CBM Perspectives
for Online and Offline Computing

Volker Friese
GSI Darmstadt

Two Different Worlds?
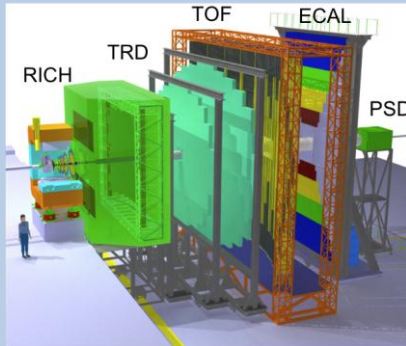
Online

From experiment to data storage

Offline

From data storage to end user analysis

Simulations

Paradigma

There shall be no separate „Online World" and „Offline World"

We want a common software environment for both.

- Any physics algorithm must be able to run on any offline / online available hardware producing the exactly same results (rucial for sstema t i cs etc...)

N. Neufeld, LHCb

Example: Offline Data Model

„Traditional" scheme

TIER0   TIER1   TIER2

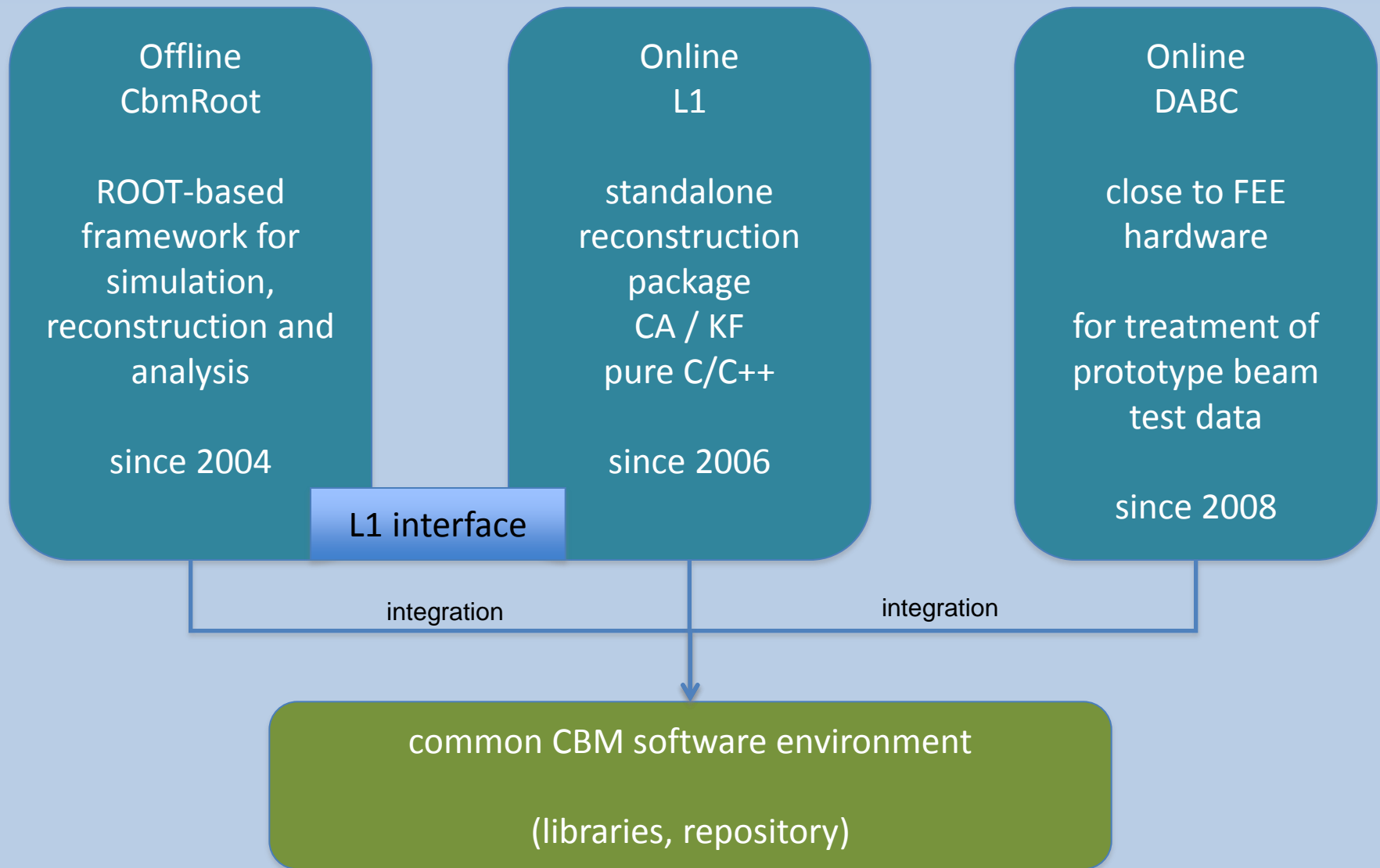| **RAW data**	→ reconstruction →	**ESD**	→ reduction →	**AOD**

CBM conditions:

- reconstruction is FAST – also offline

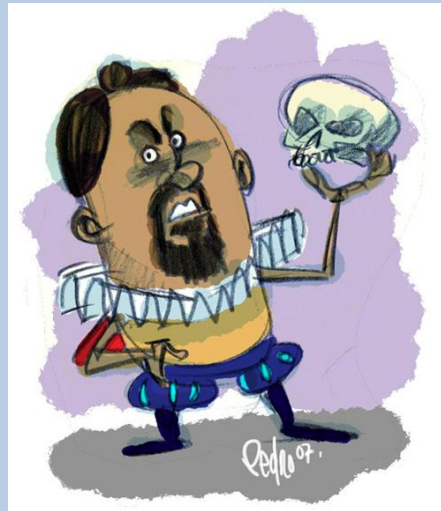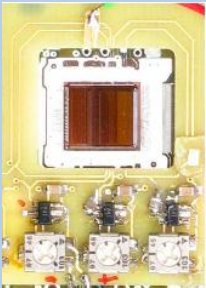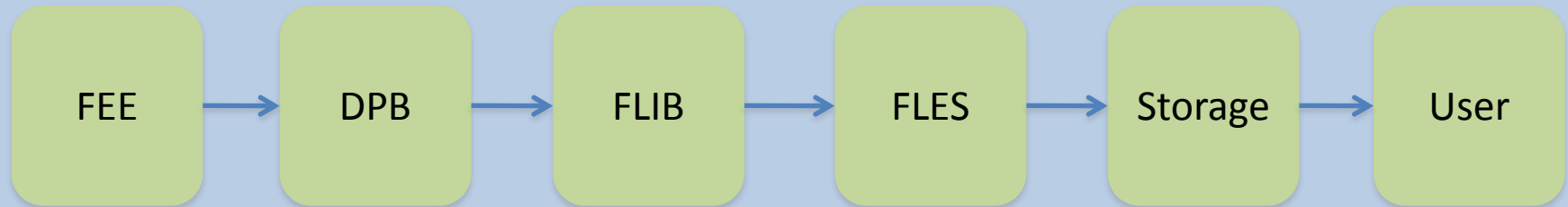- there is no large data compression from RAW to ESD

There is thus no need to store ESD – direct analysis or creation of AOD (for distributed analysis) will run reconstruction on the fly.

The online reco algorithms must thus be used offline – with refined calibrations and possibly afterburners for precision and additional track classes.

Reality (Development Phase)

Offline CbmRoot	Online L1	Online DABC
ROOT-based framework for simulation, reconstruction and analysis	standalone reconstruction package CA / KF pure C/C++	close to FEE hardware for treatment of prototype beam test data
since 2004	since 2006	since 2008

L1 interface

integration   integration

common CBM software environment

(libraries, repository)

To ROOT or Not To ROOT

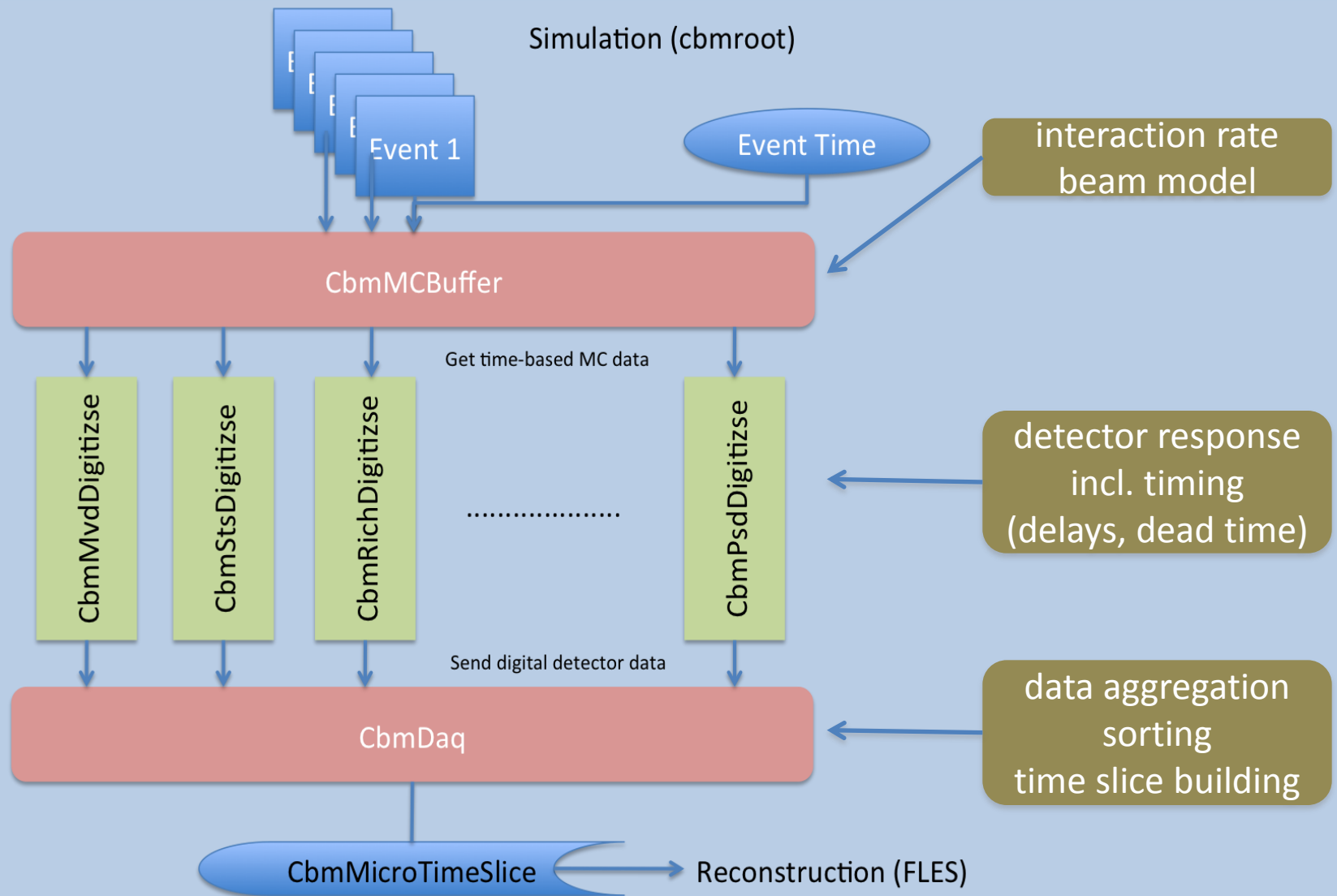| FEE	→	DPB	→	FLIB	→	FLES	→	Storage	→	User

RAW data model

- reconstruction avoids ROOT-type data
  - simple C++ classes / C structs
  - internal conversion when fed from CbmRoot
  - parallelisation with ROOT not trivial
- severe overhead if atomic data (detector messages) are TObjects
- but after reconstruction, the selected raw data must be made persistent
  - persistency model?
  - of course, everything can be wrapped into a TObject when streaming to file

Free Streaming Data
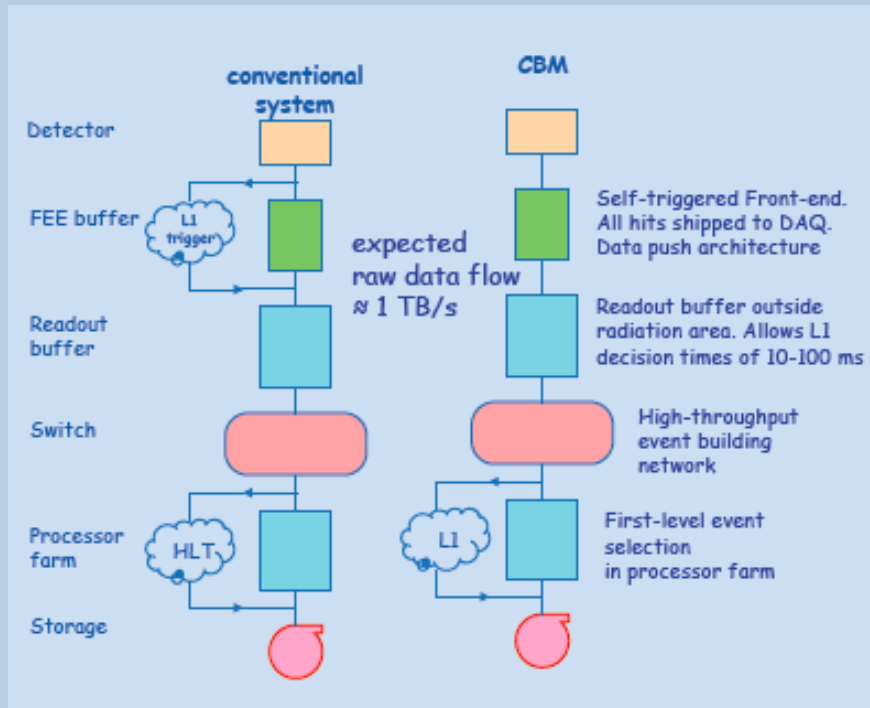
- No hardware trigger: the basic data container delivered to FLES represents a time interval, not an „event".

- For development of reconstruction, proper input must be provided from simulations.

- Modelling the data flow in CbmRoot is not trivial: simulations always start from events.

- Events may overlap in time.

- Is a ROOT TTree the proper model for this data?
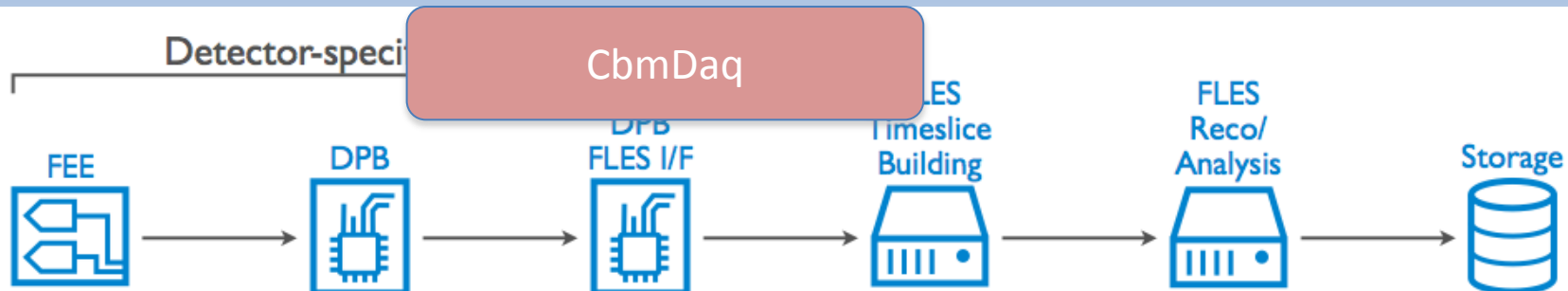
Time-Based Simulations

Simulation (cbmroot)

Event 1

Event Time

interaction rate
beam model

CbmMCBuffer

Get time-based MC data

CbmMvdDigitizse

CbmStsDigitizse

CbmRichDigitizse

....

CbmPsdDigitizse

detector response
incl. timing
(delays, dead time)

Send digital detector data

CbmDaq

data aggregation
sorting
time slice building

CbmMicroTimeSlice

Reconstruction (FLES)

Status: Time-Based Simulation

- digis (detector messages) are still TObjects
  - will move to real data format once specified from FEE / DAQ / network protocol
- buffers: TClonesArrays not well suitable; use STL for the time being
- persistency: on the container level (time slice)

Provides correct input for reconstruction on the logical level (not yet on the formal level)
- development of time-based reconstruction can continue
- future changes interfaces, not the logic

Simulation and Reality

CbmDigi

CbmStsDigi, CbmMuchDigi,...

CbmTimeSlice

CbmGlobalTrack

etc..

Instead of a Summary

2. "Revolutionary": r -think everything

„Begin at the beginning", the King said gravely, „then go on till you come to the end; then stop".

Alice in Wonderland