EUDAQ 2.0

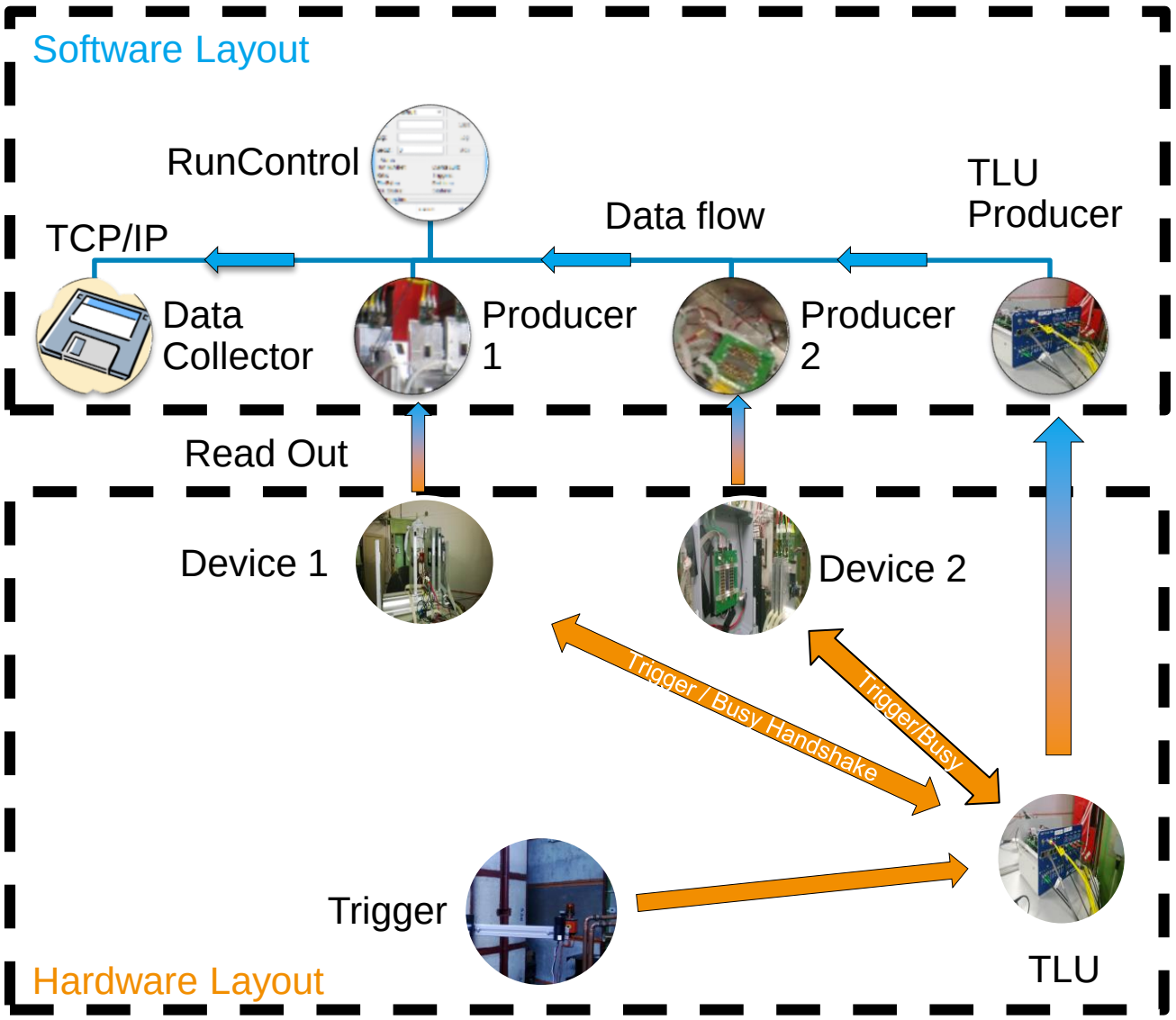
3rd Beam Telescopes and Testbeam Workshop 2015

Ulf Behrens, Alan Campbell, Francesco Crescioli, David Cussans,
Hendrik Jansen, Moritz Kiehn, Hanno Perrey, Richard Peschke,
Igor Rubinskiy, Simon Spannagel

HELMHOLTZ ASSOCIATION

DESY

EUDAQ

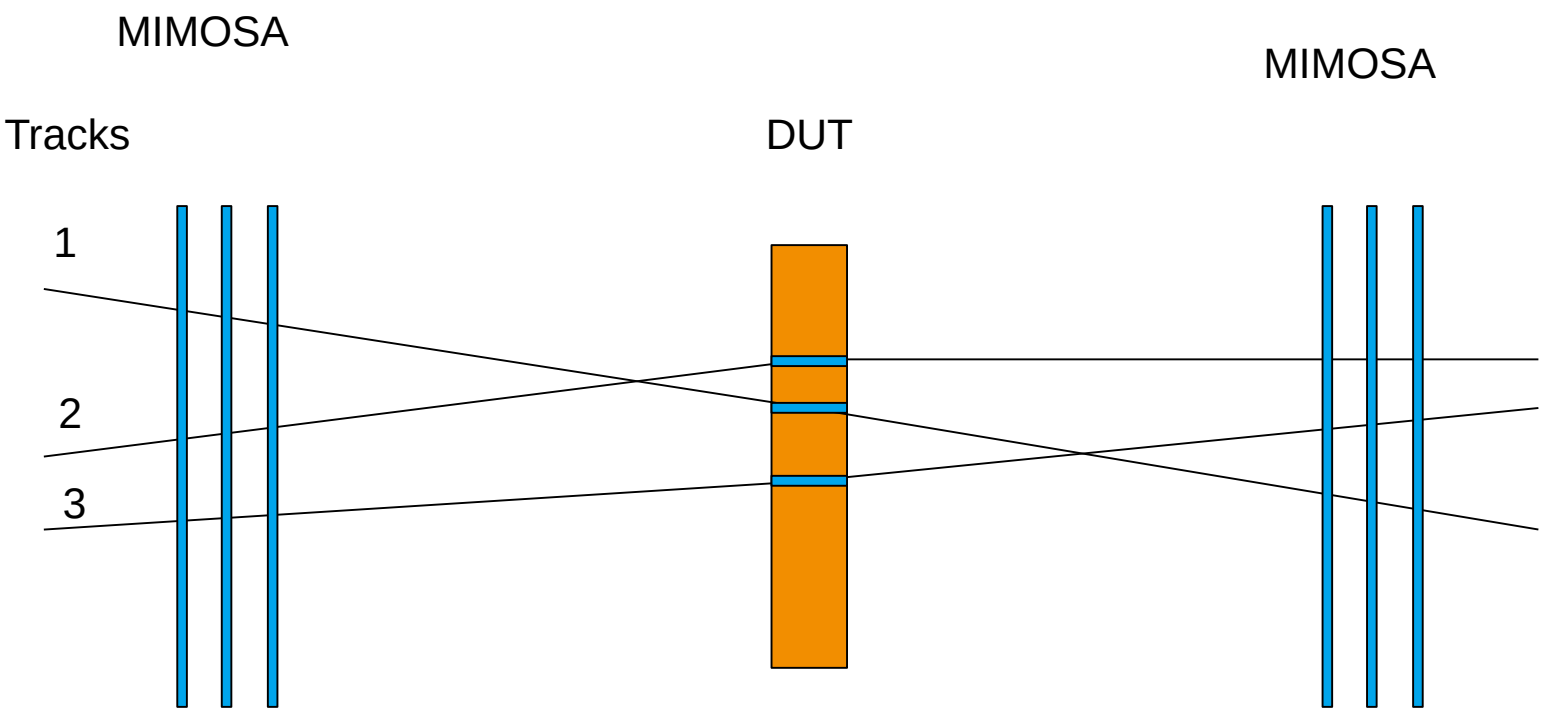


- Distributed DAQ system
  - Consists of completely independent parts / programs such as:
    - Run Control
    - Data Collector
    - Producers
- Synchronous read out
  - All devices get a common trigger
  - Every device raises a busy signal
- DAQ for MIMOSA telescope

Goals

- More flexibility in the data format
  - Possibility to store multiple readout frames in one “packet”
- Increasing of the track rate by more than 2 orders of magnitude
- Easier combination of different kind of devices
  - FE-I4 / Mimosa / Timepix(3) / Slow Control
- Resolved scalability issues
  - Decentralized data taking
  - Data can be stored locally without network overhead
- Stay backward compatible to old EUDAQ 1.x Producer and analyze readout chain (EUTelescope)
  - Only recompiling is needed
  - Changes are only need to benefit from some new features
- Cross platform

Motivation

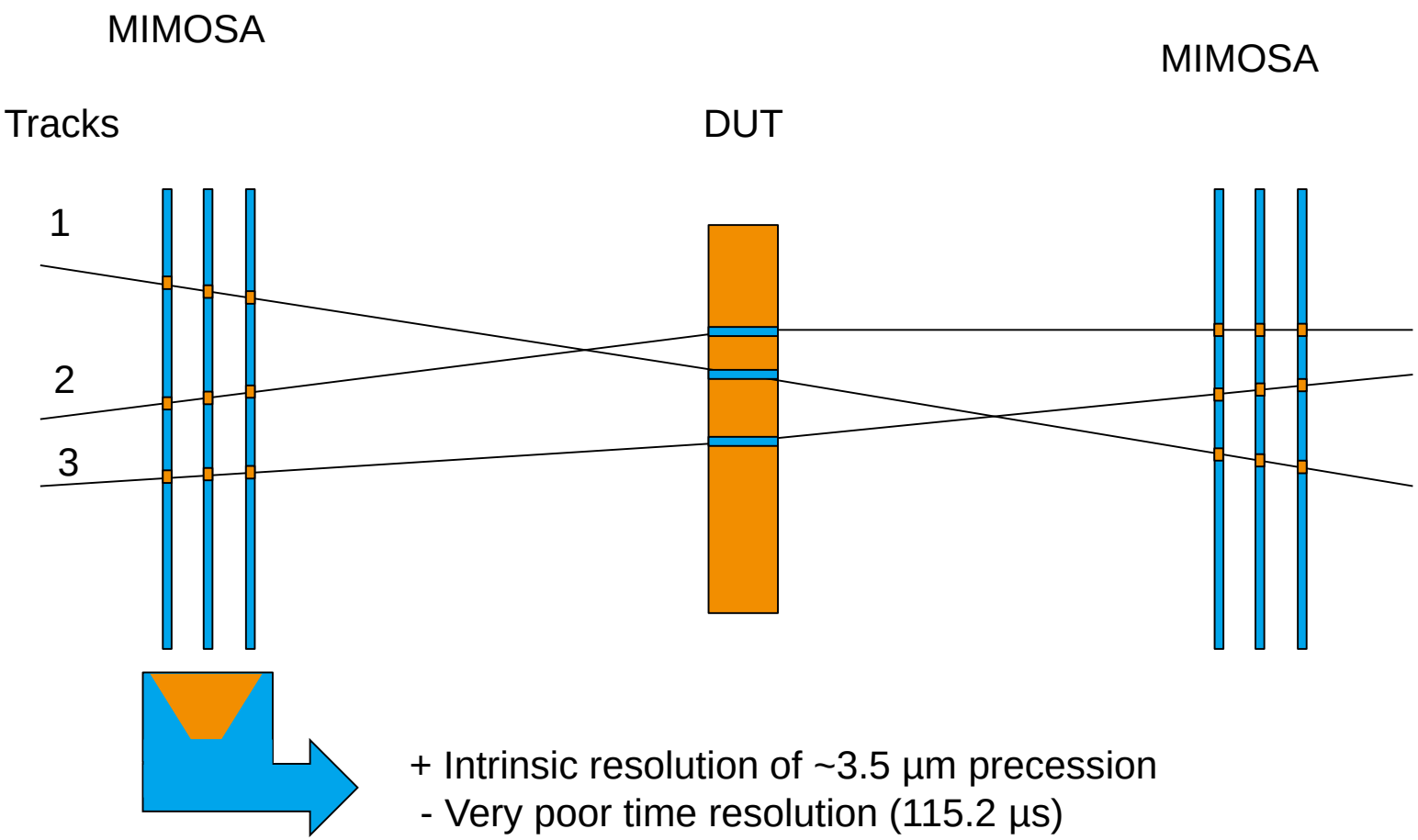


Information Needed:

- Space and time information at the position of the DUT
  - Space resolution < 3 µs
  - Separation time* < 30 ns
  - Time Resolution < 4 ns

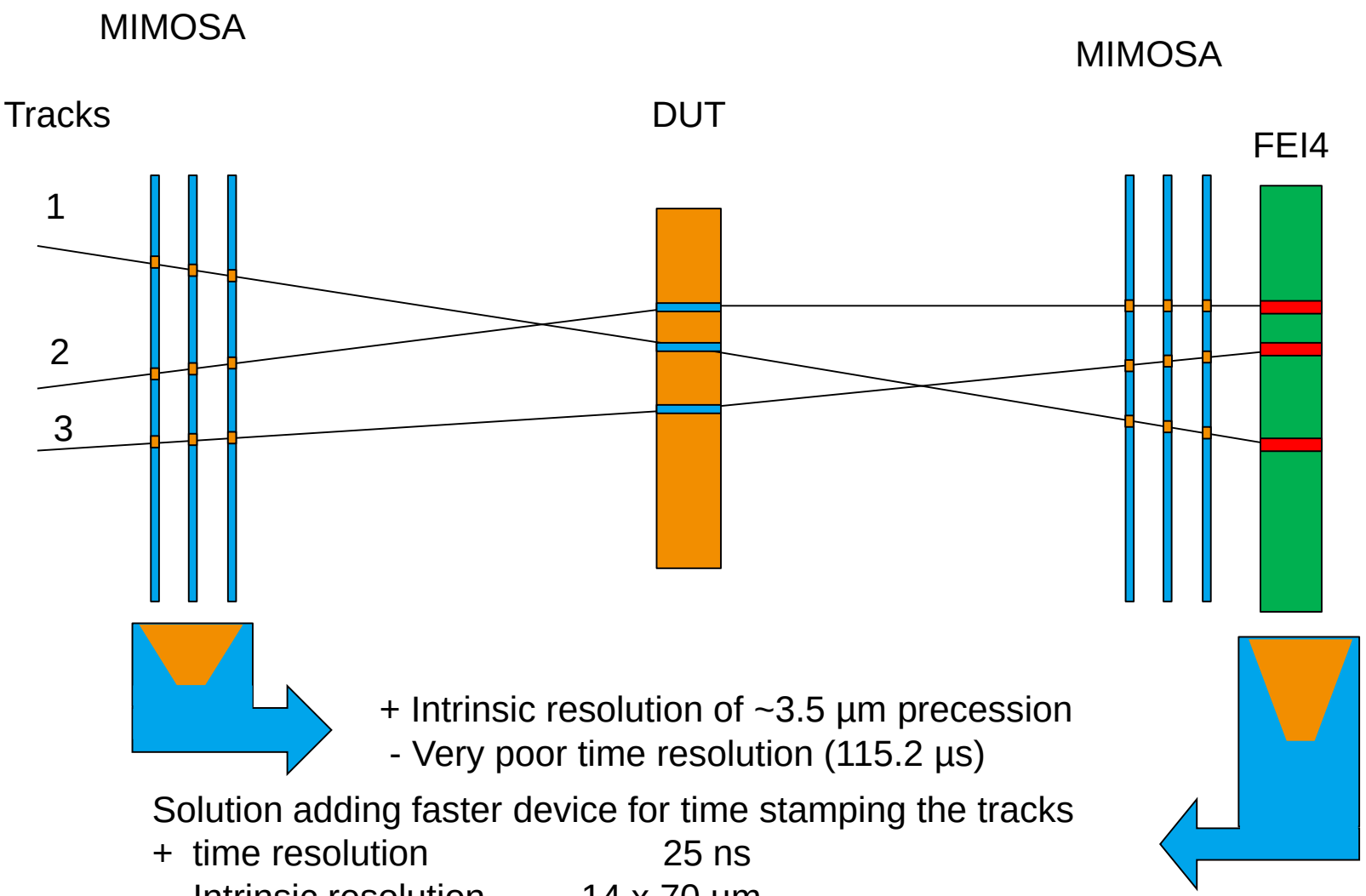
*Tracks can only be separated if their time difference is larger than the separation time

Motivation



+ Intrinsic resolution of ~3.5 µm precession
- Very poor time resolution (115.2 µs)

Motivation



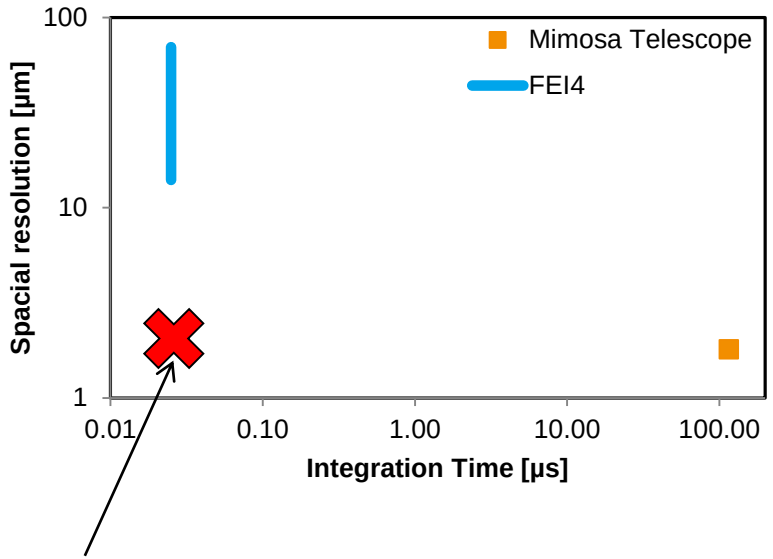
+ Intrinsic resolution of ~3.5 µm precession
- Very poor time resolution (115.2 µs)

Solution adding faster device for time stamping the tracks

+	time resolution	25 ns
-	Intrinsic resolution	14 x 70 µm

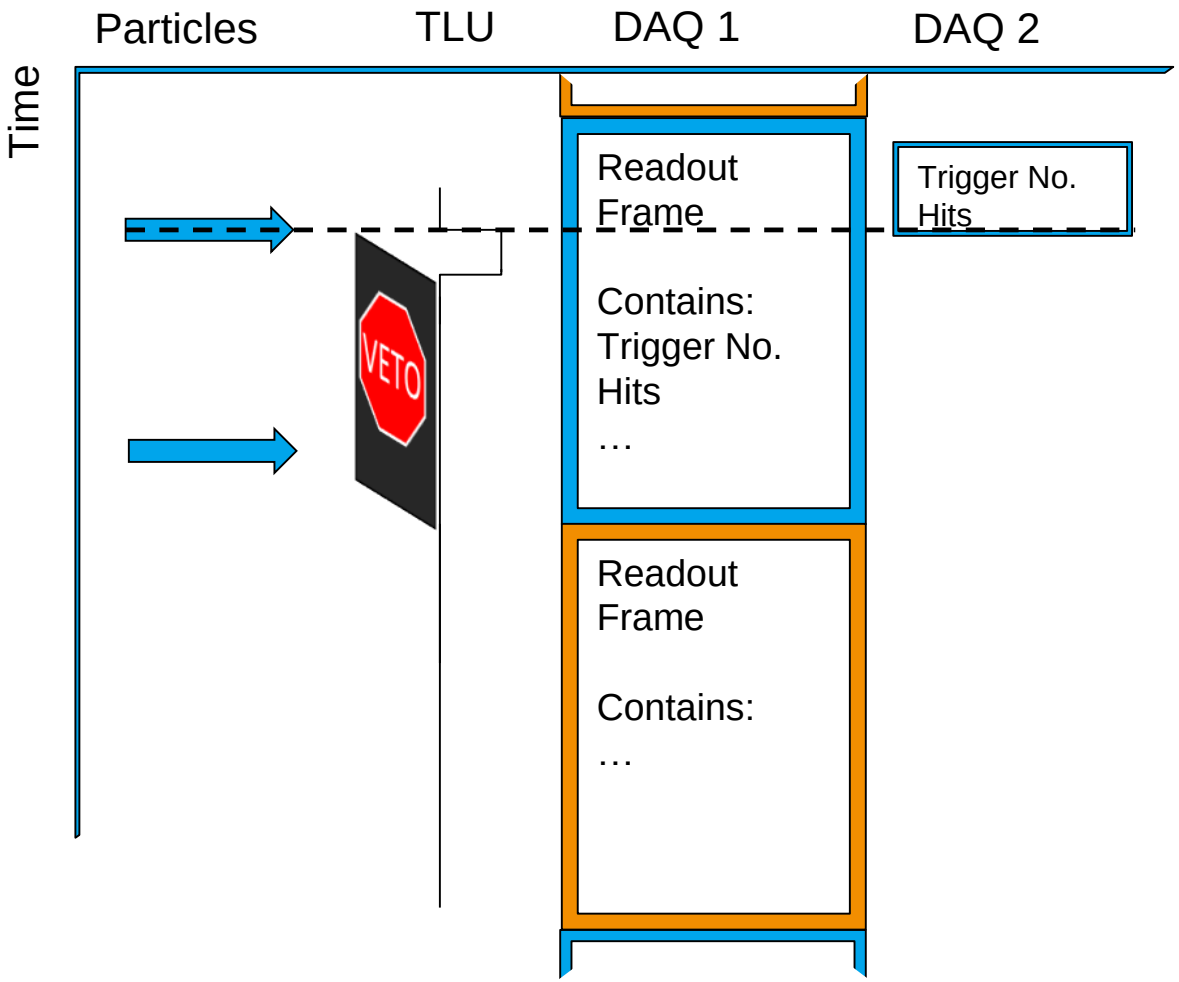
Asynchronous data streams

Combining of two types of sensors



- By adding the FE-I4 Detector the time resolution is increased by more than 3 orders of magnitude
- Gives the possibility to timestamp the individual Tracks

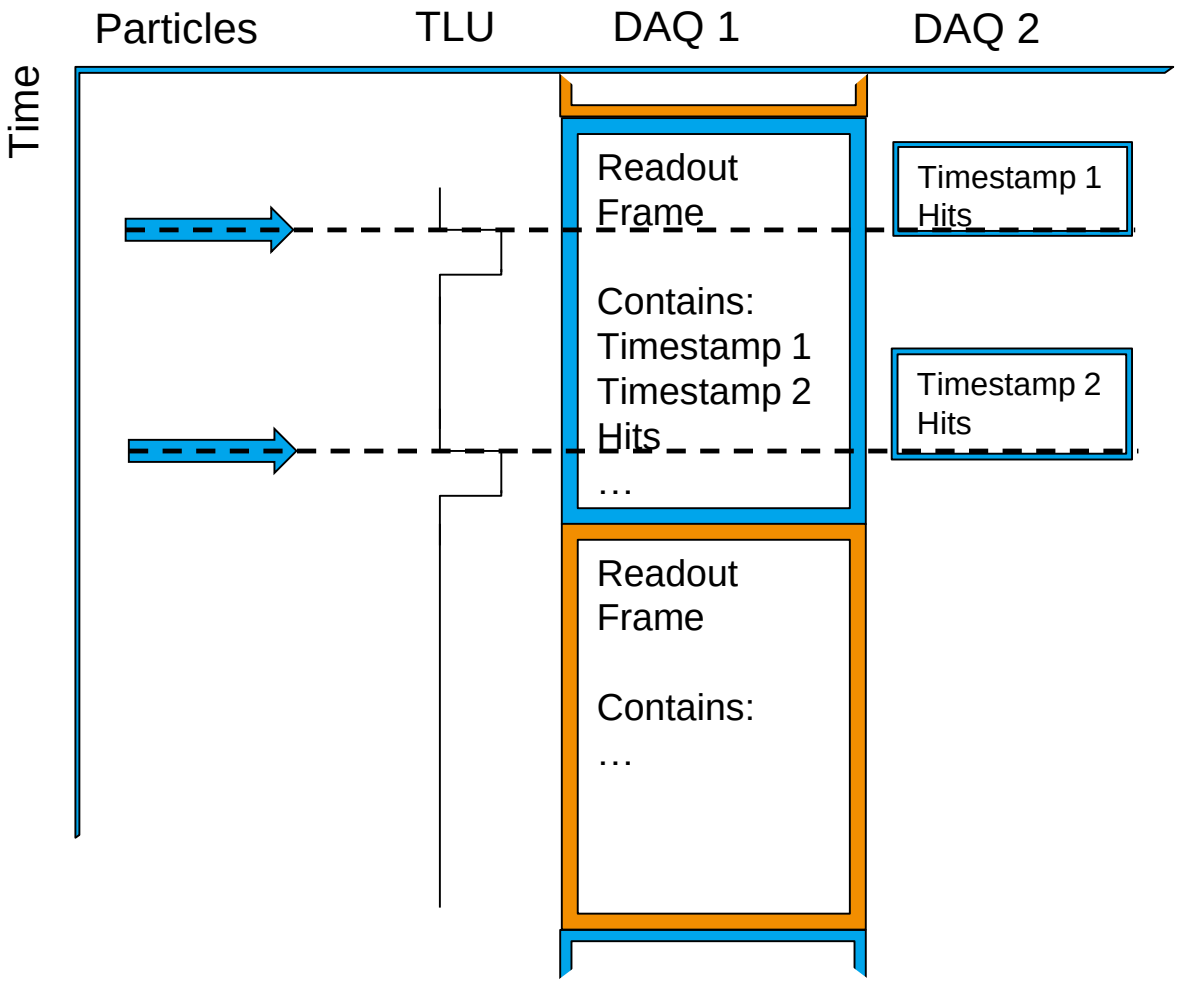
From Trigger to Timestamps



- One trigger per read out frame
- Prevents the issuing of triggers for the whole time of the read out
- All but the first particle are ignored
- Online event building
- Slowest device limits the Event rate

Marked for write to Disk
Not Marked

From Trigger to Timestamps



- Using all particles
  - Possible rate increase of two orders of magnitude
- No online event building
- Offline merging

Marked for write to Disk
Not Marked

Hardware Layout

> **Classical Trigger Handshake:**

- Device receives trigger and clocks out event number from TLU
- Device replies with a “busy” signal
- Slowest device limits trigger rate for all devices

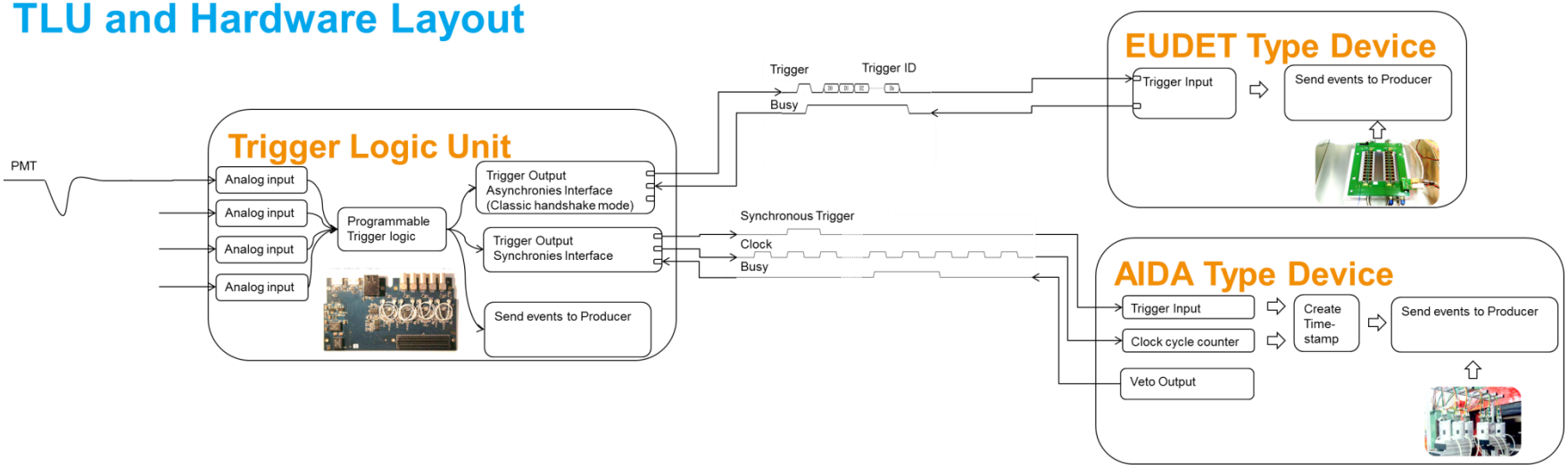
> **Synchronous Trigger interface:**

- TLU shares internal clock with DUT
- Device counts clock cycles
- On Trigger clock count is stored as timestamp
- No clock cycles used for transmitting timestamps

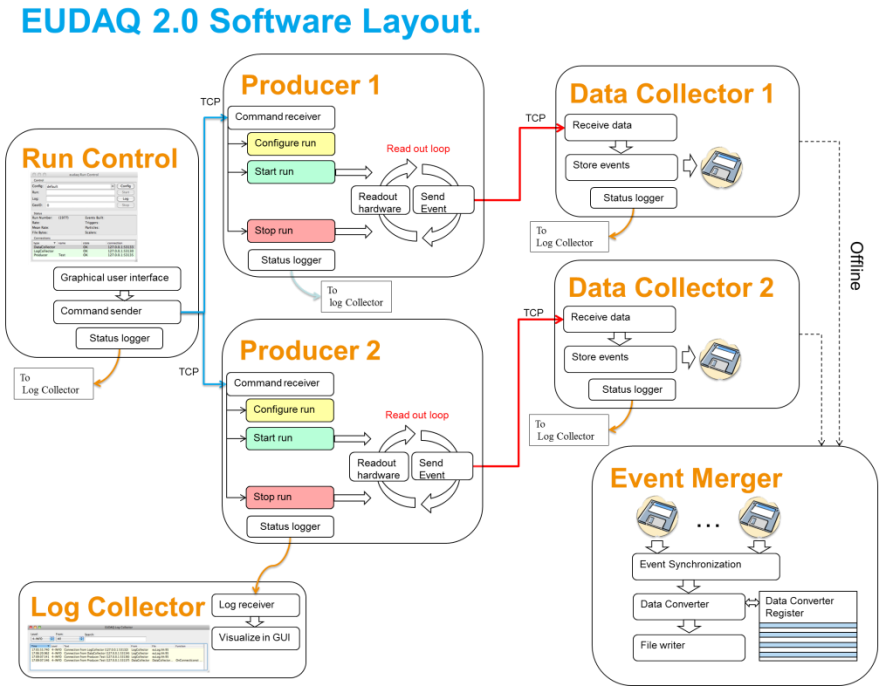
→ Trigger rate can be in the same order of magnitude as internal clock

- Device can rises busy signal to veto issuing of new trigger

TLU and Hardware Layout



Software Layout with Multiple Data Collector



> **Run Control:**

- Central authority
- Starts, stops and configures runs
- Assigns Data Collectors to Producers
- Main user interface

> **Producer:**

- Interface with user's readout system
- Receives command from Run Control
- Sends its data to the Data Collector

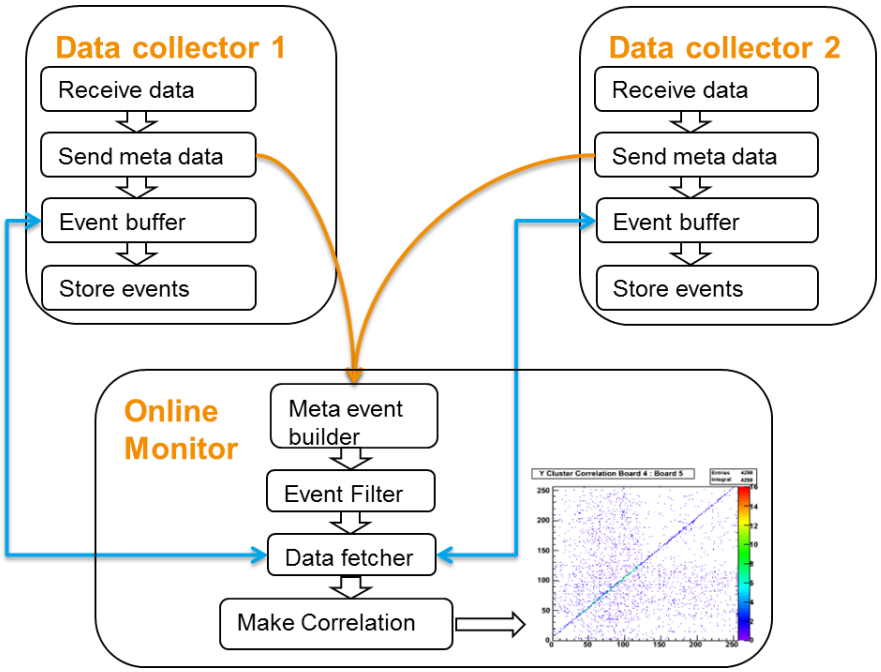
> **Data Collector:**

- Receives data from multiple producers
- Producer can have their own Data Collector on local machine

> **Log Collector:**

- Stores and displays log information
- Start time, stop time
- Errors and warnings

Online Monitor



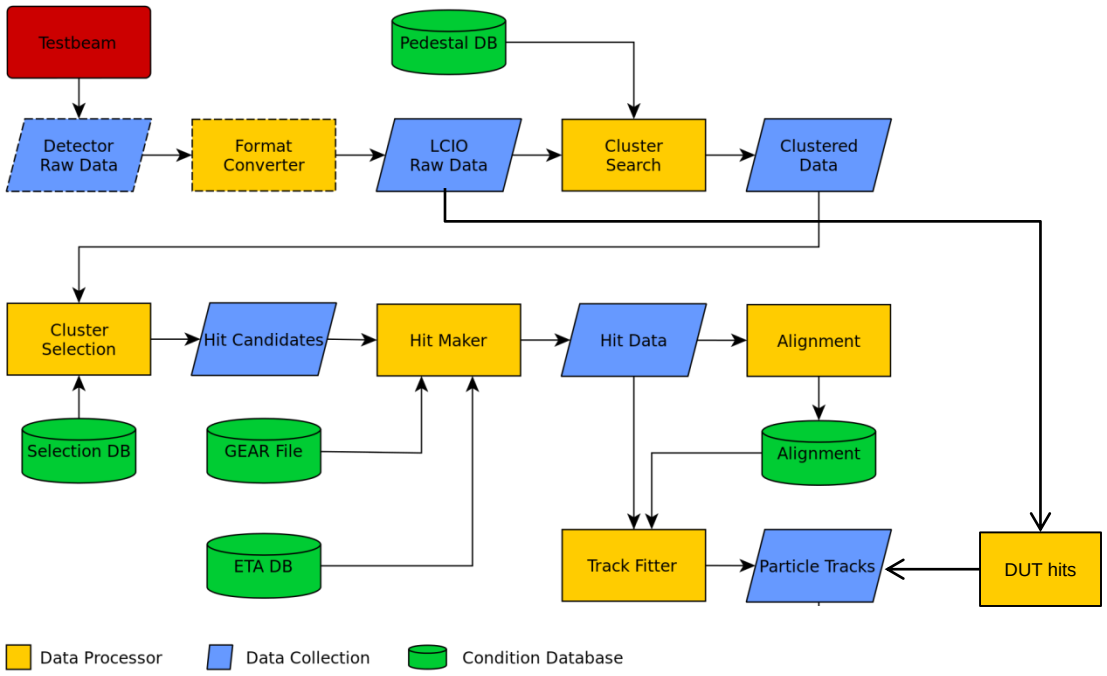
- In the new layout the data gets stored at different locations
- Data rate too high to process all events
- Receives all meta information
- Meta information contains timestamps and/or trigger number
- Uses meta information to synchronize events from different Producers
- Request a small subset of the data from
the Data Collectors
- Correlates the data from different device

Merging

- Every Data Stream is stored separately
- The TLU is the central authority for merging
- Every event gets compared to the TLU events to find the corresponding TLU event
- The Compare algorithm can easily be modified by the users. It is part of the converter plugin
- Adding the TLU timestamp to the Event
- Processing events from different producers Individually until the tracks are extracted
- Merging happens on the level of tracks and not events

Merging

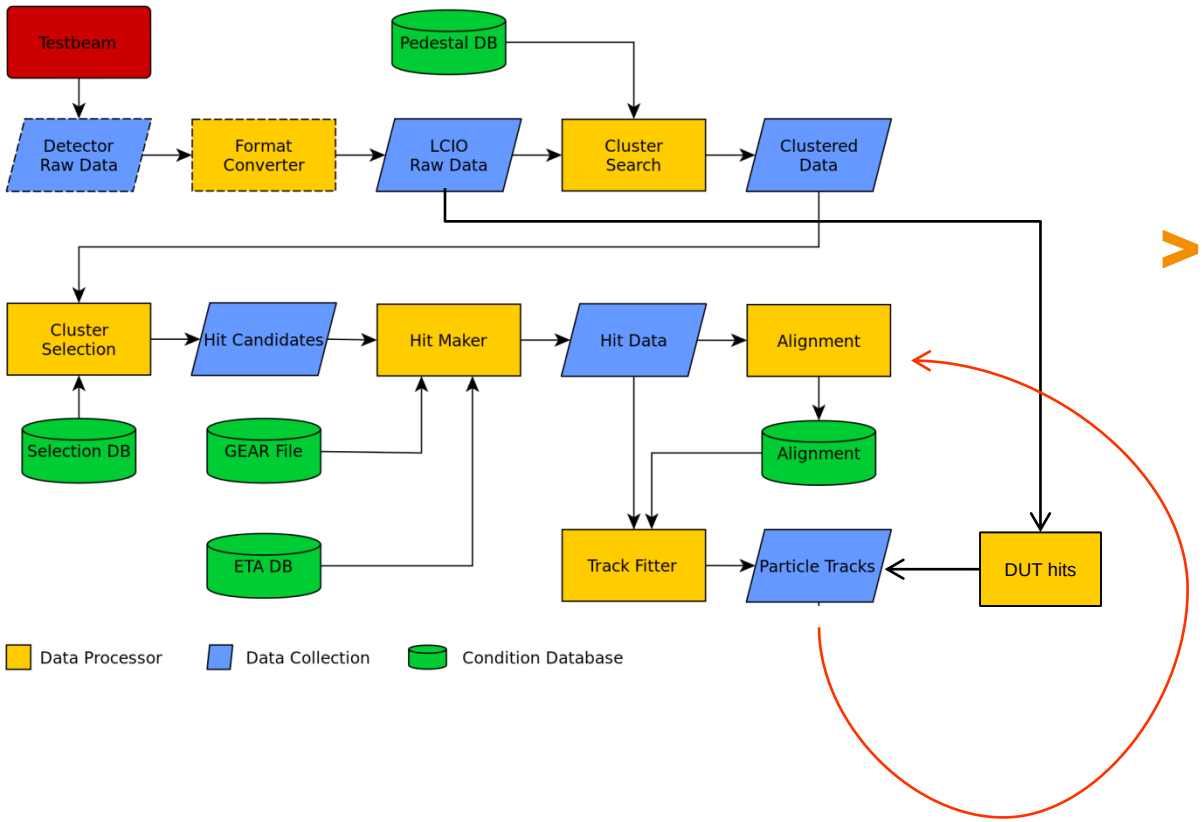
Current workflow



- The merging of the tracks with the DUT hits goes at the end.
- Reprocessing of alignment step after merging

Merging

EUDAQ 2.0 workflow



- The merging of the tracks with the DUT hits goes at the end.
- Reprocessing of alignment step after merging

Definition: Read Out Frame (ROF)

- Read Out Frame is the finest granularity one gets from a device
- It does not mean the granularity with which one reads out the Device.
  - As an Example: With every read out from the TLU one gets the information from multiple triggers. Since one can disentangle the information from the individual trigger the Read Out Frame is the individual Trigger!
  - For the FE-I4 one ROF is one LHC bunch crossing which covers a time of 25 ns
  - For the Mimosa one ROF is one frame which covers a time of 115 µs

Packets

EUDAQ 1.x

- Every Producer sends one Event per readout frame
- For some devices this leads to a lot of overhead
- With low rate (~4 kHz) the overhead is not limiting the data rate

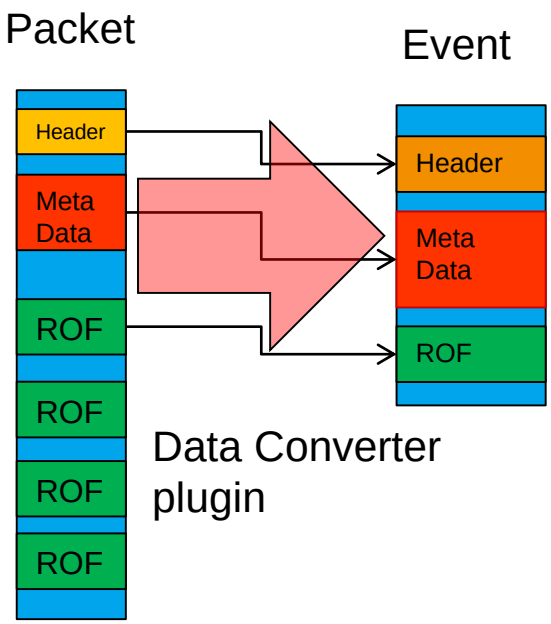
EUDAQ 2.0

- Desired rates:
  100 kHz -1 MHz
- The overhead from packing every ROF into one Event can Limit the data rate
- Allowing the use of Packet
- Packets can contain multiple ROFs.

➔ No overhead for the individual ROFs

- ROFs are extracted Offline

How to extract Events from Packets



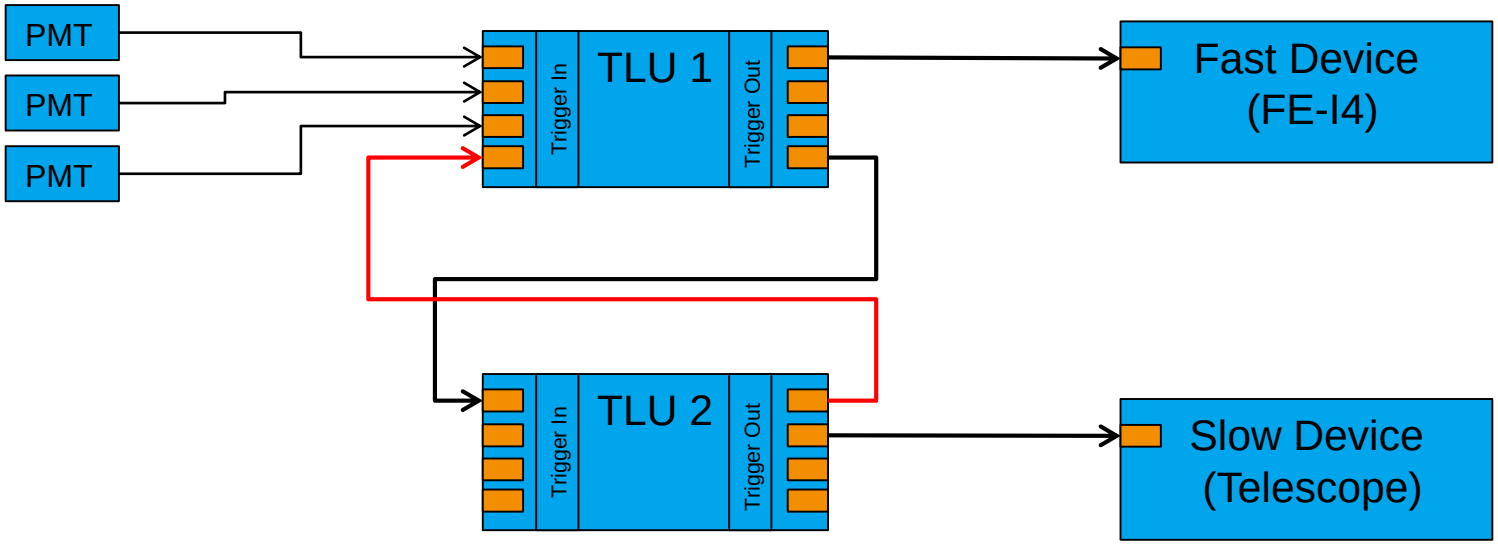
- Uses the well known mechanism of data converter plugins
- Complete flexibility how the users store their data

Example Code:

```
std::shared_ptr<eudaq::Event> FileReader::GetNextROF()
{
  std::shared_ptr<eudaq::Event> ev = getEventPtr();
  if (ev->IsPacket())
  {
    if (m_subevent_counter< PluginManager::GetNumberOfROF(*ev))
    {
      return PluginManager::ExtractEventN(ev, m_subevent_counter++);
    }
    else
    {
      m_subevent_counter = 0;
      if (NextEvent())
      {
        return GetNextROF();
      }
      else
      {
        return nullptr;
      }
    }
  }
  return ev;
}
```

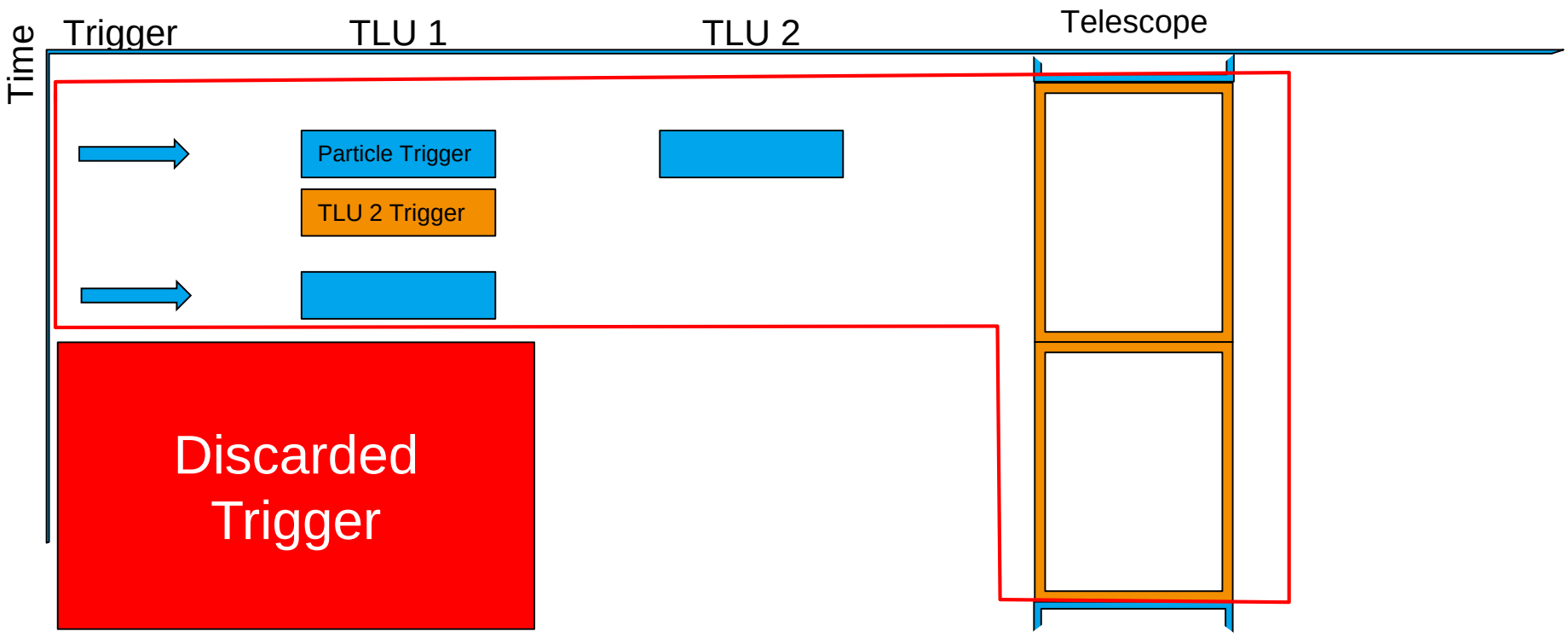
```
size_t DetectorEventConverterPlugin::GetNumberOfROF(
                                const eudaq::Event& pac
                                )
{
  size_t sum = 0;
  auto det = dynamic_cast<const DetectorEvent*>(&pac);
  for (size_t i = 0; i < det->NumEvents(); ++i)
  {
    sum += PluginManager::GetNumberOfROF(*(det->GetEvent(i)));
  }
  return sum;
}

std::shared_ptr<eudaq::Event> DetectorEventConverterPlugin::ExtractEventN(
                                std::shared_ptr<eudaq::Event> pac,
                                size_t NumberOfROF
                                )
{
  return std::dynamic_pointer_cast<DetectorEvent>(pac)->GetEventPtr(NumberOfROF);
}
```

Extra: Two TLU Setup



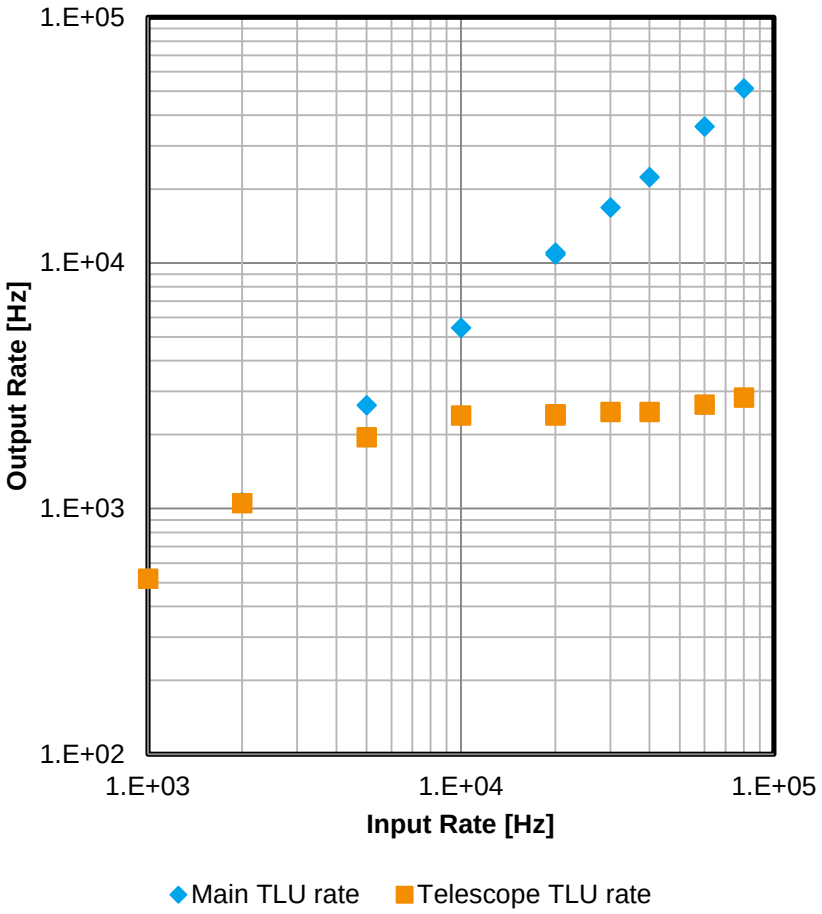
- TLU 1 is triggering TLU 2
- If the TLU 2 issues a trigger to the telescope it also triggers TLU 1 (TLU handshake)

Combination to events



- TLU 1 sees all particles
- TLU 2 only triggers when Telescope is not busy
- TLU 1 has a trigger from TLU 2 for the offline reconstruction
- Trigger which where issued after a certain time after the primary trigger will be discarded completely

Test in the Laboratory



- Input trigger from pulse generator
- The main TLU is not limited anymore by the Telescope
- The second TLU is synchronous with the Telescope
- Resynchronization of the two data streams via timestamps
  - Default approach for EUDAQ 2.0
- Not tested with beam yet

Summary

Accomplished

- New Data Format for Multiple Readout frames
- Possibility to Store Data locally to reduce the network overhead
- Merging for Online Monitor

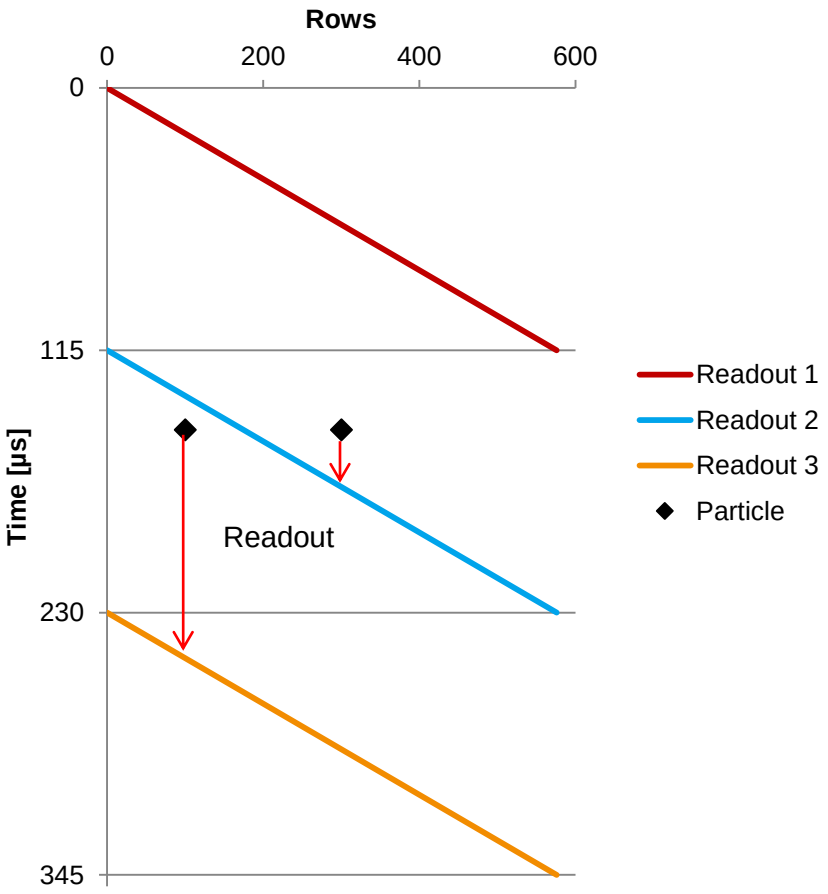
Open Task

- Track merging in EUTelescope
- Extensive beam tests
- Users need to update their producer converter to take full advantage from EUDAQ 2

End of slide show, click to exit.

Backup

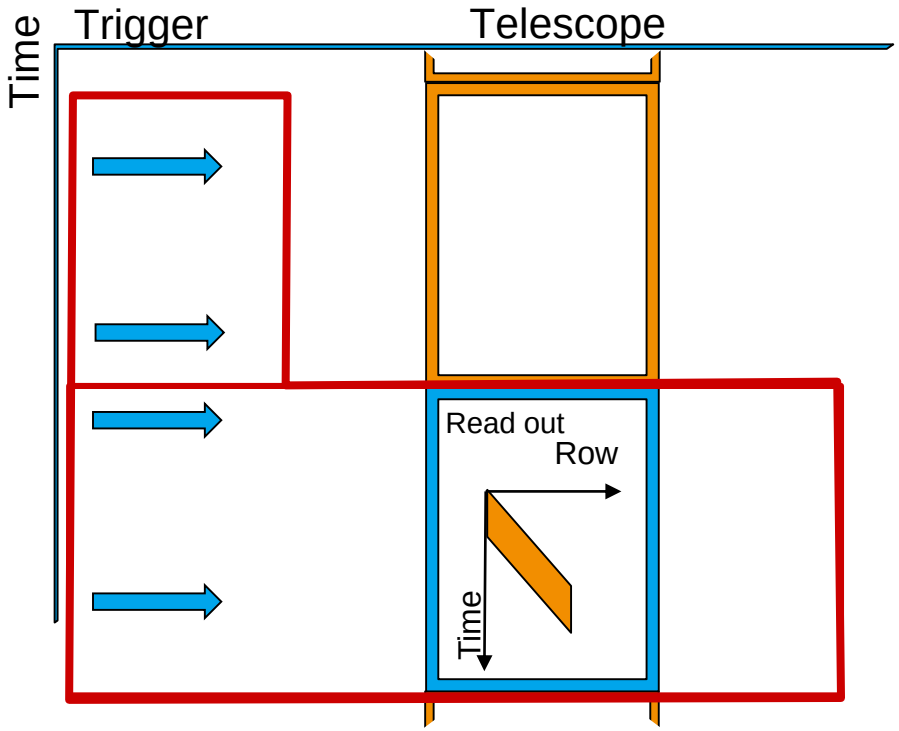
Mimosa Readout



- Rolling shutter readout
- Readout row by row
  - The readout takes 115 µs
- Hits that that appear at the exact same time can be in two different readout frames

➔ It is needed two associate two readout frames two one trigger

Telescope Type devices



- The information for one Trigger is split up on two ROF
- Limitation in the analyze framework prevents the use of references to previous events
- Instead of associating only the trigger that happened during the ROF we also Associate the one from the previous ROF to this Event
- Trigger get associated to multiple ROF