# Ongoing DAQ activities in CERN EP-DT-DI

Enrico Gamberini on behalf of

Giovanna.Lehmann@cern.ch
Enrico.Gamberini@cern.ch

CERN
EP-DT
Detector Technologies

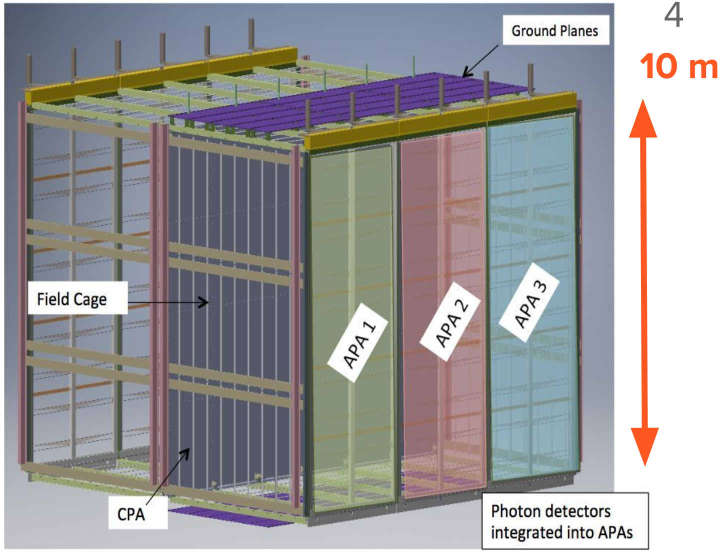
# Overview

- Presentation of the section
- ATLAS FELIX in ProtoDUNE-SP and NA62
- DAQling open-source software framework
- Storage R&D

# EP-DT-DI section

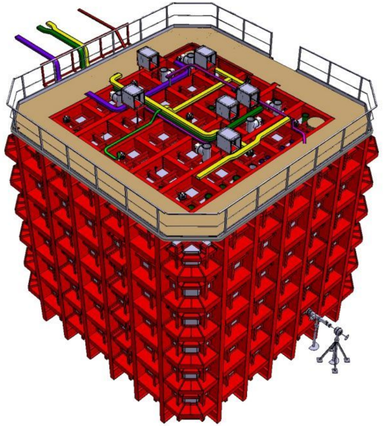
- “Provides solutions for infrastructure and detectors control and safety systems, as well as data acquisition systems for physics experiments.”
- Stabilisation and performance optimisation of the NA62 DAQ software
- Important role in the development of the NP04 (ProtoDUNE-SP) DAQ as well as its first operations on the beam line
- DAQ R&D towards DUNE
- DAQling software framework and collaboration with FASER, NA61/SHINE, and RD51 → joining effort on the DAQ development, to ease long term maintenance and support
- “Philosophy”: rely as much as possible on third-party tools and commodity hardware

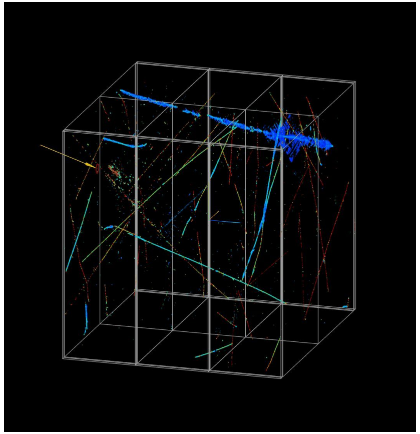
# ProtoDUNE Single-Phase




Demonstrate design, construction, and operation of the single-phase technology DUNE TPCs

- External cryostat dimensions: 10m x 10m x 10m
- 750 ton of LAr
- Charged particle beam from SPS





- Ionisation tracks are collected by the wires of the Anode Plane Assemblies (APAs)
- 6 APAs in ProtoDUNE-SP (4% of the 150 APAs of a DUNE supermodule)
- Scintillation light collected by Photon Detectors installed on APA frame
- Successful beam run in Q4 2018

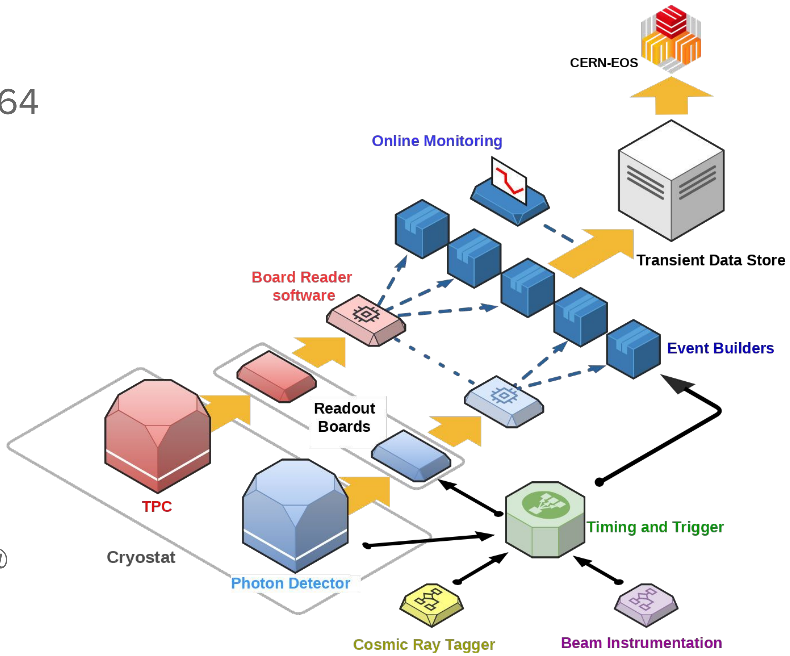
# ProtoDUNE-SP DAQ

## TPC readout

- Continuous digitization 464 Bytes @ 2MHz
- 15.360 channels
- 55 GByte/s
- Large buffers O(GBs)

## Photon Detectors

- Continuous digitization @ 150 MHz
- 240 channels
- Self triggering



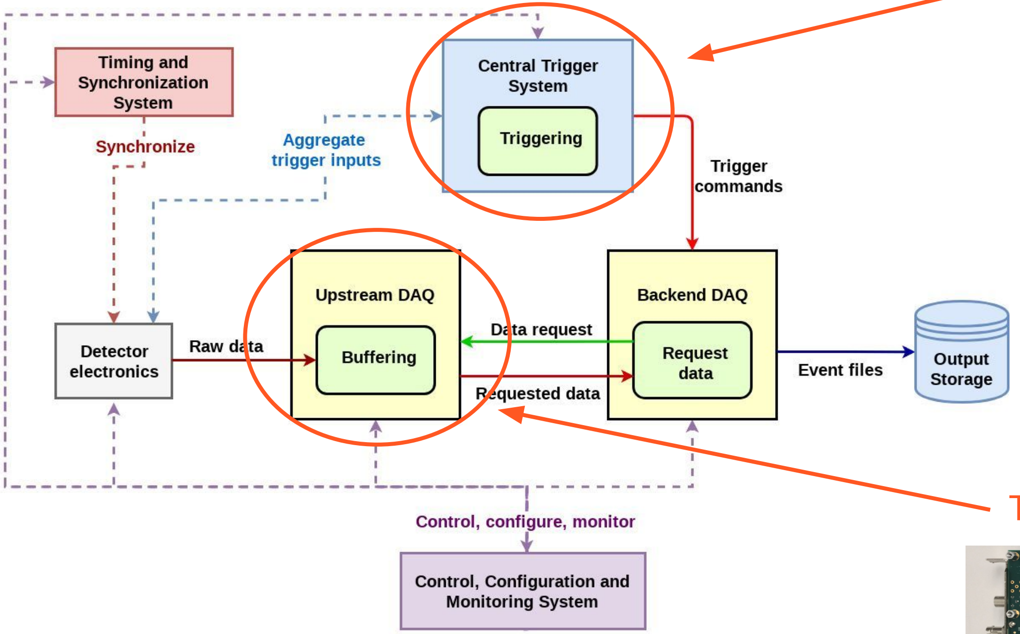

## DAQ farm

- On-detector electronics connected to DAQ via ~700 optical fibers
- ~20 high performance servers for dataflow, monitoring and control
- 700TB on-site storage
- Maximum 20 Gb/s data rate towards EOS

## Timing and trigger

- Phase-aligned master clock to all components
- Aggregate trigger inputs from CRT/PD/BI
- External trigger due to high rate of cosmic rays

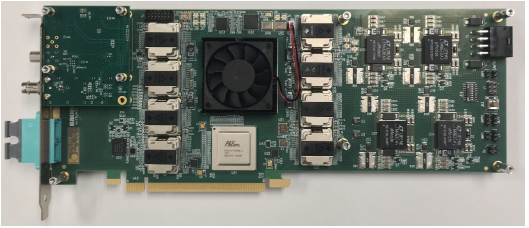
# ProtoDUNE-SP DAQ




External global trigger:

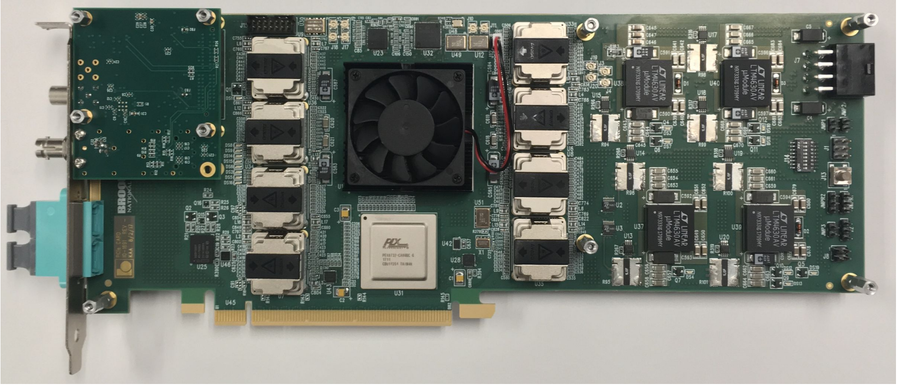
- Aggregated inputs
- ~30Hz trigger rate
- 3ms extraction windows
- Event size: ~60MB (compressed)

TPC readout with FELIX



# ATLAS FELIX

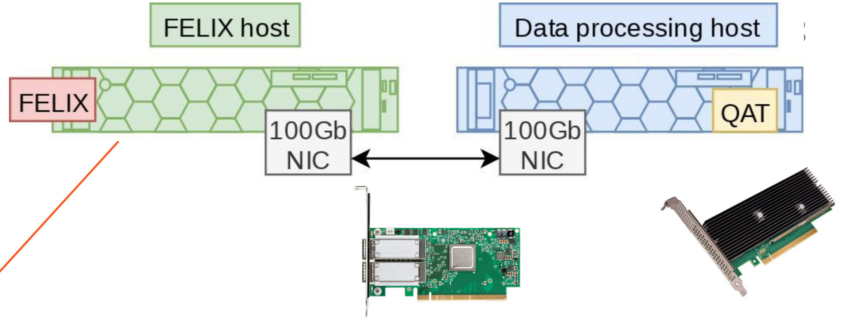
- **F**ront-**E**nd **Li**nk e**X**change
- Routes data between detector electronics and high-speed network-connected hosts (data, control, timing and trigger)
- Each FELIX system consists of one or two PCIe I/O cards hosted by a commodity server



- Interface via MTP24/48 connector, fanned out to MiniPODs
- 9.6 Gbps transceivers, support for GBT and FULL (8b/10b encoded) modes
- PCIe Gen3 x16 for communication with host server

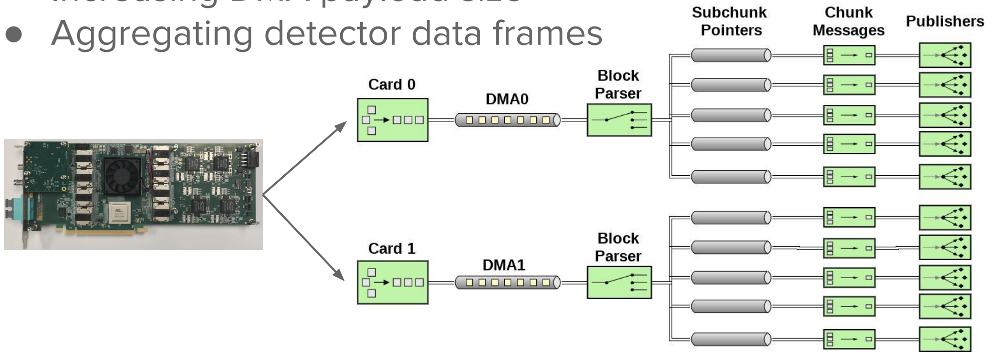
# FELIX in ProtoDUNE

1 FELIX card handles a single APA (10x ~1GB/s links)




Modest modifications on FPGA gateware
for lower memory I/O rate:

- Increasing DMA payload size
- Aggregating detector data frames

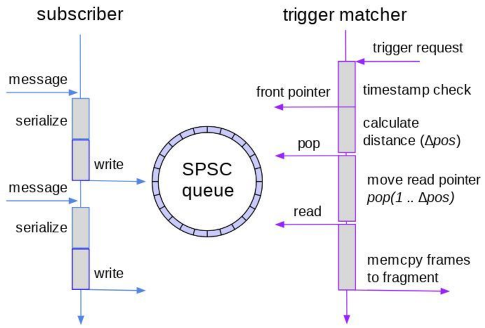



Data routing software is customized:

- Scatter-gather: collects pointers for detector data fragments
- Single copy pipeline: serialization to user buffer
- Data published on Infiniband over Ethernet

BoardReader process

- Subscribes to single link & buffers data
- Extracts data fragments for triggers
- Reorders data (AVX2 & 512)
- Hardware accelerated compression
  - Intel® QuickAssist® (QAT)



# More on FELIX in ProtoDUNE

OnHost BoardReader: merged FELIX data routing software, with data selection (trigger-matching) algorithm.

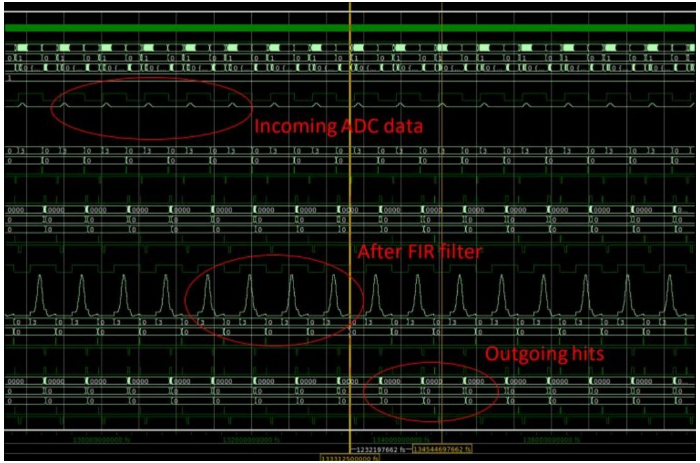
- Eliminated 100Gb P2P connection
- Reduced space and cost requirement

Covered:

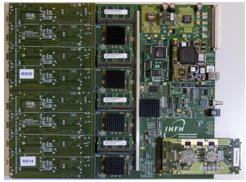
- Extensive server evaluation
  - PCIe riser configurations,
  - BIOS settings
- Optimized for memory throughput
  - Performance profiling
- Heavy NUMA balancing between processing threads and allocated memory
- Interrupt moderation of 10Gb NIC

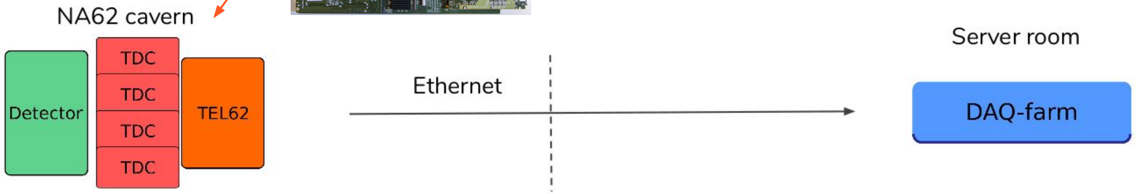
HitFinding in FELIX FPGA: FPGA gateware R&D for HitFinding

- Bitwise and byte operations are CPU heavy
- FPGA HitFinder implementation is ready
- Integration with FELIX is ongoing
- Outgoing hits have dedicated virtual links



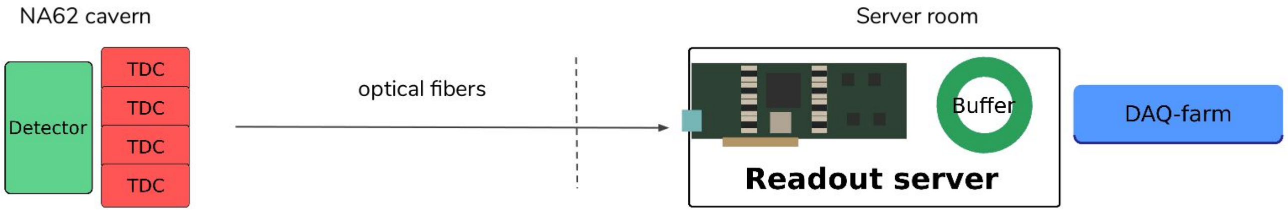
# FELIX in NA62






## Currently TEL62

- Time To Digital conversion
- buffer the data (limited)
- produce the trigger primitives
- perform trigger matching
- pack the events in UDP data frames and send the to the DAQ-farm




## New readout

On-detector:

- Time To Digital conversion
- send out the data via optical fibers

Off-detector:

- receive the data via optical fibers
- buffer the data (all the data can be cached)
- produce the trigger primitives
- perform trigger matching
- pack the events in UDP data frames and send them to the DAQ-farm

# FELIX in NA62

NA62 will restart the data-taking in 2021 after the LS2.

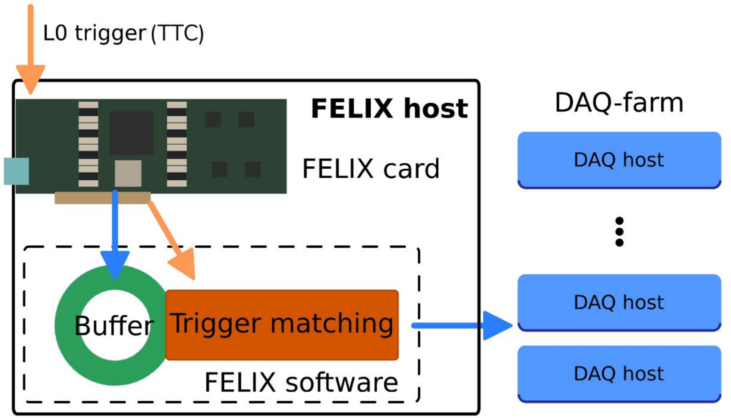
Goals:

- Equip KTAG and CHANTI, two high rate detectors, with new TDC and FELIX full readout.
- Demonstrate the capability to read data at 100% intensity
- Add NA62-specific trigger matching capabilities (L0 or L1) in the FELIX host.

The first results with trigger matching software using data at full intensity show that is possible to cope with the event rate.

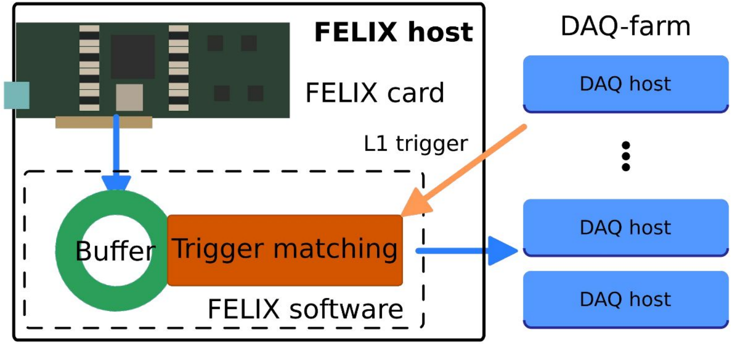
KTAG →

KTAG data are used to evaluate the L1 decision in the DAQ-farm



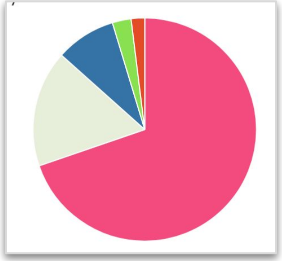

CHANTI →

CHANTI is not used in the trigger can be read at any level.



# "DAQling" software framework

- Software framework providing a generic data acquisition ecosystem
- Key features:
  - Lightweight dependencies ⇒ header-only where possible
  - Processing and data movement performance ⇒ C++17 and ZeroMQ
  - Extensible control and monitoring ⇒ Python
  - Human-readable and structured configuration ⇒ JSON
  - Easy deployment and build ⇒ Ansible automation
- Designed to scale to distributed systems
- Open-source at gitlab.cern.ch/ep-dt-di/daq/daqling
- Project started in 2019, but leveraging on third-party tools and libraries allowed for fast development time



**Programming languages used in this repository**

| | |
|---|---|
| C++ | 68.97 % |
| CMake | 16.69 % |
| Python | 8.62 % |
| Shell | 2.69 % |
| HTML | 1.94 % |

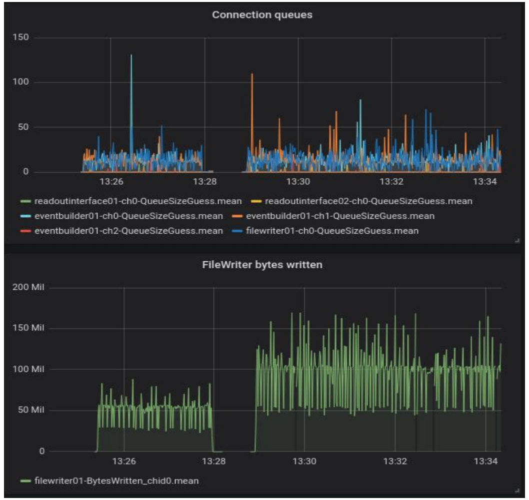
# DAQling features

## Integrated Logging

Log messages are formatted and sent to one or multiple sinks:

- stdout sink ⇒ Log file
- ZMQ publisher sink ⇒ Log collector(s)

Integrated operational monitoring

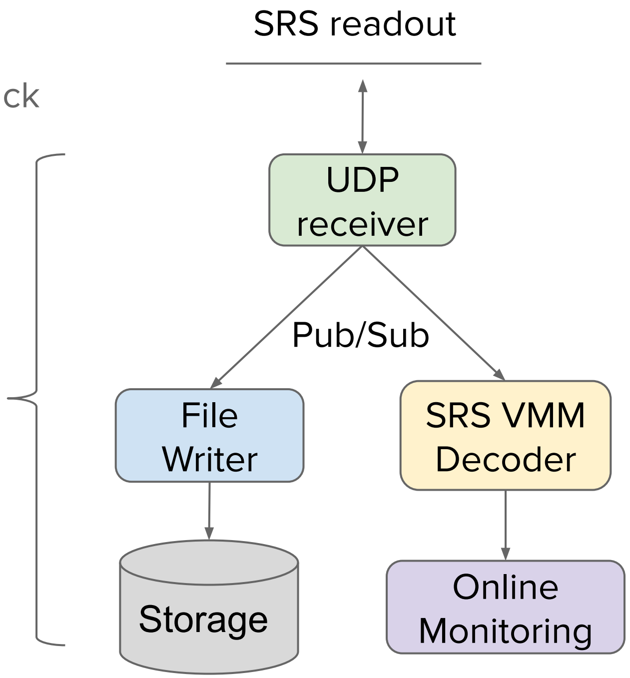



## Control for distributed system

- Process management based on “Supervisor”
- Control tree with configurable FSM rules

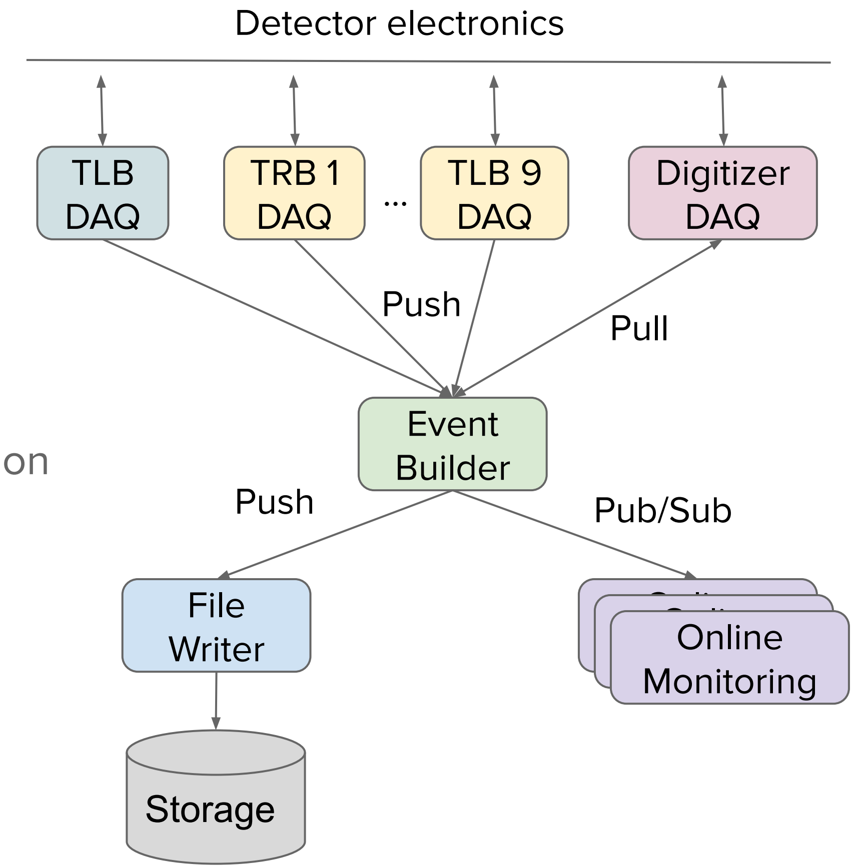
# Projects using DAQling

- FASER experiment at CERN:
  - Main user at the moment. More details in next slides…
  - First application ⇒ useful suggestions, requests, and feedback
  - FASER will acquire its first data in 2021, after the LHC LS2
  - … with no shifters 😱
- RD51 collaboration:
  - Laboratory setup for SRS readout + VMM3 ASIC
  - Raw UDP dump to file + decoder for monitoring/file writing
    - Implemented and tested in a couple of days
  - Possibility to scale up to test beam
- NA61/SHINE at CERN:
  - Use of significant part of DAQling for its DAQ upgrade



# DAQling for FASER

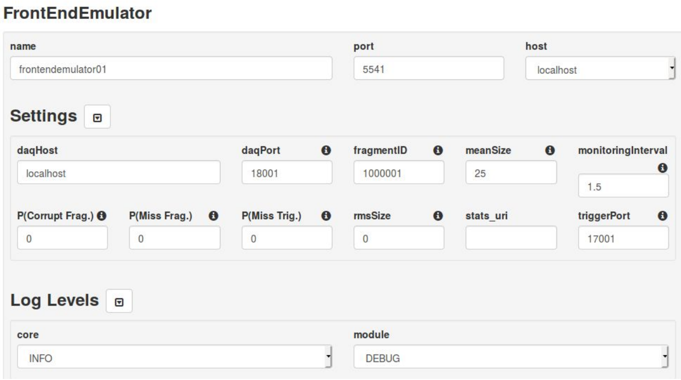
- Overview:
  - 1x Trigger Logic Board (~ 25 B fragments)
  - 9x Tracker readouts (>~ 250 B fragments)
  - 1x Digitizer (~ 15 kB fragments)
  - Trigger rate ~ 500 (peak 2k) Hz
  - Expected data on disk ~ 9 (peak 70) MB/s
- Successfully tested emulated full data flow on 2 servers
- Integration of detector readouts ongoing
- Automatic recovery manager and alerting under development
  - exploiting Logging, Monitoring, and Python Control library



# DAQling Web GUI

- Basic example developed by a FASER student:
  - Python web server based on Flask
  - Integration with op. monitoring display (Highcharts)
  - Configuration GUI based on JSON schemas


```
{
  name: "frontendemulator01",
  host: "localhost",
  port: 5541,
  type: "FrontEndEmulator",
  loglevel: {
    core: "INFO",
    module: "DEBUG"
  },
  settings: {
    meanSize: 25,
    rmsSize: 0,
    fragmentID: 1000001,
    probMissingTrigger: 0,
    probMissingFragment: 0,
    probCorruptedFragment: 0,
    monitoringInterval: 1.5,
    triggerPort: 17001,
    daqHost: "localhost",
    daqPort: 18001
  }
}
```




- Generalized version to be soon merged to DAQling and improved

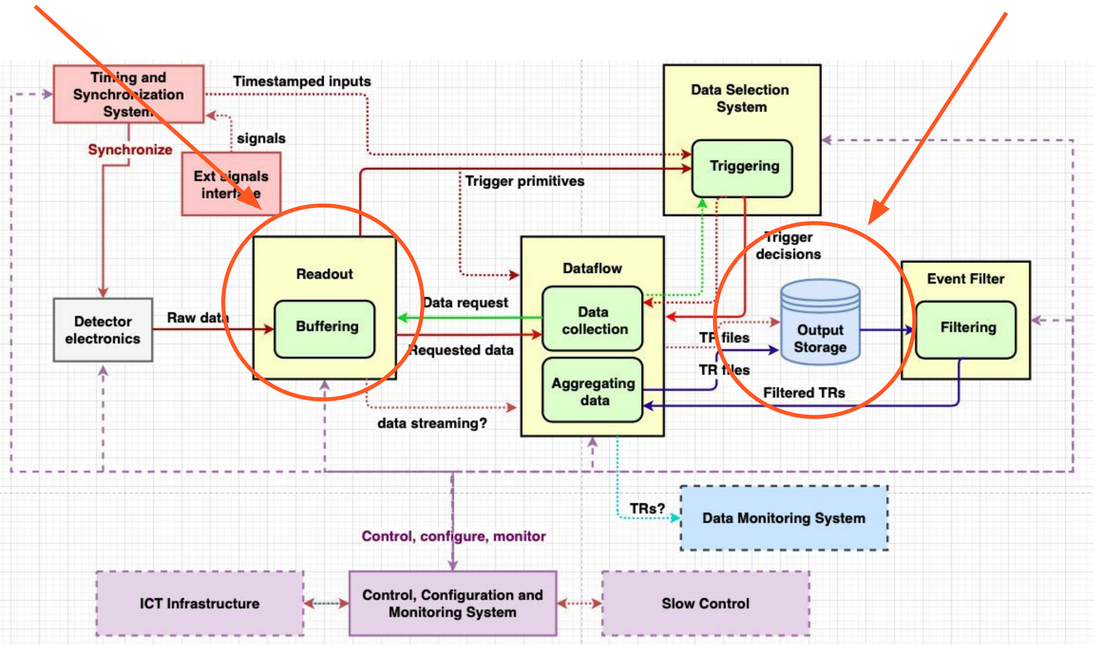
Enrico Gamberini

# Storage R&D

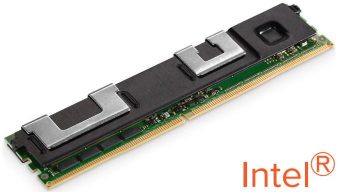
Focus on R&D for DUNE experiment on two fronts:

- Upstream buffer
  - Local high performance persistent storage
  - Keep ~2 minutes of data (4x180 TB)
- Downstream buffer
  - Distributed storage system
  - Keep data for days O(1) PB

Total readout rate O(1.5) TB/s per module



# Upstream buffer



Supernova Neutrino buffer → Persistent memory

- Critical data and high bandwidth:
    - Use of Non-Volatile Memory technology
- Successful prototype capable of buffering data from the readout system
    - Temporal buffer of 10 seconds
    - Store for over 100 seconds on SN trigger
    - Sustained a maximum throughput of ~7 GB/s
- Low level optimization lead to ~10 GB/s
    - Avoid PMDK library
    - Memcopy data and then persist to NVM
- Now: integration inside ProtoDUNE readout software

Intel® Optane Persistent Memory

- Innovative memory technology used together with DRAM
- Affordable large capacity. Single device offers 128 GB, 256 GB or 512 GB
- Support for data persistence
    - Data still available even after node failures
- Persistent Memory Development Kit (PMDK)
    - Low level software stack to develop applications for PMEM devices
    - Available for multiple use cases (libpmem, libpmemlog, libpmemblk)

# Downstream buffer

Key-Value stores for efficient logical even building

- Data fragments stored in a large distributed storage solution
- Event building is “done” by computing location of the fragments inside storage system (only metadata operation, no physical transfer)

Features

- Multi-key indexes for efficient data retrieval as needed by the experiments
- Select data events with a given event identifier, data source, data type, sub-detector
- Range queries
- Scalability to O(10) PB of data storage and support of 1M operations/second

DAQDB

- Open source project developed by Intel and experts from experiments (ATLAS, CMS)
- Goal: distributed key-value store for high bandwidth, generic data storage DAQ systems
- Adopts new technologies
  - Intel Optane Persistent Memory
  - Intel Optane SSD
- Use cases in particle physics:
  - High-Luminosity LHC experiments (e.g. ATLAS, CMS)
  - DUNE

# Summary

- EP-DT-DI provides solutions for data acquisition systems for experiments (R&D, software development, FPGA design, hardware evaluation, testing, commissioning, etc.)
- Experience from collaboration with ProtoDUNE, DUNE, ATLAS, NA62, and more.
- Interesting for COMPASS free-running DAQ (just my guesses):
  - Intel® QuickAssist for compression offloading
  - Intel® Optane NVMs for online storage
  - Experience on server evaluation and performance profiling
  - Experience with high performance networking
  - Some of the parts of DAQling might be reused or inspire development