BI LHC Sequencer tasks - Past, Present and Future

6th June 2024

Stephen Jackson
Athanasios Topaloudis

(Thanks for input from Christos Zamantzas)

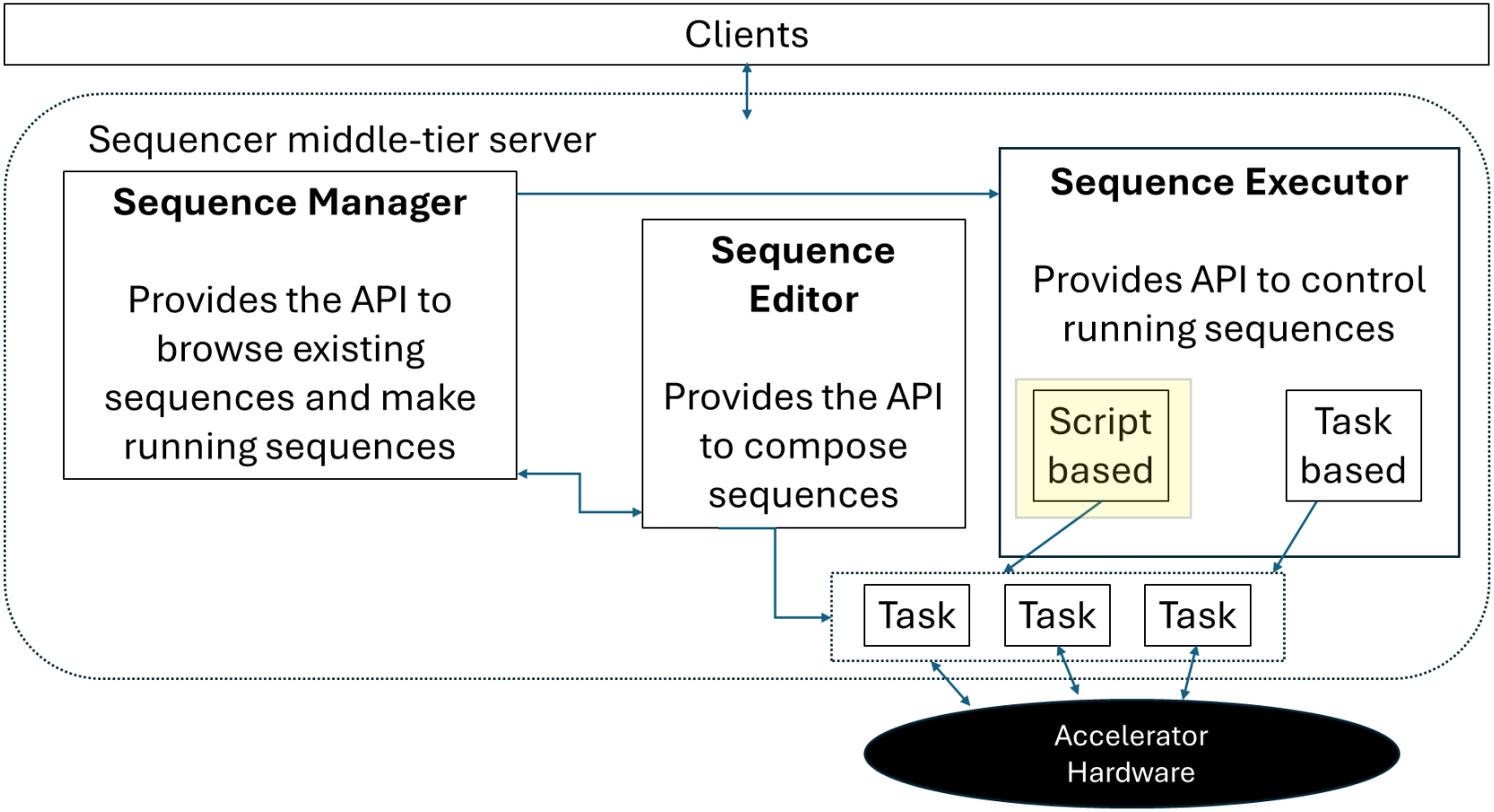
Disclaimer

For the *future* part, really at the start of discussions

… plans might change

… propose to be invited back @TB in YETS for an update

What are the BI LHC sequencer tasks?

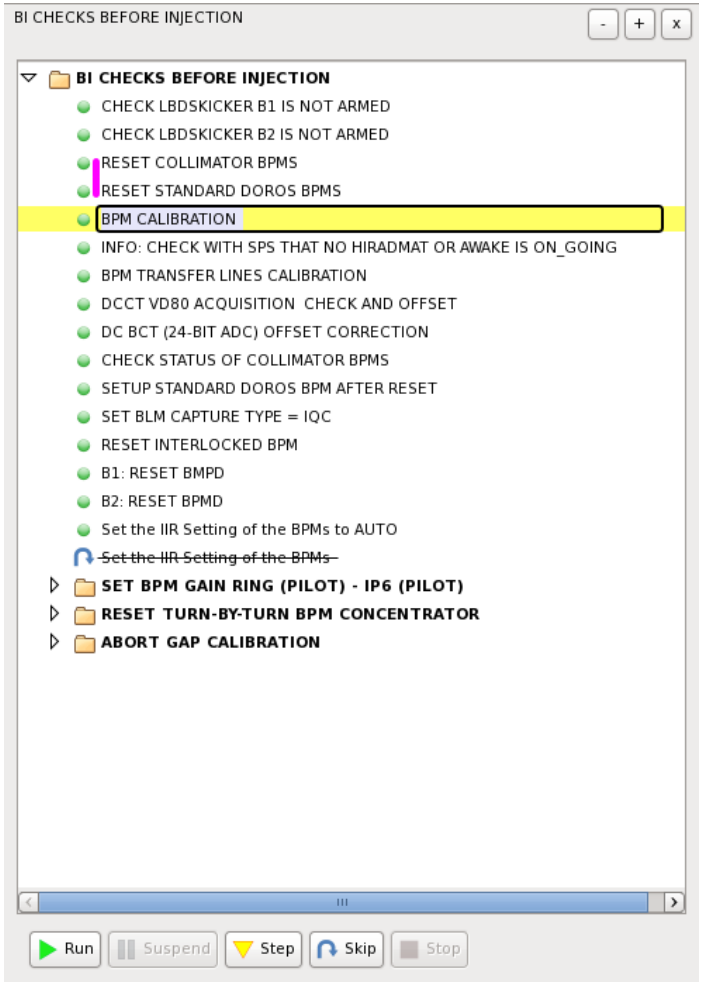


What are the BI LHC sequencer tasks?

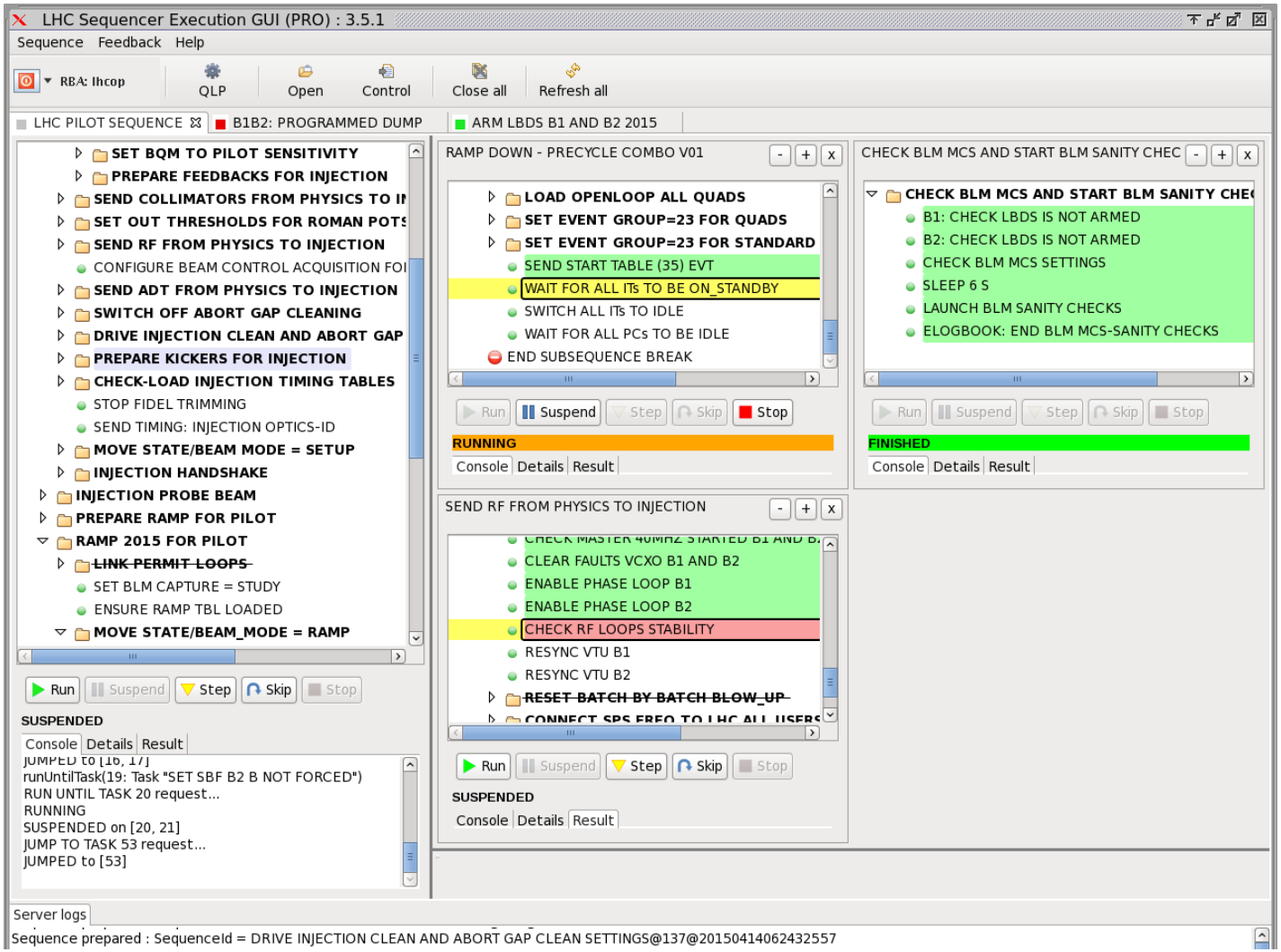
Task based - Simple	Script based - Complex
Assign on-error behavior to each task	Full power of Java :
Easily run several tasks in parallel	variables, constructs (if-else, loops, try-catch-finally blocks), type checking
Easily jump from task to task	Impossible to assign special on-error behavior to a "task"
Execute arbitrary task out of the sequence order	Impossible to jump from "task" to "task"
Impossible to use common programmatic structures like if-else, loops, try-catch-finally blocks	Not easy (but possible) to run several activities in parallel
Impossible to use variables (can use parameters defined before sequence started)	

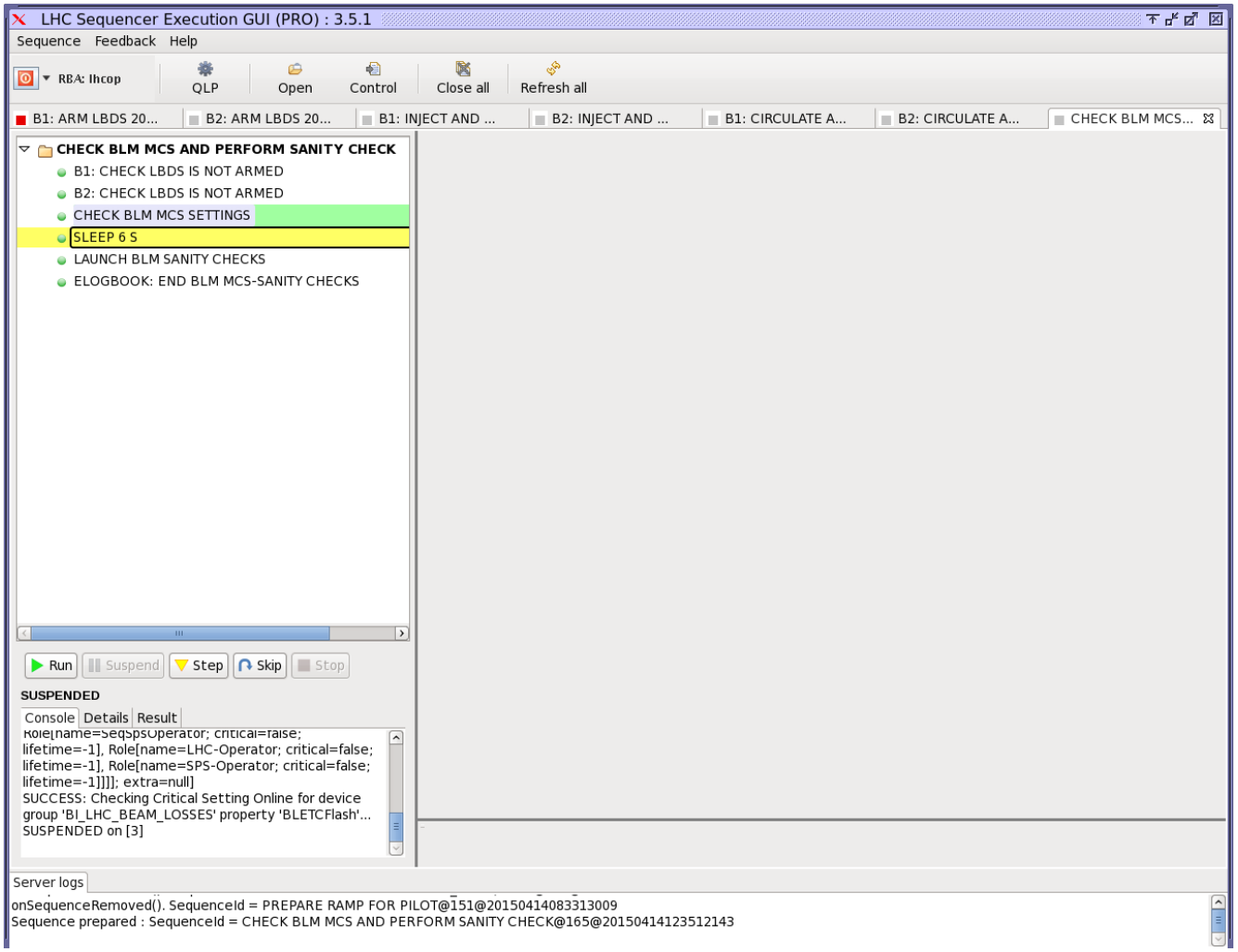
What are the BI LHC sequencer tasks?

- Most BI tasks are executed in preparation for the next fill
  - But can also be executed at any point
- Sequencer GUI -->



- Code locations in git
  - Different locations for OP written tasks and BI written tasks
    - https://gitlab.cern.ch/acc-co/seq/task/seq-task-bi/
    - https://gitlab.cern.ch/acc-co/seq/task/seq-task-blm
    - https://gitlab.cern.ch/acc-co/seq/task/seq-task-bpm
    - https://gitlab.cern.ch/acc-co/seq/task/seq-task-abortgap


LHC Sequencer Execution GUI (PRO) : 3.5.1
Sequence Feedback Help
RBA: lhcop
QLP
Open
Control
Close all
Refresh all
LHC PILOT SEQUENCE
B1B2: PROGRAMMED DUMP
ARM LBDS B1 AND B2 2015
SET BQM TO PILOT SENSITIVITY
PREPARE FEEDBACKS FOR INJECTION
SEND COLLIMATORS FROM PHYSICS TO I
SET OUT THRESHOLDS FOR ROMAN POTS
SEND RF FROM PHYSICS TO INJECTION
CONFIGURE BEAM CONTROL ACQUISITION FOI
SEND ADT FROM PHYSICS TO INJECTION
SWITCH OFF ABORT GAP CLEANING
DRIVE INJECTION CLEAN AND ABORT GAP
PREPARE KICKERS FOR INJECTION
CHECK-LOAD INJECTION TIMING TABLES
STOP FIDEL TRIMMING
SEND TIMING: INJECTION OPTICS-ID
MOVE STATE/BEAM MODE = SETUP
INJECTION HANDSHAKE
INJECTION PROBE BEAM
PREPARE RAMP FOR PILOT
RAMP 2015 FOR PILOT
LINK PERMIT LOOPS
SET BLM CAPTURE = STUDY
ENSURE RAMP TBL LOADED
MOVE STATE/BEAM_MODE = RAMP
Run
Suspend
Step
Skip
Stop
SUSPENDED
Console Details Result
JUMPED to [16, 17]
runUntilTask(19: Task "SET SBF B2 B NOT FORCED")
RUN UNTIL TASK 20 request...
RUNNING
SUSPENDED on [20, 21]
JUMP TO TASK 53 request...
JUMPED to [53]
RAMP DOWN - PRECYCLE COMBO V01
LOAD OPENLOOP ALL QUADS
SET EVENT GROUP=23 FOR QUADS
SET EVENT GROUP=23 FOR STANDARD
SEND START TABLE (35) EVT
WAIT FOR ALL ITs TO BE ON_STANDBY
SWITCH ALL ITs TO IDLE
WAIT FOR ALL PCs TO BE IDLE
END SUBSEQUENCE BREAK
Run
Suspend
Step
Skip
Stop
RUNNING
Console Details Result
CHECK BLM MCS AND START BLM SANITY CHEC
CHECK BLM MCS AND START BLM SANITY CHE
B1: CHECK LBDS IS NOT ARMED
B2: CHECK LBDS IS NOT ARMED
CHECK BLM MCS SETTINGS
SLEEP 6 S
LAUNCH BLM SANITY CHECKS
ELOGBOOK: END BLM MCS-SANITY CHECKS
Run
Suspend
Step
Skip
Stop
FINISHED
Console Details Result
SEND RF FROM PHYSICS TO INJECTION
CLEAR FAULTS VCXO B1 AND B2
ENABLE PHASE LOOP B1
ENABLE PHASE LOOP B2
CHECK RF LOOPS STABILITY
RESYNC VTU B1
RESYNC VTU B2
RESET BATCH BY BATCH BLOW_UP
Run
Suspend
Step
Skip
Stop
SUSPENDED
Console Details Result
Server logs
Sequence prepared : SequenceId = DRIVE INJECTION CLEAN AND ABORT GAP CLEAN SETTINGS@137@20150414062432557


LHC Sequencer Execution GUI (PRO) : 3.5.1
Sequence
Feedback
Help
RBA: lhcop
QLP
Open
Control
Close all
Refresh all
B1: ARM LBDS 20...
B2: ARM LBDS 20...
B1: INJECT AND ...
B2: INJECT AND ...
B1: CIRCULATE A...
B2: CIRCULATE A...
CHECK BLM MCS...
CHECK BLM MCS AND PERFORM SANITY CHECK
B1: CHECK LBDS IS NOT ARMED
B2: CHECK LBDS IS NOT ARMED
CHECK BLM MCS SETTINGS
SLEEP 6 S
LAUNCH BLM SANITY CHECKS
ELOGBOOK: END BLM MCS-SANITY CHECKS
Run
Suspend
Step
Skip
Stop
SUSPENDED
Console
Details
Result
lifetime=-1], Role[name=LHC-Operator; critical=false;
lifetime=-1], Role[name=SPS-Operator; critical=false;
lifetime=-1]]]]; extra=null]
SUCCESS: Checking Critical Setting Online for device
group 'BI_LHC_BEAM_LOSSES' property 'BLETCFlash'...
SUSPENDED on [3]
Server logs
onSequenceRemoved(). SequenceId = PREPARE RAMP FOR PILOT@151@20150414083313009
Sequence prepared : SequenceId = CHECK BLM MCS AND PERFORM SANITY CHECK@165@20150414123512143

History of tasks (scripts)

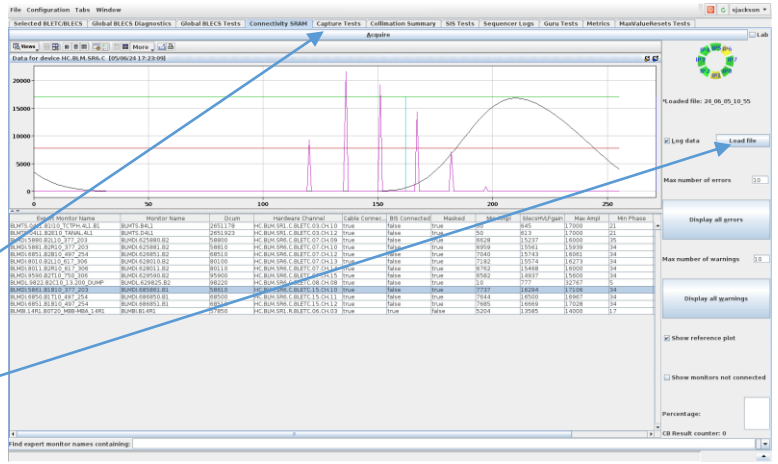
- Who made the first incarnation of scripts?
  - For BLMs
    - **MCS** Greg Kruk
    - **Connectivity and BP internal** was Laurette Ponce (with LSA & OP support(Fabio, Delphine,…))
    - **Beam Permit External** was supposed to use AccTesting (the tool TE uses for the hardware commissioning)
      - But they never found time
      - Hopefully we add this
  - Others Serkan Bozyigit (FELL ~13 years ago)
    - Subsequetly taken over by Athanasios

- In recent years, also been worked on by
  - Georges-Henry
  - Stephen
  - David Medina (PJAS + FELL)
  - Manuel + Magdalena Stachon (TECH - 2018)
    - Some changes made it into Operation (improved parallelization of devices)
    - Some changes are maybe still not Operational!
      - To clarify if improvements on hardware consistency checks (DB vs HW)

- Most recent new tasks BOMEMCHK
  - For BLM and BPM
  - 36th SY-BI Students and R&D Meeting

Relationship with Expert GUIs

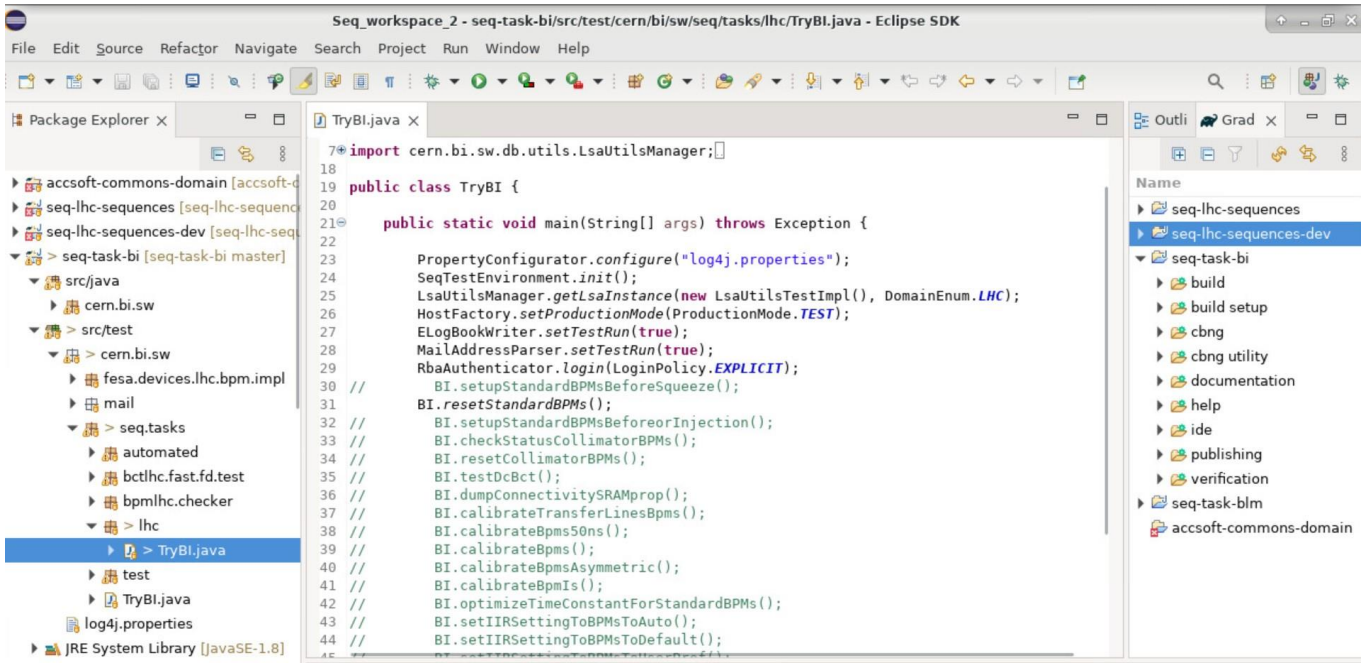
- Some sequencer tasks trigger existing code in Expert GUIs
  - BLM Connectivity SRAM tests
    - Callback behind **Acquire** button also triggered from Sequencer
    - **Load File** button loads files generated from GUI and Sequencer



  - DOROS BPMs
    - Code to read and execute the commands shared between sequencer tasks and GUI
      Finally moved into FESA
  - BPM Calibration
    - Similar to DOROS BPMs but library was never adopted by the sequencer tasks --> code duplication!

Execution of tasks outside the CCC

- Testing in the CCC is not easy, so scripts can be executed from an Eclipse workspace
  - Checked out from git and built with CBNG



Releasing new versions of sequencer tasks

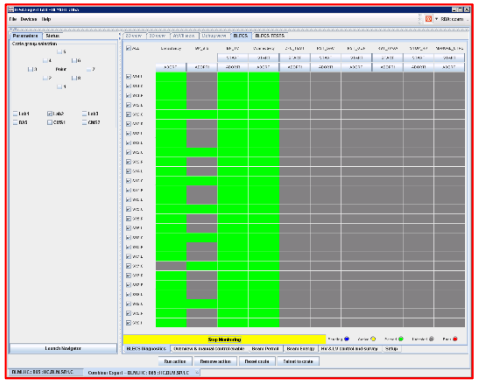
- Very complicated (IMHO)
- Initial changes are normally released to *TEST* sequencer (testing copy of the operational framework meant to test the new developments)
- Then, changes are released to *PRO* sequencer (but still not available to OP!)
- Finally, OP should give the green light so that someone from sequencer-support will *actually* make the release manually
  - The only real experts in the group are GHH and Athanasios
  - sjackson wasn't even part of the egroup sequencer-task-developers e-group until April 2024

Current list of BI tasks

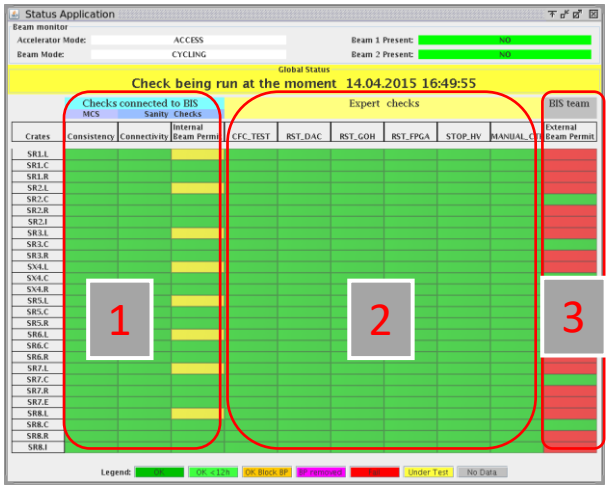
- BSRA
- BCT
  - DC
    - Quick Calibration
    - (24-bit) – Offset Correction
  - Fast
    - Test Dump Acquisition Chain (not currently maintained by BI)
- BPM
  - Calibration – (a)symmetric
  - Transfer Lines – Calibration – symmetric
- DOROS Normal & Collimator
  - Resetting BPMs, setting them up for the various machine modes, checking their status
- BOMEMCHK
  - BLM (not currently activated due to unknown issues with HW)
  - BPM
- **BLM tasks** – Historically developed outside BI
  - MCS online (checks parameters & thresholds in the electronics vs DB)
  - Connectivity (modulation signal to check detector connection and performance)
  - Beam Permit Internal (checks each card can generate interlocks / no electronics degradation)
  - Beam Permit External [Not implemented in SEQ] (checks interlocks from each crate arrive to BIS)

  - Experienced *False failures* > 2023
    - Temporary mitigation done during Q1 of 2024

System Verification & Expert Checks



Expert Application



Status Display

Three groups of checks to validate at any time the system remotely:

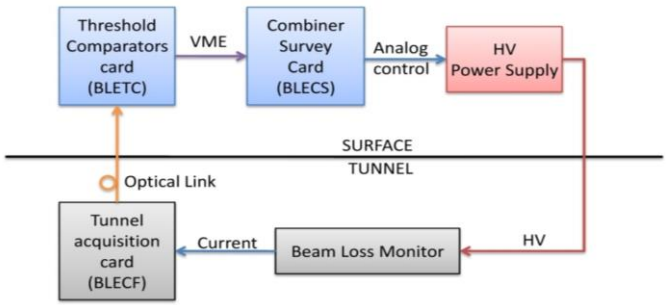
Each group assigned to different teams, i.e.

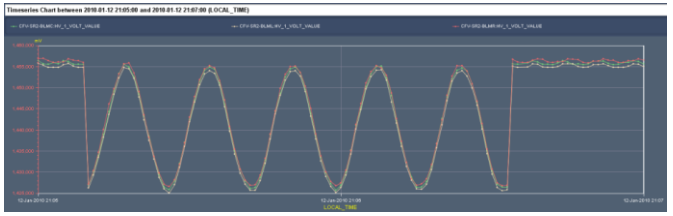
1. Operations crew & System Expert
2. System Expert
3. Beam Interlock System team

- The so-called Sanity Checks verify the consistency of the parameters, the connection & operation of all elements and ability to create interlocks
- These can be executed by the LHC Sequencer or manually by a system expert
- The BLM system ensures that this happens at least once every 24 hours.

Modulation Check (1/2)

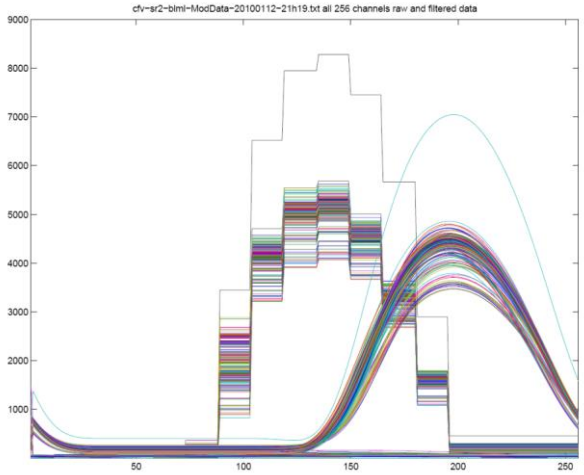
View on the LHC BLM system for the connectivity check.





The high voltage supply to the monitors is modulated with a 60mHz 30V sinusoidal signal.

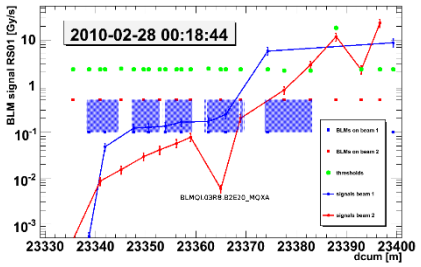
- A current on the monitors is induced by the HV modulation and can be measured by the normal BLM acquisition chain.
- The Combiner & Survey module uses the running sum (RS_09) from the Threshold Comparator modules to determine the amplitude and phase of each monitor (256 channels per crate).
- This results are compared to predefined limits to permit or block the next injection if a non conformity is detected.
- The signals of each monitor is stored to the Logging DB and can be further analysed with the dedicated application.
- The limits are unique for each monitor. They are calculated out of multiple measurements.

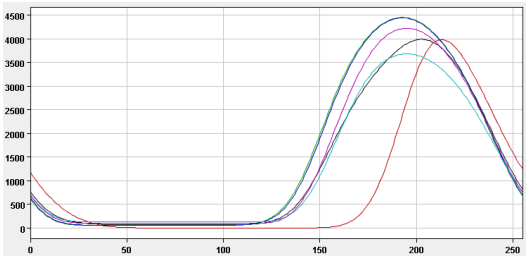


View of the SRAM data containing the original (stairs) and filtered RUNNING SUM 9 data of each channel (256 in total) of one crate. This plots represent one period of the modulation.

Modulation Check (2/2)

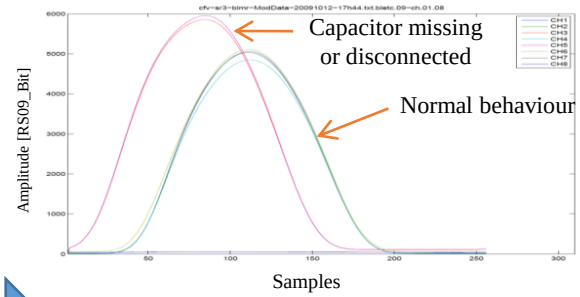
- Main tool for checking the connection & performance of each detector
- During operation/run: discovery of installation degradations

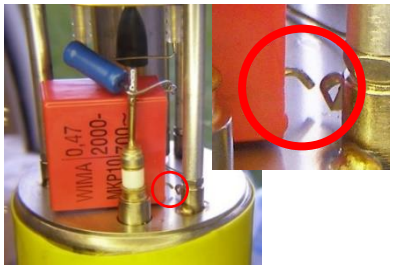




A sensitivity issue was noticed with beam and confirmed with the connectivity check measure.

Type of non-conformities	Occurrences	Deviation from the expected behavior
Chamber filter disconnected or badly soldered	27	9%-33%
Tunnel card (BLECF) non-conform behavior on one or more channel	4	10%-30%
High voltage distribution box	1	Large
Connection of monitor on the wrong channel	2	Large
Monitor not supplied with high voltage	4	Large





Non-conformity discovered during commissioning (top).

BLMLHC problems

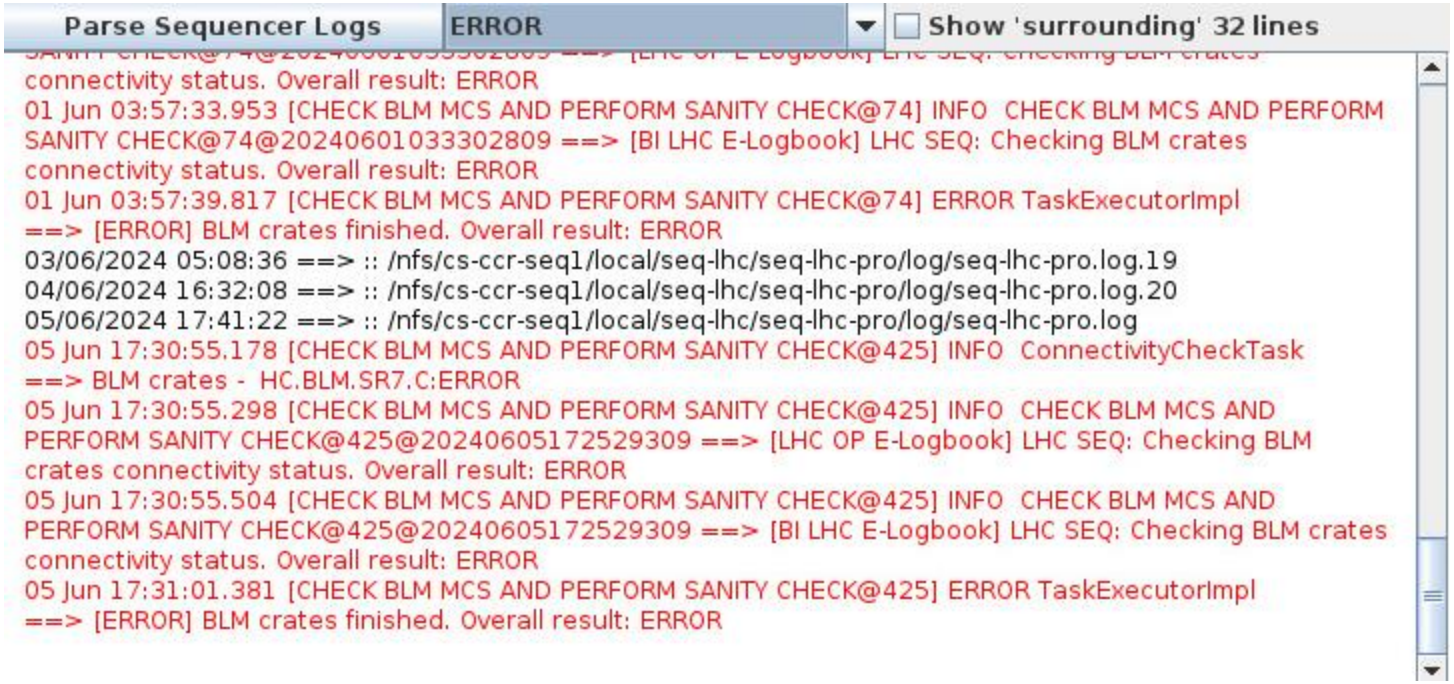
- Until 2024, BLMLHC BPTC check would fail *every-now-and-then*
  - We would test 20 times in the CCC with no errors, then 2 days later a random system would fail with a false-failure
  - The bad code in the task is now visible/failing because of the controls changes done during YETS (change in HW access time)
- Traced to a race condition between the sequencer task and the BLMLHC FESA devices
  - Example sample from sequencer task code
    1. Issues a change of device state
    2. Sleep for 1 second
    3. Reads to see if state changed
    4. If not, ERROR!
  - BLMLHC devices
    - Each 1 second (LHC BP), readout the device state and publish
- Race condition comes when the sequencer task is almost in phase with the LHC BP
  - Solution 1
    - Increase the Sleep to 2 seconds
      - This will inflict a penalty of making the tasks twice as long
  - Solution 2
    - Instead, we give the FESA device a 2nd chance
      4. Sleep another second
      5. Read again to see if state changed
      6. If not, ERROR!
- Solution 2 rolled out March 26th 2024
  - No problem seen since

BLMLHC problems

- Connectivity has often 2-3 times per week false failures.
  - This was always the case.
    - OP have instructions to try again, before asking for support.
  - Reason is the complexity to reconstruct/extract the small modulating signal inside the measurements stream for each of the 4000 channels
    - One ‘small’ spike during check can bias the results and will be resolved as ‘failed’

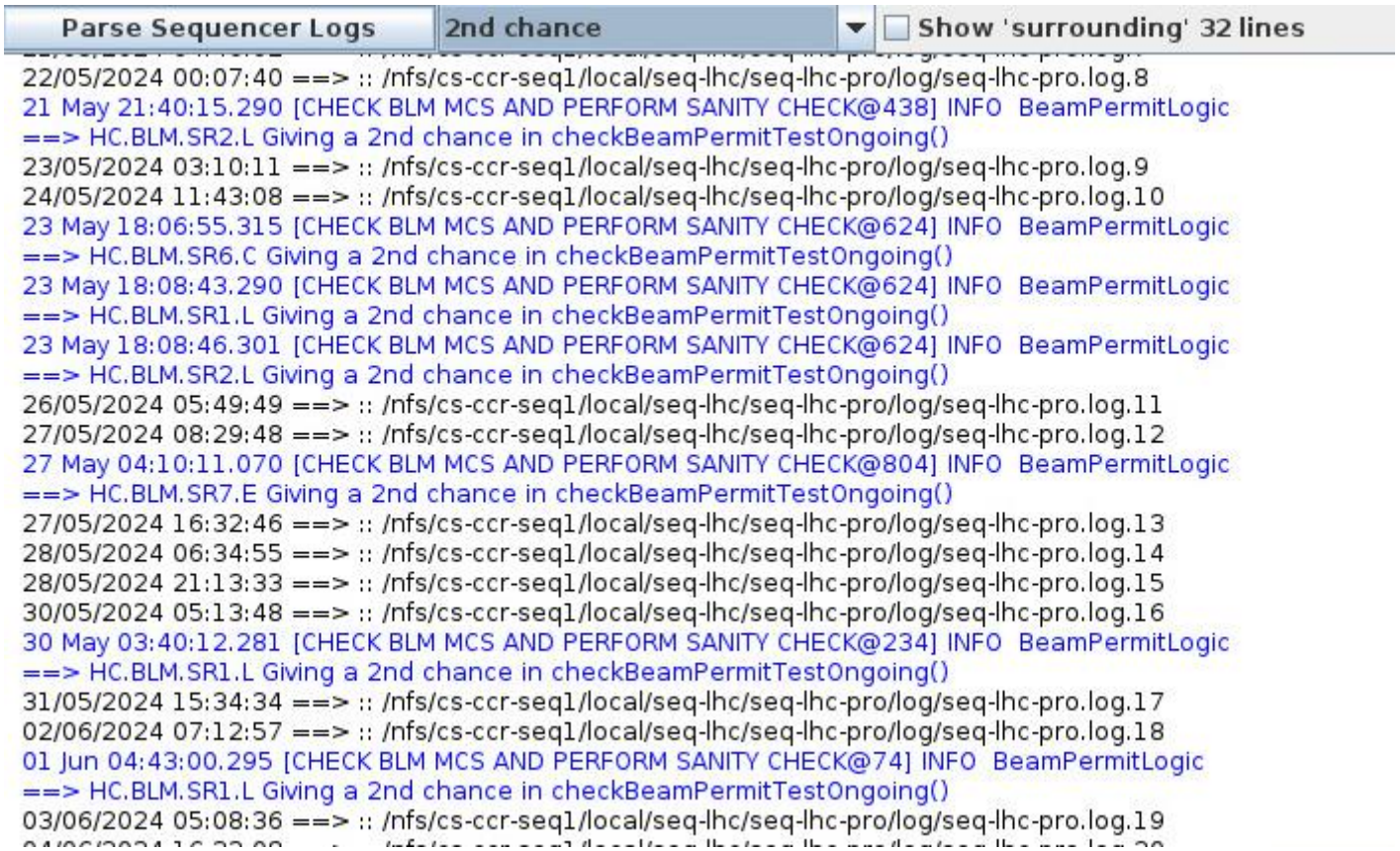
Diagnosing of problems in BLMLHC tasks

- OP Sequencer GUI gives no details on failure
  - Just **FAILED**
- Didn't have access to BI Logbook for *detailed* logging
- Needed to use syslog to work out what was going wrong



Diagnosing of problems in BLMLHC

Examples of where we now catch the race condition from syslog files



Future development

(Possibly YETS 2024, maybe LS3)

- BISW want to maintain all BI sequencer tasks
  - Why?
    - Because if the tasks are completely in our hands we can easily adapt to future changes in the FESA devices
    - … and anyway, some scripts are orphaned

- Align code to a (simplified) common standard based on tasks maintained by Athanasios
  - Many hours spent in spaghetti code
  - e.g. Existing BLM tasks
    - ➢ 3'701 lines of Java code

```
Proposed optimized version -
Only Sleeps for 2*5(locations)=10 seconds instead of for 8*5=40 seconds and should get updates correctly?
for locations in LCRIE
        Sleep 1000ms
        for points in 1..8
                if FESA::Status.offline == true
                        FESA::BLECSUserBPTCCheck.blecsEBPTCR = true
                else
                        Crate XXX is not offline. Cannot enter in test mode."
        Sleep 1000ms
        for points in 1..8
                if FESA::BLECSUserBPTCCheck.blecsABPTC != true
                        "XXX never entered in test mode. BPTC tests failed"
                isBlecsABPTC = true
                while isBlecsABPTC
                        Sleep 500ms
                        isBlecsABPTC = FESA::BLECSUserBPTCCheck.blecsABPTC
                        if isBlecsABPTC = false
                                testResultBPTC = FESA::BLECSUserBPTCCheck.blecsRBPTCP
                                timerResultBPTC = FESA::BLECSUserBPTCCheck.blecsTSTRN
                if testResultBPTC & !timerResultBPTC
                        "SUCCESS"
                else
                    if timerResultBPTC
                        "Timer for mandatory BPTC test not reset"
                    else
                        "FAILED"
```

100s of lines of Java code can be expressed in a few lines of pseudo-code

Future development
(Possibly YETS 2024)

- BLMLHC class
  - Replace the polling mechanism with a subscribe/command/response mechanism
    - Will imply small changes on the BLMLHC FESA class side
  - Unless we are very confident we will postpone operational rollout until LS3
    - But would be good to already test in YETS
  - Discussion with M. Saccani if we redo some of the logic in LS3

Impressions on current BLMLHC code-base

- Several useless layers of logic in all the code
  - Drilling down to the *actual* code is a journey
  - Often quicker to *grep* the code to find where to look
- Lots of dead-code
  - Not clear what is and isn't executed
  - Seems to be a lot of historical code which was never deleted
- Code quality quite poor in places
- Based on polling rather than subscribe/publish
- Some naming of variables and methods is incorrect
  - e.g. BLETC is commented everywhere as *Beam Permit Threshold*
    - Very confusing
- No flexibility to run *parts* of the BLM tasks
  - Need to rerun everything in case of 1 failure. Can lead to failure ping-pong
    - Each iteration takes 20 minutes
    - During interventions or errors, large amount of time is lost waiting for the tests to run everywhere
    - Aim, when needed, individual (per crate) execution.

Possible new tasks (from presentation 2016)

- Additional tasks
  - BLM Verification Checks (currently orphaned)
  - BST – Synchronicity Check (was there but removed at some point)
    - Should be revisited in new WR-based system > LS3
  - BPMs Transfer Lines – Calibration – asymmetric (if needed)
  - Other systems ? (BSRA, BSRT, BRAN, BTV, Wire Scanners, OFSU, Tune)

- “Reference” Settings checks
  - Unclear what the idea was here…

Possible new tasks (from presentation 2016)

- Results on sequencer task execution
  - OP logbook (summarised)
  - BI logbook (detailed)
    - *Broken* until recently
    - In fact it wasn't broken, but we had no easy link to it (thanks G. Trad for follow-up)
  - syslog (when things go really wrong!)

- Proposed
  - OP and BI logbook as is (OP: Summary Vs BI: Detailed)
    - Done
  - e-mail to specialists only in case of error
    - Done
  - New logging for post/statistical analysis (OAF – JJ era)
    - Define Result Structures
    - Define analysis tasks
    - Is this still needed?
    - Build on NXCALS reports already done for BOMEMCHK

Tentative roadmap

- 2024 TS1
  - Follow-up merge requests with OP (Delphine++)
    - Standardize code layout for Gradle (Roman)
    - New logbook tag for sequencer entries to improve the logbook readability (Delphine)

- 2024 Q2->Q4
  - Analyze existing codebase of all scripts for BI equipment
  - Check what is deployed and what's in the backlog
    - E.g. Changes from 2018!
  - Study any new requests for BI equipment

- 2024/5 YETS … LS3
  - BLMLHC
    - Replace the polling mechanisms with a subscribe/command/response mechanism
      - Changes in Sequencer and FESA class
    - Remove dead code
    - Make code more modular
    - Look at refactoring code to a new standard defined by Athanasios

- LS3
  - Adopt all existing scripts under common standard

Questions