# GeantV fast simulation ideas and perspectives

Andrei Gheata for the GeantV collaboration

CERN, May 25-26, 2016

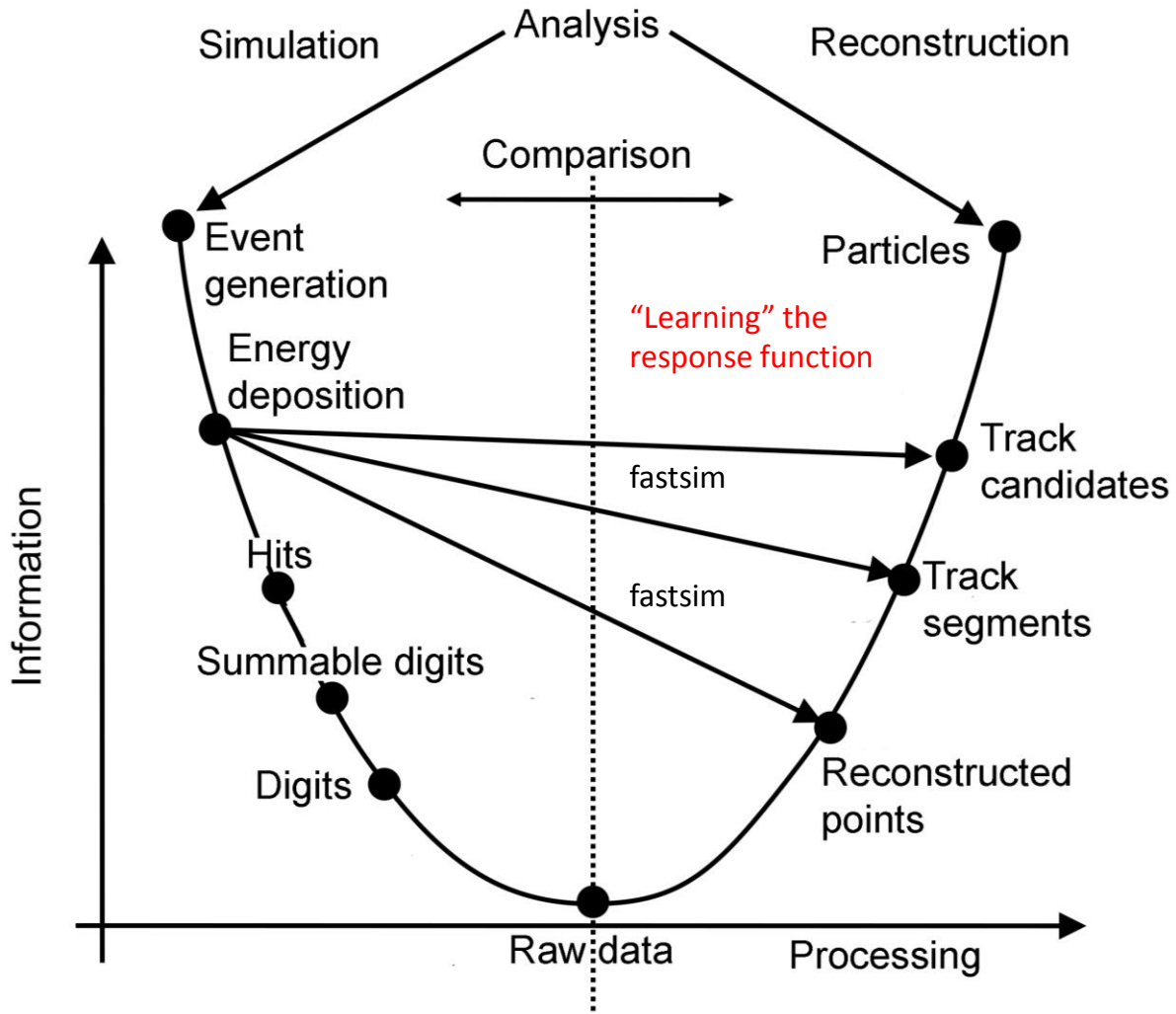
G.Amadio (UNESP), Ananya (CERN), J.Apostolakis (CERN) , A.Arora (CERN), M.Bandieramonte (CERN), A.Bhattacharyya (BARC), C.Bianchini (UNESP), R.Brun (CERN), P.Canal (FNAL), F.Carminati (CERN), L.Durhem (intel), D.Elvira (FNAL), A.Gheata (CERN), M.Gheata (CERN), I.Goulas (CERN), R.Iope (UNESP), S.Jun (FNAL), H.Kumawat (BARC), G.Lima (FNAL), A.Mohanty (BARC), T.Nikitina (CERN), M.Novak (CERN), W.Pokorski (CERN), A.Ribon (CERN), R.Sehgal (BARC), O.Shadura (CERN), S.Vallecorsa (CERN), S.Wenzel (CERN), Y.Zhang (CERN)

# THE GEANTV PROJECT

# Background

- In the best case the speedup from GeantV will be O(10)
- From the estimations collected informally, at least O(100) is needed
- This is the realm of fast simulation

# Definition


Simulation
Analysis
Reconstruction
Comparison
Event generation
Energy deposition
Particles
"Learning" the response function
Track candidates
fastsim
Hits
fastsim
Track segments
Summable digits
Digits
Reconstructed points
Information
Raw data
Processing

# History

- Fastsim has historically been introduced almost as an “afterthought” or a “user hook” in simulation programmes
  - Geant4 introduced fastsim hooks early, but they were not “abused” – Why?
- In GeantV we would like to introduce fast simulation in the framework

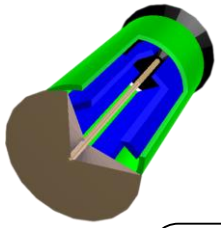
# What and how?

- It is possible to activate fast simulation on a user filter
  - e.g. region / particle /energy basis
- Provide generic tools to correlate output information (e.g. digits, rec points, or even tracks) with MC truth for selected particles
- A choice of basic fast simulation algorithms will be proposed for trackers and for calos

# Too detector specific?

- The same was said for detailed simulation before EGS came out!
- User code definitely needs to be involved, so the approach must be user code friendly…
- We do not pretend to cater for all cases, but to offer a reasonable “library” of algorithms on which to build, and a framework to easily embed user code

**GeantV:**
**Trying things differently**



Scheduler

Basket of tracks

Dispatching

Basketizer

Basket of tracks

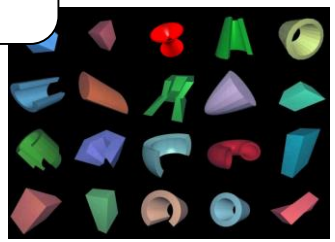
MIMD

SIMD

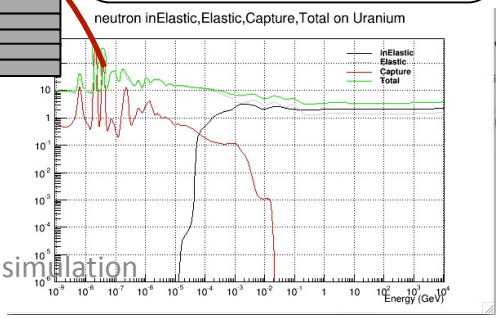
Geometry navigator

Physics

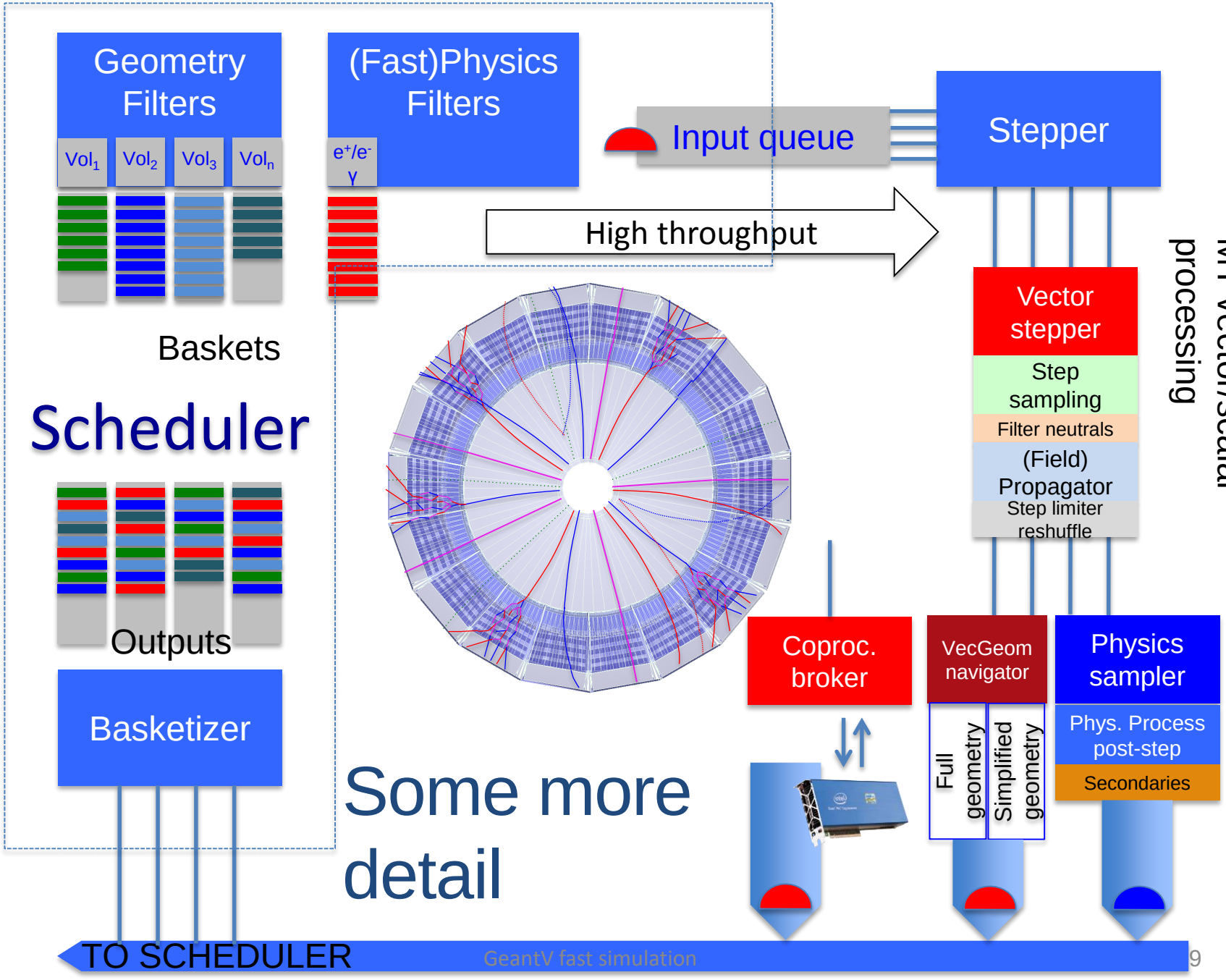
Geometry algorithms



x-sections



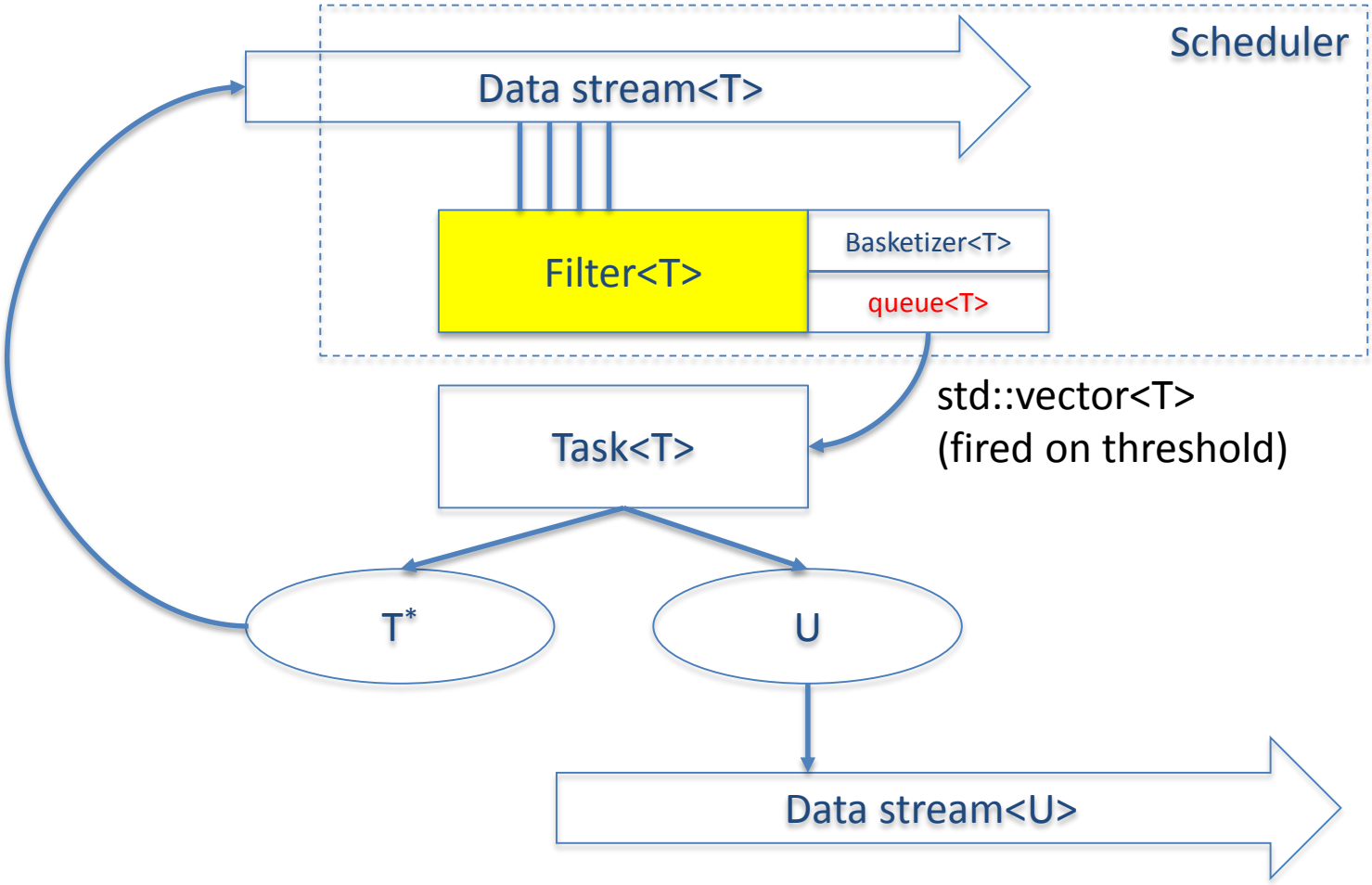

Reactions


Geometry Filters
Vol1
Vol2
Vol3
Voln
(Fast)Physics Filters
e+/e-
γ
Input queue
Stepper
High throughput
Vector stepper
Step sampling
Filter neutrals
(Field) Propagator
Step limiter reshuffle
MT vector/scalar processing
Baskets
Scheduler
Outputs
Basketizer
Coproc. broker
VecGeom navigator
Full geometry
Simplified geometry
Physics sampler
Phys. Process post-step
Secondaries
Some more detail
TO SCHEDULER

# Who controls the game?

- Fast simulation creates bridges between simulation and tracking/reconstruction
- How to interconnect in an experiment independent way these components?
- All is about scheduling the work and data flow, using a common concurrent infrastructure
- GeantV proposes tools for this kind of approach
  - Simulation is a natural place to apply these parameterizations and to check their effect
  - The library to learn the parameterizations and compare with MC truth is independent, but can be delivered with GeantV (as long as it stays generic)

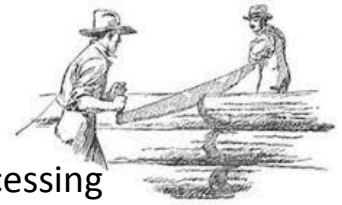
# GeantV concurrent tasks



# The complex scheduling business

- Efficient concurrent *basketizers*
  - Filtering tracks by several possible locality criteria
    - Geometry, (fast)physics process
  - Giving reasonable size vectors all along the simulation

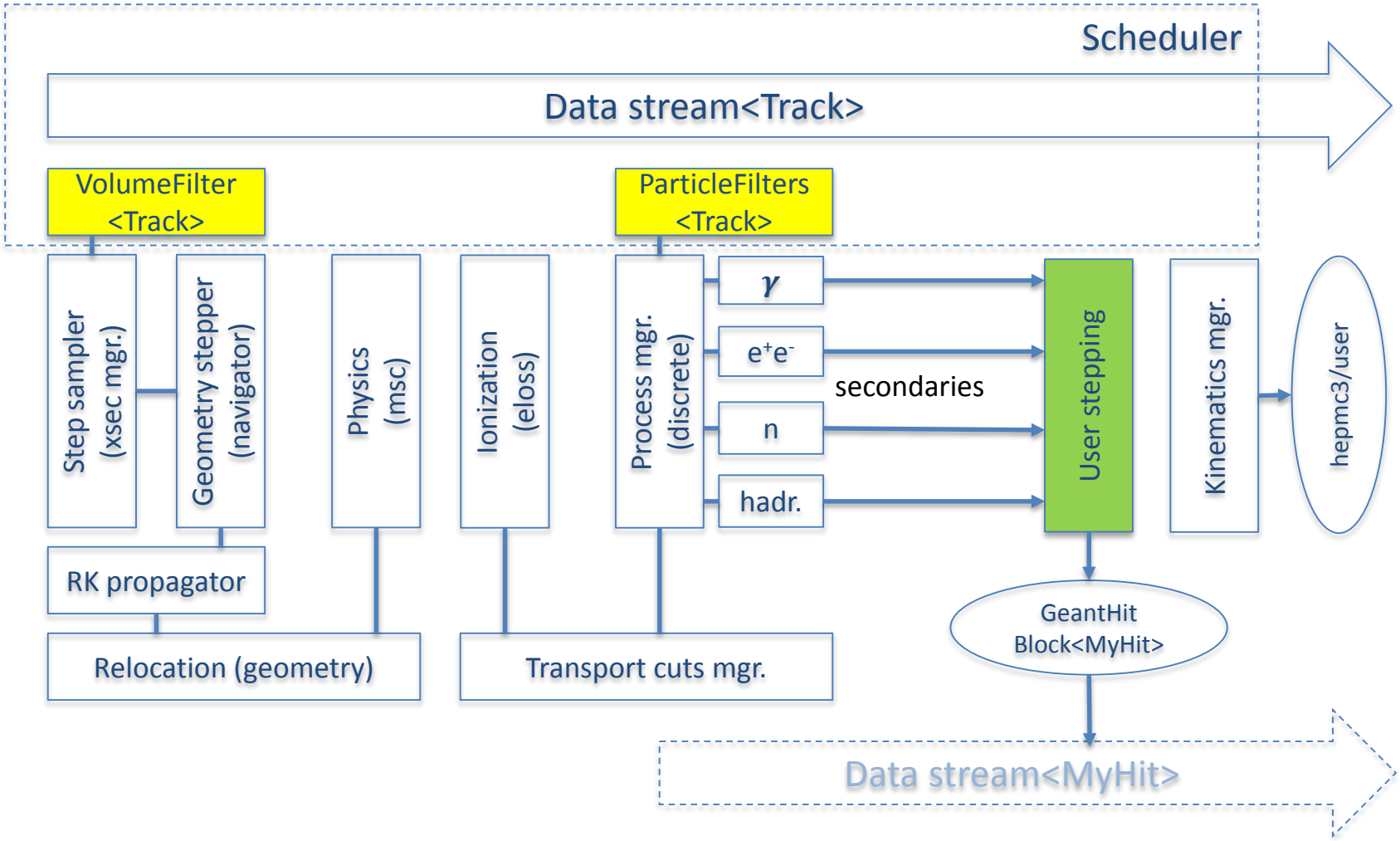
- Provide scalable & balanced workload
  - Minimize memory footprint
  - Minimize cool-down phase (tails)
    - Using concurrent event injection and prioritized processing



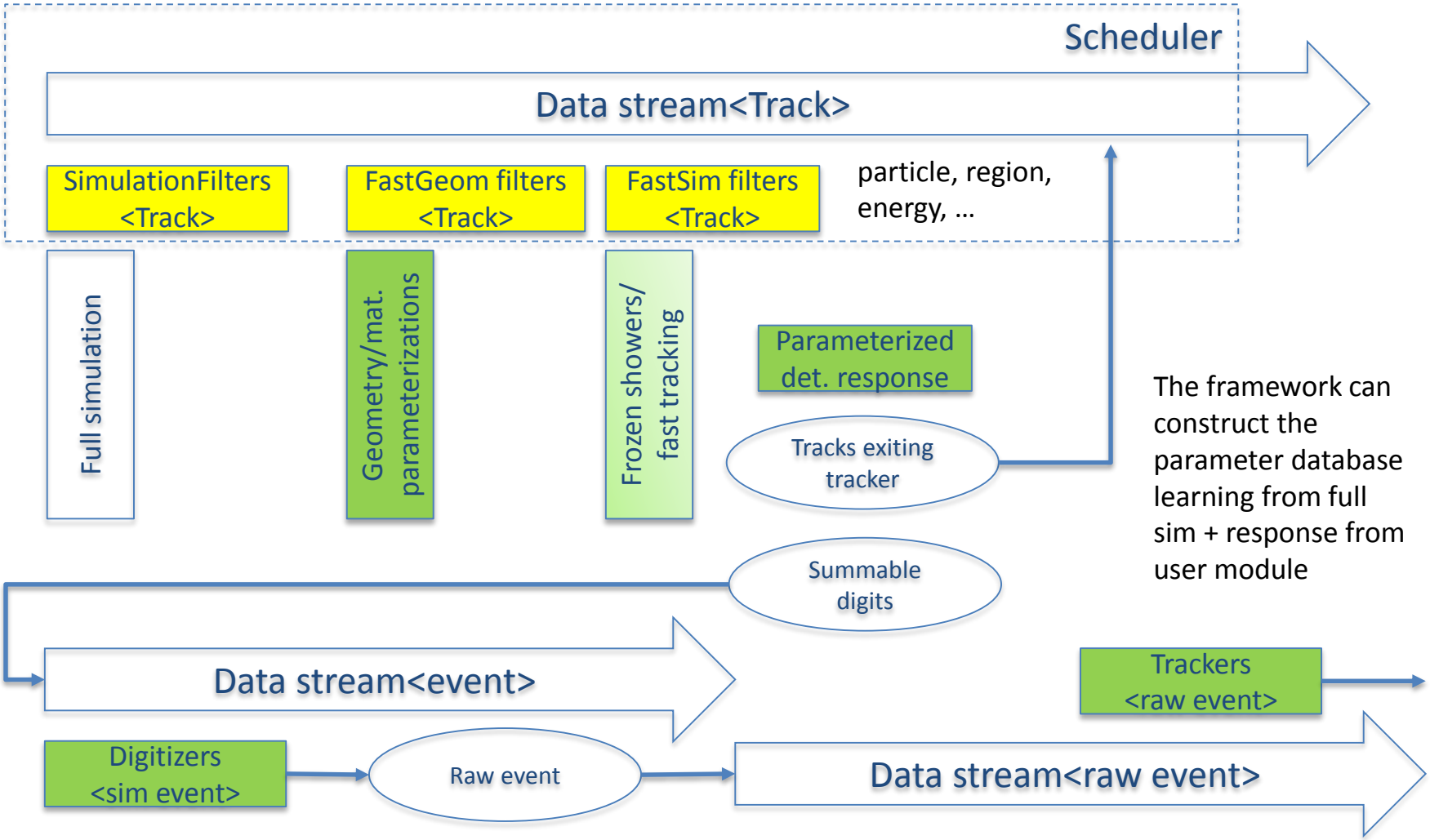
- Adaptive behavior to maximize performance
  - Dynamic switch of vector/scalar processing
  - Learning dynamically the “important” filters
  - Adjust dynamically event slots to control memory

- Accommodate additional concurrent processing in the ~~simulation~~ workflow
  - Hits/digits/kinematics I/O
  - Digitization/reconstruction tasks

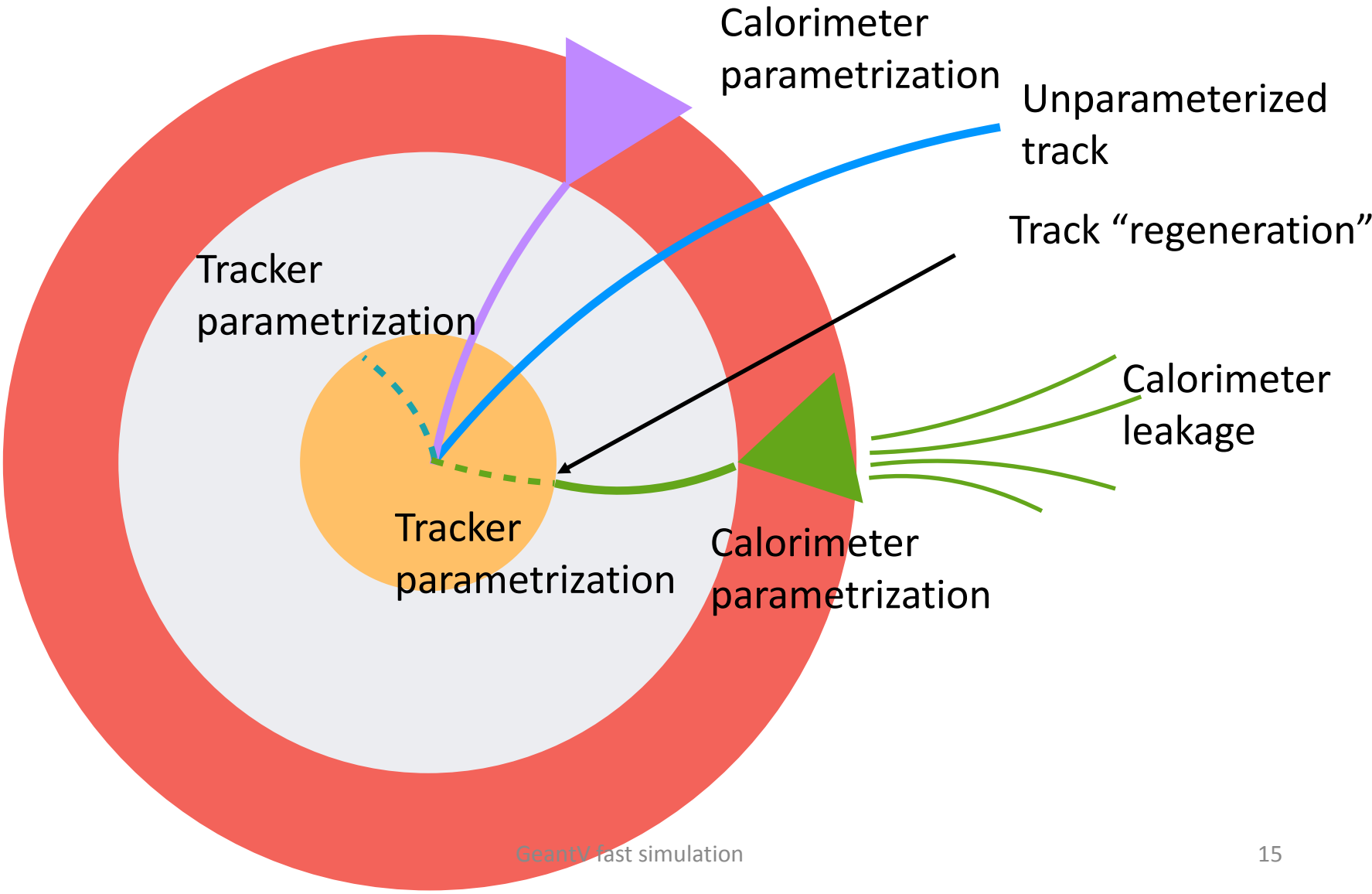
# GeantV simulation flow



# Embedding fast simulation



# Combined simulation



# Framework functions

- Provide tools to
  - Build libraries of parametrized quantities from detailed simulation
  - Compare detailed and parametrized simulation
  - Allow user-driven switch between different options (full, parametrization, param+regeneration)
- Apply these tools to simple examples (provided)
  - Generalizing in experiment-independent manner starting from these

# And…

- Parametrized simulation is considerably more “vector friendly” then detailed one

# In summary

- We are not proposing a “one fit all” solution
- We are proposing a “toolbox” providing scheduling and interconnection hooks, plus simple examples
- We want to learn from you new use cases, discuss how to accommodate them