

Integration of CALICE DAQ in common DAQ

*Vincent Boudry**

Frédéric Magniette, Miguel Rubio-Roy

École polytechnique, palaiseau

Taikan Suehara

Kyushu U.

Jiri Kvasnicka, Mathias Reinecke

DESY

Rationale of SW for common DAQ

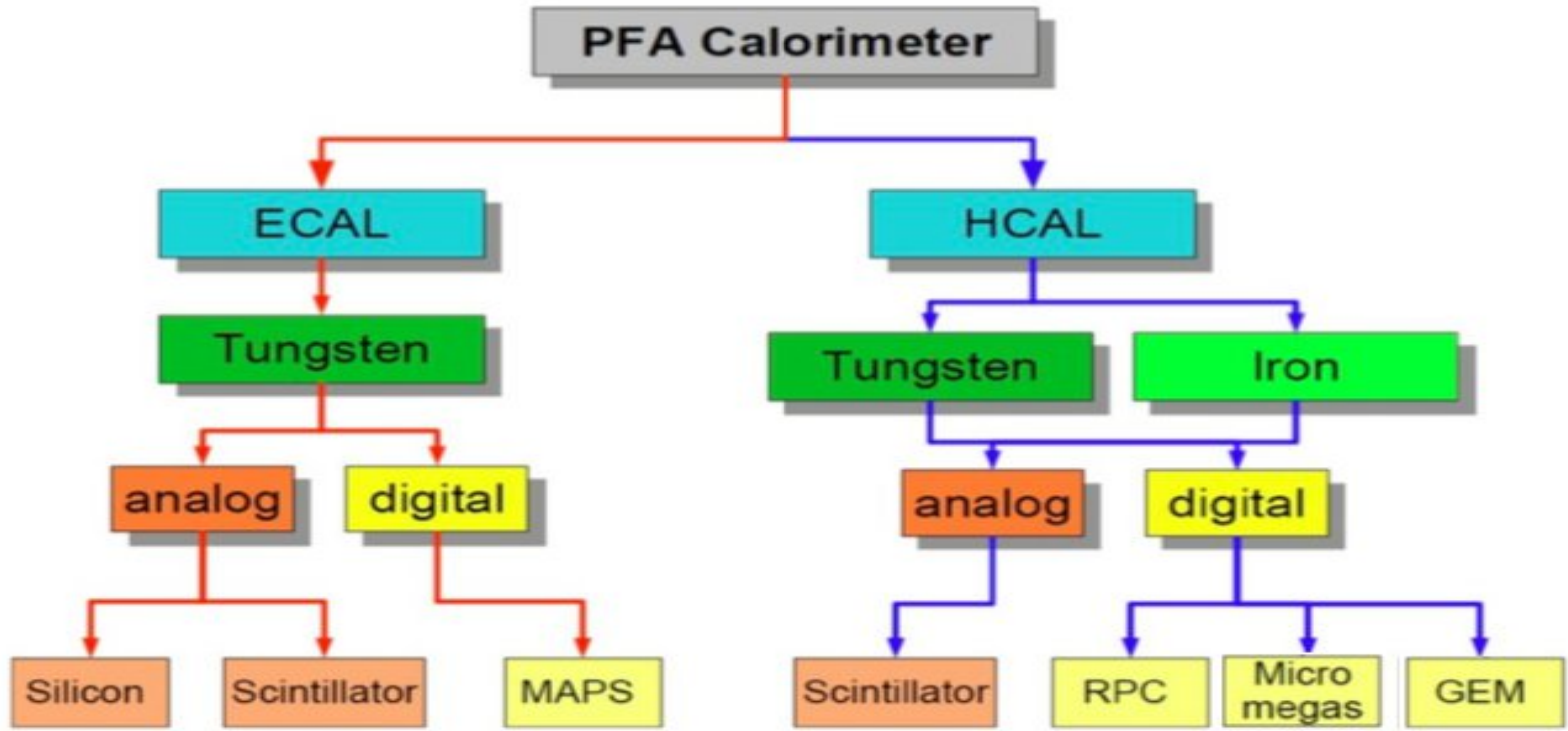
Merge Acquisition code with minimal modifications

- **common global control** → EUDAQ
 - ⊃ global monitoring:
 - DAQ: data fluxes, event numbers, number of readout elements
 - Physics on a subsample (tuned not to slow the DAQ):
 - » Histograms of critical quantities
 - » event display of a sub-subsample
- **common data containers** → LCIO
 - data stream should not necessarily go on a single computer (bottleneck)
 - but at least minimal data for monitoring

Sociology: within CALICE the DAQ SW strongly diverged (last 3 years).

- **Base from 1st generation (ex: ConditionDB, Reconstruction framework) exits**
- **Technological prototypes (new HW) require new low level SW (interface to HW)**

CALICE DAQ landscape



SKIROC2
 ASU
 ECAL-DIF
 HDMI+HDMI
 PYRAME/CALICOES
 LCIO
LLR (Palaiseau)

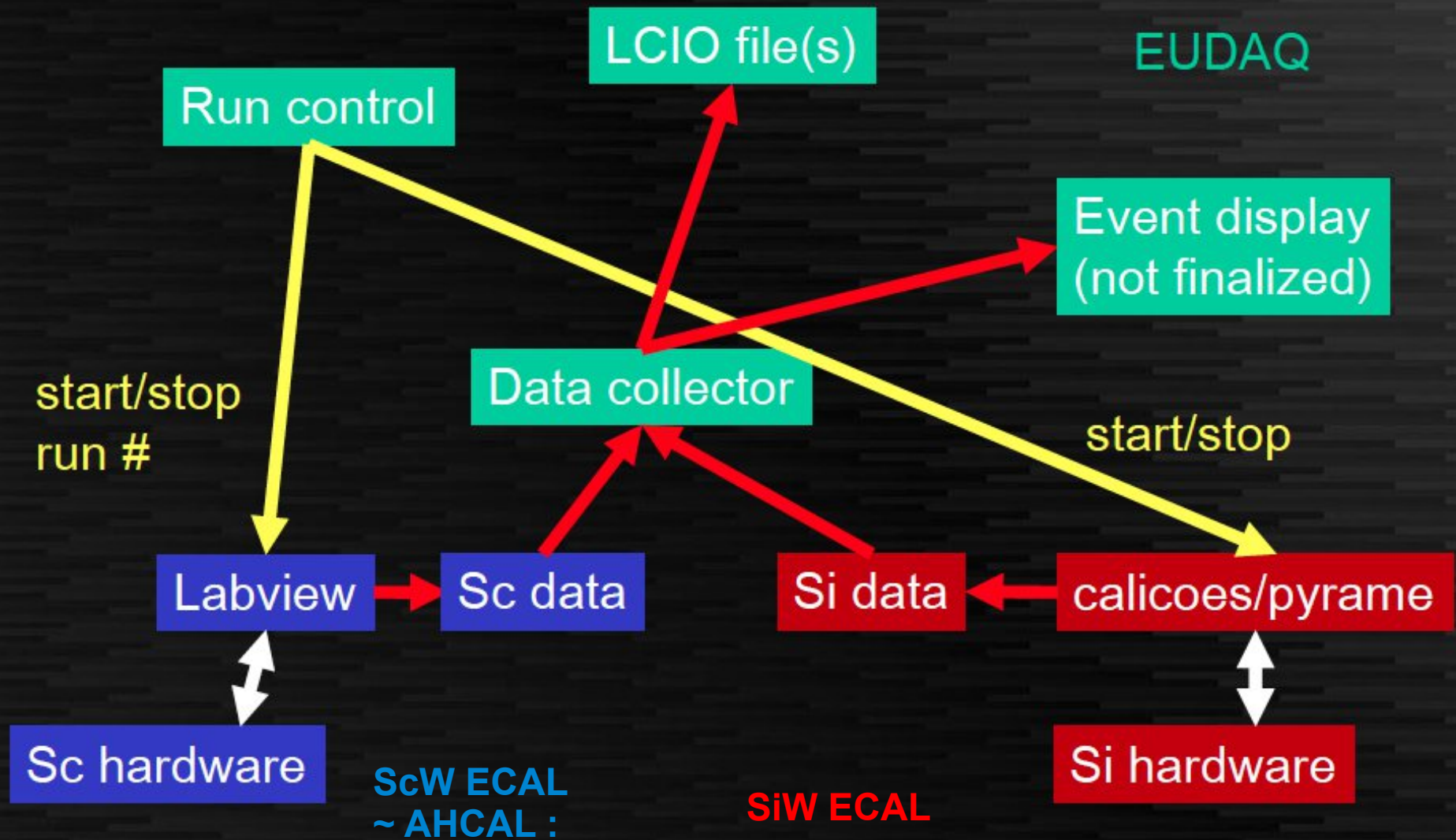
SPIROC2
 EBU / HBU
 AHCAL-DIF
 HDMI+USB
 LabView
 Raw/LCIO
Kyushu + DESY

HARDROC / MICROROC
 "PCB"
 SDHCAL-DIF
 HDMI+USB
 XDAQ
 LCIO
IPNL (Lyon)

← Same interface
 PCB
 InterFace
 Connection
 SW
 ← Same data format

Integration with EUDAQ

Combined DAQ for Si + Sc

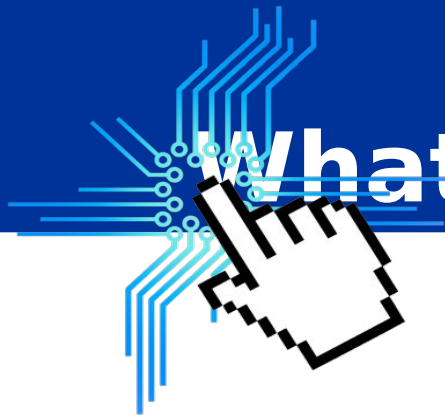


CALICE-SiW ECAL

Early version used here

Re-written last summer and released: Frédéric Magniette & Miguel Rubio-Roy (LLR)

- a “generic” low level SW → PYRAME
- a CALICE SiW-ECAL specific part → CALICOES



What's new in Pyrame ?

New configuration module : keep in real time an analog of every configuration values in the system → able to save the running configuration at any time

New bindings : C++, Labview, Xdaq

New generic class : ps (power supply), pg (pulse generator) to reduce the specificity of the drivers

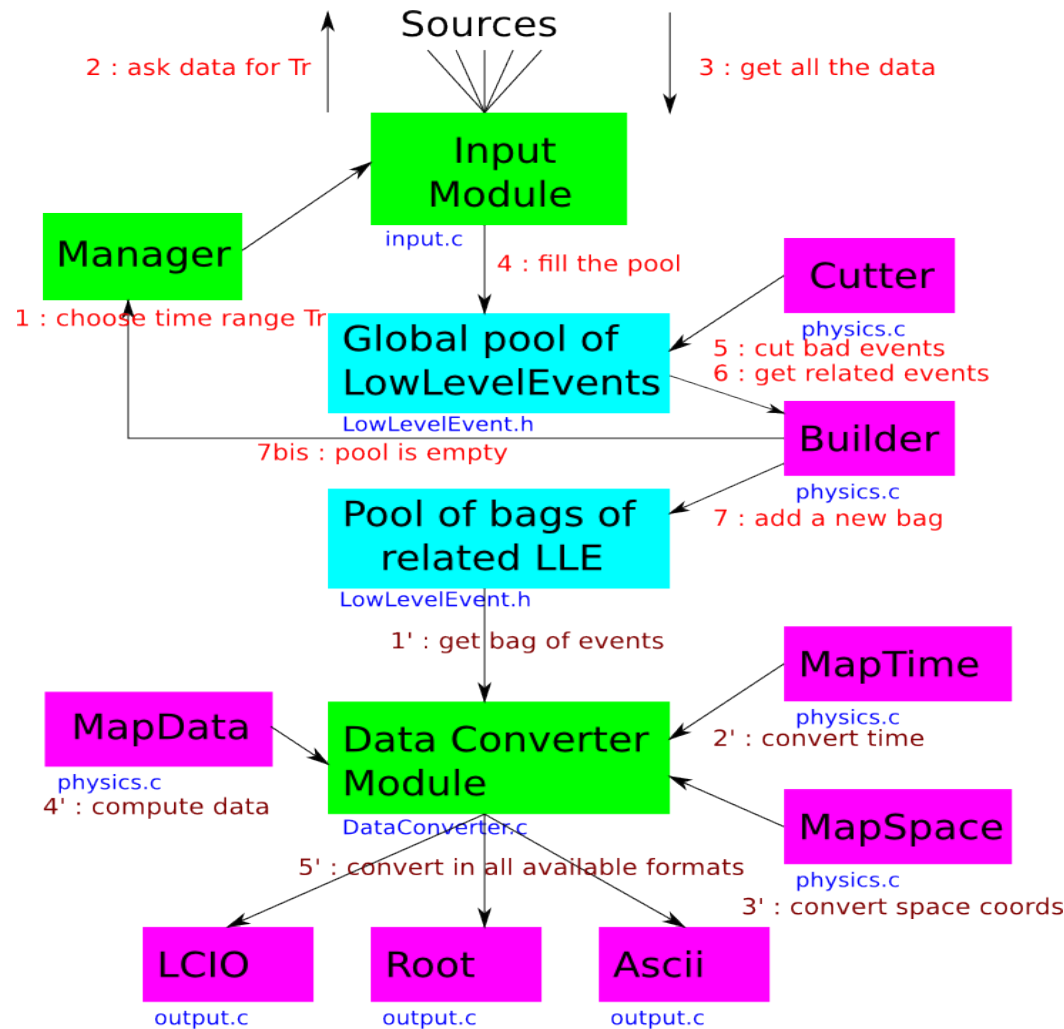
Lot's of new drivers (oscilloscopes, new power supplies, new pulse generators)

The acquisition time is now multi-media through a plugin system supporting (Raw ethernet, TCP client and server , UDP and USB)

A time tagging system has been implemented to share clock between multiple channels (extraction and injection)

The CalXML tool is now completely generic (not related to calicoes anymore)

What's new ? : the event builder

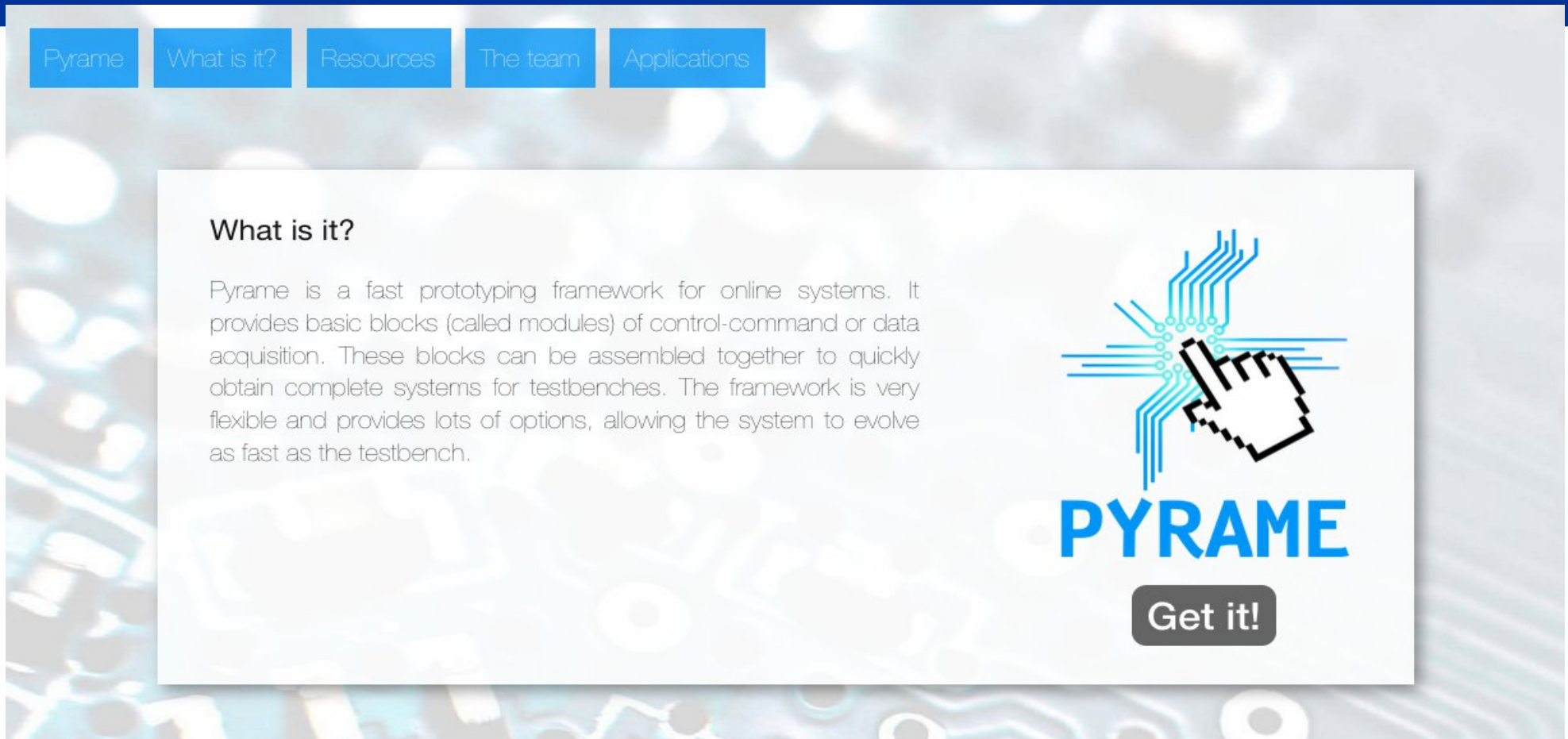


Online Event Builder

Tran Trong Hieu, Frederic Magniette

- Legend
- Generic Module
 - Specific Module
 - Data structure
 - Code File
 - Building actions
 - Converting actions

What's new in Pyrame ?



The screenshot shows the Pyrame website with a navigation bar containing five tabs: 'Pyrame', 'What is it?', 'Resources', 'The team', and 'Applications'. The 'What is it?' tab is selected. Below the navigation bar, the 'What is it?' section is displayed, featuring a description of Pyrame as a fast prototyping framework for online systems. To the right of the text is the Pyrame logo, which consists of a stylized hand cursor pointing to a central point where several blue lines representing circuit traces converge. Below the logo, the word 'PYRAME' is written in large, bold, blue capital letters, and a dark grey button with the text 'Get it!' is positioned below the name.

Pyrame

What is it?


Resources

The team

Applications

What is it?

Pyrame is a fast prototyping framework for online systems. It provides basic blocks (called modules) of control-command or data acquisition. These blocks can be assembled together to quickly obtain complete systems for testbenches. The framework is very flexible and provides lots of options, allowing the system to evolve as fast as the testbench.



PYRAME

Get it!

Complete online documentation

– Available as open-source software (LGPL Licence): <http://lr.in2p3.fr/sites/pyrame/>

Some publications (TWEPP'13, IN2P3 Computing letter)

External users : CPPM in Marseilles



What's new in Calicoes ?

Lot's of internal modifications to cope with Pyrame V2

Lot's of modifications in the high level part to ease the use :

- Removal of the OFF state, unused
- Full support of the **multiple PCs acquisition**
- New **loading/saving configuration primitives** through integration of Pyrame V2 config module. Ability to dump the whole configuration in a single file at any time
- Automatic configuration saving at startacq
- New naming system of the components to be able to call actions on the slab names instead of the dif number
- A **GUI** for the high level part of the detector (state machine and loading/saving config + statistics)

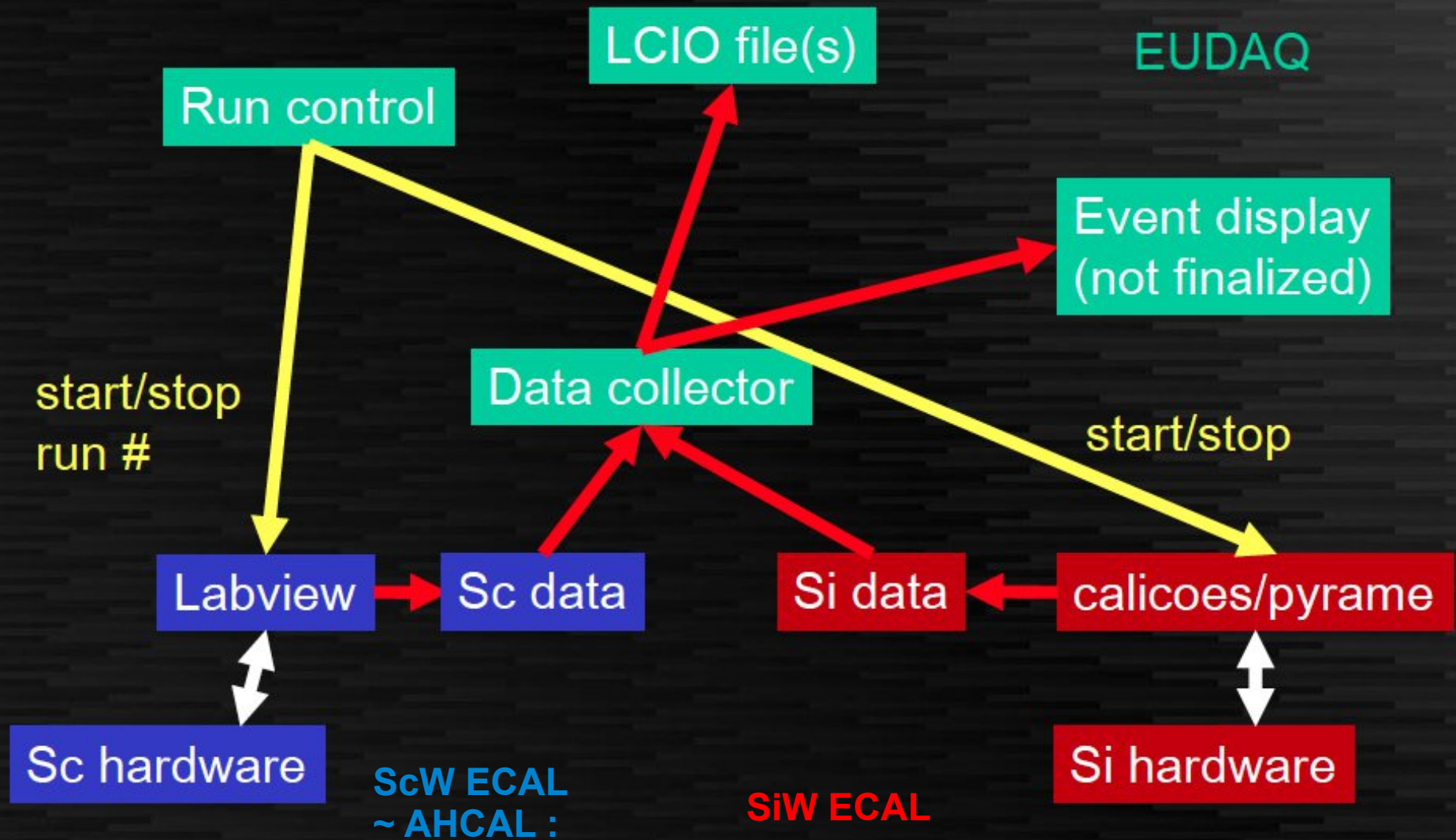
The Software test-beam

Cosmic test-beam :

- Basic test (1 slab) : success
- Stress test (6 slabs on a GDCC) : GDCC is very robust (0% of packet loss)
- Repartition test (2 PC with 2 GDCCs and 3 slabs for each) : full validation of the distributed scheme
- Frequency scanning (scan the spill freq and see corruption status) : done (no analysis yet)
- Calibration survey :
 - fast calibration algorithm to pycaldaq
 - Fully automatic S-curves implementation
- Complete cosmic calibration :
 - Scurve with noise
 - Comparison of the different calibration algorithm
- Cosmic data taking (if it remains time)

Integration with EUDAQ

Combined DAQ for Si + Sc



EUDAQ in action with calos

Technically Successful
Combined Test Beam at
CERN last week

- 1 layer of SiW ECAL (ASU)
- 3 layers of ScW ECAL (EBU)
- 12 Layers of AHCAL (24 HBU)

Using EUDAQ v1.3.1
(v2 “too complicated”)

The screenshot displays the EUDAQ v1.3.1 software interface. The main window is titled "eudaq Run Control v1.3.1" and contains several control panels. The "Control" panel shows the configuration "141202-combrun" and buttons for "Config", "Start", "Log", and "Stop". The "Status" panel displays "Run Number: (20550)", "Events Built:", "Rate:", "Triggers:", "File Bytes: 0 B", "Particles:", and "TLU Status: Scalers:". The "Connections" panel shows a table of active connections:

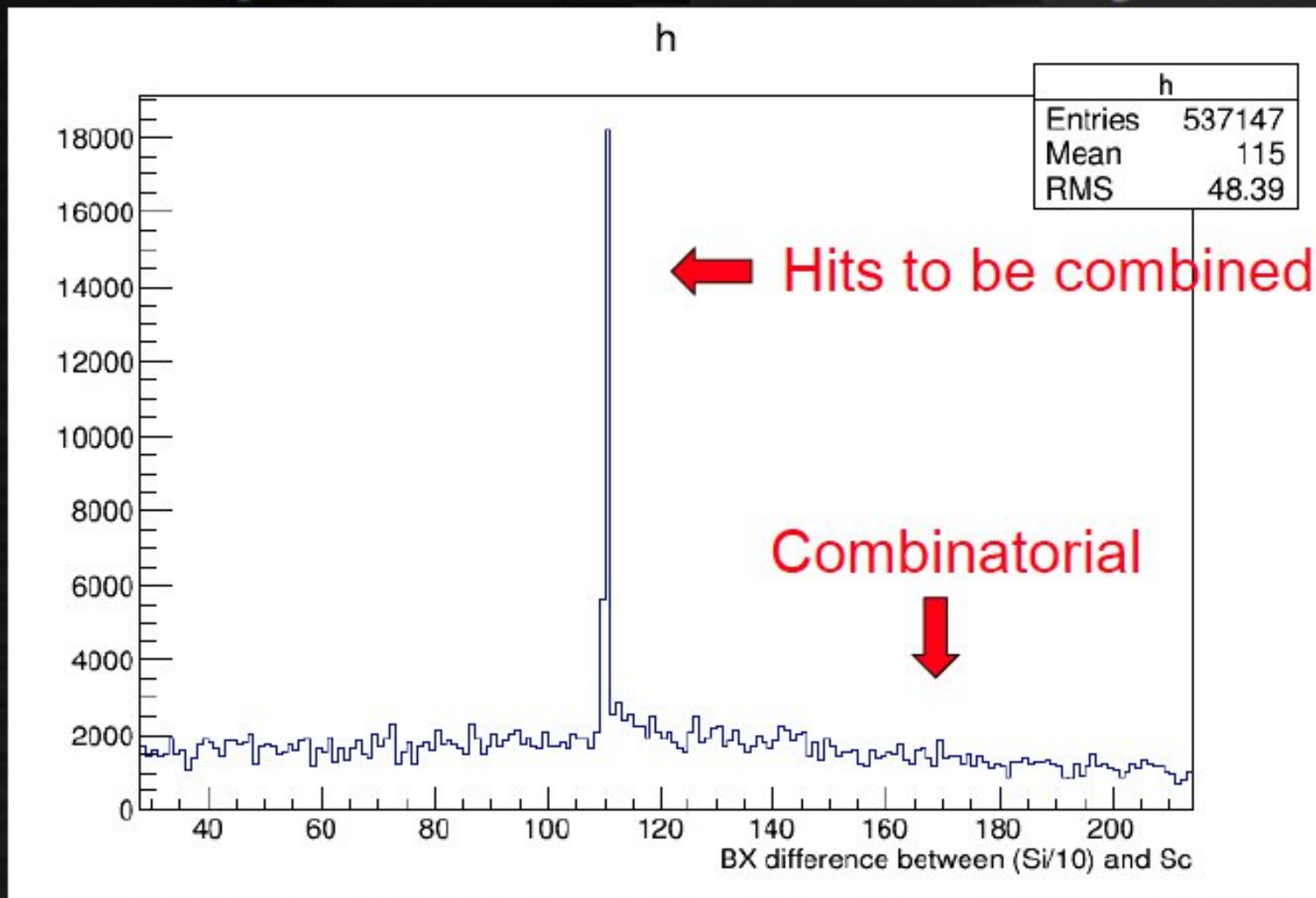
type	name	state	connection
DataColle...	CaliceDot...	OK: Confi...	127.0.0.1:45810
LogCollec...	CaliceDot...	OK: Confi...	127.0.0.1:45807
Producer	CaliceSi	OK: Confi...	127.0.0.1:45815
Producer	CaliceSc	OK: Confi...	127.0.0.1:45812

Below the main window is the "EUDAQ Log Collector" window, which shows a log of network connections and configuration events. The log entries include:

```
Received ^ Sent Level Text From File Function
06:16:02... 06:16:02... 4-INFO Connection from LogCollector (127.0.0.1:35999) LogCollec... eulog.hh:98 OnConnect(const eudaq::ConnectionInfo&)
06:16:06... 06:16:06... 4-INFO Connection from DataCollector.CaliceDataCollect... LogCollec... eulog.hh:98 OnConnect(const eudaq::ConnectionInfo&)
06:16:08... 06:16:08... 4-INFO Connection from Producer.CaliceSc (127.0.0.1:36... LogCollec... eulog.hh:98 OnConnect(const eudaq::ConnectionInfo&)
06:16:08... 06:16:08... 4-INFO Connection from Producer.CaliceSi (127.0.0.1:36... LogCollec... eulog.hh:98 OnConnect(const eudaq::ConnectionInfo&)
06:16:08... 06:16:08... 4-INFO Connection from Producer.CaliceSc (127.0.0.1:37... DataColle... DataColle... OnConnect(const eudaq::ConnectionInfo&)
06:16:08... 06:16:08... 4-INFO Connection from Producer.CaliceSi (127.0.0.1:37... DataColle... DataColle... OnConnect(const eudaq::ConnectionInfo&)
06:16:27... 06:16:27... 4-INFO Configuring (141202-combrun) RunControl RunContro... Configure(const string&, int)
```

Courtesy of T. Suehara

A quick result – BX sync



Muon beam, All Sc hits to be compared with Si hits
~ 1000 Readout cycles accumulated

Steps forward

After much circonvolution convergence process seems to resume...

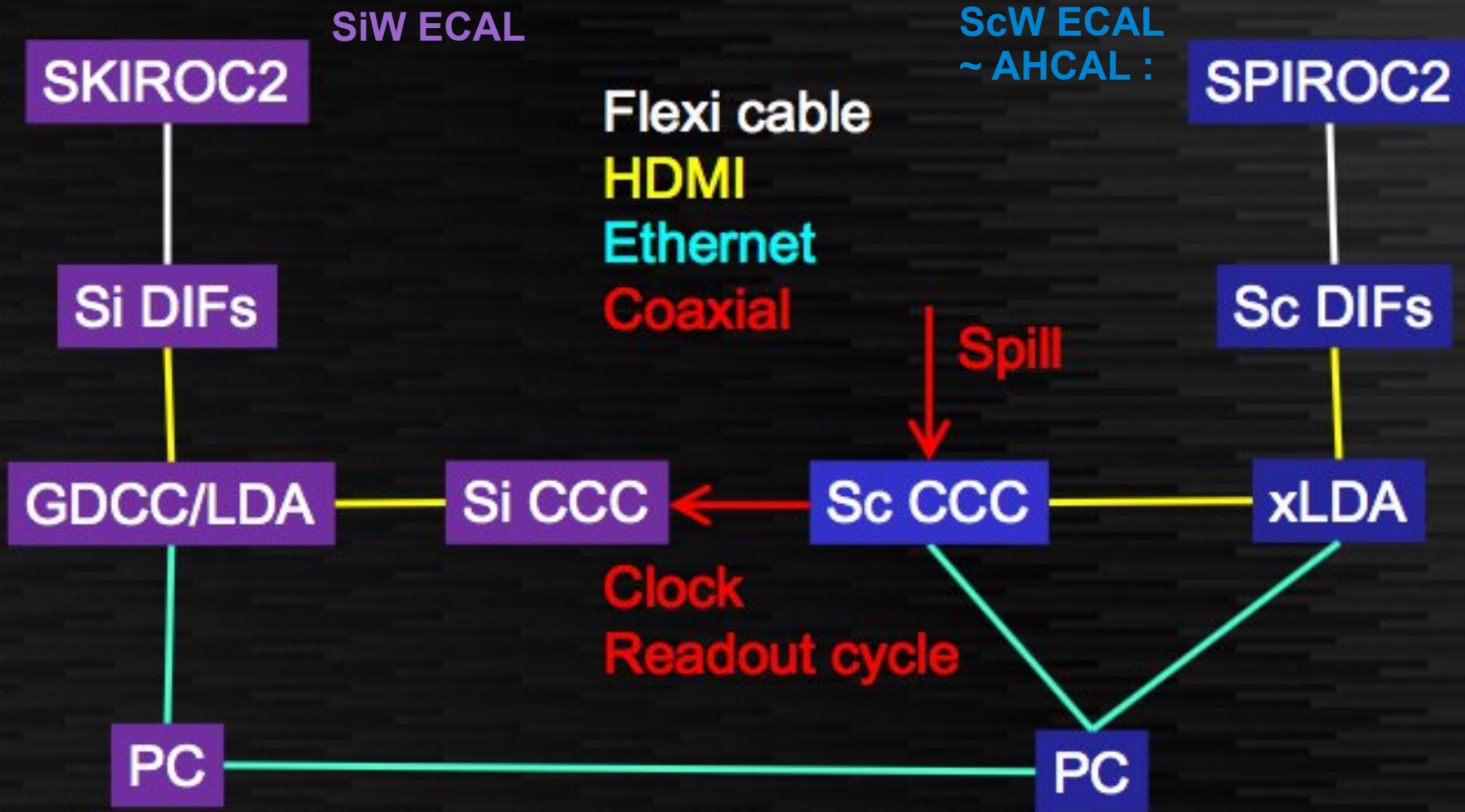
- A DAQ Task Force has been set-up (launch meeting 1 hour (!) ago) to recollect the effort within CALICE.

Technically:

- Move to EUDAQ2
 - Integrate latest version of PYRAME+CALICOES
 - Develop Monitoring and Event Display
 - Interface with historic CALICE tools (CondDB, logging, ...)
- Develop Configuration tools
 - Several tools exists: to be extended for larger set-up

Extras

HW sync



Synchronization of clock/spill

- Independent CCC board for Si and Sc
 - Difference on fast command specification (should be fixed later)
- Clock - Sc clock output connected to Si
- Spill & busy
 - Sc CCC creates output of “Readout cycle (RC)” TTL signal by (Spill & !Busy) to Si CCC
 - All busy from Sc layers combined at Sc CCC
 - Si converts RC up to start, down to stop
 - Si busy is not treated