

# Online meeting 02.07.26

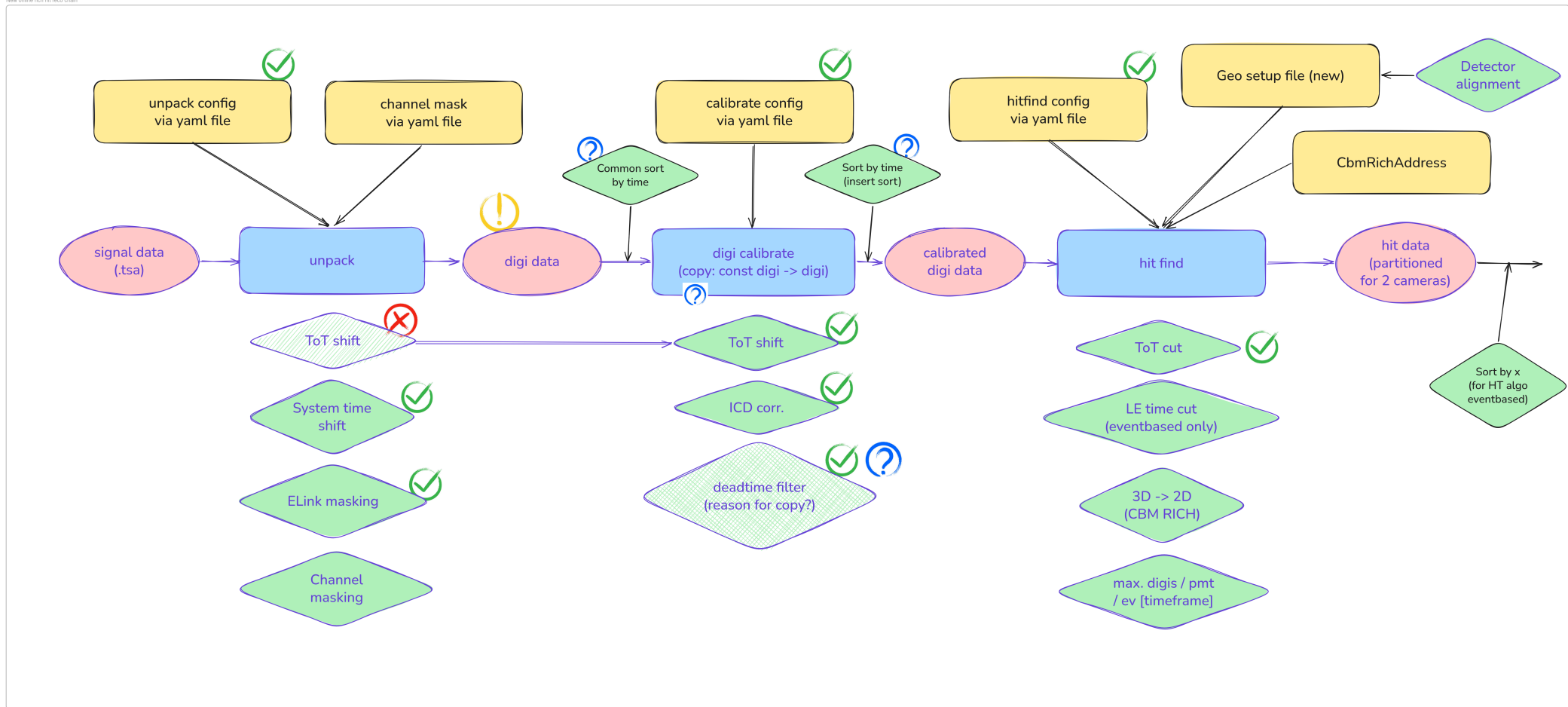
Martin Beyer

Status: online RICH Hit reconstruction



# algo RICH Hit reco chain (not final)

new online rich hit reco chain



# Questions

Note: follows the chain of Bmon/Tof

- Copy of digi input for calibrate step needed?
  - Deadtime filter as the reason?  
Unsure if really needed for RICH -> will be checked
- Sorting in time:
  - After unpack, common for all detector systems
  - After digi calibrate, since timestamps were modified  
-> Why sort twice? (skip first sort?)

Sorting in general required for digi event building?

(also very beneficial to count digis / PMT / timeframe )

# online qa

- How to run online reco qa **without** a histogram server?  
(currently did some implementation locally and converting the output via `cbm::qa::OnlineInterface::ConvertOutput`)

# CbmRichDigi class size reduction #4063

Currently:

Member	Type	Offset	Size
fAddress	int32_t	0	4
(padding)	-	4	4
fTime	double	8	8
fToT	double	16	8

Total 24 Bytes

New (fToT [double -> float] + reorder):

Member	Type	Offset	Size
fAddress	int32_t	0	4
fToT	float	4	4
fTime	double	8	8
--			

Total 16 Bytes -- 33% reduction

- Backwards compatibility looking good so far (both boost/ROOT streamer)
- Only problem if (double) fToT is out-of-range of float range? (more?)
  - For simulated data is it always 0.0
  - For unpacked and stored mCBM data it should be in range [-20.0, 120.0], also mRICH data rarely used -> Check not needed in my opinion

# boost/ROOT streamer changes

```
// New layout: fToT is float and reordered for tight packing.
class CbmRichDigi {
    int32_t fAddress;
    float fToT;
    double fTime;

    template<class Archive>
    void serialize(Archive& ar, const unsigned int version)
    {
        ar & fAddress;
        if (version < 1) {
            // Legacy stream (class version 0): fTime before fToT, ToT stored as double.
            double tot = 0.;
            ar & fTime;
            ar & tot;
            fToT = static_cast<float>(tot);
        }
        else {
            ar & fToT; // float since class version 1
            ar & fTime;
        }
    }
};

#ifdef NO_ROOT
// changed 3 -> 4
// ROOT matches members by name, converting double->float and handles the reordering automatically?
ClassDefNV(CbmRichDigi, 4);
#endif
```