L1 and Velo Software Status

- Current status in GAUDI
- Ongoing projects
- Quality checks

Current Status in GAUDI

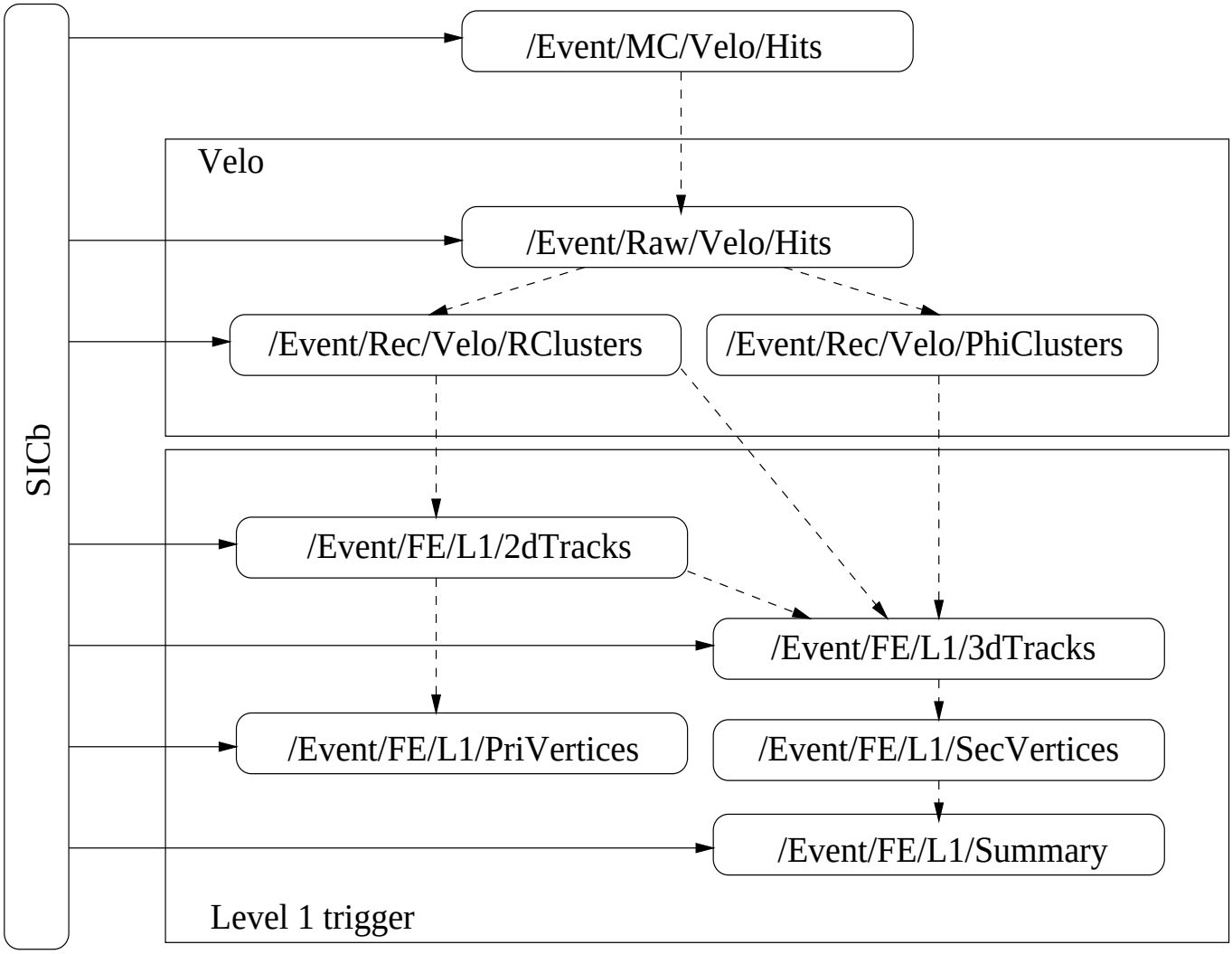
- SICb converters written for all Level 1/Velo banks (including links) - see note LHCb 2000-009

LHCbEvent directory	SICb bank
/Event/MC/ Velo/Hits	VRPR
PileupHits	VPRW
Links/MCHit2MCParticles	none
MCParticle2PhiClusters	none
MCParticle2RClusters	none
PhiCluster2MCParticles	none
RCluster2MCParticles	none
RawHit2MCParticles	none
RawHit2MCVeloHits	none
Raw/Velo/Hits	VRHT
Rec/ Velo/RClusters	VSCR
PhiClusters	VSCP
FE/ L1/ 2dTracks	TVT21
3dTracks	TVT31
PriVertices	TVTX1
SecVertices	TVTX1
Summary	TVTS1

- can access all data which is available in SICb
- currently *not* in tagged release of GAUDI - no public libraries
  - this makes it awkward to use as it requires you to check out 8 packages!

What we have now:

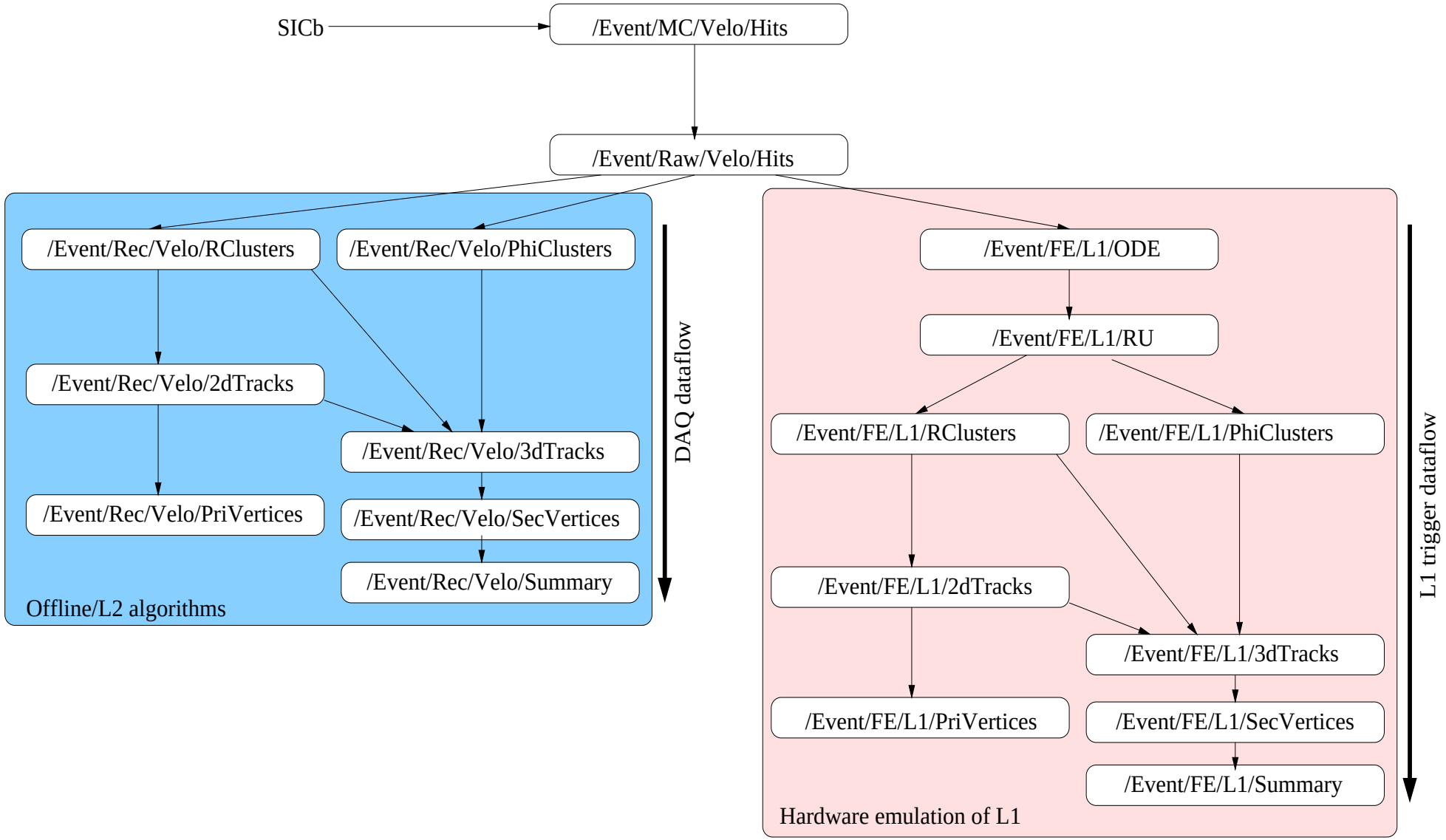
- everything is converted - nothing calculated



- the dashed lines represent the virtual dataflow as exists in SICb - the algorithms to make e.g. tracks from clusters are the next stage

Just a facade with no algorithms!

What we aim to have:



Ongoing projects:

- L1 trigger code development in C (Heidelberg)
  - build ODE and RU banks for hardware emulation
  - 2D track finding as in SICb (triplets)
  - ... reproduce TP results within GAUDI framework
- 3D track finding in C++ (Cracow)
  - required for finding beam when detector retracted
- Analytic primary multi-vertex finder
- Reproduce all the SICb algorithms in the framework - this includes “fast” C routines destined for the L1 trigger plus C++ algorithms for offline reconstruction
  - requires geometry (read cdf files/XML?)
  - requires constants (ditto)
- What we should add at the same time:
  - inefficiency
  - noise
  - overlay pileup events (in SICb or GAUDI?)
  - overlay spillover from previous event (ditto)

aim to implement first version by summer 2000

Quality checks:

- What we can check
  - banks exist - better done centrally: same code for all systems
  - physics distributions: hits, tracks
  - some code already exists to make histograms (M. McCubbin - spin-off from VELO optimisation) can add more