Software framework for the Super Charm-Tau factory detector project

Anastasiia Zhadan

Budker Institute of Nuclear Physics, Novosibirsk, Russia

19 May 2021

BINP

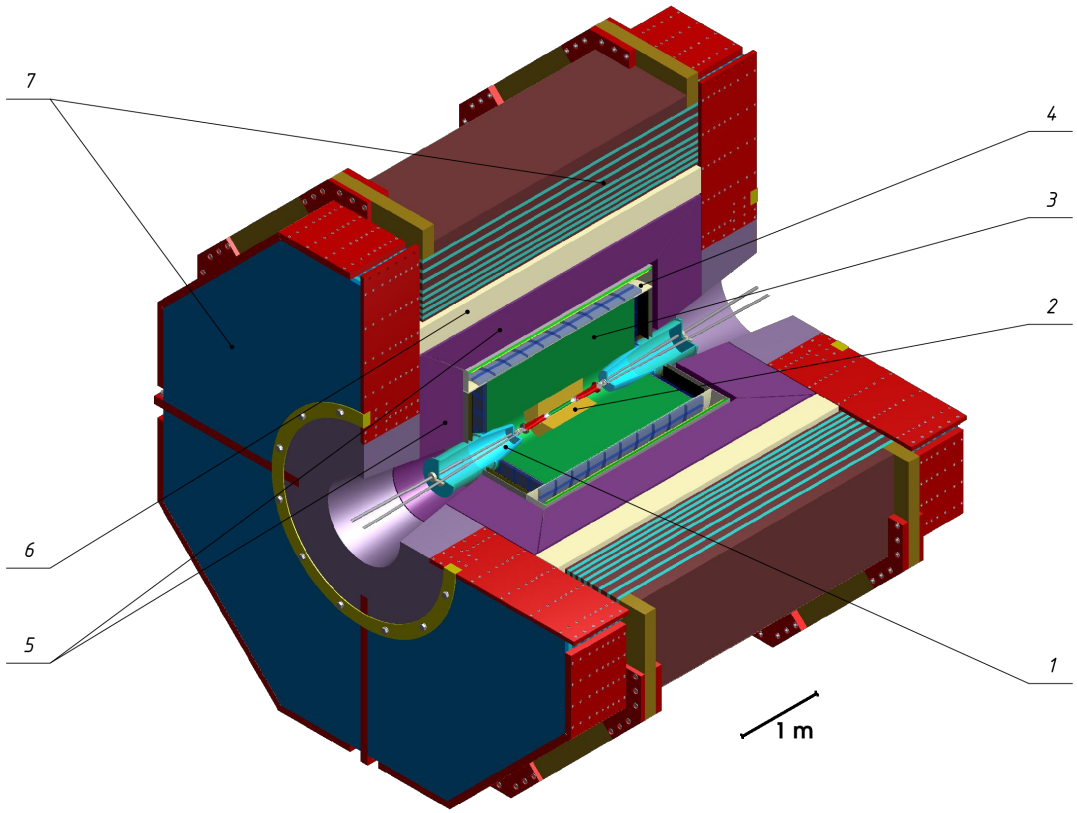
CREMLIN PLUS
Connecting Russian and European Measures for Large-scale Research Infrastructures

vCHEP 2021

Detector overview

Requirements:

- Occupancy 350 kHz
- Good energy and momentum resolution
- High detection efficiency of soft tracks
- Best possible π/K and π/μ separations
- Minimal CP detection asymmetry



	subsystem	options		subsystem	options
1	Beam pipe	beryllium	**2**	Inner tracker	TPC, cGEM, Si-strip
3	Main tracker	drift chamber	**4**	PID system	FARICH, DIRC
5	Calorimeter	CsI, LYSO, LXe	**6**	Magnet	thin coil?
7	Muon system	Scintillators, RPC, ...			

Software for the project

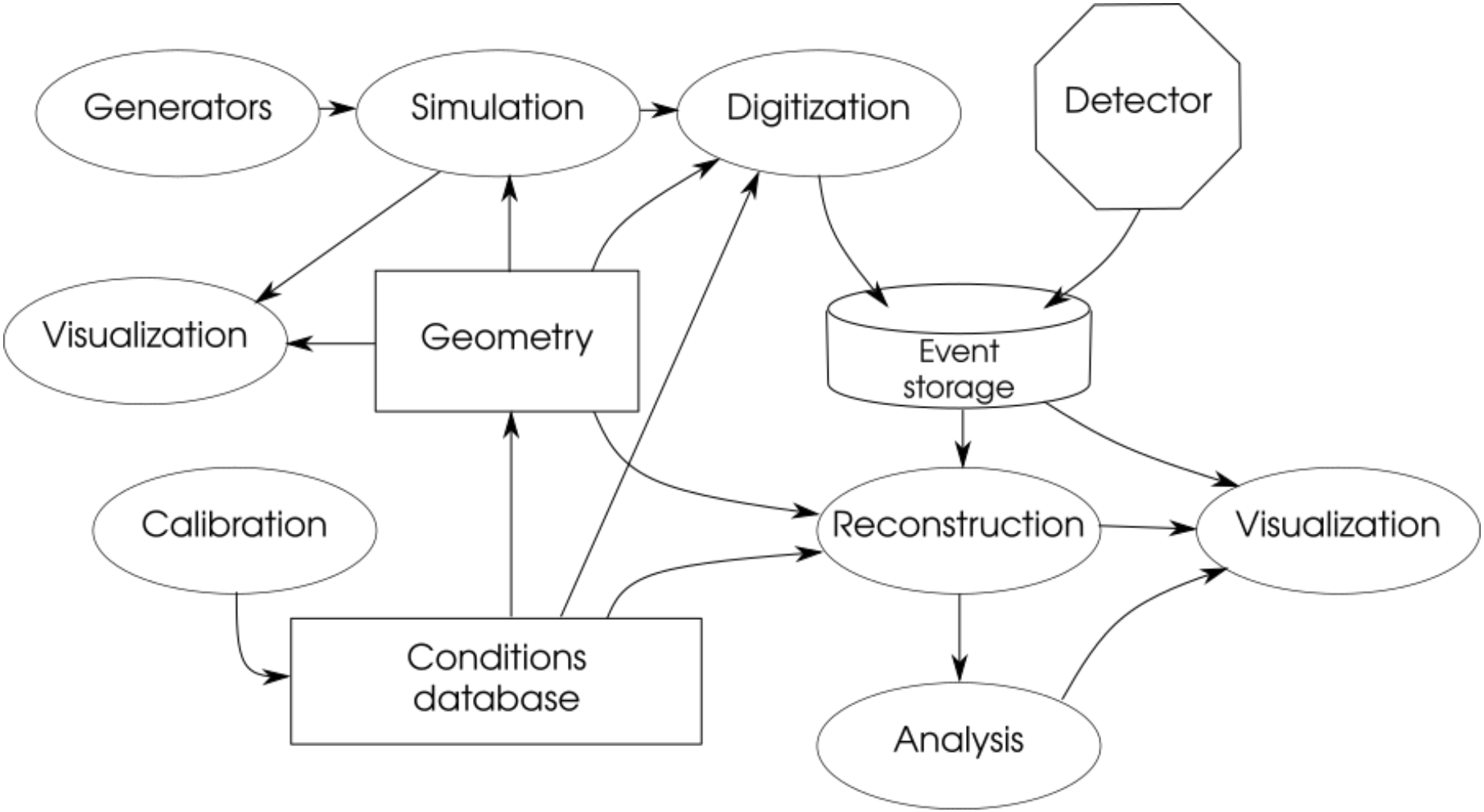
A HEP software framework

A typical HEP experiment requires complete stack of relevant software:

- event generators,
- parametric and full detector simulation,
- event reconstruction algorithms,
- online event interpretation for trigger decisions,
- event data model (EDM),
- I/O interface to conditions data base,
- I/O interface to data storage,
- offline data analysis algorithms,
- build system and release management software.

Software for the project

Framework elements and data flows



The Aurora framework

- Based on Gaudi
- Conventional and recently emerged HEP software tools:
  - ROOT, Geant4
  - DD4Hep (Key4HEP)
- Other experiments software
  - Belle II, ILC, FCCSW...
- Build & configuration system inspired by ATLAS Athena
- `lcgcmake` system to build external packages
- Nightly builds
- Standard computing environment is Scientific Linux 7 x86_64, GCC9 + Python2&3

The Aurora framework

- Based on Gaudi
- Conventional and recently emerged HEP software tools:
  - ROOT, Geant4
  - DD4Hep (Key4HEP)
- Other experiments software
  - Belle II, ILC, FCCSW...
- Build & configuration system inspired by ATLAS Athena
- `lcgcmake` system to build external packages
- Nightly builds $\rightarrow$ **CI**
- Standard computing environment is Scientific Linux 7 x86_64, GCC9 + Python2&3

The Aurora framework

- Based on Gaudi
- Conventional and recently emerged HEP software tools:
  - ROOT, Geant4
  - DD4Hep (Key4HEP)
- Other experiments software
  - Belle II, ILC, FCCSW...
- Build & configuration system inspired by ATLAS Athena
- `lcgcmake` system to build external packages
- Nightly builds $\rightarrow$ **CI**
- Standard computing environment is Scientific Linux 7 x86_64, GCC9 + Python2&3 $\rightarrow$ **pure Python3**

Standalone studies

- Parametric simulation tool for quick estimations of the detector response
- Background simulations with Fluka
- Gas mixture studies and electric field simulations with Garfield for TPC and DC
- CERN team develops TPC variant & adopts track finding algorithms from iLCSoft

Standalone studies

- Parametric simulation tool for quick estimations of the detector response
  $\rightarrow$ **now merged into the framework**
- Background simulations with Fluka

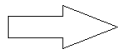
- Gas mixture studies and electric field simulations with Garfield for TPC and DC

- CERN team develops TPC variant
  & adopts track finding algorithms from iLCSoft
  $\rightarrow$ **have a prototype, now incorporating into the framework**

Event Data Model

- ROOT-based
- PODIO to generate C++ classes using yaml file
- Under development → Rebase to EDM4hep?

```
sct::G4Hit:
  Description: "A track hit and with its global pre step and post step positions."
  Author: "A.Zhadan"
  Members:
    - unsigned long long cellId   // Cell id
    - double energy               // Energy
    - double globalTime           // Global time
    - double localTime            // Local time
    - sct::Point preStepPosition  // The pre step point in global frame
    - sct::Point postStepPosition // The post step point in global frame
    - int trackId                 // Track Id from Geant4 of the particle that leave the hit
    - int pdgId                   // PDG Id of the particle that leave the hit
    - sct::LorentzVector momentum // Momentum of the particle that leave the hit
```



h G4Hit.h	C++ G4Hit.cc
h G4HitCollection.h	C++ G4HitCollection.cc
h G4HitConst.h	C++ G4HitConst.cc
h G4HitData.h	C++ G4HitObj.cc
h G4HitObj.h	

Event Generators

The conventional set of event generators available

- Exclusive decays of hadrons and tau lepton
  - EvtGen, Tauola, PHOTOS, Pythia
- Inclusive generators for $e^+e^- \rightarrow$ hadrons
  - preliminary solution based on Pythia
- Generators for luminosity measurements and calibrations
  - MCGPJ, BabaYaga, BBBREM, KKMC...

Status of the software

Geometry in Aurora

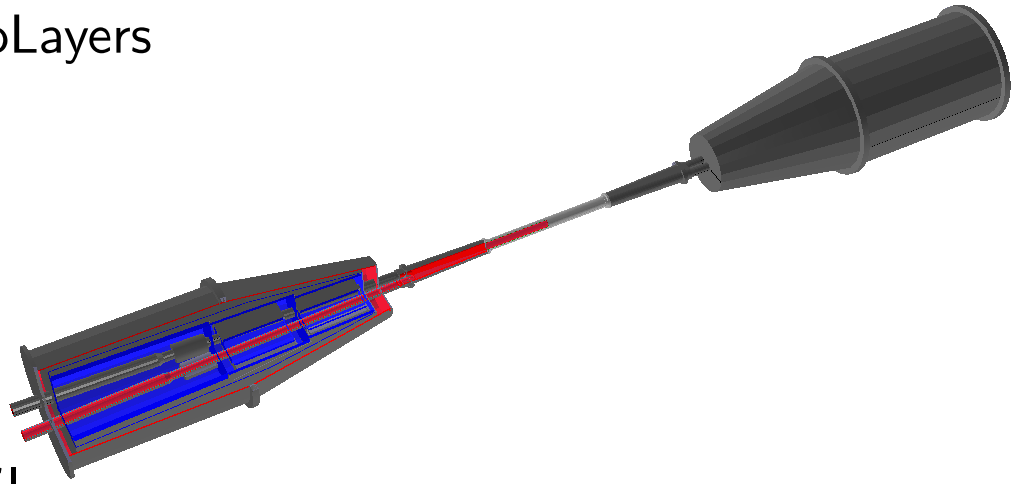
- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI

  (overlaps, material scans. . .)

Status of the software

Geometry in Aurora

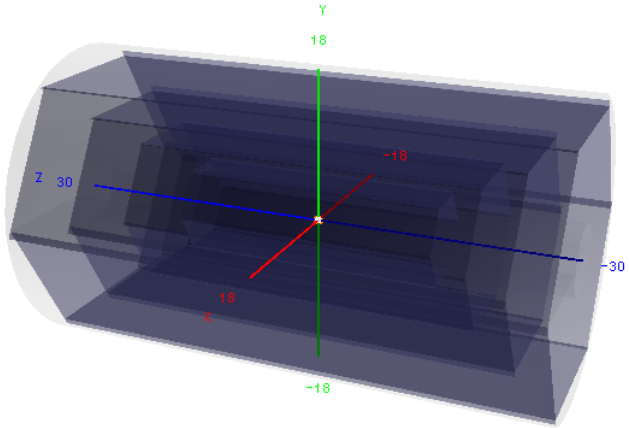
- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)



Status of the software

Geometry in Aurora

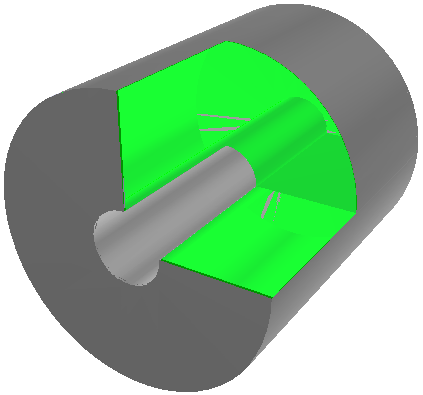
- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)

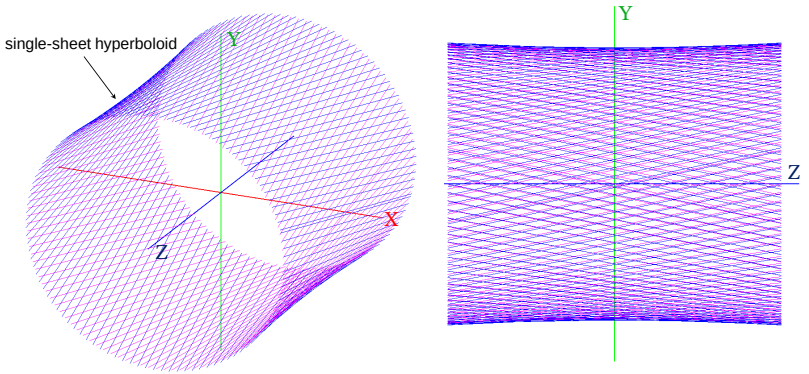


Status of the software

Geometry in Aurora

- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)

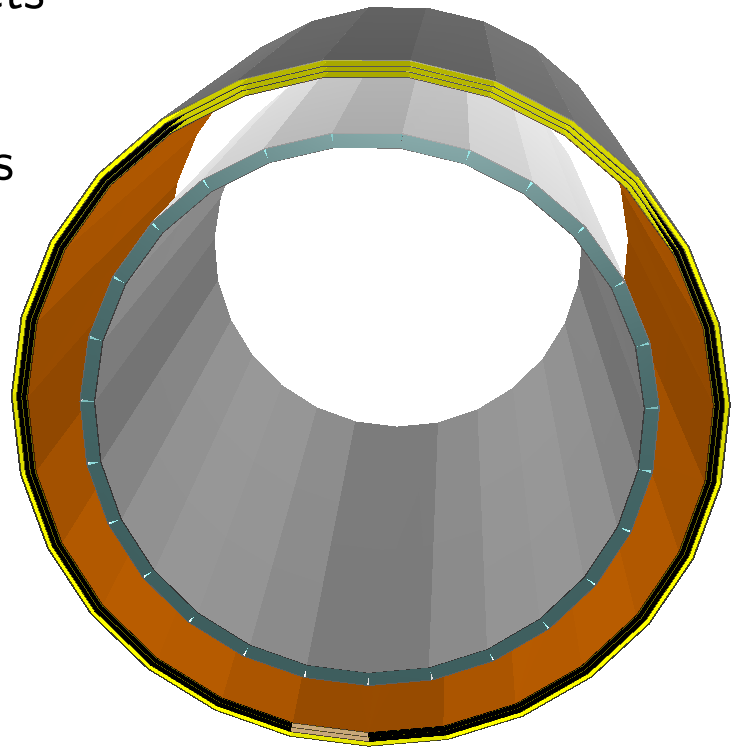




Status of the software

Geometry in Aurora

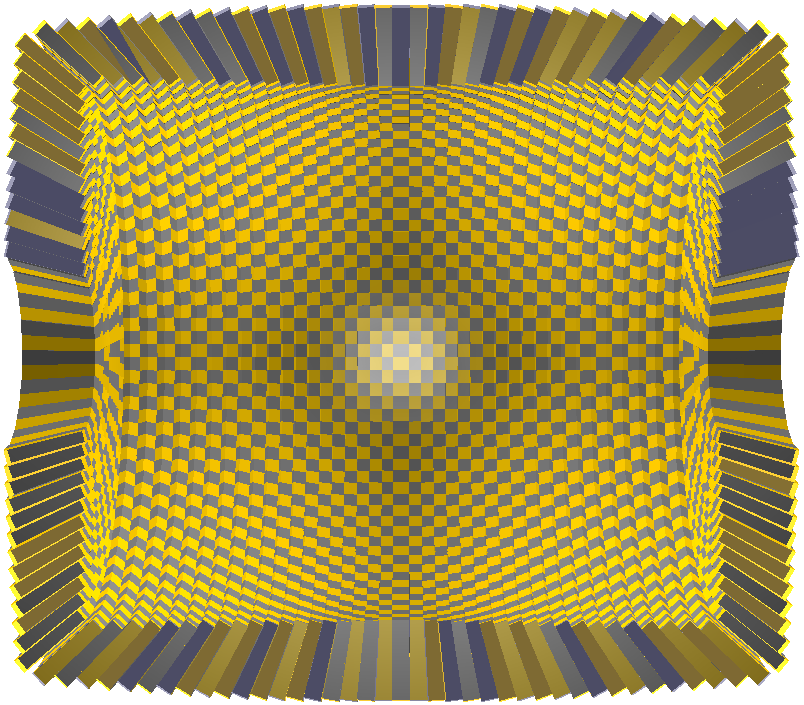
- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)



Status of the software

Geometry in Aurora

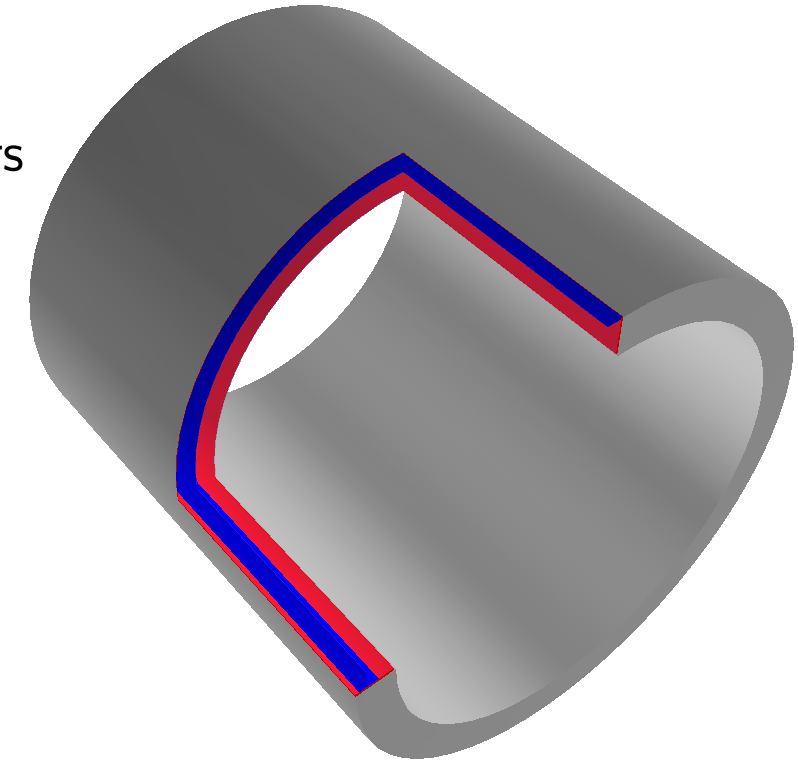
- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)



Status of the software

Geometry in Aurora

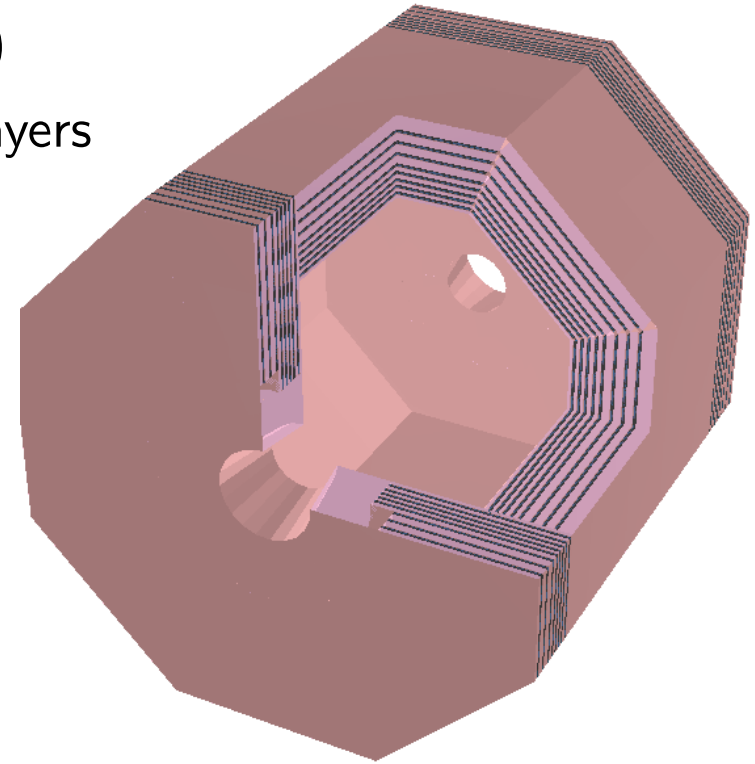
- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)



Status of the software

Geometry in Aurora

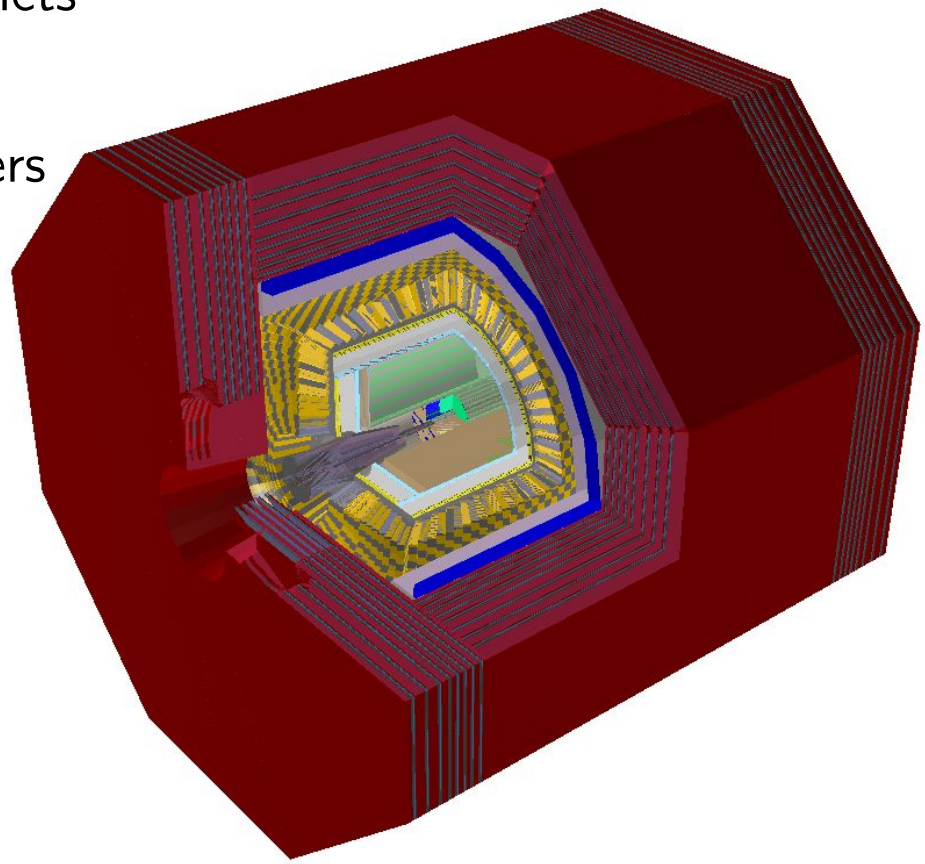
- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)



Status of the software

Geometry in Aurora

- Subsystems implemented at the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools for CI (overlaps, material scans. . .)



We have geometry for at least one option for each subsystem

Status of the software

Simulation

- Set initial parameters via job options file:
  - generate primary particles / read pre-generated events
  - choose active subsystems and select variants
  - tools to save output collections
  - ...
- Geant4 is used for the particle propagation and hit generation
  - G4Hit — information about hit, time, energy deposit, track ID and etc.
  - Special Gaudi tools to save G4Hit for each sensitive detector subsystem

Status of the software

Digitization & Reconstruction

- most subsystems miss separate Digitization stage yet
  - integrated into reconstruction
  - based on standalone studies
  - sample module prepared for Silicon Strip
- 1st stage Reconstruction: individual subsystem level
  - in preparation by subsystem groups
  - Calorimeter and DC most advanced at the moment
- 2st stage Reconstruction: combining subsystems, PID...

Status of the software

Digitization & Reconstruction

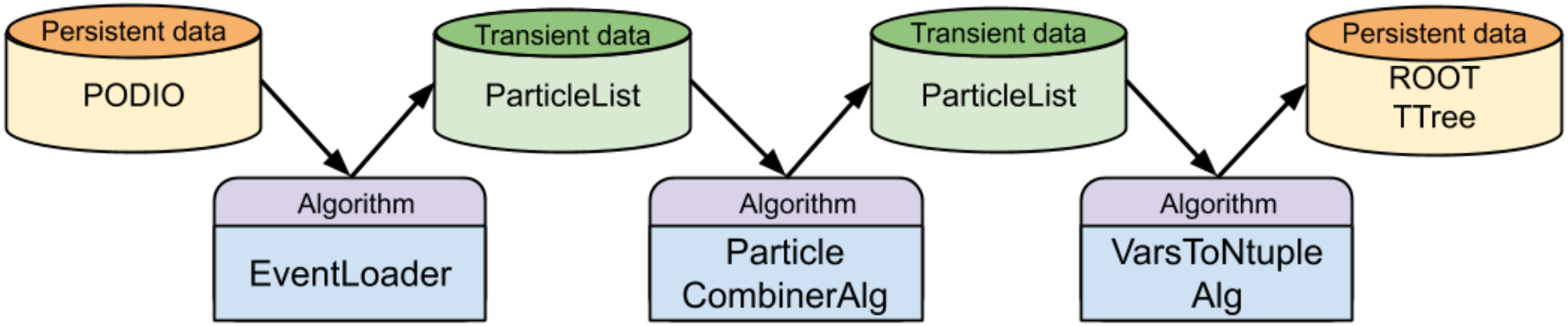
- most subsystems miss separate Digitization stage yet
  - integrated into reconstruction
  - based on standalone studies
  - sample module prepared for Silicon Strip
- 1st stage Reconstruction: individual subsystem level
  - in preparation by subsystem groups
  - Calorimeter and DC most advanced at the moment
- 2st stage Reconstruction: combining subsystems, PID...

  $\rightarrow$ scheduled for this year

Status of the software

Data Analysis

- Adopting Belle II recipes and solutions for analysis
- Base set of analysis algorithms ready:



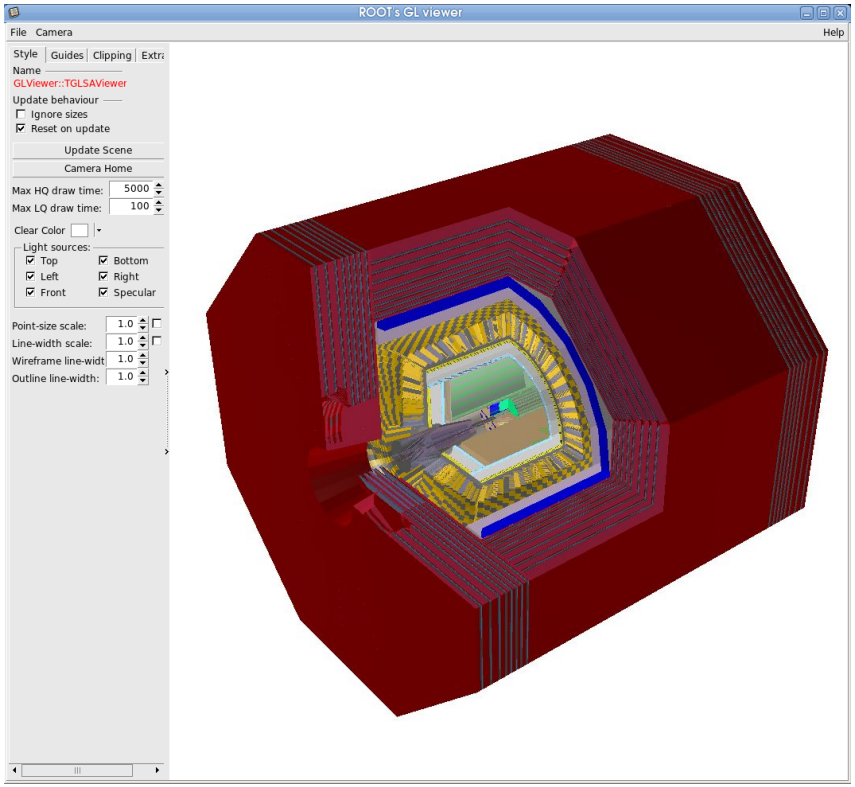
- final particles processing
- kinematic cuts

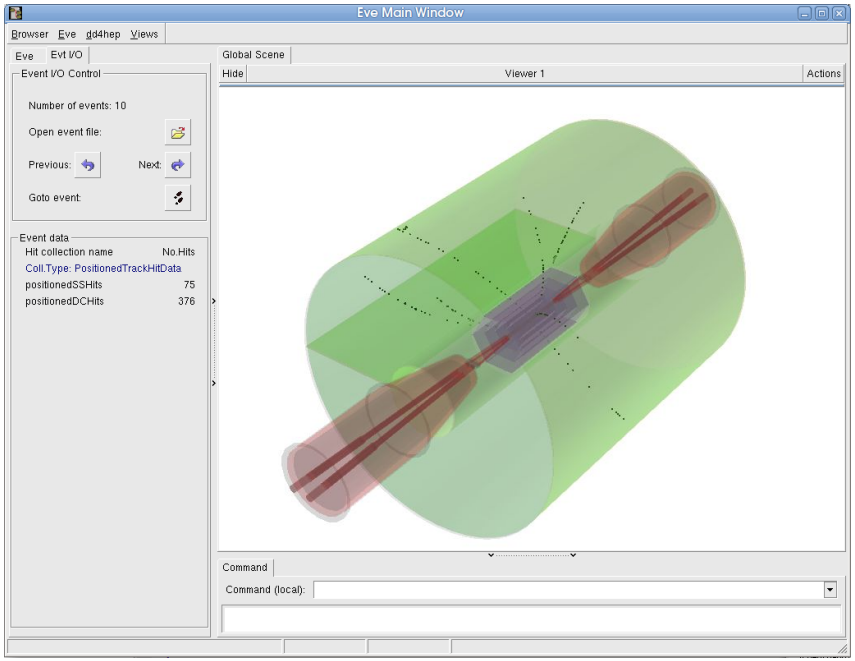
- reconstructuion of decayed particle
- kinematic cuts

- kinematic parameters calculation
- particle selection

Status of the software

Detector/Event Display





- Geometry display tool is ready
- Base Event display (DDEve-based) available, lots of things to improve

Conclusions

The Aurora framework now contains all components minimally required at the present stage of the SCT detector project development:

- set of primary event generators,
- usable in analysis parameterized simulation,
- detector geometry (with at least basic description for all detector elements, and several options for some subsystems),
- full Geant4-based simulation,
- sample digitization module,
- reconstruction modules (from basic to really advanced, depending on subsystem),
- analysis and job configuration tools,
- test and service tools.

Conclusions

The Aurora framework now contains all components minimally required at the present stage of the SCT detector project development:

- set of primary event generators,
- usable in analysis parameterized simulation,
- detector geometry (with at least basic description for all detector elements, and several options for some subsystems),
- full Geant4-based simulation,
- sample digitization module,
- reconstruction modules (from basic to really advanced, depending on subsystem),
- analysis and job configuration tools,
- test and service tools.

This allows us to announce the Aurora 1.0.0 release

Acknowledgments

We are grateful to the Belle II collaboration and to the FCCSW project for access to their software

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871072 (the CREMLINplus project)

Thank you for attention

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871072

Backup

Further software development

The nearest goals for the software development are:

- implementation of digitization modules for all subsystems
- further reconstruction improvements, including adoption of some high-level tools, i. e. track finding,
- improvement of detector and event visualization tools. The underlying DDEve has been not actively developed recently, so this is an area of possible backward contribution to DD4Hep
- distribution of the software via CvmFS