

Data acquisition: Virgo legacy



*Introduction
Virgo DAQ architecture
From Virgo to ET*



*S. Viret (IP2I Lyon)
Virgo DAQ group
(s.viret@ip2i.in2p3.fr)*



→ Virgo online context:

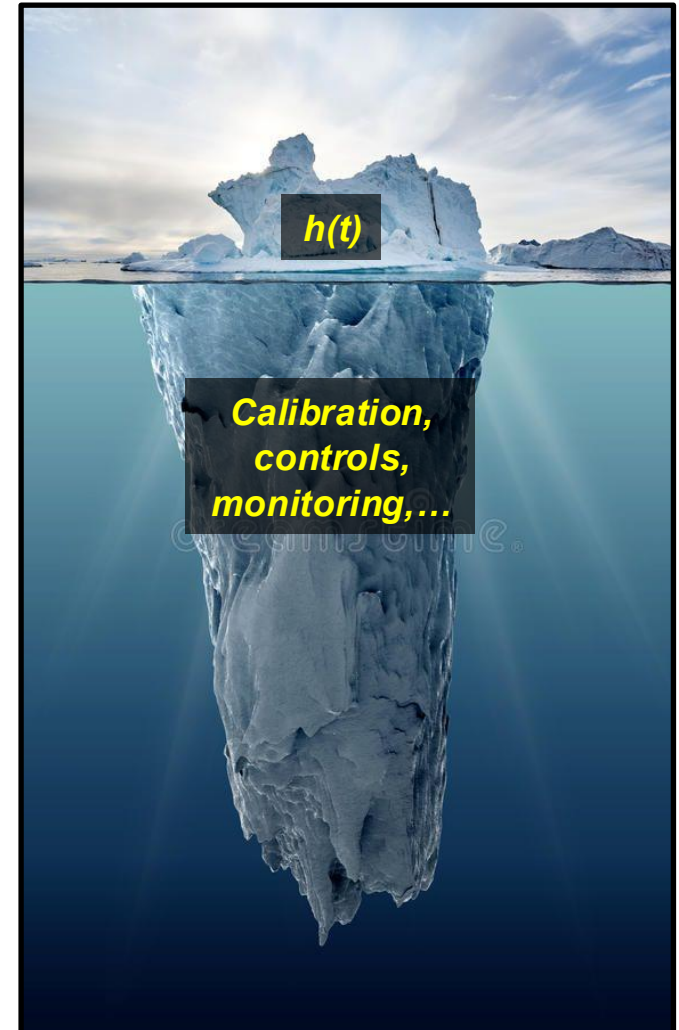
⇒ **GW interferometers are triggerless by design.** The strain is continuously recorded as soon as the ITF is locked.

⇒ One 4kHz signal per ITF... Looks simple on paper, but:

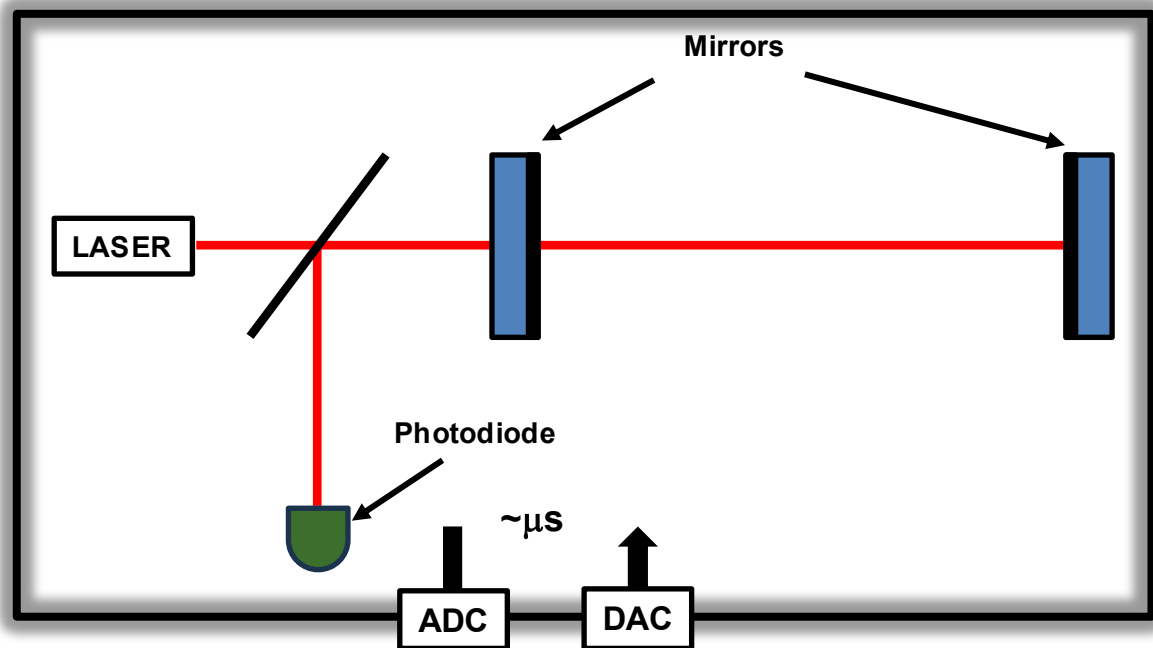
- Complex inter-calibration of the system is constantly needed
- Strain should be shared online with other ITFs
- Low latency data processing is required to alert other facilities

⇒ Online infrastructure should allow for **robust, simple, and smooth $h(t)$ production, distribution, and use.** The aim is to forget iceberg's underside

⇒ Lot of that will also be true for ET, just add an order of magnitude to the data flow to exchange...



→ ITF strawman view: an optical cavity which has to stay locked

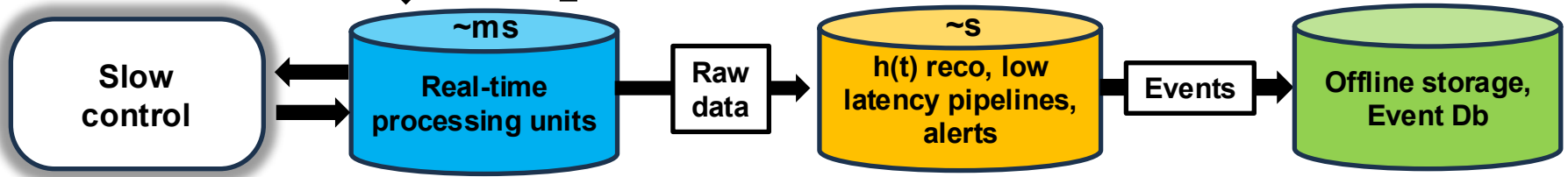


This presentation

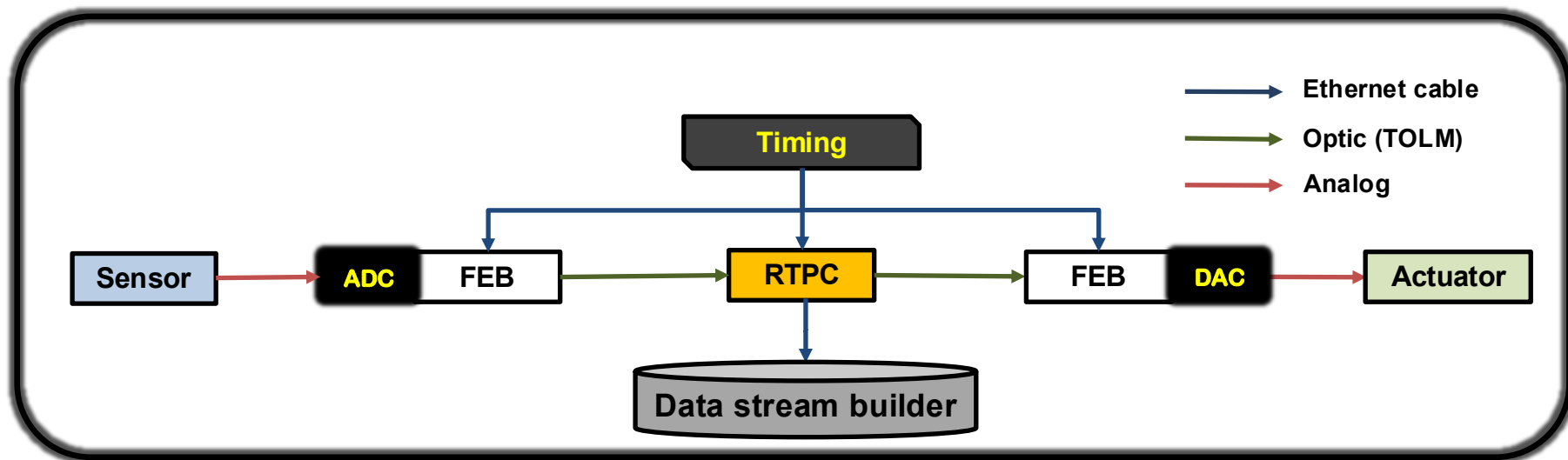
- ⇒ Input signals: up to **400MHz** for RF, **1MHz** for the rest.
- ⇒ Hard-real time processing (**down to 20-10kHz**): control loops
- ⇒ Interfaced with slow control (*lock acquisition, monitoring, operation*)

- ⇒ Full info stored in raw data stream (**~100(rawfull)+40(raw)MB/s**)
- ⇒ $h(t)$ and quality flag (**<0.1MB/s**) are then distributed by the ITFs to computing centers for low latency analysis.

Not in this talk



1. VIRGO DAQ architecture

→ Main principle


⇒ Analog signals are converted/processed by custom Front-End (FE) systems. Two main types of FE in Virgo:

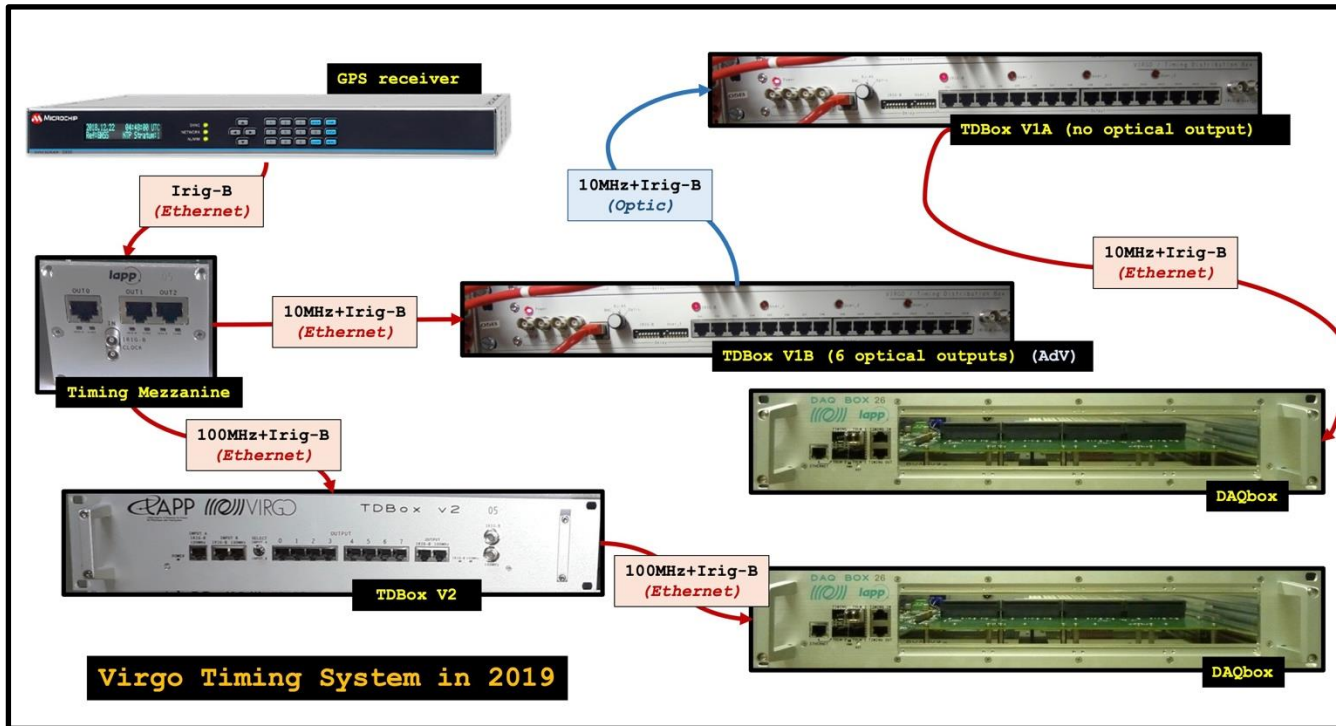
- **DSP boards** (*INFN*) for real-time control of the main mirrors position.
- **DAQboxes** (*CNRS/IN2P3*) for the rest of the control/detection electronics.

⇒ In both cases, signal from/to the FEs is encapsulated in a common custom format (**TOLM**) and transmitted via optical high-speed links. **TOLM is also an IN2P3 contribution (LAPP)**. As is the current **timing distribution network**.

⇒ In the next slides I will focus on IN2P3 contributions to Virgo DAQ

→ Timing system:

⇒ Custom timing distribution system developed by LAPP. Provides **IRIG-b signal** and **10/100MHz clocks** to the different DAQ elements. In place since 2008, upgraded in 2016 (*100MHz clock distribution*) in order to use digital demodulation.

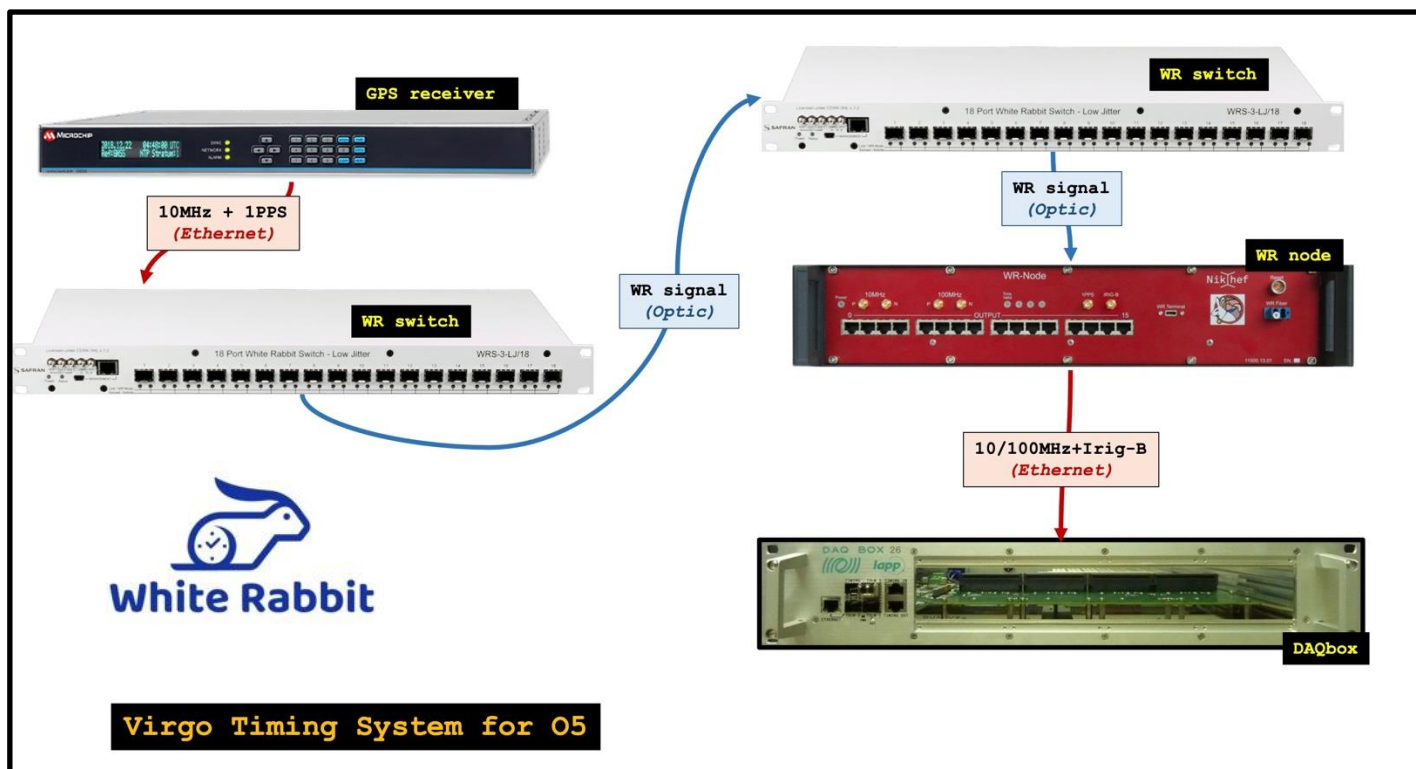


⇒ $1\mu\text{s}$ synchronization accuracy over the whole detector

⇒ Most of the controls are based on signals modulated at high frequency (MHz). Precise demodulation put high constraints on timing resolution (**phase noise**)

→ Timing system for O5:

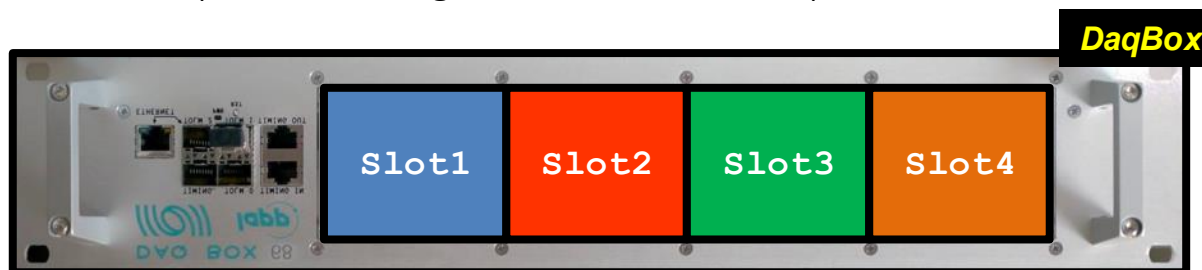
⇒ Current system will be replaced next year by a more generic approach based on [White Rabbit](#) protocol (*CERN-based open standard*). A dedicated board developed by Nikhef ([WR node](#)), converts the WR signal to ensure backward compatibility with current frontend, but **future FE boards will directly use WR signal**.



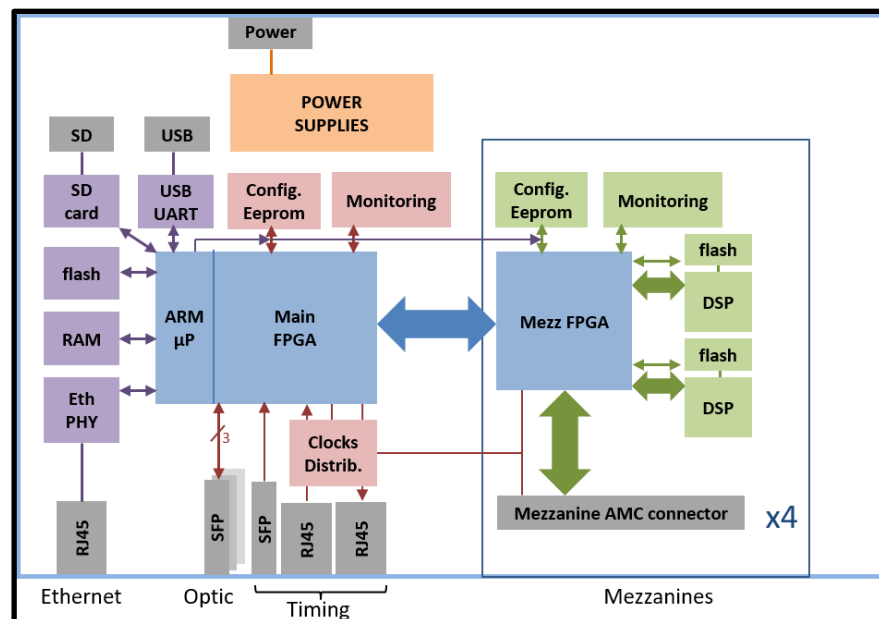
⇒ Certainly a very good starting point for ET (*used in ET-Pathfinder*).

→ DAQboxes

⇒ Low power 2U system with one motherboard and up to 4 mezzanines. 7 different mezzanines flavors available (**ADC, DAC, digital demodulation, ...**).



⇒ **Highly modular.** Subsystems can adapt the box to their needs with the mezzanines.



Motherboard block diagram

⇒ **High adaptability.** One can create a new mezzanine afterwards to adapt to new needs.

⇒ **Low power.** Motherboard power consumption is 24W, and mezzanines power range from ~6W (ADC) to 20W (Demod). **Can work without any active cooling**, even in vacuum (*used for control of suspended optical benches*)

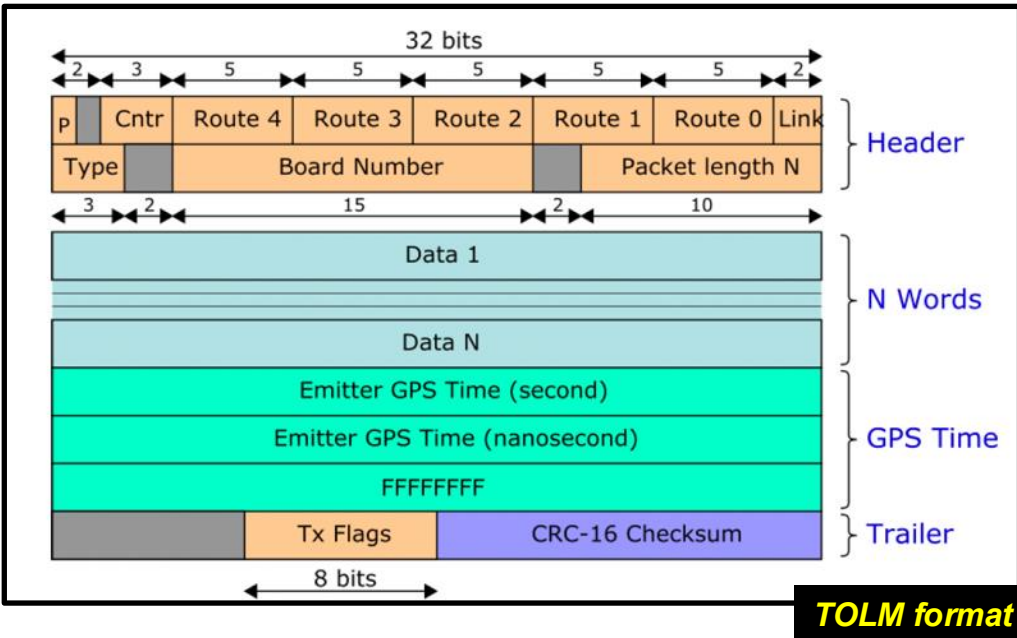
⇒ **New version in preparation for O5 (DBlite)**, will contain 8 ADC/DAC channels, and only 2 mezzanine slots. Will also handle WR signal by design. Will be used in ET-Pathfinder

→ The TOLM network and data format

⇒ Data exchanged through optical network using a custom format: **TOLM** (*Timing and Optical Link Module*). Packets are organized in 32 bit words sent through 2.5Gbps optical links.

⇒ **TOLM format simple and generic.** Every TOLM packet contains:

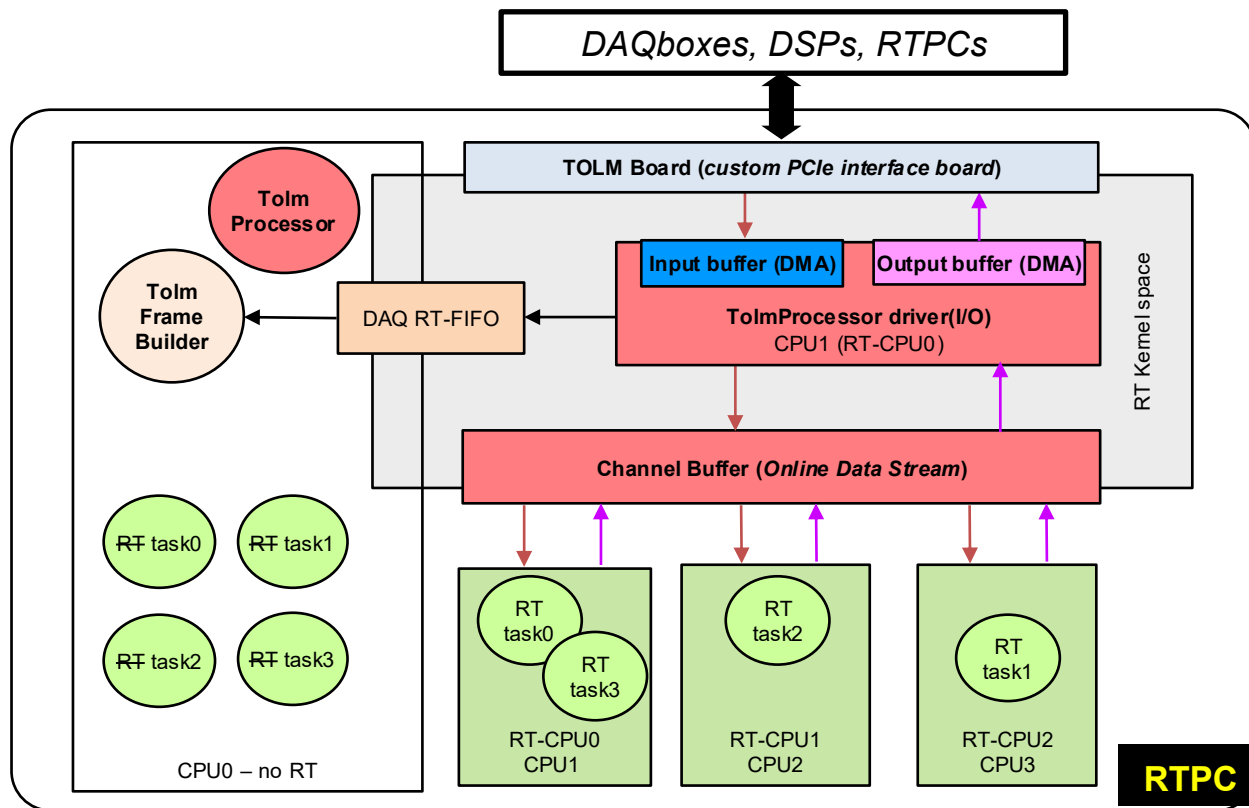
1. A header providing the packet ID and route in the optical network
2. Precise time tag, provided from the timing system.
3. A trailer with checksum info
4. A payload where the first word is a unique ID giving the data type (*needed to know how to decode the data*). The rest of the payload is the data itself.



⇒ As for the FE boards, the TOLM format can easily integrate new data types. **Particularly important for a detector which is complex to setup and characterize.**

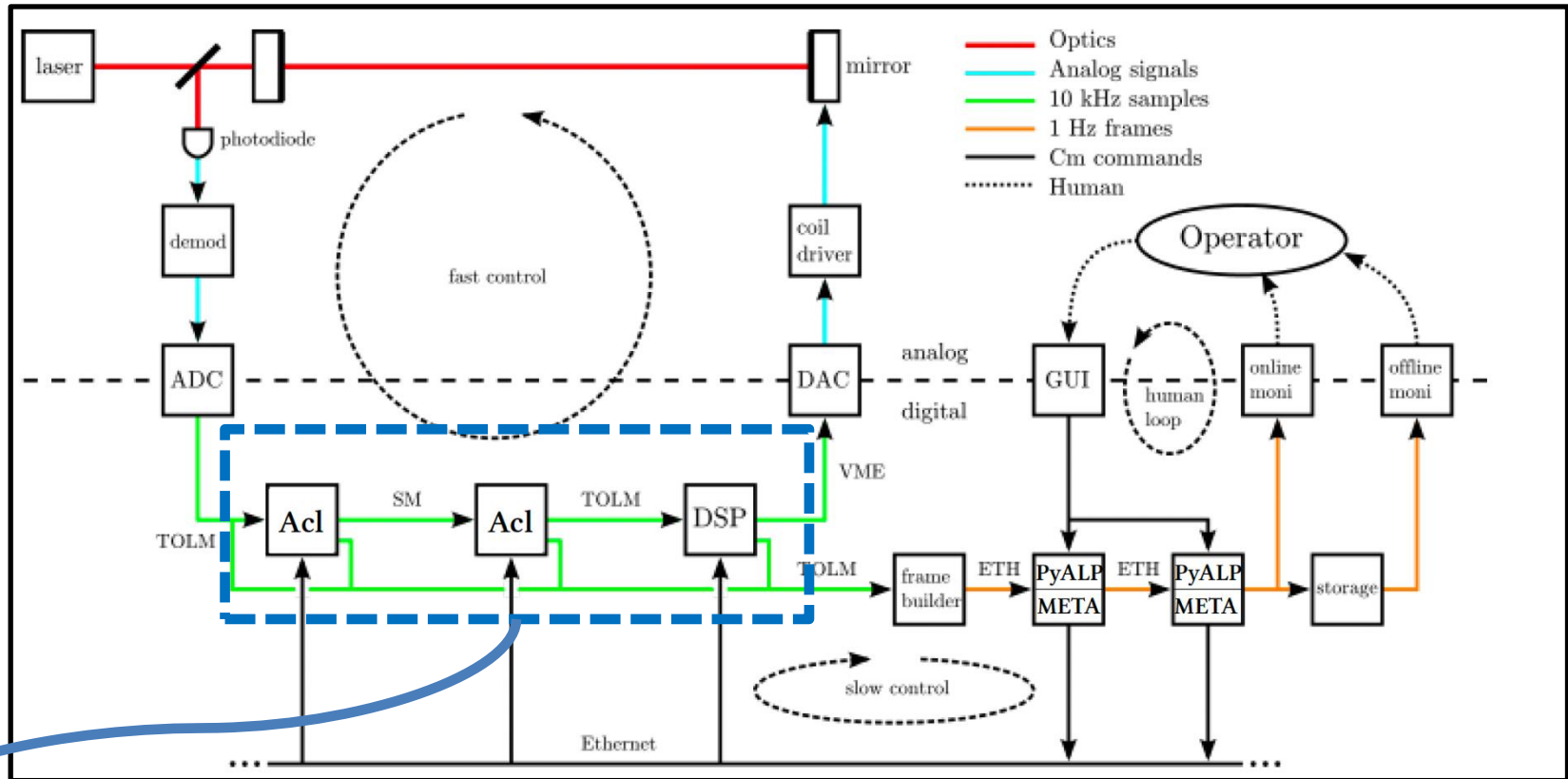
→ **Real time computing: the TOLM software**

⇒ TOLM data is retrieved by **21 real-time PCs** (4-CPU machines with RTAI Linux kernel). There a custom C software framework handles data reception/emission, and makes it available to real-time tasks (**TolmProcessor**).



⇒ Another tool (**TolmFrameBuilder**) produces the raw data frame (with 0.2 or 1s granularity). Also running on the RTPC, but not in real-time.

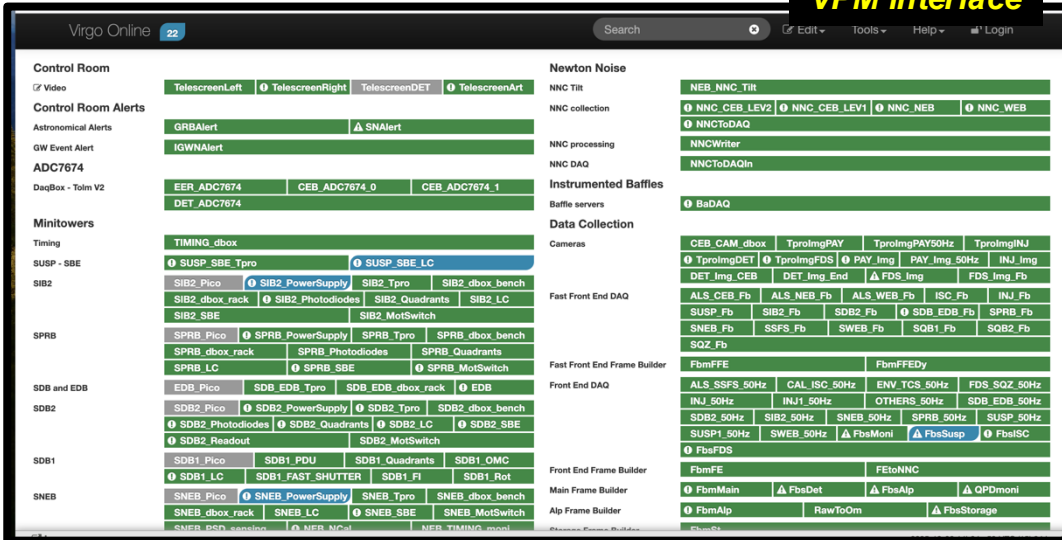
→ Online calibration, real time control



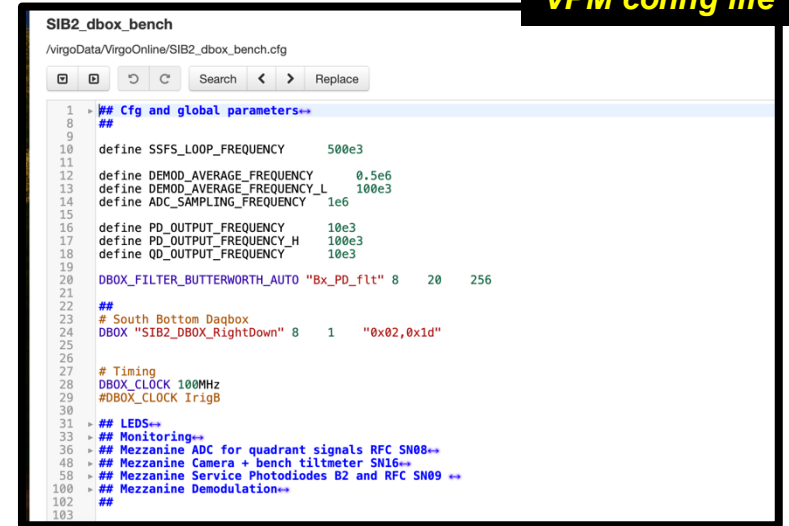
⇒ Real time control and calibration loops are handled either on **RTPCs** or on the **DSP slices** via dedicated custom low-level softwares: **ACL** for RTPCs (*C code*) and **DSPcode** for DSPs (*Assembly code*).

→ **RTPC tasks handling: the Virgo Process Monitoring (VPM) tool**

VPM interface



VPM config file



```

SIB2_dbox_bench
/VirgoData/VirgoOnline/SIB2_dbox_bench.cfg

1  >## Cfg and global parameters<<
2  ##
3
4  define SSFS_LOOP_FREQUENCY      500e3
5
6  define DEMOD_AVERAGE_FREQUENCY  0.5e6
7  define DEMOD_AVERAGE_FREQUENCY_L 100e3
8  define ADC_SAMPLING_FREQUENCY   1e6
9
10
11
12  define PD_OUTPUT_FREQUENCY      10e3
13  define PD_OUTPUT_FREQUENCY_H    100e3
14  define QD_OUTPUT_FREQUENCY      10e3
15
16
17  DBOX_FILTER_BUTTERWORTH_AUTO "Bx_pdflt" 8  20  256
18
19
20
21  ##
22  # South Bottom Dabox
23  DBOX "SIB2_DBOX_RightDown" 8  1  "0x02,0x1d"
24
25
26
27  # Timing
28  DBOX_CLOCK 100MHz
29  #DBOX_CLOCK IrigB
30
31
32  >>## LEDS<<
33  >>## Monitoring<<
34  >>## Mezzanine ADC for quadrant signals RFC SN08<<
35  >>## Mezzanine Camera + bench tiltmeter SN16<<
36  >>## Mezzanine Service Photodiodes B2 and RFC SN09 <<
37  >>## Mezzanine Demodulation<<
38  ##
39

```

⇒ **Easy to use process supervisor (C/Javascript)** developed to handle the control procedures processed on the RTPCs.

⇒ **Remote access/editing possible** via VPN (useful for on-call expert which are not on-site)

⇒ Bookkeeping of the previous configurations, we can easily retrieve the detector config at a given date

⇒ **Generic tool:** also used for all non real-time online computing (*h(t) calculation, frame production and storage,...*). Another VPM instance is handling data transfers and a low latency pipeline.

2.From Virgo to ET

→ Virgo legacy:

⇒ Most of the elements used today in Virgo DAQ were developed by IN2P3 (*mostly by LAPP and IPHC*). Just presented the low level here, but there were also significant contribution to the **computing model**.

⇒ Regarding the hard realtime part, a **complete ecosystem was developed**. This system is particularly adapted to GW interferometer constraints:

- **Low noise, high modularity, and scalability.**
- **Digital demodulation and fast digital control loop processing.**

⇒ ET will use more modern tools, but **some of the bricks built in Virgo can serve as a starting point:**

- **Modular/Low power FE board:** DAQbox lite
- **Digital demodulation**
- **Simple/scalable online data format:** TOLM format (maybe via Ethernet)
- **Simple and robust RT software framework for fast loop processing at higher frequencies:** ACL, Tolm software,...
- **Online processes monitoring tools:** VPM

⇒ **Each of these tools could become a wheel you won't have to reinvent for ET.**

⇒ Having said that: if one wants to exploit this legacy within ET, **new groups (*within IN2P3 or not*) will have to get involved.**

→ A slide on AI...

⇒ AI is not yet the common thing in LVK DAQ frameworks, but **some first applications are coming** (eg *online suspension control*).

⇒ No doubt this will take a fair share of 3rd gen ITF DAQ frameworks. **The need for real-time AI processing tools will grow quickly.**

⇒ We also can play a role here, either at the **CPU level** (*NN inference on RT software*), or closer to the detector, on **low level hardware host** (*NN inference on FPGA*).

⇒ With a DAQ infrastructure like the Virgo one, inclusion of these elements does not, in principle, requires any new hardware development.

Conclusions

⇒ Virgo online infrastructure, thanks to a very modular and generic approach, has been one of the most seamless subsystem of the interferometer during the past decades.

⇒ Improvements and more modern tools are needed, but ET and Virgo DAQ model will not be fundamentally different.

⇒ A remarkable expertise has been developed in this domain, and a central role was played by IN2P3.

⇒ This hardware/software expertise can be a real bonus for ET DAQ initial developments.

→ DSP boards

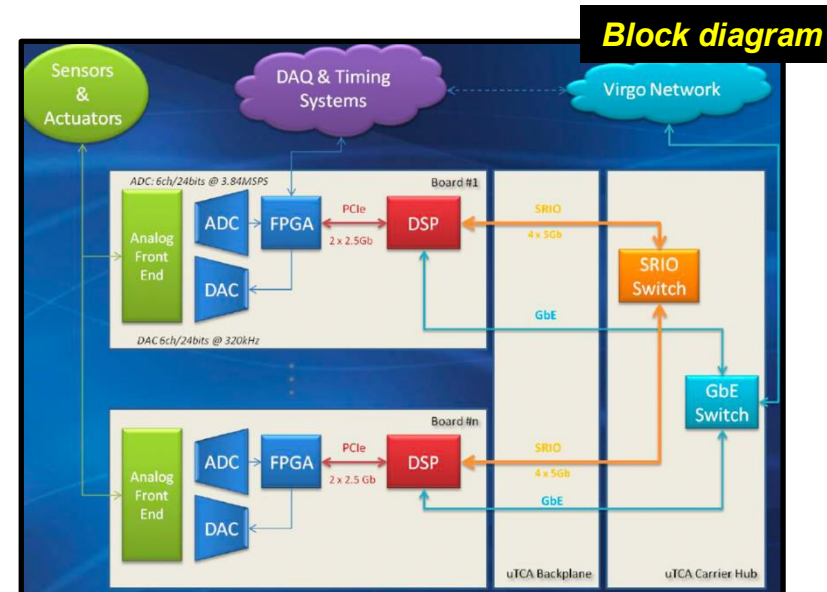
⇒ μ TCA custom board (**USDPT**) handling main mirrors suspension controls

⇒ High rate fast loops, computations handled by DSPs (*processing data up to 320kHz*)

⇒ Every suspension is driven by a set of 12 USDPT boards (**local control unit**).

⇒ Data from/to the control unit is exchanged with the TOLM format.

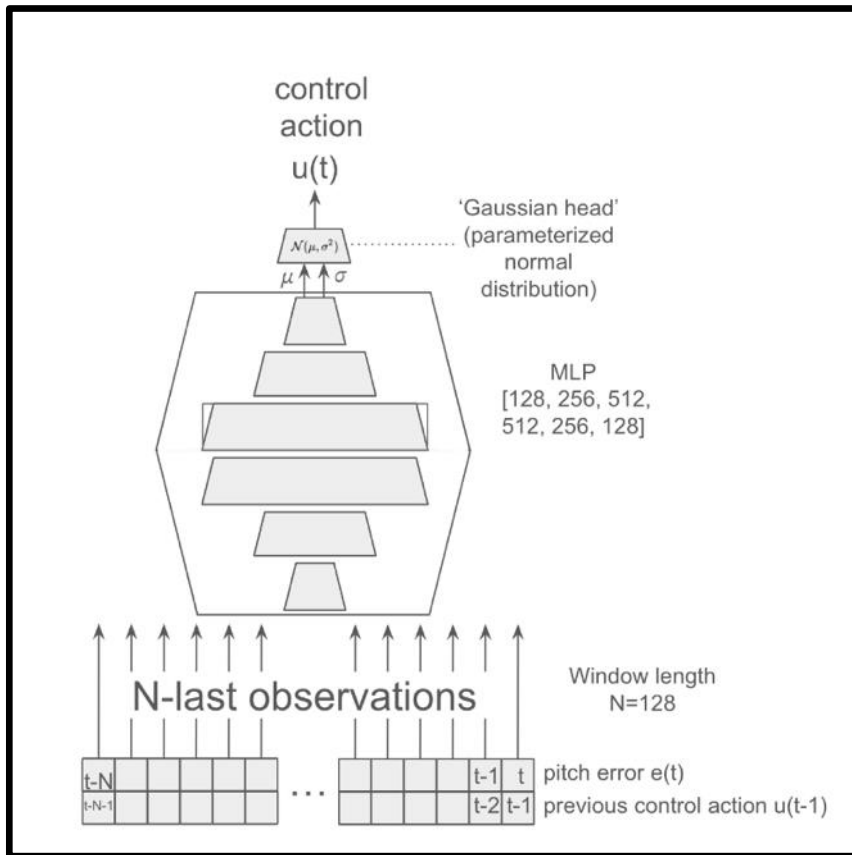
⇒ The current setup is being upgraded for O5 (*3 simpler boards will replace the generic USDPT system, native WR timing, ...*)



Control unit

→ Plans for the future: an example

⇒ Usage of reinforcement learning to replace standard control loops has been tested recently in LIGO (<https://arxiv.org/pdf/2509.14016v1>), evaluation in Virgo has started

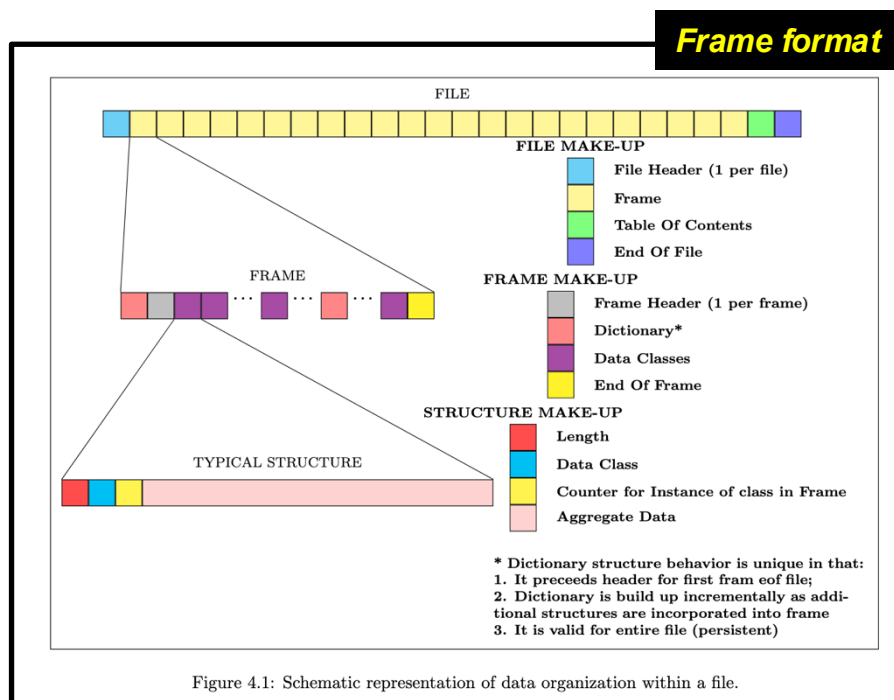


⇒ Inference should be done in real-time on RTPCs using incoming data stream.

⇒ This kind of upgrade is targeting more the next generation of ITFs, but LIGO and Virgo ITFs offer the perfect testbench to characterize them.

→ Raw data production

⇒ More related to online analysis, but produced on the RTPCs. Raw data structure base on a **common data format (frame)** used by all the LVK collaboration. Format used also for offline analysis.



⇒ Frame and TOLM data format were developed by the same team and therefore share some structural similarities

⇒ **Idea is once again to have something very modular adapted to any kind of data formats (from INT8 to hi-res images).**

⇒ Frame building and processing handled by a dedicated custom C software framework.

⇒ Raw data frames used by downstream online servers are produced with 1s granularity. Streaming ends at this point in Virgo, but this granularity is sufficient for the moment.