



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

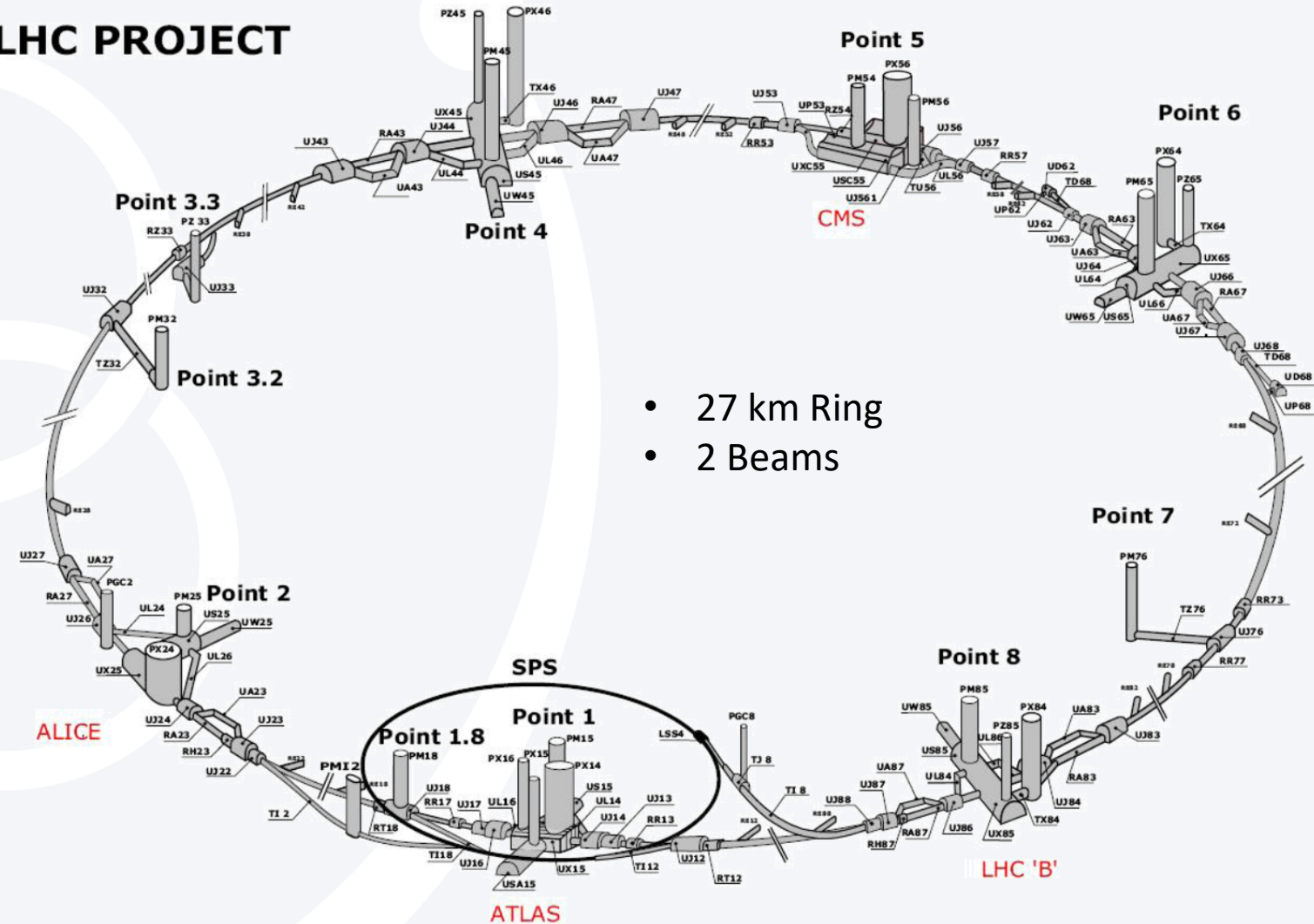
Introduction

... The LHC Circuits



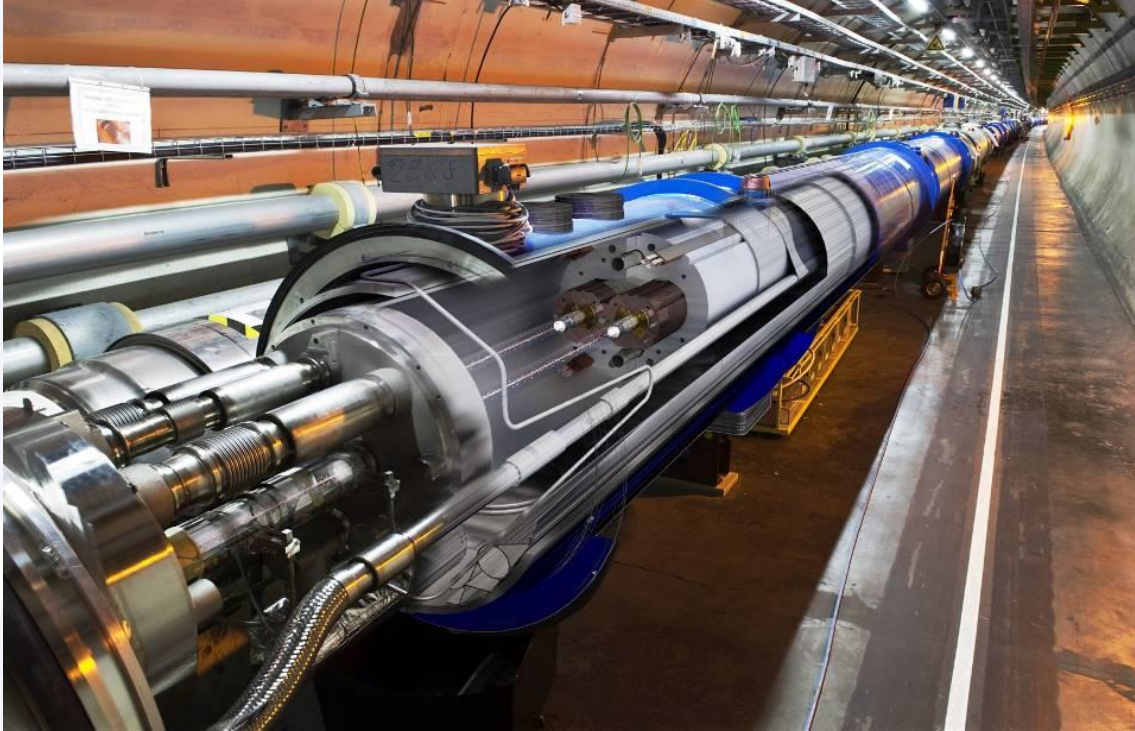
The LHC

LHC PROJECT



- 27 km Ring
- 2 Beams

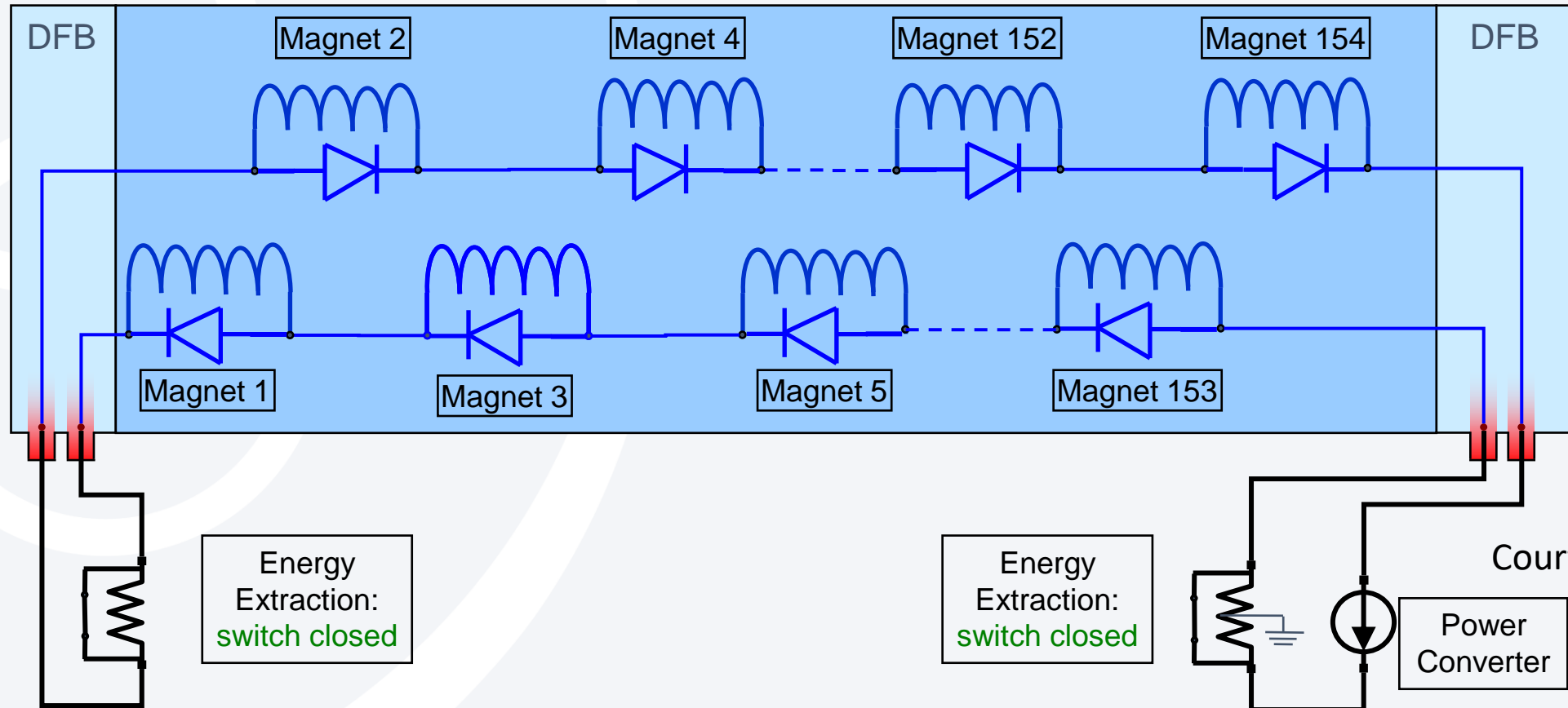
LHC magnets inventory



- 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**

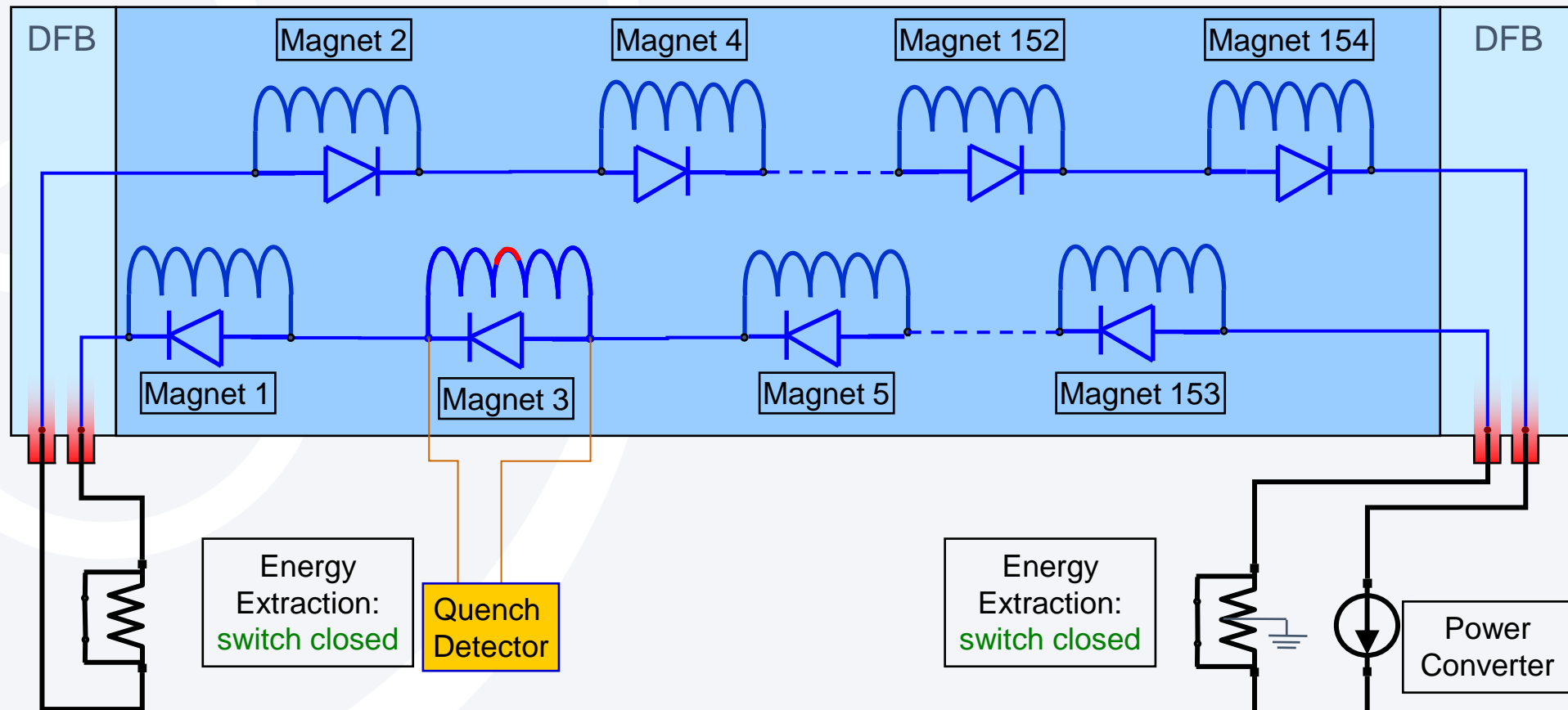
Exemplary Circuit: Main dipoles in arc cryostat

- 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)



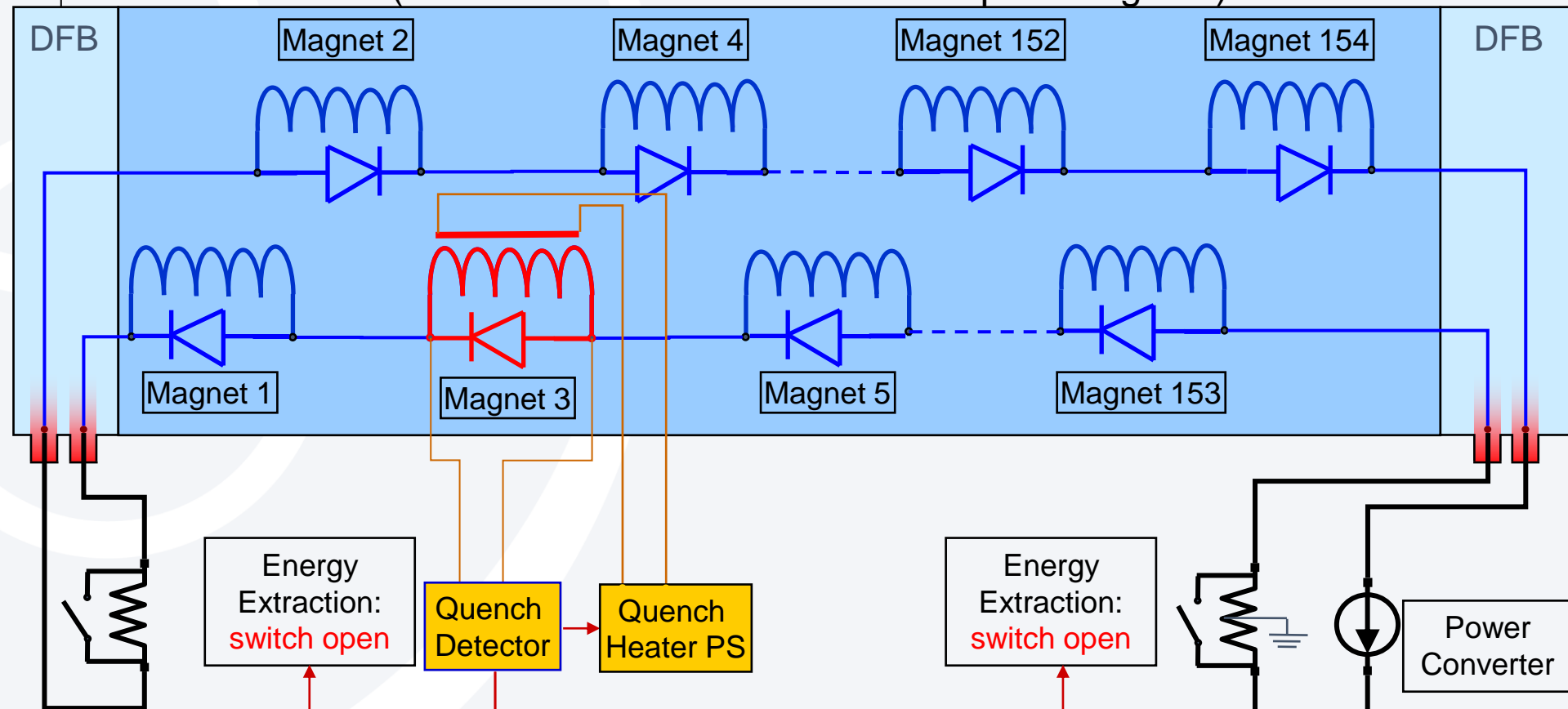
Main dipoles: quench of a magnet

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

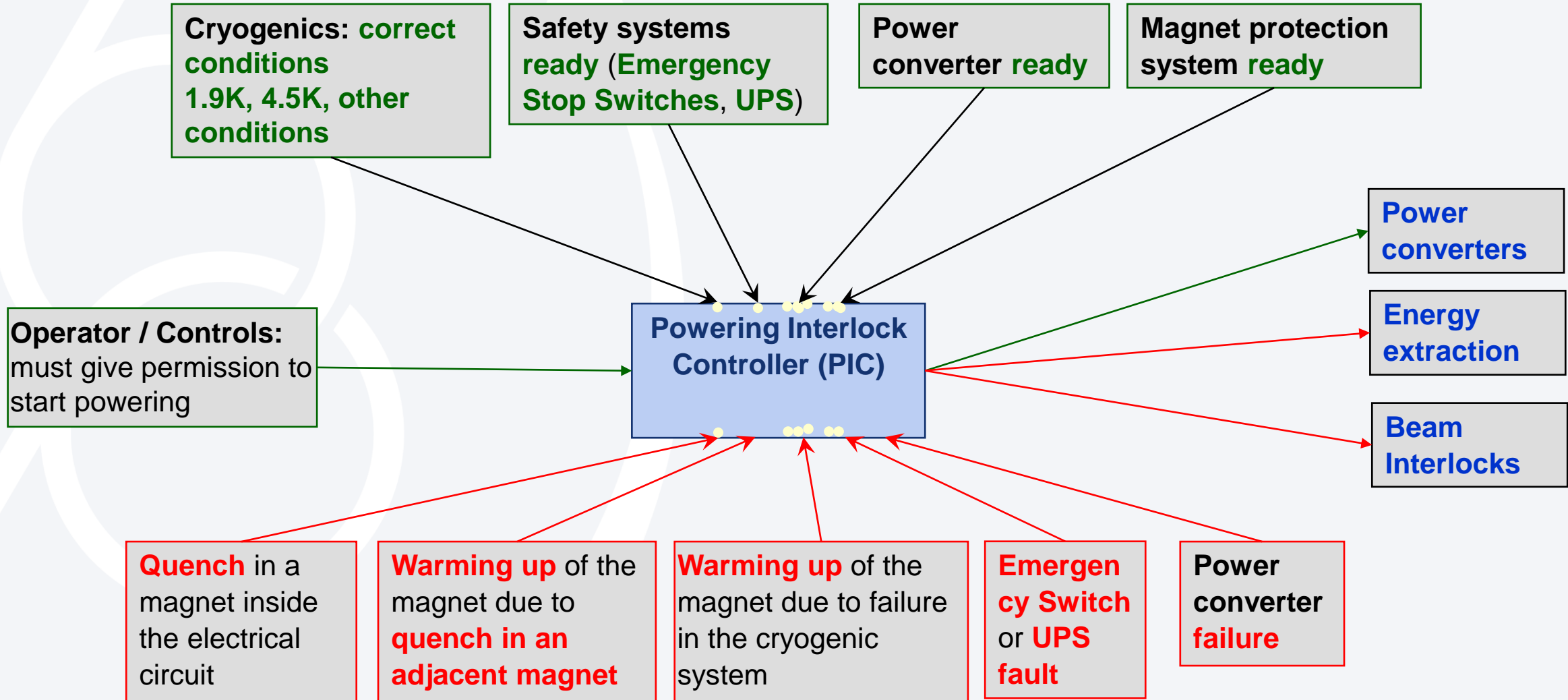


Main dipoles: magnet protection

- **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 ...



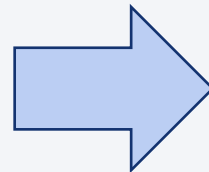
... getting from there to here ...

24th April 2007

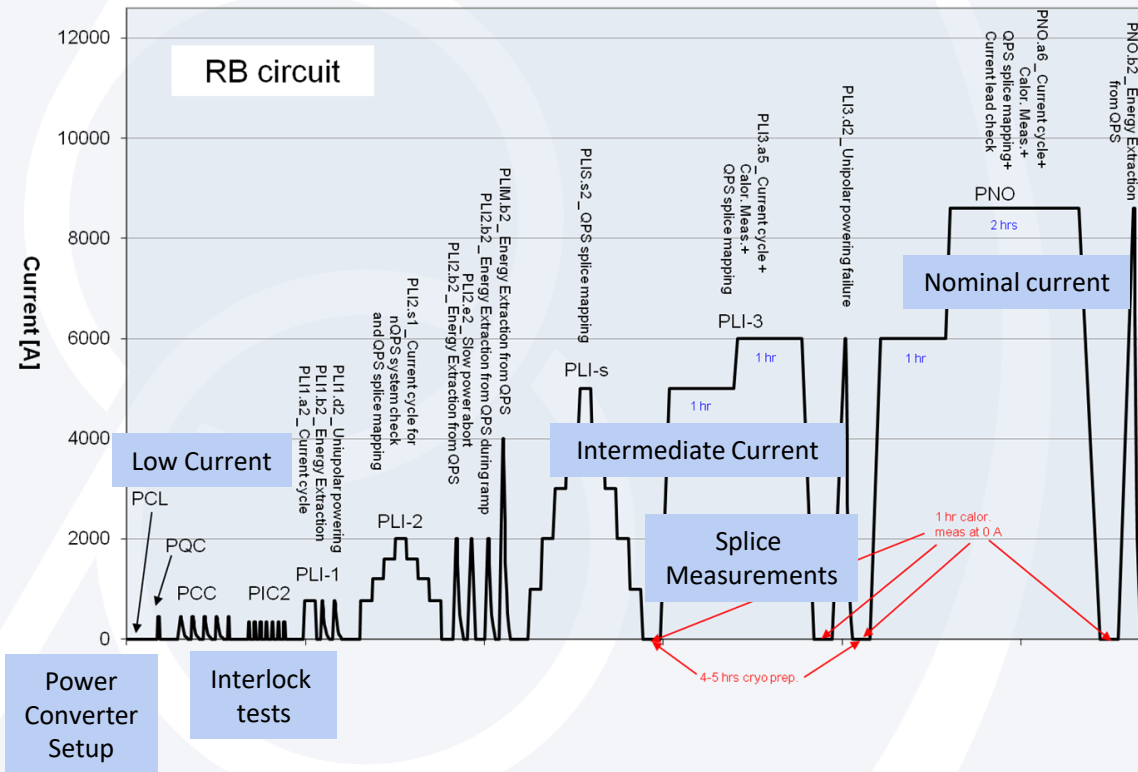
Start of LHC S/C circuits commissioning



- 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

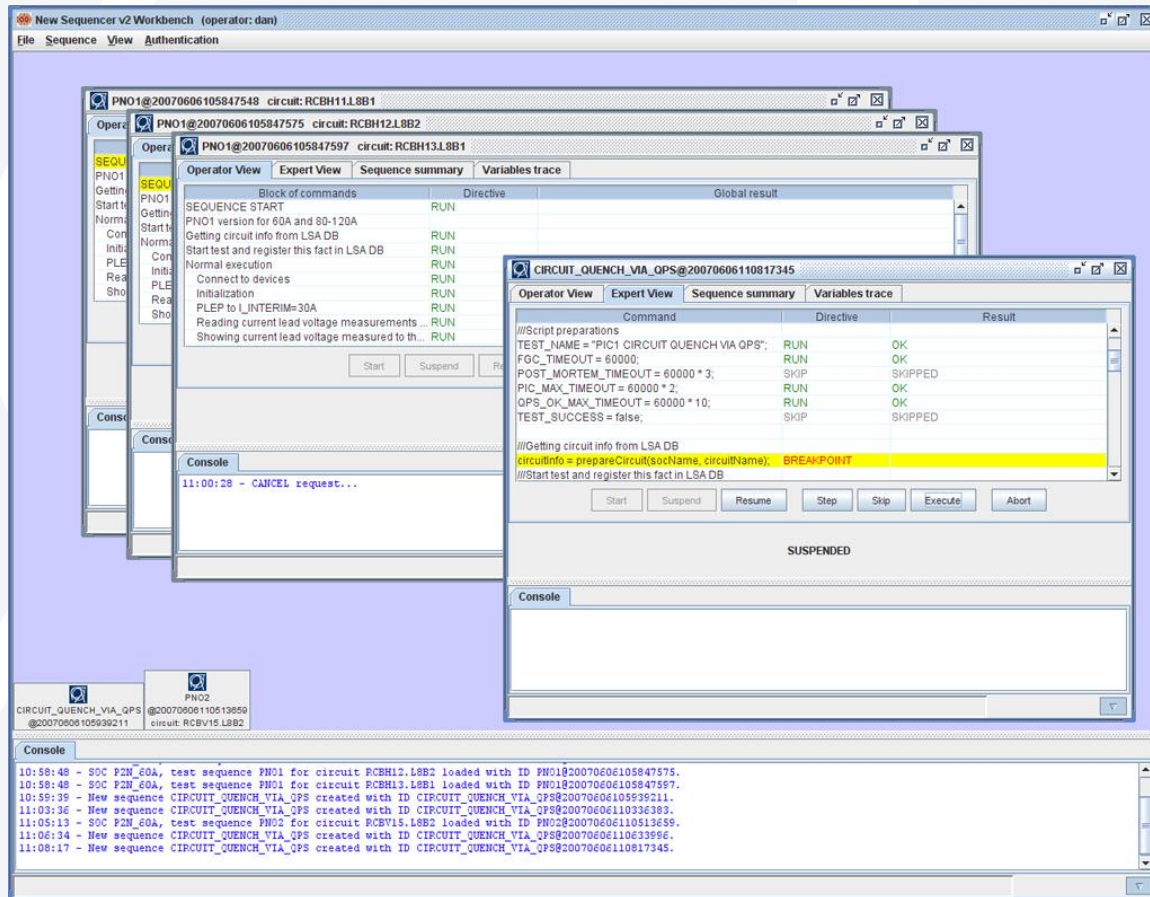


- Textual step by step Guide for particular circuit type + Guide to Analysis
- Contains Parameter for the tests (might change from campaign to campaign), potentially per circuit
- Maintained and updated before every campaign. Still today the ,only source of truth‘

→ Reference / Only source of truth

Example Procedure: 600A Circuit (Courtesy: CERN)

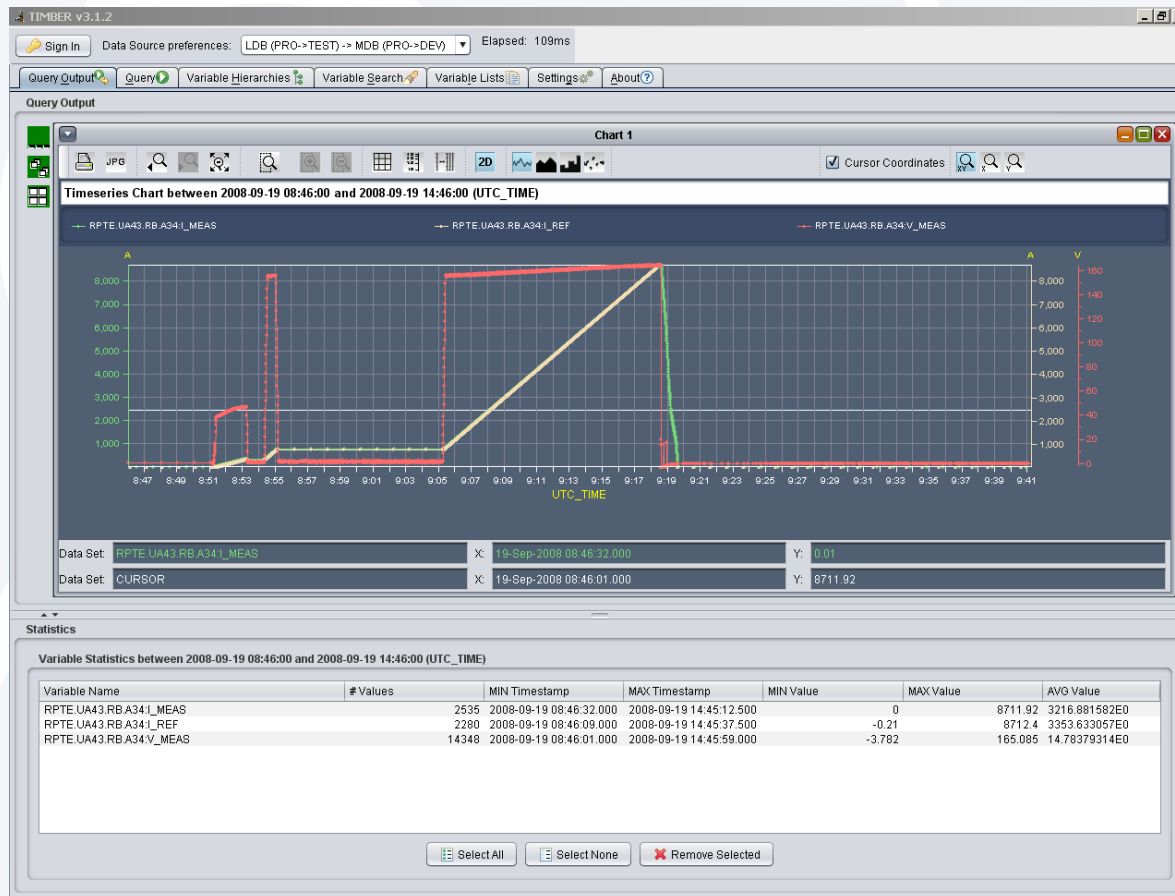
1) Efficient Execution → HWC Sequencer



- 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)

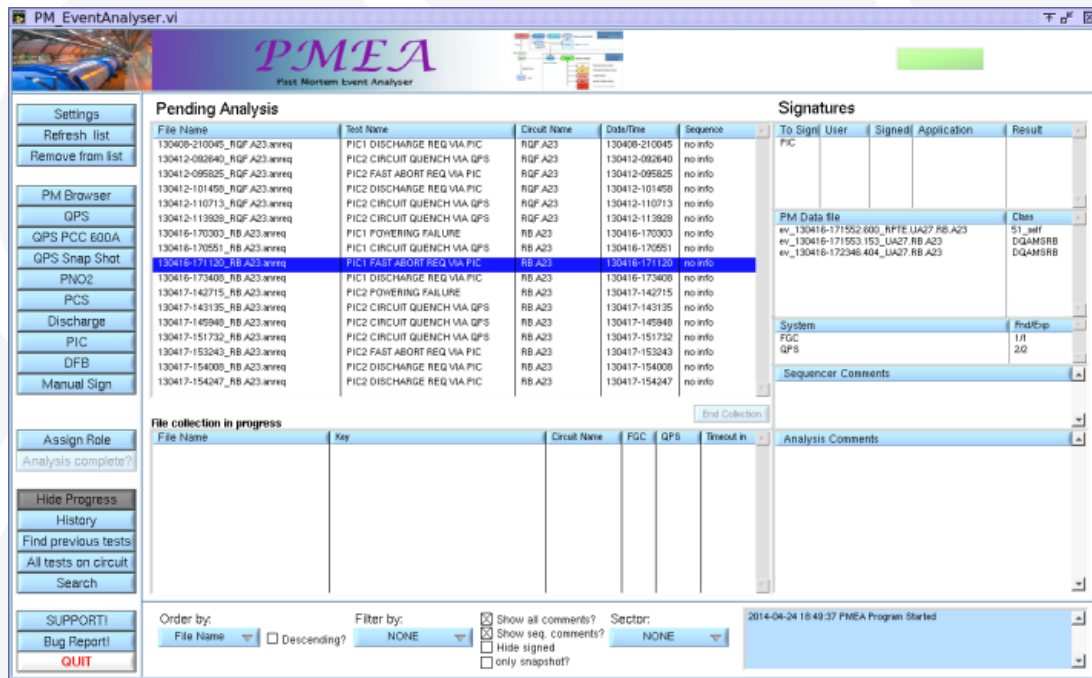
– Logging Database (CALS):

– Logged continuously

– Lower resolution; (potential) data reduction.

3) Test Analysis

- Some ideas in the beginning that sequencer could also do (some) verification → Proved 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

The screenshot shows a web browser window titled 'Powering Test Summary' displaying a table of test results. The table has columns for test ID, status, and various parameters. A pop-up window is overlaid on the table, providing details for a specific circuit.

Test ID	Status	Value	Unit	Other
RD2.R5	PNO.e2	5 / 5 (100%)		Y/WPM, 3
RQ10.L6	PNO.e7	5 / 5 (100%)		Y, 4
RQ10.R5	PNO.e7	5 / 5 (100%)		Y, 3
RQ4.L6	PNO.e7	5 / 5 (100%)		Y, 7
RQ4.R5	PNO.e7	5 / 5 (100%)		Y/WPM, 5
RQ5.L6	PNO.e7	5 / 5 (100%)		Y, 3
RQ5.R5	PNO.e7	5 / 5 (100%)		Y/WPM, 4
RQ6.R5	PNO.e7	5 / 5 (100%)		Y/WPM, 4
RQ7.R5	PNO.e7	5 / 5 (100%)		Y, 3
RQ8.L6	PNO.e7	5 / 5 (100%)		Y, 4
RQ8.R5	PNO.e7	5 / 5 (100%)		Y, 3
RQ9.L6	PNO.e7	5 / 5 (100%)		Y, 4
RQ9.R5	PNO.e7	5 / 5 (100%)		Y, 3
RQX.R5	PNO.d13	5 / 5 (100%)		Y, 1
RB.A56	PLI2.b2	12 / 12 (100%)		N/WPM, 6
RQD.A56	PLI3.b3	6 / 6 (100%)		Y, 2
ROF.A56	PLI3.b3	6 / 6 (100%)		Y, 2

PARAMETER NAME	VALUE	UNIT
ACC_PCS	.1	A/s^2
ACC_PNO	.1	A/s^2
DIDT_PCS	1	A/s
DIDT_PCS_LOW	1	A/s
DIDT_PNO	1	A/s
I_EARTH_MAX	.01	A
I_ERR_MAX	.5	A
I_ERR_PCC_MAX	.36	A
I_INTERM_2	200	A
I_PCC	45	A
I_PCC_MID	10	A
I_PCS	200	A
I_PCS_MID	80	A
I_PNO	550	A
TIME_ACTIVATION	99.493	s

- 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

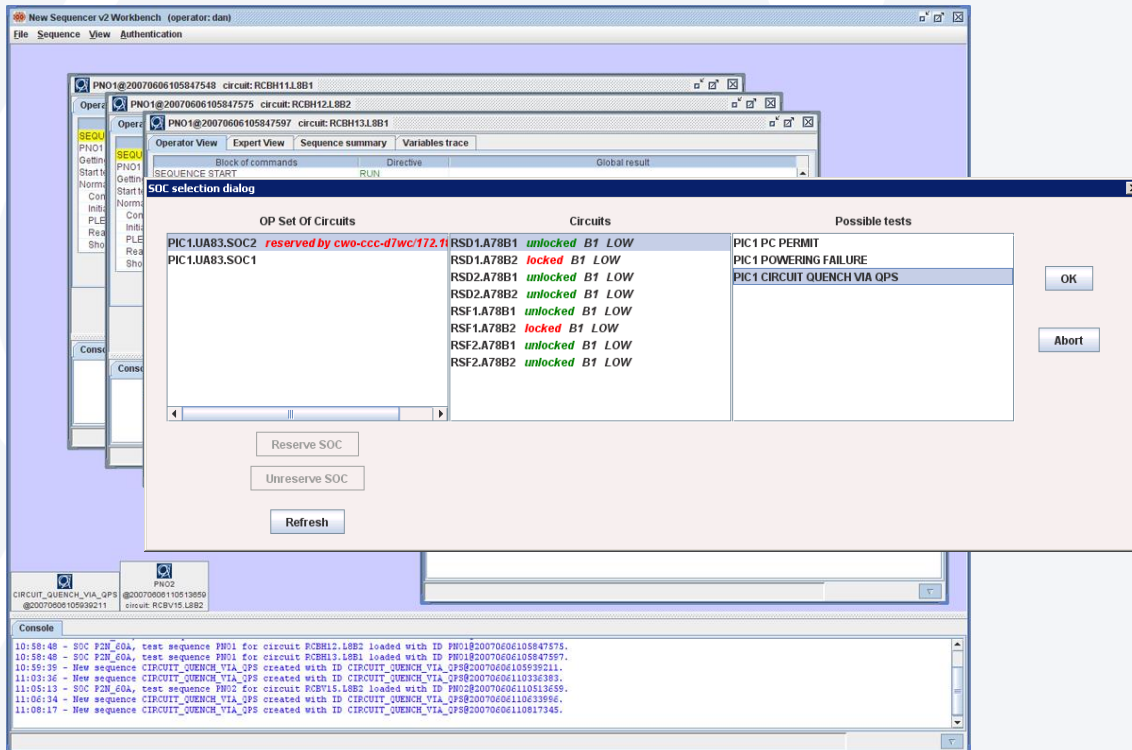
– Parallel Test execution: To avoid that several Operators would access the same hardware:

- Reservation of „Set of Circuits“
- Sequencer „locks“ (db) a circuit during execution.

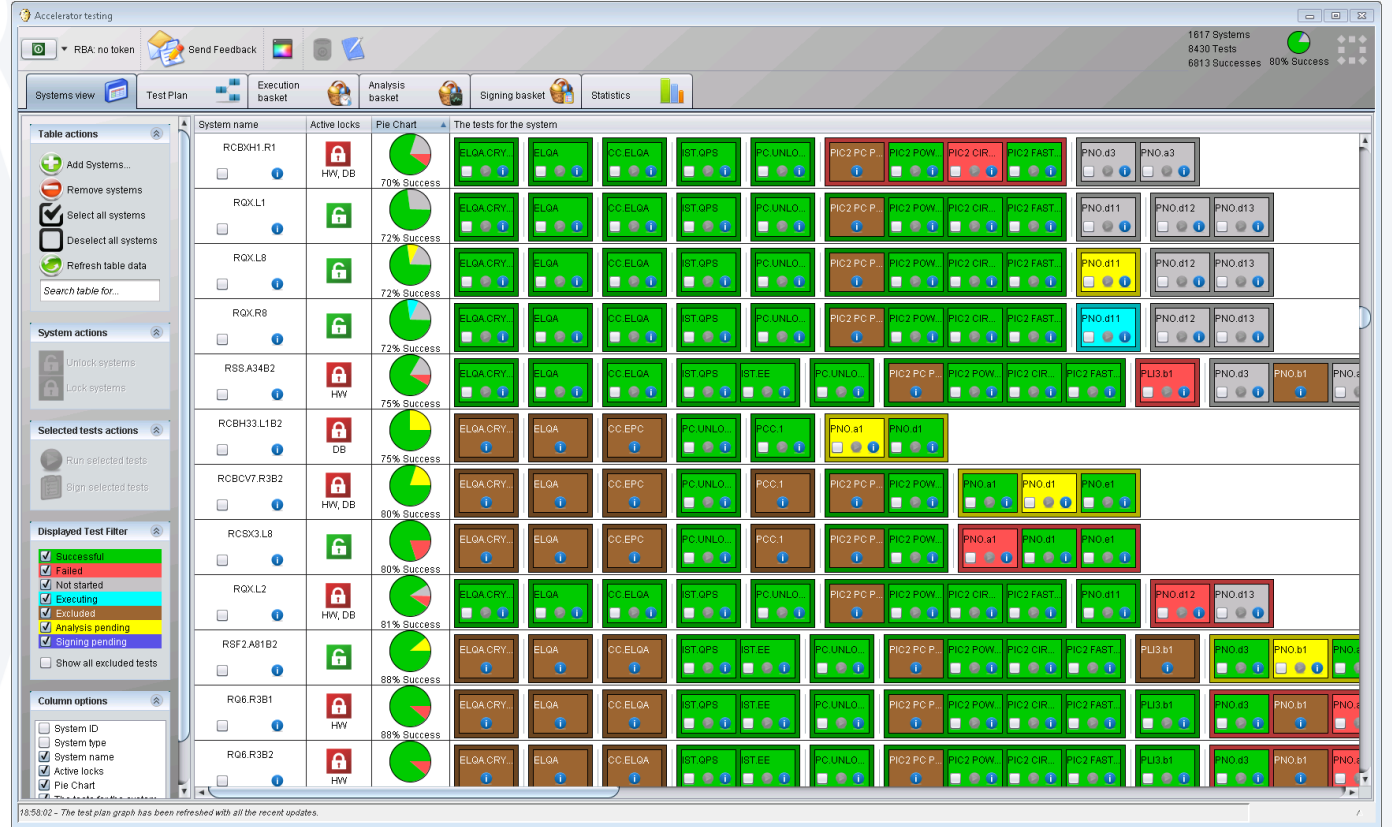
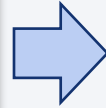
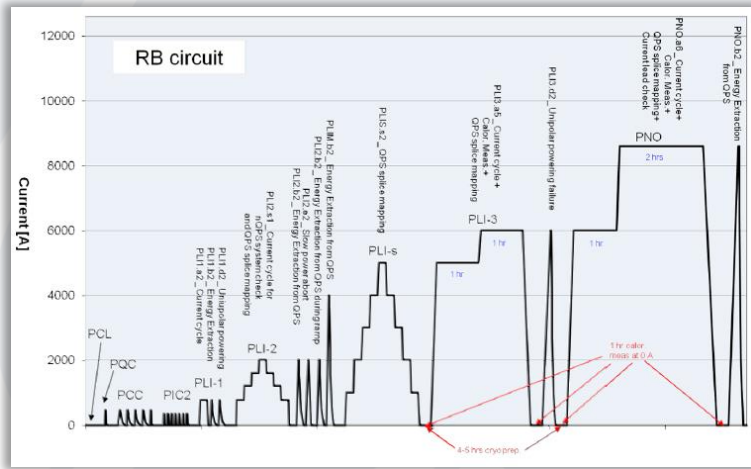
– Still much interference between the tests:

- 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.

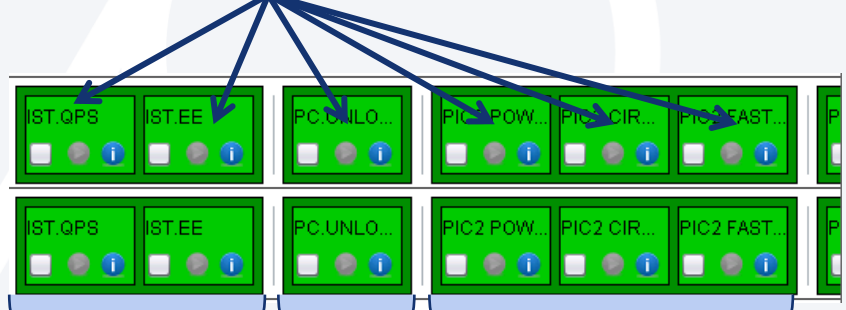
→ A lot of Knowledge required in the head of the operators!



6) Orchestration → AccTesting

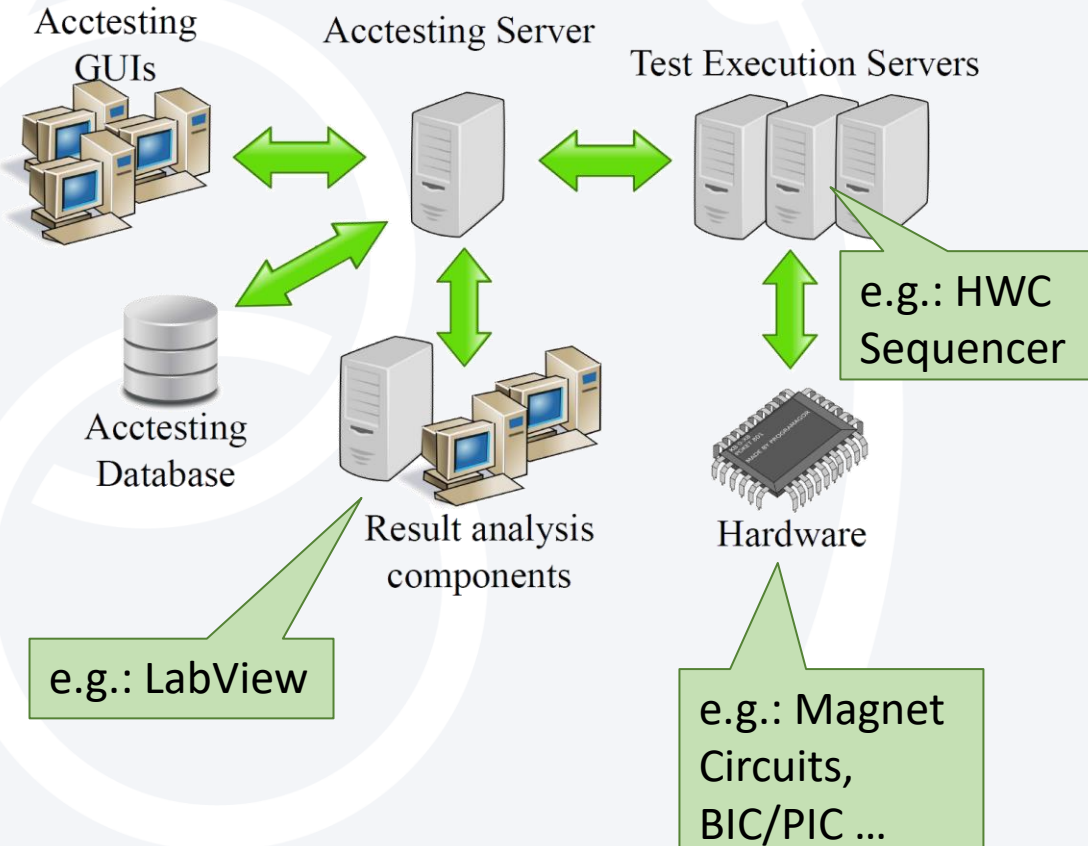


Tests



Test-Phases.

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

The screenshot displays the AccTesting software interface. At the top, it shows system statistics: 1617 Systems, 8430 Tests, 6813 Successes, and 80% Success. The main area is a grid of test results for various systems, including RCBXH1.R1, RQXL1, RQXL8, RQX.R8, R8SA34B2, RCBH33.L1B2, RCBCV7.R3B2, RCSX3L8, RQXL2, RSF2.A81B2, RQ6.R3B1, and RQ6.R3B2. Each system row contains a table of test results with columns for System name, Active locks, Pie Chart, and The tests for the system. The tests are represented by colored buttons (green, yellow, red) indicating their status. A blue callout box is overlaid on the grid with the text: "1. User 'expresses his/her wish' to execute one (or many) tests."

System name	Active locks	Pie Chart	The tests for the system
RCBXH1.R1	HW, DB	70% Success	ELQA CRY... ELQA CC ELQA IST OPS
RQXL1		72% Success	ELQA CRY... ELQA CC ELQA IST OPS
RQXL8		72% Success	ELQA CRY... ELQA CC ELQA IST OPS
RQX.R8		72% Success	ELQA CRY... ELQA CC ELQA IST OPS
R8SA34B2	HW	75% Success	ELQA CRY... ELQA CC ELQA IST OPS IST EE PC UNLO... PIC2 PC P... PIC2 POW... PIC2 CIR... PIC2 FAST... PLI3.b1 PNO.d3 PNO.b1 PNO.e
RCBH33.L1B2	DB	75% Success	ELQA CRY... ELQA CC EPC PC UNLO... PCC.1 PNO.a1 PNO.d1
RCBCV7.R3B2	HW, DB	80% Success	ELQA CRY... ELQA CC EPC PC UNLO... PCC.1 PIC2 PC P... PIC2 POW... PNO.a1 PNO.d1 PNO.e1
RCSX3L8		80% Success	ELQA CRY... ELQA CC EPC PC UNLO... PCC.1 PIC2 PC P... PIC2 POW... PNO.a1 PNO.d1 PNO.e1
RQXL2	HW, DB	81% Success	ELQA CRY... ELQA CC ELQA IST OPS PC UNLO... PIC2 PC P... PIC2 POW... PIC2 CIR... PIC2 FAST... PNO.d11 PNO.d12 PNO.d13
RSF2.A81B2		88% Success	ELQA CRY... ELQA CC ELQA IST OPS IST EE PC UNLO... PIC2 PC P... PIC2 POW... PIC2 CIR... PIC2 FAST... PLI3.b1 PNO.d3 PNO.b1 PNO.e
RQ6.R3B1	HW	88% Success	ELQA CRY... ELQA CC ELQA IST OPS IST EE PC UNLO... PIC2 PC P... PIC2 POW... PIC2 CIR... PIC2 FAST... PLI3.b1 PNO.d3 PNO.b1 PNO.e
RQ6.R3B2	HW	88% Success	ELQA CRY... ELQA CC ELQA IST OPS IST EE PC UNLO... PIC2 PC P... PIC2 POW... PIC2 CIR... PIC2 FAST... PLI3.b1 PNO.d3 PNO.b1 PNO.e

AccTesting - Workflow II

Accelerator testing

957 Systems
12300 Tests
434 Successes 3% Success

Systems view Test Plan Execution basket Analysis basket Signing basket Statistics

Basket filter

Only my systems
 All systems

Search table for...

Basket actions

- Refresh basket
- Abort selected
- Remove all selected
- Remove sel. unscheduled
- Trigger scheduling

System name	Test name	Request status	Request ID	In basket since	Scheduler comment	Requested from
ROD.A56B1	PIC2 CIRCUIT QUENCH VIA QPS	WAITING_FOR_SCHEDULING	68849	84 d	The phase of the test can not yet be executed	cwe-513-vmw175
RQTL9.L3B1	CC.ELQA	WAITING_FOR_SCHEDULING	10081662	91 d	The phase of the test can not yet be executed	cwe-513-vmw175
RCS.A23B2	CC.ELQA	WAITING_FOR_SCHEDULING	10081664	91 d	The phase of the test can not yet be executed	cwe-513-vmw175
RQTL9.L3B2	CC.ELQA	WAITING_FOR_SCHEDULING	10081668	91 d	The phase of the test can not yet be executed	cwe-513-vmw175
RCS.A23B1	CC.ELQA	WAITING_FOR_SCHEDULING	10081673	91 d	The phase of the test can not yet be executed	cwe-513-vmw175
RSS.A23B1	CC.ELQA	WAITING_FOR_SCHEDULING	10081680	91 d	The phase of the test can not yet be executed	cwe-513-vmw175
RSS.A23B2	CC.ELQA	WAITING_FOR_SCHEDULING	10081681	91 d	The phase of the test can not yet be executed	cwe-513-vmw175
RQTD.A23B1	CC.ELQA	WAITING_FOR_SCHEDULING	10081709	91 d	The phase of the test can not yet be executed	cwe-513-vmw175

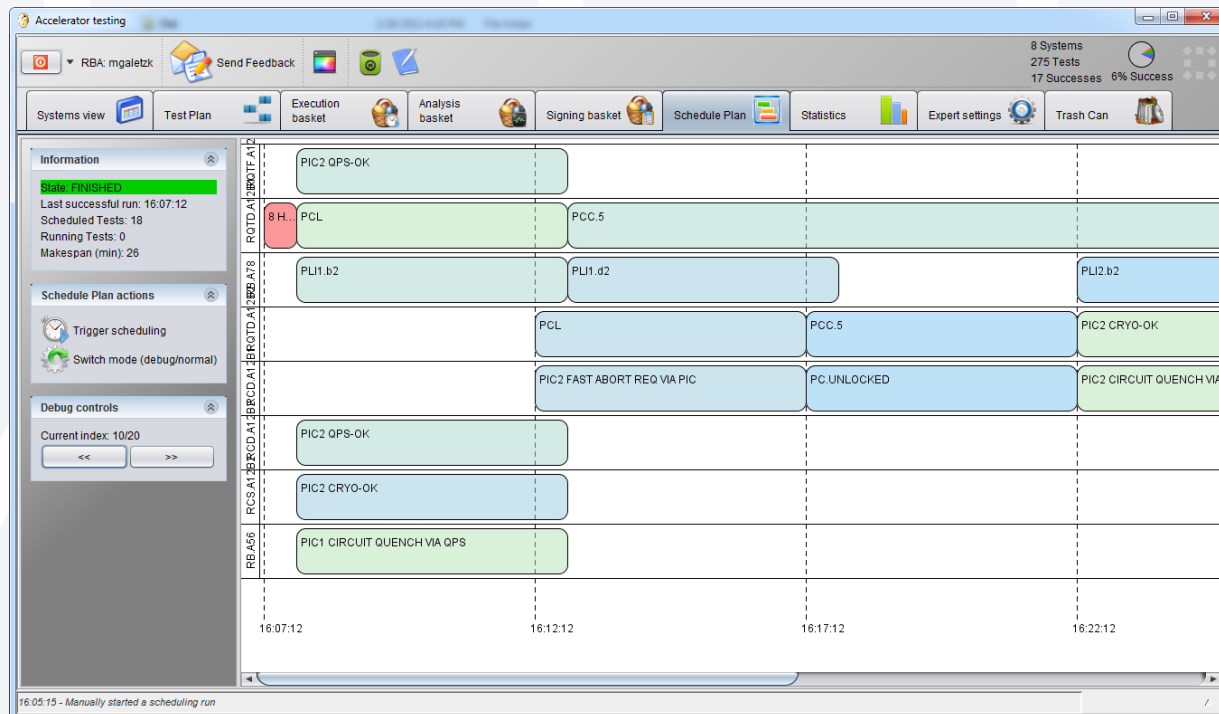
23:11:06 - The test plan graph has been refreshed.

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, grammar, 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 (numerous) small circuits.

– Takes away „monkey analysis“ from the experts and saves their energy for the „tricky cases“

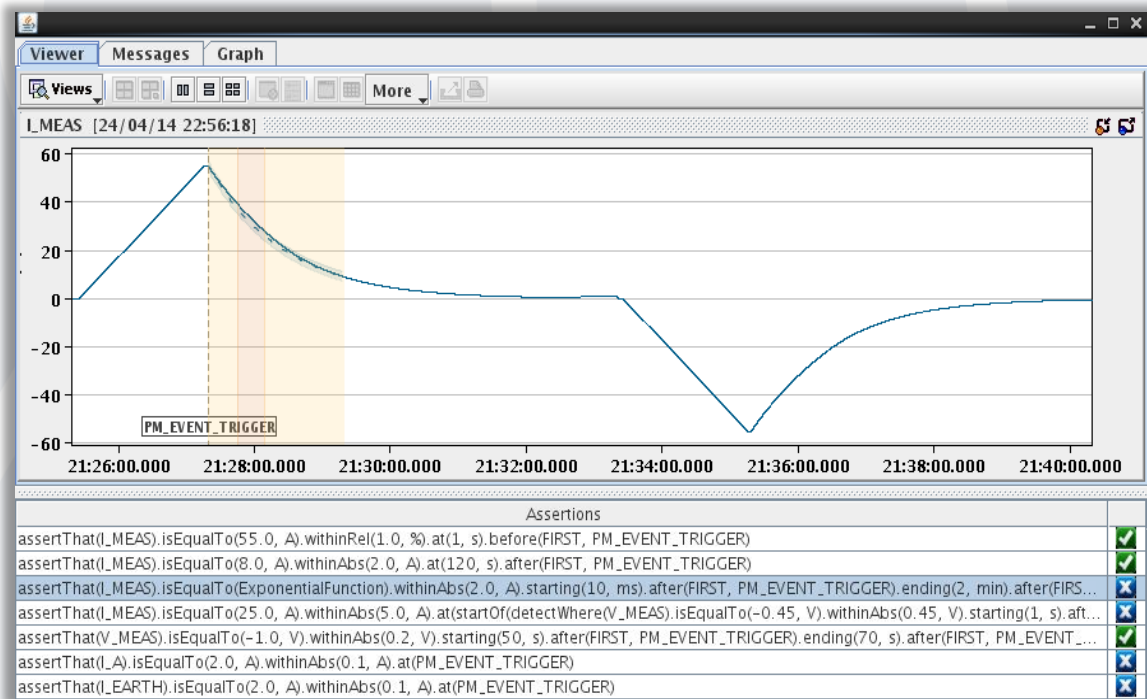
17 // Check ST_CIRCUIT OK OPS and opening of EE switches: ST_SW_A_OPEN in case of 600A with EE system...
18 assertThat(circ_DOEMC_LOCATION_CIRCUIT_ST_FPA_REC_0).changesTo(true).starting(200, MILLI(SECOND))
19 .before(PM_EVENT_TRIGGERER).after(PM_EVENT_TRIGGERER);

Specify the offset to apply to a point in time for the end time of the timestamp range

Parameters:
time the time before or after the upcoming defined point in time
timeUnit the unit for this time

Returns:
the [TimeOffsetDescription](#) allowing to specify the end point in time of the timestamp range.

- ending(long time, Unit<Duration> timeUnit) : TimeOffsetDescription<Void>
- endingAt(PointInTime pointInTime) : Void - TimeRangeEndDescription
- endingAt(Occurrence order, NamedEvent namedEvent) : Void - TimeRange
- equals(Object obj) : boolean - Object
- getClass() : Class<?> - Object
- hashCode() : int - Object



Summary

... and lessons learnt



Summary and Lessons learnt

- Good (not necessary 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!