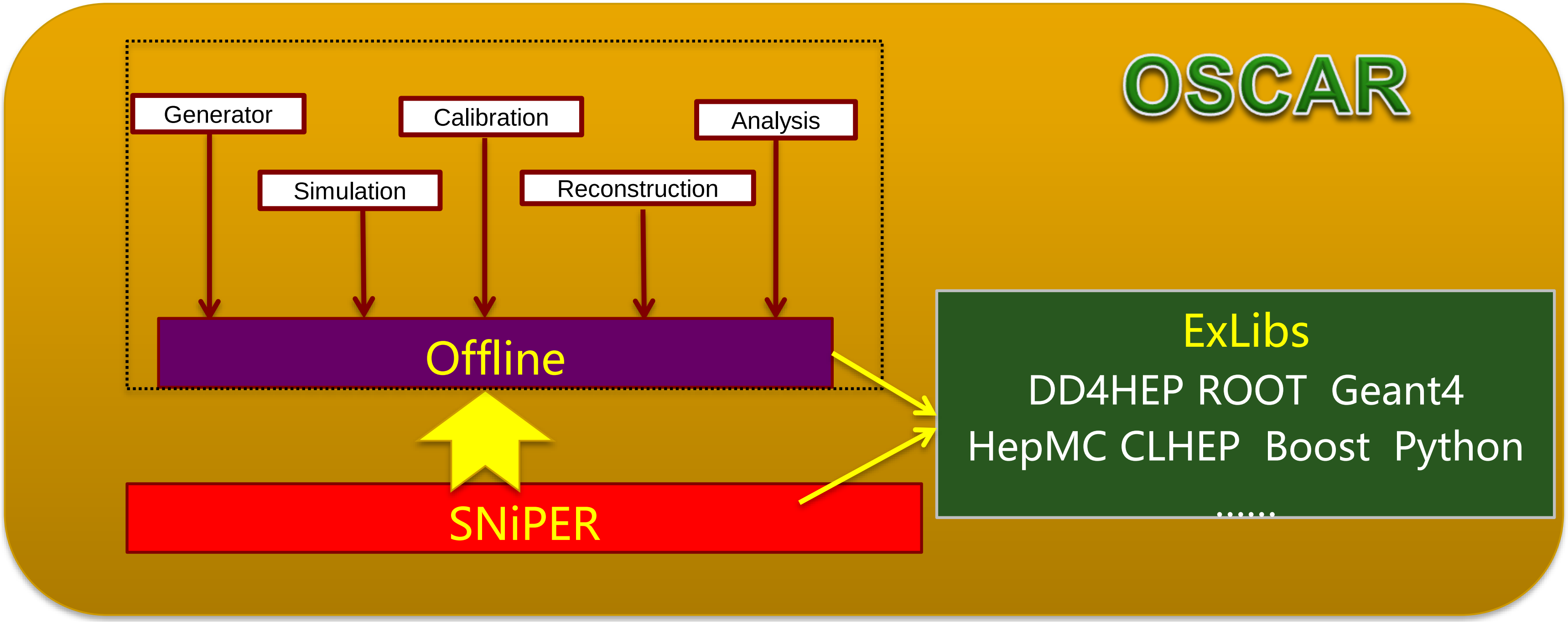
Geometry management requirements:

- Provide a consistent geometry description for offline applications
- Provide a unified and user-friendly software environment for detector designers and scientists
- Support flexible geometry combinations of sub-detectors and easy switching among different designs

Design of Geometry Management System(GMS)

Software environment

OSCAR: Offline Software of Super Tau-Charm Facility

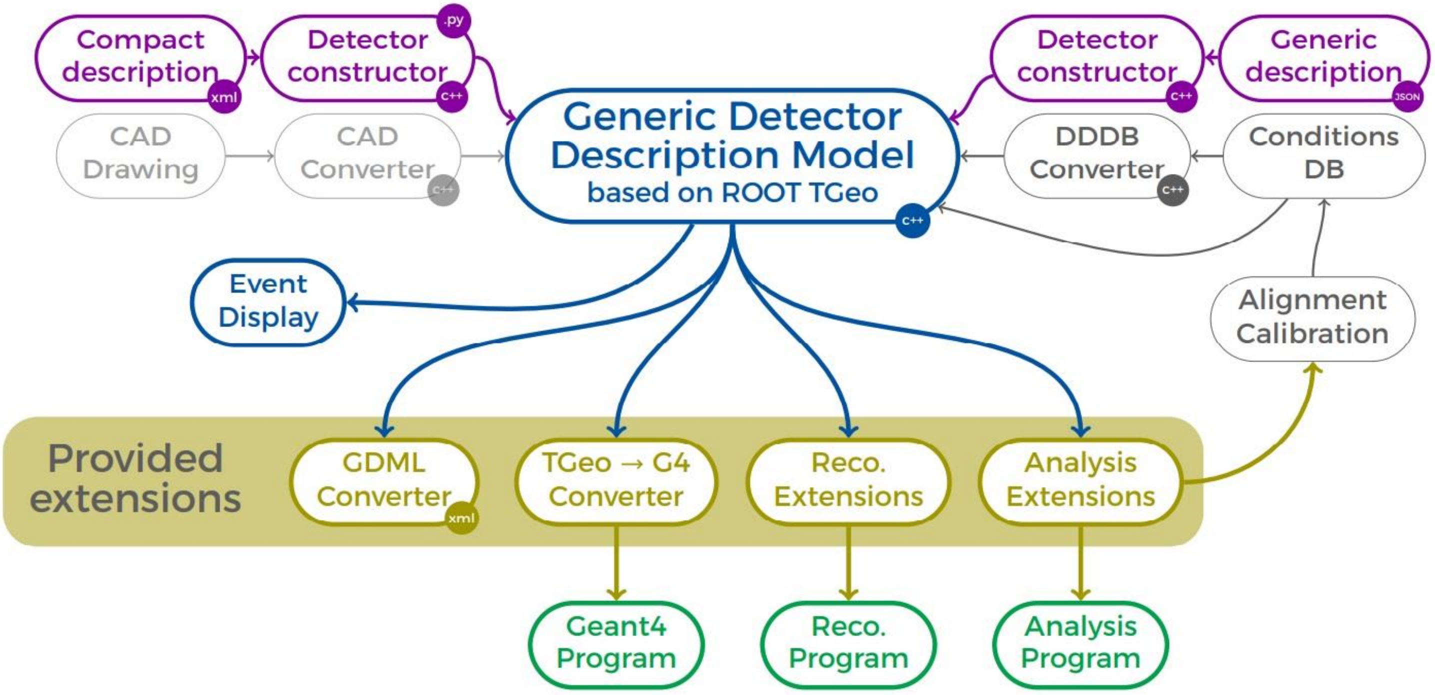


- **ExLibs:** **include frequently used third-party software and tools**
- **SNiPER:** **the framework to provide core functionalities and common services.**
- **Offline:** **all software specific to STCF**

Design of Geometry Management System(GMS)

Detector description toolkit

DD4hep: a general Detector Description tool for high energy physics, from LHCb

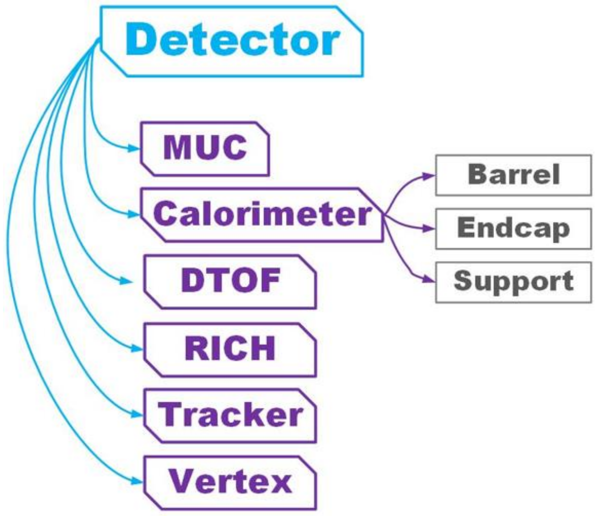


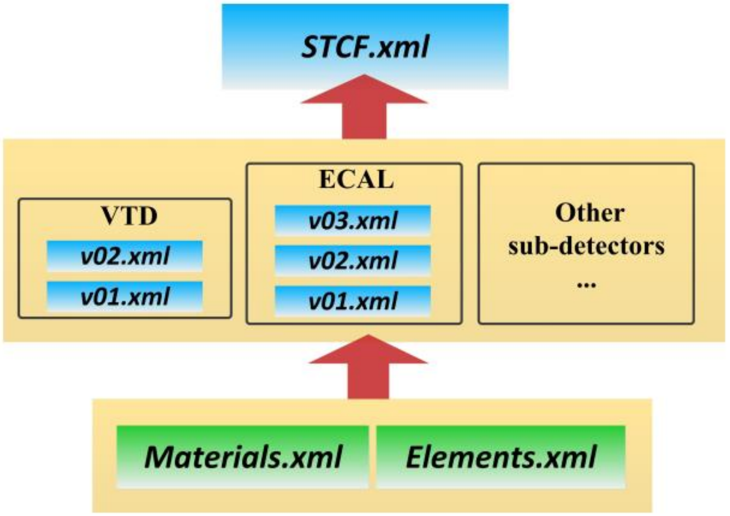
- **The geometries of different sub-detectors can be independently constructed and easily assembled into a full detector.**
- **Supports geometry conversion to Geant4 geometry objects.**
- **Provides a sophisticated 3D visualization plug-in to conveniently display and check detector designs.**
- **Offers full technical support and maintenance from the developers.**

Design of Geometry Management System(GMS)

Detector parameters repository

Tree-like hierarchy structure





Detector components description:

- Elements and materials
- Volume parameters
- Display, readout setup
- Fields…

Design of the prarameters repository

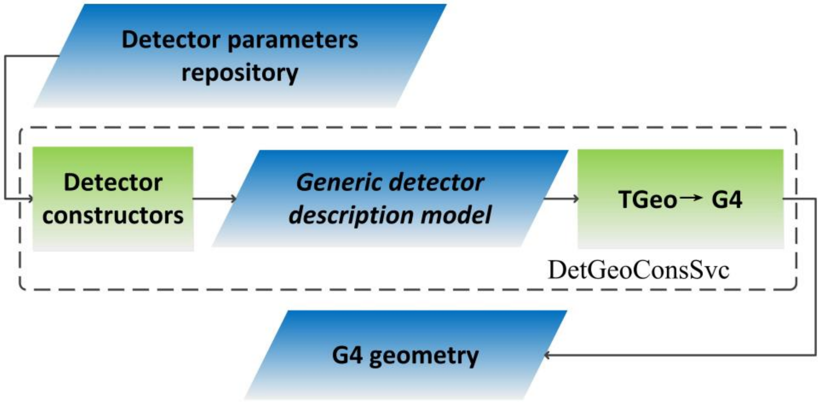
- Parameters are stored in XML files, human readable.
- Elements and materials are shared.
- Support multiple designs and flexible configuration of the geometry.
- Gitlab for version control based on apache

Design of Geometry Management System(GMS)

Geometry model construction

Detector constructor palette:

- **Specialized C++ codes**
- **Declare the detector name**
- **Parse detector parameters**
- **Placement of the volumes**




```
static Ref_t create_detector(Detector& theDetector,
                             xml_h e, SensitiveDetector sens) {

  xml_det_t x_det = e;
  Layering layering(x_det);
  xml_comp_t staves = x_det.staves();
  xml_dim_t dim = x_det.dimensions();
  DetElement sdet(det_name, x_det.id());
  Volume motherVol = theDetector.pickMotherVolume(sdet);

  PolyhedraRegular polyhedra(numSides, rmin, rmax, detZ);
  Volume envelopeVol(det_name, polyhedra, air);

  for (xml_coll_t c(x_det, _U(layer)); c; ++c) {
    xml_comp_t x_layer = c;
    int n_repeat = x_layer.repeat();
    const Layer* lay = layering.layer(layer_num - 1);
    for (int j = 0; j < n_repeat; j++) {
      string layer_name = _toString(layer_num, "layer%d");
      double layer_thickness = lay->thickness();
      DetElement layer(stave, layer_name, layer_num);
        ...}
  }
  DECLARE_DETELEMENT(GenericCalBarrel_o1_v01, create_detector)
```

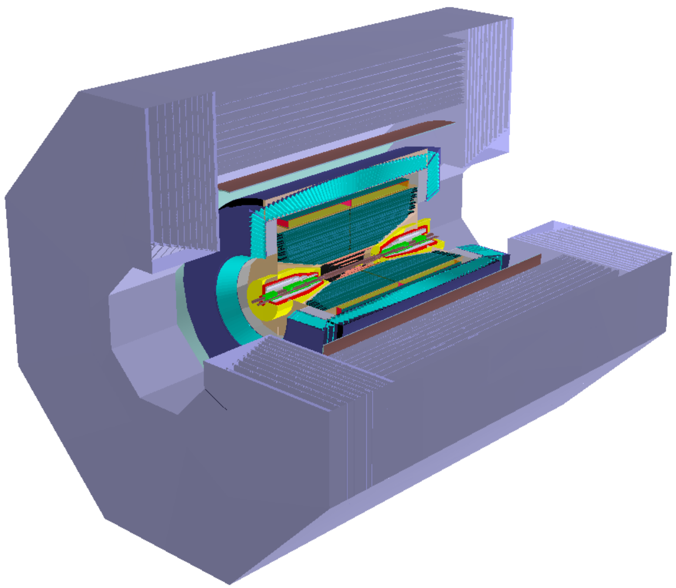
Geometry construction

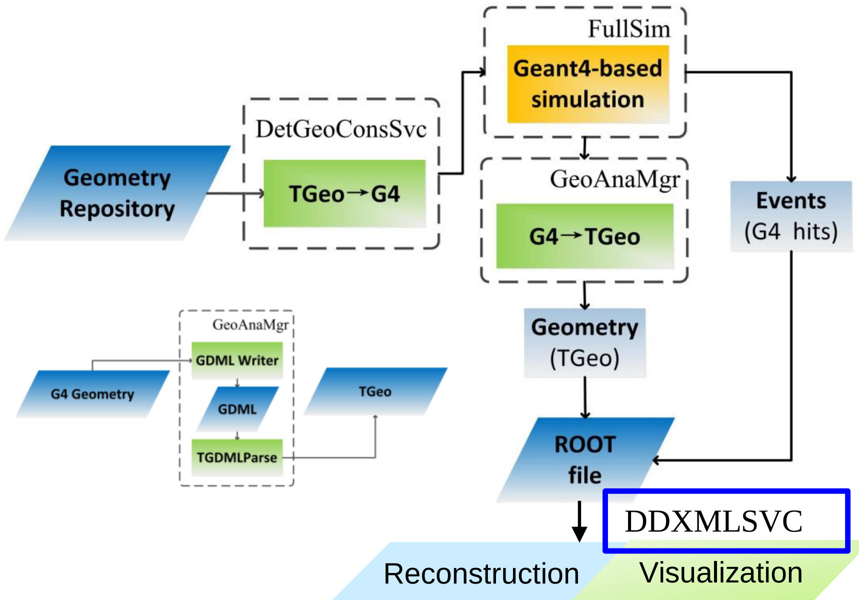
encapsulated in SNIPER-Type modules

- **detector constructors read and parse the geometry parameters.**
- **construct the generic detector description model.**
- **the generic detector description model is converted to G4 geometry.**

Design of Geometry Management System(GMS)

Workflow of GMS





STCF detector Geometry

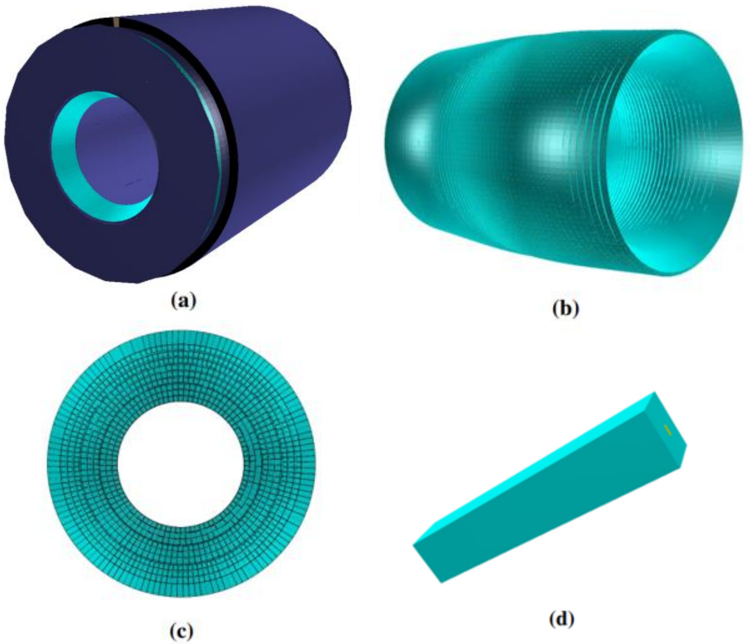
- **Easy switch and assembly of different sub-detectors**
- **Single sub-detector scenario**
- **Full detector scenario**

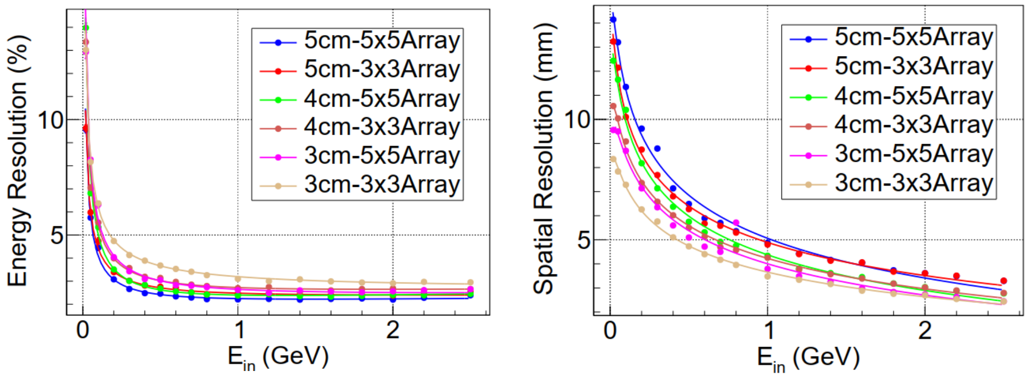
Workflow of GMS in simulation

- **Detector model was initialized by DetGeoConsSvc**
- **FullSim: SNiPER-Type algorithms based on Geant4 simulation**
- **After simulation, detector geometry is saved into TGeo in ROOT for further usage of reconstruction, visualization**

Test and performance of GMS

Workflow of GMS





Energy and spatial resolutions versus incident energy under different crystal sizes

Basic performace

With the GMS and FullSim:

- **Evaluate and optimize the detector design**
- **Different CsI crystal unit sizes are tested**

Table 1. Time consumption of the detector geometry construction.

Steps	Geometry construction	Geometry conversion	Total
Time consumption	1.12 s	0.16 s	1.28 s

Table 2. Comparison of XML and ROOT file sizes.

File	XML	ROOT
Size	100 KB	4 MB

Summary

- GMS was developed in OSCAR to provide a consistent geometry description for different offline applications
- A unified and user-friendly software environment for detector designers with several SNiPER-Type packages.
- Detector simulations with GMS are performed to help optimization of the detector.
- DDXML service is developed to extract geometric information for Reconstruction and Visualization.

Thank a lot !