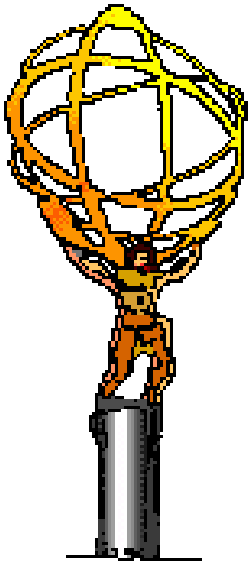
Trig. Perf. & Benchmarking



Julius Hrivnac
Matthias Sessler
Maria Smizanska

TDAQ Chamonix Workshop 22 Oct. 1998

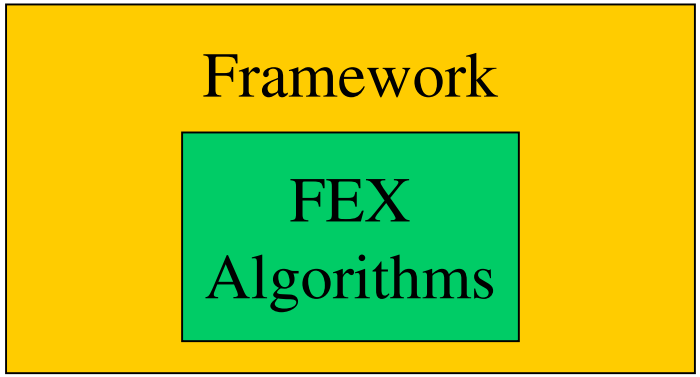
Contents

- General Trigger Performance Framework
- Aims of Existing Frameworks:
  - ➜Standalone C FEX Algorithms
  - ➜ATRIG
  - ➜CTRIG
  - ➜LVL2 Reference Software
- Aims of New Framework
  - ➜TRIG++

General Trigger Performance Framework

- Interface between FEX Algorithms and Framework depending on Task:
  - ➜full analysis / benchmarking / event display
- Algorithms potentially return Tracks/Clusters, Digits, Kine

(Digit = Hit)
(Kine = Monte Carlo
Information = Truth)



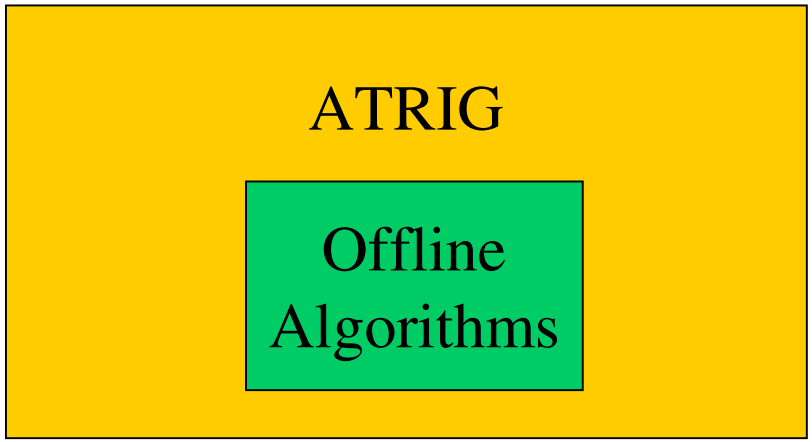
Standalone C FEX Algorithms

- Aim: Benchmarking of LVL FEX Algorithms (and Quality Tests)
- But: Need also Framework to read in Data and to test Quality

Every Algorithm had its own Framework

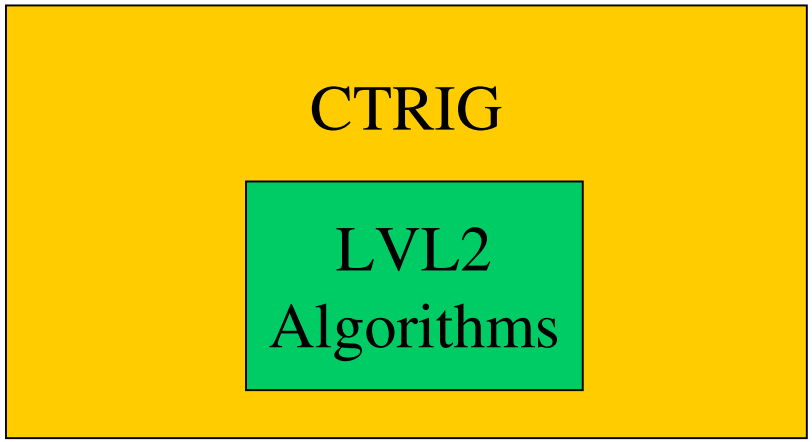
ATRIG

- So far: RoI-guided high-p_T Trigger Performance Studies
- Algorithms return neither Kine nor Digits

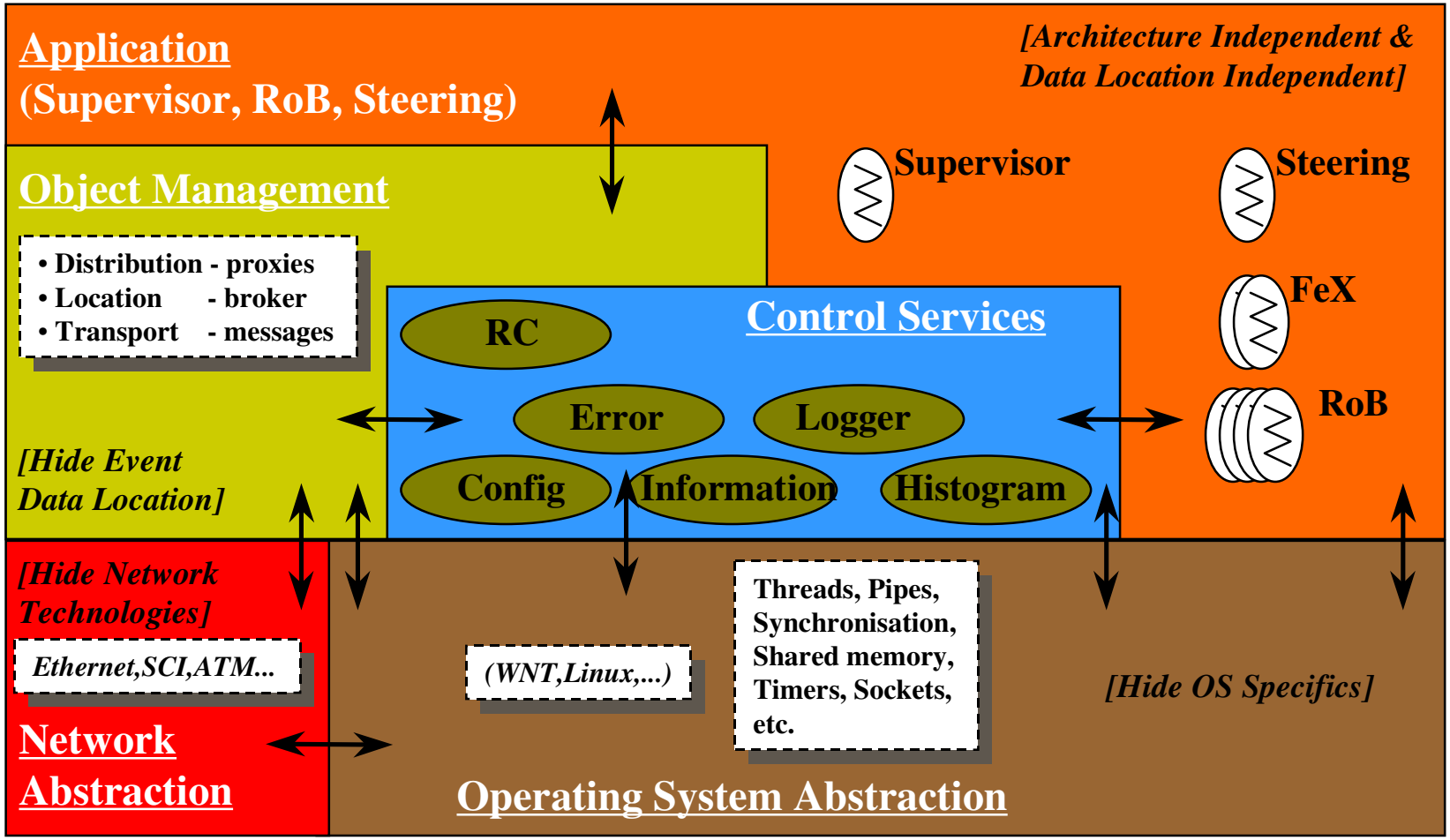


CTRIG

- RoI-guided and non-RoI-guided (B-physics) -> Kine correlation needed
- Algorithms return Kine information



Reference Software



TRIG++

- RoI-guided and non-RoI-guided -> Kine or Digit
- Interface to Event Display -> Digit
- Combined Tracker Fit -> Digit
- Benchmarking -> neiter Kine nor Digit

-> Algorithms return Digits of Tracks on Demand

Algorithm / Framework Interfaces

- ATRIG: Algorithms return Tracks always without Kine/Digit information
- CTRIG: Algorithms return Tracks always with Kine information
- TRIG++: Algorithms return Tracks with optional Digit information
- Ref. Soft.: Algorithms return Tracks with optional Digit information

Summary

- Why optional Digit info?
  - ➜Event Display: Digit info needed
  - ➜Kine Association: Digit or Kine info needed
  - ➜Combined Tracker Fit: Digit info needed
  - ➜Inside Tracking Detector: Digit info needed for Fit anyway
  - ➜Benchmarking: neither Digit nor Kine

-> optional Digit info solves all Tasks