

4D Tracking studies

Valentina Cairo

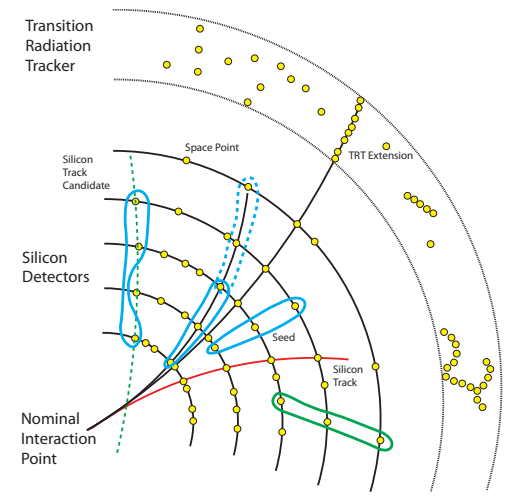
April 28th 2021

Our first goal is to find a framework that best suits our needs for simulation/reconstruction studies

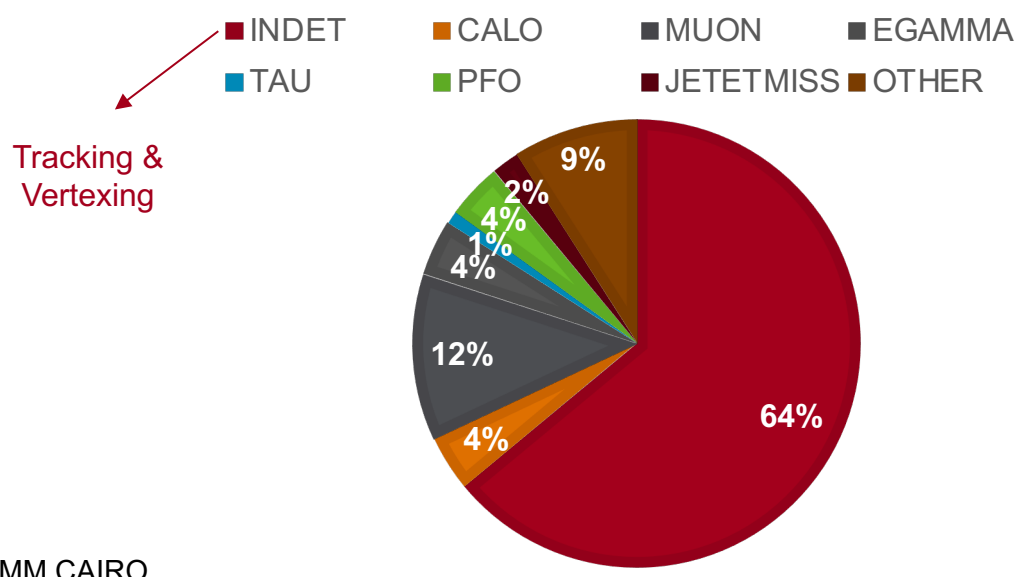
- Ideally, we want to be able to:
 - create certain detector geometries which include 4D tracking sensors
 - 2 things to keep in mind: those sensors may have different passive material than 3D sensors and they imply being able to run 4D tracking
 - simulate and reconstruct objects based on these layouts
 - ultimately, we would like to see the impact of such layouts on some physics benchmarks
- These questions depend on a variety of possibilities, i.e. **which detector layout do we want to use as a benchmark?** One of the detectors at future colliders or shall we start from the HL-LHC scenario?
 - Most of what we learn for HL-LHC can be ported over future detectors.

Tracking in ATLAS

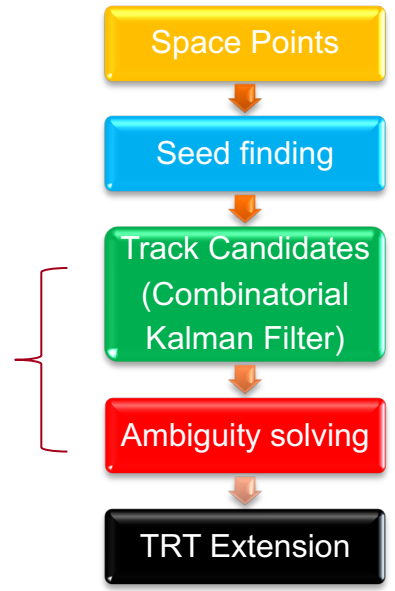
- **ATLAS core software lives in the Athena framework**
 - Tested and used for over 15 years now
 - Based on C++ with python configurations
 - Deals with the entire data & simulation workflow
 - Event generation, simulation, digitization, reconstruction etc
- **Tracking is the most CPU intensive reconstruction component (and it is fundamental)!**



RUN 2 RECONSTRUCTION CPU TIME

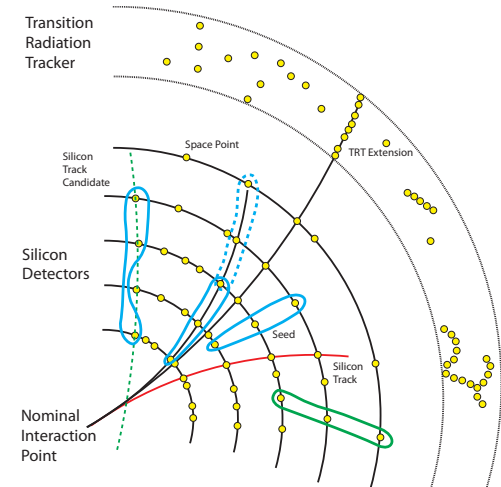


Most of the time is spent here



Tracking in ATLAS

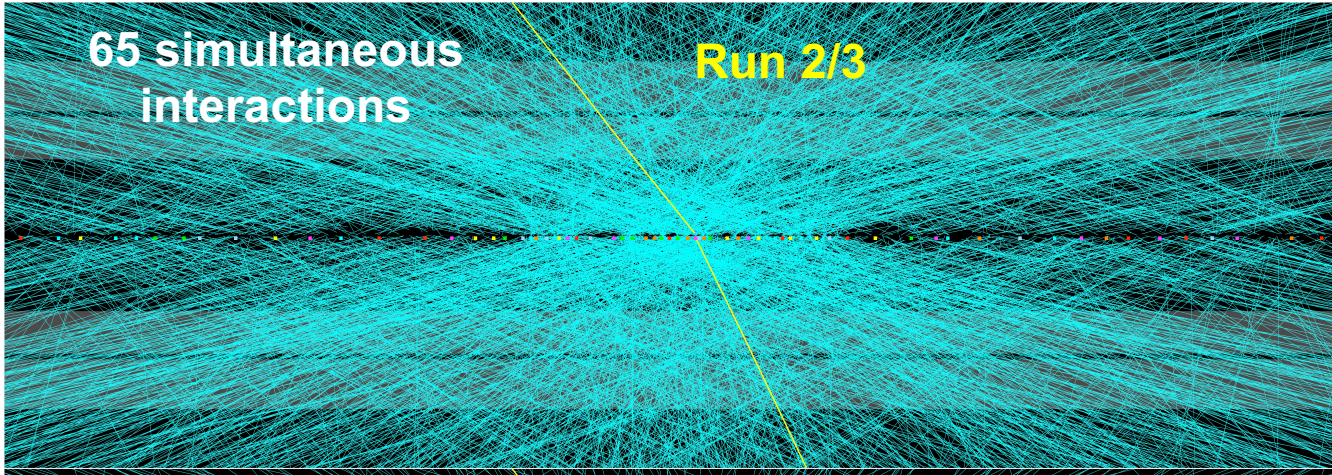
- **ATLAS core software lives in the Athena framework**
 - Tested and used for over 15 years now
 - Based on C++ with python configurations
 - Deals with the entire data & simulation workflow
 - Event generation, simulation, digitization, reconstruction etc
- **Tracking is the most CPU intensive reconstruction component (and it is fundamental)!**



- Run 3 → x2 faster tracking!
- But... More drastic solutions are needed for Run 4 (HL-LHC, <pile-up> ~200)
- **Time information can help reduce the combinatorics!**

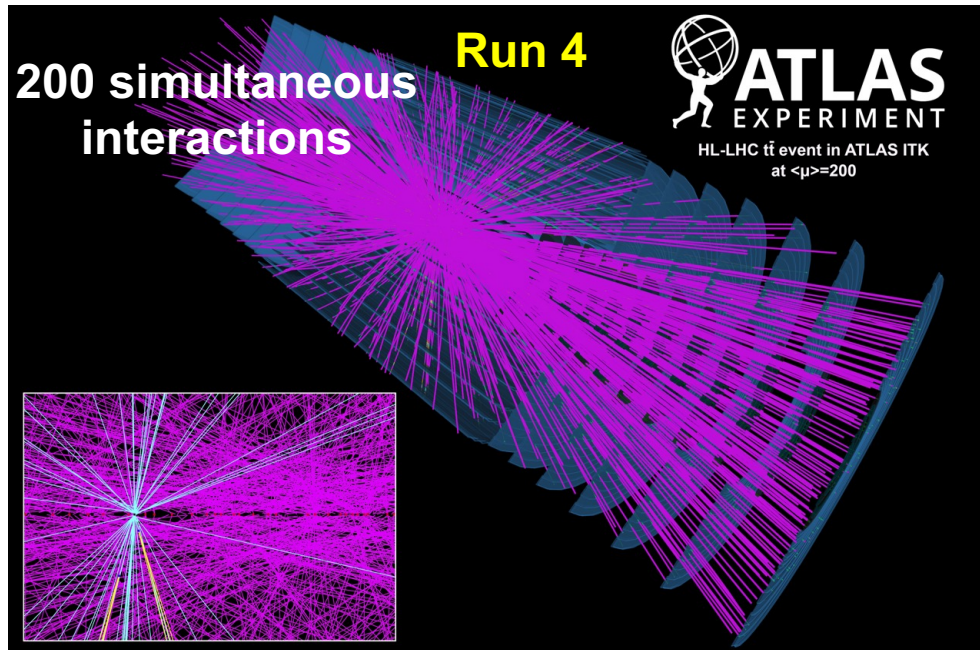


Tracking & Vertexing from Run 2/3 to Run 4



Run Number: 336852, Event Number: 883966264

Date: 2017-09-29 09:19:23 CEST



The challenge

Tracking and vertexing present the greatest challenges to CPU when hit combinatorics explodes

→ Fast and accurate tracking software is necessary

Write **thread-safe code** which can be executed concurrently to maximize memory access

e.g. on-going migration of ATLAS's software framework (Athena) to a multi-threading framework (AthenaMT)

Use **performant software** technology

Improve tracking & vertexing strategy

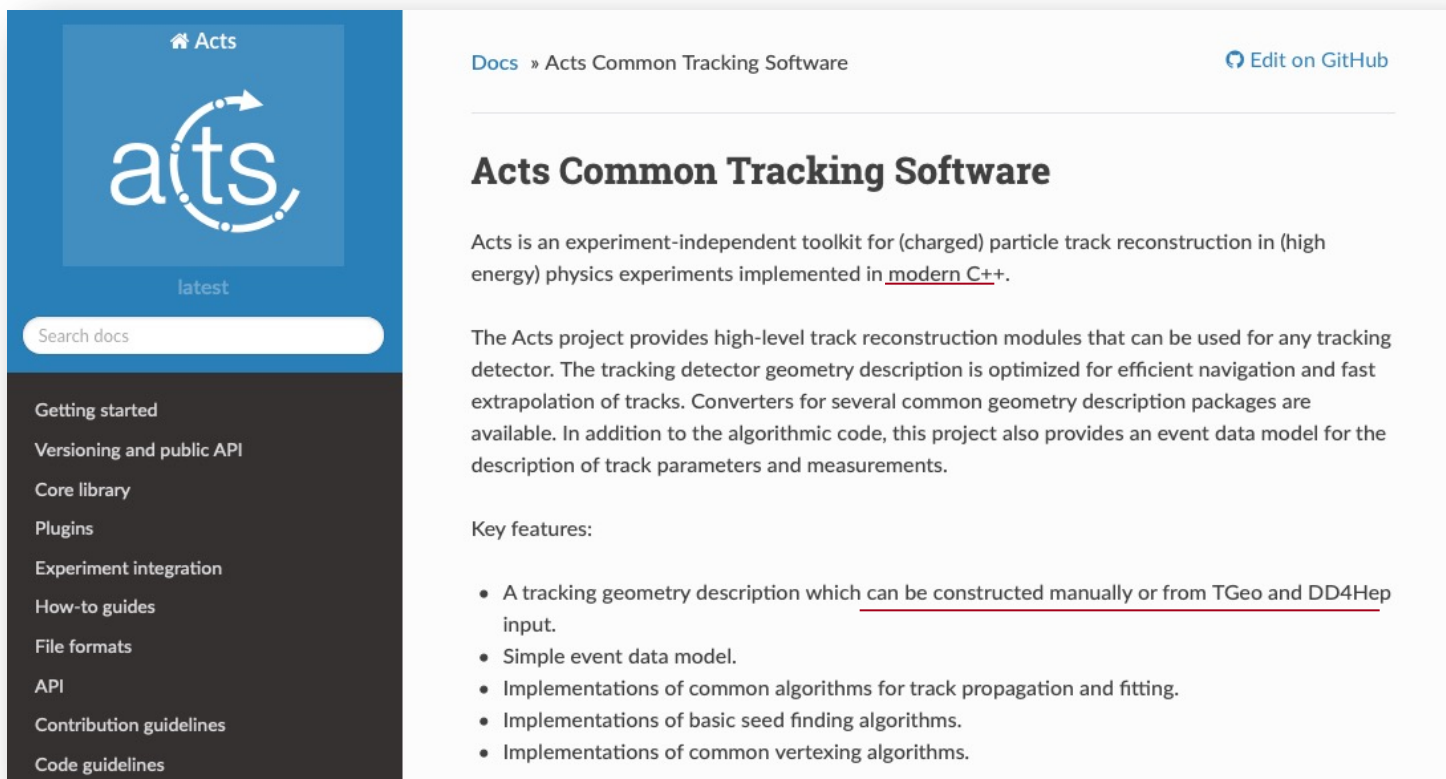
We want to do this!

Adapt to modern hardware architecture

ACTS: A Common Tracking Software

A couple of years ago, a new initiative took off in ATLAS targeting HL-LHC:

- A Common Tracking Software, **ACTS**
 - An **open-source platform for tracking algorithms R&D**
 - Collaboration across experiments:
 - ATLAS, LHCb, FCC-hh, FASER, CEPC and external contributors



Acts

Docs » Acts Common Tracking Software [Edit on GitHub](#)

Acts Common Tracking Software

Acts is an experiment-independent toolkit for (charged) particle track reconstruction in (high energy) physics experiments implemented in modern C++.

The Acts project provides high-level track reconstruction modules that can be used for any tracking detector. The tracking detector geometry description is optimized for efficient navigation and fast extrapolation of tracks. Converters for several common geometry description packages are available. In addition to the algorithmic code, this project also provides an event data model for the description of track parameters and measurements.

Key features:

- A tracking geometry description which can be constructed manually or from TGeo and DD4Hep input.
- Simple event data model.
- Implementations of common algorithms for track propagation and fitting.
- Implementations of basic seed finding algorithms.
- Implementations of common vertexing algorithms.

What about hit time propagation?

A couple of years ago, a new initiative took off in ATLAS targeting HL-LHC:

- A Common Tracking Software, **ACTS**
 - An open-source platform for tracking algorithms R&D
 - Collaboration across experiments:
 - ATLAS, LHCb, FCC-hh, FASER, CEPC and external contributors



Docs » Acts Common Tracking Software

Edit on GitHub

Acts Common Tracking Software

Acts is an experiment-independent toolkit for (charged) particle track reconstruction in (high energy) physics experiments implemented in modern C++.

The Acts project provides high-level detector. The tracking detector geometry, track extrapolation of tracks. Converter and other utilities are available. In addition to the algorithm description of track parameters and

Key features:

- A tracking geometry description
- Simple event data model.
- Implementations of common algorithms
- Implementations of basic seed generation
- Implementations of common vertexing

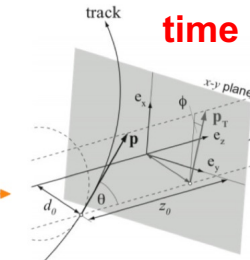
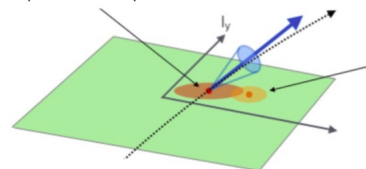
Track Parameter and Measurement EDM

- Coefficients and covariance are described using fixed-size Eigen::Matrix

TrackParameters: $(l_0, l_1, \phi, \theta, q/p, t)$

- l_0, l_1 : Coordinate in local surface frame (Bound Parameters) or curvilinear frame (Curvilinear Parameters)
- Meaning of l_0, l_1 varies depending on surface type
- p, ϕ, θ : Momentum and direction
- q : Charge
- t : Per-track timing info

Supports multi-component track parameters representation



e.g. perigee track parameters at perigee surface $l_0 = d_0, l_1 = z_0$

Measurement:

- Contains a SourceLink to original detector measurement
- Uses std::variant to as a wrapper of heterogeneous measurement (1D, 2D, ...)

ACTS supports time propagation!

ACTS workshop

ACTS components and functionalities

Infrastructure consolidation and new features since last ACTS workshop



- Fast simulation engine (see M. Kiehn's tutorial)
- Integrated into acts-core
- A light-weight Gaudi style test framework for event processing, integration and concurrency test
- Integration into acts-core to ease maintenance and performance monitoring

Collaborating with the ACTS Team

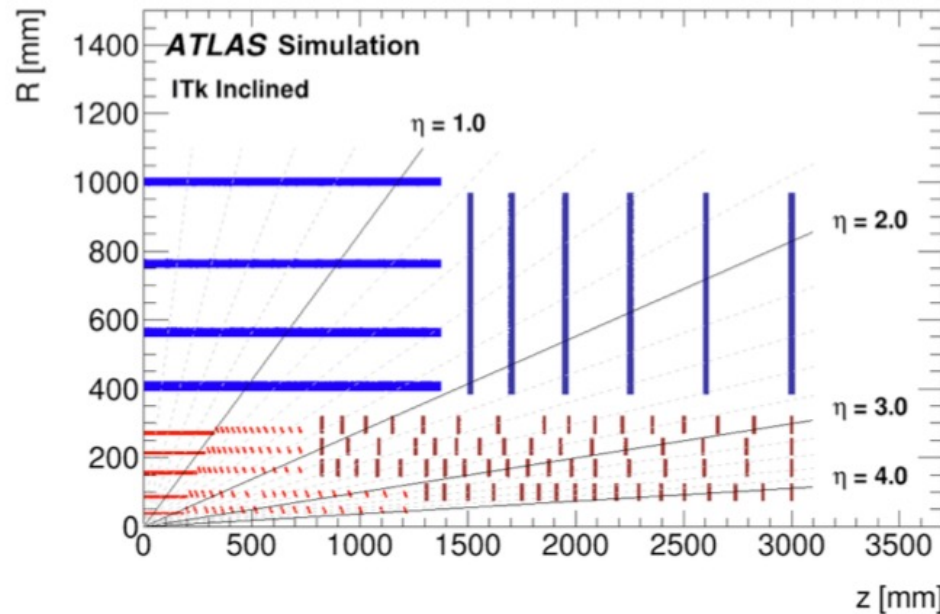
Ariel and I has a meeting with the a CERN team of ACTS developers (Andreas Salzburger and Fabian Klimpel)

- The time component is fully integrated in the ACTS propagation (and measurement structure), i.e. **you can already now define the time as a measurement parameter (with error), and the propagation engine will use it as a standard parameter to be propagated** (including error propagation).
- Also the ACTS Kalman Filter can already swallow time measurements and would simply perform update and filtering steps on it
- **Very Recent** PR (<https://github.com/acts-project/acts/pull/461>) for the new **ActsFatras**/Digitization where, as a start, at least time smearing will be supported for measurements
- The developers are also happy to collaborate with us on **how to use (and thus validate) the code extending their current unit test validation**

Various potential areas of collaboration

See what we discussed last week during [Ariel's presentation](#)

4D pixels in LS4 ITk



3 key ideas:

Improve track reconstruction:

Speed, efficiency, purity, and IP resolution

Improve physics reconstruction by reducing pileup effects:

Pileup jet suppression, particle flow, b-tagging (track-to-vertex association)

Improve forward jets (VBF):

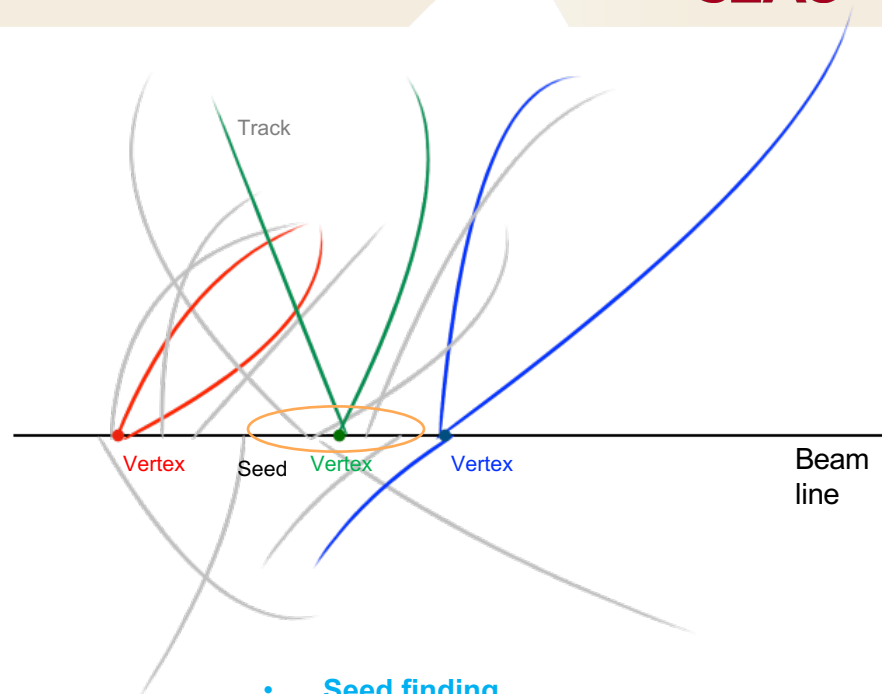
Provide vertex t_0 for HGTD

Tracking in time-ROI: outline of the workflow

One of the studies we proposed is to see if the hit time available in a limited number of layers can **improve vertex reconstruction**:

Outline of the plan:

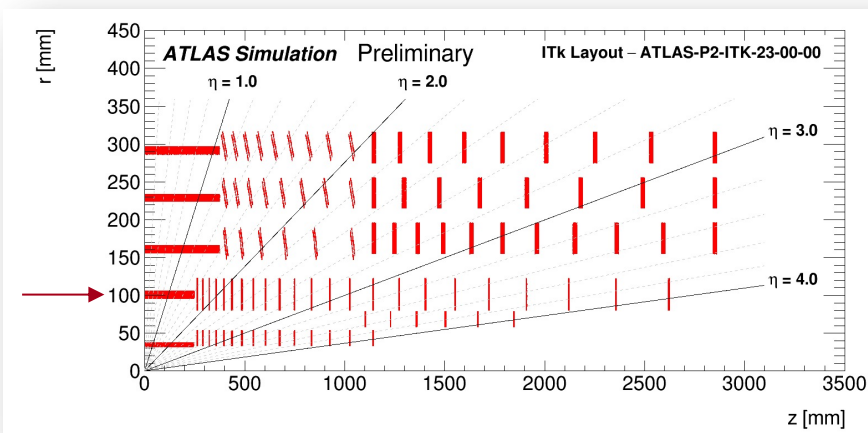
- Do vertex clustering (*z-scan*), return the average hit time and define the time-ROIs to run tracking
 - Warning: PU tracks accidentally included in the *z*-clusters lead to non-Gaussian tails in the time distribution.
 - Pre-selecting high p_T tracks would allow us to remove those contributions if we find that this is an issue.



- **Seed finding**
- **Track-to-Seed assignment**
- **Fitting**
- **Acceptance/Rejection**

Tracking in time-ROI: outline of the workflow

- First step:
 - On CERN:
 - get us some HowTo instructions to be able to run ACTS with the Itk geometry (focus on Pixel layers at first)
 - We ran the machinery with a generic detector geometry, see back-up slides
- Once we are able to run the full chain with the Itk geometry:
 - On SLAC:
 - Change the timing resolution, layer position, number of layers, etc
 - For the initial 1-layer study we will consider the **second innermost pixel layer**.



ACTS Integration

- As said, ATLAS is planning to switch to ACTS components for tracking in HL-LHC
 - Integration in Athena has started (ACTS Vertexing for RUN3 already!)
 - Possibility for long term involvement: once ACTS will be fully integrated, the impact of these studies can be thoroughly tested in the entire data processing chain

NEW IMPROVED US Atlas LXR Cross Reference of Nightly Git Clones
 head/athena/Tracking/

Last indexation completed on 2020-09-22 20:09:45 UTC

Name	Size	Date (UTC)	Last indexed	Description
Parent directory	-	2020-09-22 19:24:17		
Acts/	-	2020-09-22 19:24:16		
THAlgorithms/	-	2020-09-22 19:24:16		
THAlignment/	-	2020-09-22 19:24:16		
THConditions/	-	2020-09-22 19:24:16		
THConfig/	-	2020-09-22 19:24:16		
THDetDescr/	-	2020-09-22 19:24:16		
THDigitization/	-	2020-09-22 19:24:16		
THEvent/	-	2020-09-22 19:24:16		
THEventCnv/	-	2020-09-22 19:24:16		
THExtrapolation/	-	2020-09-22 19:24:16		
THFilter/	-	2020-09-22 19:24:16		
THG4Components/	-	2020-09-22 19:24:16		
THTools/	-	2020-09-22 19:24:16		
THTruthTracks/	-	2020-09-22 19:24:16		
THUtilityPackages/	-	2020-09-22 19:24:16		
THValidation/	-	2020-09-22 19:24:16		
THVertexFinder/	-	2020-09-22 19:24:16		

NEW IMPROVED US Atlas LXR Cross Reference of Nightly Git Clones
 head/athena/Tracking/Acts/

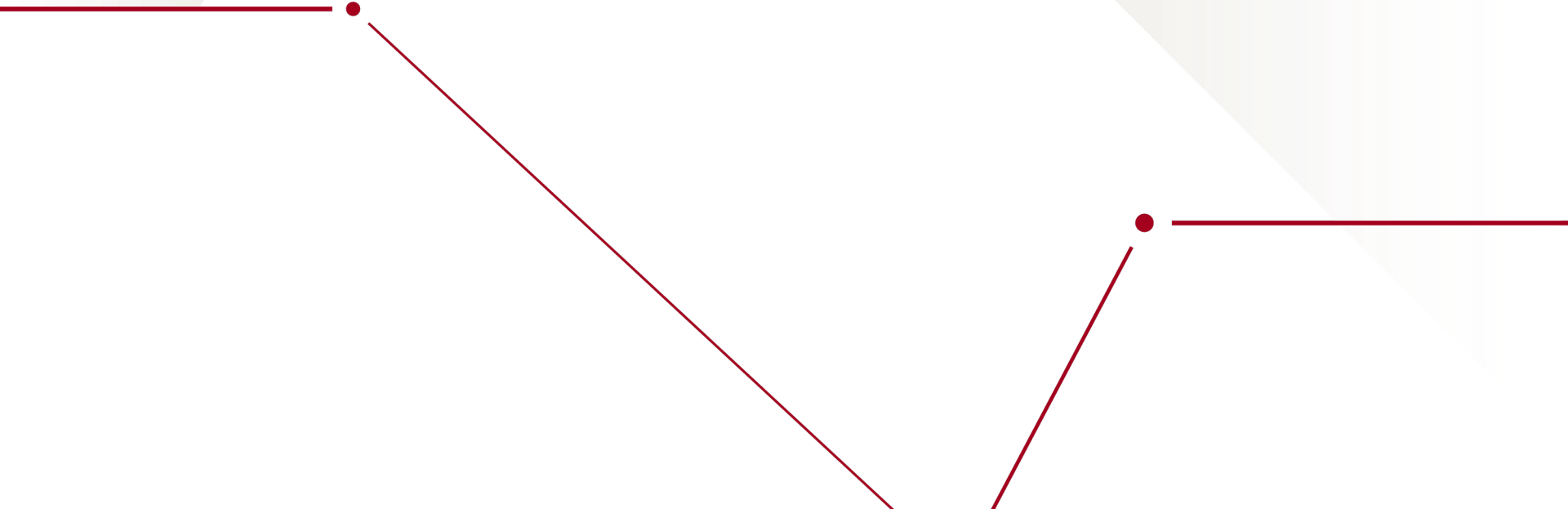
Last indexation completed on 2020-09-22 20:09:45 UTC

Name	Size	Date (UTC)	Last indexed	Description
Parent directory	-	2020-09-22 19:24:16		
ActsGeometry/	-	2020-09-22 19:24:16		
ActsGeometryInterfaces/	-	2020-09-22 19:24:16		
ActsInterop/	-	2020-09-22 19:24:16		
ActsPrivtxFinder/	-	2020-09-22 19:24:16		

Wrap up and open questions

- **From a track-reconstruction perspective, using ACTS to reconstruct tracks with a parametrization that accounts for time is a very promising approach**
 - One might easily hack other existing platforms, but then those developments would only serve this project and will not be used in the future
 - Other experiments will use ACTS – in principle we can change the geometry and see the impact of our studies in other detectors!
- We have defined a plan to check the impact of **tracking in time-ROIs**
- The result of this first step will indicate what direction to take
- If the ultimate goal is to get to a full **physics prospect study** (for instance, what is the impact on btagging and thus on certain physics topologies?) then the question becomes more complex (and is constrained by the available software infrastructure)... **We would need to use ACTS within a framework that allows to build other objects**
 - Full ACTS integration in Athena may not arrive on time for a Snowmass paper due next Summer
 - We can have multiple milestones: start with the basics (i.e. tracking and vertexing, and in the meanwhile see if we can build on the existing knowledge used for the HGTD TDR to perform higher level studies)

Extra Slides



Instructions

As a reference, look at

https://acts.readthedocs.io/en/latest/getting_started.html

And <https://indico.cern.ch/event/917970/overview> (but some of the options used in the April workshop are already superseded, so you can use the instructions below as a reference)

On lxplus

```
git clone https://github.com/acts-project/acts source
cd source/
source CI/setup_lcg96.sh
cmake -B build -S . -DACTS_BUILD_EVERYTHING=on
cmake --build build
cd build/bin
```

Instructions - Geometry

```
./ActsExampleGeometryGeneric -help (for curiosity)
```

```
./ActsExampleGeometryGeneric -n1 --output-obj true --geo-detector-volume
```

```
BeamPipe Pixel SStrip LStrip
```

(This uses a generic geometry created for the ML challenge and it will create .obj files that you can visualize in a 3D visualizer, like MeshLab)

Instructions - Fatras

`./ActsExampleFatrasGeneric --help (for curiosity)`

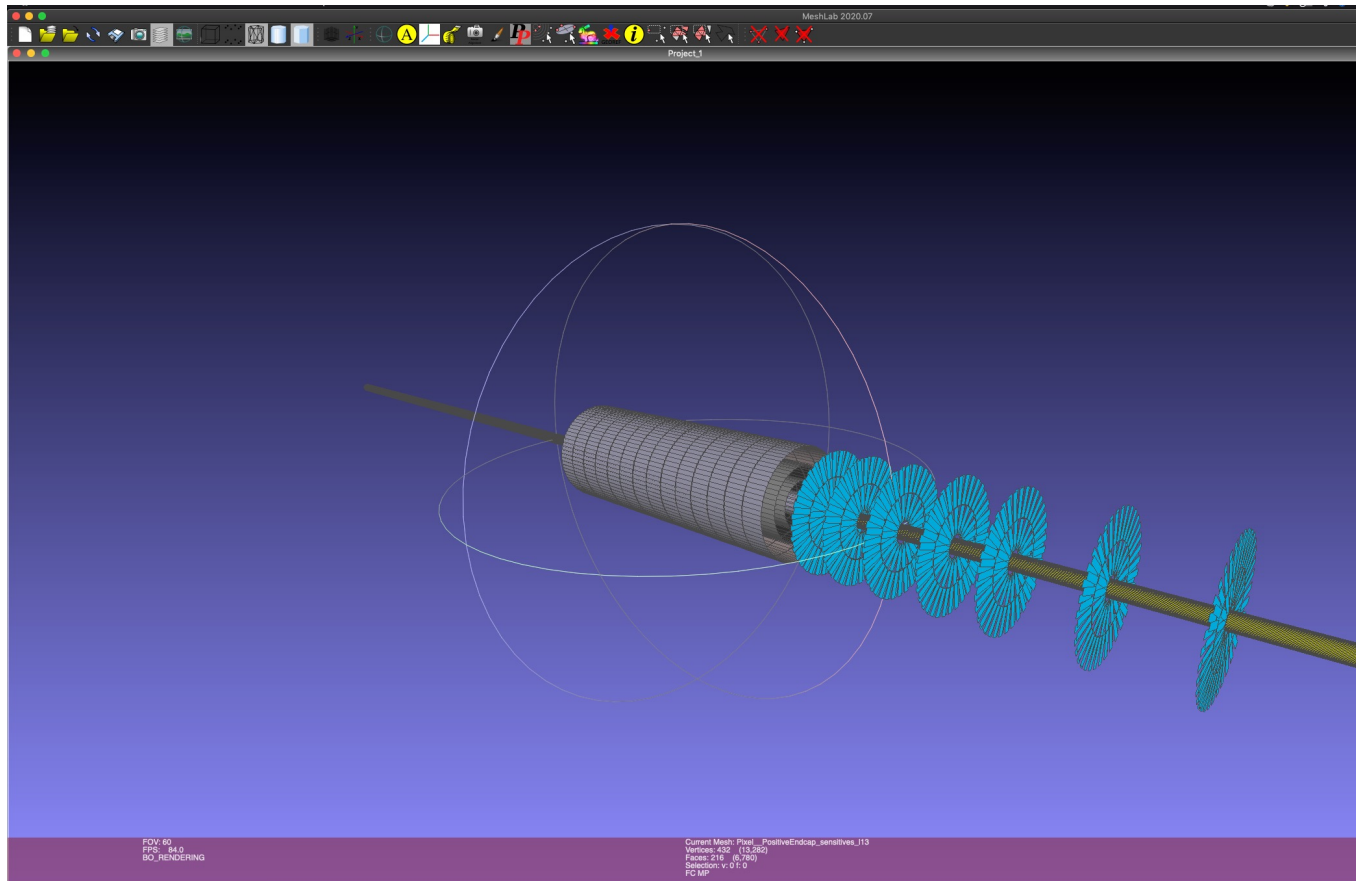
`./ActsExampleFatrasGeneric -n10 --bf-values 0 0 2.5 --output-root true --output-csv true`

```
[vcairo@lxplus768 bin]$ ./ActsExampleFatrasGeneric -n10 --bf-values 0 0 2.5 --output-root true --output-csv true
PYTHIA Warning in PhaseSpace2to2tauyz::trialKin: maximum for cross section violated
PYTHIA Warning in MultipartonInteractions::init: maximum increased by factor 4.172
03:38:36 Sequencer INFO Added reader 'EventGenerator'
03:38:36 Sequencer INFO Added algorithm 'FlattenEvent'
03:38:36 Sequencer INFO Added writer 'CsvParticleWriter'
03:38:38 Sequencer INFO Added writer 'RootParticleWriter'
- BField (scalar to/in) Tesla set to: 1
03:38:38 Sequencer INFO Added algorithm 'ParticleSelector'
03:38:38 Sequencer INFO Added algorithm 'FlattenEvent'
03:38:38 Sequencer INFO Added algorithm 'FatrasAlgorithm'
03:38:38 Sequencer INFO Added writer 'CsvParticleWriter'
03:38:38 Sequencer INFO Added writer 'CsvParticleWriter'
03:38:38 Sequencer INFO Added writer 'RootParticleWriter'
03:38:39 Sequencer INFO Added writer 'RootParticleWriter'
03:38:42 Sequencer INFO Added writer 'RootSimHitWriter'
03:38:43 Sequencer INFO Added algorithm 'DigitizationAlgorithm'
03:38:43 Sequencer INFO Added writer 'CsvPlanarClusterWriter'
03:38:45 Sequencer INFO Added writer 'RootPlanarClusterWriter'
03:38:45 Sequencer INFO Processing events [0, 10)
03:38:45 Sequencer INFO Starting event loop with 10 threads
03:38:45 Sequencer INFO 0 services
03:38:45 Sequencer INFO 0 context decorators
03:38:45 Sequencer INFO 1 readers
03:38:45 Sequencer INFO 5 algorithms
03:38:45 Sequencer INFO 9 writers
PYTHIA Warning in SimpleSpaceShower::pT2nextQCD: weight above unity
PYTHIA Error in StringFragmentation::fragment: stuck in joining
PYTHIA Error in Pythia::next: hadronLevel failed; try again
PYTHIA Warning in StringFragmentation::fragmentToJunction: bad convergence junction rest frame
03:39:23 Sequencer INFO finished event 3
03:39:25 Sequencer INFO finished event 2
03:39:31 Sequencer INFO finished event 4
03:39:33 Sequencer INFO finished event 5
03:39:33 Sequencer INFO finished event 8
03:39:33 Sequencer INFO finished event 1
03:39:36 Sequencer INFO finished event 0
03:39:36 Sequencer INFO finished event 7
03:39:37 Sequencer INFO finished event 9
03:39:37 Sequencer INFO finished event 6
03:39:37 RootParticle INFO Wrote particles to tree 'particles' in 'particles.root'
03:39:37 RootParticle INFO Wrote particles to tree 'particles' in '/eos/home-v/vcairo/ACTS/source/build/bin/particles_initial.root'
03:39:37 RootParticle INFO Wrote particles to tree 'particles' in '/eos/home-v/vcairo/ACTS/source/build/bin/particles_final.root'
03:39:39 RootPlanarCL INFO Wrote particles to tree 'clusters' in 'clusters.root'
03:39:39 Sequencer INFO Processed 10 events in 54.838948 s (wall clock)
03:39:39 Sequencer INFO Average time per event: 48.100153 s/event
```

Just a quick look to understand better...

- Many examples are currently provided. Tried to run a few on **geometry building** and Fatras (see back-up for instructions if you want to try it yourself!)

MeshLab visualizer



Just a quick look to understand better...

- Many examples are currently provided. Tried to run a few on geometry building and **Fatras** (see back-up for instructions if you want to try it yourself)

Generation and simulation are done in a single step for now

Some key options for running Fatras (`./ActsExampleFatrasGeneric --help`):

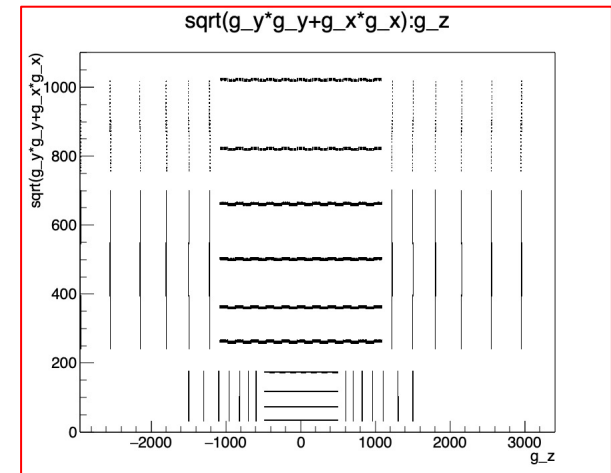
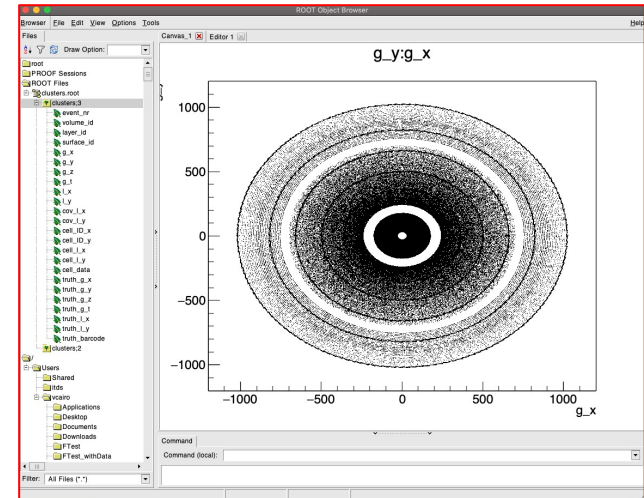
- `--evg-cms-energy arg (=14000)` Center-of-mass energy collision in GeV
- `--evg-beam0 arg (=2212)` PDG number of the first beam particle
- `--evg-beam1 arg (=2212)` PDG number of the second beam particle

- `--evg-input-type arg (=pythia8)` Type of evgen input 'gun', 'pythia8'
- `--evg-hard-process arg (=HardQCD:all = on)` Pythia8 process string for the hard scatter
- `--evg-pileup-process arg (=SoftQCD:all = on)` Pythia8 process string for the pile-up
- `--evg-pileup arg (=200)` Number of instantaneous pile-up events

- `--pg-pdg arg (=13)` PDG number of the particle, will be adjusted for charge flip.
- `--bf-values arg (=0 0 0)` In case no magnetic field map is handed over. A constant magnetic field will be created automatically. The values can be set with this options. Please hand over the coordinates in cartesian coordinates: {Bx,By,Bz} in Tesla.
- `--bf-map arg` Set this string to point to the bfield source file. That can either be a '.txt', a '.csv' or a '.root' file. Omit for a constant magnetic field.
- `--bf-name arg (=bField)` In case your field map file is given in root format, please specify the name of the TTree.

`./ActsExampleFatrasGeneric -n10 --bf-values 0 0 2.5 --output-root true --output-csv true`
(still uses the generic geometry from the previous page)

clusters.root hits.root particles_final.root particles_initial.root particles.root



From the tutorial

https://indico.cern.ch/event/917970/contributions/3862003/attachments/2045143/3426537/zoom_0.mp4

Truth content

```
In [5]: h = pd.read_csv('/home/msmk/code/acts/sim-gun-test/event000000000-truth.csv')
        h.head()
```

Out[5]:

	hit_id	particle_id	geometry_id	tx	ty	tz	tt	tpx	tpy	tpz	te	deltapx	deltapy
0	0	4503599644147712	504403295704419033	17.785538	82.659142	-1502.5	5.021653	0.045841	0.208956	-3.811316	3.818778	-0.000439	0.000588
1	1	4503599644147712	504403433143402505	15.380033	71.694099	-1302.5	4.353219	0.045776	0.208962	-3.811317	3.818778	0.000064	-0.000005
2	2	4503599644147712	504403570582355977	12.977911	60.728767	-1102.5	3.684784	0.045843	0.208788	-3.811326	3.818778	-0.000067	0.000173
3	3	4503599644147712	504403708021309449	11.293963	53.059425	-962.5	3.216882	0.046107	0.209114	-3.811305	3.818778	-0.000263	-0.000326
4	4	4503599644147712	504403845460262921	9.600341	45.378079	-822.5	2.748976	0.045741	0.209426	-3.811292	3.818778	0.000366	-0.000312

FATRAS

- A few words about FATRAS

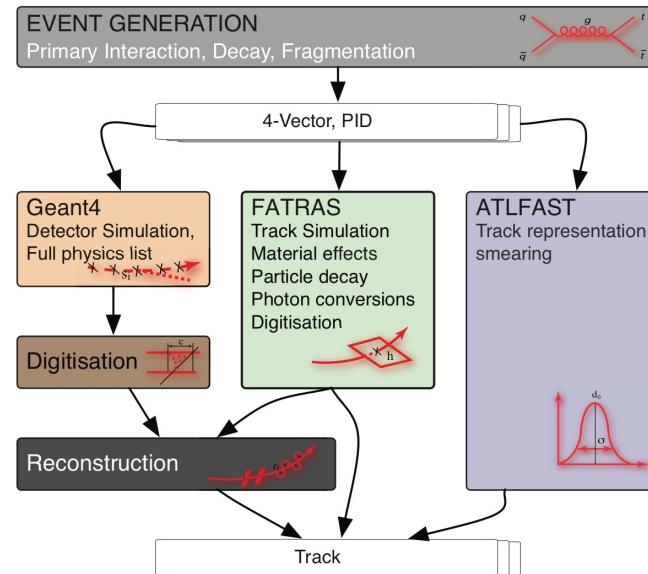
README.md

FASt TRAck Simulation package

This packages provides tools to run a fast track simulation on top of the core library. The fast track simulation (Fatras) uses the actor plug-in mechanism of the `Acts::Propagator` and its predictive navigation to simulate particle trajectories through the tracking detector.

As a fast track simulation, it uses the surface-based reconstruction geometry description, i.e. `Acts::TrackingGeometry`, as a simulation geometry instead of a detailed volumetric description. Interactions and material effects are simulated using parametrized models:

- Multiple Coulomb scattering is simulated by Gaussian (mixture) approximations.
- Ionisation loss is simulated using the Bethe-Bloch formalism.
- Radiation loss is simulated using the Bethe-Heitler formalism.



(a) Track simulation in ATLAS

HGTD ([link](#))

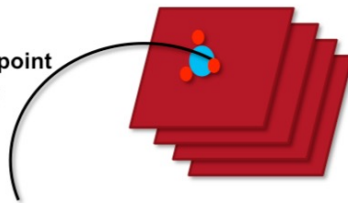
Time Association Strategy

SLAC

Starting point: Extrapolate from **last hit on track** to HGTD layer 0

Then, for each layer:

1. Look for **surfaces** near the **extrapolated crossing point**
 - Pick up all surfaces in a $\sim 5 \times 5 \text{cm}$ region around the crossing
 - Currently optimised for efficiency rather than speed
2. Look for the **cluster best matching the track**
 - Candidates: All clusters on surfaces selected in 1.)
 - For each cluster: Use KalmanUpdater to attempt to add measurement to track
 - Keep the one cluster with the best incremental χ^2
3. Keep cluster if **satisfactory fit outcome**
 - Cutting at $\chi^2/n.d.f < 5$
4. In case of **successful** extension: Use **updated** track parameters from Kalman forward filter for extrapolation to next layer



Result: Set of 4 outcomes (either one cluster or failure) for the 4 layers.

VMM CAIRO

3

Information Collected for Later Analysis

SLAC

Result of this procedure:

Decorated to xAOD::TrackParticles for analysis in InDetPhysValMonitoring

Stored information:

Track quality based on presence/absence of hits on last Pixel/Strip layer

True production time from truth particle's production vertex

Reco track time based on χ^2 weighted average of cluster times on track

- Also available: Version with arithmetic mean
- Also storing RMS of times of associated clusters – sensitive to PU contamination!
- See below for def of cluster time!

Information per layer:

- **Presence/absence** of associated cluster
- **Cluster time** – Defined as time measured in HGTD corrected for time of flight from track origin (0,0,z₀)
- **Incremental χ^2**
- **Location** in x,y,z,R
- **Truth classification of cluster** – see extra slides
- Information on **existence** of a cluster on the layer from the track regardless of association outcome
 - Using truth – see extra slides

VMM CAIRO

4

Snowmass Efforts

- For what concerns future detector scenarios, **a set of tutorials is being put in place by the Snowmass MC task force** to help new users ramp up (most of these tutorial still have to take place, so we will learn more in the coming weeks)

<https://snowmass21.org/montecarlo/energy>

MC/Simulation Framework Tutorial Series

We are holding a series of tutorials for a variety of proposed future colliders. The goal of these tutorials is to prepare outside collaborators to conduct physics studies for these machines.

Machine	Date	Link
ILC	Aug 28	https://indico.fnal.gov/event/45031/
CEPC	Sept 8	https://indico.fnal.gov/event/45183/
FCC-ee/hh	Sept 22-23	https://indico.cern.ch/event/945608/
LHeC/FCC-eh	Sep 25	https://indico.fnal.gov/event/45185/
Whizard for e+e-	Sept 28	https://indico.fnal.gov/event/45413/
FCC-ee/hh	Sept 29	https://indico.cern.ch/event/949950/
Muon Collider	Sep 30	https://indico.fnal.gov/event/45187/

Snowmass Efforts

- For what concerns future detector scenarios, a set of tutorials is being put in place by the Snowmass MC task force to help new users ramp up (most of these tutorial still have to take place, so we will learn more in the coming weeks)

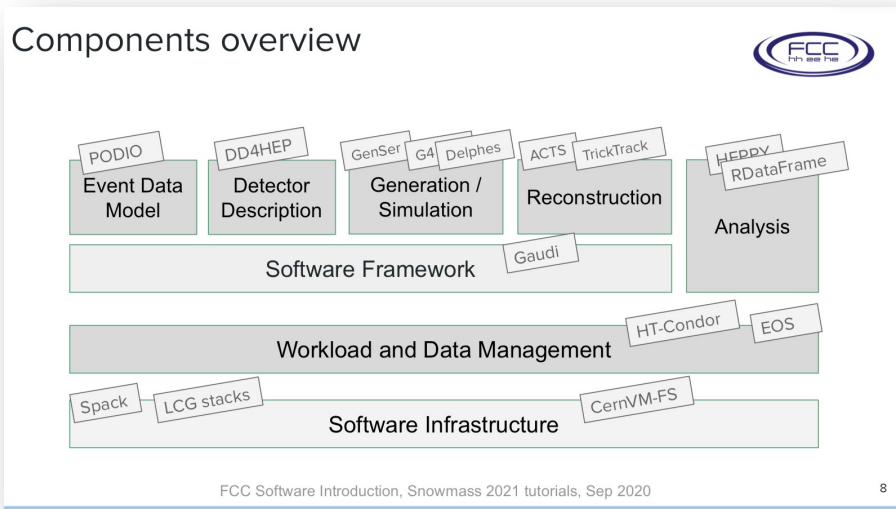
Most of these frameworks support **fast simulation**

Snowmass EF MC Task Force - Framework/MC Info				
File Modifica Visualizza Inserisci Formato Dati Strumenti Componenti aggiuntivi Guida				
Solo visualizzazione				
Which proposed collider are you responding on behalf of?				
	A	C	D	E
1	Which proposed collider are you responding on behalf of?	Have you developed a simulation framework based on GEANT?	Is this framework publicly-available? If so, where?	Have you developed a DELPHES detector card (or other form of fast and/or parameterized simulation)?
2	ILC	Yes.	https://github.com/iLCSoft	Work is currently ongoing.
3	CLIC	Yes, DD4hep+DDG4+ddsim	e.g: /cvmfs/clicdp.cern.ch/iLCSoft/builds/2020-02-07/x86_64-slc6-gcc62-opt/init_ilcsoft.sh github.com/aidasoft/dd4hep	yes, CLICdet cards Stage1/2/3, documentation in https://twiki.cern.ch/twiki/bin/view/CLIC/DelphesMadgraphForBSMReport and https://arxiv.org/abs/1909.12728 (contains concrete instructions)
4	CEPC	Yes, CEPC has the Mokka-C, a Geant4 Full simulation toolkit.	The CEPCsoft are capsuled into a package on cvmfs, the installation can be found at http://cepcsoft.ihep.ac.cn/guides/scratch/docs/cvmfs/	Yes, we have developed a Delphes card, which is also validated on the Full simulation tools. In addition, many small fast simulation tools are developed.
5	FCC-ee	Yes	Yes, the code is on github: https://github.com/HEP-FCC/ and it is installed centrally on cvmfs (same as FCC-ee). Also for the CDR, full simulation samples where produced with CMSSW.	Yes
6	FCC-hh	yes	Yes, the code is on github: https://github.com/HEP-FCC/ and it is installed centrally on cvmfs	yes
7	LHeC/FCC-eh	Yes: DD4HEP interface to GEANT	Yes: http://dd4hep.web.cern.ch/dd4hep/	Yes
8	Muon collider	Yes	yes, github	No DELPHES, no parametric simulation

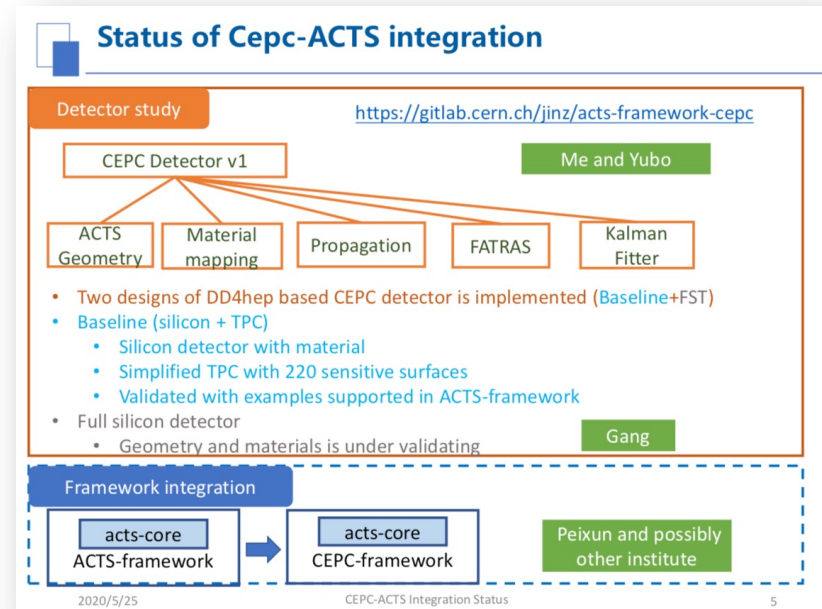
ACTS Integration

- Ongoing experiments like ATLAS are planning to switch to ACTS components for tracking in HL-LHC
- Future detectors are also integrating ACTS in their software infrastructure

FCC



CEPC



As a side note (it should not affect us, but worth mentioning) ACTS is very well supported for VMM CAIRO cylindrical geometries, but there is little support at the moment for test-beam geometries