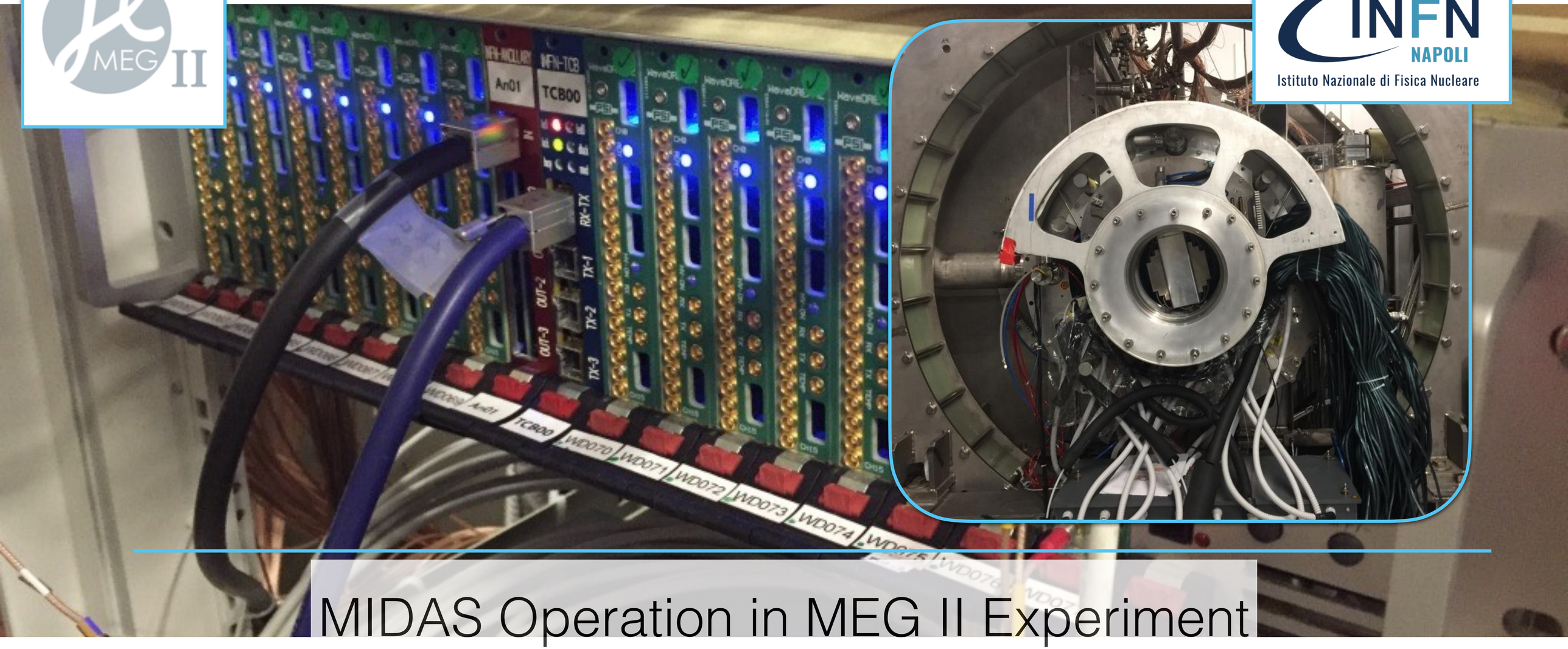
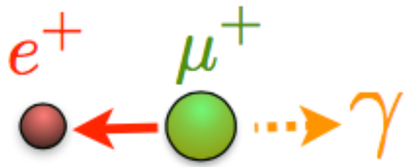


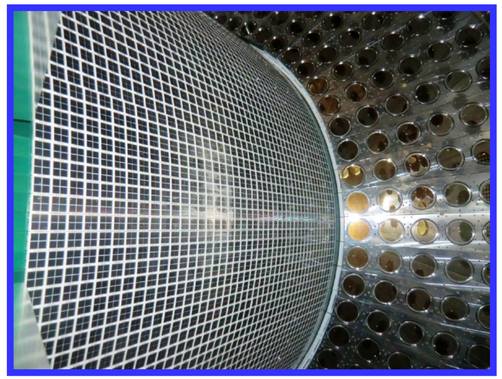
MIDAS Operation in MEG II Experiment

Marco Francesconi

MIDAS Workshop, ZOOM, 12 Sep 2023

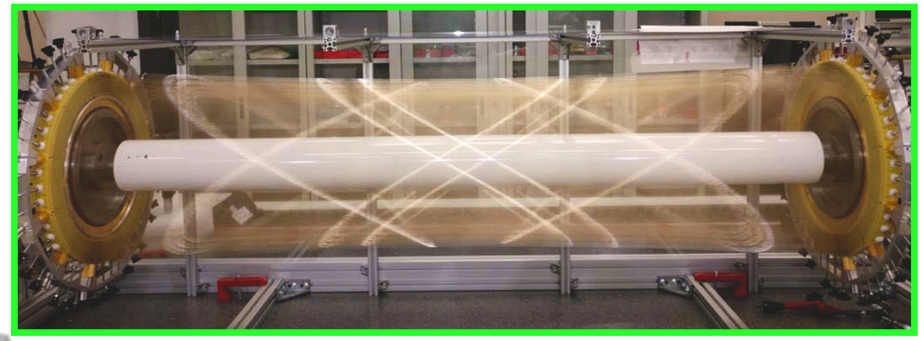
The MEG II Detectors





LXe

Scintillation Detector
4092 **SiPMs** +
668 **PMTs**



Drift Chamber

1728 square **drift cells**

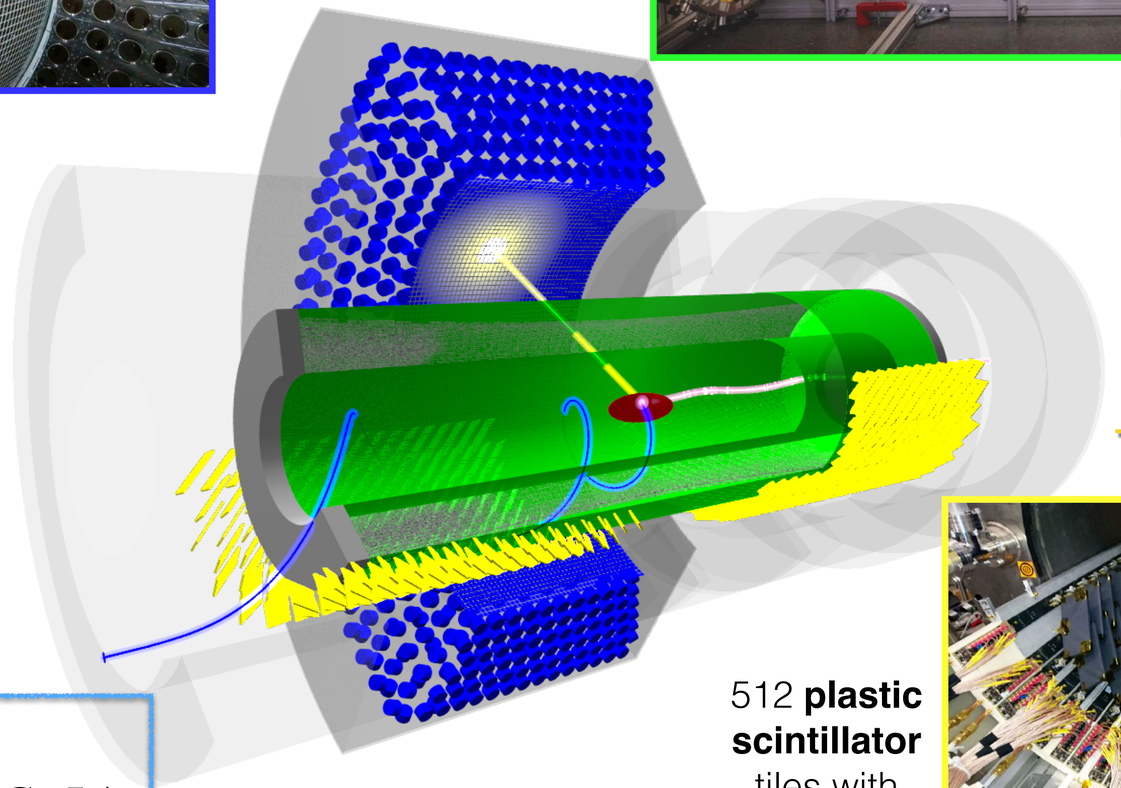
MEG Collaboration aims to observe Lepton-Flavour forbidden $\mu^+ \to e^+\gamma$ **decay**

- Impossible to observe in the SM ($BR \sim 10^{-52}$)
- Very sensitive to **Beyond SM** physics.

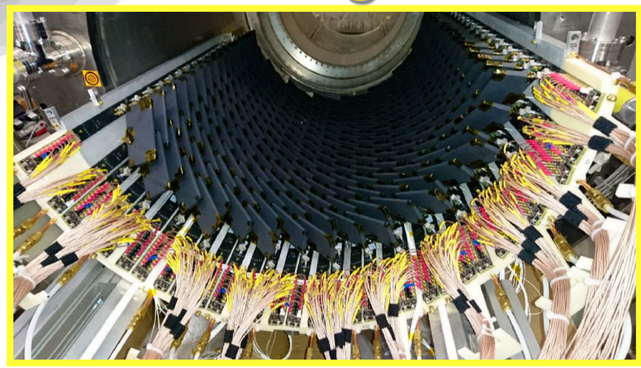
Complementary measurement to $\mu^- \to e^-$ conversion (Comet/Mu2e), $\mu^+ \to e^+e^-e^+$ (Mu3e)

Detection strategy is the same of MEG

"Faster, more segmented" detectors



Timing Counter



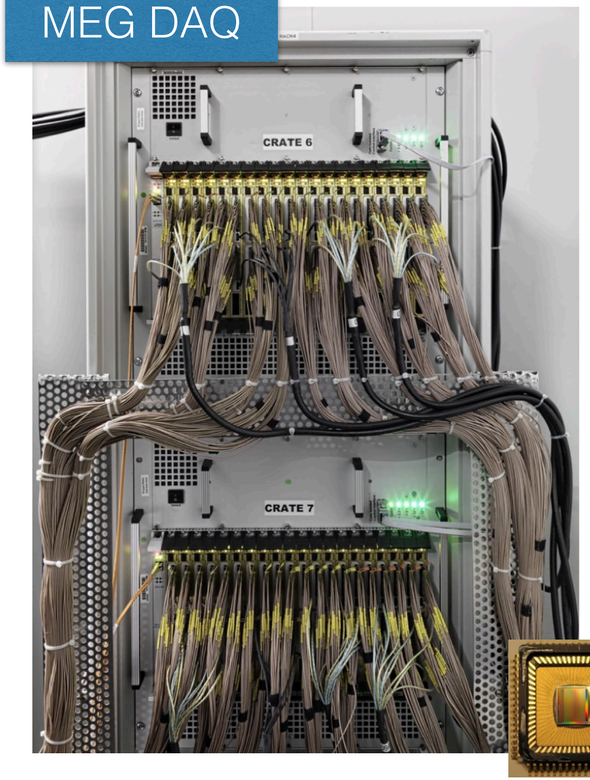
512 **plastic scintillator** tiles with SiPMs.

MEG II final sensitivity

$$BR(\mu^+ \to e^+ + \gamma) \leq 6 \cdot 10^{-14}\ (@\ 90\,\%\ C.L.)$$

The MEG and MEG II Trigger and acquisition system

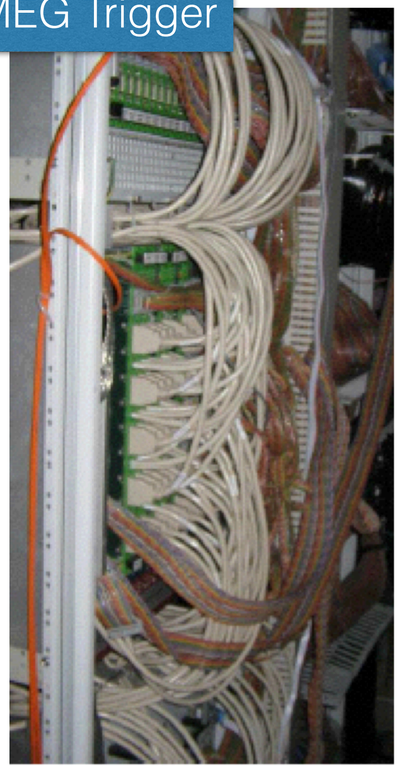
MEG DAQ



5 **VME** Crates (6+3 U) x
640 DRS4 Channels/Crate

Readout limited due to VME Bus

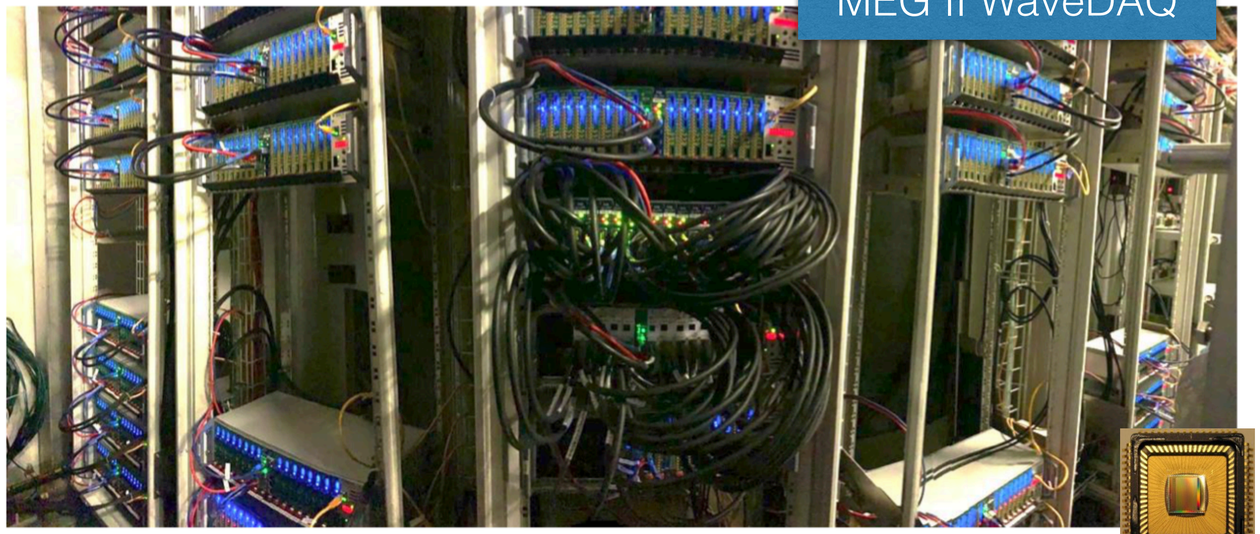
MEG Trigger



2 **VME** Crates with ADC sampling
+
1 **VME** crate for trigger decision

MIDAS VME readout driver

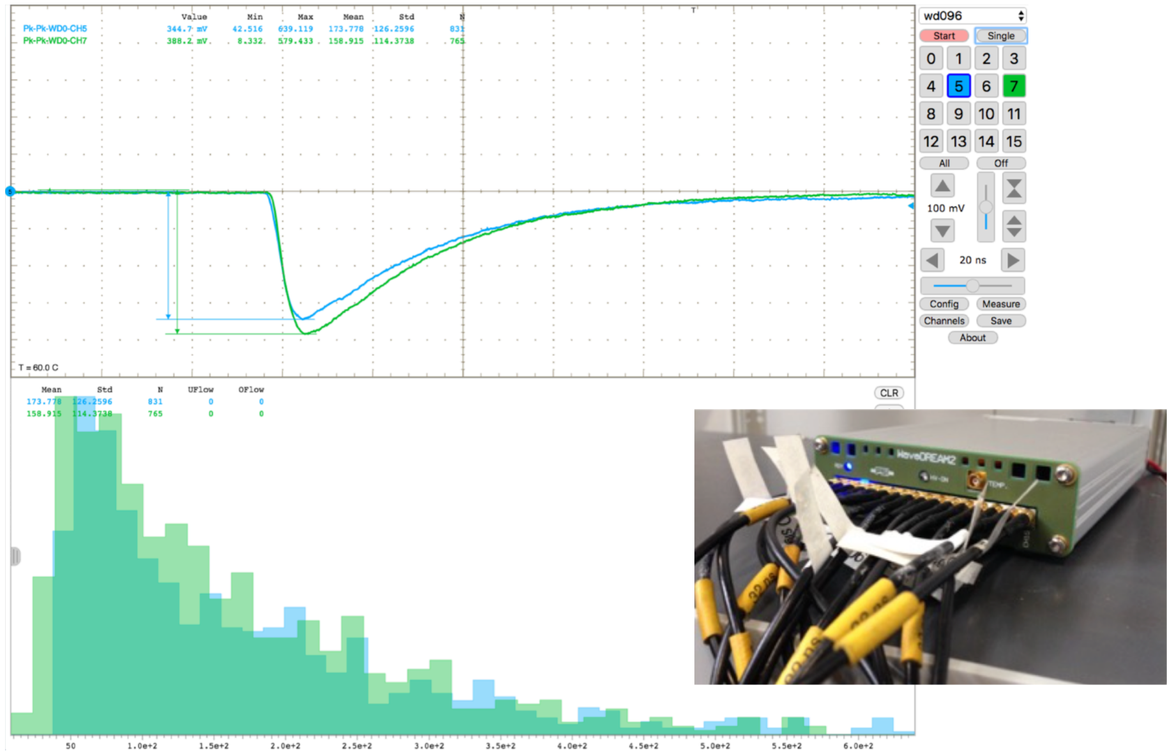
MEG II WaveDAQ



34 Crates (3U) x 256 Channels/Crate
"x3 channel density"

Consolidation:
- DRS4 everywhere
- SiPM biasing
- Configurable amplification (x0.25-x100)
- Integrating both Trigger and DAQ
- “off the shelf" **ethernet** readout
- Only a single backend server

WaveDAQ philosophy



Each board can work as a standalone 16 channel scope directly inside a browser
(WaveDream Server: by Stefan!)
http://elog.psi.ch/scope

Other people (not expert) may use the WaveDAQ boards:
Should work "out of the box"

Limit the requirements on MIDAS code:
only MSC-MXML as git submodules,
to address slow control facilities (HV settings and Crate control)

All DAQ chain was rewritten
for **portability** using C++11 std::thread-std::mutex

```
marco ~/Documents/work/git/wavedaq/sw $ git clone ...
marco ~/Documents/work/git/wavedaq/sw $ mkdir build && cd build
marco ~/Documents/work/git/wavedaq/sw/build $ cmake ..
marco ~/Documents/work/git/wavedaq/sw/build $ make install
marco ~/Documents/work/git/wavedaq/sw/build $ ../bin/wds
WDS starting
WDS GIT revision: Tue May 9 12:49:01 2023 +0200 - d913f08 on branch develop
WDS starting HTTP server at port 8080
Listening on UDP port 60916
```

Worked out-of-box also on Apple M1

Static libraries can then be either used with simple "command line" tools or integrated in more complex environments

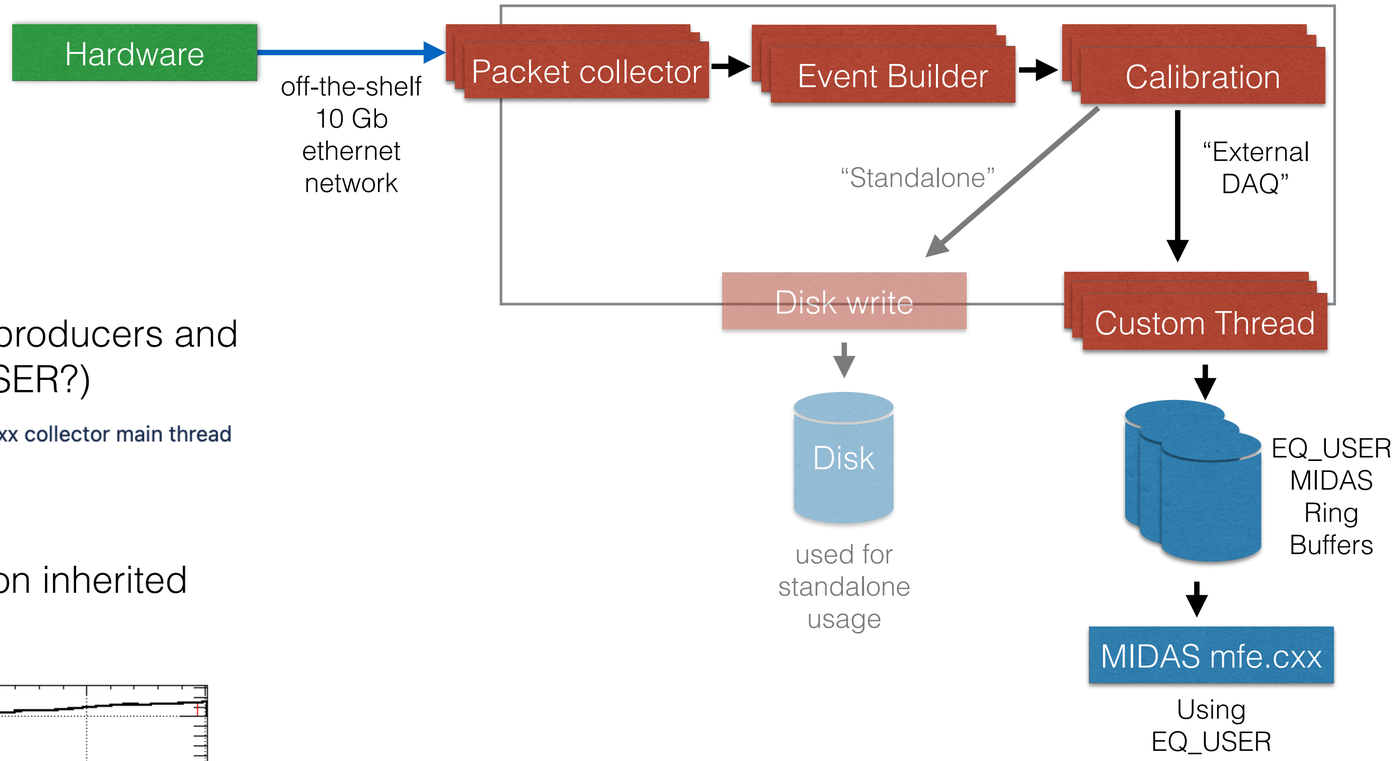
Integration in MIDAS frontend

Developed before new C++ tmfe frontend interface:
old mfe.cxx framework
“EQ_USER”: only expose MIDAS Ring Buffers to user code

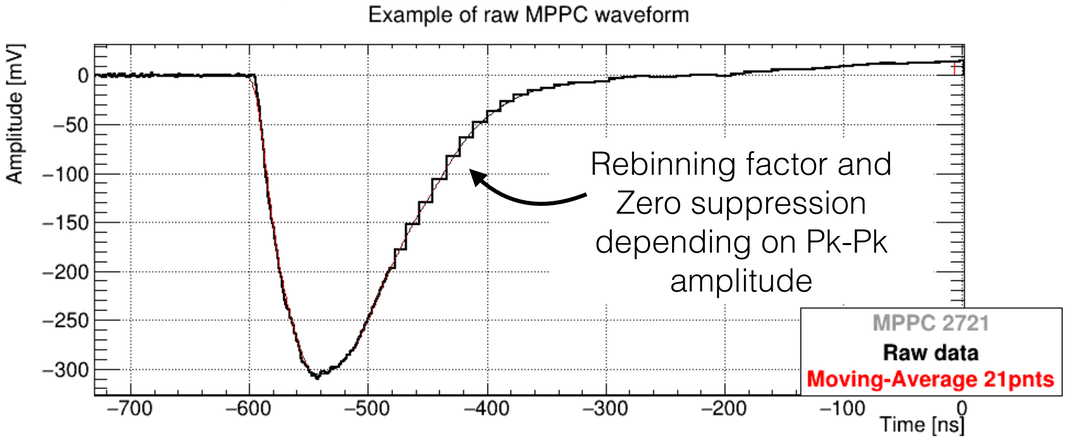
Custom interface threads (only piece linking MIDAS libraries)
write into Ring Buffers, based on mtfe.cxxe xample

Modifications by Stefan to mfe.cxx to support multiple producers and thread-safe bank creation (first user od EQ_USER?)

Stefan Ritt f52c28e Implemented sorting of EQ_USER events in mfe.cxx collector main thread

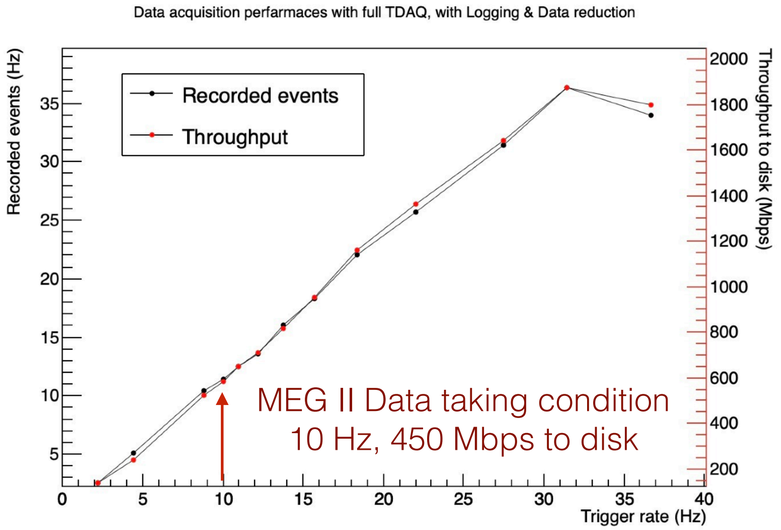


Custom threads also performs also data suppression/reduction inherited from MEG



**Data transmission based on UDP:
packet drops may happens!**

Performances

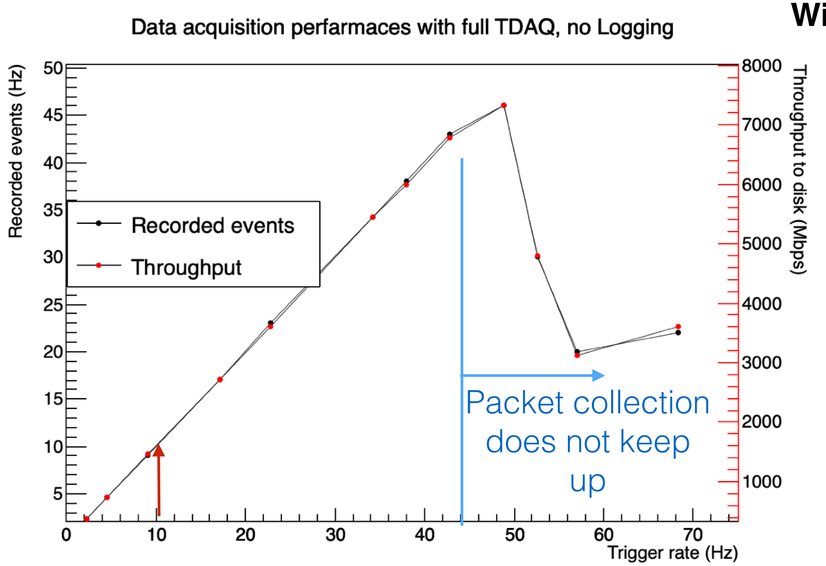


With Logging

NOTE:
Data compression disabled in mlogger,
pbzip do not scale enough

Backend system: 2x Intel Xeon Gold 5218 CPUs @ 2.30GHz
X710 10Gbps Intel NIC
1.8TB NVMe SSD

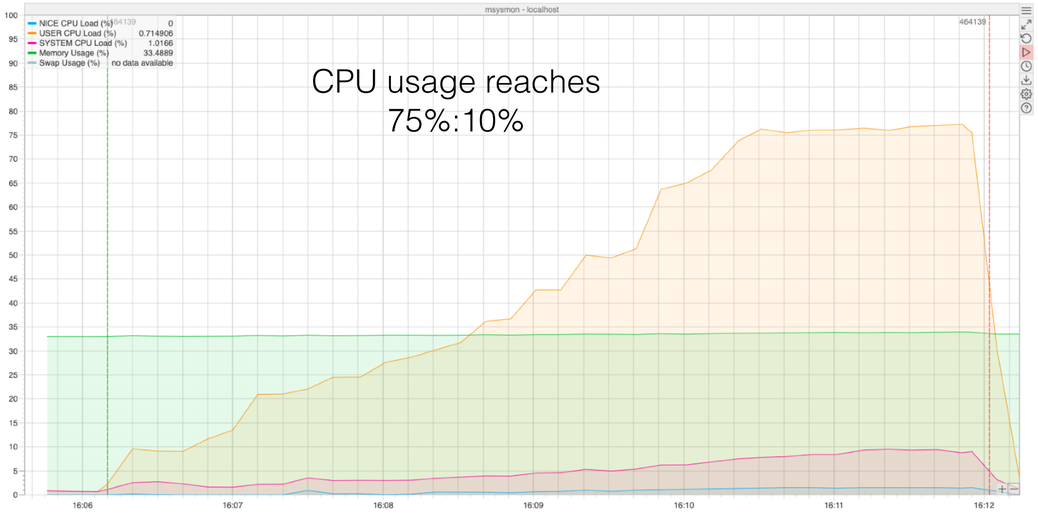
MEG I: Farm of 10 nodes + 1 event builder



Without Logging

At ~50 Hz hardware
limit on the
WaveDAQ side

Takeaway message:
beyond 1 Mpacket/s linux network stack show its limits



Why DAQ speed matter

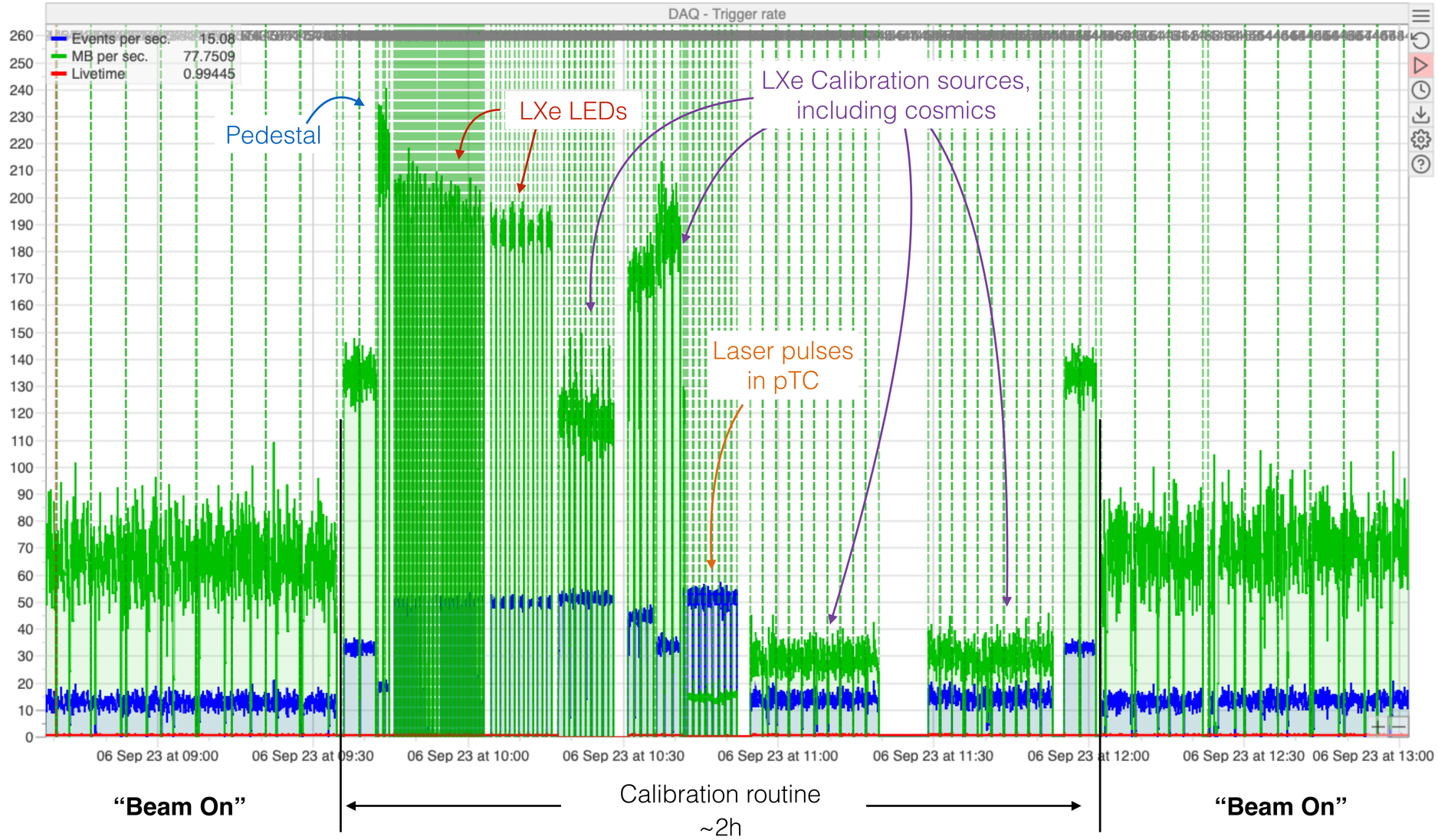
Faster DAQ

Quicker calibration

More “Beam On” statistics

Currently can digest all data produced up to 50 Hz during calibration

Do not overengineer:
x5 wrt physics needs is reasonable



XML Config files, ODB parameters and MSL Libraries

```
<System>
   <Crate Group="PMT" Name="PMT0">
      <WDB Name="PMT0-0" Slot="0">
         <TriggerGain>236,231,0,216,222,226,239,237,0,0,0,0,0,0,0,0</TriggerGain>
         <TriggerLevel>-0.00230506, -0.00642326</TriggerLevel>
         <TriggerTdcOffset>2,0,0,0,0,2,2,2,0,0,0,0,0,0,0,0</TriggerTdcOffset>
         <TriggerTdcMask>ffff</TriggerTdcMask>
      </WDB>
   </Crate>
</System>-
```

At frontend init

Online Database Browser

equipment / trigger / settings / wavedaq / PMT0 / PMT0-0

Key	Value
Slot	0 (0x0)
MscbName	WD838
FrontendGain	*
FrontendPzc	*
FrontendPzcLevel	7 (0x7)
SamplingFrequency	1400 (0x578)
TriggerLevel	*
TriggerGain	*
TriggerTdcMask	0xffff (65535)
TriggerTdcOffset	*
DRSChannelTxEnable	0x0003ffff (262143)
ADCChannelTxEnable	0x0000 (0)
TDCChannelTxEnable	0x0000 (0)
TRGTxEnable	No
ZeroSuppressionEnable	No
SwZeroSuppressionPrescaling	0x0000 (0)
SwZeroSuppressionClkROIStart	0x0000 (0)
SwZeroSuppressionClkROIEnd	0x03ff (1023)
SwZeroSuppressionSignalROIStart	0x0000 (0)
SwZeroSuppressionSignalROIEnd	0x03ff (1023)
SwZeroSuppressionRegionStart	0x0000 (0)
SwZeroSuppressionRegionEnd	0x03ff (1023)
SwZeroSuppressionRebinFactor	0x0000 (0)
SwZeroSuppressionRebinSignalThr	0
SwZeroSuppressionThr	0
SwZeroSuppressionRebinThr	*
	[0] 0
	[1] 0

During commissioning it was very hard to follow TDAQ board reassignment:
List of values from external XML File

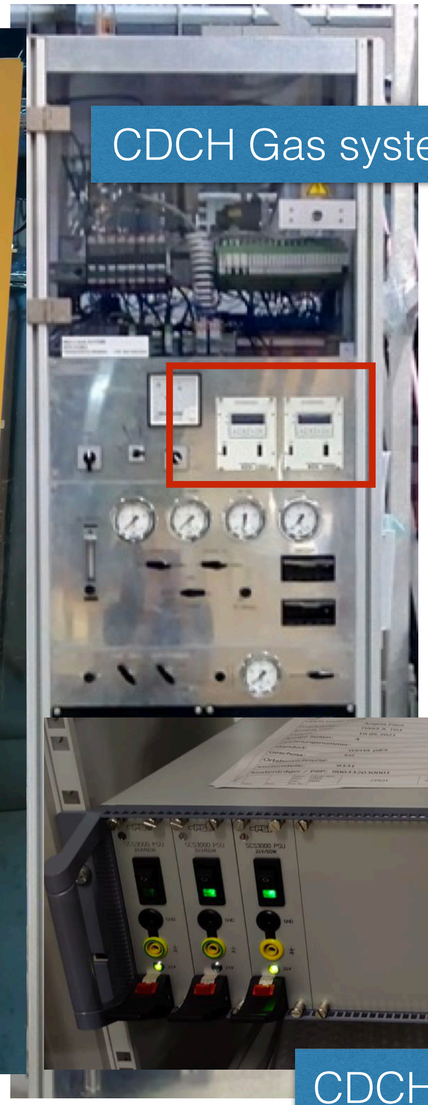
Same file for standalone uses and with MIDAS

In particular:

- Automatic population of ODB (~14000 parameters)
- Script for generation of Experiment-wide MSL Libraries:
  `CALL SetPMTTriggerLevel, -0.1`
  instead of
  `ODBSET /Equipment/Trigger/Settings/WaveDAQ/PMT*/*/TriggerGain[*], -0.1`
  (In the best case)
- External script can calculate best thresholds without access to ODB
  and merge them with main XML file

Slow control

LXe cryogenics slow control



CDCH Gas system

CDCH Slow control

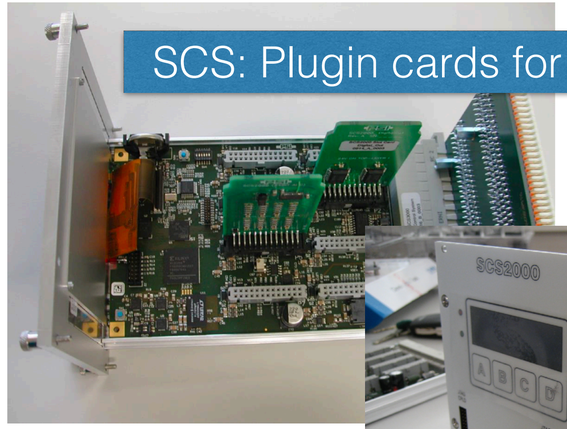
Large use of SCS2000 and SCS3000 from PSI using **MSCB** over ethernet

More than 20 units in use (monitoring & HW loops)

+34 MSCB endpoints for the WaveDAQ crates
+3 MSCB endpoints for old PMT HV dividers
+EPICS for beamline control
+few non-standard device (ISEG, Raspberry PIs, LXI)

600 GB of MIDAS History files since 2015

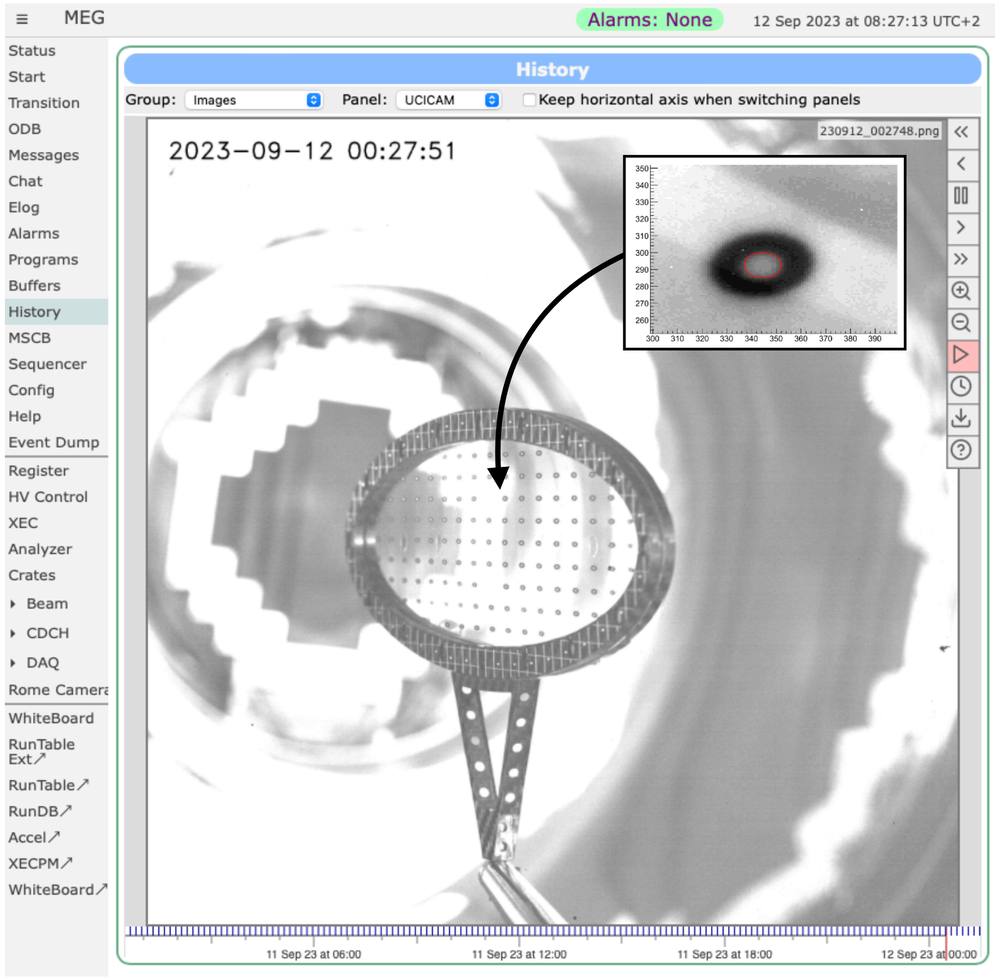
SCS: Plugin cards for different I/O



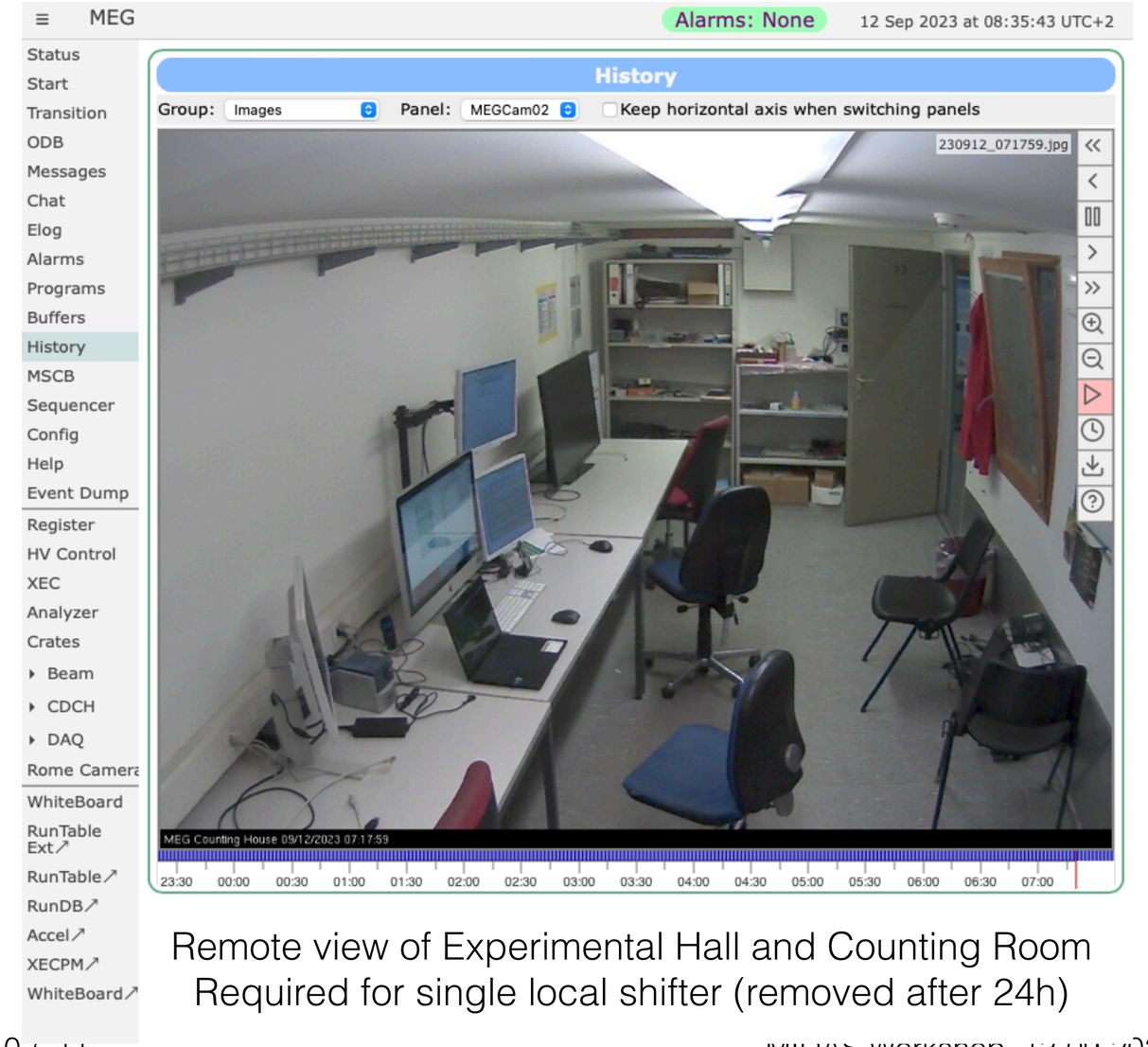
Large use of custom pages for monitoring and action:
Very rare need for a shifter to use SSH

Image History, use cases

Previously using custom frontend (fetching images) + custom page (remote view)



Photogrammetric measurement of target position and deformation



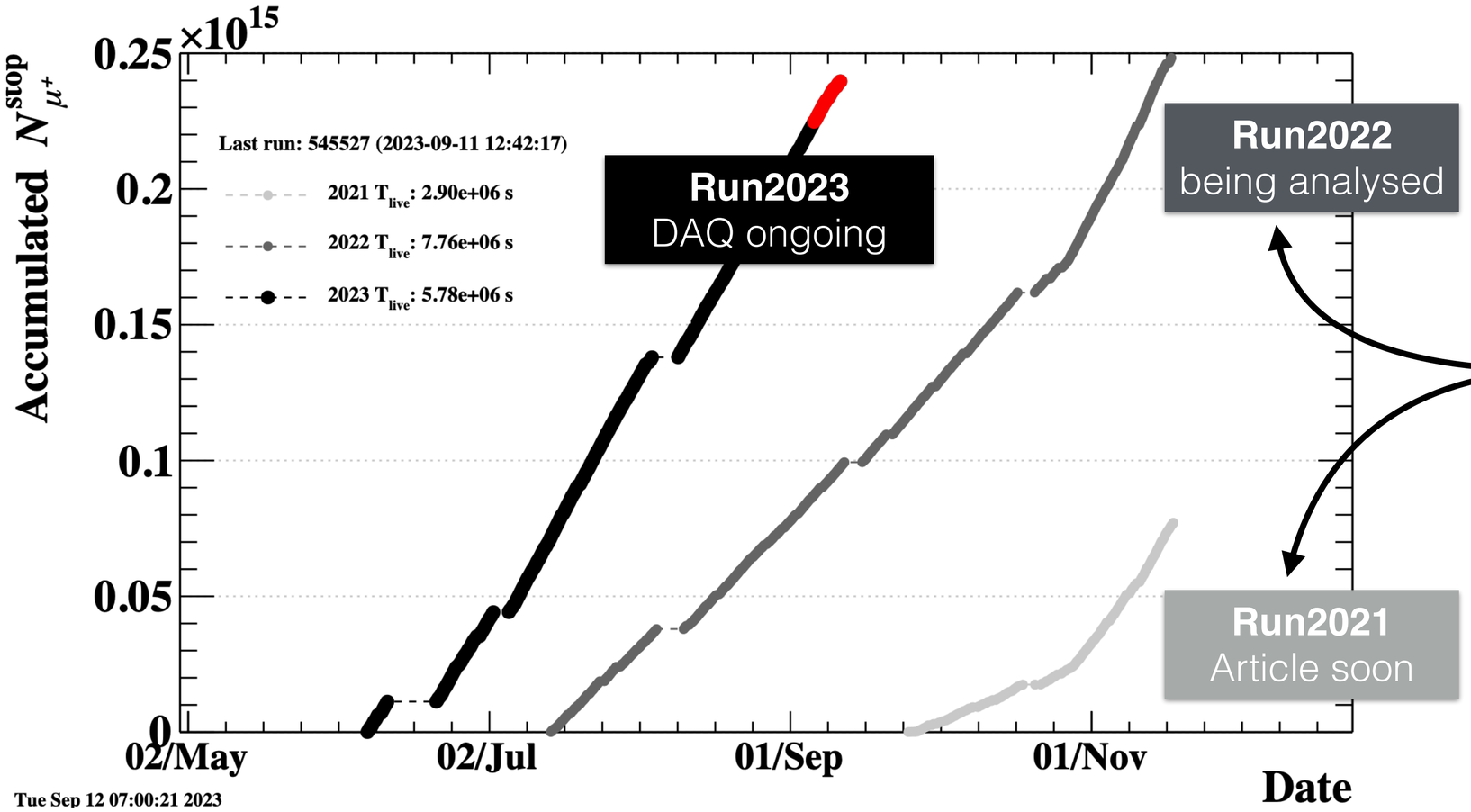
Remote view of Experimental Hall and Counting Room
Required for single local shifter (removed after 24h)

Conclusions

MEG II is currently taking data, approximately 500 TB/Run (100 TB/month)

Experiment commissioning and first physics runs during covid!

DAQ expected to end in 2026



During covid time: remote shifts were crucial!