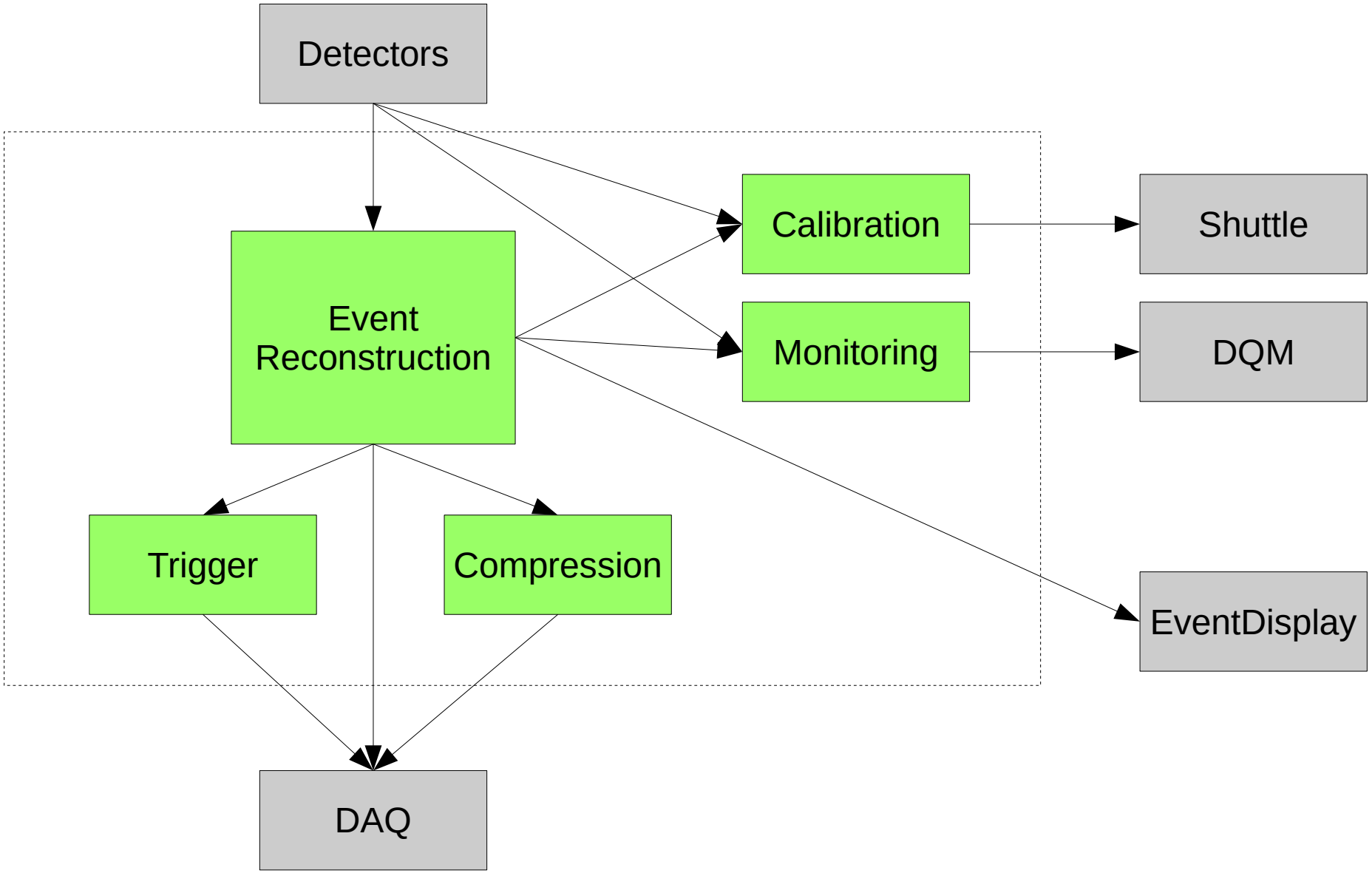
ALICE HLT Tutorial: Overview

Timo Breitner, University of Frankfurt

November 5, 2013

HLT Core Functions



HLT Capabilities

- Input:
  - Access to raw data, partial and full (ESD) reconstructed data on all levels
  - Access to “online” OCDB (HCDB)
  - Access to DCS data
- Output:
  - Raw event stream (DAQ)
  - Shuttle (OCDB)
  - DQM
  - Event Display

Existing HLT Components

- Event reconstruction:
  - Active: TPC tracking, ITS-TPC tracking, Vertexing
  - Lots of other detectors currently unmaintained
- Compression:
  - TPC clusterization + Huffman
- Trigger:
  - TPC filter (for compression)
  - Many examples/prototypes exist, never fully commissioned
- Calibration:
  - HLT internal
  - Prototypes
- Monitoring:
  - Active: TPC Cluster compression, tracking
  - Lots of others unmaintained (Multiplicity, Correlations)

Before you get started...

- What input do you need?
  - Don't just expect a filled ESD!
  - Do you need a certain detector to be reconstructed? Is there a reco path in HLT?
- Think about calibration!
  - HCDB != OCDB
  - How recent do OCDB objects have to be?
  - How good does the calibration have to be?
  - Do you need online calibration (e.g. TPC drift time)?

A Word on Triggers...

- Why triggering?
  - Physics case?
  - Benefit?
    - Alice is typically dead-time limited
    - Small benefit in smaller storage needs (apparently not a problem...)
    - Faster reconstruction of rare/interesting events
- Alternative: Tagging
  - Rare events are tagged and put in separate stream for fast reconstruction
  - Proposed several times, never fully developed
  - Only small changes in HLT framework needed, but affects core parts of DAQ and Offline

General Requirements

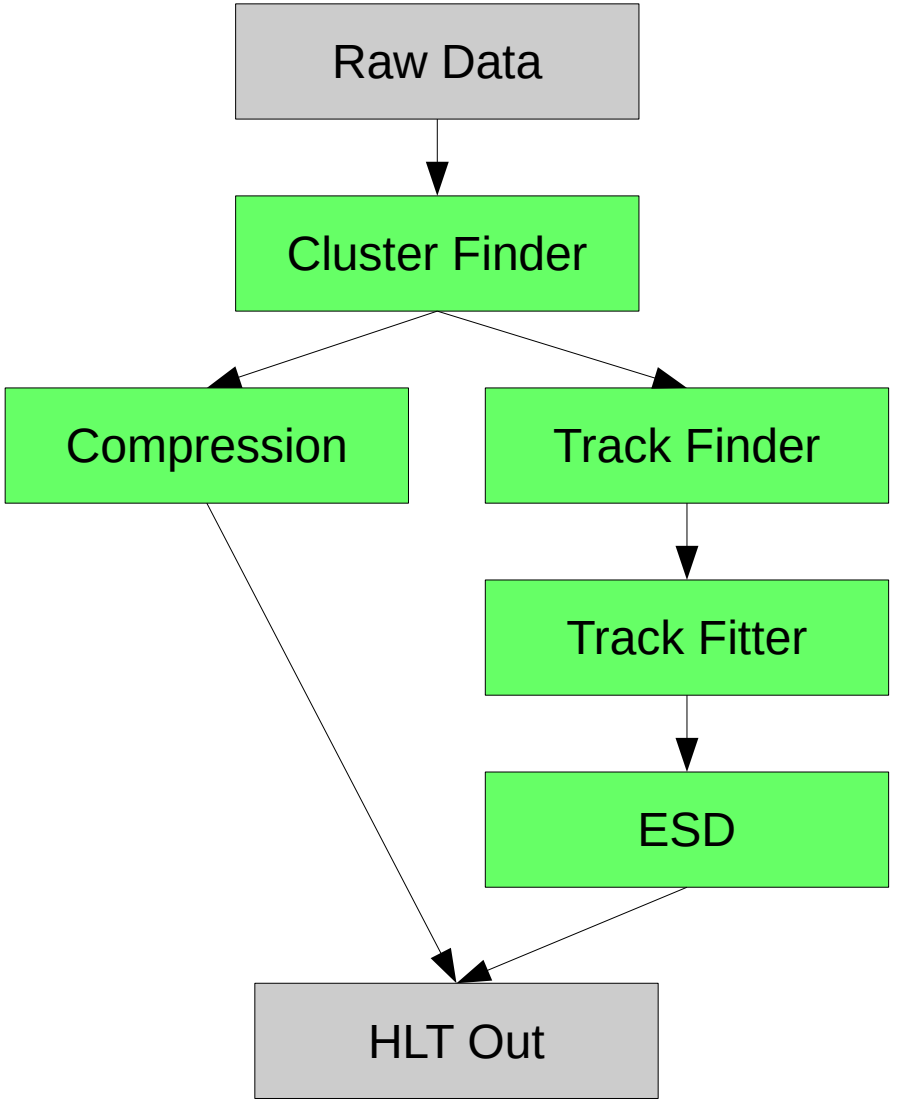
- Stability
  - We can't afford run failures due to crashing processes (e.g. segmentation faults, memory leaks)
  - Test your components thoroughly, use appropriate tools (e.g. valgrind)
  - Always expect corrupt input data, make consistency checks
  - HLT follows a “zero tolerance” policy

General Requirements

- Resource consumption
  - Do not use excessive amounts of memory and/or CPU time
  - Try to optimize performance, use tools (profilers)
  - Use modern techniques (e.g. vectorization, see next tutorial about Vc package of Matthias Kretz)
  - Try to avoid ROOT objects as exchange format

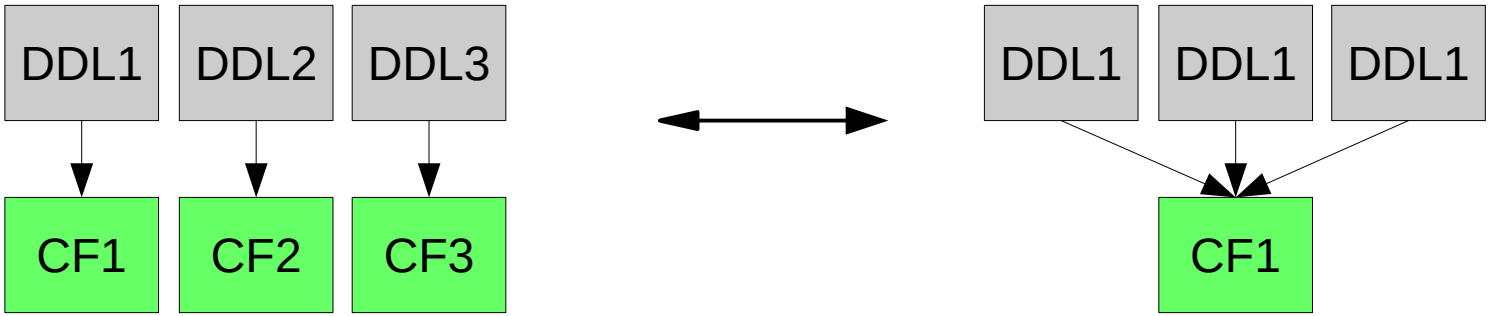
HLT Component Model

- Reconstruction is split into independent modules, called “components”
- Encapsulates reconstruction step
- Well-defined input and output data format
- Each running in a separate process (online case)



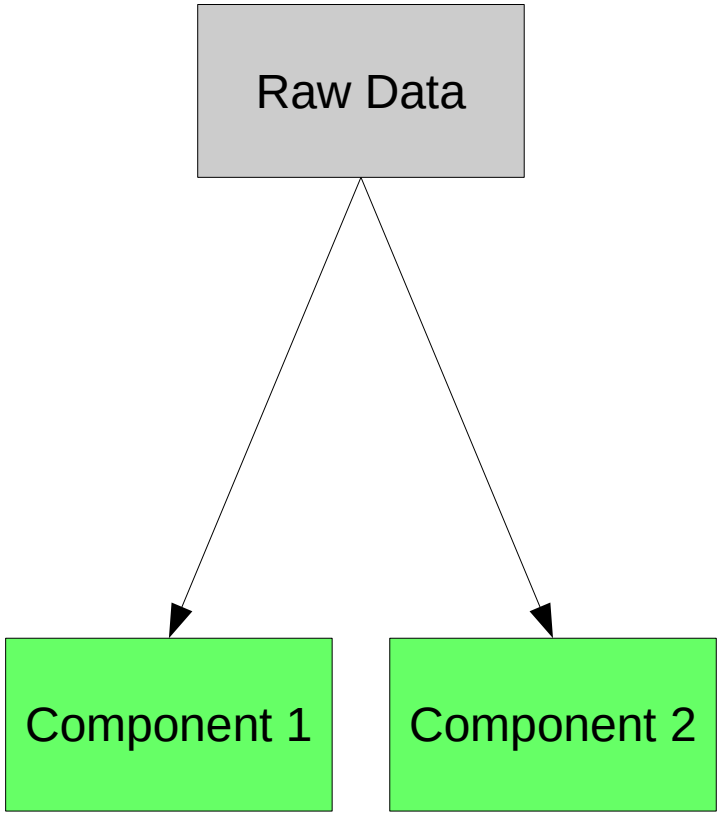
HLT Component Model

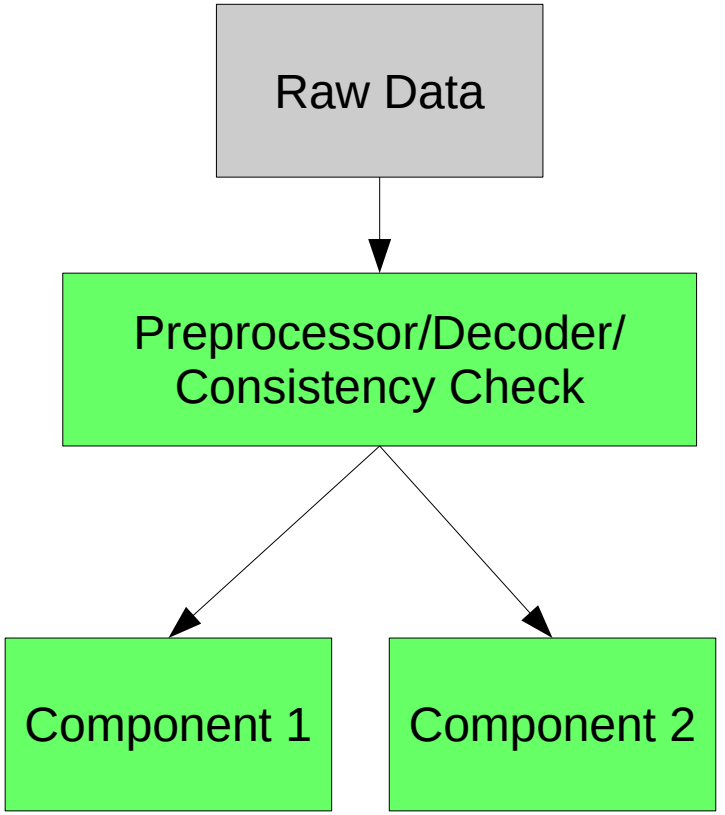
- Components should be able to handle various number of input blocks
- Allows flexible process placement



HLT Component Model

- Put common functionality into dedicated component





Getting started

- Download tutorial.tbz

- `cd $ALICE_ROOT/HLT`
- `tar xvf ~/tutorial.tbz`
- `touch CMakeLists.txt`
- `cd $ALICE_ROOT/build`
- `make install`