Experience from LHC Hardware-Commissioning (HWC)

K. Fuchsberger

Introductory Remarks

– Thanks to M. Zerlauth, R. Schmidt, M. Solfaroli Camillocci and M. Pojer for providing material and ideas!
– Copyright CERN for all pictures and Schematics/Graphics in this presentation.
– Several screenshots were done by myself (K. Fuchsberger), as I was for a long period part of the team, developing and improving several concepts and tools mentioned in this presentation (e.g. AccTesting, eDSL)

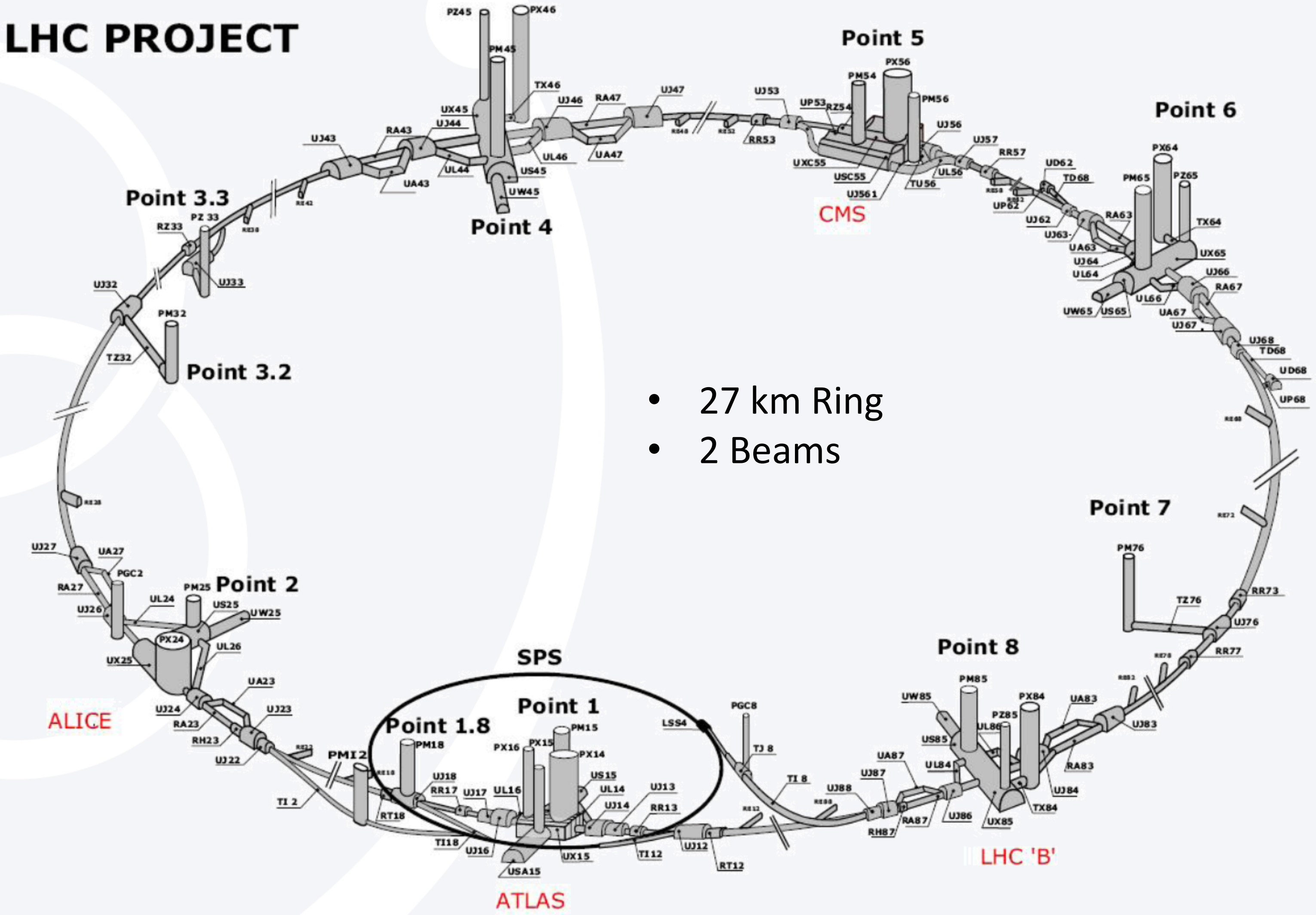
Content

- Introduction: The LHC Circuits
- Key Approaches/Tools for HWC
- Summary

Introduction

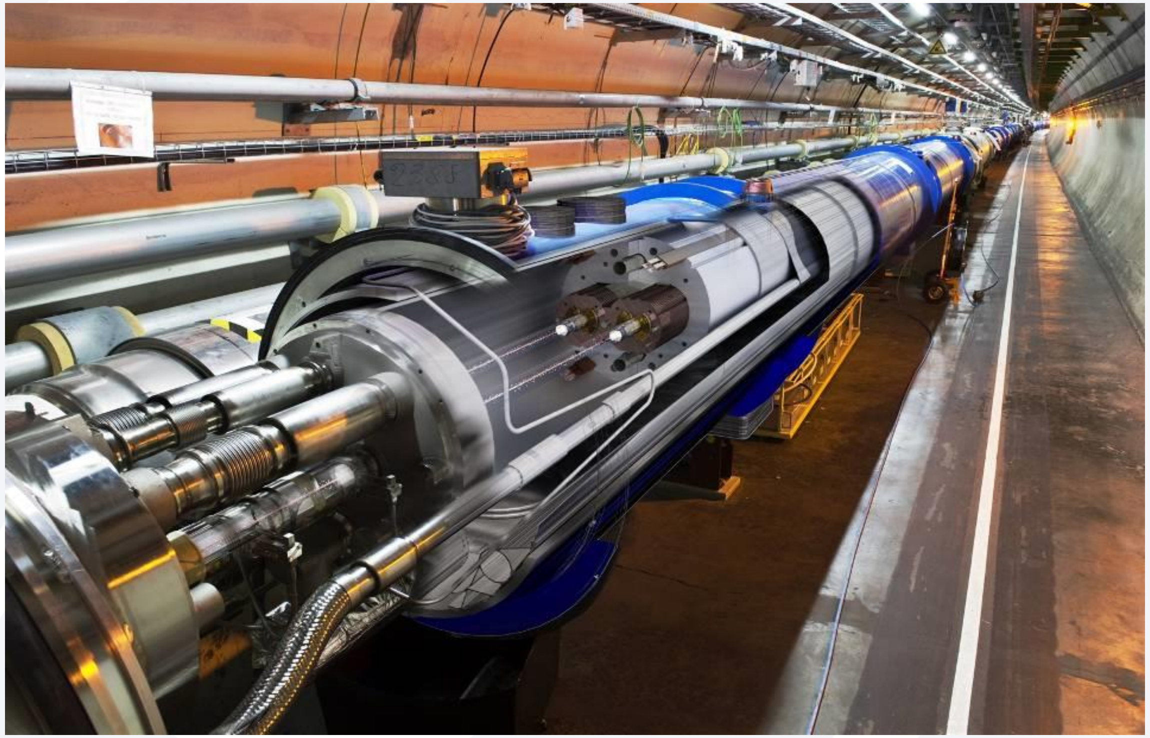
… The LHC Circuits

The LHC



- 27 km Ring
- 2 Beams

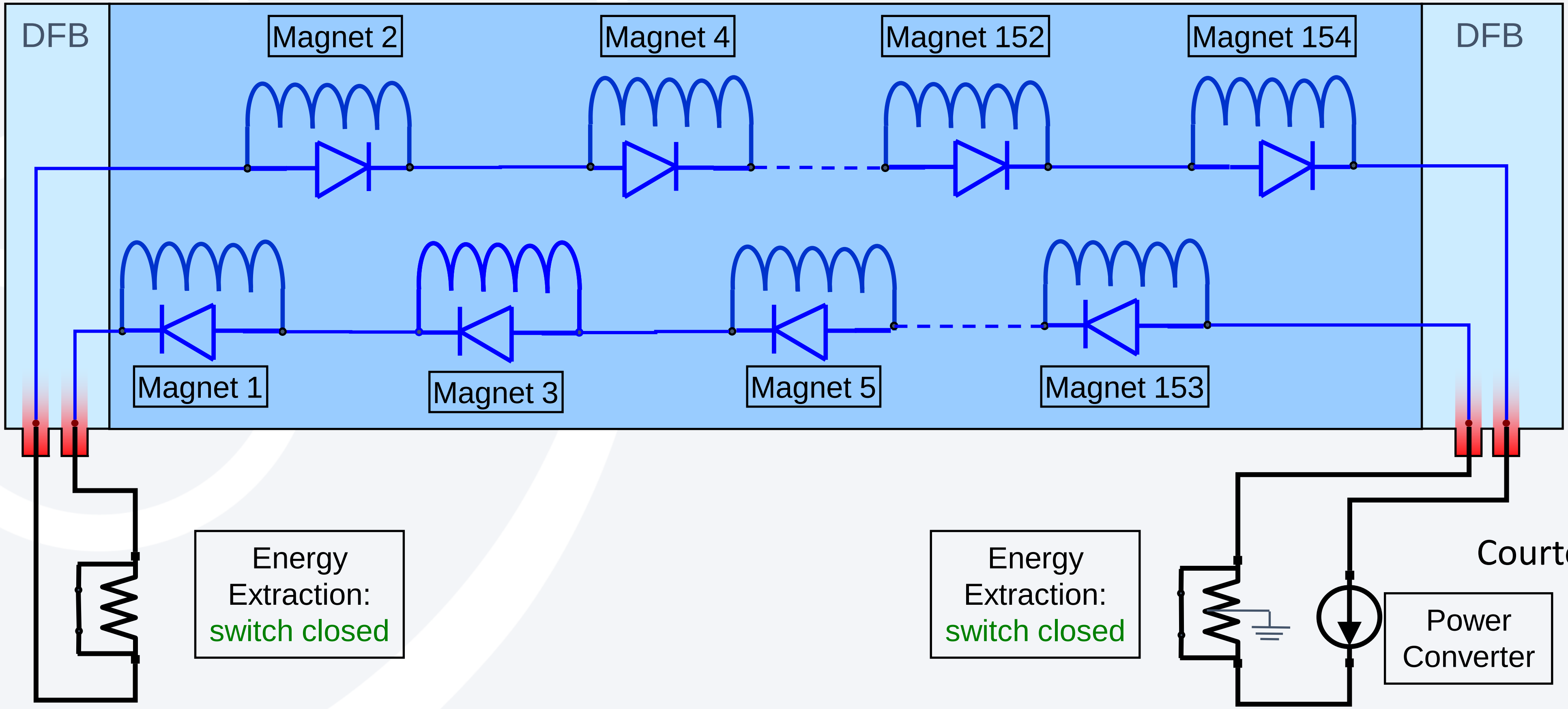
LHC magnets inventory



- More than 10.000 superconducting magnets of 50 different types, e.g.:
  - 1232 main dipoles (powered in series)
  - 752 orbit corrector magnets (powered individually)
  - 392 Main Quadrupole magnets (powered in series)
  - Triplets at Interaction points, correctors, …
- → In total 1618 Electrical Circuits
- → 9 Circuit types, depending on:
  - Stored Energy, Risks
  - Protection Elements (for magnets, busbars and current leads)
- Examples:
  - Individual orbit corrector: 60 A, 9 kJ
  - 154 mains (series): 12 kA, 1.2 GJ

Exemplary Circuit: Main dipoles in arc cryostat

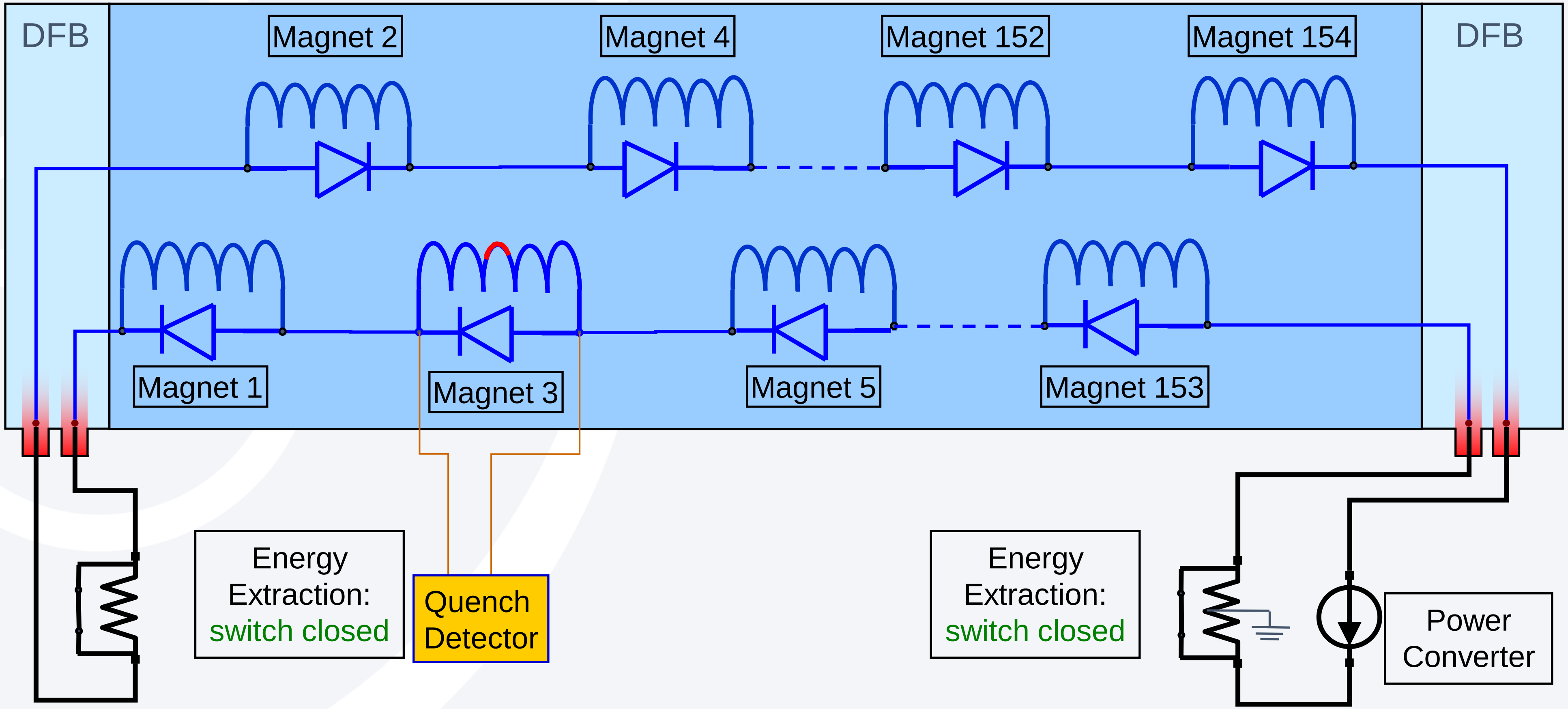
- Time for the energy **ramp** is about **20-30 min** (Energy from the grid)
- Time for regular discharge is about **the same** (Energy back to the grid)



Courtesy: R. Schmidt

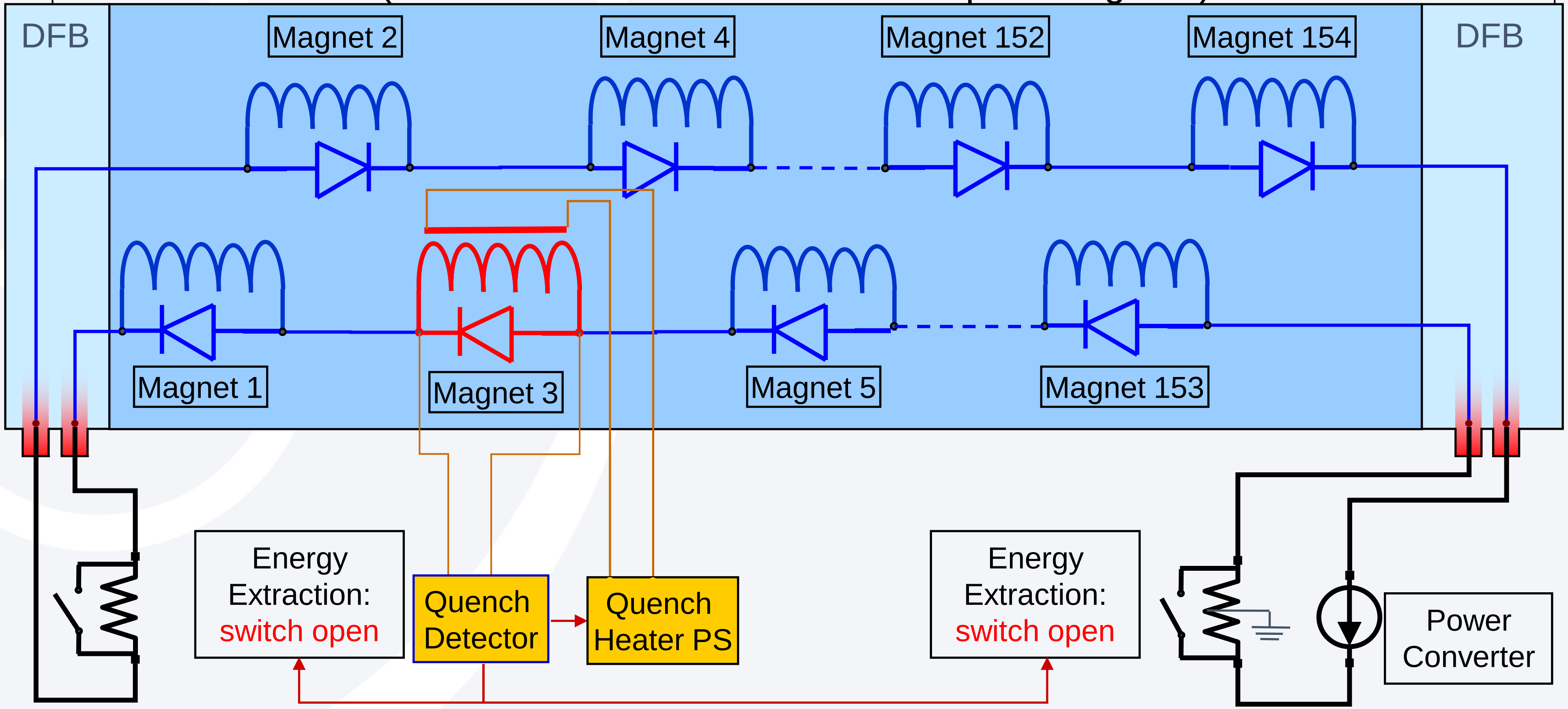
Main dipoles: quench of a magnet

- **Quench in one magnet**: Resistance and voltage drop across quenched zone
- **Quench is detected:** Voltage across magnet exceeds 100 mV for >10 ms

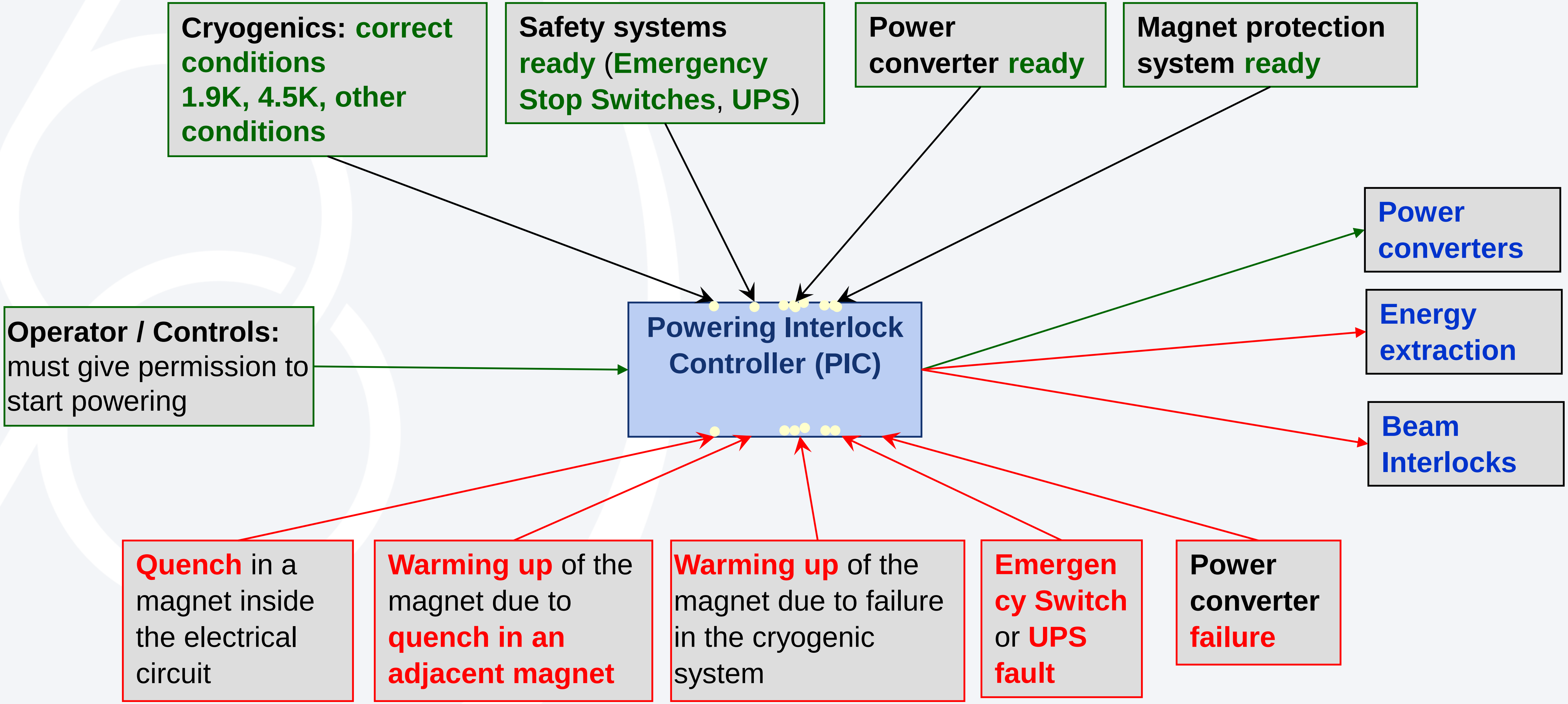


Main dipoles: magnet protection

- Quench heaters warm up the entire magnet coil: energy stored in magnet dissipated inside the magnet (time constant of 200 ms)
- Diode in parallel becomes conducting: current of other magnets through diode
- Resistance is switched into the circuit: energy of 153 magnets is dissipated into the resistance (time constant of 100 s for main dipole magnets)



Conditions for powering



Summary: LHC HWC Challenges

- 1618 Superconducting Magnet Systems (!!!):
  → „Integration Testing“ (in the real environment)
  Magnet + Power Converter + QPS + EE + …
- ~ 7000 individual Tests in total
  - Magnets performance
  - Protection functionalities
- To be reproducibly repeated each year after Christmas Shutdown and/or whenever a sector is warmed up.

Key Approaches/Tools for LHC HWC

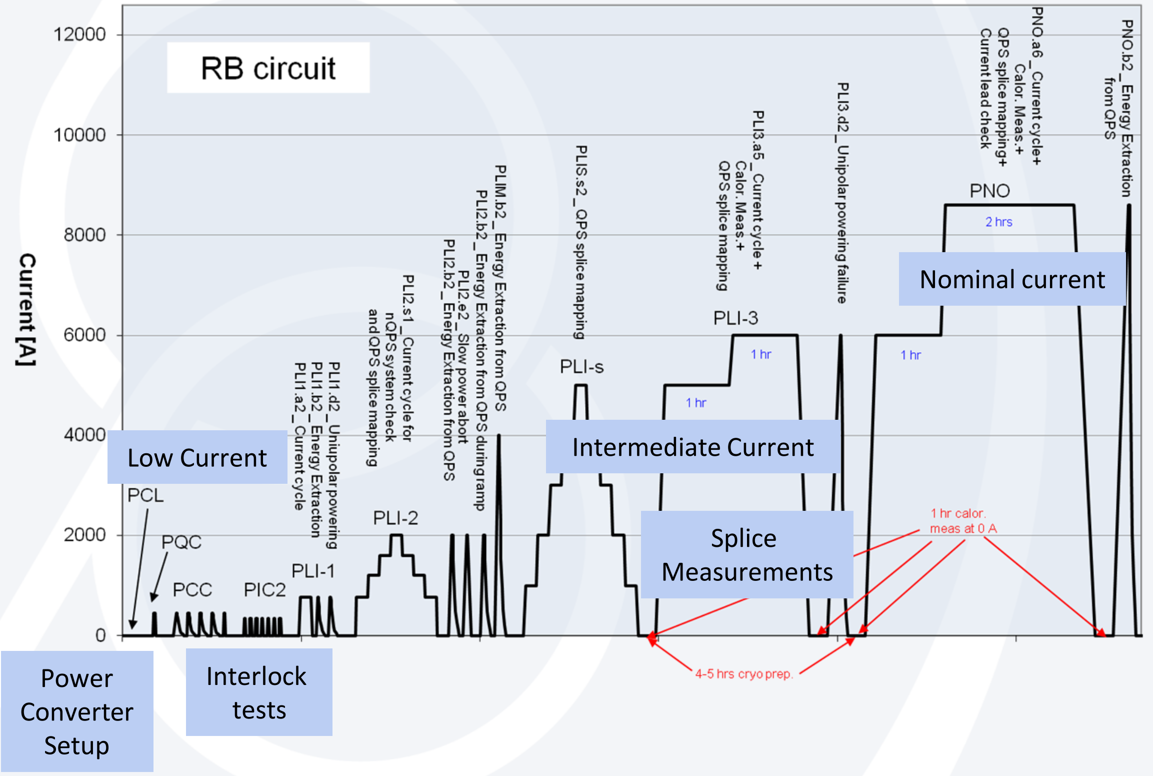
… It‘s been a long road …

… getting from there to here …

24th April 2007
Start of LHC S/C circuits commissioning

- Initial plan: HWC in the tunnel (close to the electronics).
- Soon decided, that this is not necessary (System built to be remotely controlled ;-), and not super-comfortable …

0) How to test? → Test Procedures

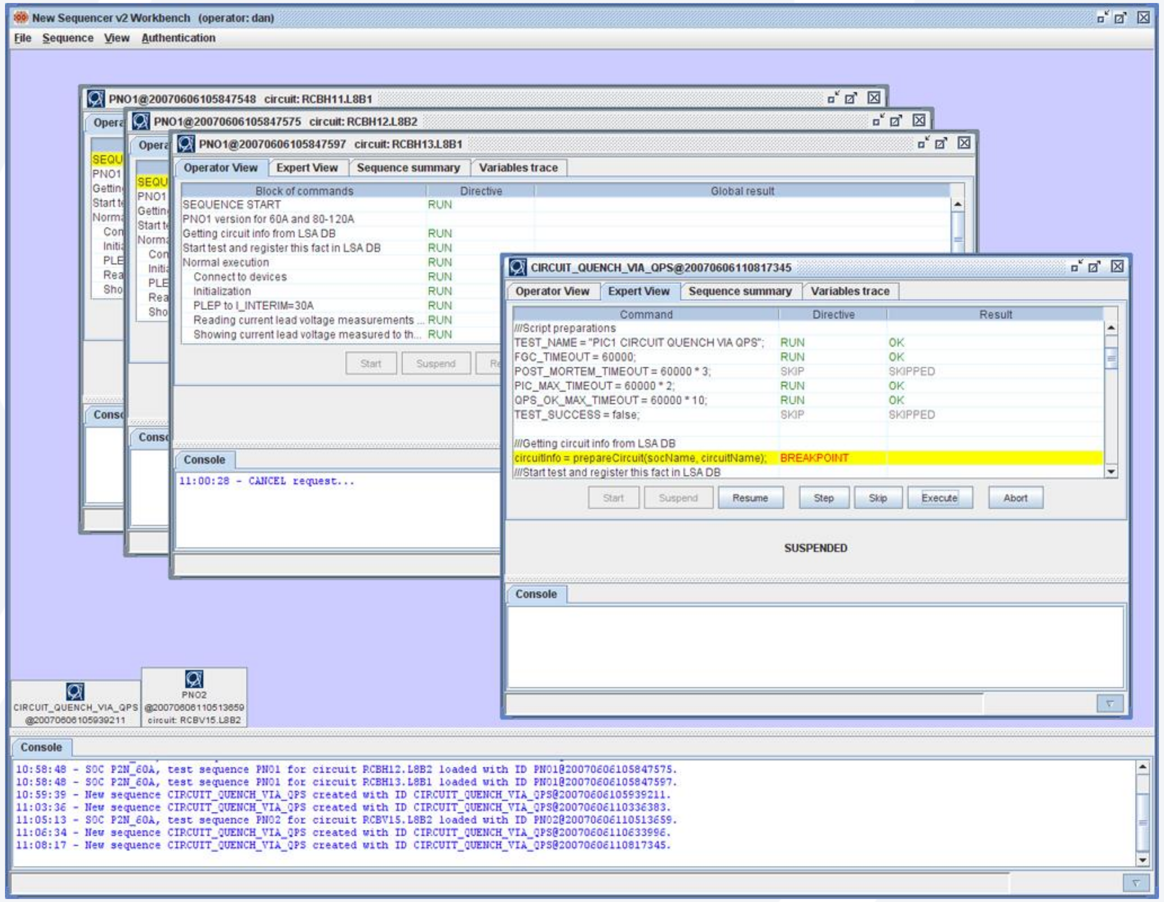


Example Procedure: 600A Circuit (Courtesy: CERN)

- Textual step by step Guide for particular circuit type + Guide to Analysis
- Contains Parameter for the tests (might change from campaign to campaign), potentially per circuit
- Maintained and updated before every campaign. Still today the ‚only source of truth‘

→ Reference / Only source of truth

1) Efficient Execution → HWC Sequencer



– Kind of ‚scripting' of repetitive tasks → „Sequences".

– Possibility to step through the Tests

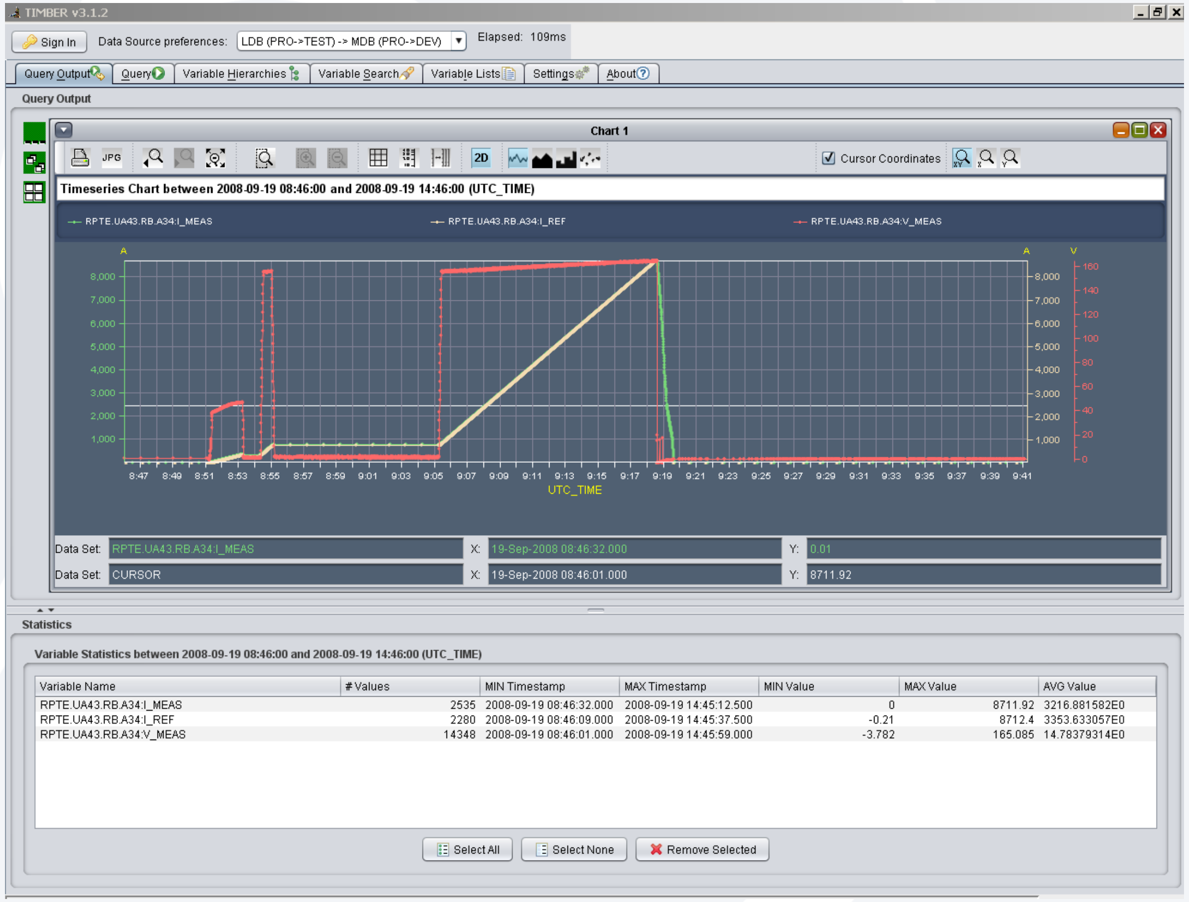
– Different modes:

  – Strict (fails the full test on any problem)

  – „Expert" mode (e.g. Failed tasks can be retried)

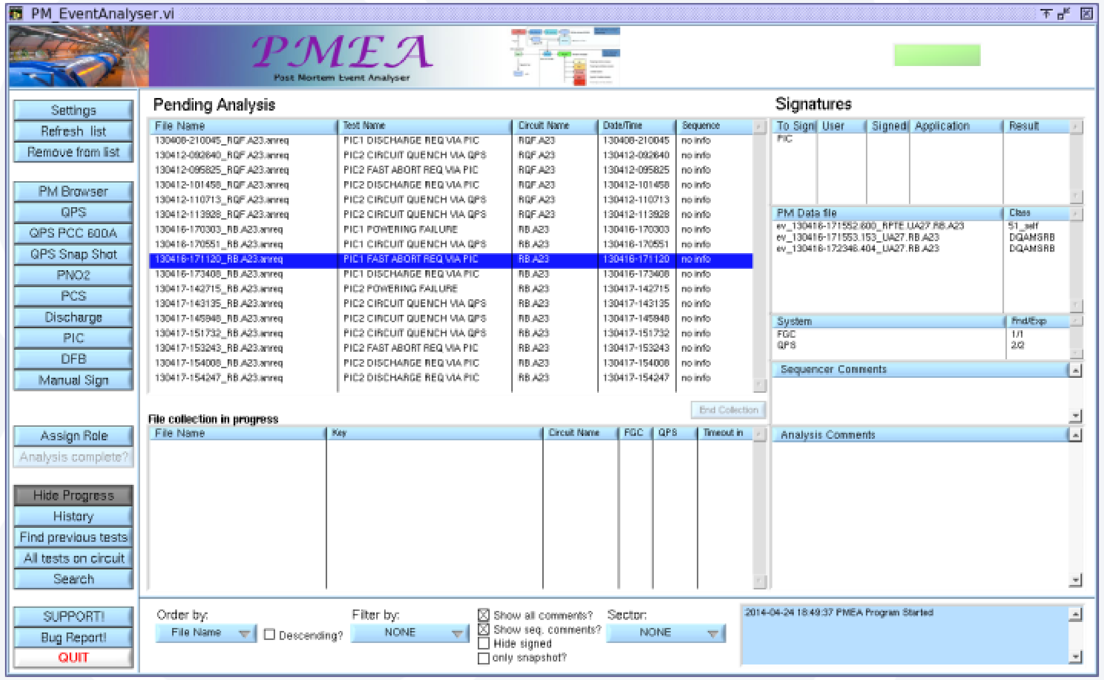
→ Very similar Concept already almost ready at GSI.
→ See presentation of S. Krepp

2) Knowing what happened → Signal Data



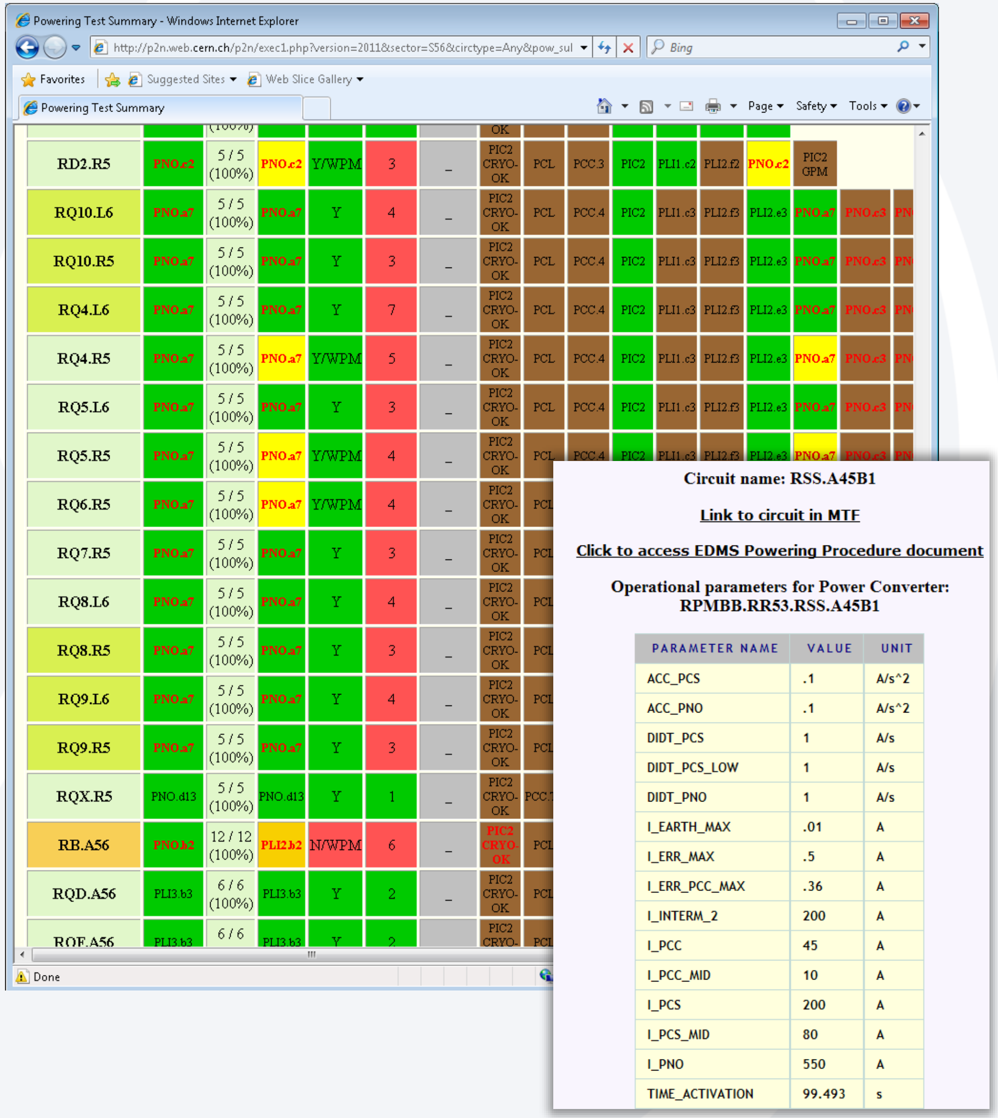
- (Powering) Post Mortem System:
  - Event driven (e.g. Triggered by power aborts)
- Logging Database (CALS):
  - Logged continuously
  - Lower resolution; (potental) data reduction.

3) Test Analysis



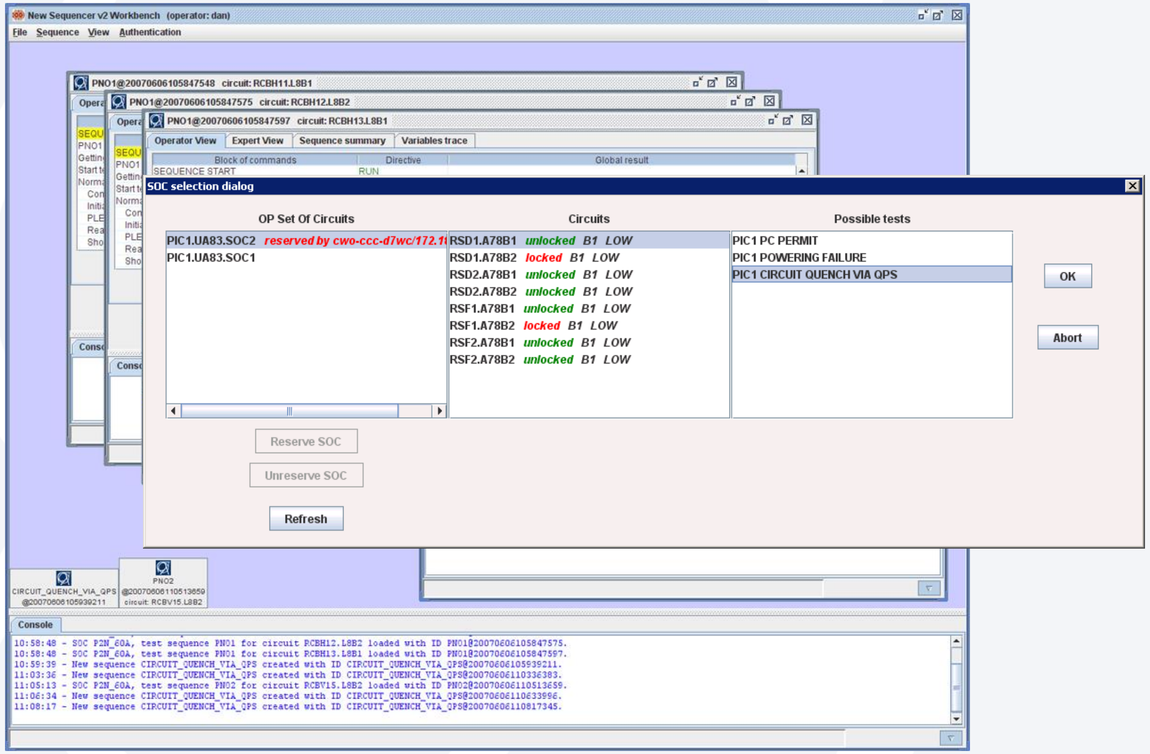
- Some ideas in the beginning that sequencer could also do (some) verification → Proofed to be difficult (e.g. Long time until data arrives, Complexity of Analysis)
- → Analysis was kept separate from Execution
- Analysis in the beginning done manually (PMEA – LabView tools)
- More and more automation over time:
  - Semi-Automation (e.g. Pulling data together and visualizing appropriate for the specific Test)
  - Some fully automated Test modules in LabView

4) Keeping the overview



- In the beginning: Excel Spreadsheets by Rüdiger
- Later: “Alvaro’s pages”: Dedicated website. (php, directly accessing LSA DB – where test results were stored)

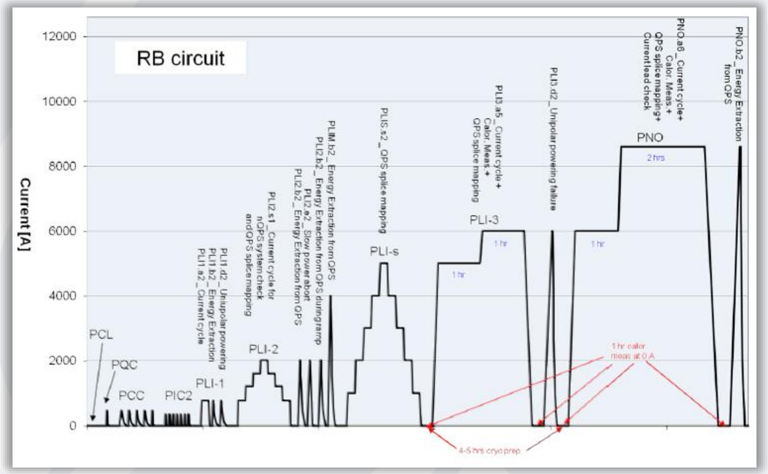
5) Avoiding Testing Crosstalk

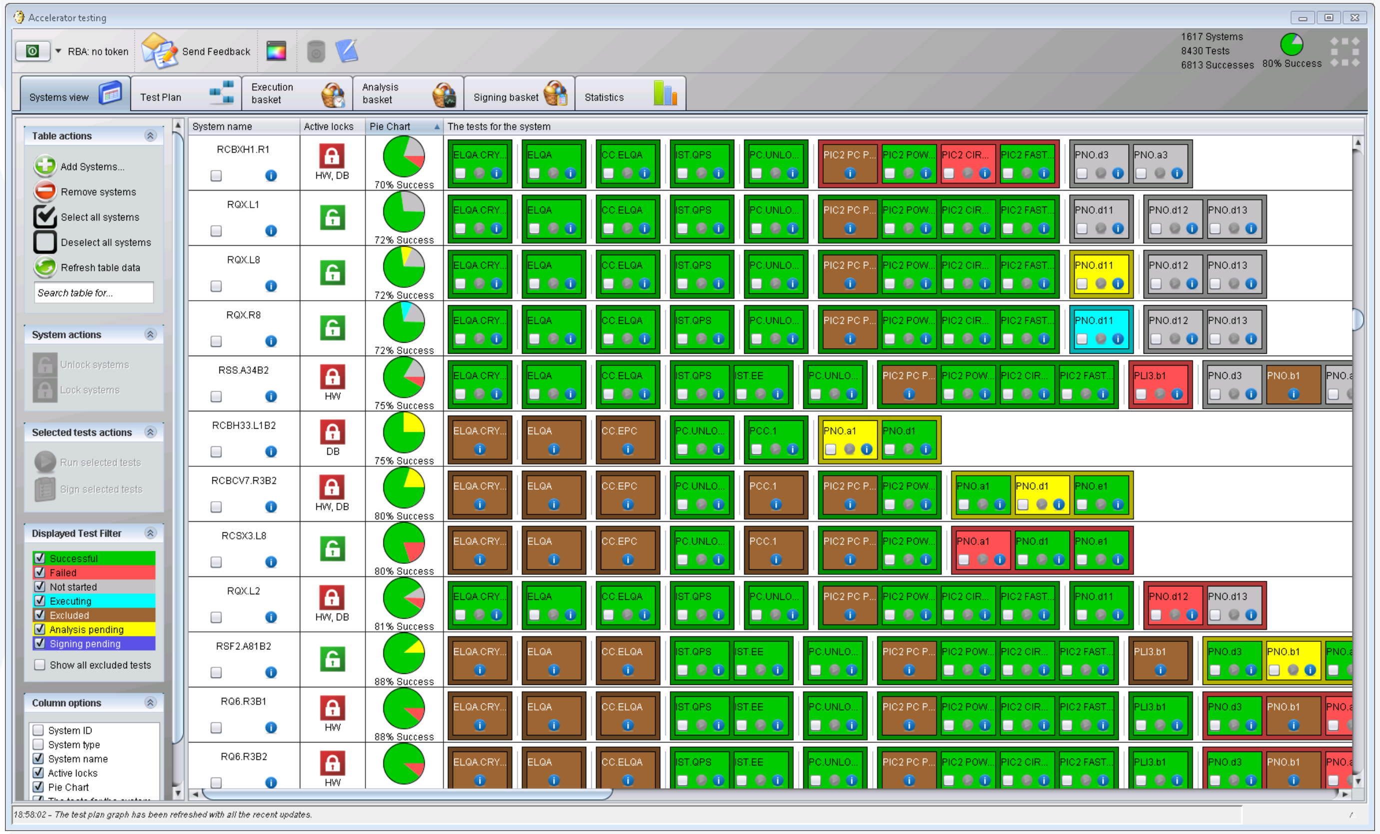


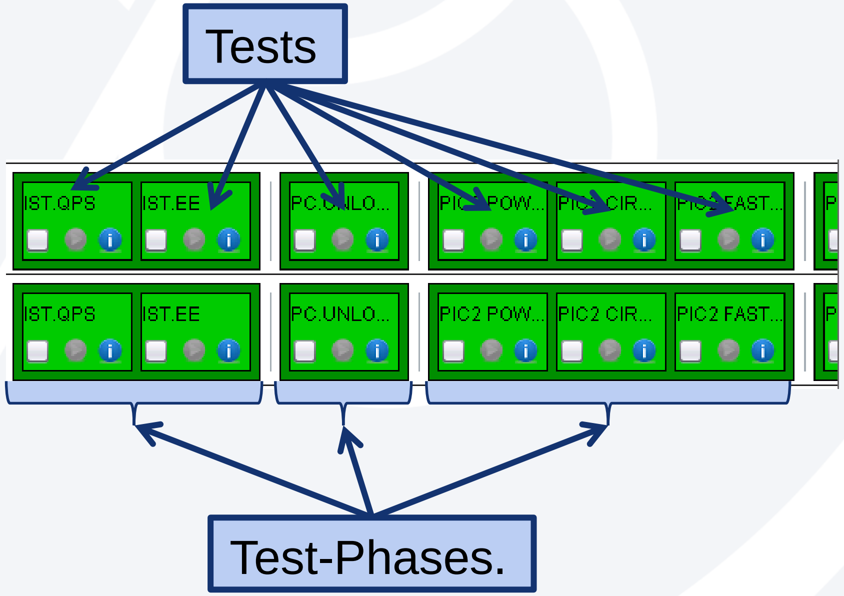
- Parallel Test execution: To avoid that several Operators would access the same hardware:
  - Reservation of „Set of Circuits“
  - Sequencer „locks“ (db) a circuit during execution.
- Still much interference between the tests:
  - Some Tests (e.g. of close circuits, or same QPS controller) must not be executed at the same time (fail otherwise)
  - Some tests on different circuits MUST be executed at the same time .
  - …

→ A lot of Knowledge required in the head of the operators!

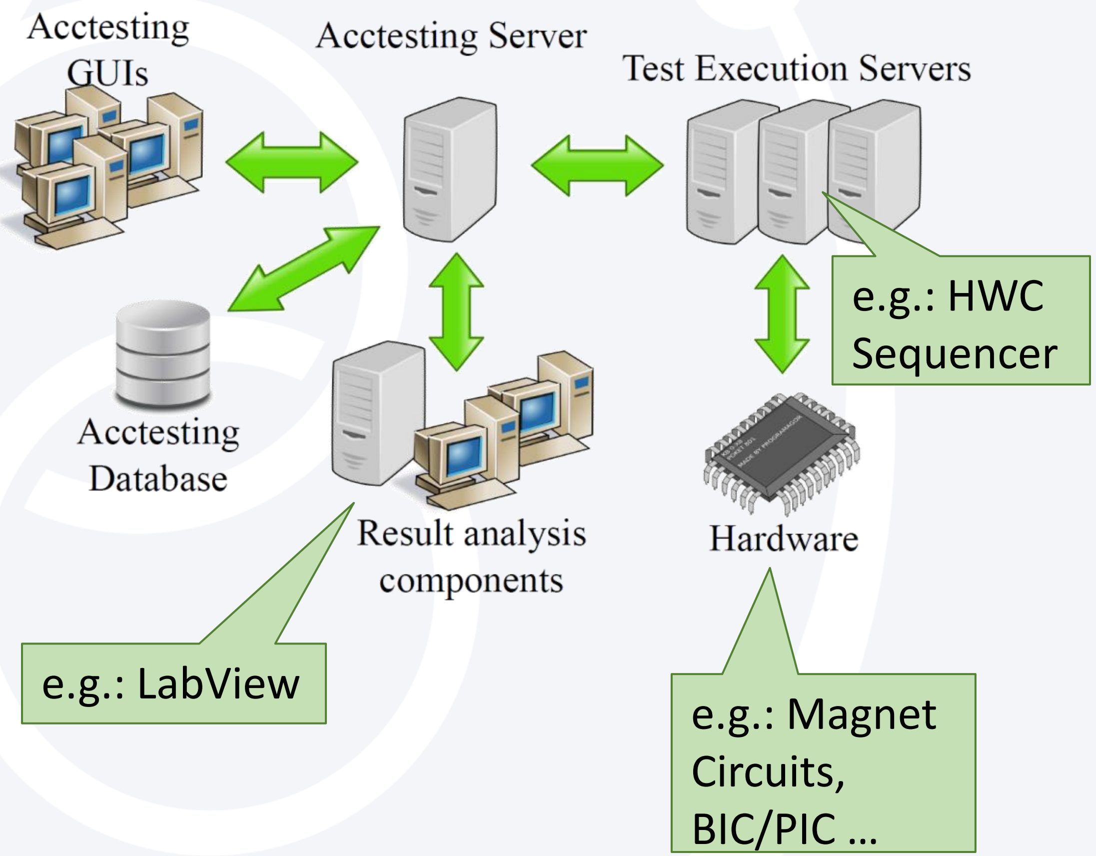
6) Orchestration → AccTesting





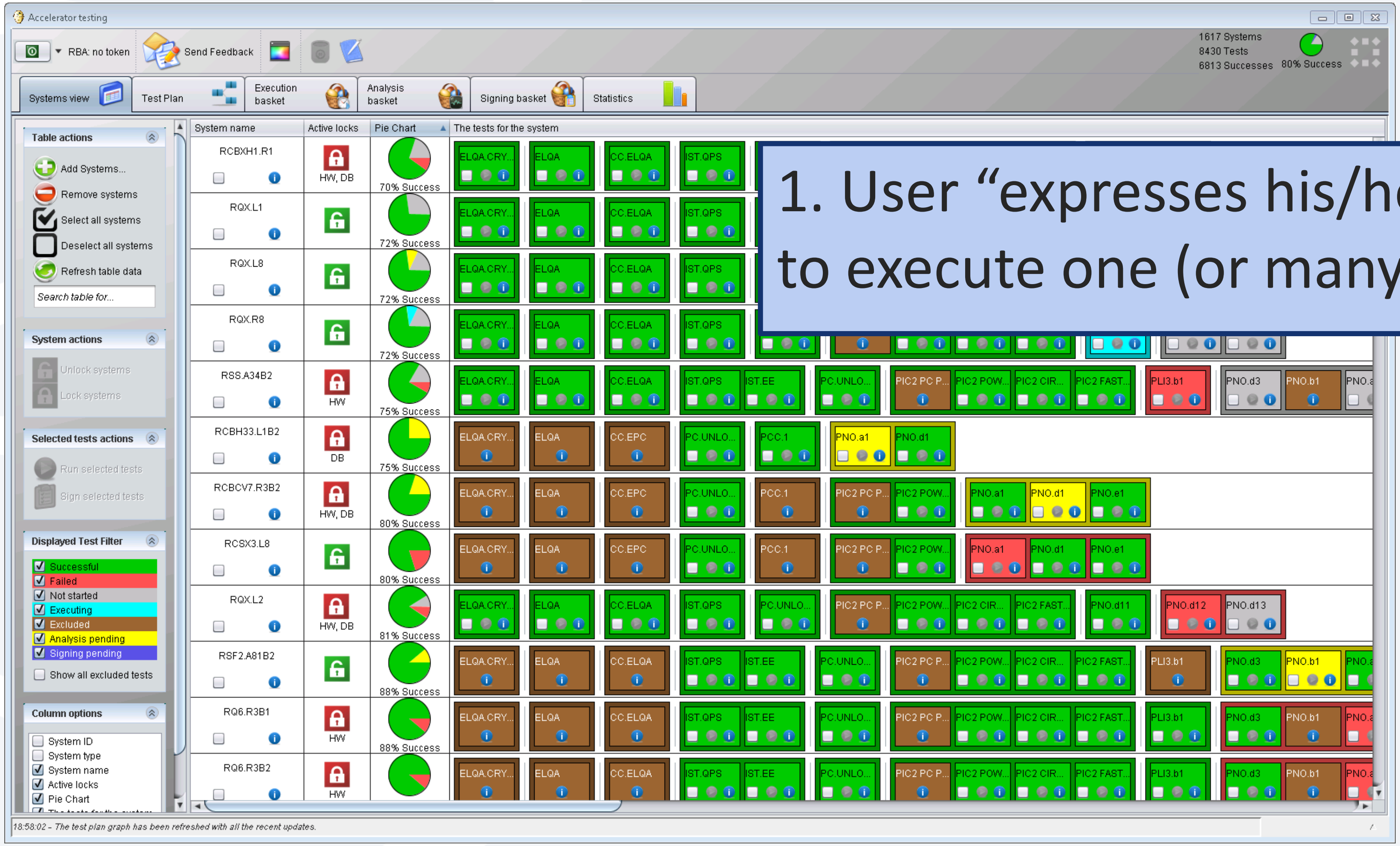


AccTesting Concepts



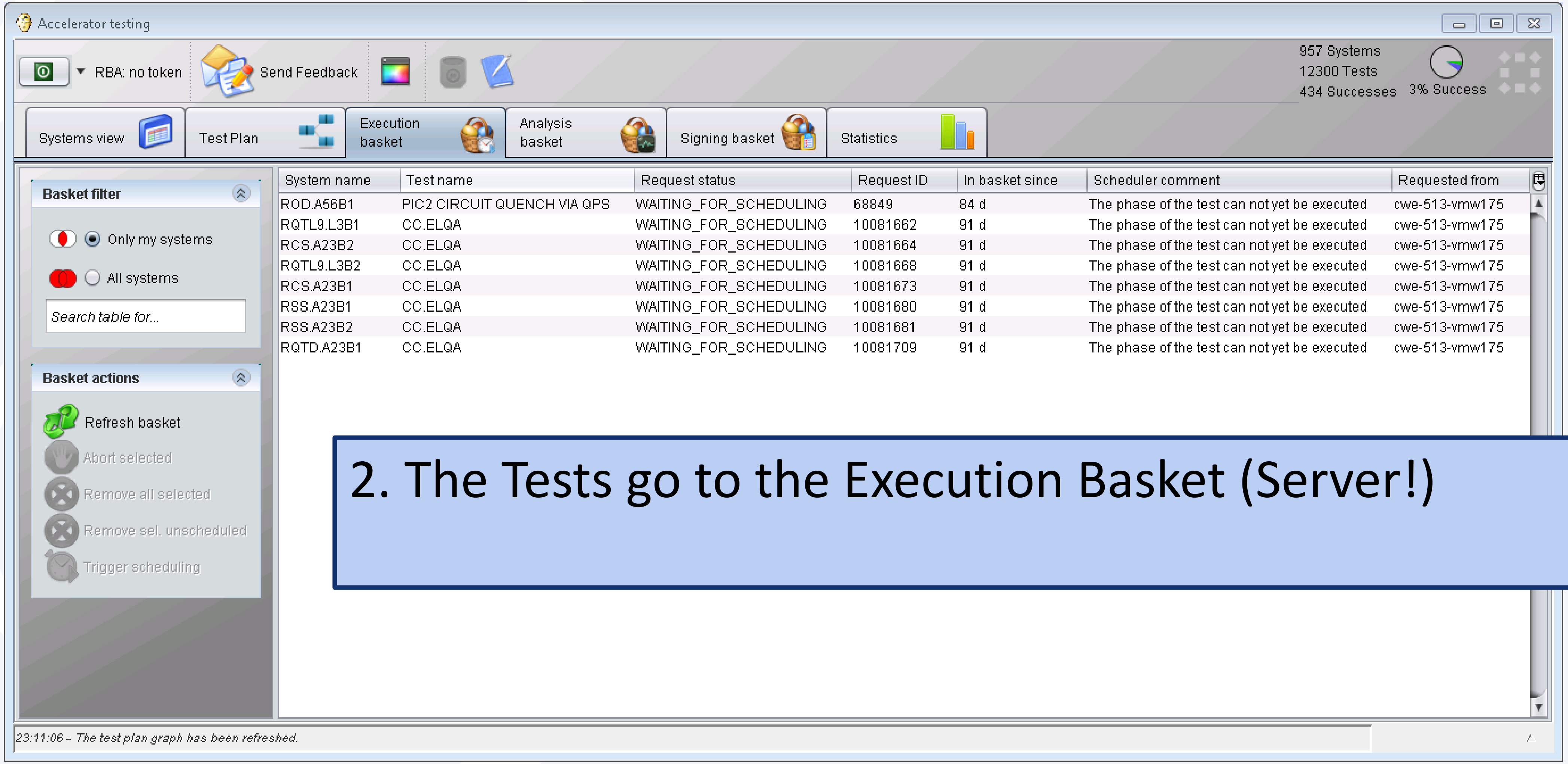
- Central Server with many Uis
- Open Architecture, which supports ‚plugins' for execution and Analysis.
- →Integrates all the existing tools.
- Takes care of all the tricky things:
  - Correct Order of execution
  - Locking of Circuits
  - Tracking of results
  - Avoiding Crosstalks (through „Constraints")
  - Signatures

AccTesting - Workflow I



1. User "expresses his/her wish" to execute one (or many) tests.

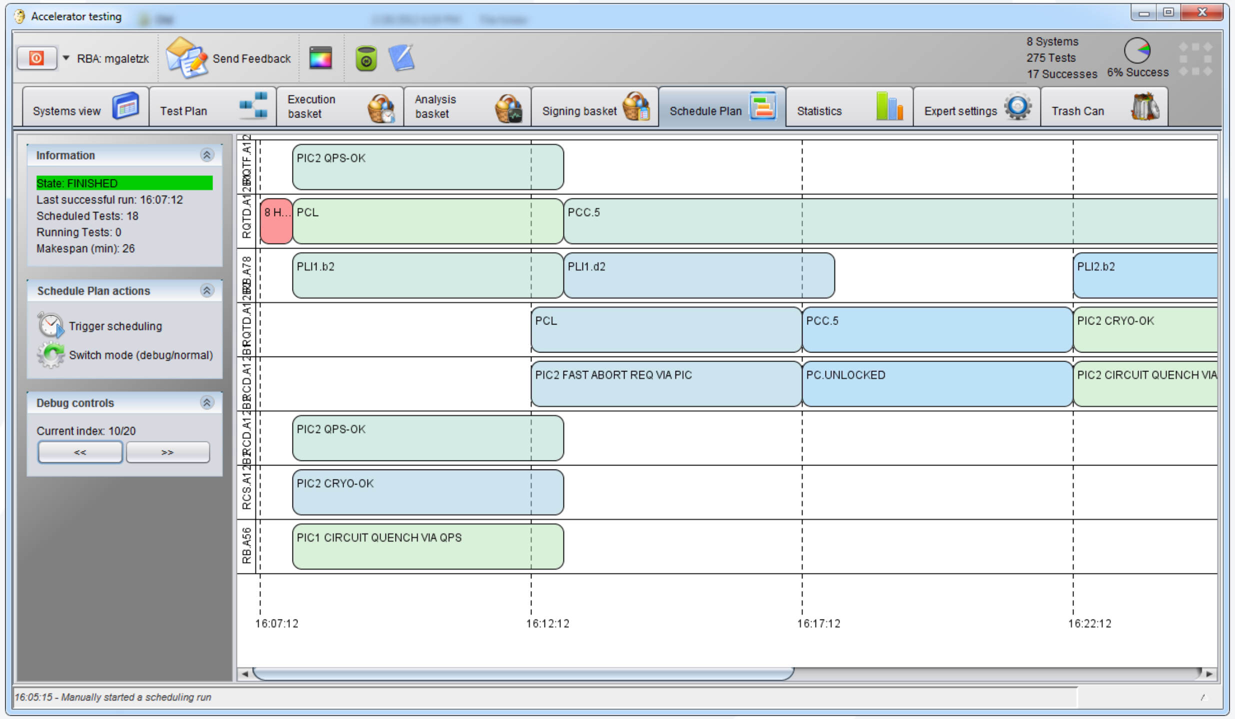
AccTesting - Workflow II



2. The Tests go to the Execution Basket (Server!)

AccTesting - Workflow III

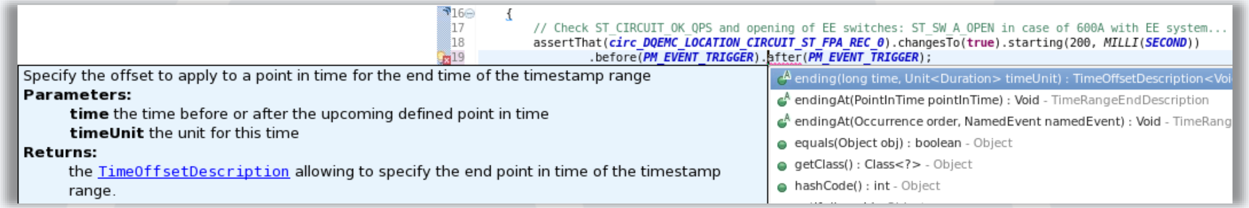
3. The Scheduler (on the Server) will decide when to start which test(s).

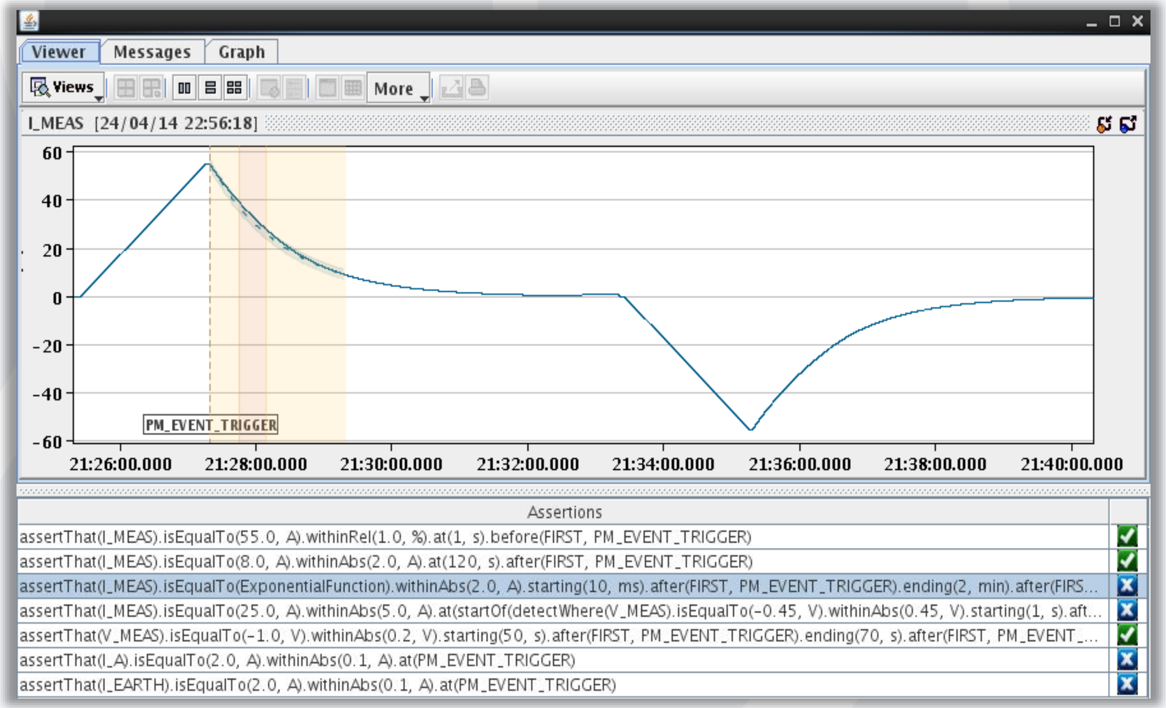


- Central Scheduling can respect all the conditions (Phases, Locks, Constraints …), even if requests come from different GUIs.
- No Test started, if preconditions are not fulfilled.
- When conditions are fulfilled later, the tests are started automatically (No delays).
- No Need for reservation of Systems

→ Big boost in Testing throughput!

7) Analysis Automation → Analysis DSL





- Small domain specific language, integrated in java.
  - Code completion, grammer, javadoc → out of the box
  - Supports fixed set of conditions
  - Can be run on extra servers (scalable)
- GUI plugin directly for AccTesting → direct presentation of assertion Results.
- Brought large gain in time, in particular for tests of the (numeroes) small circuits.
- Takes away ‚monkey analysis' from the experts and saves their energy for the „tricky cases"

Summary

… and lessons learnt

Summary and Lessons learnt

- Good (not neccesary long) procedures are the basics. If kept up to date, together with the results of the test, serve as essential documentation.
- Automation pays off! (Tests are frequently repeated)
  - Execution: Sequencer (see also Stefan‘s talk)
  - Analysis
- Dumping data is easy, finding it again might be tricky …
  - Archiving System → see Vitaliy‘s presentation
- Orchestration avoids mistakes and saves time!
- Not everything has to be perfect from the beginning (it anyway wont;-). Better:
  1. Start from „the simplest thing that might possibly work“
  2. Try + use it → learn
  3. Iterate („goto 1“)

Thank you!