DAQ: status and future

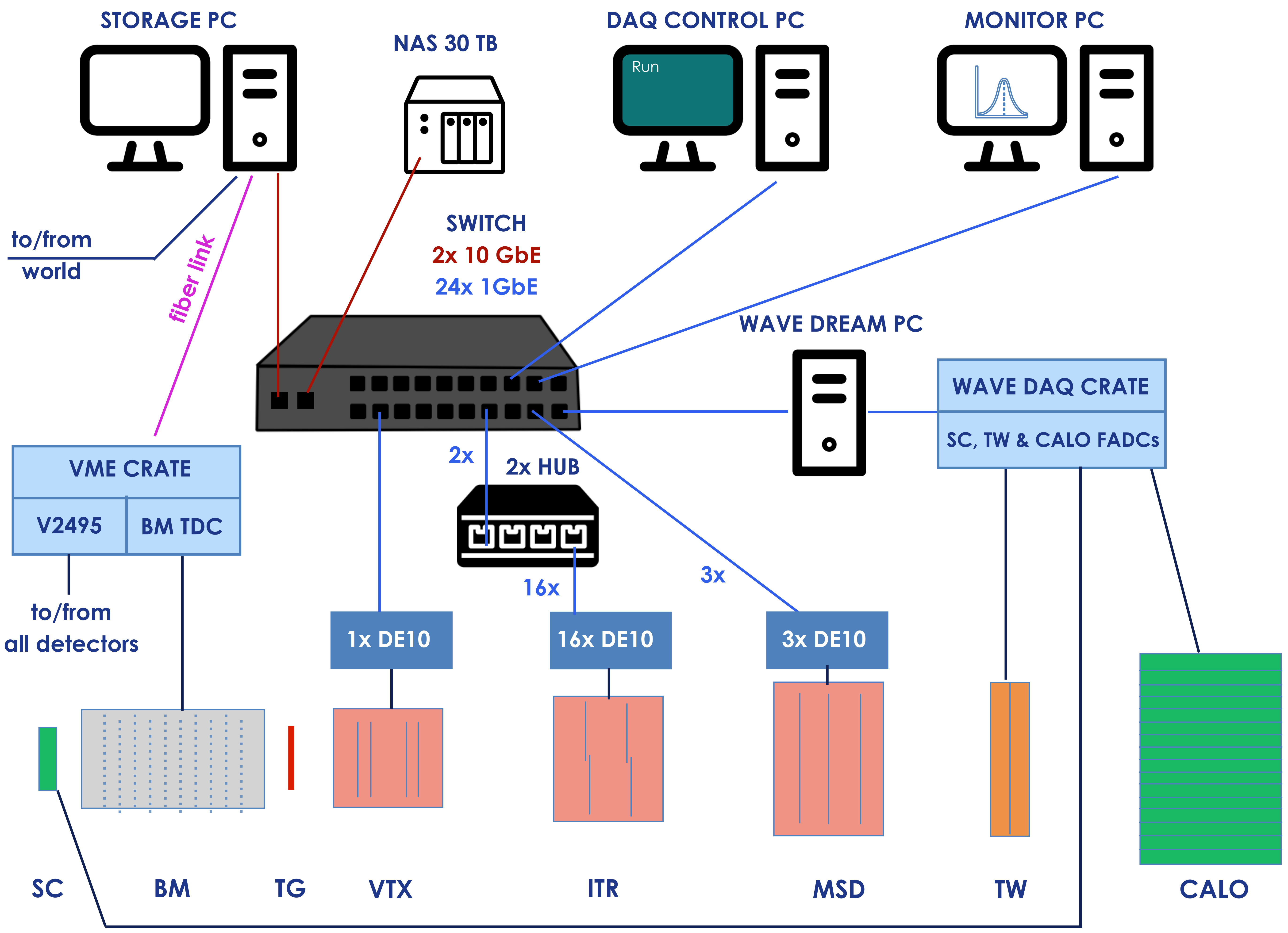
Silvia Biondi, Mauro Villa, Riccardo Ridolfi
University & INFN of Bologna

Outline

- The DAQ system
- Status of DAQ integration
- What we are working on now
- What is still missing
- Conclusions

New DAQ structure

- **VME crate:**
  - Trigger and BM boards
- **3 central DAQ PCs:**
  - for steering the whole process, storing the data and monitoring the runs both online and offline
- **detector integrated readout systems:**
  - SC, VTX, IT, MSD and TW
- for some sub-detectors we plan to have an **intermediate PC** to interface with the central DAQ:
  - Wave DAQ;
  - VTX and MSD used now;
  - collect data, to store additional information which is not included into the final data format and to reduce the size of the event.

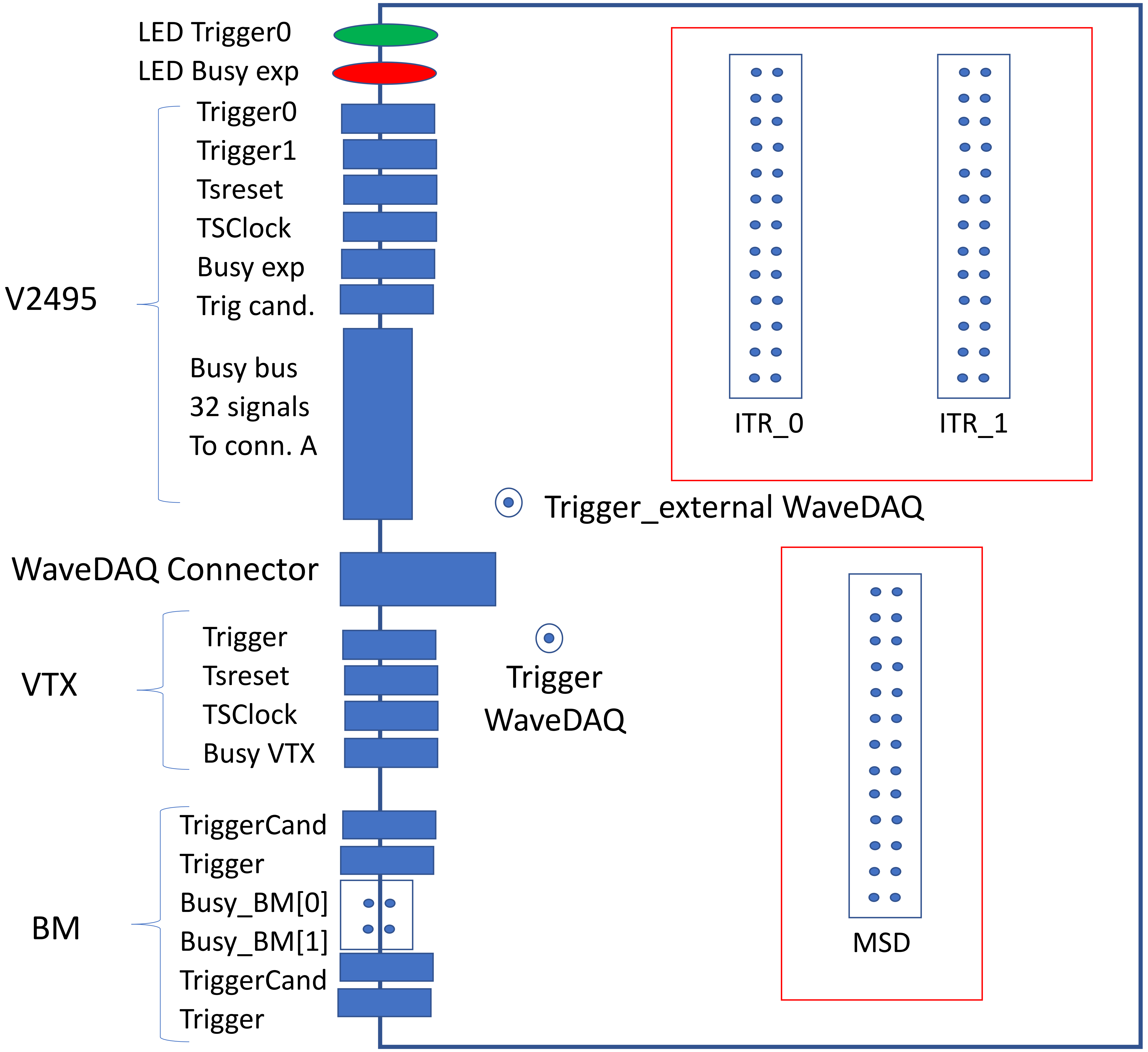


Status of DAQ integration

Sub-detector	What we will use	What we need to work on	From which institute	What we have now
Start Counter	Wave Dream	PC interface	Roma+Pisa	working system
Beam Monitor	TDC	parameters for board configuration	Milano+Roma	TDC (V1190B)
Vertex	DE10 (std)	software for TCP connection (CPU)	Frascati	DE10 (nano)
IT	DE10 (nano)	software for TCP connection (CPU)	Frascati	DE10 (nano)
Micro Strips	DE10 (nano)	software for board connection (CPU + FPGA)	Perugia	DE10 (nano)
DE/TOF	Wave Dream	PC interface	Roma+Pisa	working system
Calorimeter	Wave Dream	PC interface	Torino	working system

Patch panel design and development

- **Reason:** distribution of trigger, timestamp and busy signals necessary to synchronise all the sub-detectors and to handle the trigger;
- **different types of signals** are designed for different sub-detectors necessities;
- designed size like a **VME 6U board**, so we can put it in the VME crate where the power can be steered by the crate itself;
- went through all the boards and readout systems of each sub-detector to finalise the **amount of signals to be received and distributed by the patch panel**.
- **ITR and MSD** are treated in the same way (with same connectors)
  - each of them will provide an **interface board to receive and send signals**.



- **STATUS:**
  - produced but with some errors being fixed, not usable now;
  - changes to be introduced in the framework for its usage are already implemented and being tested;
  - schedules and tests slowed down by covid and restrictions.

MSD DAQ

- Working on a general FPGA+CPU framework together with detector experts;
- strong collaboration to finalise the whole setup from the sensors to the DAQ;
- start thinking about online monitoring.
- **Workplan:**
  - **started a joint DAQ-MSD HW tests**
    - intermediate PC for tests has just been configured;
    - a standalone version of the DAQ system has been installed for foreseen remote tests;
    - final touches and (quick) tests need to be done before sending back to Perugia;
    - after the remote tests (for long runs) work smoothly, we need to switch to in-person-tests in order to check all the cables and connections also with patch panel included.

VTX/IT DAQ

- (remote) integration in Bologna started;
- 2-day test beam with DAQ+VTX was not possible due to covid;
- start thinking about online monitoring
- **Workplan:**
  - **started a joint DAQ-VTX HW tests**
    - intermediate PC for tests in Frascati;
    - a standalone version of the DAQ system has been prepared (same as for MSD) for foreseen remote tests;
    - finalising configurations from both parts;
    - after the remote tests (for long runs) work smoothly, we need to switch to in-person-tests in order to check all the cables and connections also with patch panel included.

Work-plan for other systems

From the last FOOT Collaboration meeting, no updates have been done for these items yet

- **Beam Monitor:**
  - HV to be added in central DataBase;
  - improve online monitoring.
- **WaveDAQ:**
  - improve online monitoring;
  - test event building to check no event loss happens anymore;
  - more controls on threading;
  - starting procedure needs to be automatic!
- **Calo:**
  - ~~readout system yet to be designed (320 channels read by FADCs 17xx);~~
  - ~~DAQ will be tuned on estimations from the designed setup.~~

Readout with WaveDAQ

Online monitoring

- **GNAM**
  - need to add all the info from different detectors;
  - possibility to check directly comparisons between different systems.

- **Online Histograms (OH)**
  - some already implemented, with very basics parameters to check;
  - for now, event size (for all the integrated modules) and time measurement and hit channels for BM.

- **Information Service (IS)**
  - widely used so far to check detector-specific parameters during the run;
  - each Readout Module can set the frequency of parameters updating.

- **DataBase (DB)**
  - used to check some interesting parameters at the end of the run;
  - for now only WaveDream readout module uses this tool.

- All this info can be **used by shifters during data taking**
  - not to overload the DAQ shifters too much (from GSI experience);
  - all detector expert should be "shifters" to check their own info.

Data flow - connection

connections going from the same client to the same server

Readout module = CLIENT

1. send **configuration** parameters to server
2. request **monitoring parameters** from server
3. tell the server **when to send data**
4. put the data in the relative **DataChannel** through parallel threads

"Slow" connection

starts when "Configure" mode and uses > 1 thread to handle different simultaneous activities

"Fast" connection

only if the system goes in "RUN" mode,
stops when "stopDC" mode

Sub-detector = SERVER

1. receive **configuration** from client
2. send **monitoring parameter** to client
3. set a device parameter to "run" status and **prepare itself for the data flux**
4. **send the data** to the client

Slow control - Slow connection

connections going from the same client to the same server

Readout module = CLIENT

1. send **configuration** parameters to server
2. request **monitoring parameters** from server
3. tell the server **when to send data**
4. put the data in the relative **DataChannel** through parallel threads

"Slow" connection

starts when "Configure" mode and uses > 1 thread to handle different simultaneous activities

"Fast" connection

only if the system goes in "RUN" mode, stops when "stopDC" mode

Sub-detector = SERVER

1. receive **configuration** from client
2. send **monitoring parameter** to client
3. set a device parameter to "run" status and **prepare itself for the data flux**
4. **send the data** to the client

this is handled exclusively by the Slow connection, so it has no effect on the Fast connection for the data flow!

Online monitoring

- **Working on**
  - slow-control system for online monitoring;
  - some meetings with detector experts (before June) to define how to monitor the temperature/voltage of different sensors in the DAQ system;
  - also plan to increase the information stored in the DB, both for initial and end-of-run configuration of all the sub-detectors.

- **SC**
  - significant variations in temperature during the run not expected, no need to monitor the temperature.

- **MSD**
  - for each DE10: monitoring currents and voltages is necessary, temperature is expected to be monitored as well;
  - detector experts will check (and likely test) temperature sensors used by TW.

- **TW**
  - temperature already stored in monitoring for this sub-detector since GSI;
  - the number of needed sensors yet to be defined (max 10 sensors);
  - an intermediate board needs to be designed and placed between sensors and WD boards.

- **Calo**
  - variations in temperature expected to be not negligible;
  - proposal to have one sensor for each crystal, but yet to be defined.

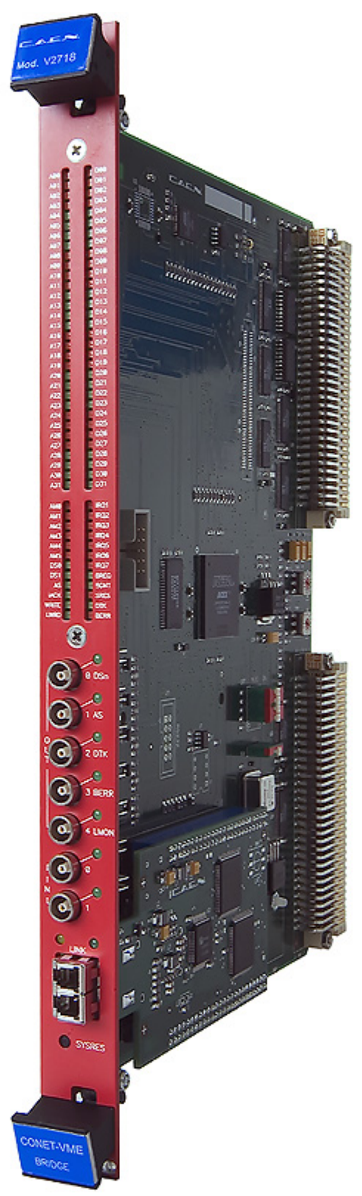
Conclusions

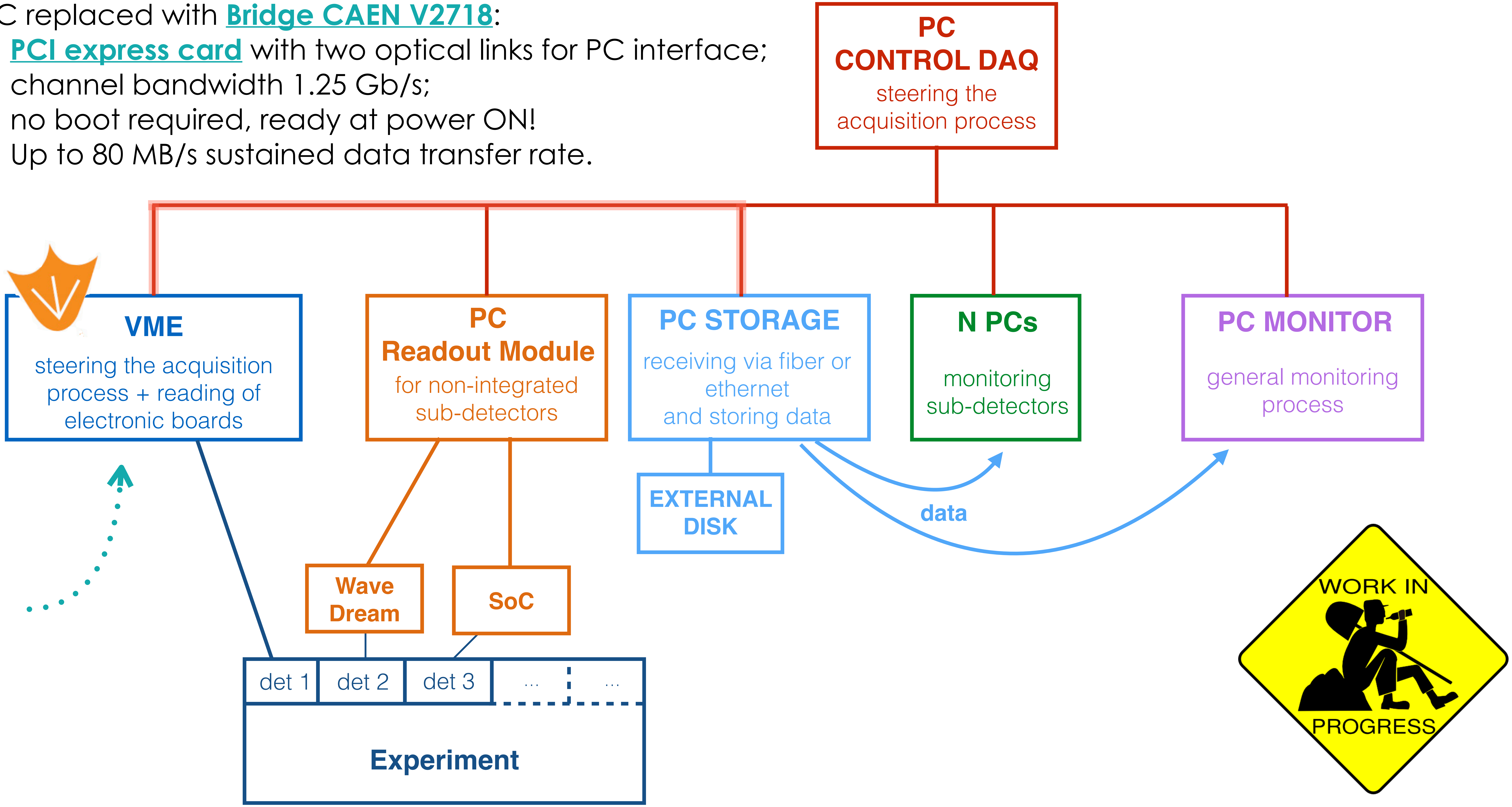
- **All the lab activities have been stopped due to the Covid situation**
  - no in-situ integrations or joint activities for sub-detectors.
- **WD system integration**
  - still some work needed but in good shape.
- **VTX and MSD integration**
  - ongoing with remote tests with intermediate PCs;
  - waiting for possibility to make more detailed HW tests.
- **Other systems (IT, Calo)**
  - several tests needed with detector experts;
  - need to define many technical points for all these systems;
  - integration will proceed as soon as they will be available (and when we are allowed to meet again):
    - we need to plan for a long integration phase well in advance ($\geq$ **5 months**) wrt a test beam.
- **Works ongoing to finalise the DAQ structure (both HW and SW)**
  - online monitoring;
  - reducing levels in DAQ structure;
  - patch panel finalisation for the whole system.

Supporting material

New DAQ structure

- SBC replaced with **Bridge CAEN V2718**:
  - **PCI express card** with two optical links for PC interface;
  - channel bandwidth 1.25 Gb/s;
  - no boot required, ready at power ON!
  - Up to 80 MB/s sustained data transfer rate.





Reminder: from our CDR

from CDR

Detector	Board(s)	DAQ channels	max event rate (kHz)	Event size (bytes)
Trigger	V2495	1	10	40 B
Start Counter	DreamWave	4	1	8.2 kB
Beam Monitor	TDC	36	5	0.1 kB
Vertex detector	SoC on DEx	$4 \cdot 10^6$	2	0.9 kB
Inner tracker	SoC on DEx	$28 \cdot 10^6$	2	2.1 kB
Outer tracker	Custom	$6 \cdot 10^3$	2	0.5 kB
$\Delta E/\Delta x$	DreamWave	80	1	8.4 kB
Calorimeter	QDC	400	2	1.7 kB
Total DAQ	Storage PC	-	1	22 kB

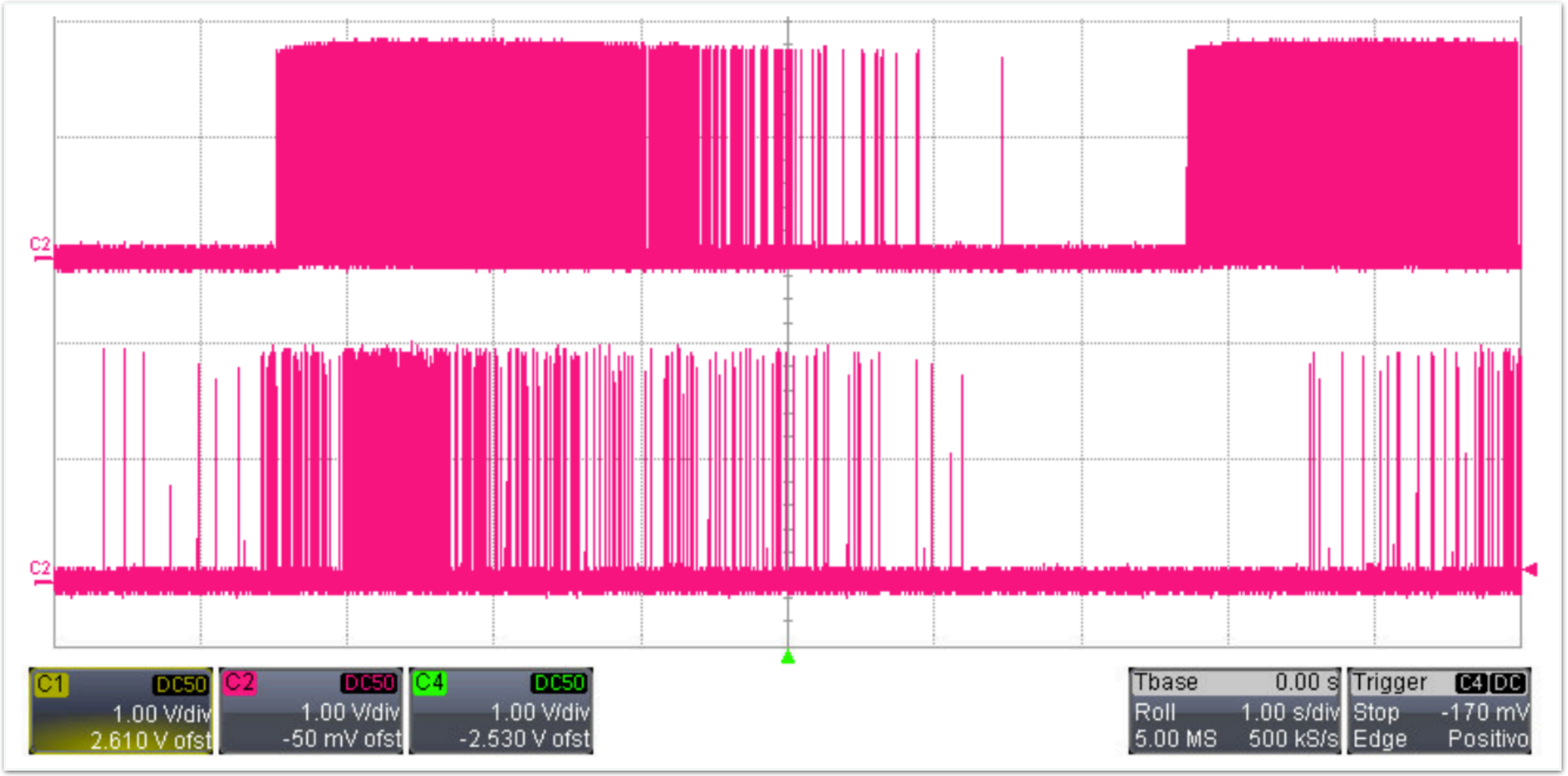
- Numbers from GSI experience
  - DAQ (trigger+BM+file structure): 530 B
  - VTX: 650 B
  - SC+TOFW: 29 kB

→ 30 kB

Beam simulator

- Simulation of the beam we had at GSI;
- **more realistic environment** for debugging systems integrations;
- **simulations are not enough.**

- 6-8 s periods, varying intensity, random trigger.



Patch panel design and development

CAEN V2495

D connector
Input signal: Trigger_Candidate – TTL - LEMO

E connector
Output signals: TTL – LEMO
Trigger0
Trigger1
TSReset (Time Stamp reset)
TSClock (Time Stamp Clock)
Busy_Experiment

A connector
Input signals: LVDS
BusyBus (32 signals)

Wave DAQ/Wave Dream

FCI connector 8 coppie - LVDS
Output signals:
- Trigger_Candidate
- Busy_WD
- Trig_marg

Input signals:
- Trigger_ext
- TSReset
- TSClock
- Busy_Experiment

Beam Monitor, CAEN V1x90, V895

V1x90
Input signal:
- Trigger1 – LEMO - NIM

Output signal:
- Busy_BM – ECL – 110 Ohm (x2)

V895
Input signal:
- Trigger Candidate – LEMO - analog

VTX, IT, MSD - DE10 board or similar (DE10 nano)

VTX
Input signals:
- Trigger0, TSReset, TSClock – LEMO – TTL

Output signal:
- Busy_VTX – LEMO - TTL

IT, MSD → connettore 2x13
Input signals:
- Trigger0, TSReset, TSClock – LVDS

Output signals:
- Busy_ITR[0..15], Busy_MSD[0..5] - LVDS