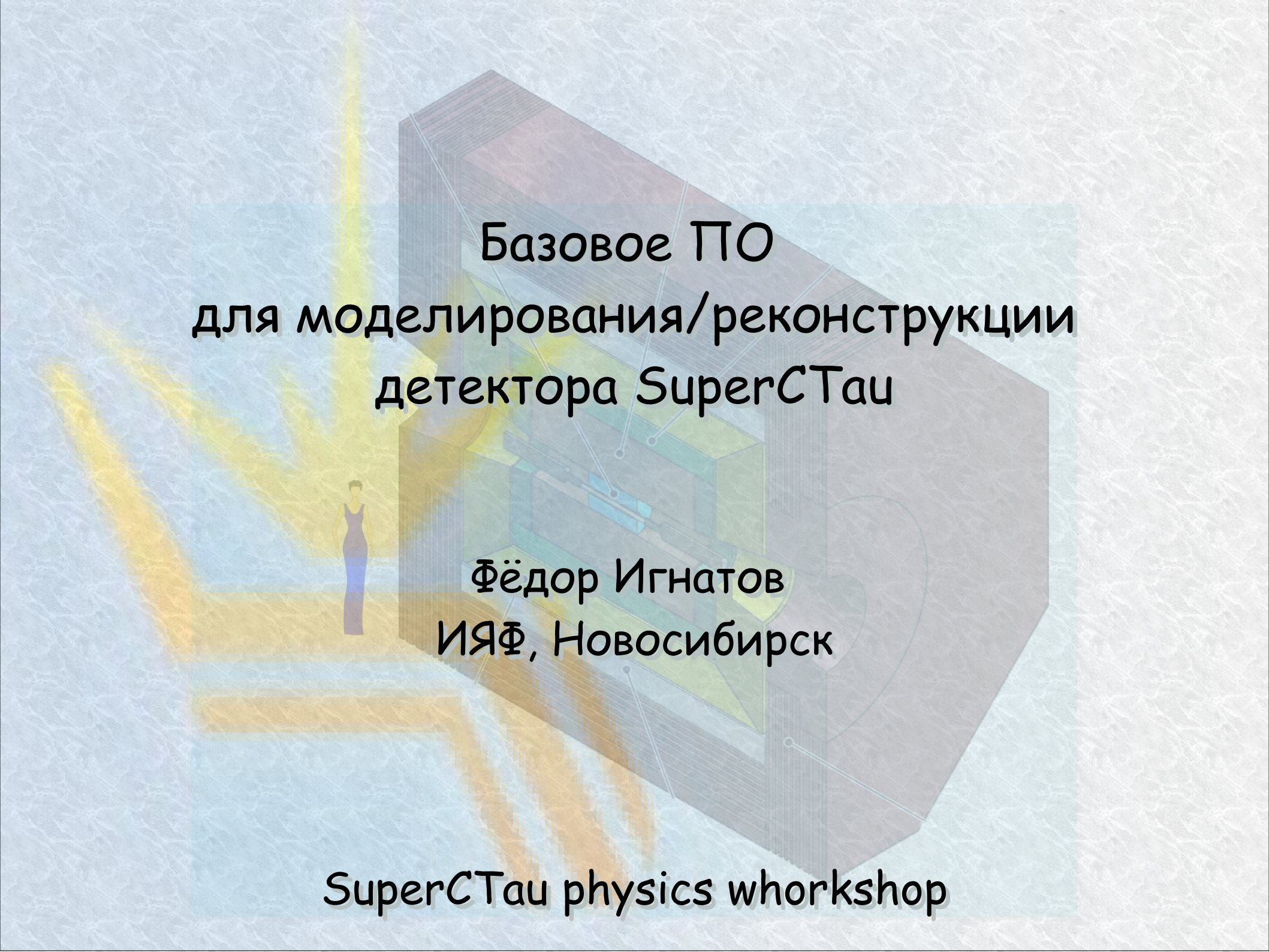


A 3D cutaway diagram of the SuperCTau detector, showing its complex internal structure with various colored layers and components. A small figure of a person in a blue dress stands to the left of the detector for scale. The background is a light blue and yellow abstract pattern.

Базовое ПО для моделирования/реконструкции детектора SuperCTau

Фёдор Игнатов
ИЯФ, Новосибирск

SuperCTau physics whorkshop

Было проведено несколько совещаний в рамках рабочей группы по моделированию

- × Наш ресурс по людям ограничен,
- × единственная возможность использовать пакеты которые:
используются и развиваются для других больших проектов
являются де-факто стандартом в HEP

в качестве базового софта для моделирования детектора/реконструкции SuperCTau
было выбрано:

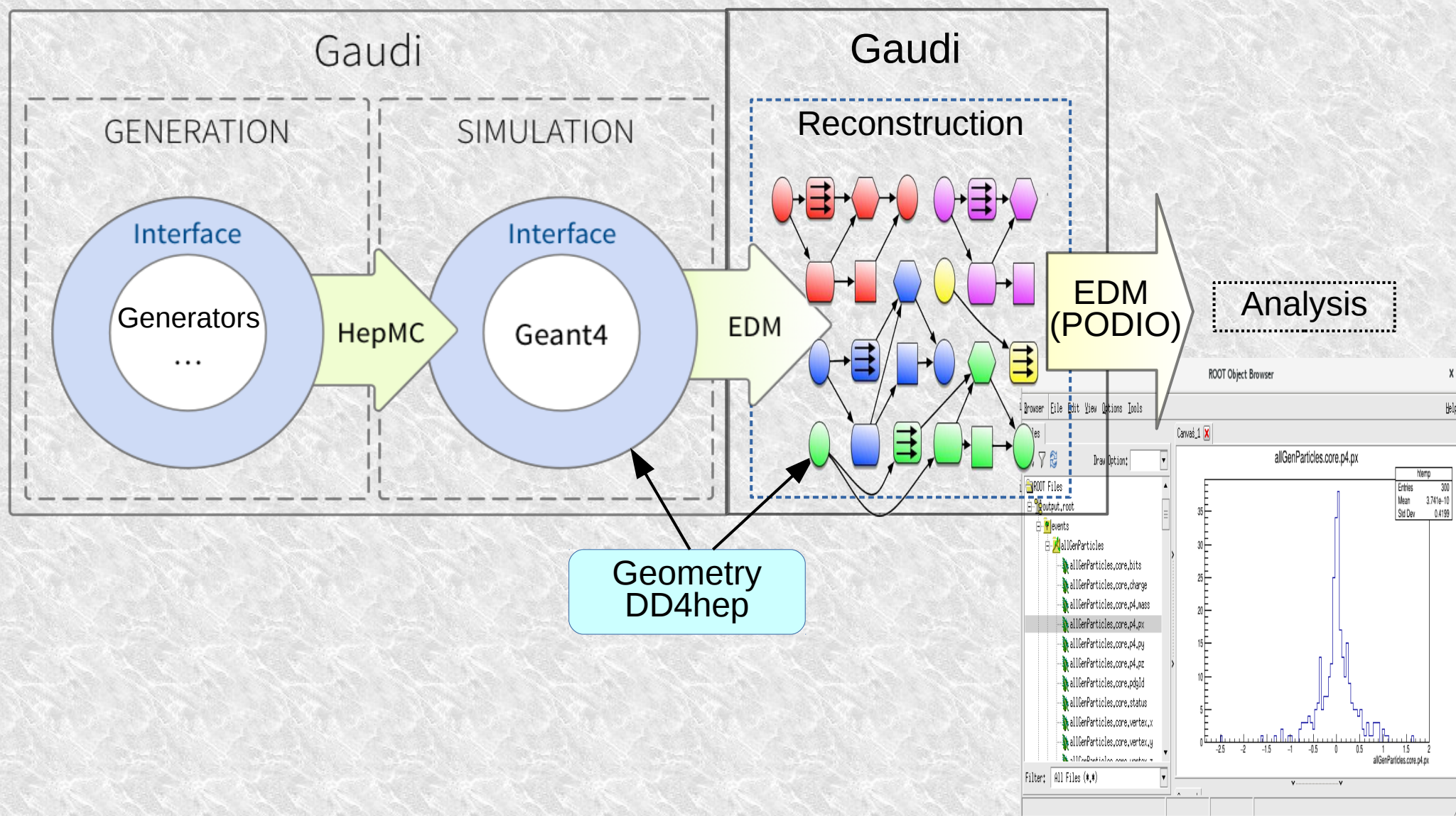
ROOT, Geant4 - без альтернатив

- | | |
|------------------------|---|
| Gaudi | - фреймворк обработки событий |
| HePMC3 | - интерфейс взаимодействия генераторов |
| PODIO | - модель данных (EDM) |
| DD4hep | - описание геометрии для целей
моделирования/реконструкции |
| cmake | - система сборки софта |
| git | - репозиторий на основе GitLab (http://git.inp.nsk.su) |
| wiki , e-mail рассылка | (https://star.inp.nsk.su/mailman/listinfo/ctd-software) |

Всё это модель ПО для FCC
FCCSW software в полном
виде взяты за основу
<http://fccsw.web.cern.ch/>

<https://indico.cern.ch/event/556692/contributions/2484280/>





Пакет PODIO - Plain Old Data I/O:

дата-центричная модель данных

+FCC-EDM(набор готовых классов)

без переизбытка с++-овской абстракции

Event Data Model (FCCSW decision)

After reviewing existing solutions:

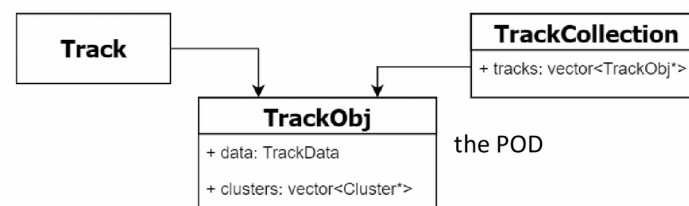
- LHC experiments: Complicated and not extractable
- Linear Collider I/O: Starting point, hard to adapt

Decided to invest: Plain Old Data I/O (PODIO)

- Focus on re-usability and flexibility
- Code fully generated from text files
- Simply describe your data, PODIO does the rest
- Easily adapt data model to changing requirements
- Python & C++ supported on the same footing

```
fcc::Track:
Members:
- float chi2
- unsigned ndf
OneToManyRelations:
- fcc::TrackCluster clusters
- fcc::TrackState states
```

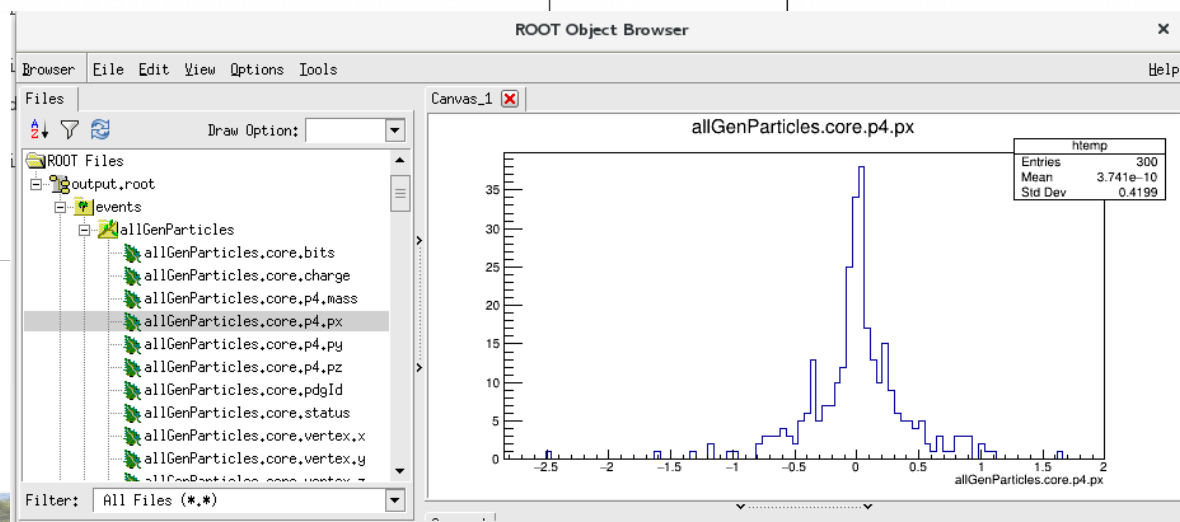
user interface



Основной потребитель FCC,
рассматривают LHCb, LCIO

Простой доступ в ROOT-файле:

<https://indico.cern.ch/event/505613/contributions/2228322>



В качестве фреймворка для обработки событий выбран пакет **Gaudi**

Gaudi in a nutshell



- Gaudi is an event based experiment data processing framework
 - Designed to marshall physics code in an organised way to manage experiment workflows
- Originally developed by LHCb ~2000
 - Later adopted by ATLAS
 - Other users: Daya Bay, LZ, FCC
- Essential concepts are
 - Separate data and algorithms
 - Different persistent and transient views of data
 - User code encapsulated in Algorithms and Tools
 - Well defined interfaces
- Code in CERN GitLab
 - <https://gitlab.cern.ch/gaudi/Gaudi>

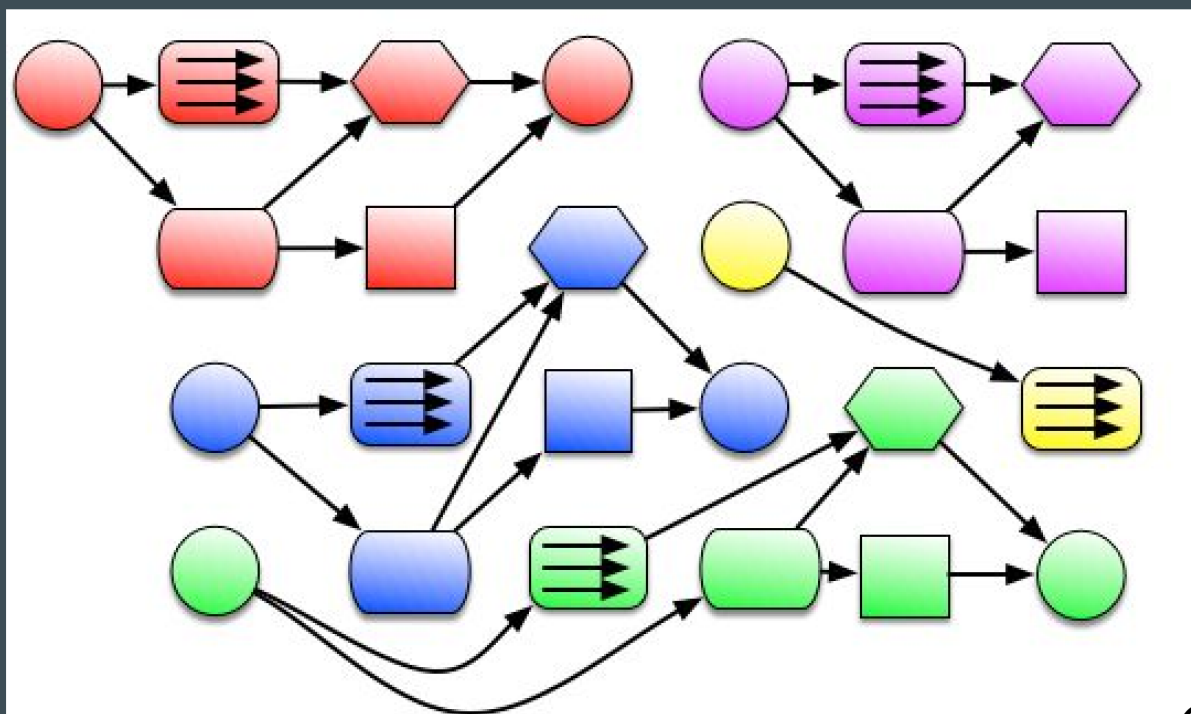
Используется как в ЦЕРН-е:
Atlas, LHCb, FCC, ...

Так и в большом количестве
других экспериментов
КМД-3, BESIII и т.д.

Бурно развивается в последнее время:

В рамках более тесной кооперации ATLAS ↔ LHCb,
для стандартизации и параллельной обработки на многих ядрах

Concurrent Gaudi processing sketch



- Essential idea is simple
 - Exploit parallelism between events
 - Within events
 - Within algorithms
- Of course, there is a lot of devil in the details...

вектор развития:

Code cleanup, modernization to c++17
parallelized ROOT I/O

Разные модели параллелизма

Functional approach

Colours represent different events, shapes different algorithms

<https://indico.cern.ch/event/668607/>

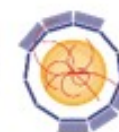
DD4hep - Generic Detector Description for simulation and reconstruction

Overview

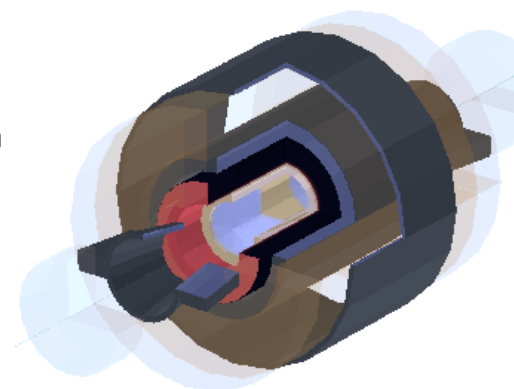
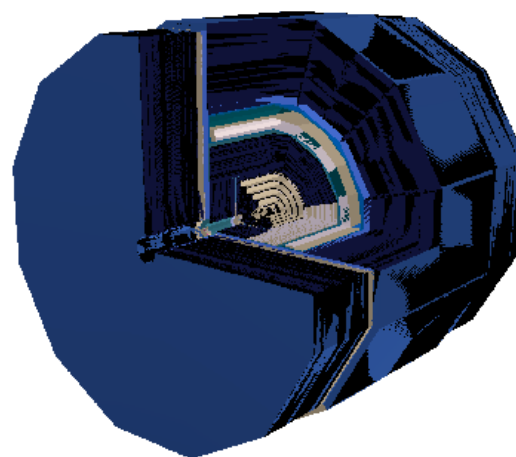
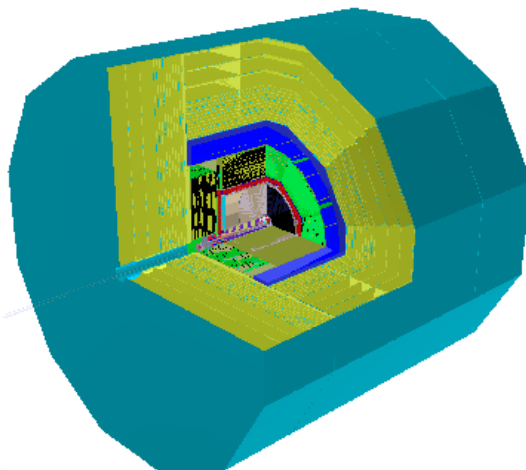
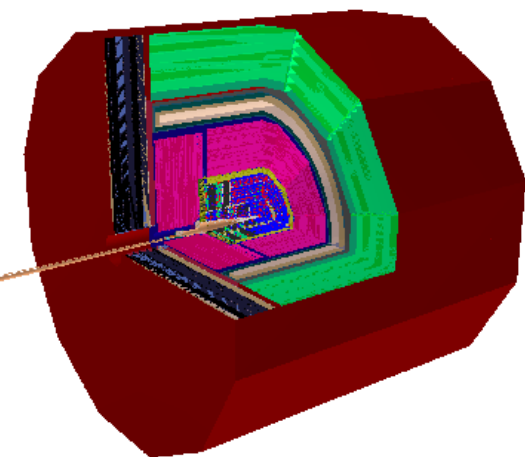
<http://dd4hep.web.cern.ch/>

<http://fccsw.web.cern.ch/fccsw/tutorials/FCCSW/Detector/doc/DD4hepInFCCSW.html>

- ▶ **Complete Detector Description**
 - ▶ Providing geometry, materials, visualization, readout, alignment, calibration...
- ▶ **Supports full experiment life cycle**
 - ▶ Detector concept development, detector optimization, construction, operation
 - ▶ Facile transition from one stage to the next
- ▶ **Single source of information → consistent description**
 - ▶ Use in simulation, reconstruction, analysis, etc.
- ▶ **Ease of Use**
- ▶ **Few places for entering information**
- ▶ **Minimal dependencies**
- ▶ **AIDA-2020 and HSF member project**



Users of DD4hep



DD4hep
DDG4

DD4hep
DDG4

DD4hep
DDG4

DD4hep

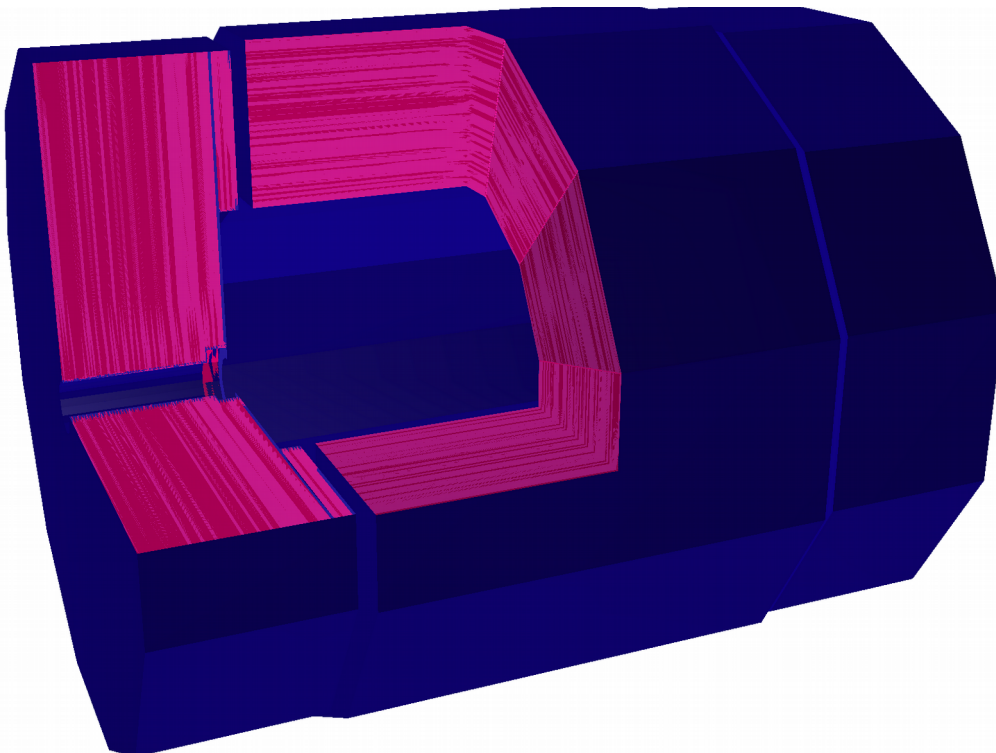
► The user base for DD4hep is growing
LHCB, CMS Under investigation

The Generic Detector Palette

- ▶ Generic driver available → scalable and flexible
- ▶ Parameters are provided in compact XML files, e.g.

```
<detector id="15" name="HCal" type="GenericCalBarrel_o1_v01" readout="HCalCollection">
  <envelope vis="HCALVis">
    <shape type="PolyhedraRegular" numsides="HCal_sym" rmin="HCal_rmin" rmax="HCal_rmax" dz="HCal_dz" material="Air"/>
    <rotation x="0*deg" y="0*deg" z="90*deg-180*deg/HCal_symmetry"/>
  </envelope>
  <dimensions numsides="HCal_sym" rmin="HCal_rmin" z="HCal_dz*2"/>
  <layer repeat="(int) HCal_layers" vis="HCalLayerVis">
    <slice material="Steel235" thickness="0.5*mm" vis="HCalAbsorberVis" radiator="yes"/>
    <slice material="Steel235" thickness="19*mm" vis="HCalAbsorberVis" radiator="yes"/>
    <slice material="Polystyrene" thickness="3*mm" sensitive="yes" limits="cal_limits"/>
    <slice material="Copper" thickness="0.1*mm" vis="HCalCopperVis"/>
    <slice material="PCB" thickness="0.7*mm" vis="HCalPCBVis"/>
    <slice material="Steel235" thickness="0.5*mm" vis="HCalAbsorberVis" radiator="yes"/>
    <slice material="Air" thickness="2.7*mm" vis="InvisibleNoDaughters"/>
  </layer>
</detector>
```

- ▶ You can scale, change layers, radii and compositions...
- ▶ Propagate visualization attributes to Display



XML description

+

C++ driver code

(python driver)

already large palette of simple
'standard HEP detectors' provided
in DDDetectors

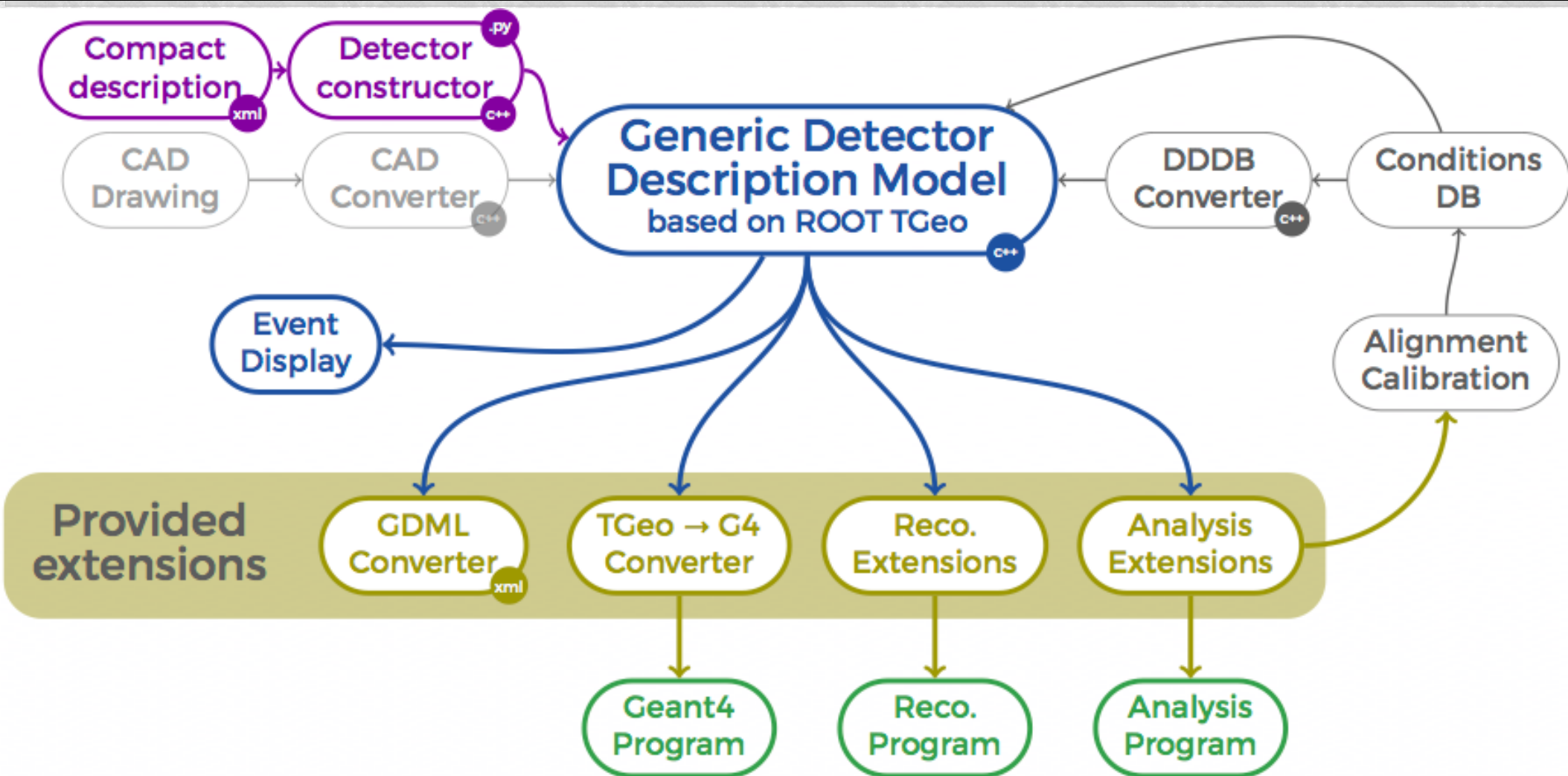
C++ driver code

```
static Ref_t create_detector(LCDD& lcdd,
                             xml_h e, SensitiveDetector sens) {
```

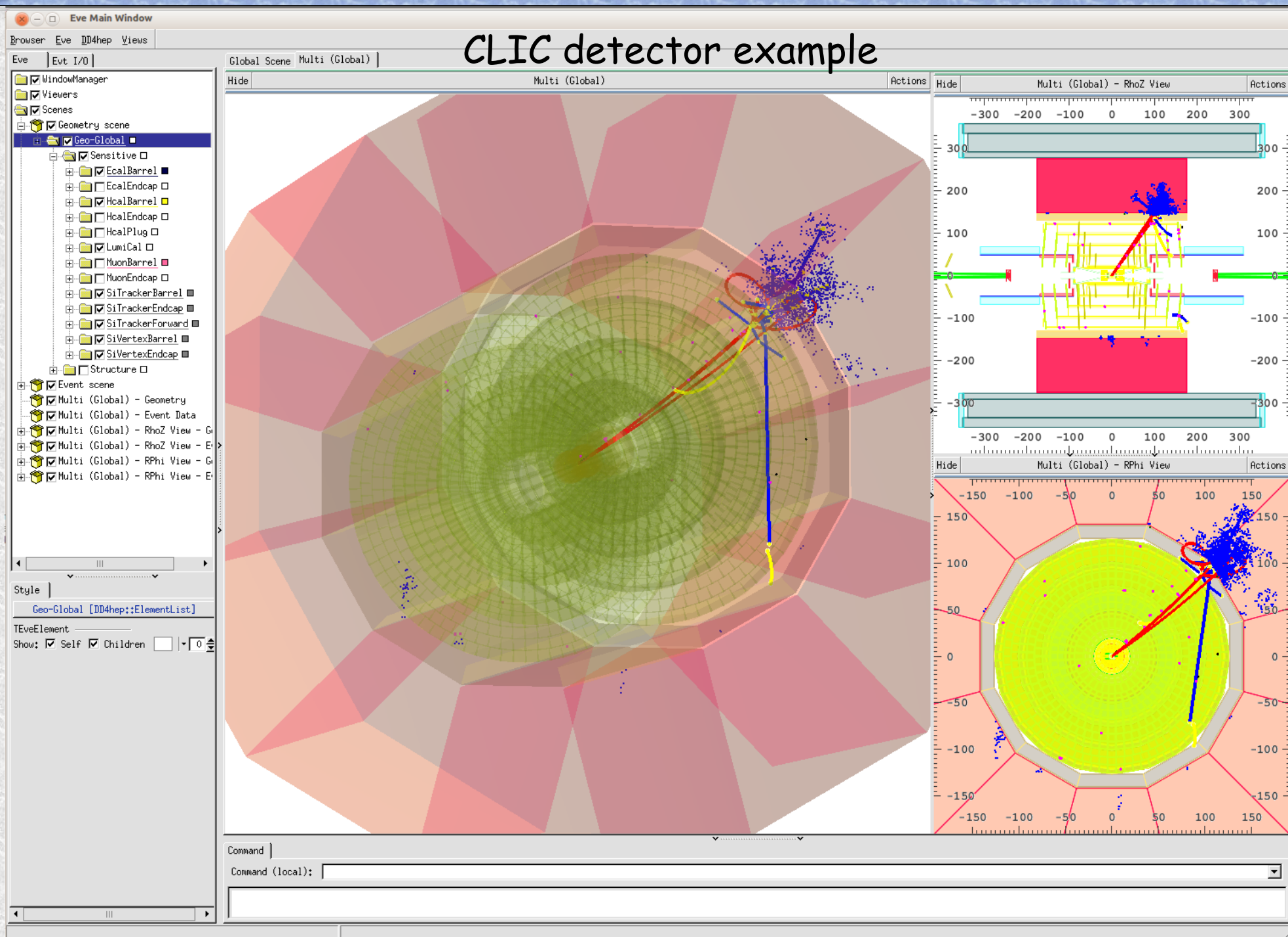
```
    xml_det_t x_det = e;
    Layering layering(x_det);
    xml_comp_t staves = x_det.staves();
    xml_dim_t dim = x_det.dimensions();
    DetElement sdet(det_name, x_det.id());
    Volume motherVol = lcdd.pickMotherVolume(sdet);
```

```
    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);
```



DDRec - для целей реконструкции
DD Eve - визуализация
DDAlign - alignment
DDCond - interface to conditions DB



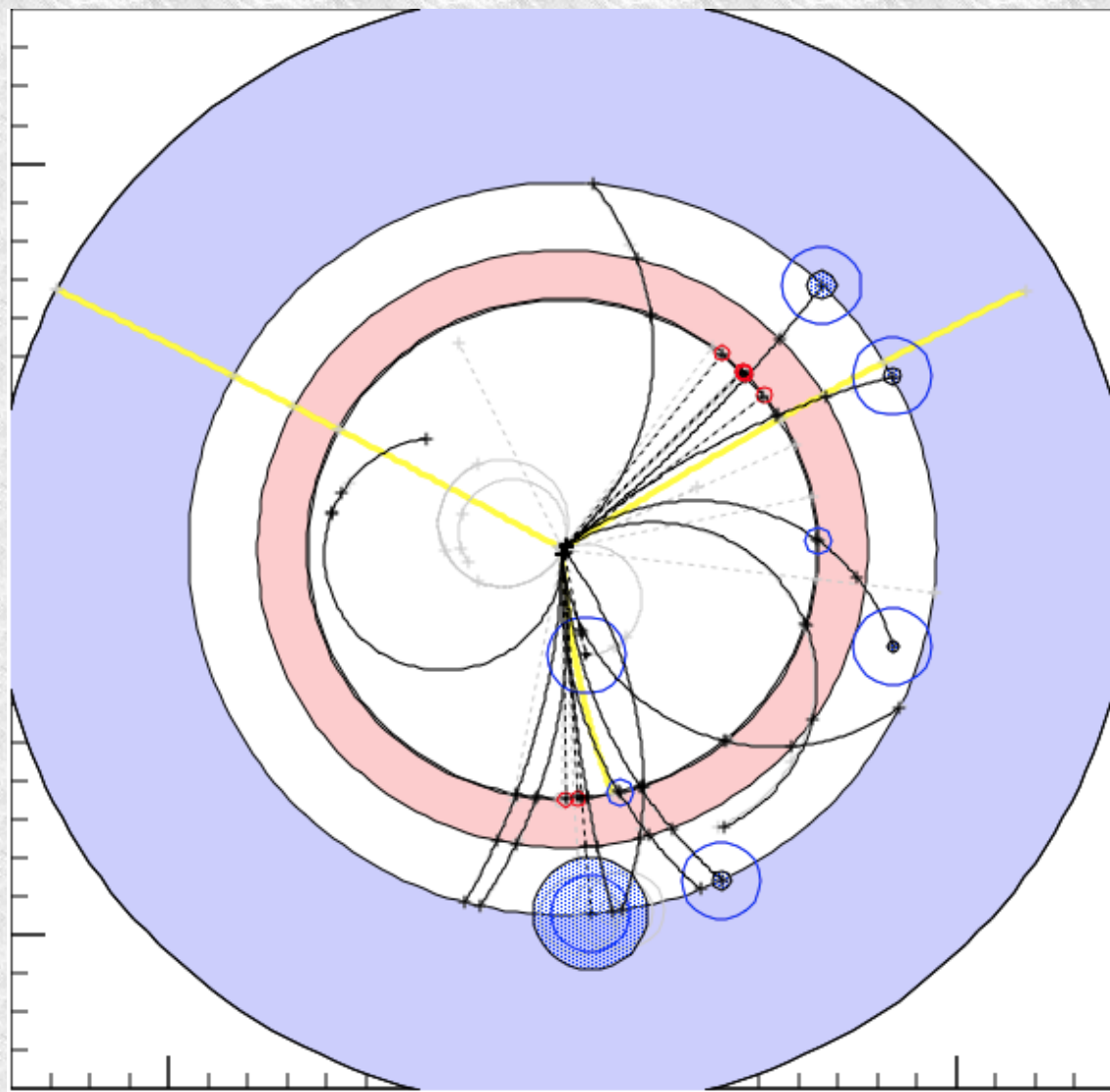
пакет **PAPAS** (The PArametrized PArticle Simulation)

Развивается для **FCC-ee**

Достаточно быстро можем
сделать параметрическое
моделирование:

Описание подсистем детектора:

- × Длина, радиус
- × Геометрическая эффективность,
- × Эффективность
- × Разрешение
- × характерный размер кластера



<https://github.com/HEP-FCC/papas>

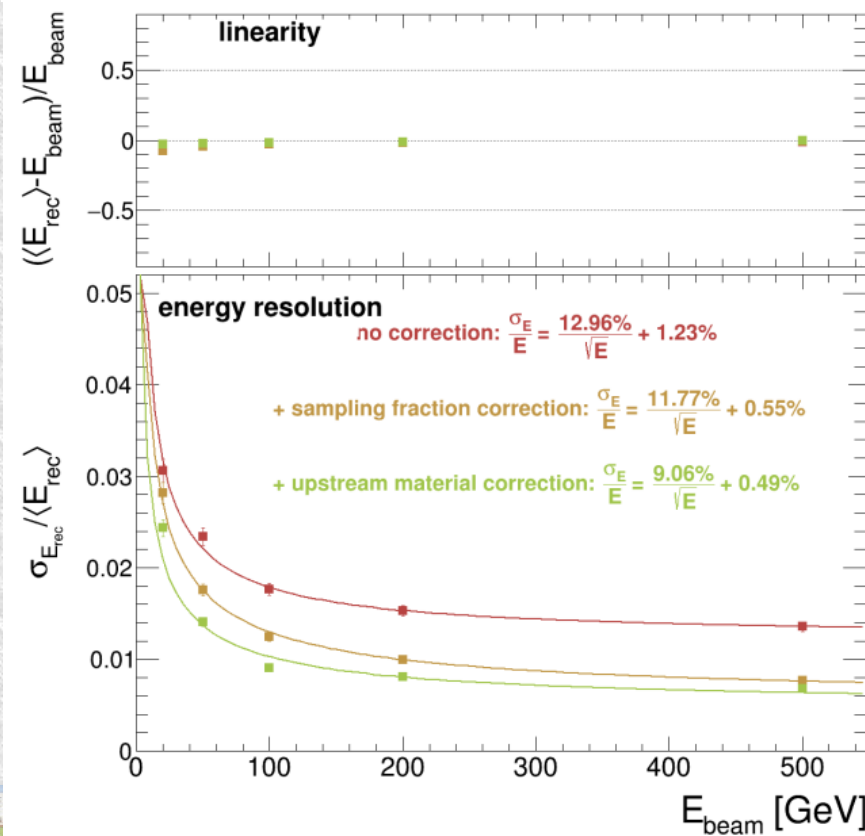
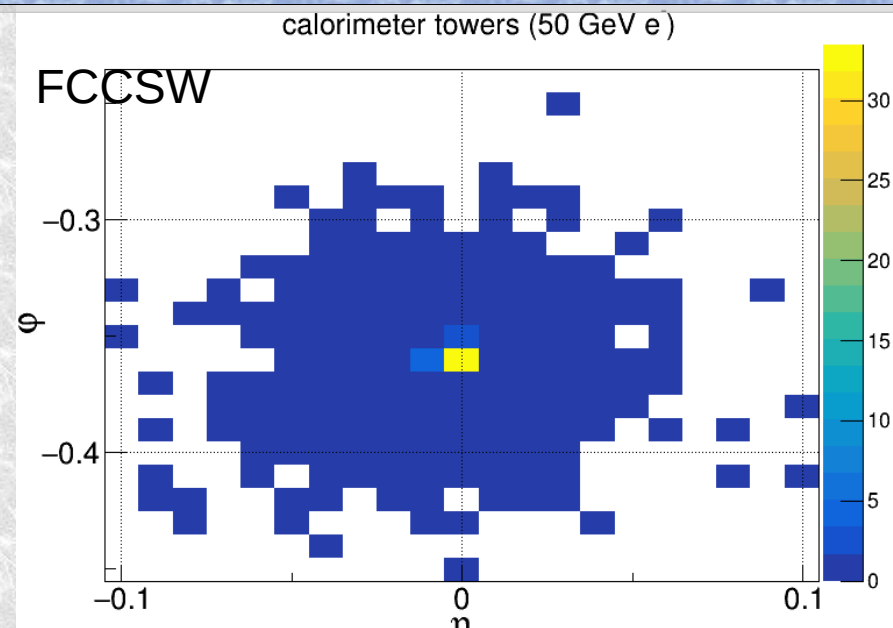
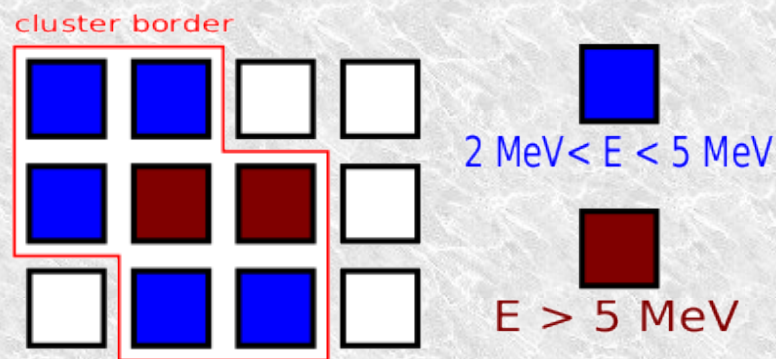
<https://indico.cern.ch/event/556692/contributions/2592513/>

12

Калориметр

Есть алгоритм поиска кластеров в FCCSW
(вдохновленный ATLAS-овским)

Можем относительно быстро адаптировать
используемые у нас в ИЯФ на KMD3,.....



Трековая часть

ATLAS вынес и развивает трековый код в отдельный эксперимент независимый пакет:

ACTS - A Common Tracking Software — возможно будет лучший пакет из доступных на рынке для фитирования с Калмановским фильтром (с точки зрения производительности) <http://acts.web.cern.ch/>

x есть интеграция с FCCSW (Gaudi + fcc-EDM)

x геометрия оптимизированная для реконструкции из DD4hep через плагин

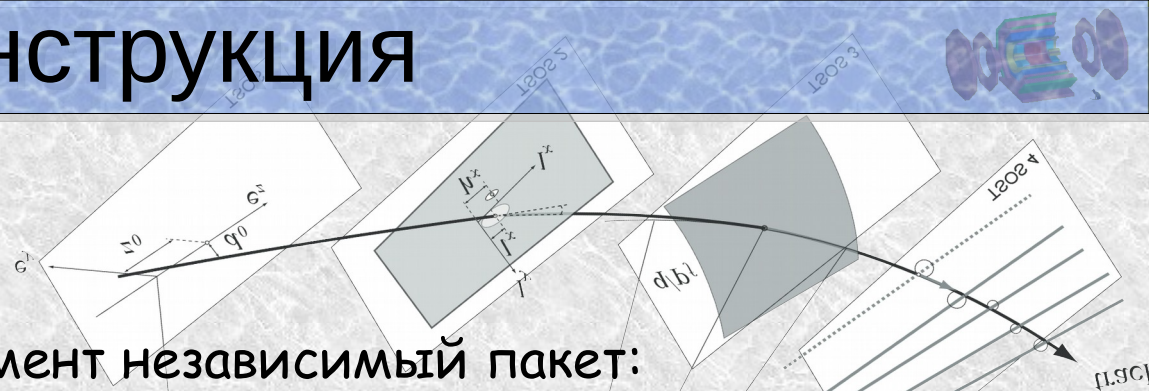
Pattern recognition — поиск треков

За относительно недолго можно адаптировать программу используемую для fullstereo ДК MEG2

PatRec пакет

корни из Alice, развивался/использовался для MEG1, ILC 4th concept, mu2e DCH proposal

Возможно надо будет и так встраивать в FCCSW для ДК детектора IDEA FCC-ee



Относительно быстро мы сможем прикрутить генераторы:
Rhadr с КМД-3, BESIII, EvtGen

Написать параметрическое быстрое моделирование
Для быстрого анализа каких-то процессов

Для полного моделирования надо люди которые будут
описывать геометрию подсистем, оцифровка

Реконструкция в подсистемах

Детальный анализ ключевых процессов

**У Кого есть желание и возможность внести вклад
в развитие моделирования, реконструкции, сделать анализ процессов
Вы нам нужны, присоединяйтесь!**

