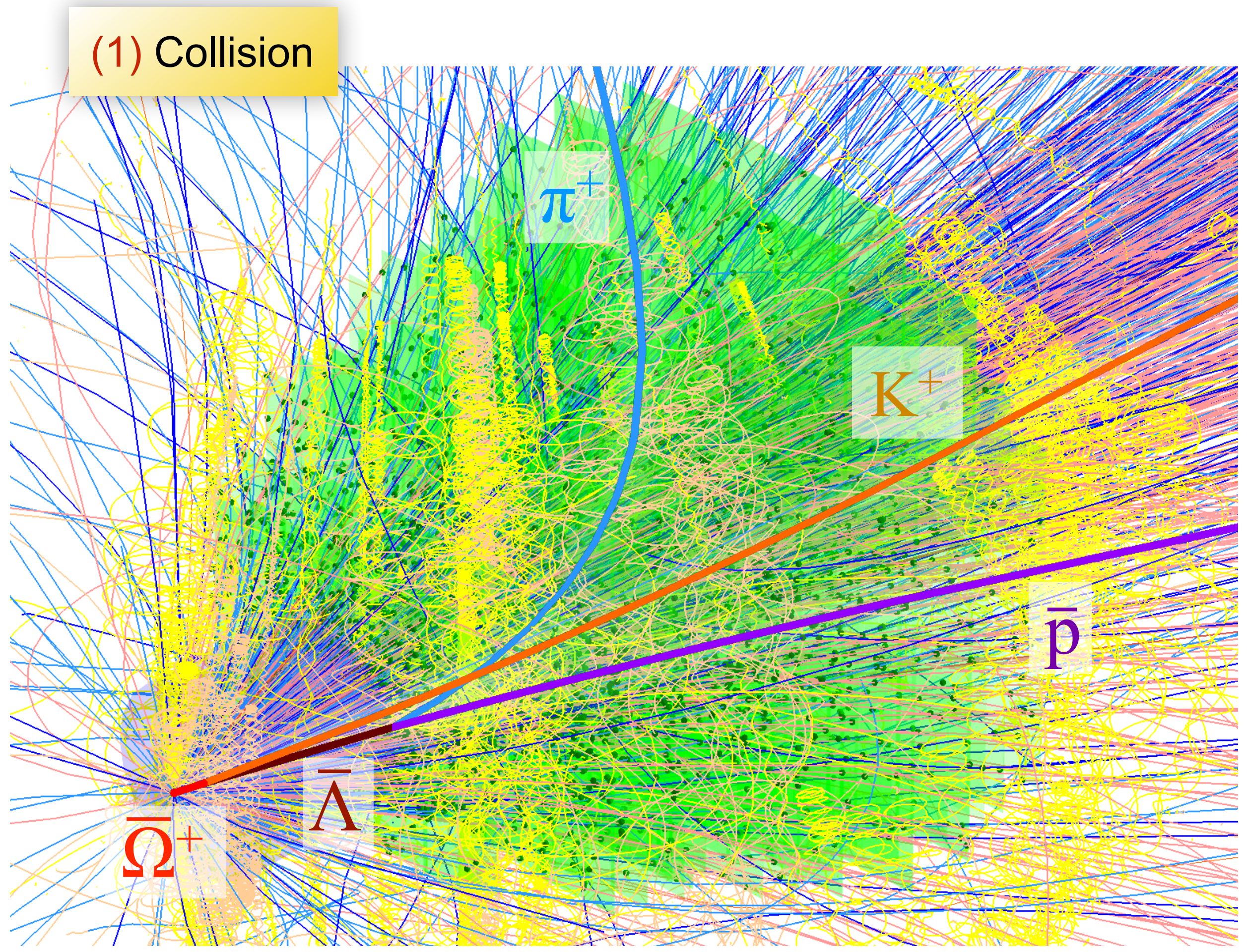
FLES - First-Level Event Selection package for fast online event reconstruction in CBM experiment at FAIR

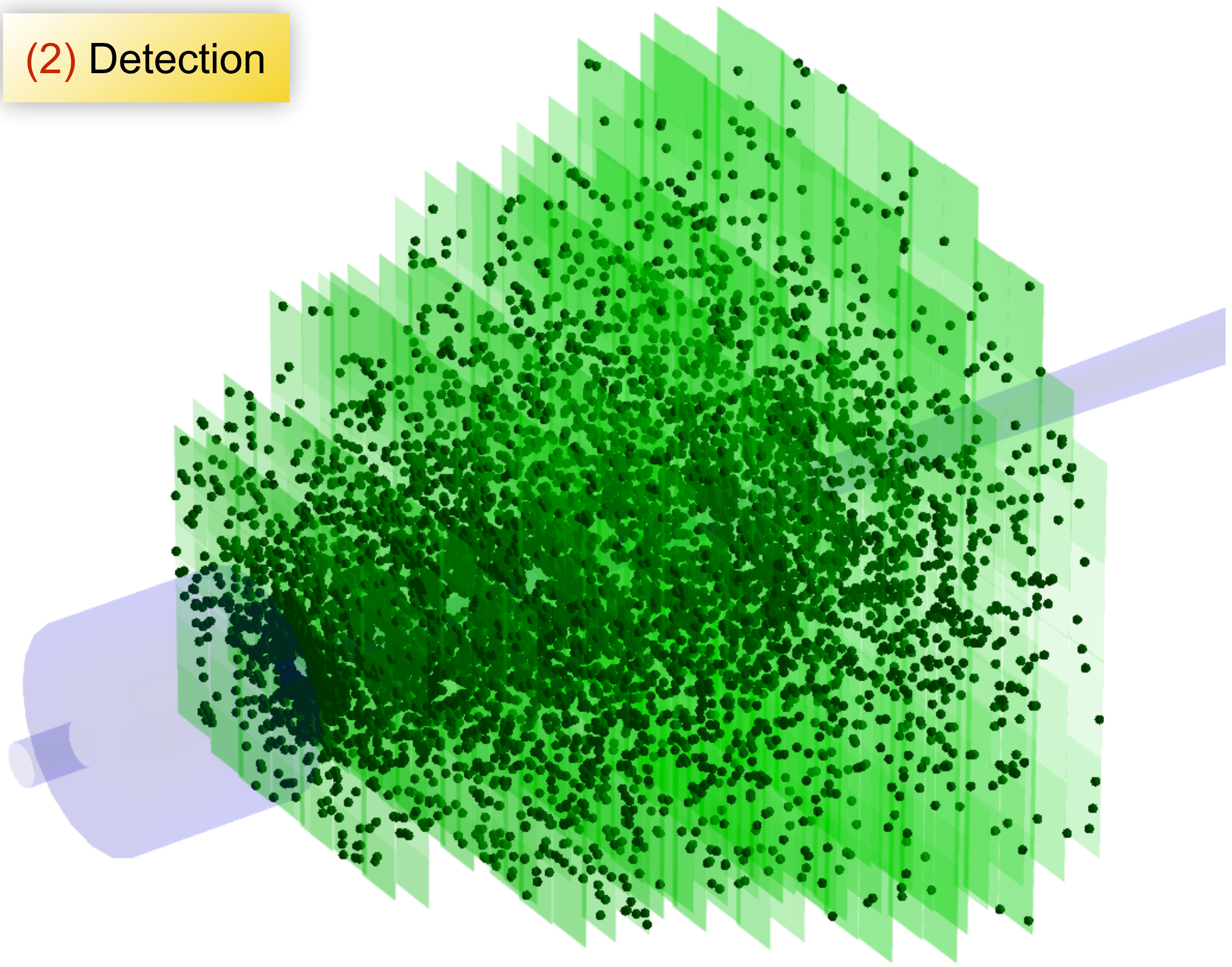
Ivan Kisel

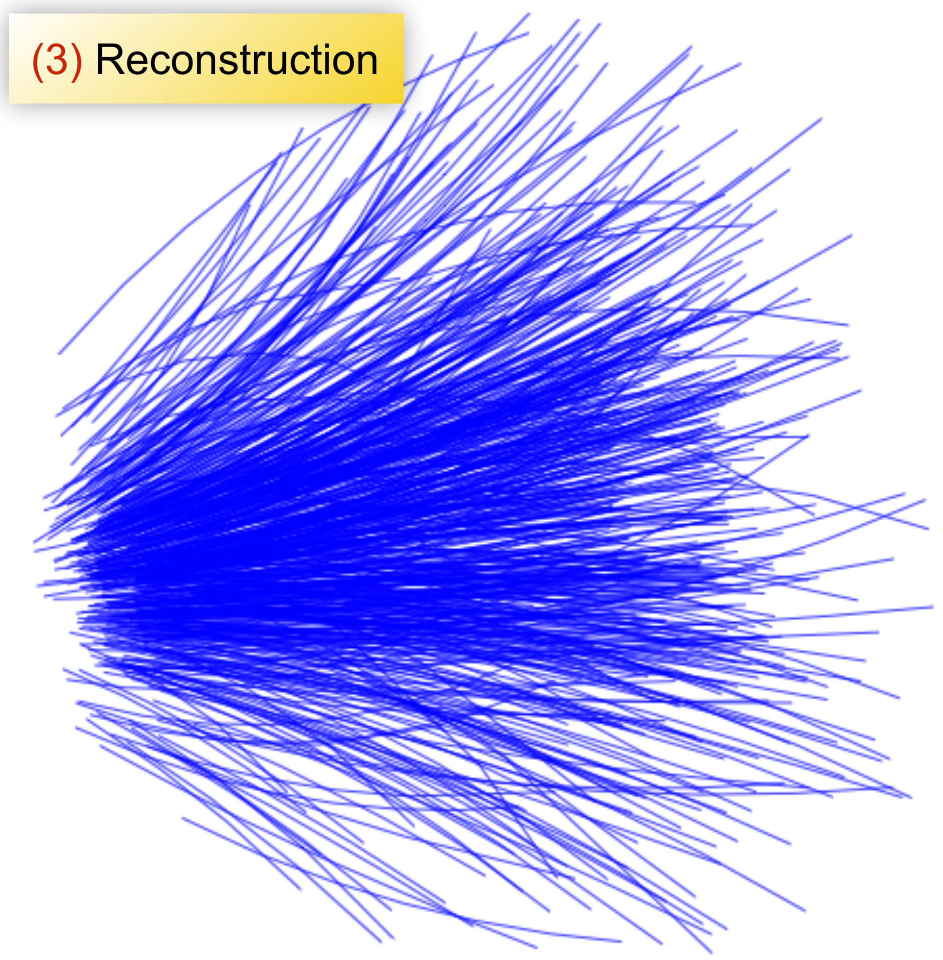
(for the CBM Collaboration)

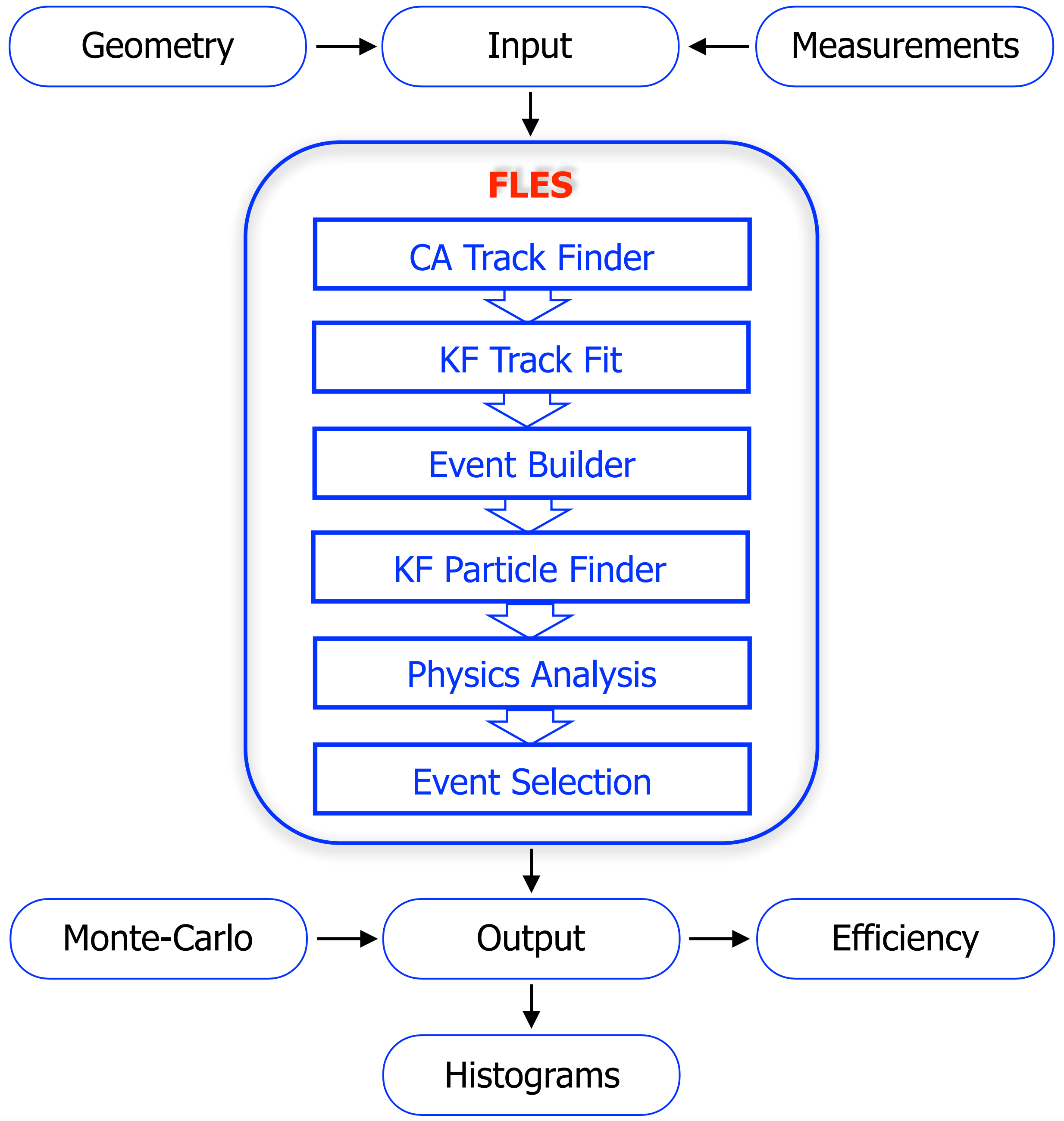
Goethe University Frankfurt am Main
FIAS Frankfurt Institute for Advanced Studies
GSI Helmholtz Center for Heavy Ion Research

Reconstruction Challenge in CBM









- CBM - Compressed Baryonic Matter
- Future fixed-target heavy-ion experiment at FAIR
- Explore the phase diagram at high net-baryon densities
- 10^7 Au+Au collisions/sec
- ~ 1000 charged particles/collision
- Non-homogeneous magnetic field
- Double-sided strip detectors
- 4D reconstruction of time slices.

Full event reconstruction and physics analysis will be done by the First-Level Event Selection (FLES) package:

- Cellular Automaton (CA) Track Finder
- Kalman Filter (KF) Track Fitter
- KF short-lived Particle Finder

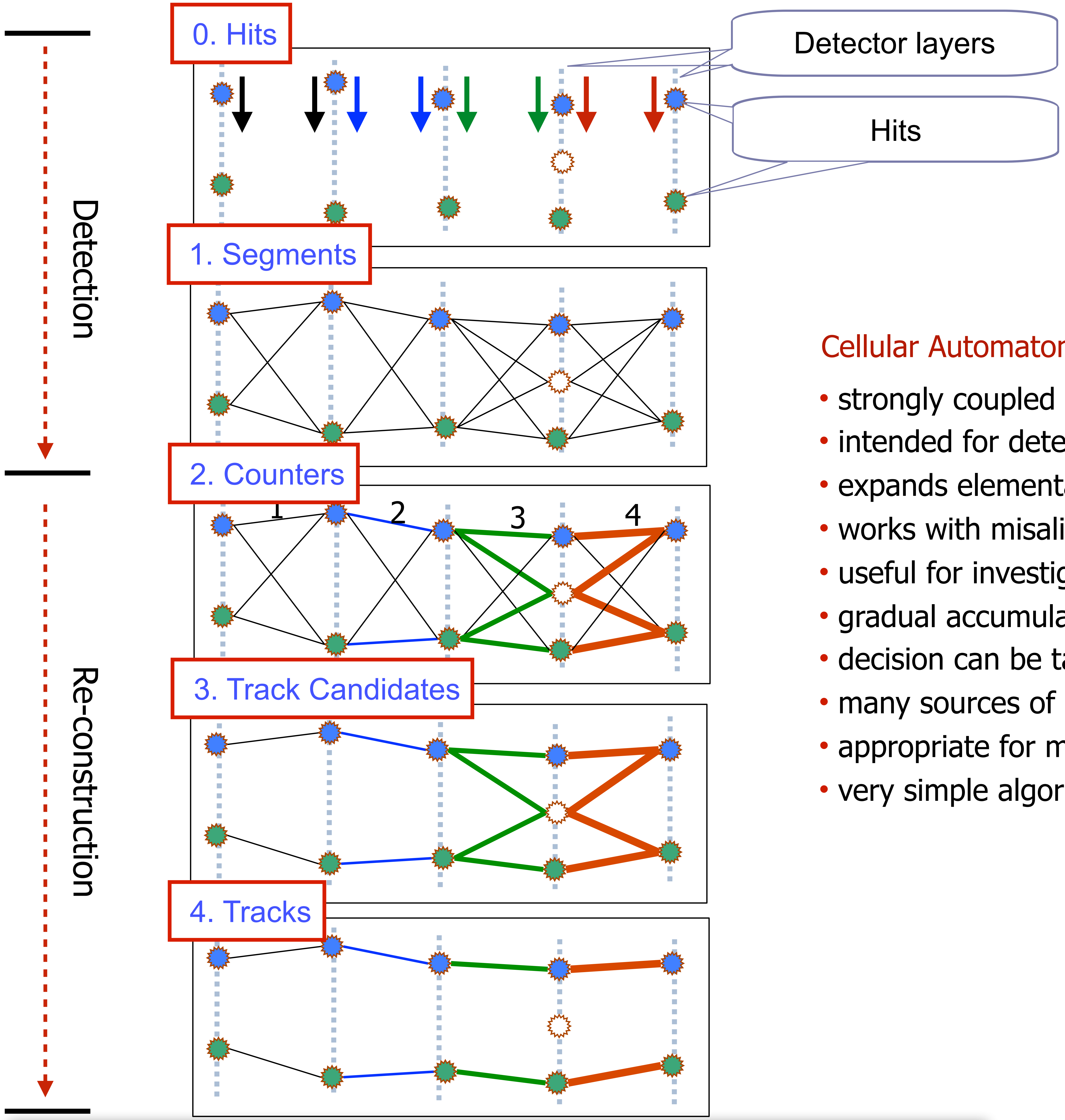
All reconstruction algorithms are vectorized and parallelized.

Event reconstruction will be done on-line and off-line using the same FLES reconstruction package

Cellular Automaton (CA) Track Finder

Cellular Automaton:

1. Build short track segments.
2. Connect them according to the track model,
   estimate their location in a track.
3. Tree structures appear,
   collect segments into track candidates.
4. Select the best track candidates.

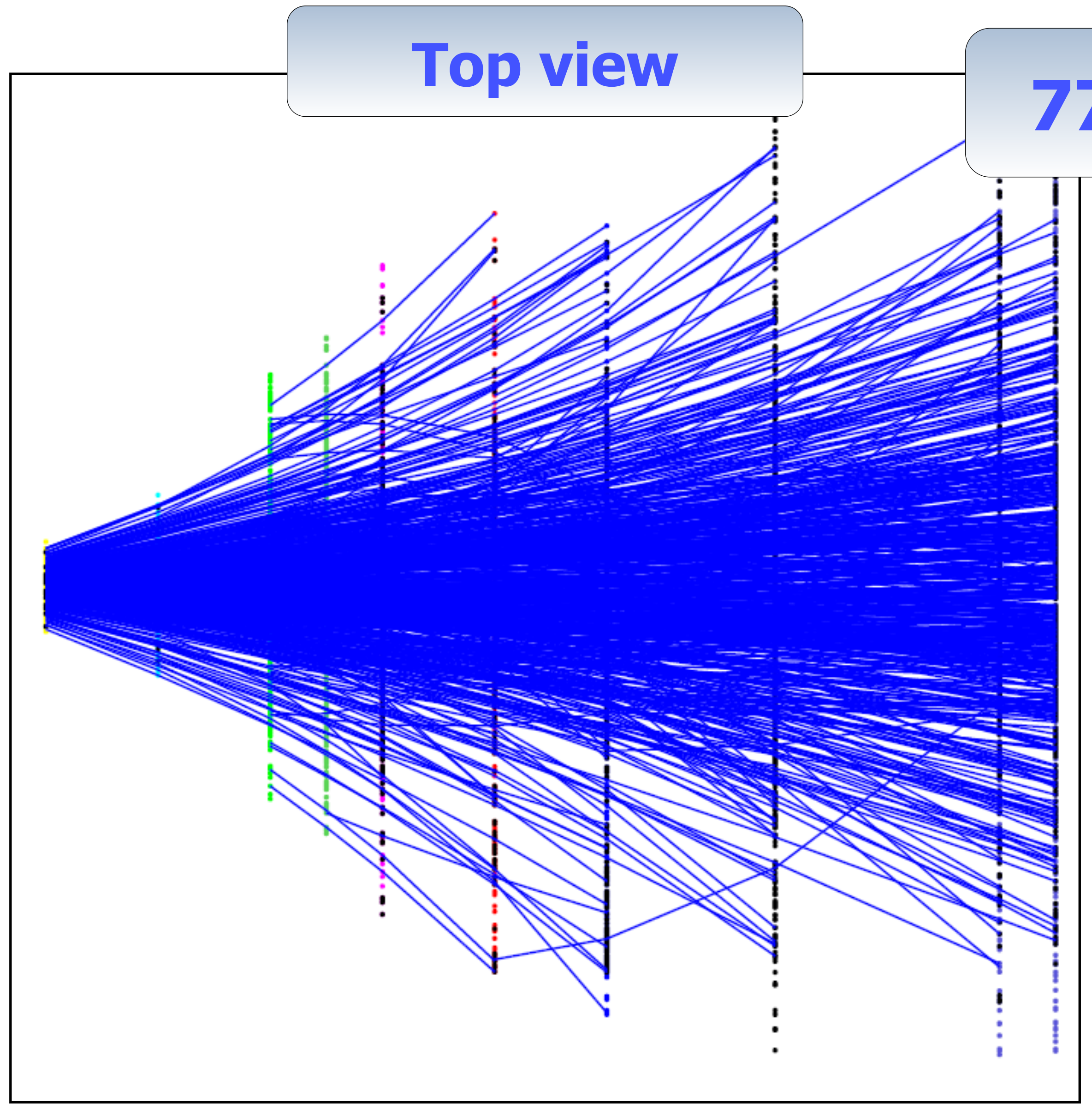


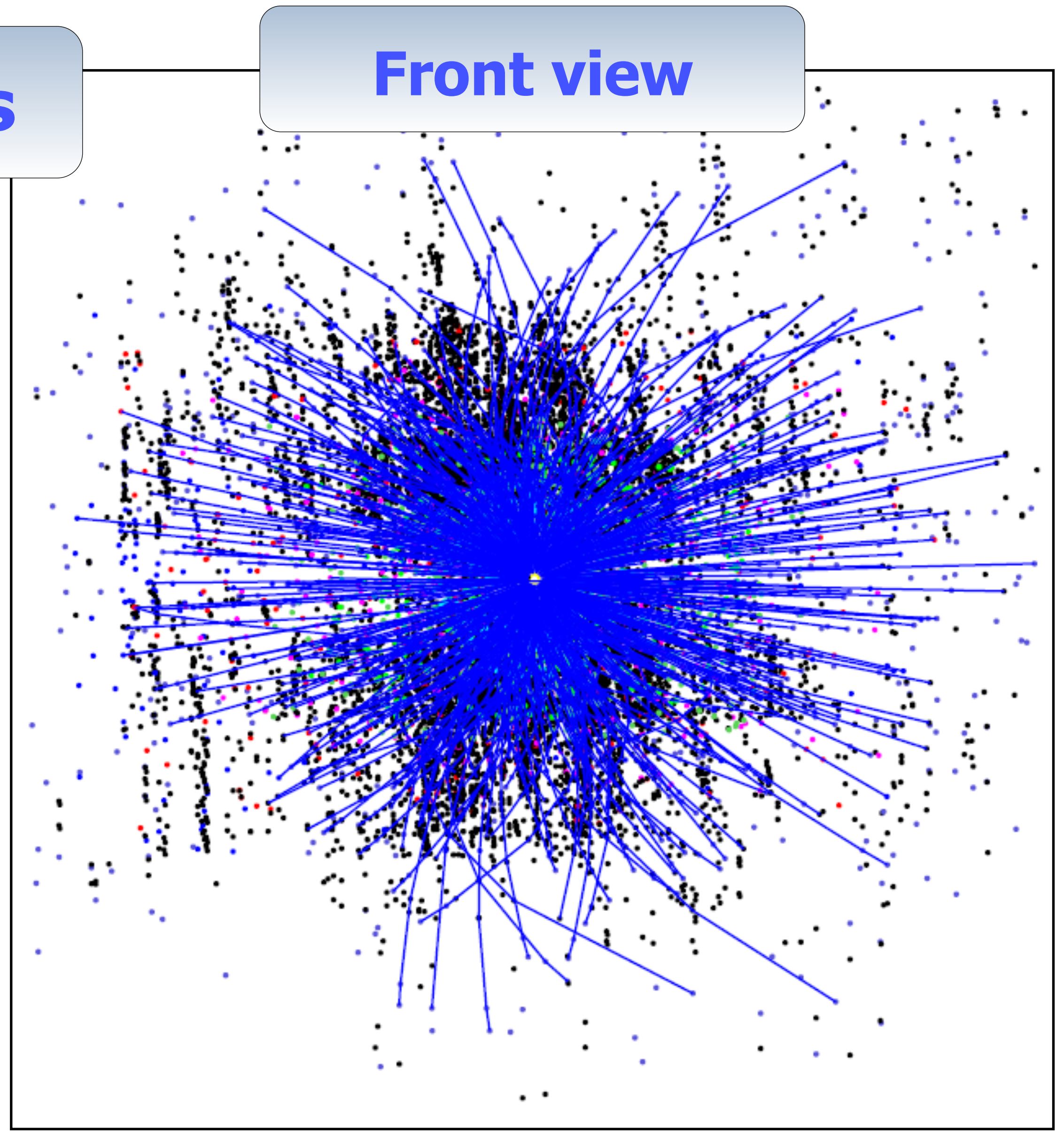
Cellular Automaton:

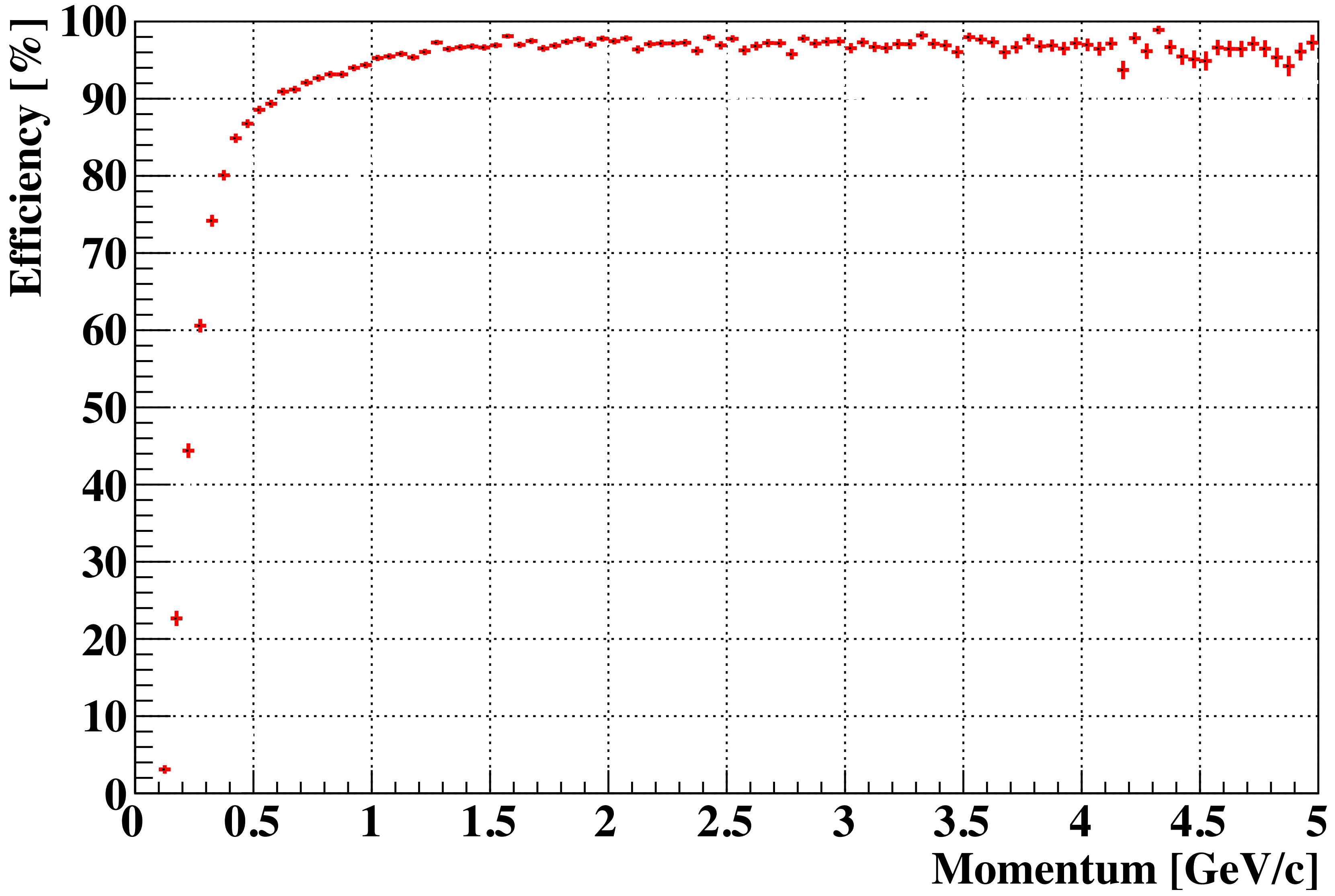
- strongly coupled to the detector system as its extension
- intended for detection of particle trajectories
- expands elementary measurements over 2-3 detectors
- works with misaligned or faulty detector systems
- useful for investigation of the detector performance
- gradual accumulation and extraction of information
- decision can be taken at any stage
- many sources of intrinsic parallelism
- appropriate for many-core CPU/GPU
- very simple algorithm

Useful for complicated event topologies with heavy combinatorics

Cellular Automaton (CA) Track Finder





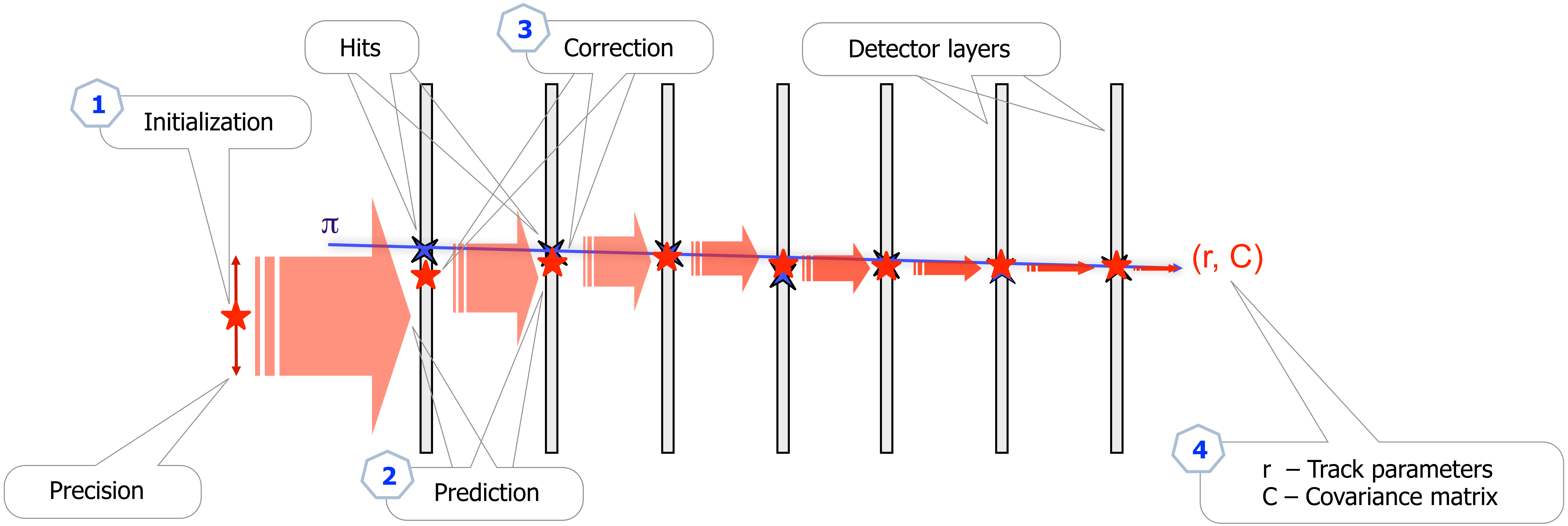


Track category	Eff, %
All tracks	90.9
Primary high-p	97.5
Primary low-p	92.6
Secondary high-p	91.1
Secondary low-p	63.8
Clone level	0.4
Ghost level	5.9
MC tracks found	134
Time, ms/ev	10

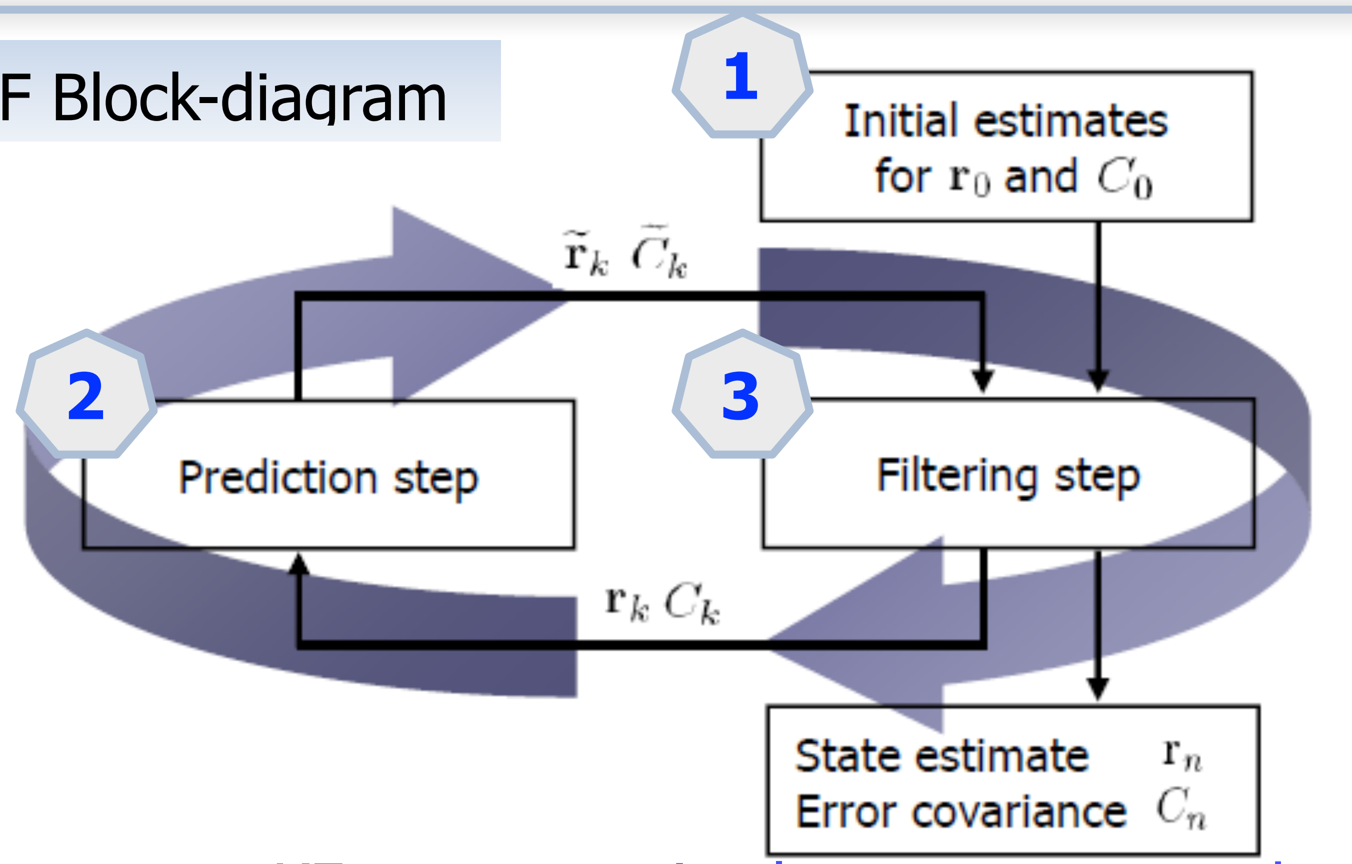
Fast and efficient track finder (100 μs/core/track)

Kalman Filter (KF) based Track Fit

Estimation of the track parameters at one or more hits along the track – Kalman Filter (KF)



KF Block-diagram



KF as a recursive least squares method

Kalman Filter:

1. Start with an arbitrary initialization.
2. Add one hit after another.
3. Improve the state vector.
4. Get the optimal parameters after the last hit.

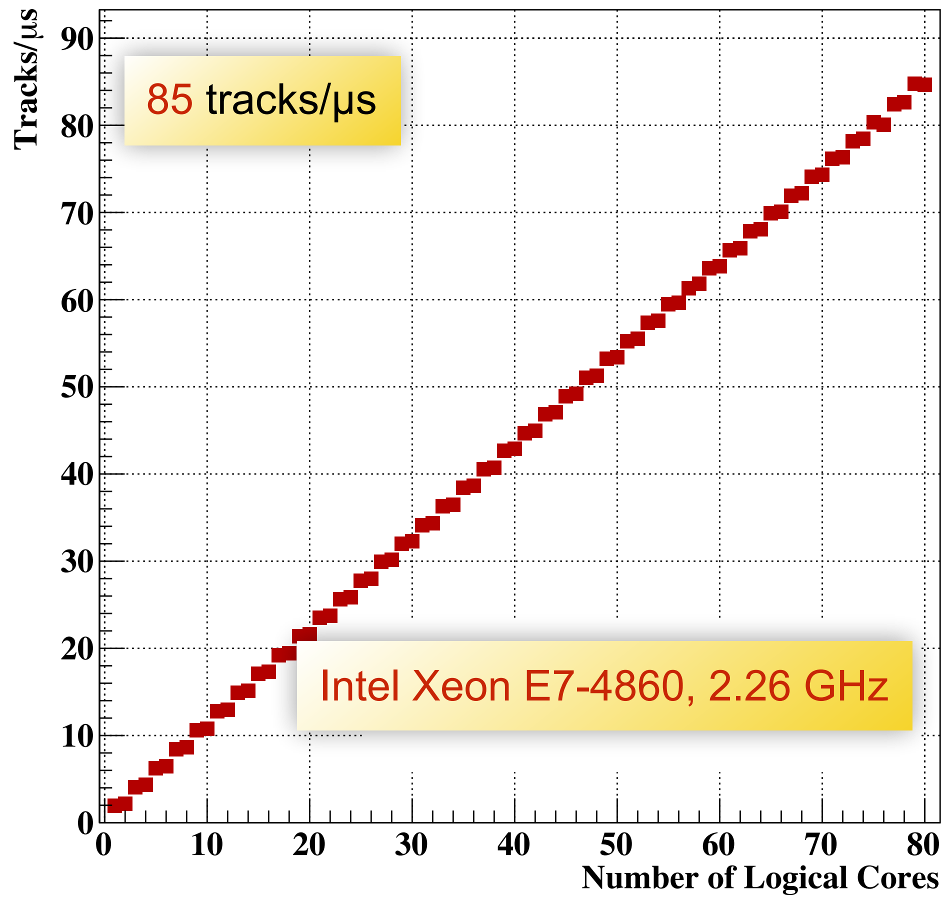
State vector

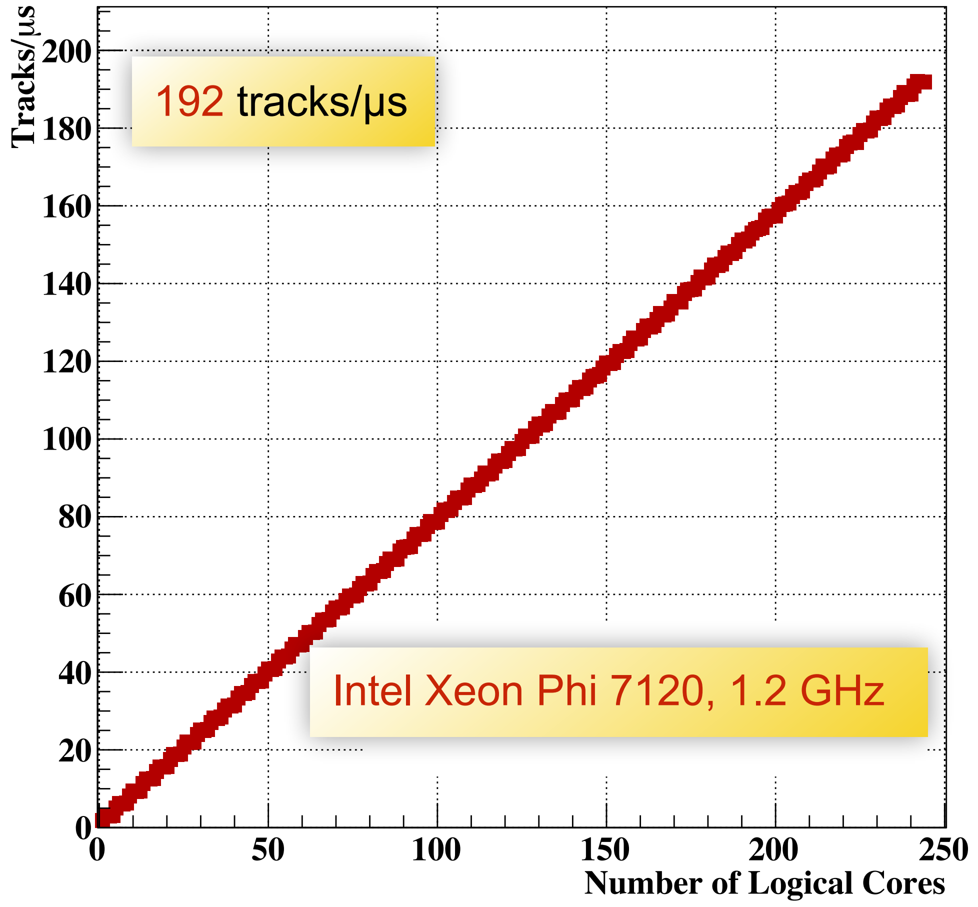
Position, direction and momentum

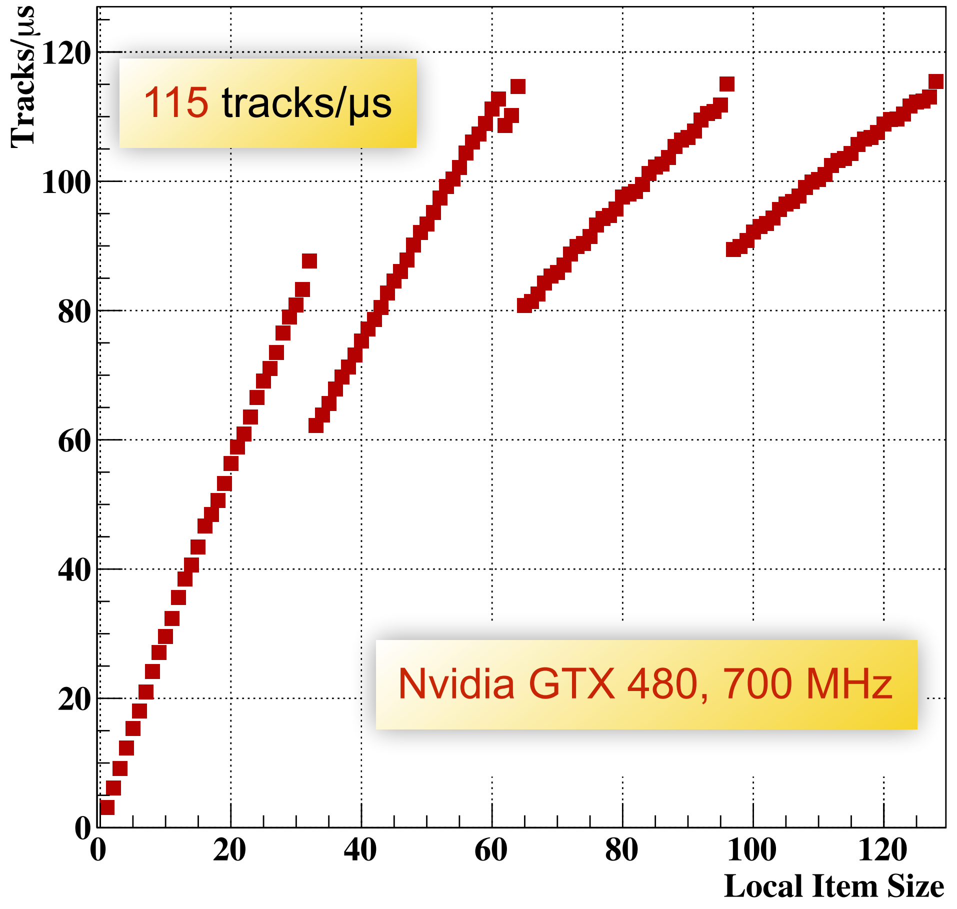
$r = \{ x, y, z, p_x, p_y, p_z \}$

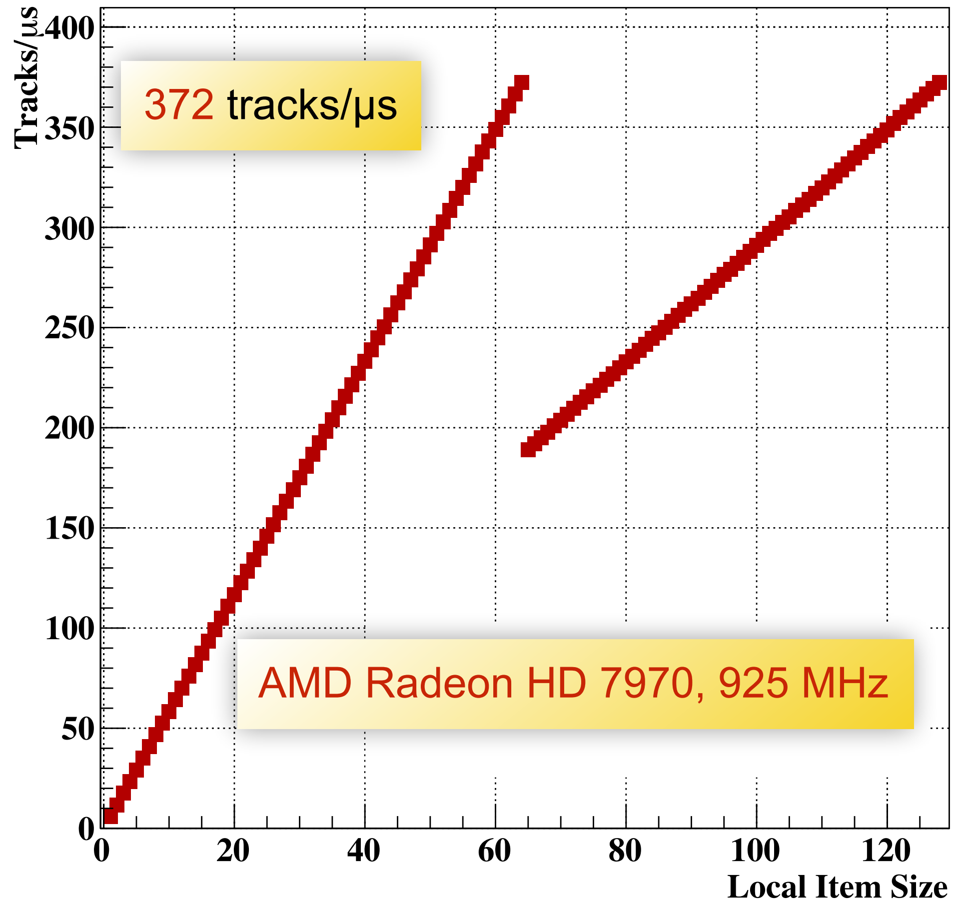
Nowadays the Kalman Filter is used in almost all HEP experiments

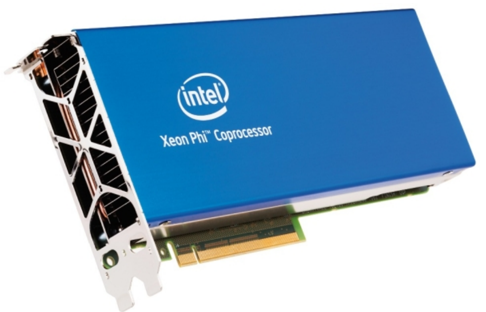
Kalman Filter (KF) Track Fit









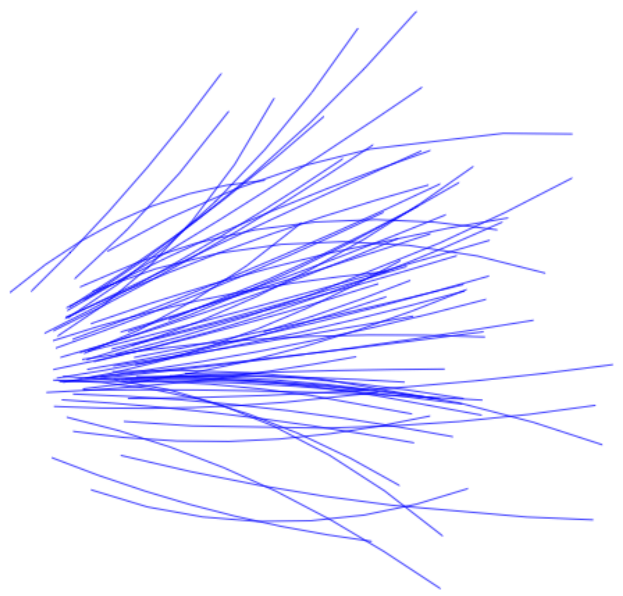


- Fast and precise estimation of the parameters of particle trajectories is the core of the reconstruction procedure in CBM.
- The KF track fitting algorithm is implemented as a single code for all CPU/GPU platforms.
- Scalability with respect to the number of logical cores in a CPU is one of the most important parameters of the algorithm.
- The scalability on the Intel Xeon Phi coprocessor is similar to the CPU, but running four threads per core instead of two.
- In case of the graphics cards the set of tasks is divided into working groups of size *local item size* and distributed among compute units (or streaming multiprocessors) and the load of each compute unit is of the particular importance.
- The track fit performance on a single node: 2*CPU+2*GPU = 10^9 tracks/s = (100 tracks/event)* 10^7 events/s = 10^7 events/s.
- A single compute node is sufficient to estimate parameters of all particles produced in CBM at the maximum 10^7 interaction rate!

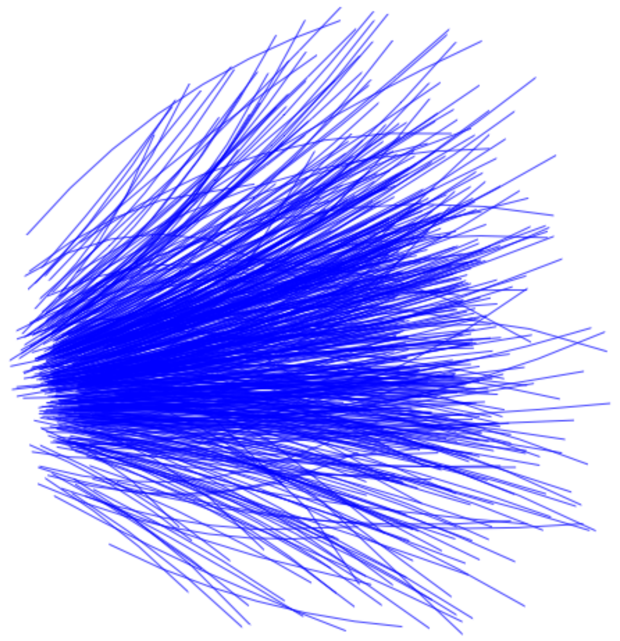
The fastest implementation of the Kalman filter in the world (0.5 μs/core/track)

CA Track Finder at High Track Multiplicity

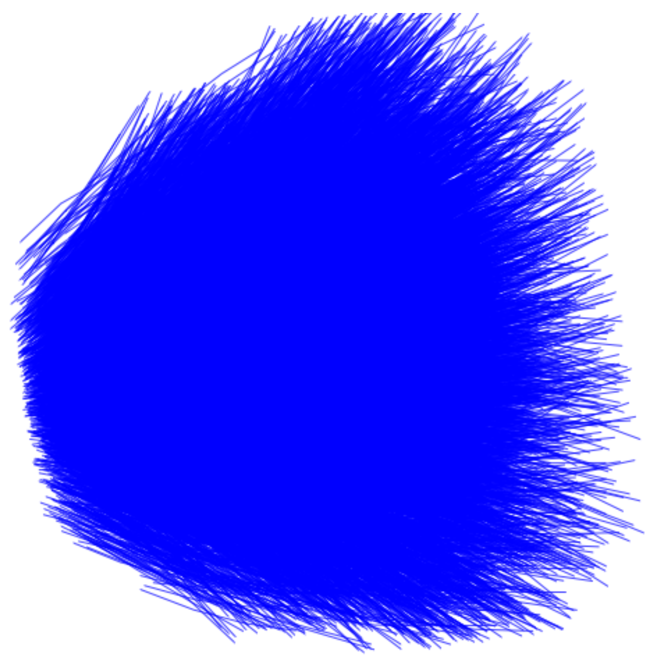
A number of minimum bias events is gathered into a group (super-event), which is then treated by the CA track finder as a single event.



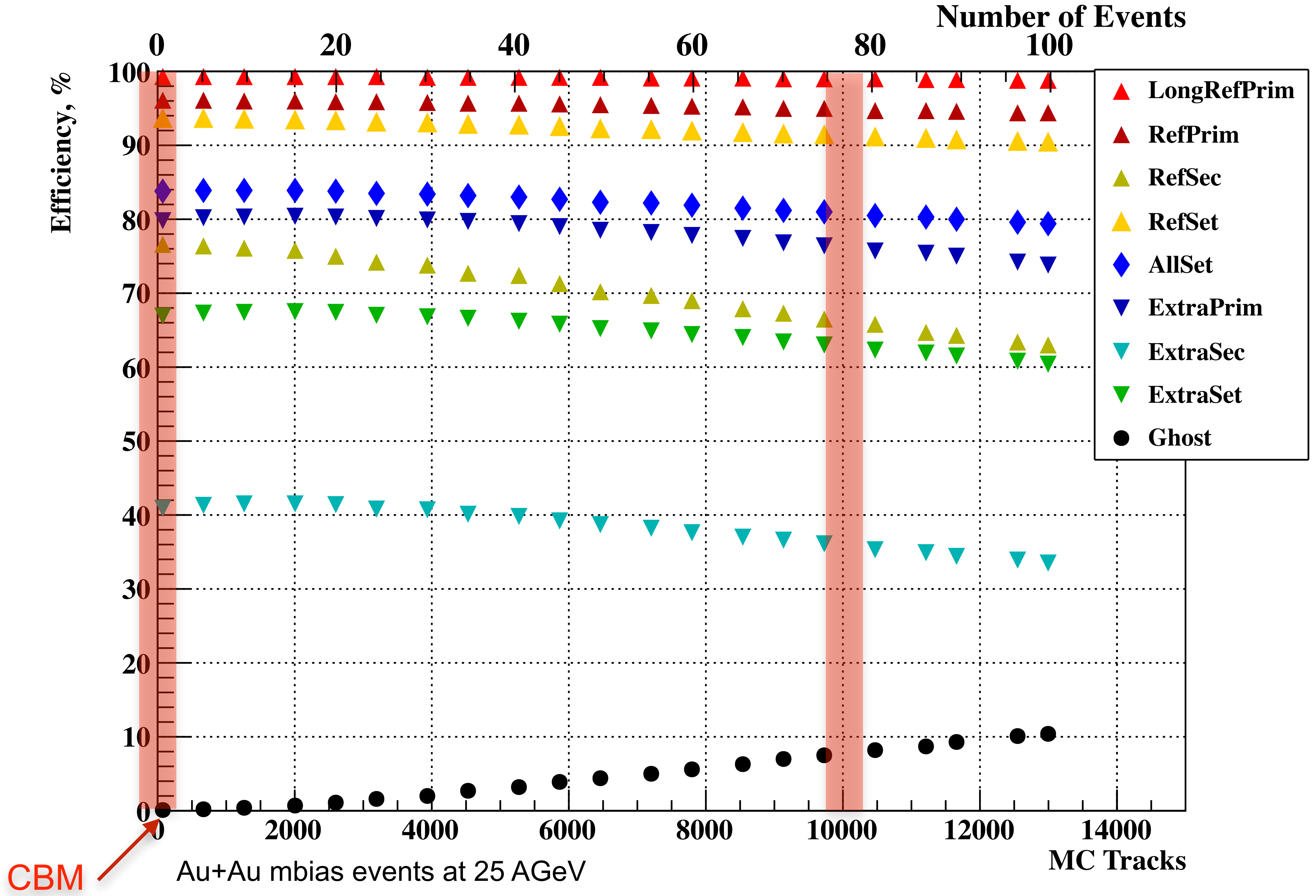
1 mbias event, $\langle N_{reco} \rangle$ = 109



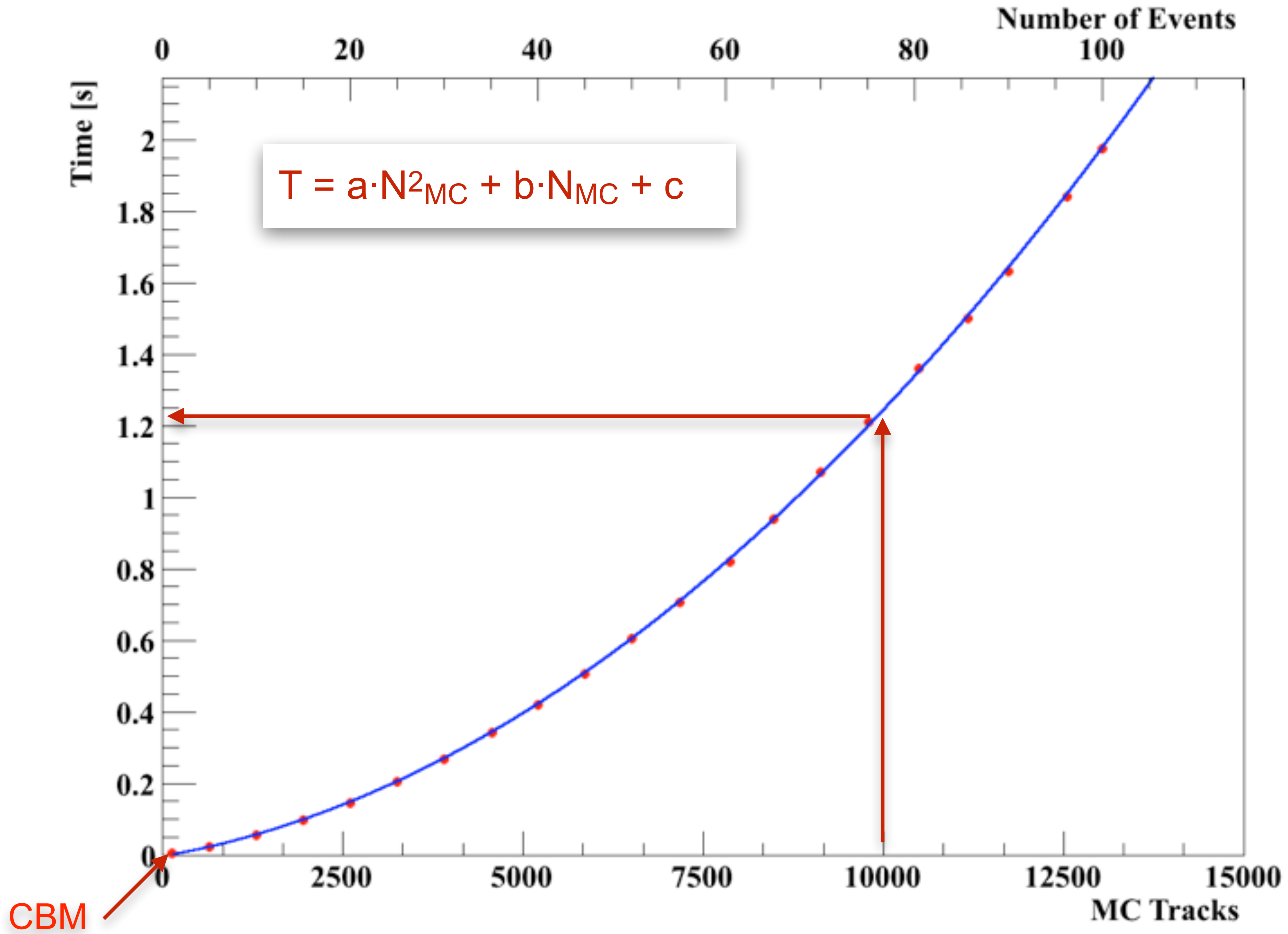
5 mbias events, $\langle N_{reco} \rangle$ = 572



100 mbias events, $\langle N_{reco} \rangle$ = 10340



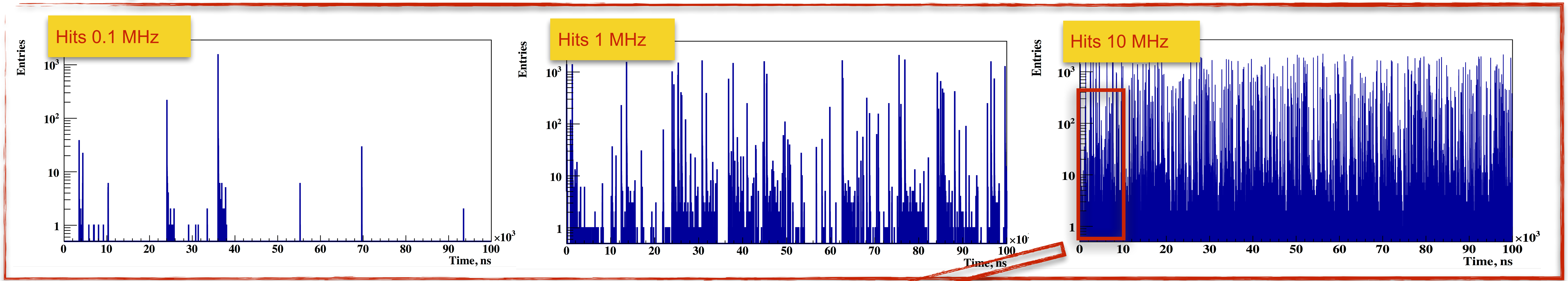
Au+Au mbias events at 25 AGeV



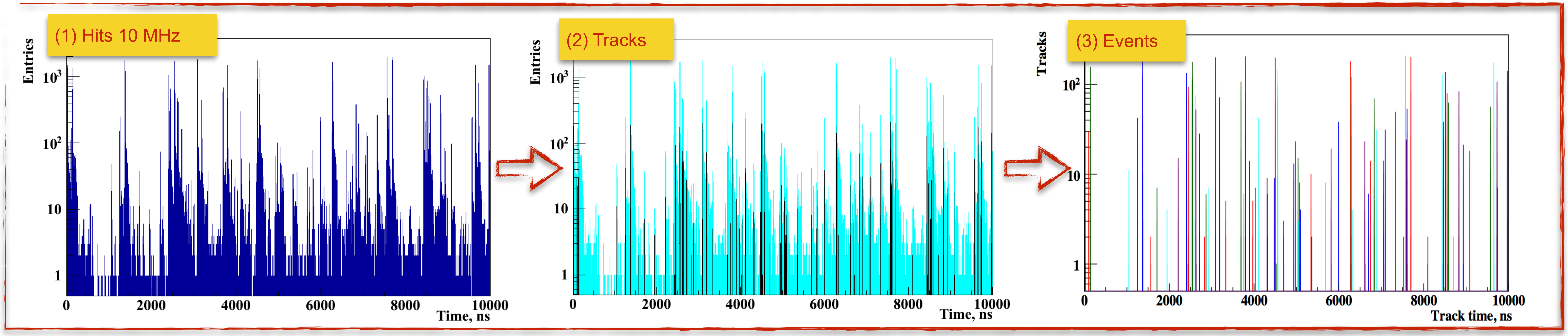
Reliable reconstruction efficiency and time as a second order polynomial w.r.t. to the track multiplicity

4D Event Building at 10 MHz

Hits at high input rates

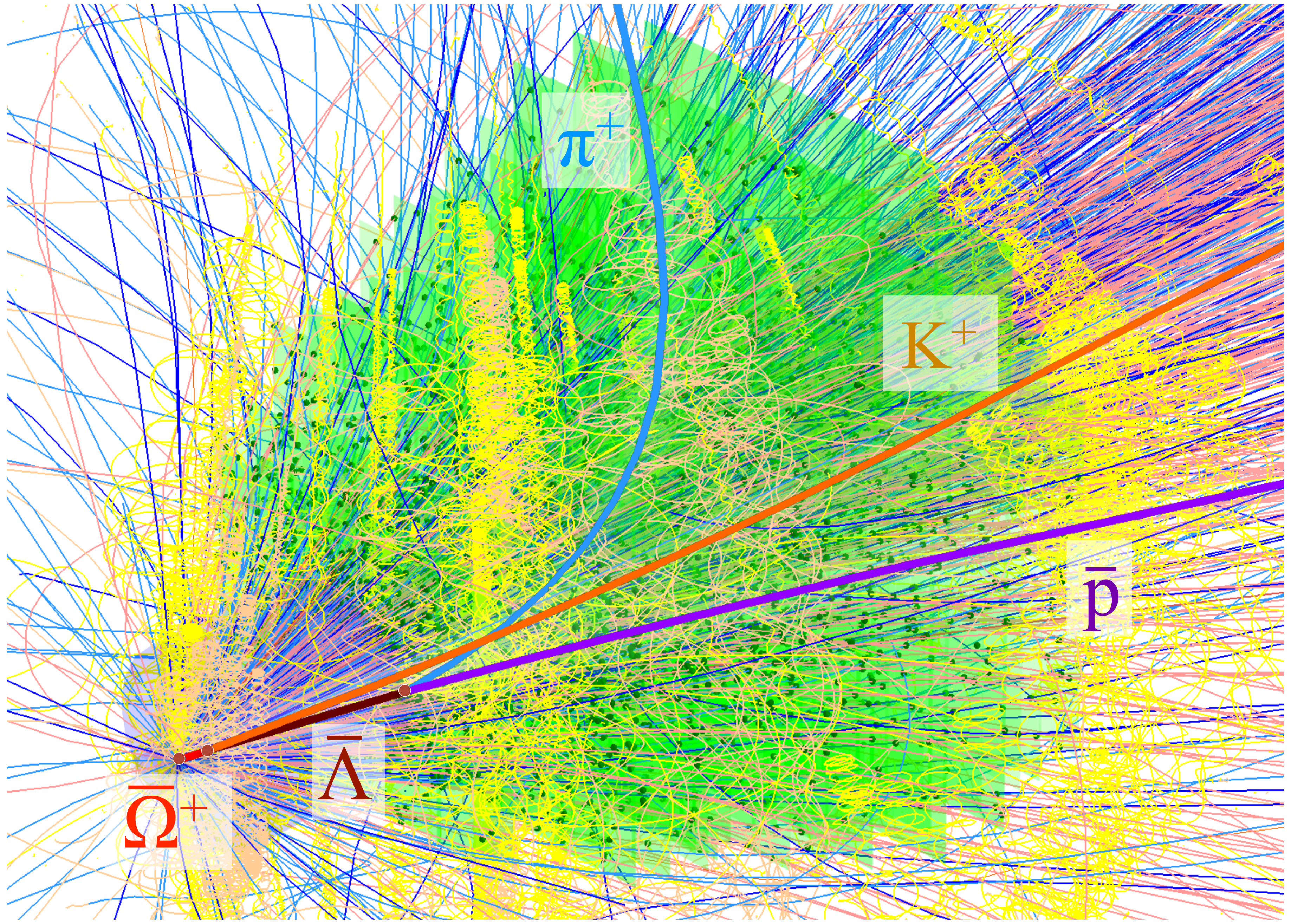


From hits to tracks to events



Reconstructed tracks clearly represent groups, which correspond to the original events

KF Particle: Reconstruction of short-lived Particles



Simulated AuAu collision at 25 AGeV

$$\bar{\Omega}^+ \leftarrow \bar{\Lambda}\, K^+$$
$$\hookrightarrow \bar{p}\, \pi^+$$

```
KFParticle Lambda(P, Pi);                        // construct anti Lambda
Lambda.SetMassConstraint(1.1157);                // improve momentum and mass
KFParticle Omega(K, Lambda);                     // construct anti Omega
PV -= (P; Pi; K);                                // clean the primary vertex
PV += Omega;                                     // add Omega to the primary vertex
Omega.SetProductionVertex(PV);                   // Omega is fully fitted
(K; Lambda).SetProductionVertex(Omega);          // K, Lambda are fully fitted
(P; Pi).SetProductionVertex(Lambda);             // p, pi are fully fitted
```

$$\mathbf{r} = \{\, x, y, z, p_x, p_y, p_z, E \,\}$$

State vector

$$\mathbf{C} = \langle \mathbf{r}\mathbf{r}^T \rangle = \begin{bmatrix} \sigma_x^2 & C_{xy} & C_{xz} & C_{xp_x} & C_{xp_y} & C_{xp_z} & C_{xE} \\ C_{xy} & \sigma_y^2 & C_{yz} & C_{yp_x} & C_{yp_y} & C_{yp_z} & C_{yE} \\ C_{xz} & C_{yz} & \sigma_z^2 & C_{zp_x} & C_{zp_y} & C_{zp_z} & C_{zE} \\ C_{xp_x} & C_{yp_x} & C_{zp_x} & \sigma_{p_x}^2 & C_{p_xp_y} & C_{p_xp_z} & C_{p_xE} \\ C_{xp_y} & C_{yp_y} & C_{zp_y} & C_{p_xp_y} & \sigma_{p_y}^2 & C_{p_yp_z} & C_{p_yE} \\ C_{xp_z} & C_{yp_z} & C_{zp_z} & C_{p_xp_z} & C_{p_yp_z} & \sigma_{p_z}^2 & C_{p_zE} \\ C_{xE} & C_{yE} & C_{zE} & C_{p_xE} & C_{p_yE} & C_{p_zE} & \sigma_E^2 \end{bmatrix}$$

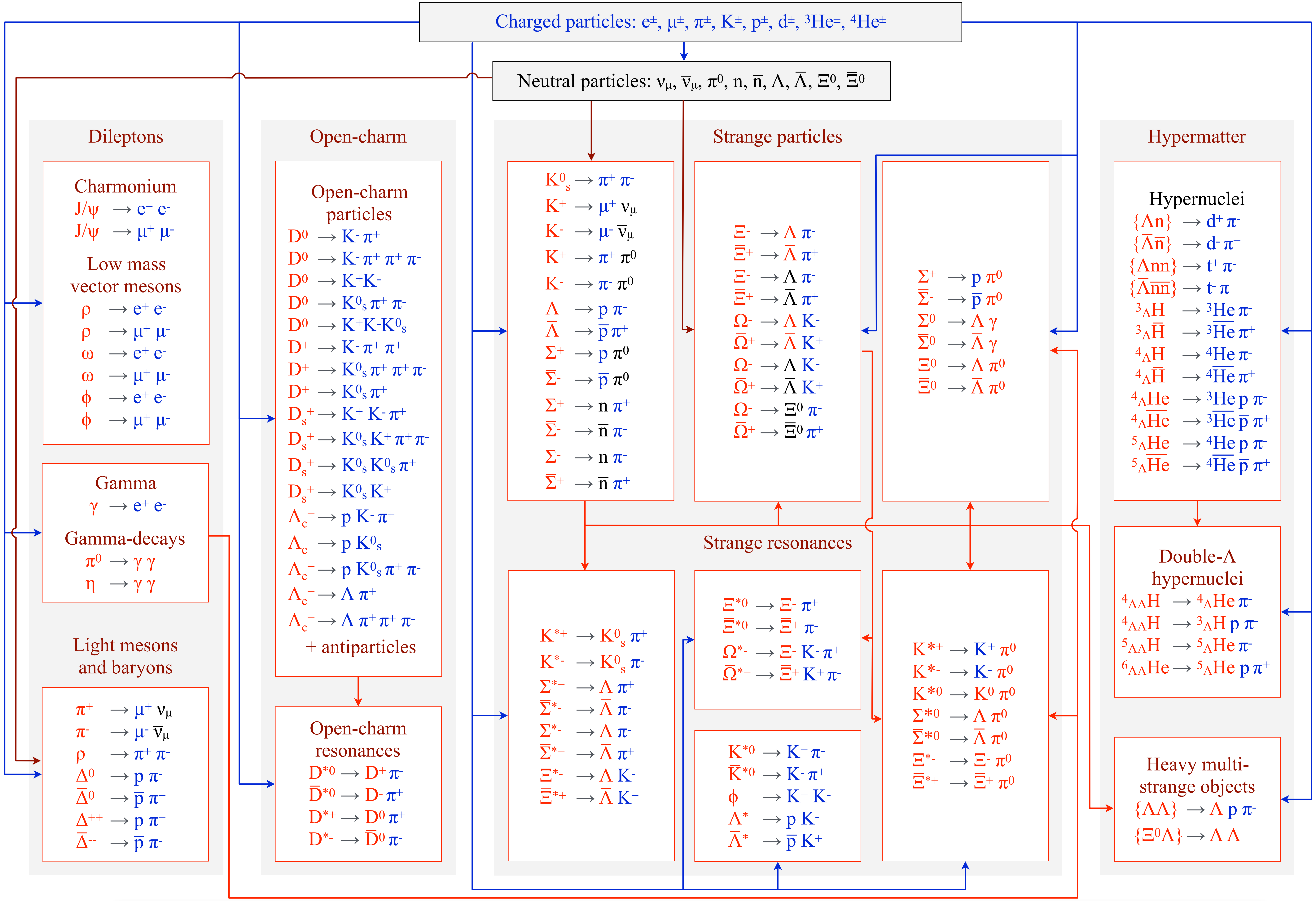
Covariance matrix

Features:

- KF Particle class describes particles by the state vector and the covariance matrix.
- Covariance matrix contains essential information about tracking and detector performance.
- The method for mathematically correct usage of covariance matrices is provided by the KF Particle package based on the Kalman filter (KF).
- Heavy mathematics of KF requires fast and vectorised algorithms.
- Mother and daughter particles are treated in the same way.
- The natural and simple interface allows to reconstruct easily complicated decay chains.
- The package is geometrically independent and can be adapted to other experiments (CBM, PANDA, ALICE, STAR, …).

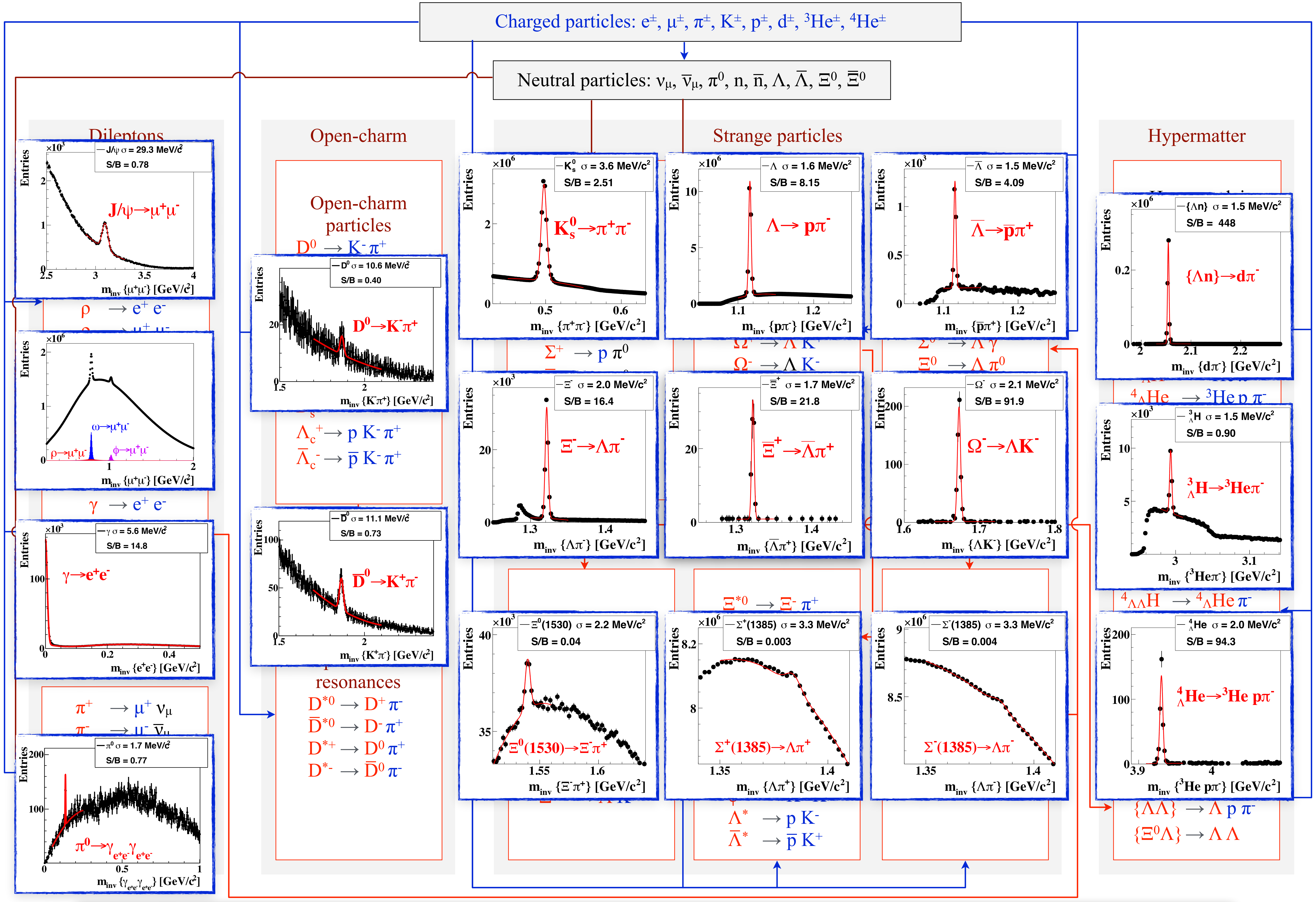
KF Particle provides a simple and very efficient approach to physics analysis

KF Particle Finder for Physics Analysis and Selection



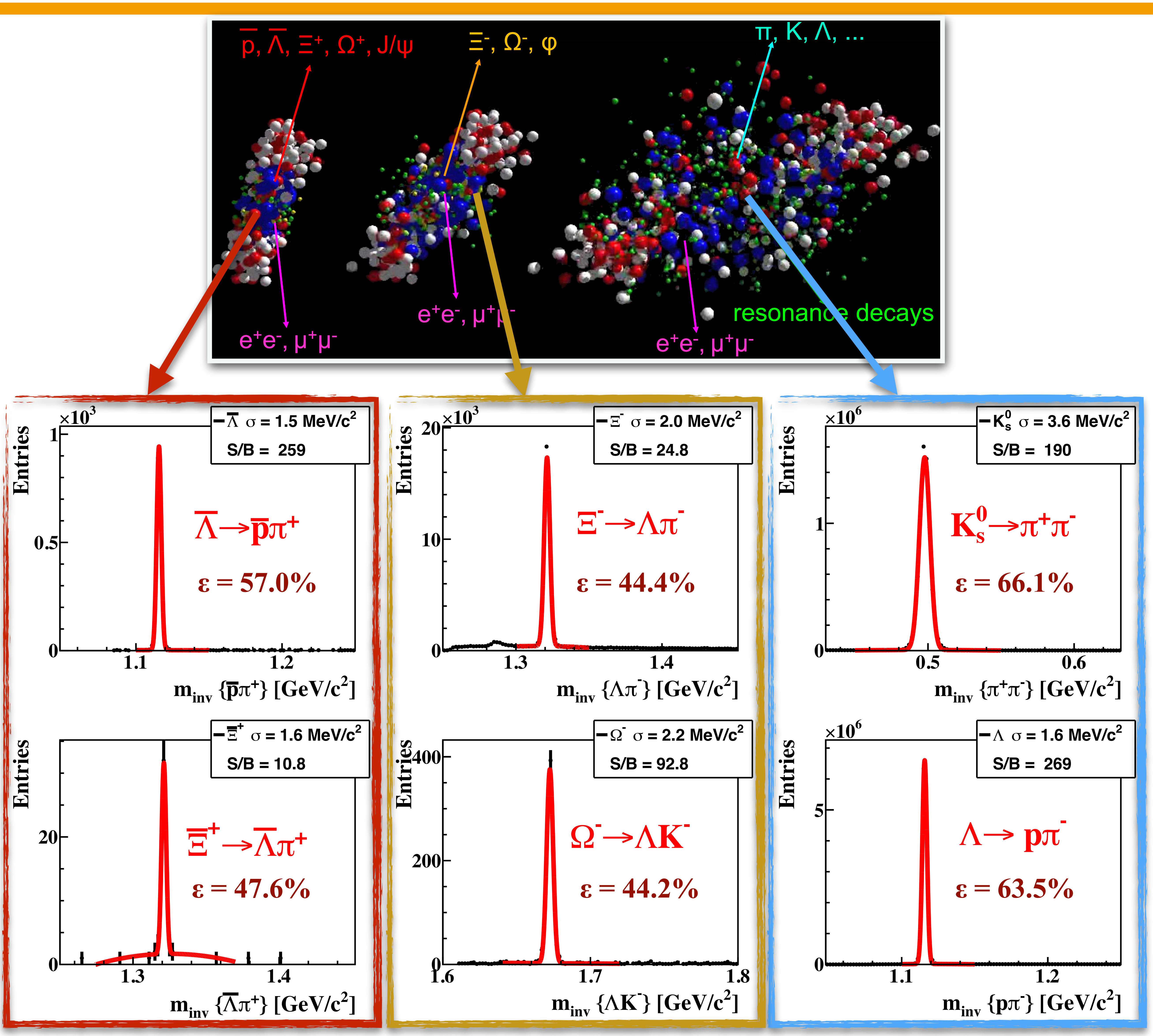
The KF Particle Finder package has implemented more than 150 decay channels

KF Particle Finder for Physics Analysis and Selection



A common platform for offline physics analysis and for real-time express analysis (100 μs/core/decay)

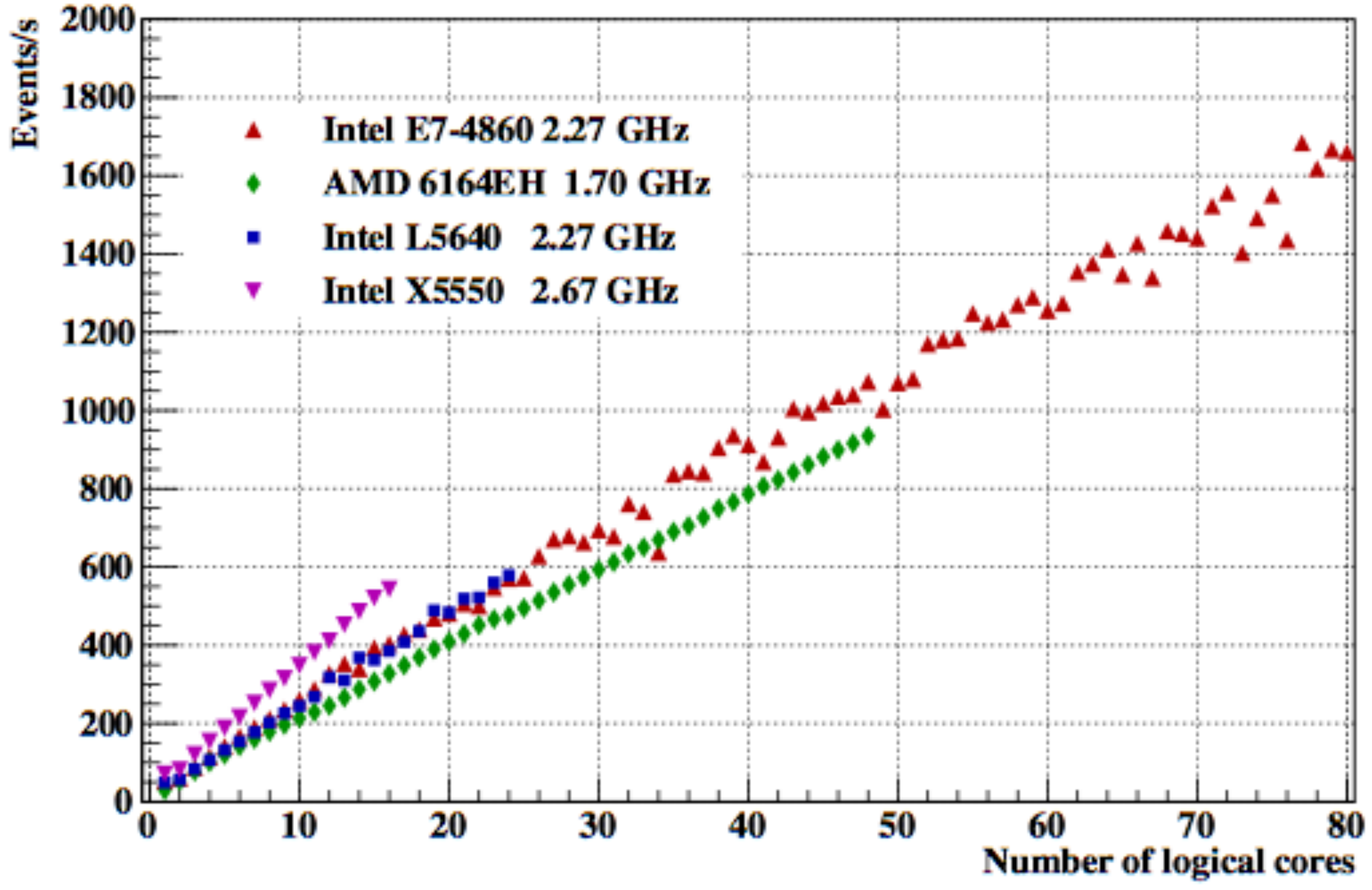
Very Clean Probes of Collision Stages

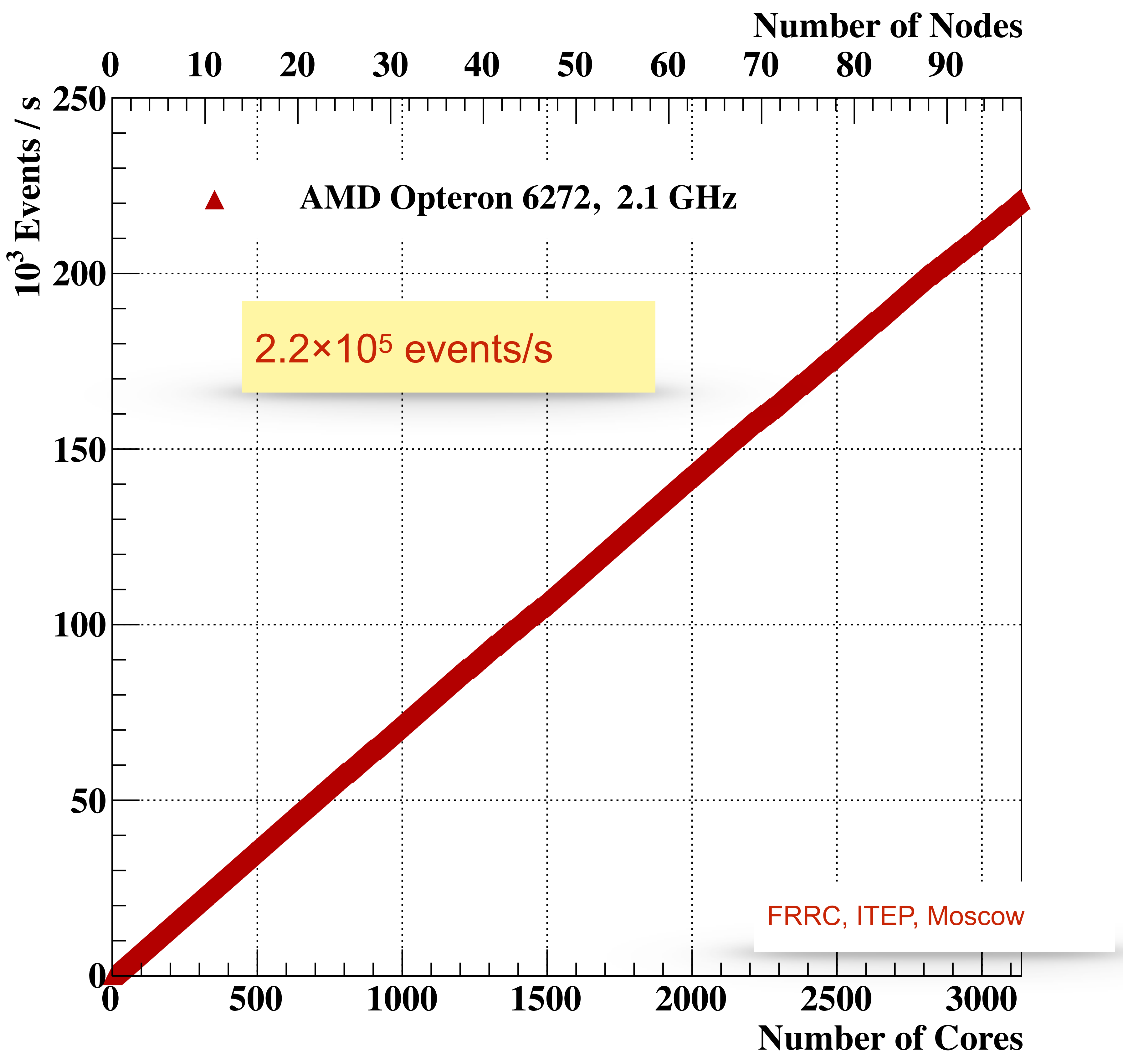


AuAu, 10 AGeV, 3.5M central UrQMD events, **MC PID**

Clean reconstruction of short-lived particles produced at different stages of heavy-ion collisions

Running FLES on HPC Node/Farm



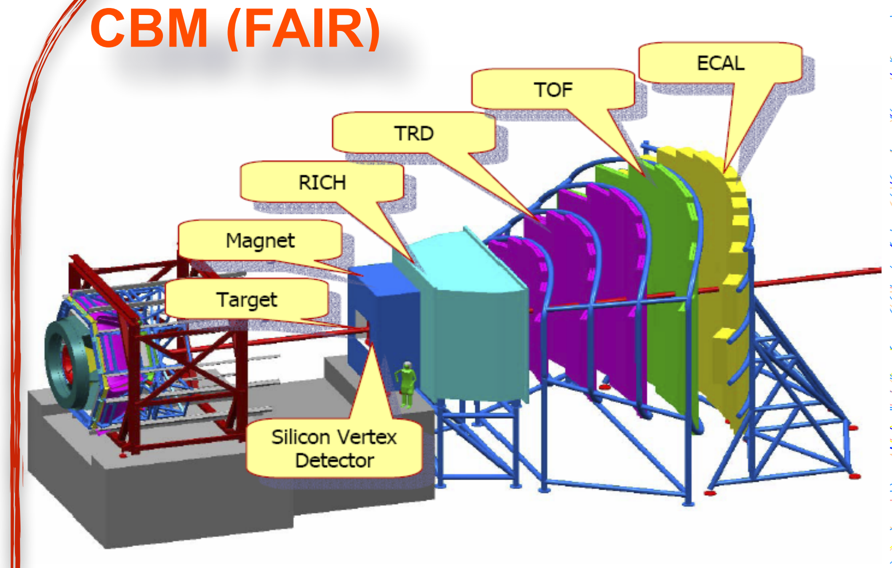


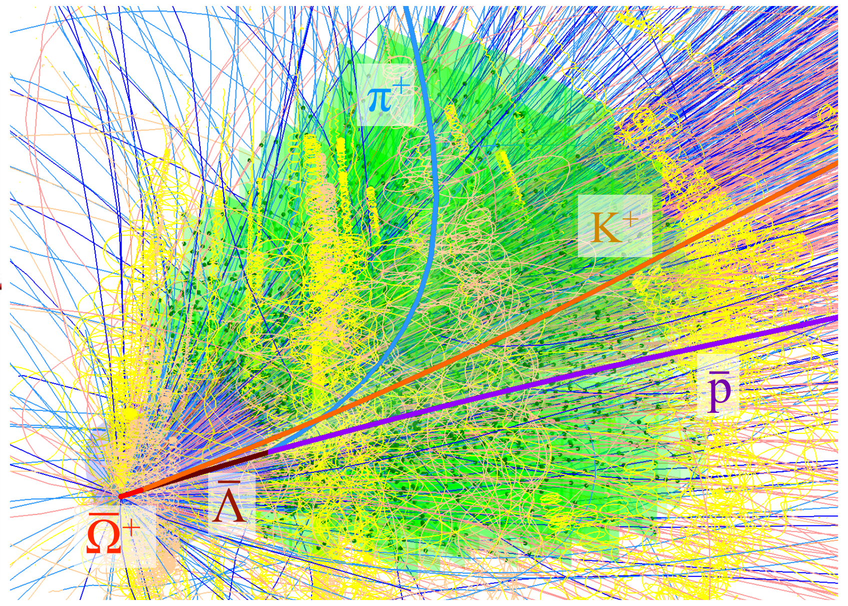
The FLES package is vectorized, parallelized, portable and scalable: 2.2×10^5 events/s on 3 200 CPU cores

CBM -> STAR: Reconstruction and Analysis Software

Within the FAIR Phase-0 program the CBM KF Particle Finder has been adapted to STAR and applied to real data of 2014, 2016 and BES-I.

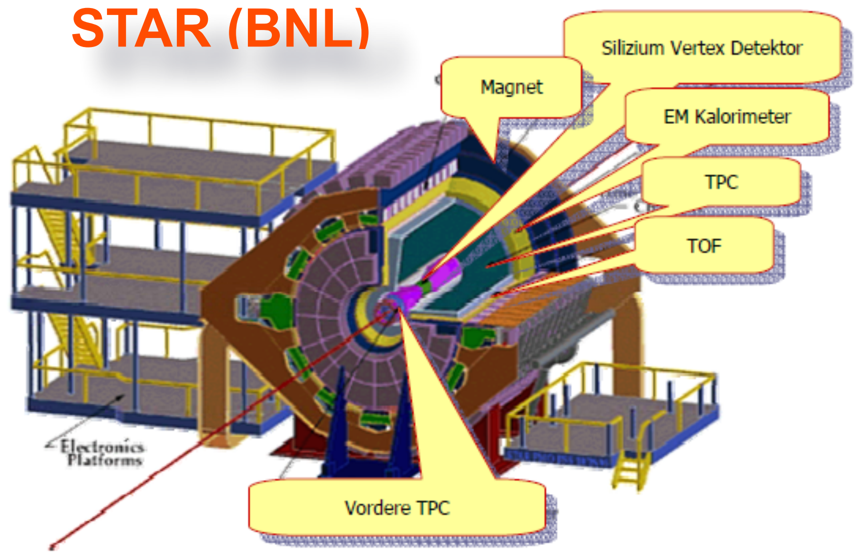
CBM (FAIR)

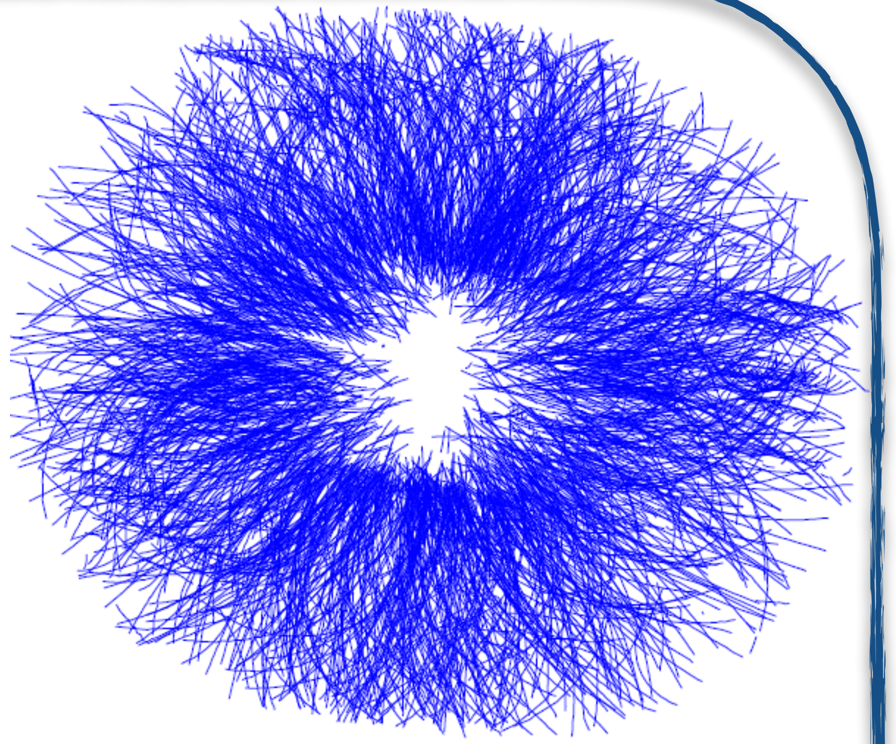




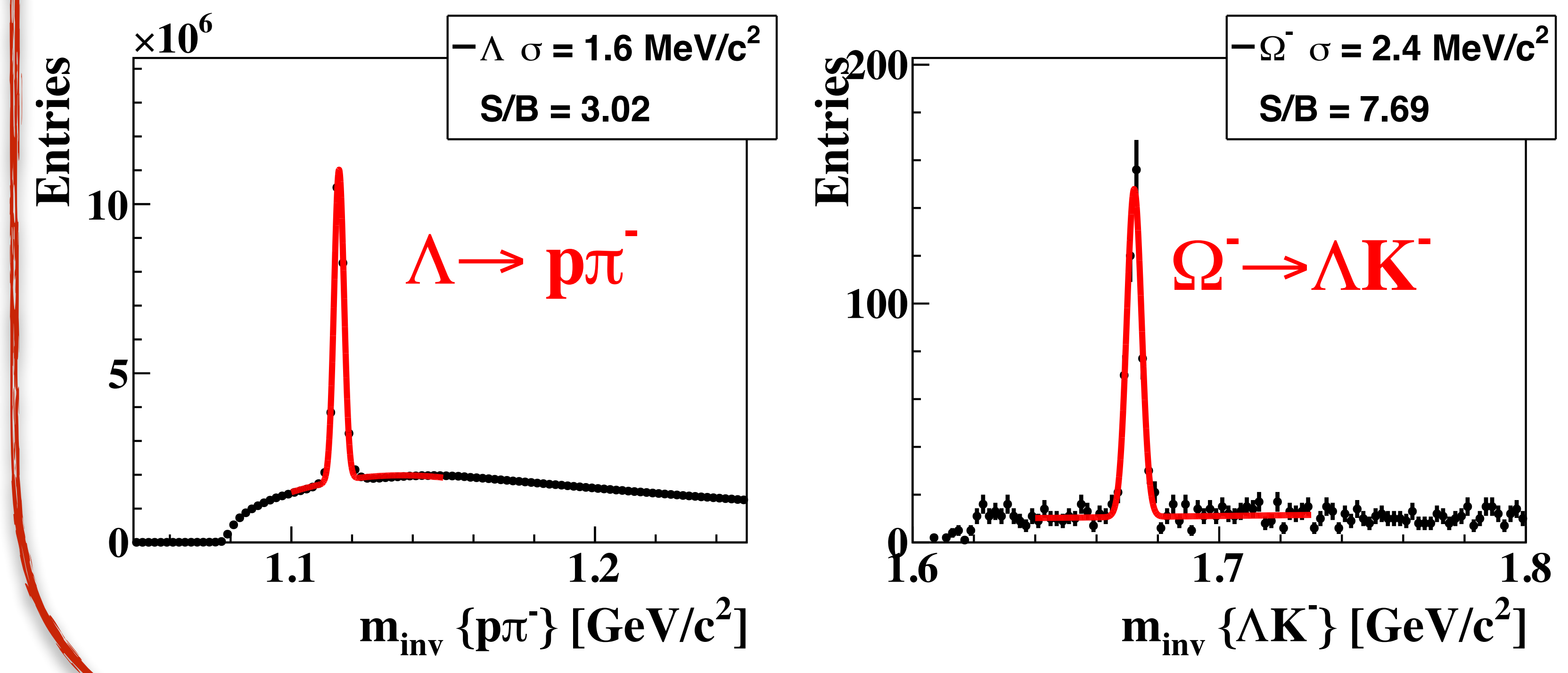
$\bar{\Omega}^+ \rightarrow \bar{\Lambda} K^+$
$\hookrightarrow \bar{p}\, \pi^+$

STAR (BNL)

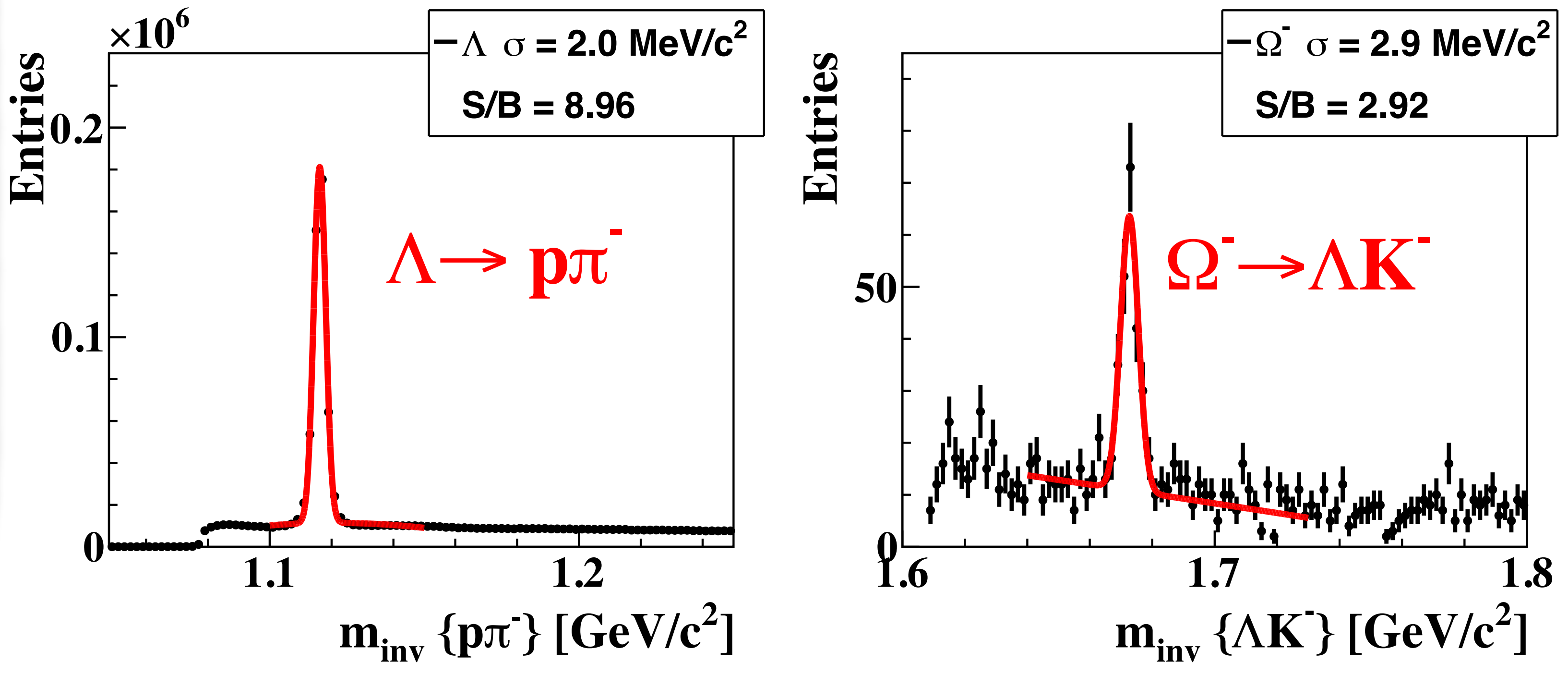




CBM, 5M central Au+Au, 10 AGeV, PHSD



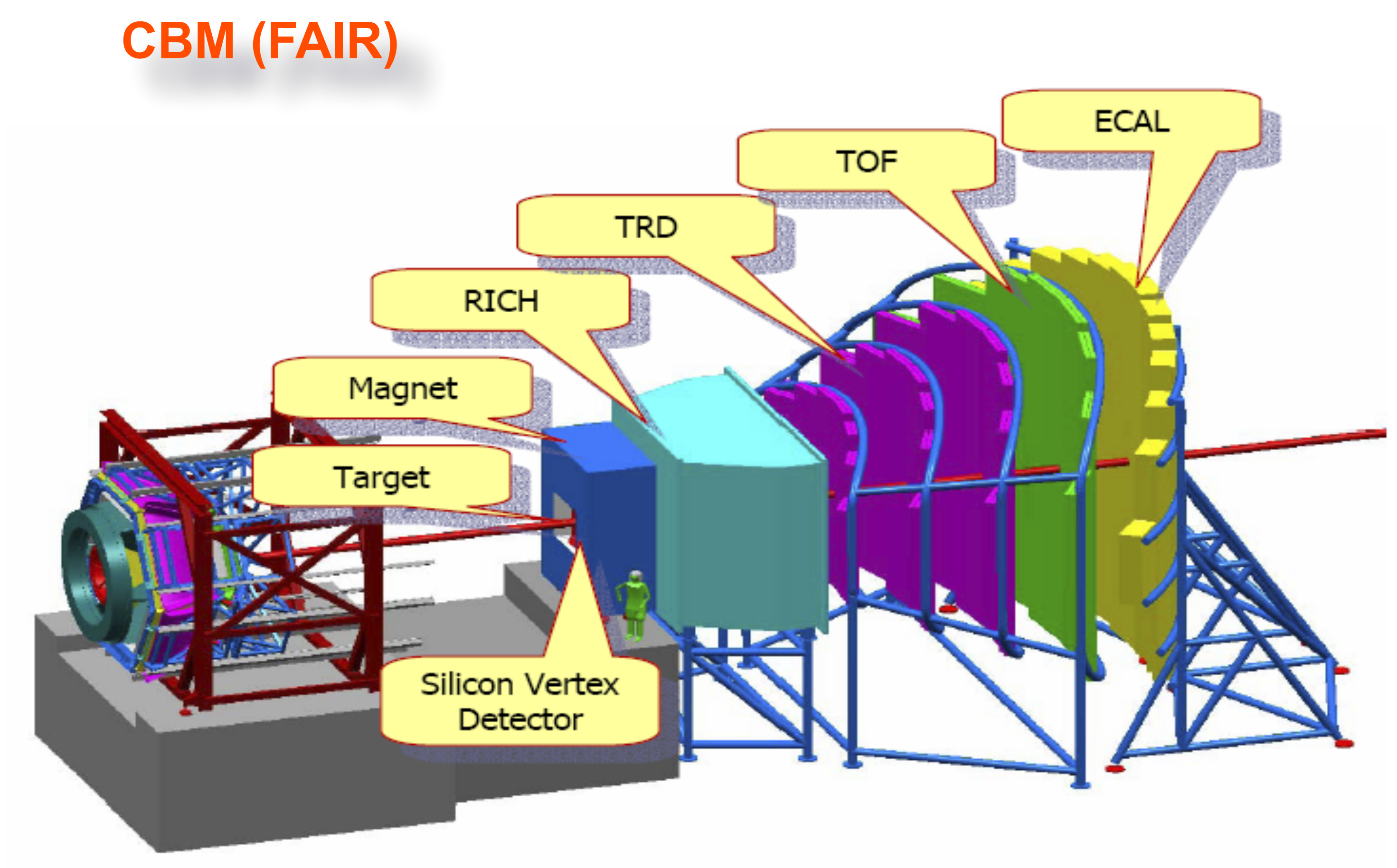
STAR, 1.3M mbias Au+Au, 200 AGeV, Run 2016

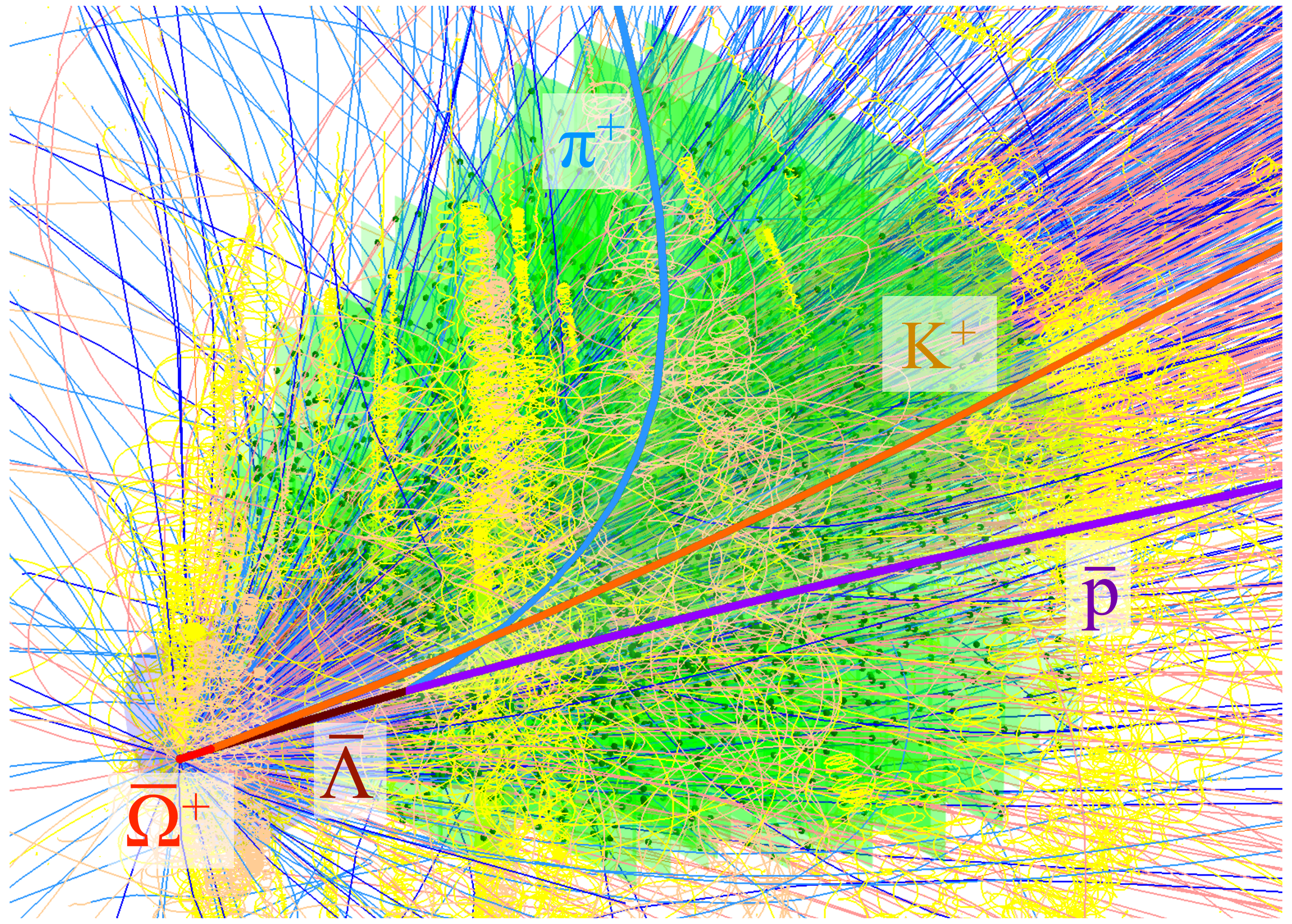


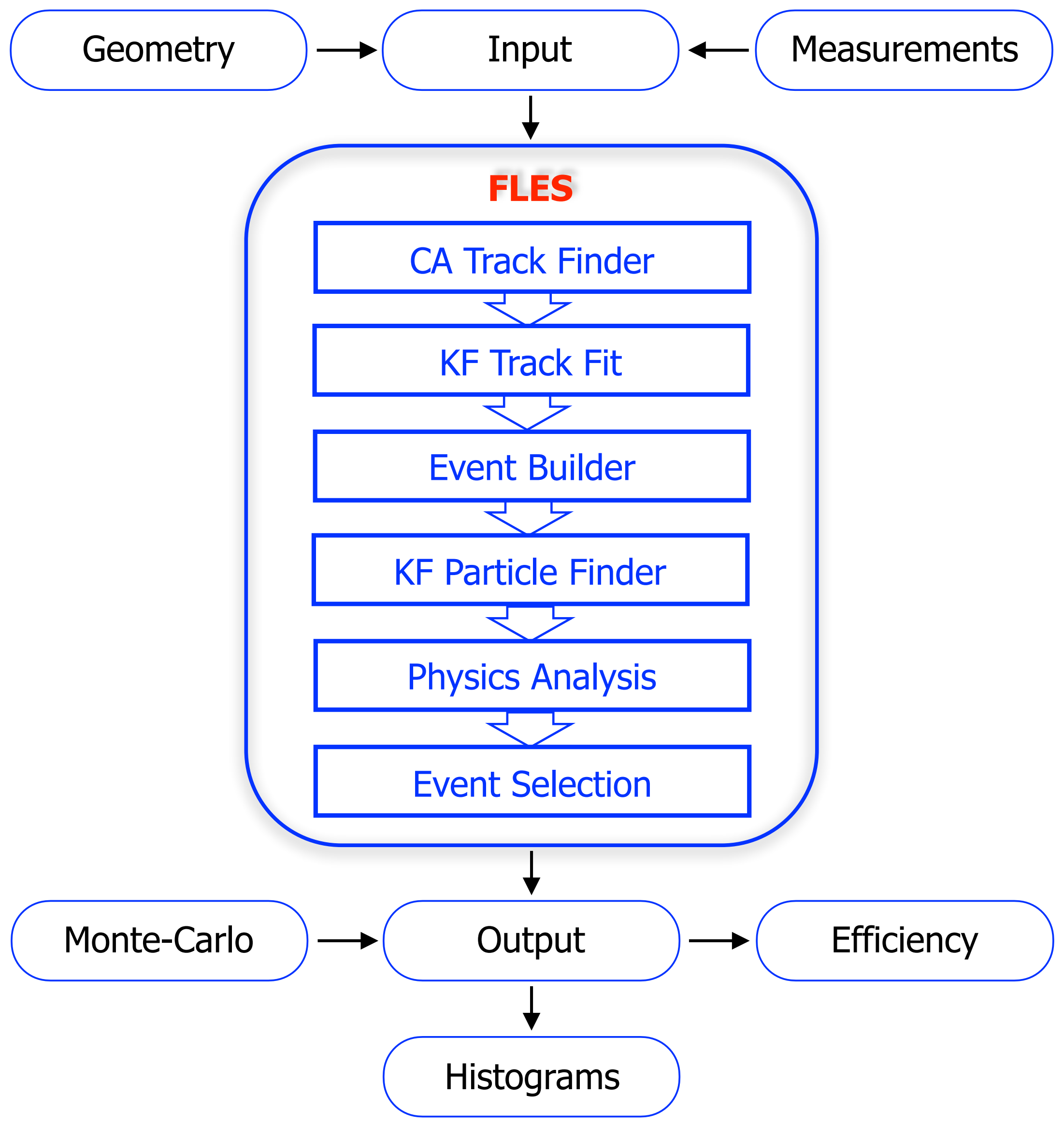
- ✓ Since 2013 (online) and 2016 (offline) the CA track finder is the standard STAR track finder for data production. Use of CA provides 25% more D0 and 20% more W with respect to the traditional track following approach.
- ✓ The KF particle finder provides a factor 2 more signal particles due to the use of covariance matrices. The integration of the KF particle finder into the official STAR repository for physics analysis is currently in progress.

Preparing for the real-time express physics analysis during the BES-II runs (2019-2020)

Conclusion







1. The CBM experiment with 10^7 input rate will require the full event reconstruction and physics analysis of the experimental data online. As the same HPC farm will be used for offline and online processing of experimental data, the main reconstruction and analysis algorithms will work both offline and online. Errors and insufficient accuracy in online data processing, physics analysis or selection of interesting collisions by these algorithms will lead to complete loss of all experimental data, since only the incorrectly selected data will be stored in this case.
2. Hence the need to redesign the existing offline algorithms for their efficient, fast and reliable online operation, and to adapt them for use of vector (SIMD) registers as well as for their work in the parallel mode on many-core CPU and GPU architectures.
3. We have demonstrated, that the core algorithms of the FLES package, the Cellular Automaton for searching for particle trajectories (100 μs/core/track) and the Kalman Filter to estimate their parameters (0.5 μs/core/track), have a very high level of intrinsic parallelism for their fast and efficient implementation on many-core CPU/GPU architectures.
4. The KF Particle Finder package with more than 150 decay channels implemented (100 μs/core/decay) is a common platform for offline physics analysis and for real-time express analysis at 10^7 interaction rate in CBM.
5. Adaptation of the FLES algorithms within the CBM Phase-0 program to the STAR experiment with its excellent detector performance, high quality experimental data and a well established reconstruction chain is the first and successful step in preparing the FLES algorithms for reconstruction and analysis of CBM real data at Day-1.