Discussion on DAQ Command & Control

Vincent Boudry

École polytechnique

LLR / Japan meeting 16/04/2014

Needs for DAQ2

For SiW ECAL test and, immediate needs are

- Debug & Basic Monitoring (online)
  - Shape to be defined...
  - Some tools exists, ... ~extensive list here.
- Configuration tools
  - Several tools exists: to be extended for larger set-up

For Testbeam and AIDA & longer term (~fall ?)

- Integration in a common interface
  - For future would precognise EUDAQ.
  - Other tools already exist...

Online Monitoring

Needs ~ 4 levels:

- Acquisition (R&D, GDCC validation)

  ⇒ Stats from PC and cards (rates/counters), check of a reproducable signal [injection], step-by-step validation of connectivity

- SLAB functionnality: transmission of signals, connection with ROC

  + step-by-step validation of config loading,

  + “basic” data analysis with ~ stable config (& tools to set-it).

- SLAB performances (cosmics, injection)

  + complex data analysis, scripting

- TestBeams:

  + long term monitoring of key parameters, logging, run keeping, ConditionDB

Statistics tools

DAQ stat ⇒ in shared mem (ext. screen printing & in histograms)

- Statmod from Pyrame (push on subscription)
  - Calicoes/Pyrame counters
    - Number of lost packets at each level
  - nb of packet (total & /DIF)
    - data_size / RO (total & / DIF)
      - nb of ROC / RO (total & / DIF)
      - nb ROC evts / RO (total & / DIF)
- Post-run stats (commands). → on GUI (PhyGUI … C++ work in progress...)
  - Card registers:
    - DIF counters (with enough length to avoid overfill)
    - GDCC registers
  - External counters (via Pyrame)

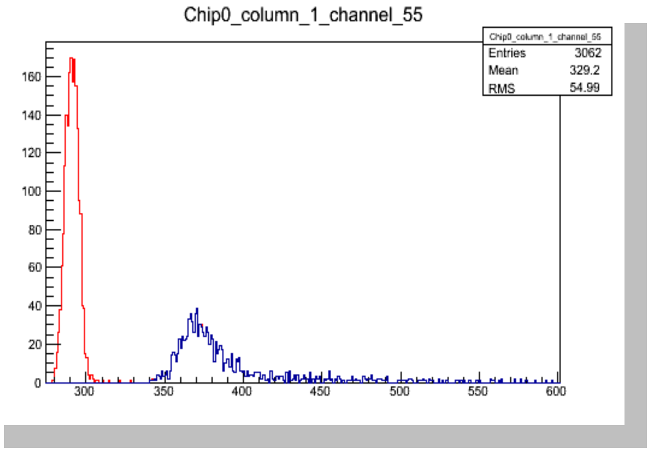
Data Quality Existing

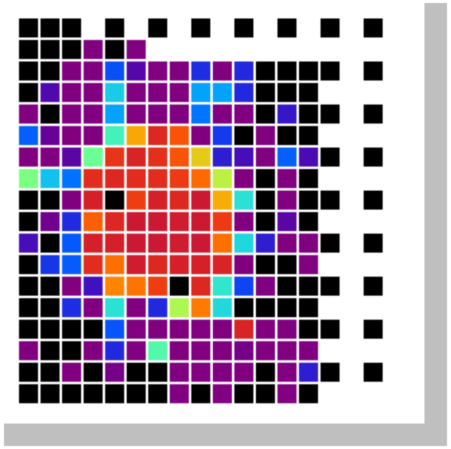
Existing Online (RT)

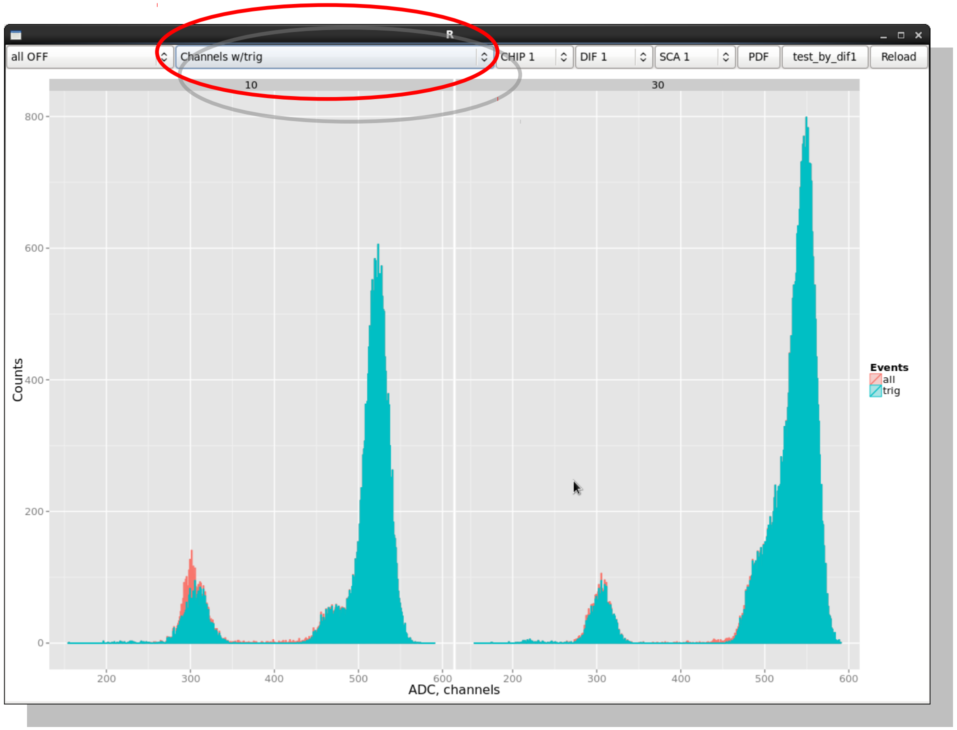
- Fast monitoring by Frédéric (beam profile) in Python
  - Circular buffer (concept proof, in Root)
  - HitCam

Existing Offline (after run)

- Skiroc_stat (Python)
- Raw2Root (by Thibault)
  - last TB : many histograms
    - **Needs better “hierarchical” presentation.**
      - » Example exists in SDHCAL code (Qt) to be adapted.
- R analysis (by Vladik)
  - Complete analysis framework





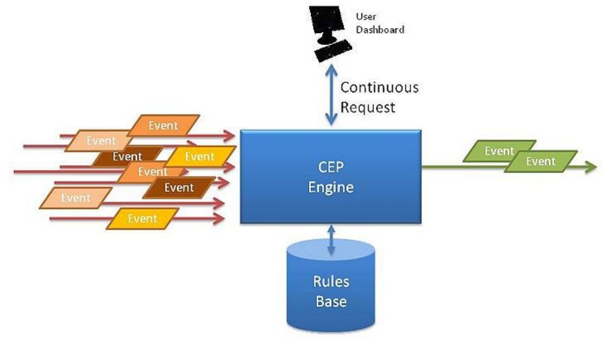


for online needs decoupling of decoding & histos

- data from Calicoes evt builder ⇒ subsampling socket
  - on request (random, on evt#)

Vincent.Boudry@in2p3.fr

Debug & monitoring



Complex Event Processing - Auteur : Patrick Gantet

Identification of errors, signalisation & actions:

- Existing: Failsafe version mode of configs
  - Timing in XML
- Review of error messages → log parser ? all to be evaluated
  - Check for message counts/ missing critical things...
    - Some examples:
      - » pylogsparser https://pypi.python.org/pypi/pylogsparser/0.2 to be evaluated]
      - » Analog [http://www.analog.cx/]
      - » DejaVu (from memory) [https://www.systems.ethz.ch/node/198]
      - » or Complex Event Processing tool
- High Level Error msg:
  - in cmd_det: Transitions errors
    - In PHYGUI

Configuration

Needs:

- GUI for individual chips
- Hierarchical setup of parameters
- Scripting
  - PyCalDaq interface
    - Above CmdDet
  - Bindings Pyrame in C / C++ / Python / R / LabView
  - Reconfigure(param) in CmdDet
    - per chip, with regexp...
      - » on bitstream in memory

Config Gui Existing tools

Vladik:

- Ruby & GtK GUI
  ⇒ Python + Gnome GTK / Qt [SystemAgnostic ?]
- edition of outputs
- extended to 4 chips

Muriel:

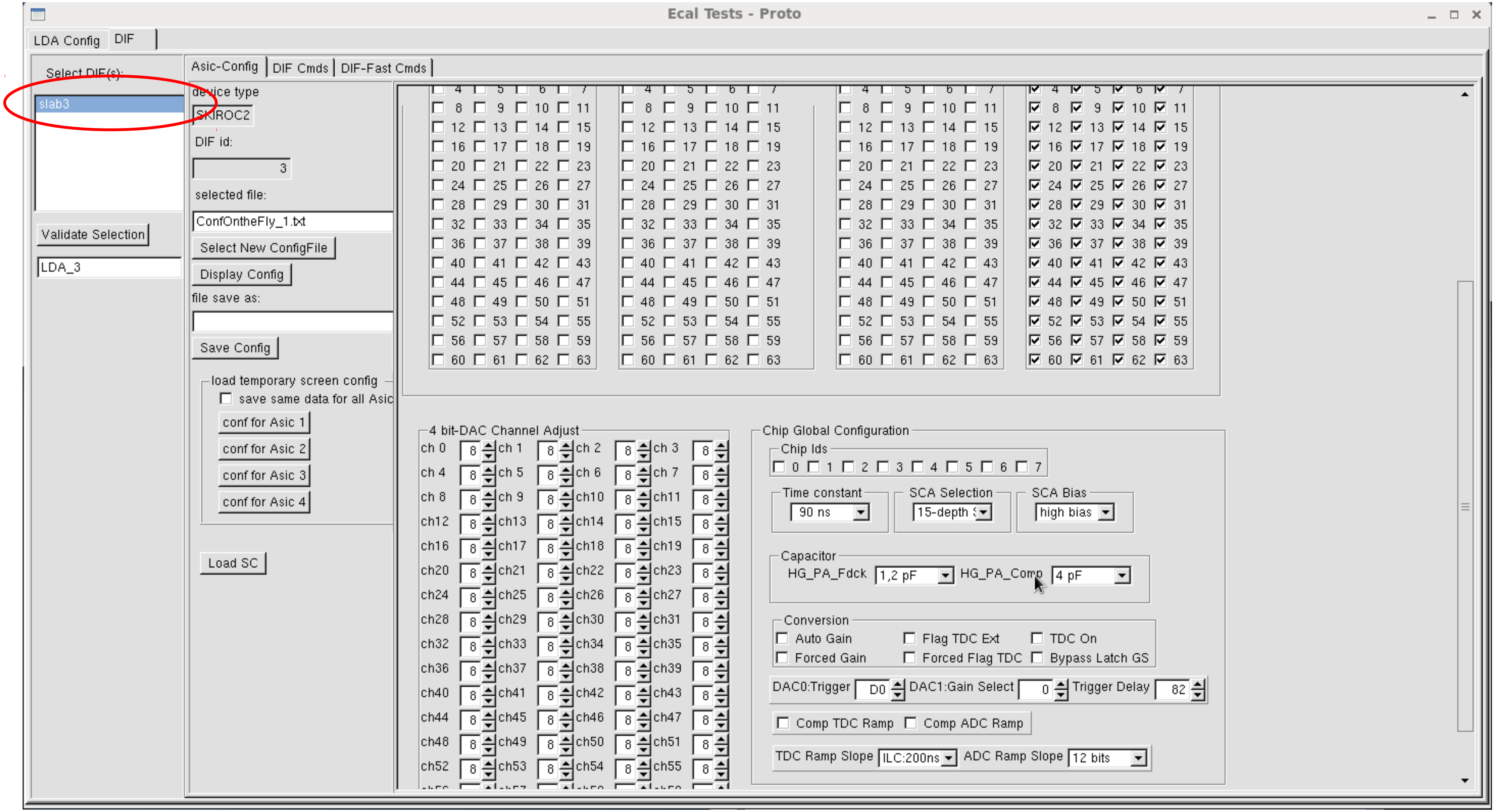
- C++ & Root
  - Hierarchical structure foreseen

IPNL tools (SDHCAL):

- ORACLE DB ↔ XDAQ; + xml hierarchy
- Scripts for modification (scans) of xml files
  -

Muriel's GUI

C++ + Root



Ecal Tests - Proto

Select DIF(s):

slab3

Validate Selection

LDA_3

device type

SKIROC2

DIF id:

3

selected file:

ConfOntheFly_1.txt

file save as:

load temporary screen config

□ save same data for all Asic

□ 8 □ 9 □ 10 □ 11 □ 8 □ 9 □ 10 □ 11 □ 8 □ 9 □ 10 □ 11 ☑ 8 ☑ 9 ☑ 10 ☑ 11
□ 12 □ 13 □ 14 □ 15 □ 12 □ 13 □ 14 □ 15 □ 12 □ 13 □ 14 □ 15 ☑ 12 ☑ 13 ☑ 14 ☑ 15
□ 16 □ 17 □ 18 □ 19 □ 16 □ 17 □ 18 □ 19 □ 16 □ 17 □ 18 □ 19 ☑ 16 ☑ 17 ☑ 18 ☑ 19
□ 20 □ 21 □ 22 □ 23 □ 20 □ 21 □ 22 □ 23 □ 20 □ 21 □ 22 □ 23 ☑ 20 ☑ 21 ☑ 22 ☑ 23
□ 24 □ 25 □ 26 □ 27 □ 24 □ 25 □ 26 □ 27 □ 24 □ 25 □ 26 □ 27 ☑ 24 ☑ 25 ☑ 26 ☑ 27
□ 28 □ 29 □ 30 □ 31 □ 28 □ 29 □ 30 □ 31 □ 28 □ 29 □ 30 □ 31 ☑ 28 ☑ 29 ☑ 30 ☑ 31
□ 32 □ 33 □ 34 □ 35 □ 32 □ 33 □ 34 □ 35 □ 32 □ 33 □ 34 □ 35 ☑ 32 ☑ 33 ☑ 34 ☑ 35
□ 36 □ 37 □ 38 □ 39 □ 36 □ 37 □ 38 □ 39 □ 36 □ 37 □ 38 □ 39 ☑ 36 ☑ 37 ☑ 38 ☑ 39
□ 40 □ 41 □ 42 □ 43 □ 40 □ 41 □ 42 □ 43 □ 40 □ 41 □ 42 □ 43 ☑ 40 ☑ 41 ☑ 42 ☑ 43
□ 44 □ 45 □ 46 □ 47 □ 44 □ 45 □ 46 □ 47 □ 44 □ 45 □ 46 □ 47 ☑ 44 ☑ 45 ☑ 46 ☑ 47
□ 48 □ 49 □ 50 □ 51 □ 48 □ 49 □ 50 □ 51 □ 48 □ 49 □ 50 □ 51 ☑ 48 ☑ 49 ☑ 50 ☑ 51
□ 52 □ 53 □ 54 □ 55 □ 52 □ 53 □ 54 □ 55 □ 52 □ 53 □ 54 □ 55 ☑ 52 ☑ 53 ☑ 54 ☑ 55
□ 56 □ 57 □ 58 □ 59 □ 56 □ 57 □ 58 □ 59 □ 56 □ 57 □ 58 □ 59 ☑ 56 ☑ 57 ☑ 58 ☑ 59
□ 60 □ 61 □ 62 □ 63 □ 60 □ 61 □ 62 □ 63 □ 60 □ 61 □ 62 □ 63 ☑ 60 ☑ 61 ☑ 62 ☑ 63

4 bit-DAC Channel Adjust

ch 0 8 ch 1 8 ch 2 8 ch 3 8
ch 4 8 ch 5 8 ch 6 8 ch 7 8
ch 8 8 ch 9 8 ch10 8 ch11 8
ch12 8 ch13 8 ch14 8 ch15 8
ch16 8 ch17 8 ch18 8 ch19 8
ch20 8 ch21 8 ch22 8 ch23 8
ch24 8 ch25 8 ch26 8 ch27 8
ch28 8 ch29 8 ch30 8 ch31 8
ch32 8 ch33 8 ch34 8 ch35 8
ch36 8 ch37 8 ch38 8 ch39 8
ch40 8 ch41 8 ch42 8 ch43 8
ch44 8 ch45 8 ch46 8 ch47 8
ch48 8 ch49 8 ch50 8 ch51 8
ch52 8 ch53 8 ch54 8 ch55 8

Chip Global Configuration

Chip Ids: □ 0 □ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7

Time constant: 90 ns
SCA Selection: 15-depth
SCA Bias: high bias

Capacitor: HG_PA_Fdck 1,2 pF HG_PA_Comp 4 pF

Conversion
□ Auto Gain □ Flag TDC Ext □ TDC On
□ Forced Gain □ Forced Flag TDC □ Bypass Latch GS

DAC0:Trigger D0 DAC1:Gain Select 0 Trigger Delay 82

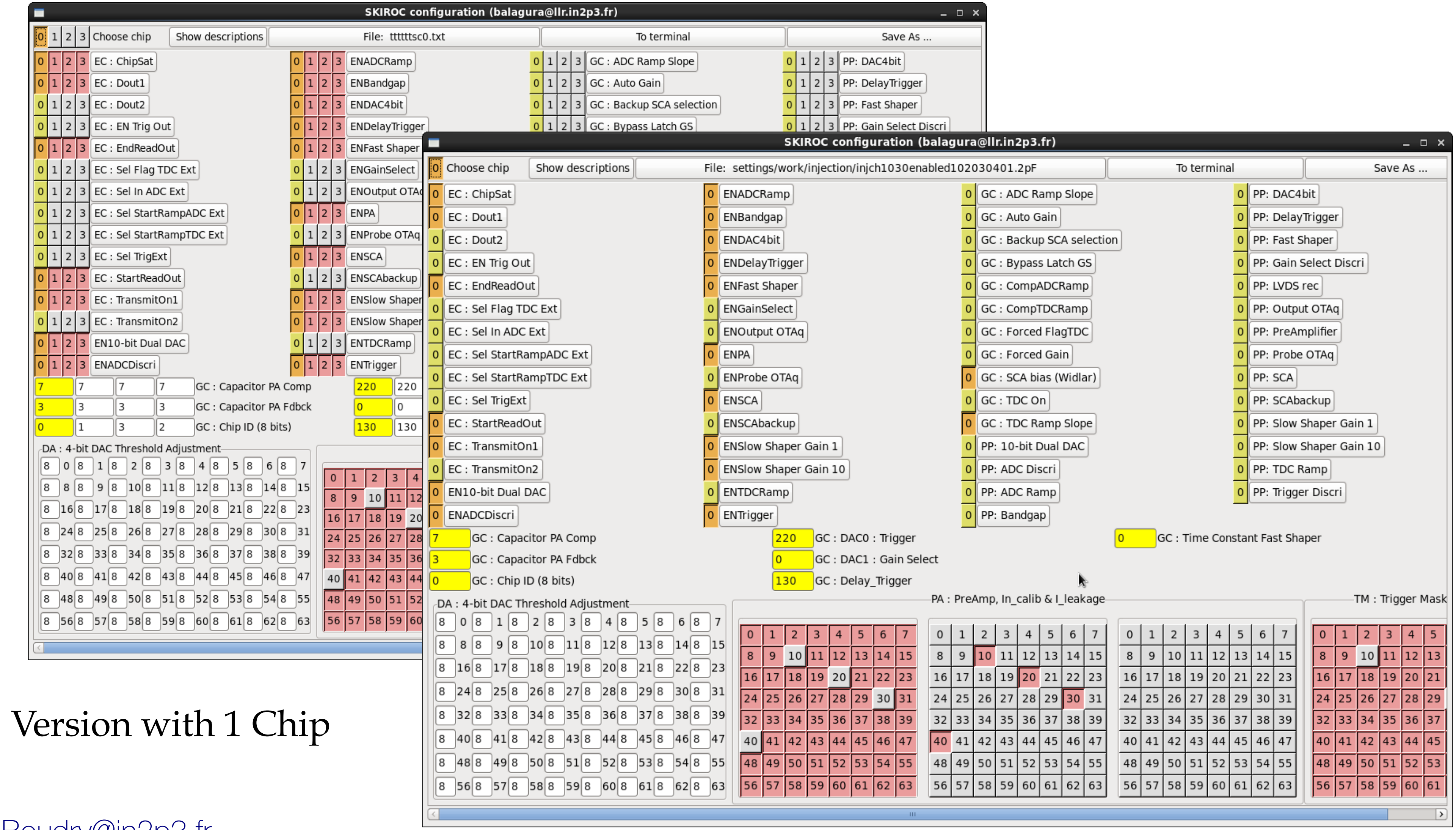
□ Comp TDC Ramp □ Comp ADC Ramp

TDC Ramp Slope ILC:200ns ADC Ramp Slope 12 bits

Ruby + GnomeGTK (Vladik)

Edition of chip configuration files

Version with 4 Chips



SKIROC configuration (balagura@llr.in2p3.fr)

File: ttttttsc0.txt

7	7	7	7	GC : Capacitor PA Comp
3	3	3	3	GC : Capacitor PA Fdbck
0	1	3	2	GC : Chip ID (8 bits)

220	220
0	0
130	130

SKIROC configuration (balagura@llr.in2p3.fr)

File: settings/work/injection/injch1030enabled102030401.2pF

7	GC : Capacitor PA Comp
3	GC : Capacitor PA Fdbck
0	GC : Chip ID (8 bits)
220	GC : DAC0 : Trigger
0	GC : DAC1 : Gain Select
130	GC : Delay_Trigger
0	GC : Time Constant Fast Shaper

DA : 4-bit DAC Threshold Adjustment

8	0	8	1	8	2	8	3	8	4	8	5	8	6	8	7
8	8	8	9	8	10	8	11	8	12	8	13	8	14	8	15
8	16	8	17	8	18	8	19	8	20	8	21	8	22	8	23
8	24	8	25	8	26	8	27	8	28	8	29	8	30	8	31
8	32	8	33	8	34	8	35	8	36	8	37	8	38	8	39
8	40	8	41	8	42	8	43	8	44	8	45	8	46	8	47
8	48	8	49	8	50	8	51	8	52	8	53	8	54	8	55
8	56	8	57	8	58	8	59	8	60	8	61	8	62	8	63

Version with 1 Chip

Vincent.Boudry@in2p3.fr

Interface

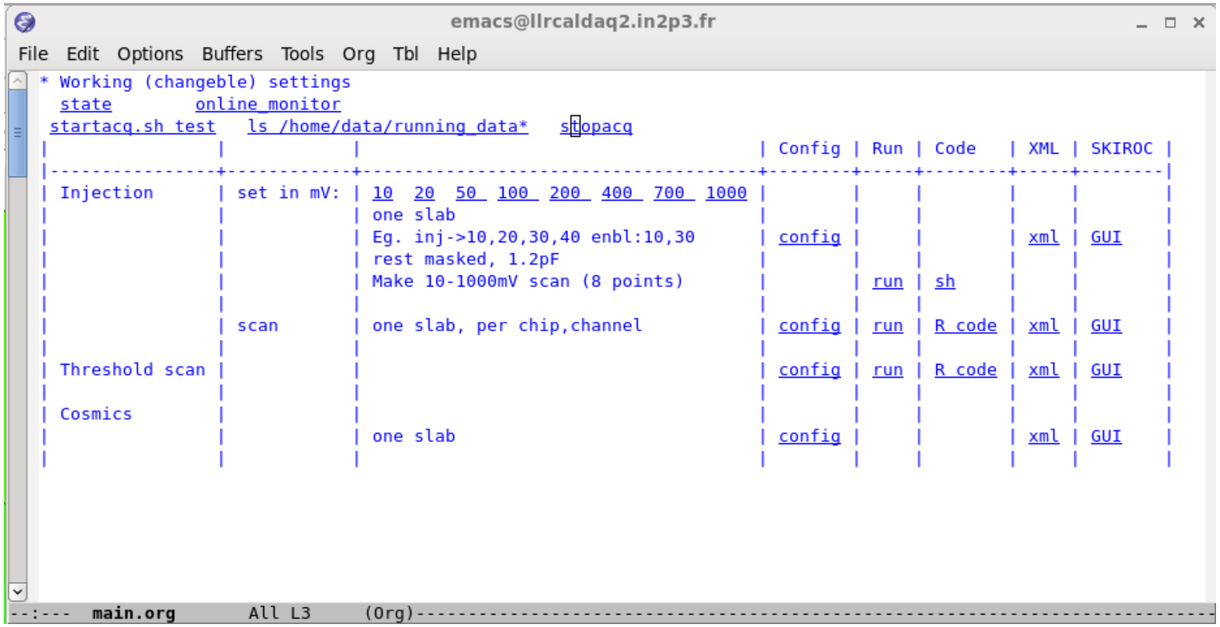
Scripts

(not so) poor's man:

- Emacs & Org
  - Clickable interface

EUDAQ:

-



emacs@llrcaldaq2.in2p3.fr
File Edit Options Buffers Tools Org Tbl Help

```
* Working (changeble) settings
  state        online_monitor
 startacq.sh test   ls /home/data/running_data*   stopacq
|                |            |                                      | Config | Run | Code   | XML | SKIROC |
|----------------+------------+--------------------------------------+--------+-----+--------+-----+--------|
| Injection      | set in mV: | 10  20  50  100  200  400  700  1000 |        |     |        |     |        |
|                |            | one slab                             |        |     |        |     |        |
|                |            | Eg. inj->10,20,30,40 enbl:10,30      | config |     |        | xml | GUI    |
|                |            | rest masked, 1.2pF                   |        |     |        |     |        |
|                |            | Make 10-1000mV scan (8 points)       |        | run | sh     |     |        |
|                |            |                                      |        |     |        |     |        |
|                | scan       | one slab, per chip,channel           | config | run | R code | xml | GUI    |
|                |            |                                      |        |     |        |     |        |
| Threshold scan |            |                                      | config | run | R code | xml | GUI    |
|                |            |                                      |        |     |        |     |        |
| Cosmics        |            |                                      |        |     |        |     |        |
|                |            | one slab                             | config |     |        | xml | GUI    |
|                |            |                                      |        |     |        |     |        |
--:---  main.org       All L3     (Org)------------------------------------------------------------------------
```

Integration → AIDA

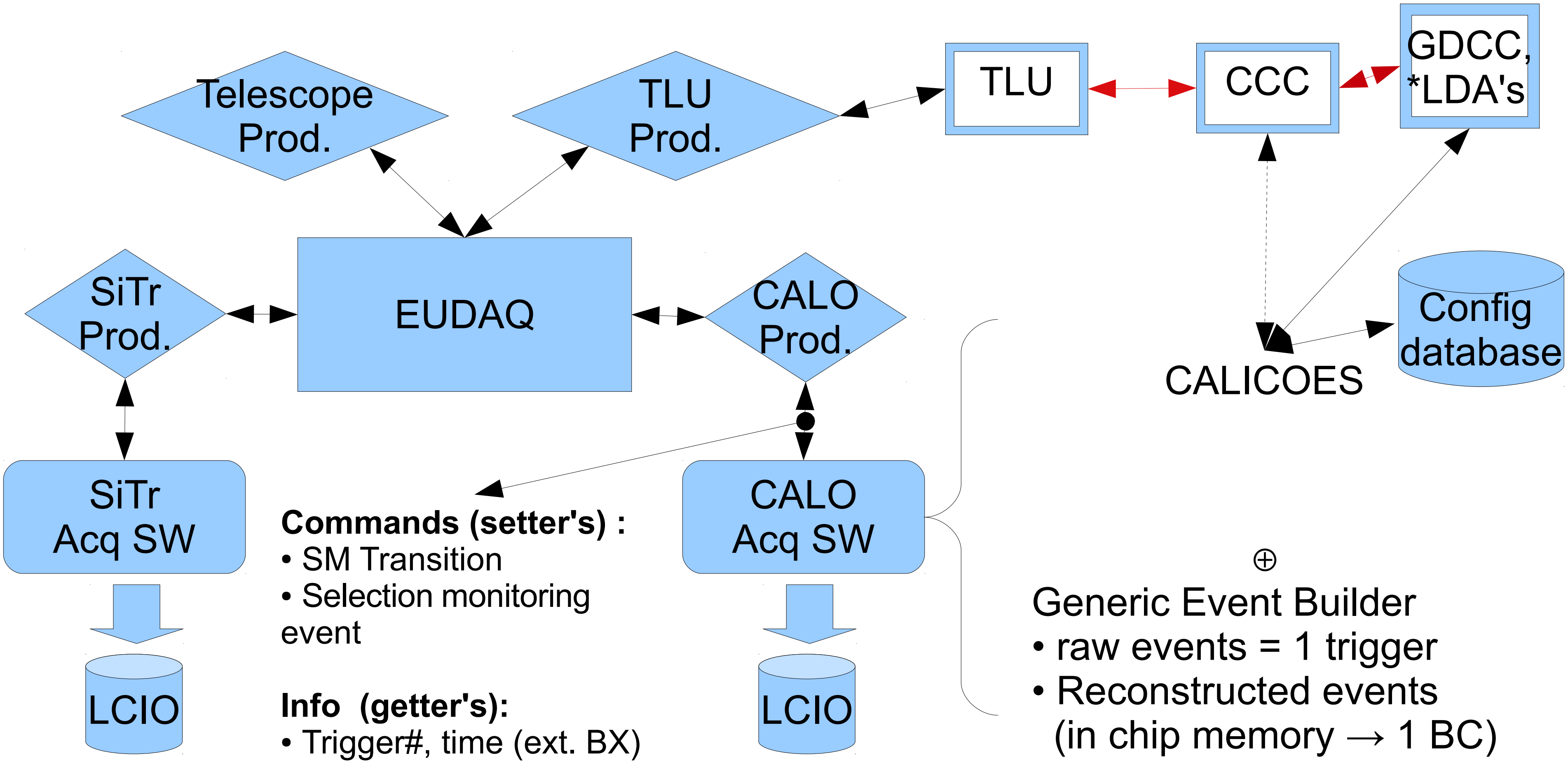
AIDA « specification » document delivered

- Crude specifications : ~rationale ⇒ real implementation to be precised
  - http://cds.cern.ch/record/1666866?ln=en
- Basis :
  - HW Sync by TLU ↔ CCC (2 versions)
  - SW Sync : High level by EUDAQ (control)
    - interface ↔ CALICOES
      - » DIF FW
    - Data written in // in LCIO files
      - » 1 event = 1 readout (spill, trigger, …)

Technical discussion planned at AIDA 3rd annual meeting (next week in Vienna)

- practical implementation, test bench… ECAL+SiTra (EUDAQ)

Integration of CALICE ECAL & EUDAQ (Small system, Software side)



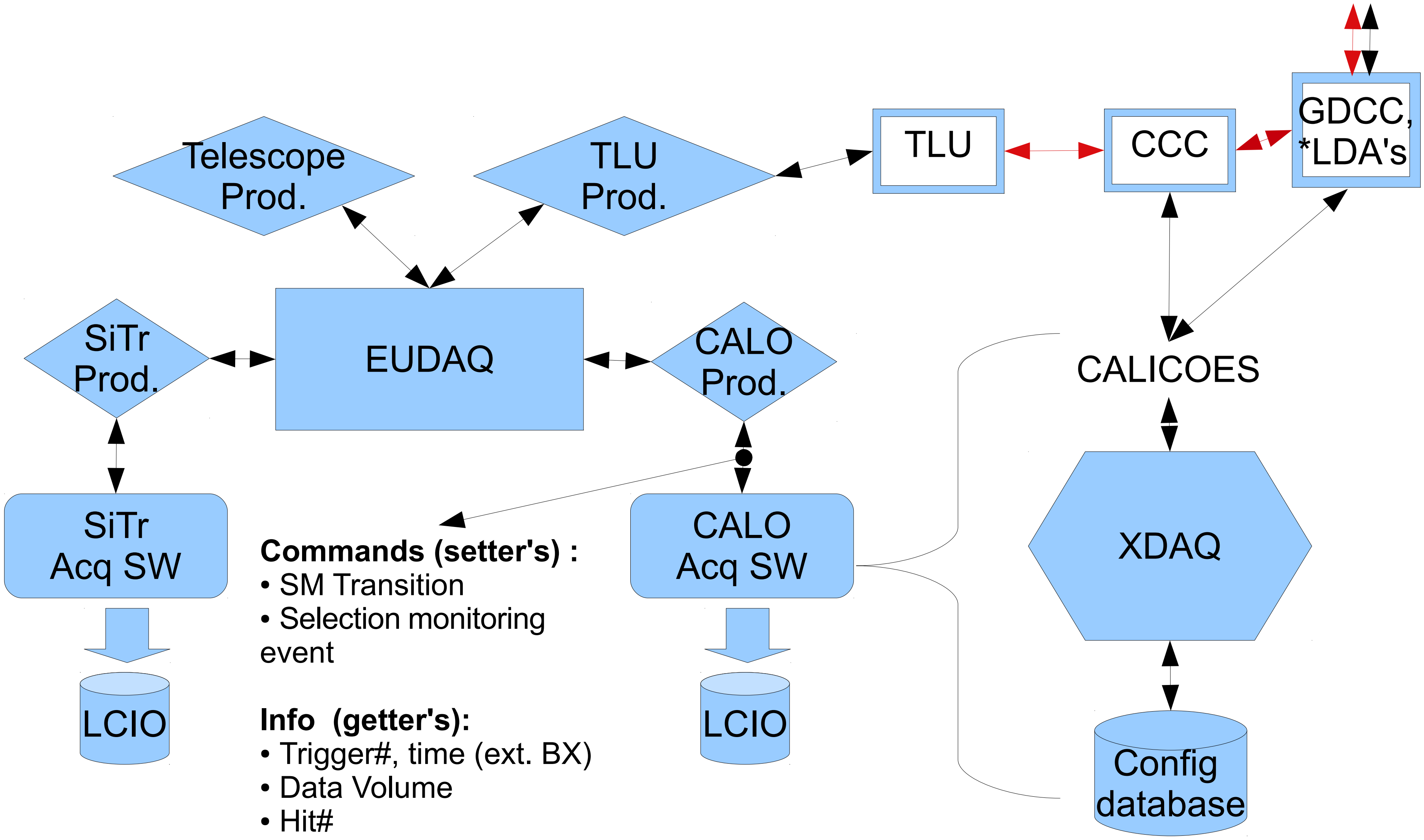
Commands (setter's) :
- SM Transition
- Selection monitoring event

Info (getter's):
- Trigger#, time (ext. BX)
- Data Volume
- Hit#
- …

⊕

Generic Event Builder
- raw events = 1 trigger
- Reconstructed events (in chip memory → 1 BC)

CALICE HCAL & EUDAQ (ex. large system, SW side)



Commands (setter's) :
- SM Transition
- Selection monitoring event

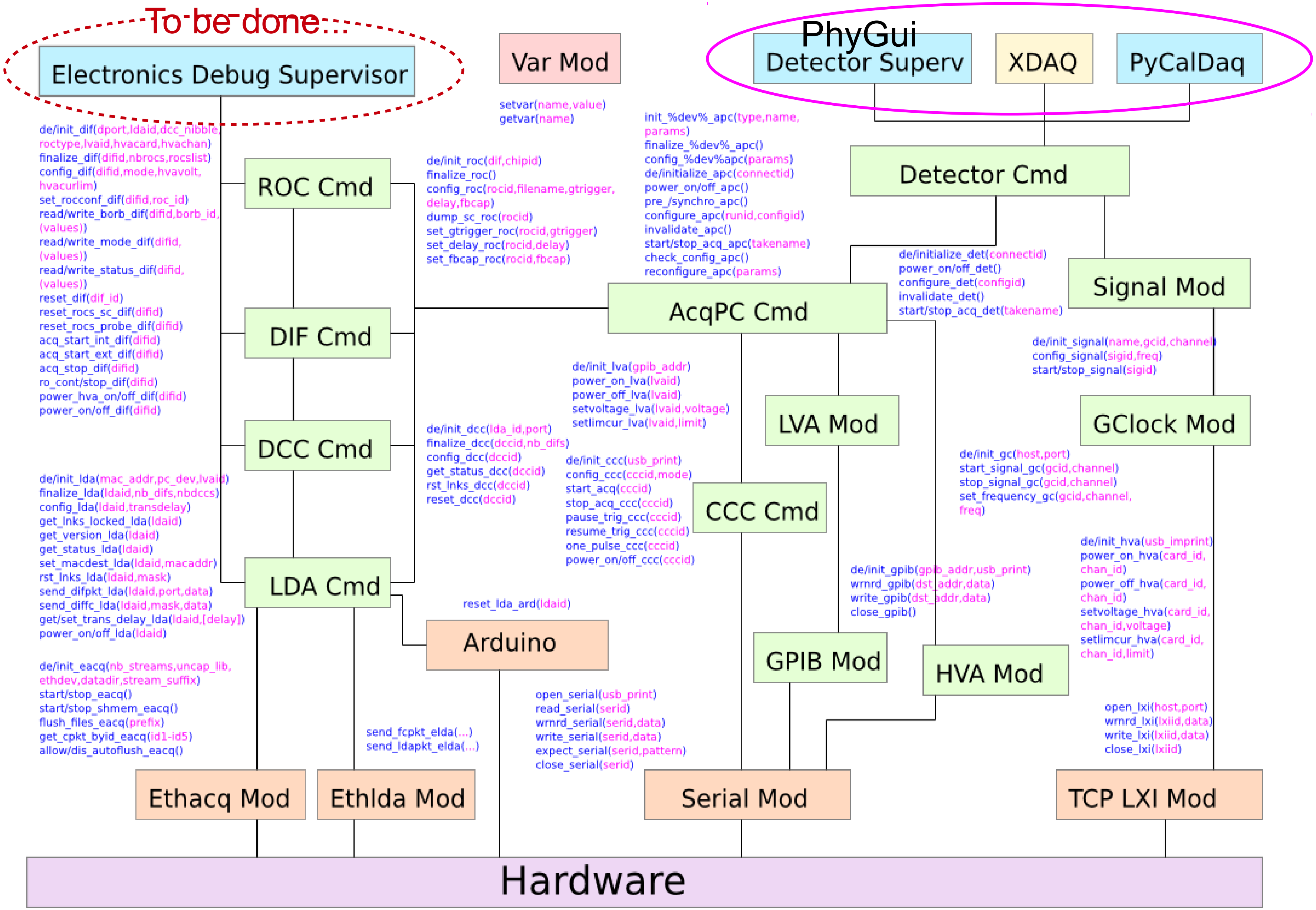
Info (getter's):
- Trigger#, time (ext. BX)
- Data Volume
- Hit#
- …

Integration in « AIDA-2 »

Contact persons: D. Cussans, M. Wing

- preparatory meeting in Vienna / Vidyo
- Toward ?? : To be discussed:
  - 1) A central DAQ **system** which would provide the interfaces (hardware, software) needed to detectors to work together in a **combined beam-test** during AIDA2. There would also need to be corresponding effort by the detector work-packages to interface their DAQ to a central DAQ.
  - 2) Common DAQ **components** - it would be valuable for some groups to have a "label", even if there is very little funding, for development of ge DAQ components and adapting them for Linear Collider purposes.

CALICOES big picture



```
de/init_dif(dport,ldaid,dcc_nibble,roctype,lvaid,hvacard,hvachan)
finalize_dif(difid,nbrocs,rocslist)
config_dif(difid,mode,hvavolt,hvacurlim)
set_rocconf_dif(difid,roc_id)
read/write_borb_dif(difid,borb_id,(values))
read/write_mode_dif(difid,(values))
read/write_status_dif(difid,(values))
reset_dif(dif_id)
reset_rocs_sc_dif(difid)
reset_rocs_probe_dif(difid)
acq_start_int_dif(difid)
acq_start_ext_dif(difid)
acq_stop_dif(difid)
ro_cont/stop_dif(difid)
power_hva_on/off_dif(difid)
power_on/off_dif(difid)
```

```
de/init_lda(mac_addr,pc_dev,lvaid)
finalize_lda(ldaid,nb_difs,nbdccs)
config_lda(ldaid,transdelay)
get_lnks_locked_lda(ldaid)
get_version_lda(ldaid)
get_status_lda(ldaid)
set_macdest_lda(ldaid,macaddr)
rst_lnks_lda(ldaid,mask)
send_difpkt_lda(ldaid,port,data)
send_diffc_lda(ldaid,mask,data)
get/set_trans_delay_lda(ldaid,[delay])
power_on/off_lda(ldaid)
```

```
de/init_eacq(nb_streams,uncap_lib,ethdev,datadir,stream_suffix)
start/stop_eacq()
start/stop_shmem_eacq()
flush_files_eacq(prefix)
get_cpkt_byid_eacq(id1-id5)
allow/dis_autoflush_eacq()
```

```
setvar(name,value)
getvar(name)
```

```
de/init_roc(dif,chipid)
finalize_roc()
config_roc(rocid,filename,gtrigger,delay,fbcap)
dump_sc_roc(rocid)
set_gtrigger_roc(rocid,gtrigger)
set_delay_roc(rocid,delay)
set_fbcap_roc(rocid,fbcap)
```

```
init_%dev%_apc(type,name,params)
finalize_%dev%_apc()
config_%dev%apc(params)
de/initialize_apc(connectid)
power_on/off_apc()
pre_/synchro_apc()
configure_apc(runid,configid)
invalidate_apc()
start/stop_acq_apc(takename)
check_config_apc()
reconfigure_apc(params)
```

```
de/initialize_det(connectid)
power_on/off_det()
configure_det(configid)
invalidate_det()
start/stop_acq_det(takename)
```

```
de/init_signal(name,gcid,channel)
config_signal(sigid,freq)
start/stop_signal(sigid)
```

```
de/init_lva(gpib_addr)
power_on_lva(lvaid)
power_off_lva(lvaid)
setvoltage_lva(lvaid,voltage)
setlimcur_lva(lvaid,limit)
```

```
de/init_dcc(lda_id,port)
finalize_dcc(dccid,nb_difs)
config_dcc(dccid)
get_status_dcc(dccid)
rst_lnks_dcc(dccid)
reset_dcc(dccid)
```

```
de/init_ccc(usb_print)
config_ccc(cccid,mode)
start_acq(cccid)
stop_acq_ccc(cccid)
pause_trig_ccc(cccid)
resume_trig_ccc(cccid)
one_pulse_ccc(cccid)
power_on/off_ccc(cccid)
```

```
de/init_gc(host,port)
start_signal_gc(gcid,channel)
stop_signal_gc(gcid,channel)
set_frequency_gc(gcid,channel,freq)
```

```
de/init_hva(usb_imprint)
power_on_hva(card_id,chan_id)
power_off_hva(card_id,chan_id)
setvoltage_hva(card_id,chan_id,voltage)
setlimcur_hva(card_id,chan_id,limit)
```

```
de/init_gpib(gpib_addr,usb_print)
wrnrd_gpib(dst_addr,data)
write_gpib(dst_addr,data)
close_gpib()
```

```
reset_lda_ard(ldaid)
```

```
send_fcpkt_elda(...)
send_ldapkt_elda(...)
```

```
open_serial(usb_print)
read_serial(serid)
wrnrd_serial(serid,data)
write_serial(serid,data)
expect_serial(serid,pattern)
close_serial(serid)
```

```
open_lxi(host,port)
wrnrd_lxi(lxiid,data)
write_lxi(lxiid,data)
close_lxi(lxiid)
```