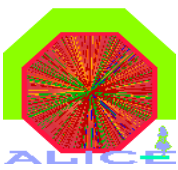

High Level Trigger

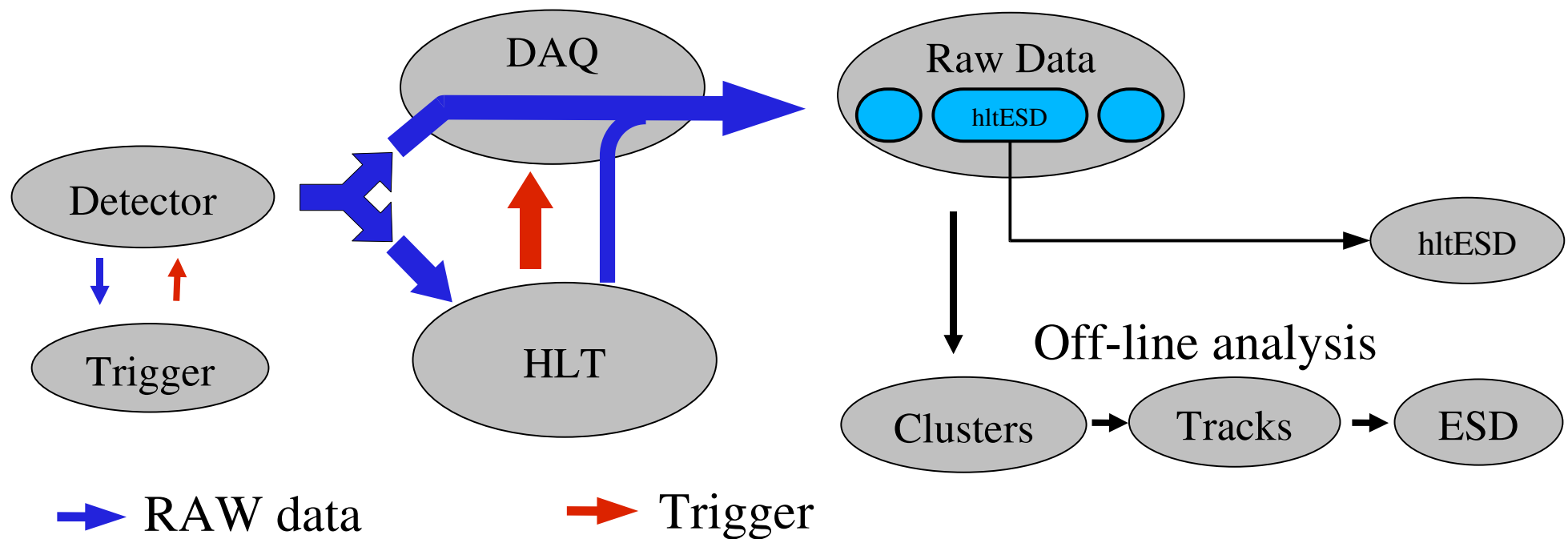
@ AliRoot simulation/reconstruction

Recent changes in AliRoot sim/rec



- AliReconstruction: common event loop
 - removed from the detectors code
- Removal of run-loaders from the detector reconstruction code
 - Processing of simulated data not longer possible in AliReconstructor
- AliSimulation: HLT simulation added
- AliHLTSystem:
 - Encapsulates HLT reconstruction for AliSimulation/Reconstruction
 - Added definition of HLT configurations via macros
 - Added analysis chain customization

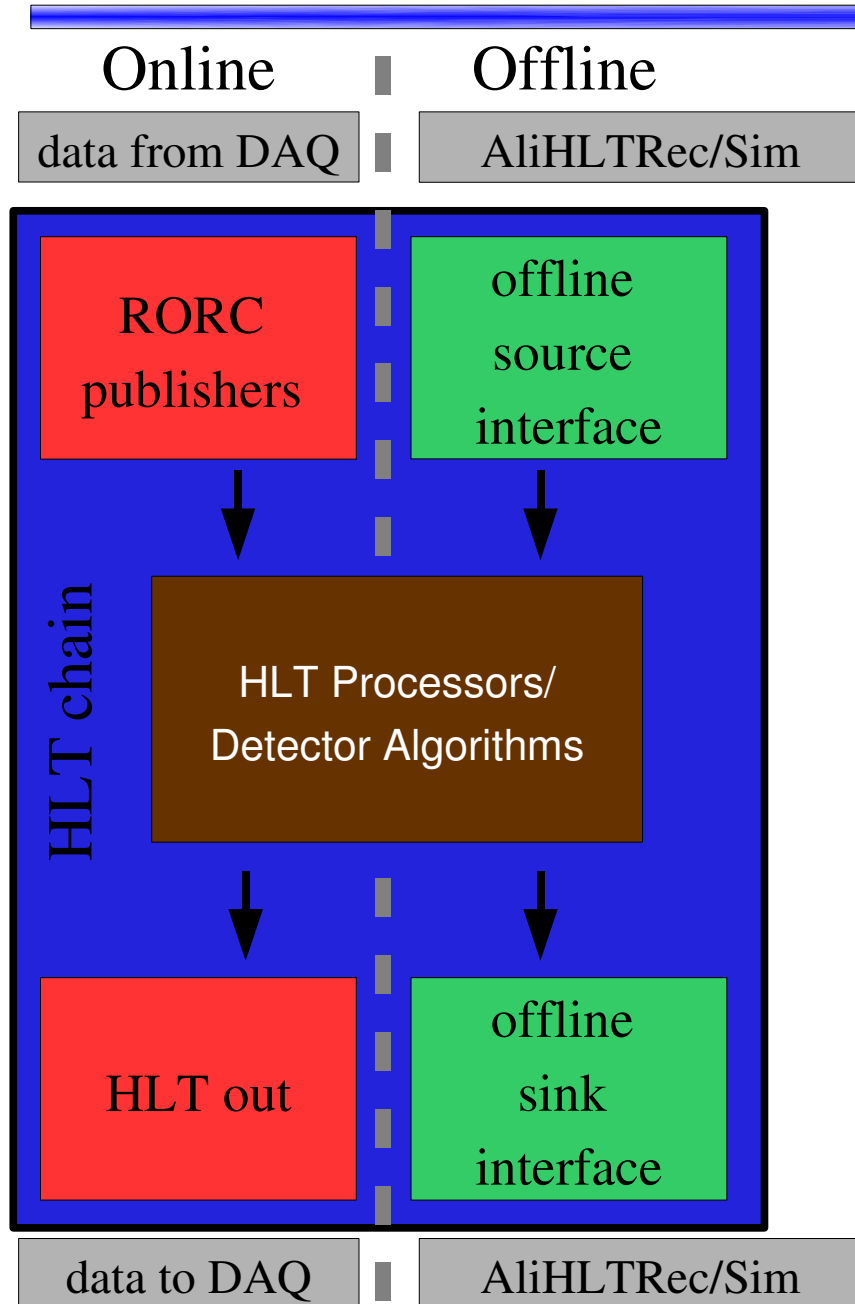
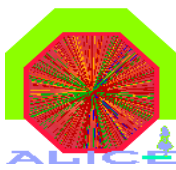
HLT data processing flow



HLT produces:

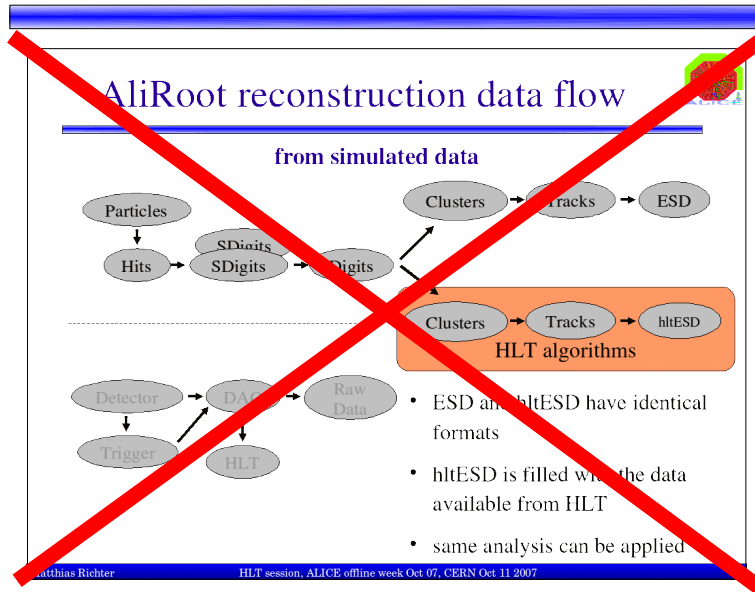
- Compressed data
- hltESD is calculated on-line on the HLT cluster and shipped together with the RAW data stream

How to run HLT in AliRoot?



- Completely identical HLT processing can run both in the online and offline framework
- HLT reconstruction in AliRoot encapsulated in AliHLTSystem
 - Stand-alone
 - Simulated data
 - Raw data (simulated or real)
- Offline Source/Sink components provide data input/output

HLT simulation data flow

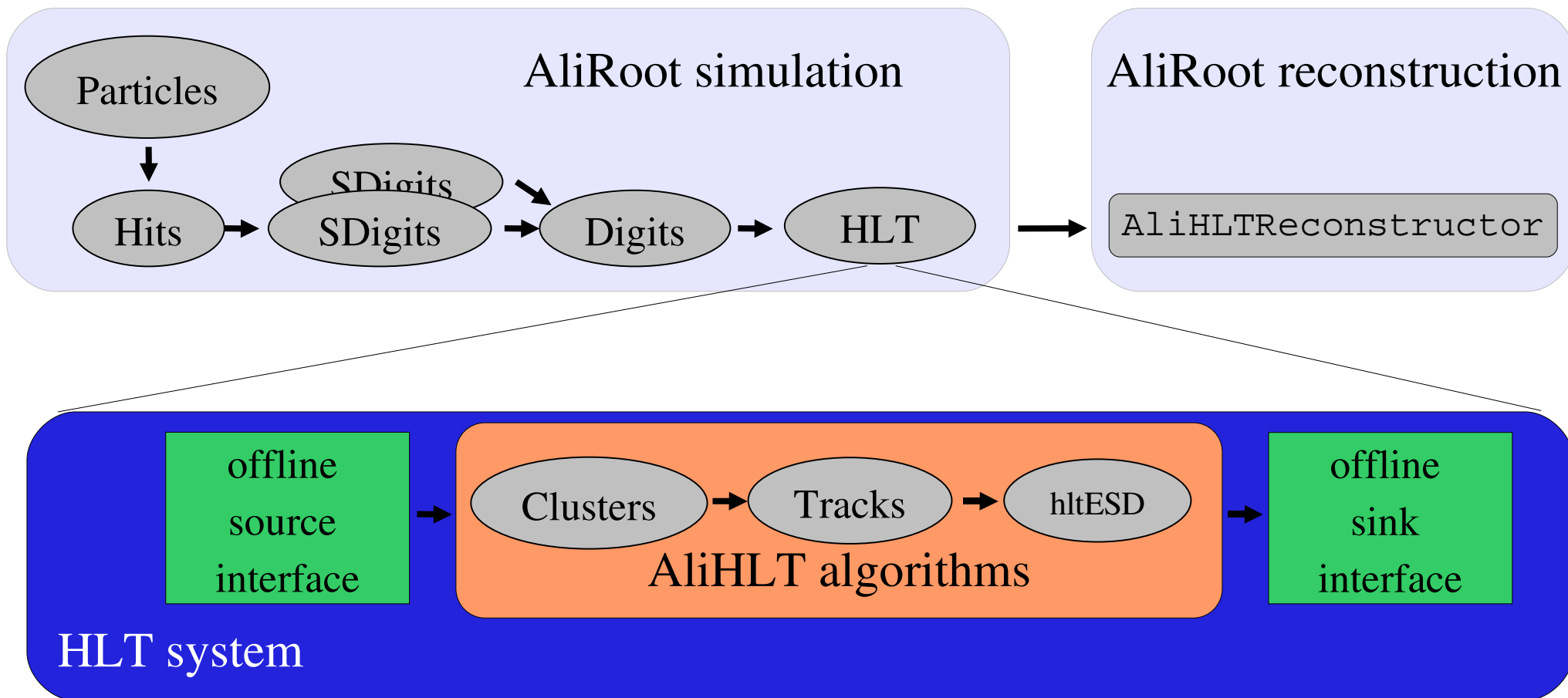
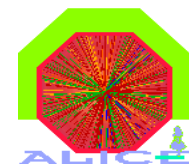


Old scheme as presented in several meetings before is deprecated!

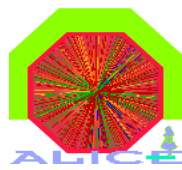


- Running at the end of the AliRoot simulation
- HLT simulation = HLT reconstruction on simulated data
- Same output as the real HLT

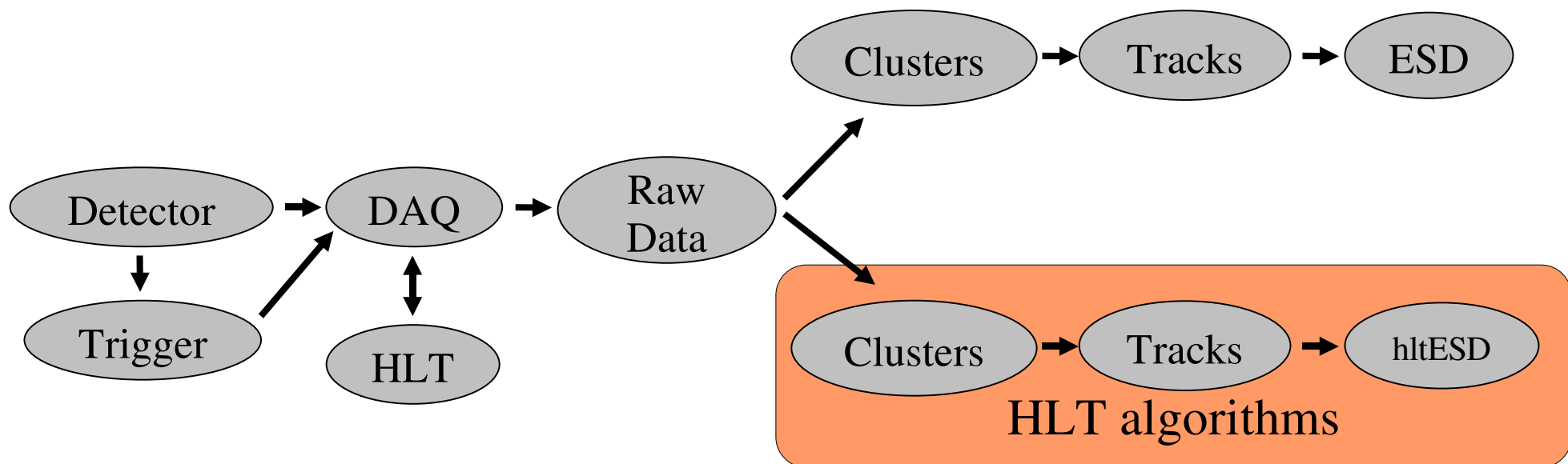
HLT simulation data flow



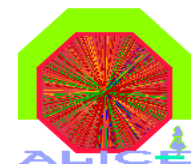
HLT raw data processing in AliRoot



- Default operation of AliHLTRecreconstruction extracts HLT output (real or simulation)
- Development/debugging feature
 - Reconstruction in parallel to the offline reconstruction

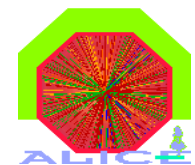


HLT output (AliRoot)



- Simulation
 - Concatenated data blocks in the HOMER format
 - Under development: common offline sink component generating the HLT output like the HLTOUT nodes
- Reconstruction
 - ESD files
 - ESDs merged into hltESD provided by AliReconstruction (under development)

HLT Reconstruction interface



- AliHLTSystem

- Loads the HLT component libraries
- Module agents define chains to run
- Identical for AliRoot Simulation and reconstruction

- Options

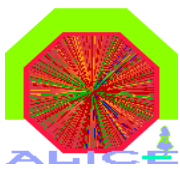
<code>config=conf-macro.C</code>	configurations defined by macro instead of agents
<code>chains=chain1,chain2</code>	custom chains
<code>loglevel=0x<level></code>	HLT loglevel (0x3f all, 0x3c info and higher)
<code>libAliHLT<...>.so</code>	load component library instead default libs

- http://web.ift.uib.no/~kjeks/doc/alice-hlt/group__alihlt__tutorial.html

Library agents

- Implemented by each module to provide properties of the module
- Module agent: `AliHLTModuleAgent`
 - `RegisterComponents`: define and register components
 - `CreateConfigurations`: define tasks
 - Component
 - Component arguments
 - Data sources (parent publishers)
 - `GetReconstructionChains`: chains to run
 - AliRoot simulation or event reconstruction
 - Chains can be defined depending on the availability of the `AliRunLoader` or `AliRawReader`

HLT @ AliSimulation



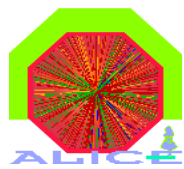
- AliHLTSimulation
 - Last step of AliRoot simulation
 - Loads the HLT component libraries
 - Module agents define configurations and chains to run
 - Custom configurations and chains possible

- Example (processing of already simulated data)

```
AliReconstruction sim;           // the simulation instance
sim.SetRunGeneration(kFALSE);    // do not generate particles
sim.SetMakeDigits("");          // disable
sim.SetMakeSDigits("");         // disable
sim.SetMakeDigitsFromHits("");  // disable
sim.SetMakeTrigger("");
sim.SetRunHLT("libAliHLTPC.so  config=conf-tpc.C  chains=sink1");
sim.Run();
```

- HLT Simulation = HLT Reconstruction of simulated data

HLT Simulation: components



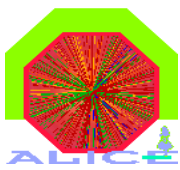
INPUT

- Detector specific offline source component(s) convert(s) digits to native input format of the first component in the chain
 - e.g. TPCDigitPublisher
- Common component of the framework can publish digit tree:
 - AliHLTLoaderPublisherComponent

OUTPUT

- Common component (underdevelopment) of the framework takes all incoming data blocks, adds HOMER descriptor and stores it in the appropriate place of the RunLoader
 - Identical to HLTOUT nodes
- Detector specific offline sink component(s) add the data to some tree in the AliRunLoader

Custom HLT system configuration



```
{// conf-tpc.C
int iMinSlice=0;
int iMaxSlice=35;
int iMinPart=0;
int iMaxPart=5;
TString writerInput;
for (int slice=iMinSlice; slice<=iMaxSlice; slice++) {
  TString trackerIn;
  for (int part=iMinPart; part<=iMaxPart; part++) {
    TString arg, pub, cf;

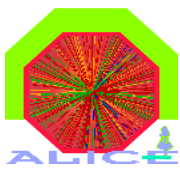
    // digit publisher components
    arg.Form("-slice %d -partition %d", slice, part);
    pub.Form("DP_%02d_%d", slice, part);
    AliHLTConfiguration pubconf(pub.Data(), "TPCDigitPublisher", NULL , arg.Data());

    // cluster finder components
    cf.Form("CF_%02d_%d", slice, part);
    AliHLTConfiguration cfconf(cf.Data(), "TPCClusterFinderUnpacked", pub.Data(), "pp-run timebins 446");
    if (trackerIn.Length(>0) trackerInput+=" ";
    trackerIn+=cf;
  }
  TString tracker;
  // tracker finder components
  tracker.Form("TR_%02d", slice);
  AliHLTConfiguration trconf(tracker.Data(), "TPCSliceTracker", trackerIn.Data(), "pp-run bfield 0.5");

  if (writerInput.Length(>0) writerInput+=" ";
  writerInput+=tracker;
}

// the esd writer configuration
AliHLTConfiguration esdwconf("esd-writer", "TPCEsdWriter" , writerInput.Data(), "-datafile AliESDs.root")
}
```

HLT @ AliReconstruction



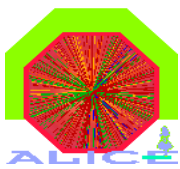
- AliHLTReconstructor
 - Loads the HLT component libraries
 - Module agents define chains to run
 - Chains run during FilleSD to avoid intermediate data storage

- Example

```
AliReconstruction rec;           // the reconstruction instance
rec.SetRunTracking("");         // switch off tracking
rec.SetFilleSD("HLT");          // run rec only for HLT
rec.SetOption("HLT", "libAliHLTSample.so");
rec.Run();
```

- Reconstruction of RAW data

HLT Reconstruction: components



INPUT

- Common component of the framework publishes data from RawReader:
 - AliHLTRawReaderPublisherComponent
 - Identical to real input from DDL and RORC publisher

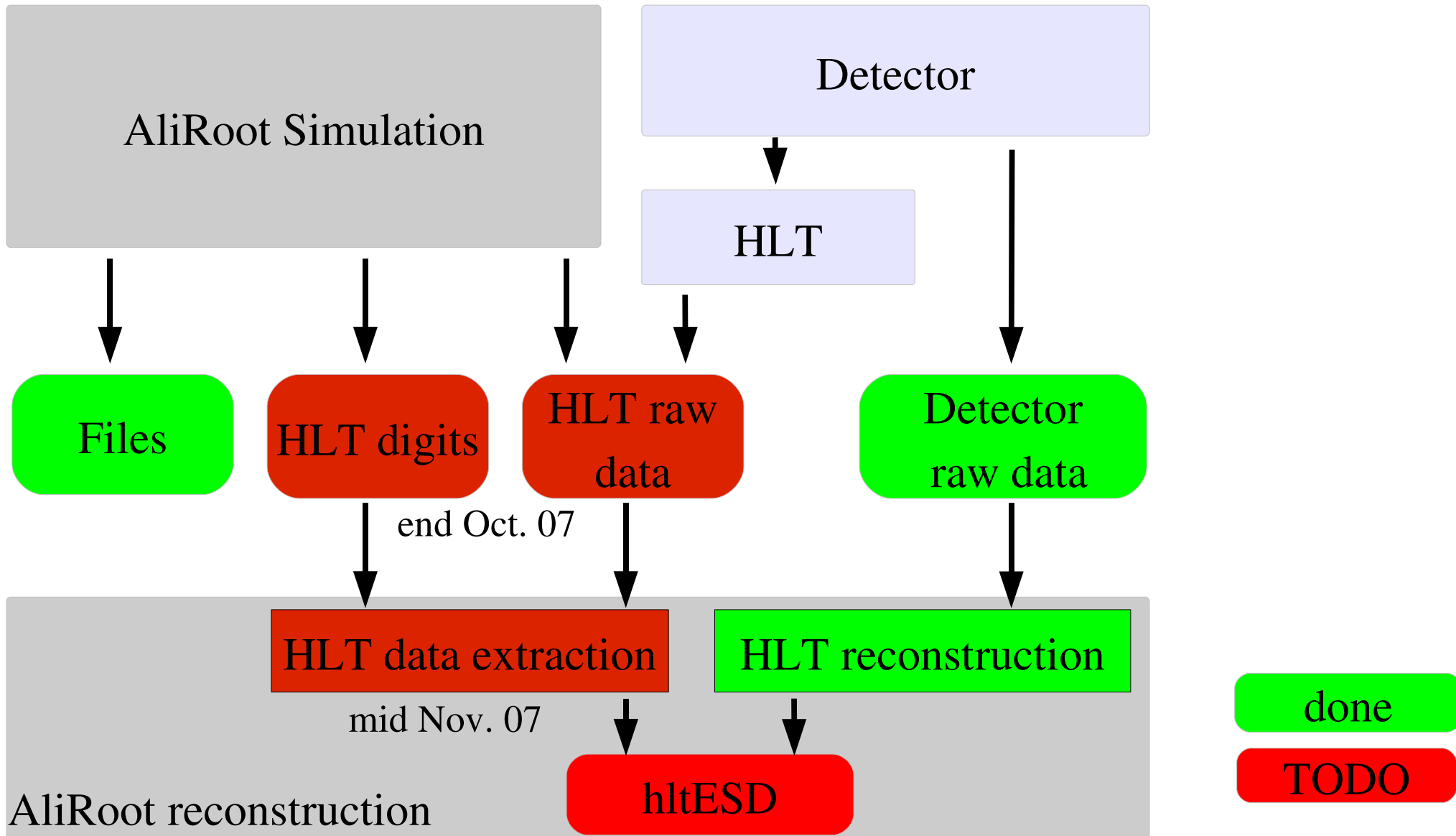
PROCESSING

- HLT components/chains identical to real HLT

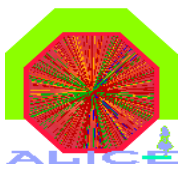
OUTPUT

- ESD Merger writes data back to hltESD of AliReconstruction
- ESD file writers

Roadmap



TODO



-
- Storage of data from HLT simulation -> HLT.Digits.root (Oct 2007)
 - AliHLTLoader to run properly in AliReconstruction (Oct 2007)
 - AliHLTReconstructor framework
 - Unpacking/handling of detector data
 - Handling of hltESDs (Nov 2007)
 - RawStreams for compressed data (Nov 2007)
 - Implementation of detector specific source and sink components
 - Definition of configurations to run
 - Definitions of chains to run
 - Framework is ready for that development