Accelerator Controls

Stephane Deghaye (BE-CSS-FST)

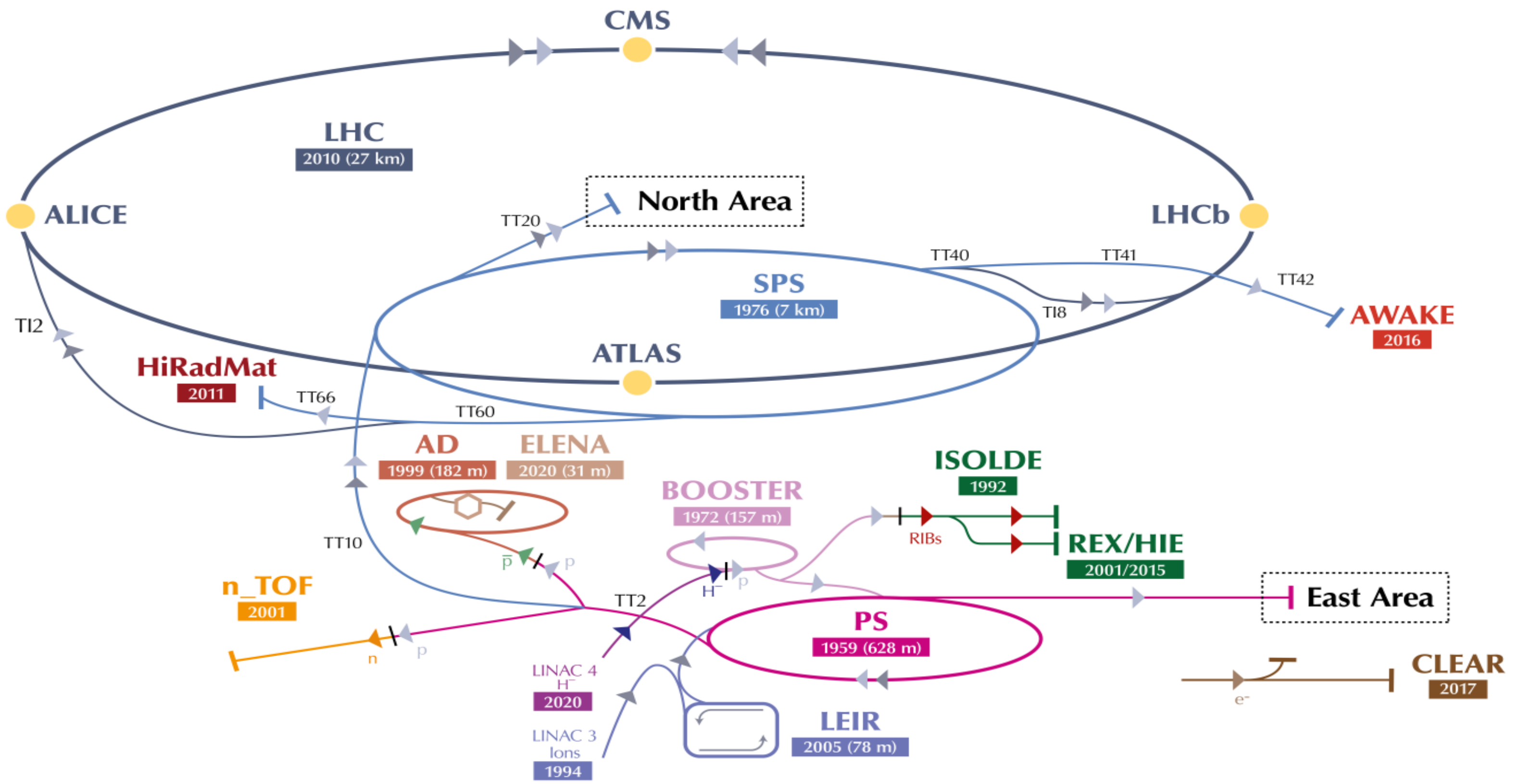
29th & 30th of November 2021

Agenda

1. What is an accelerator made of?
2. Control System Requirements
3. Implementation Philosophy
4. A bit of History
5. Hardware & Software Architecture
6. CERN examples & Key Components
7. Detailed implementations and concepts

CERN Accelerator Complex

CERN Accelerator Complex



H⁻ (hydrogen anions) · p (protons) · ions · RIBs (Radioactive Ion Beams) · n (neutrons) · p̄ (antiprotons) · e⁻ (electrons)

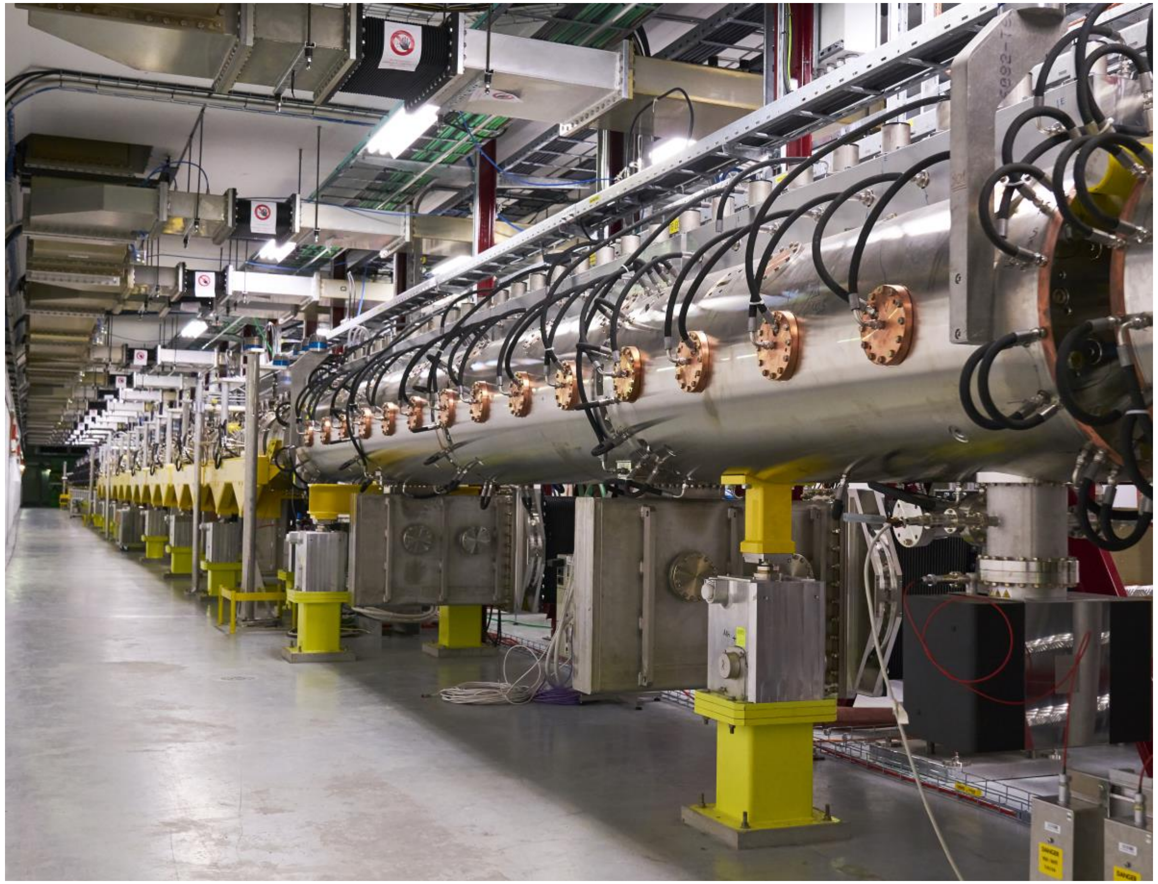
LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE - Radioactive EXperiment/High Intensity and Energy ISOLDE // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator // n_TOF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials

What is an accelerator made of?

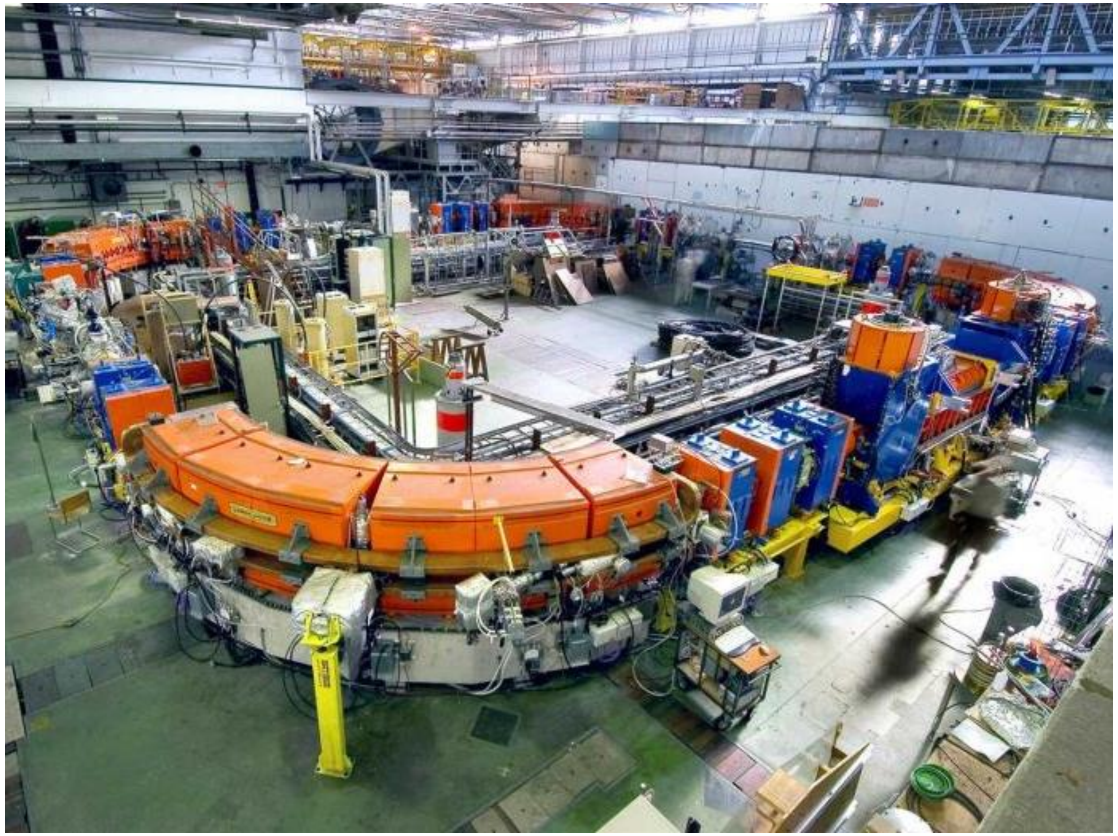
Particle accelerators are either linear or circular

What is an accelerator made of?

Linac4

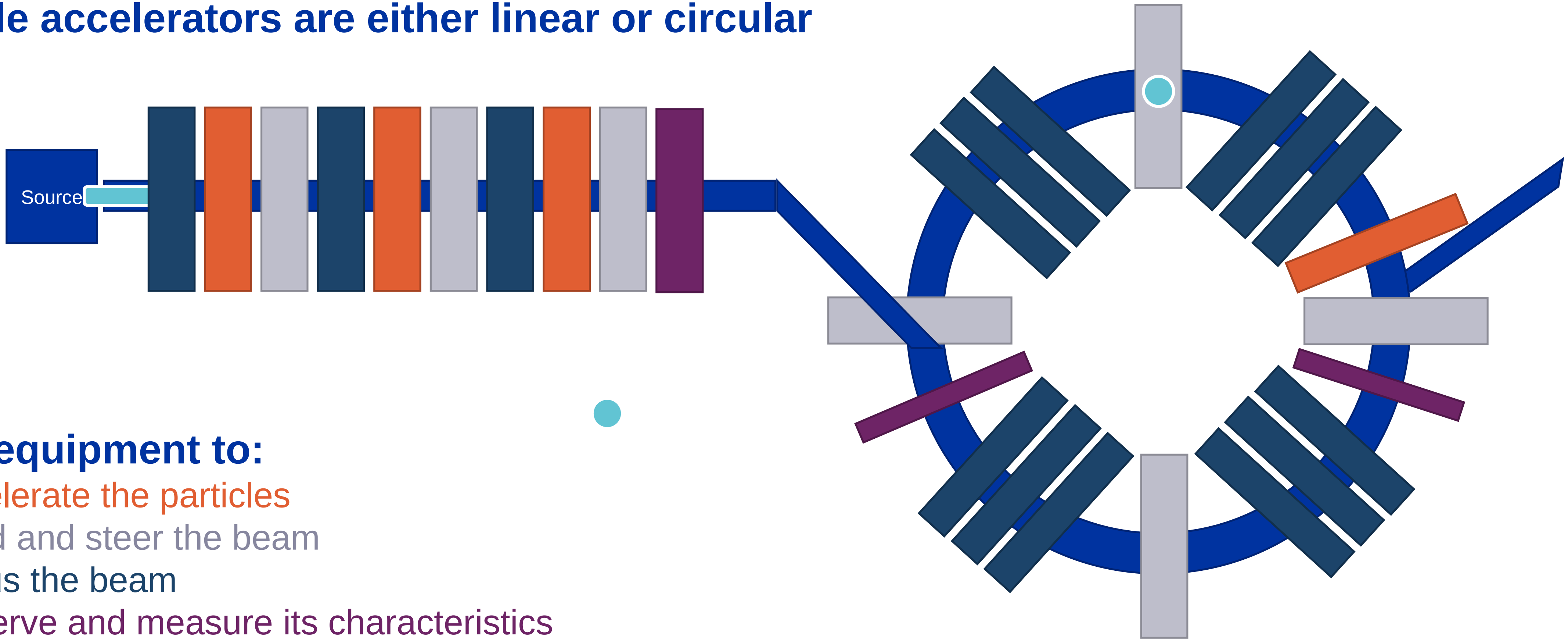


LEIR



What is an accelerator made of?

Particle accelerators are either linear or circular



Need equipment to:

- Accelerate the particles
- Bend and steer the beam
- Focus the beam
- Observe and measure its characteristics
- Inject in and extract from the accelerator

What is an accelerator made of?

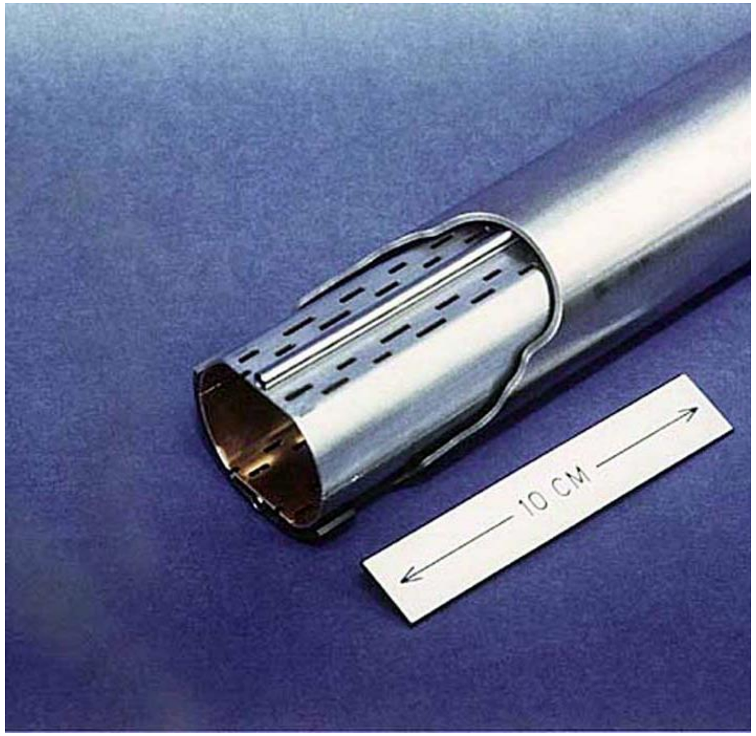
A beam pipe in which the beam circulates

+

Elements in and around the pipe to:

- Steer and focus the beam
- Accelerate the beam and give it structure
- Measure the beam
- Inject and extract the beam
- …

Oversimplified view !!



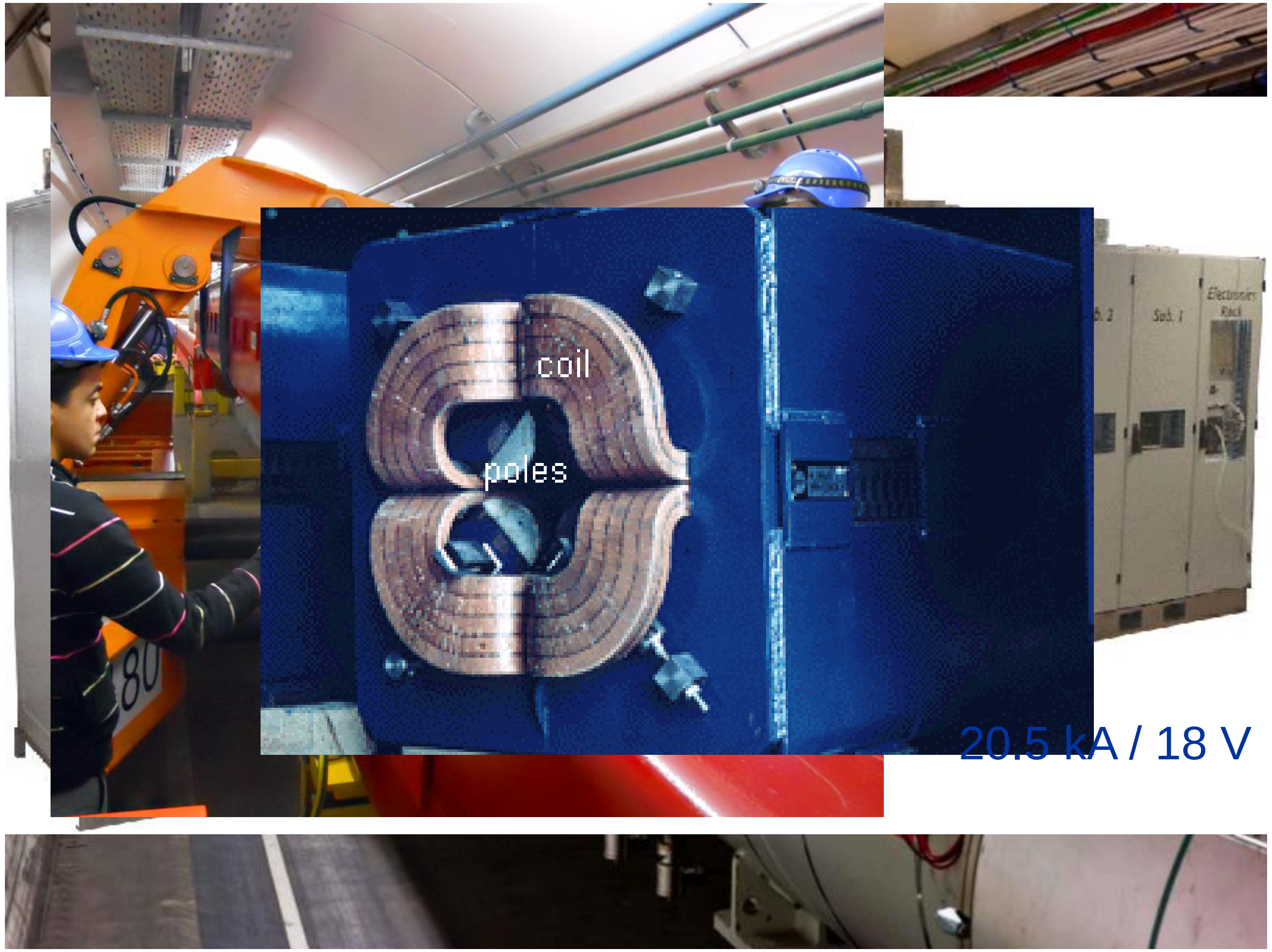
What is an accelerator made of?

A beam pipe in which the beam circulates

+

Elements in and around the pipe to:

- Steer and focus the beam
- Accelerate the beam and give it structure
- Measure the beam
- Inject and extract the beam
- …



Oversimplified view !!

What is an accelerator made of?

A beam pipe in which the beam circulates

+

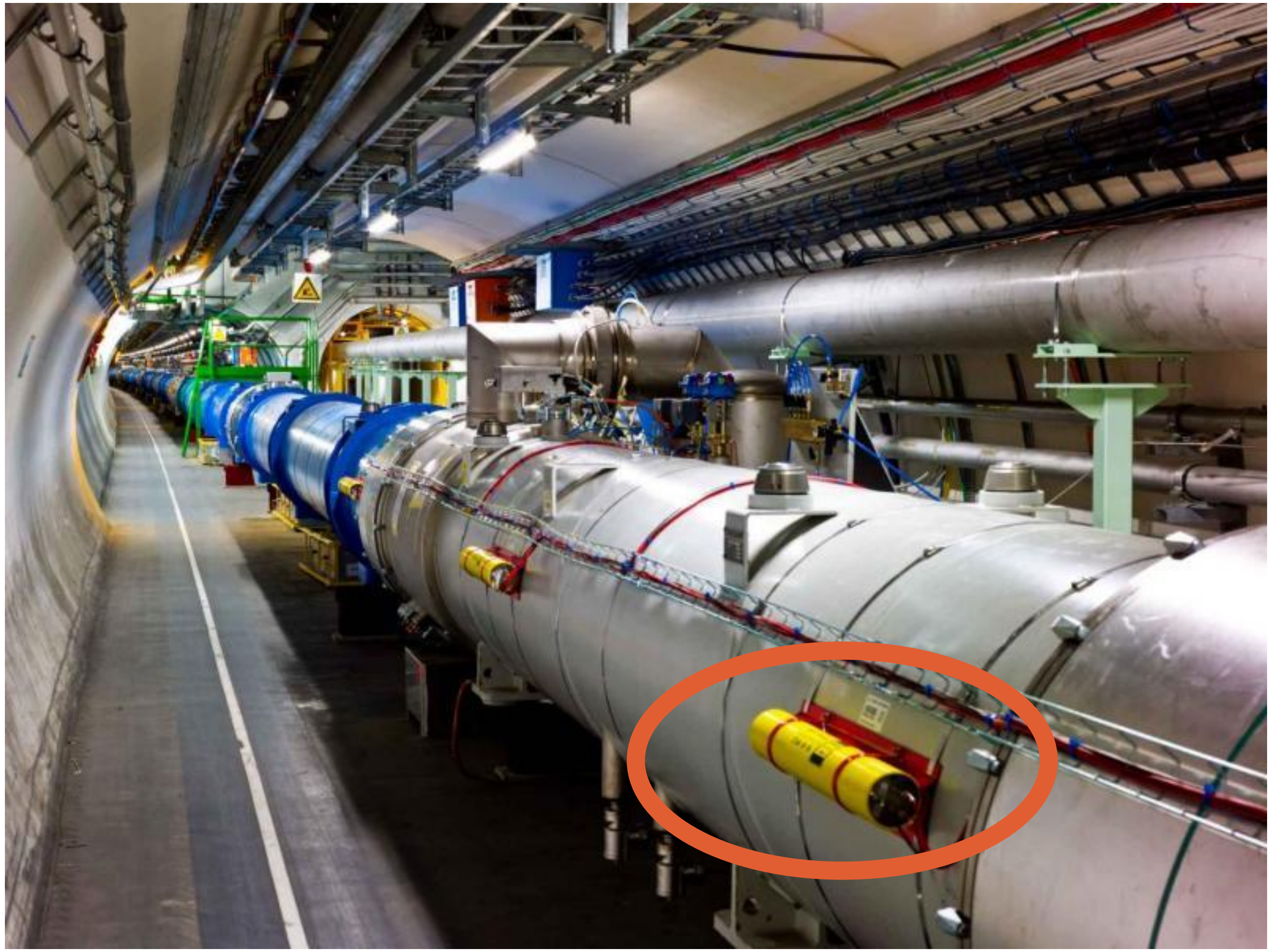
Elements in and around the pipe to:

- Steer and focus the beam
- Accelerate the beam and give it structure
- Measure the beam
- Inject and extract the beam
- …

Oversimplified view !!

What is an accelerator made of?

Beam Position Monitor
Beam Position Monitor



A beam pipe in which the beam circulates

+

Elements in and around the pipe to:

- Steer and focus the beam
- Accelerate the beam and give it structure
- Measure the beam
- Inject and extract the beam
- …

Oversimplified view !!

What is an accelerator made of?

A beam pipe in which the beam circulates

+

Elements in and around the pipe to:

- Steer and focus the beam
- Accelerate the beam and give it structure
- Measure the beam
- Inject and extract the beam
- …

Oversimplified view !!

Transfer line

Septum

Why a Control System?

- Particle accelerators are made of many components to control and monitor the beams produced.

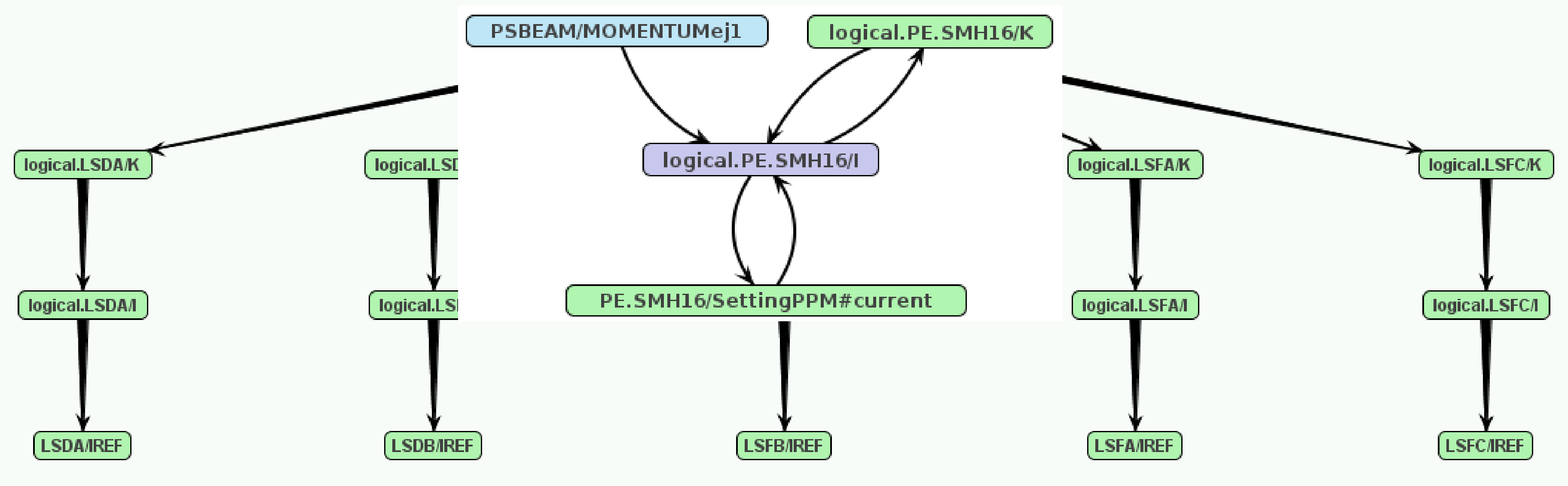
- The physicists and operators need to be able to remotely control and monitor these elements ➔ this is the role of the **Control System**.

Control System Requirements

Control System Key Requirements

Act on accelerator elements (settings & states)

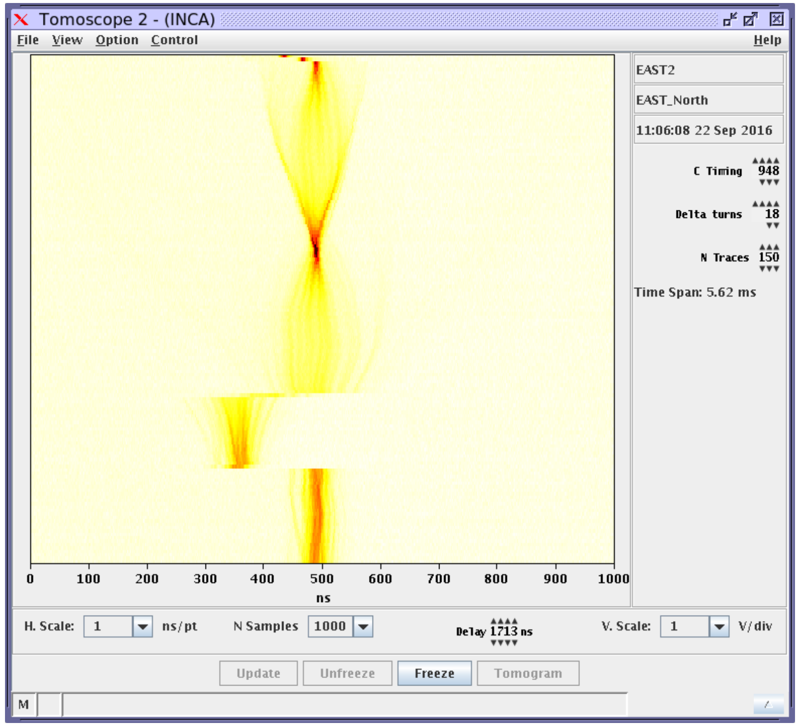
- Minimum: direct access to the hardware values
- Ideal:
  - Model-driven control to work at a higher level
  - Global transactional synchronisation



Control System Key Requirements

Monitor the elements (instruments & actuators)

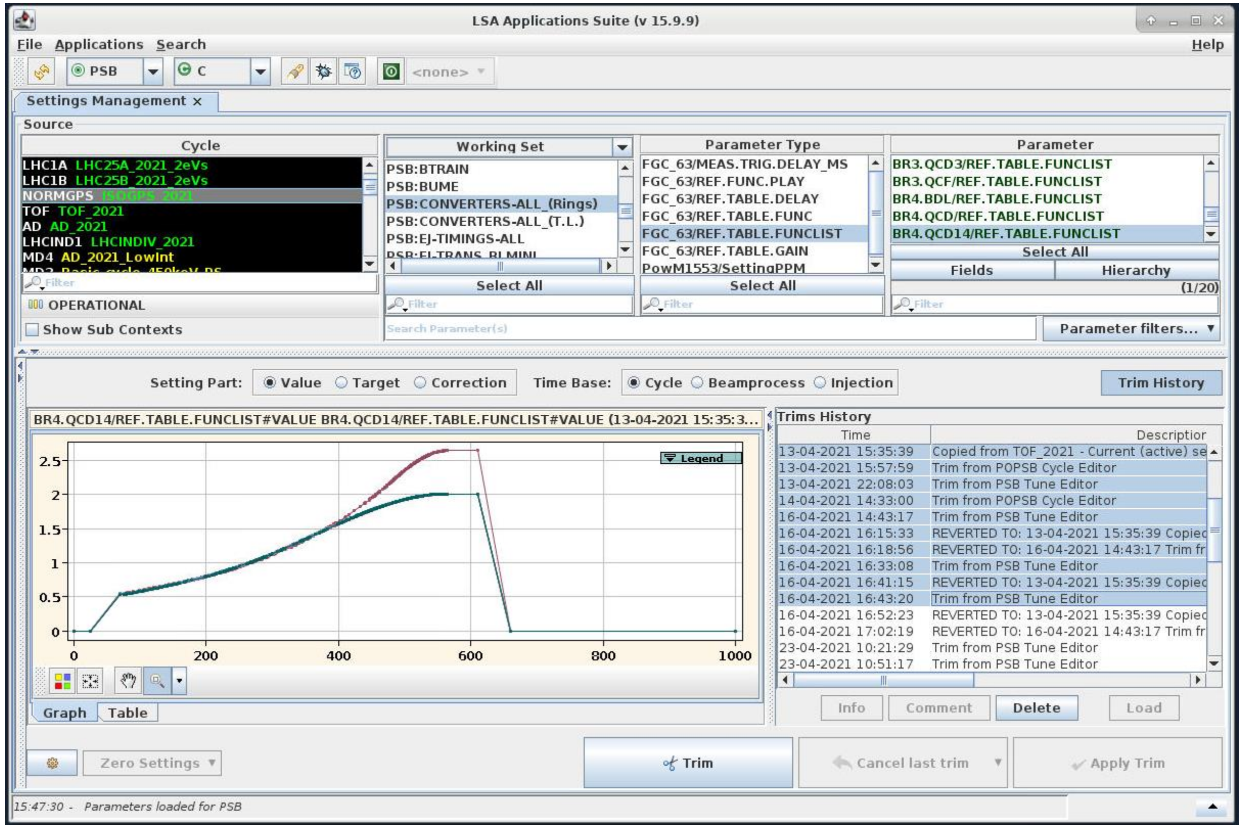
- Minimum: Display raw acquisitions
- Ideal: Time-tagged, coherent acquisitions, post-processing for quick detection of abnormal situations



Control System Key Requirements

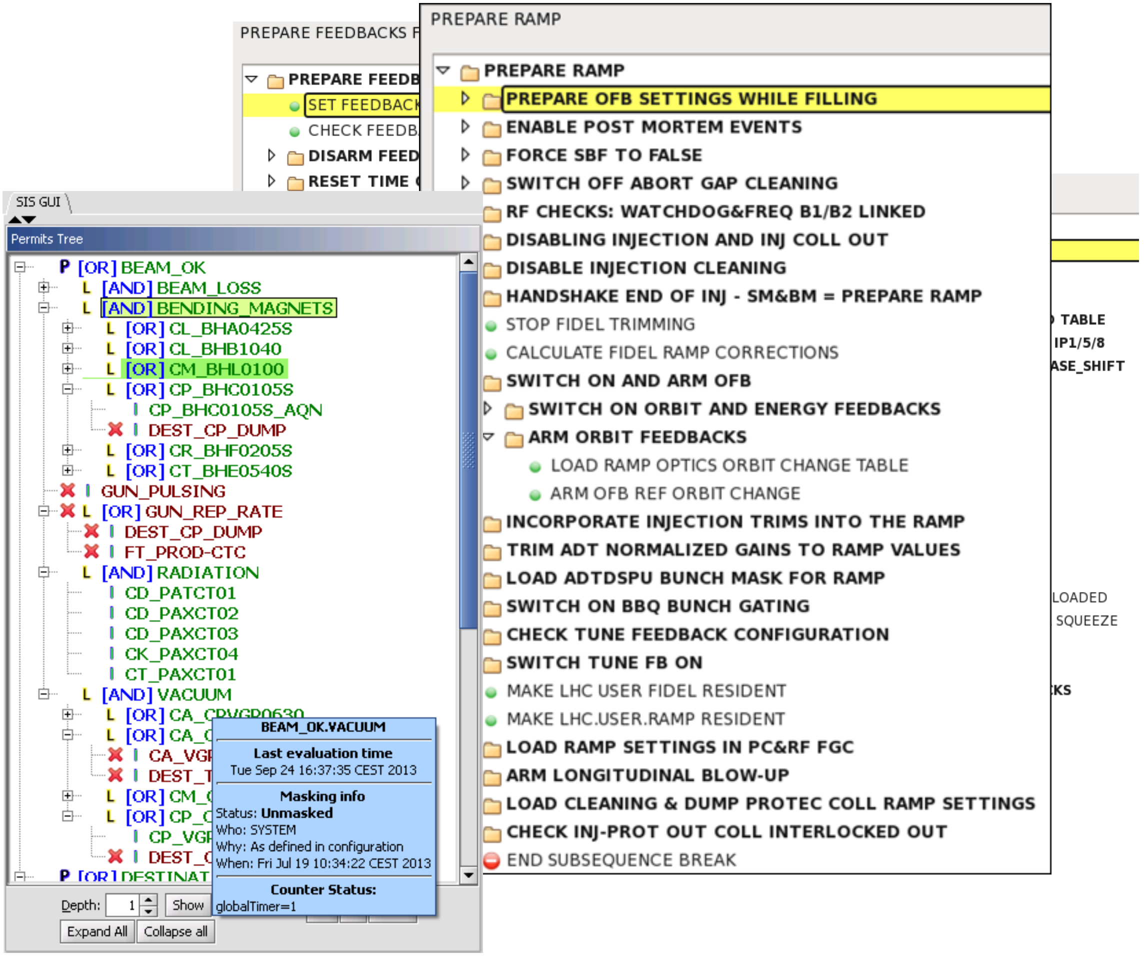
Long-Term Memory of Settings & Acquisitions

- AKA Logging
- Accelerator performance & post-mortem analysis, fine tuning of the machines
- Minimum: structured time-series in a simple format (CSV, SDDS, etc.)
- Ideal: Years of data (settings & acquisitions) with performant data extraction & analysis tools



More Control System Requirements

- **Automation**
  - **Generate initial values, play sequences, feedback loops, etc.**
  - **Minimum: Non-interactive scripts**
  - **Ideal: Model-driven generation, flexible sequencer (almost like a debugger), automated actions (decision tree, machine learning)**

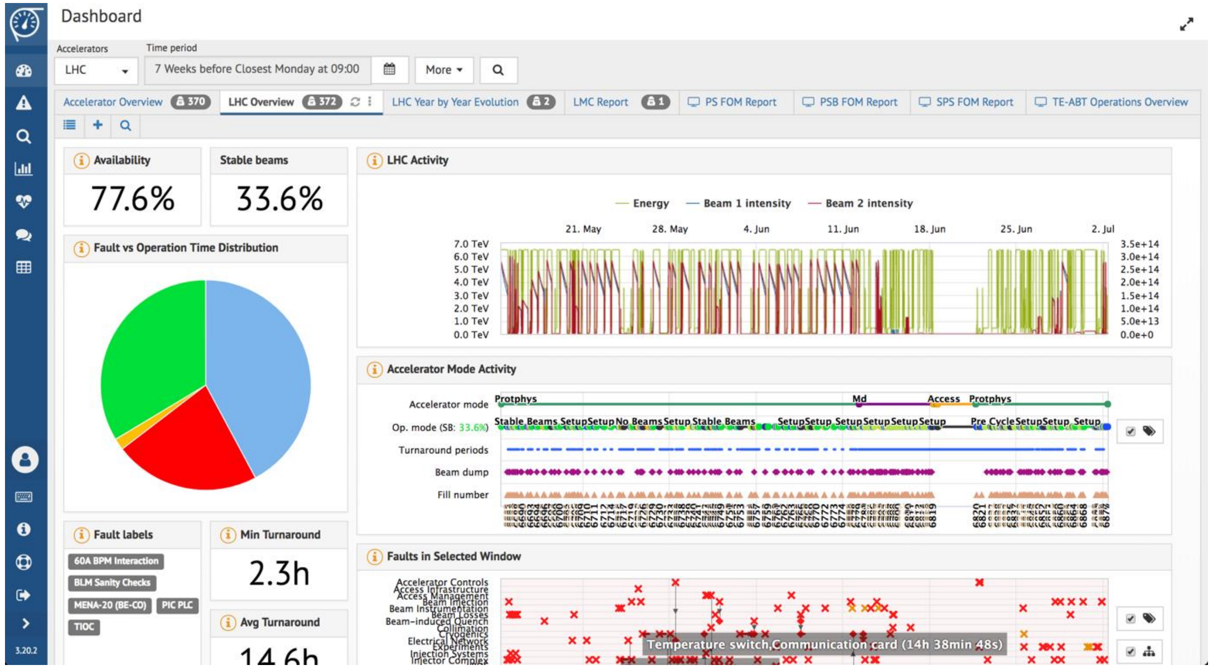


More Control System Requirements

- **Safety for machine protection & operational availability**
  - **Minimum: Machine interlock to protect the hardware**
  - **Ideal: High-level fast-reaction interlocks and role-based access to prevent the wrong action at the wrong time**

More Control System Requirements

- **Diagnostics**
  - **Detection, identification, and follow-up of problems in the controls infrastructure**
  - **Minimum: Non-interactive status screens**
  - **Ideal: Online monitoring, remote interventions (e.g. power cycle), failure prediction (Machine Learning), analysis tools**



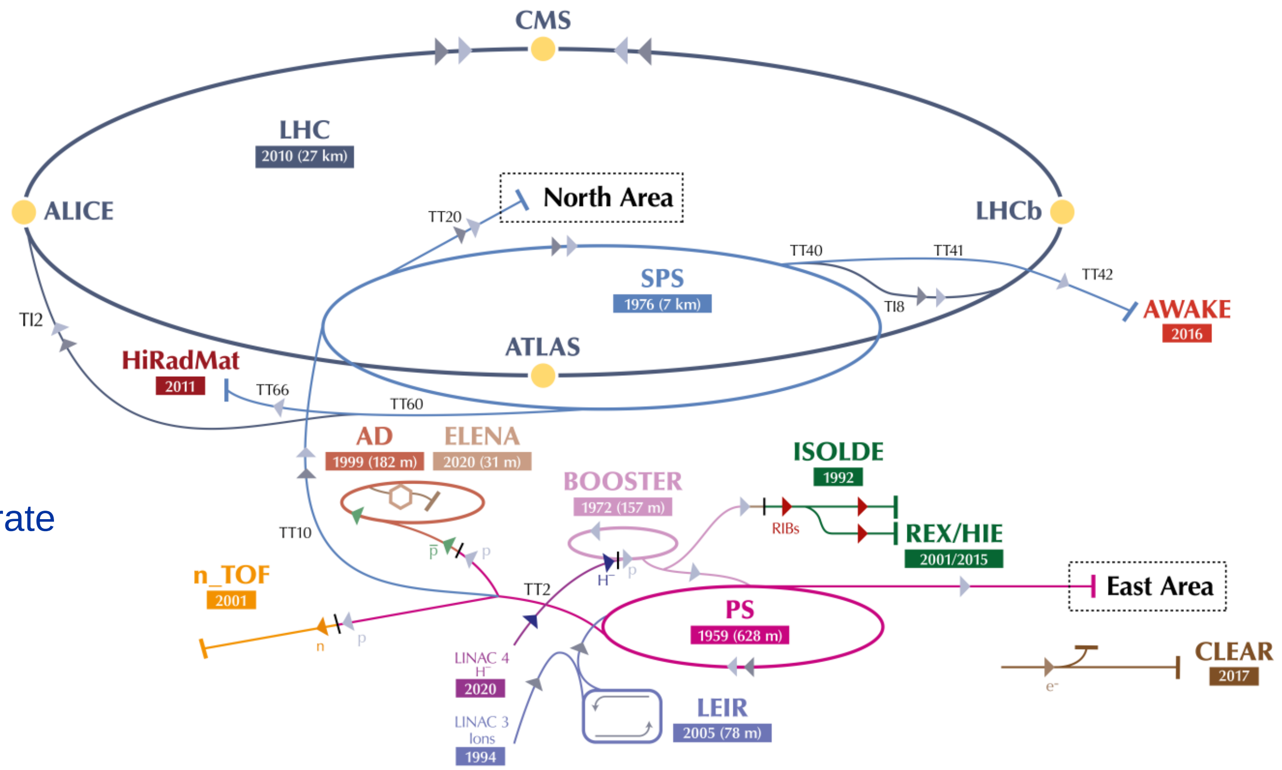
And many more…

Controls Complexity

- **Many requirements from physicists and operators**
- **Accelerators made of many elements**
  - Early accelerators, e.g. Proton Synchrotron (PS), were small

    < 5’000 devices
  - Latest accelerators, e.g. LHC, are much more complex to operate

    30’000+ elements



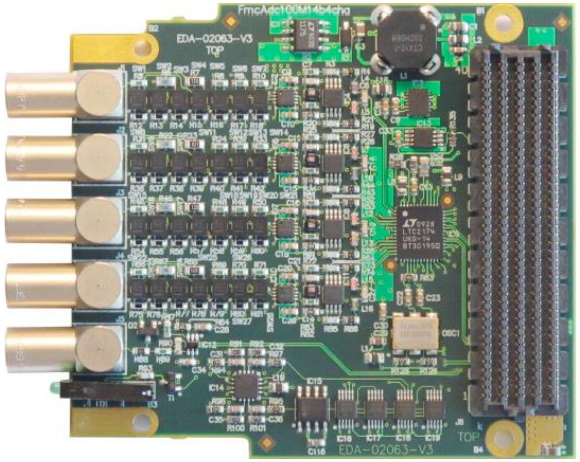
- **The Control System’s job is to hide the complexity and help you to do your job as efficiently as possible.**

Implementation Philosophy

Implementation Philosophy - Hardware

As much as possible:

- **Apply vertical industrial control system solutions**
  - PLCs for industry-like process control (electricity, cooling & ventilation, vacuum, cryo)
- **Restrict home-made HW development to specific applications**
  - Beam optics controls (i.e. all power converters),
  - Injection and Extraction systems,
  - Beam instrumentation,
  - RF,
  - Collimation,
  - Timing Systems,
  - Etc.



Don't Reinvent The Wheel !!

- **Base the HW architecture on available standards and Commercial Off-The-Shelf (COTS)**
  - Standards for complex embedded I/O systems with high performance demands
  - COTS electronic modules for generic features (CPUs, serial controller boards, ADCs, etc.)
  - COTS desktop PCs & servers for control rooms and application servers
  - Standard fieldbuses for applications requiring real-time features and radiation hardness (e.g. WorldFIP), and less stringent applications (Profinet/Ethercat)
  - Standards for cost-effective I/O systems for networking (fieldbus controllers)
  - GPS for time stamping and overall accelerator synchronization

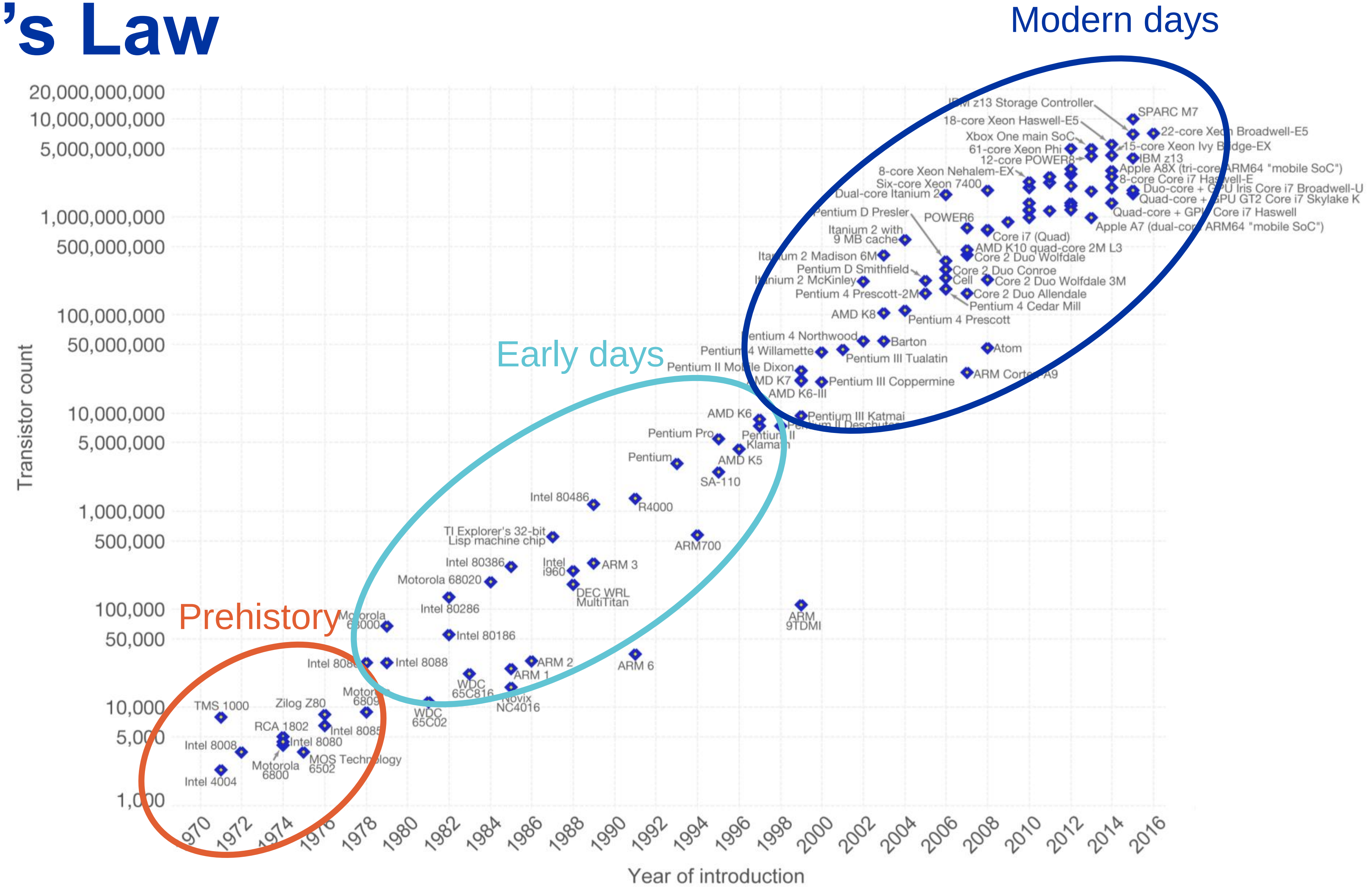
Implementation Philosophy - Software

As much as possible:

- **Apply vertical industrial control system solutions**
  - Supervisory Control and Data Acquisition Systems (SCADA) for commands, graphical user interfaces, alarms, etc. of industrial systems
- **Rely on common technologies and tools**
  - Important for aspects such as recruitment, education & training
  - DBs & Storage solutions (e.g. Hadoop)
  - Communication protocols
  - Monitoring solutions used in the industry (e.g. ICINGA)
- **Privilege Open-Source Software**
  - Avoid vendor lock-in
  - Control license cost
  - Manage the Total Cost of Ownership (TCO)

A bit of History

Moore's Law

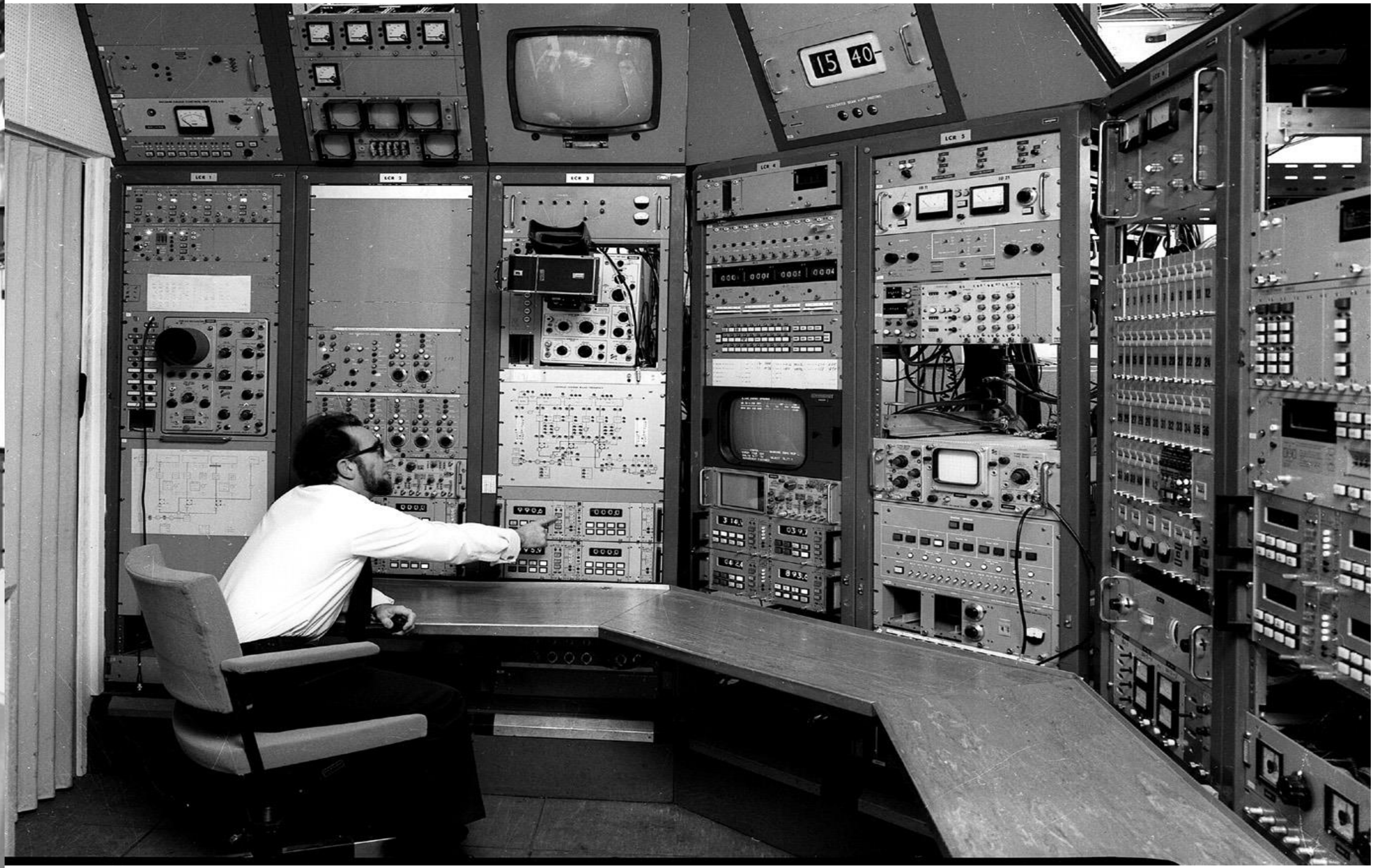


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Control System Prehistory

- **Accelerators are small and overall less complex (e.g. no superconducting magnets)**
  - No more than a few thousands of devices to control
- **No computing infrastructure and limited possibility to model**
- **Actuator and monitors are physically in the local control rooms (e.g. buttons, knobs, analogue oscilloscopes, etc.)**

Control System Prehistory



The Early Days

Beginning of remote controls

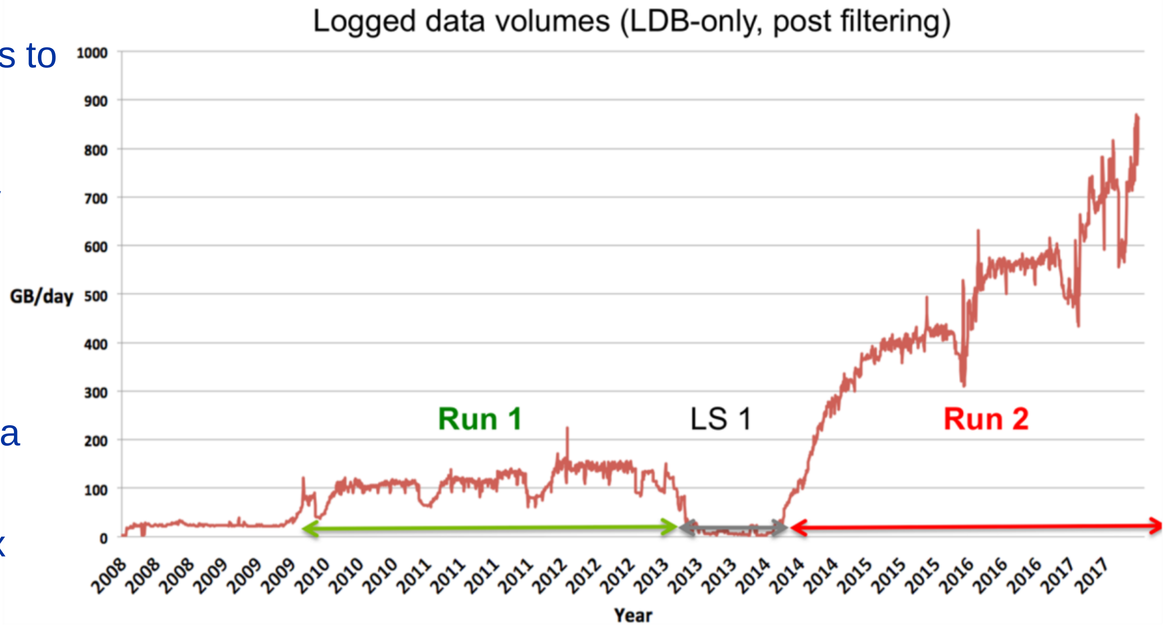
- Still limited by the available performance
- Lack of standards and common frameworks → more DIY and custom solutions
- Emergence of several controls solutions, aiming at different types of accelerators (at first)
  - EPICS (driven by US labs),
  - Tango (driven by ESRF (Fr) – synchrotron light sources)
  - CERN[1]

[1] non-exhaustive list

Modern days

- **Hardware has become powerful**
  - E.g. embedded systems at CERN in late 90s had **64 MB of RAM**; Nowadays, they have **8 GB**
  - Most of the needs are covered
  - Yet, users want more and more data (turn-by-turn acquisitions, big-data solution for the long-term storage, etc.)

- **Software industry has become a major actor worldwide.**
  - We can rely on many readily available technologies that open the doors to much more powerful systems



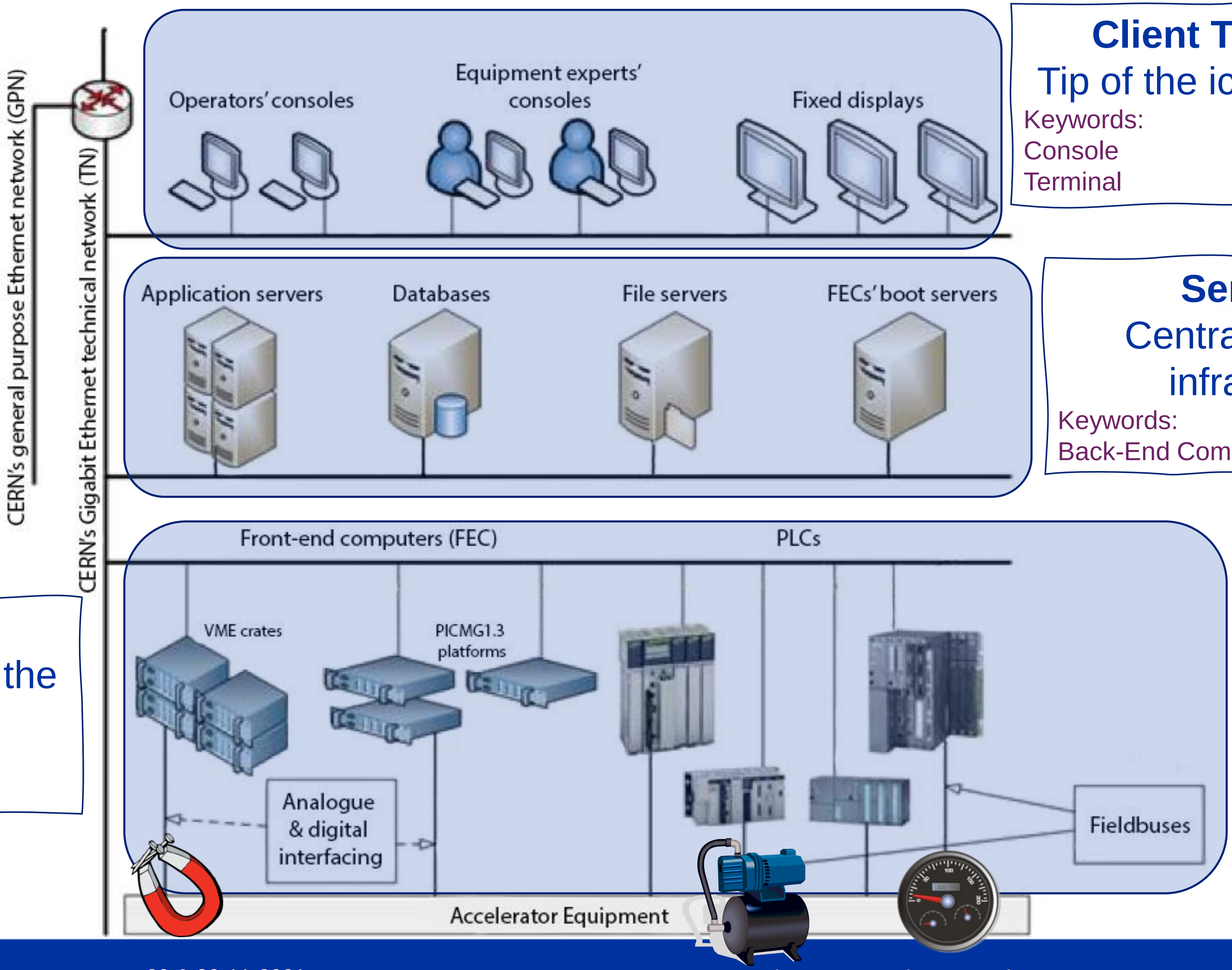
- **We still need to integrate and customise them to the very specific domain of particle accelerator controls**
  - Not all solutions are appropriate; Need to remember accelerator controls ≠ selling plane tickets
  - Mastering the different solutions with their evolution, limitations, etc. is a major challenge
  - The rhythm of updates is no longer under our control. E.g. recent Linux CentOS changes

50 years of technology evolution

High-Level Architecture

of a modern control system

High-Level Hardware Architecture



Client Tier
Tip of the iceberg
Keywords:
Console
Terminal

Server Tier
Central computing infrastructure
Keywords:
Back-End Computer (BEC)

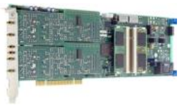
Resource Tier
Electronics close to the accelerator
Keywords:
Front-End Computer (FEC)

High-Level Hardware Architecture

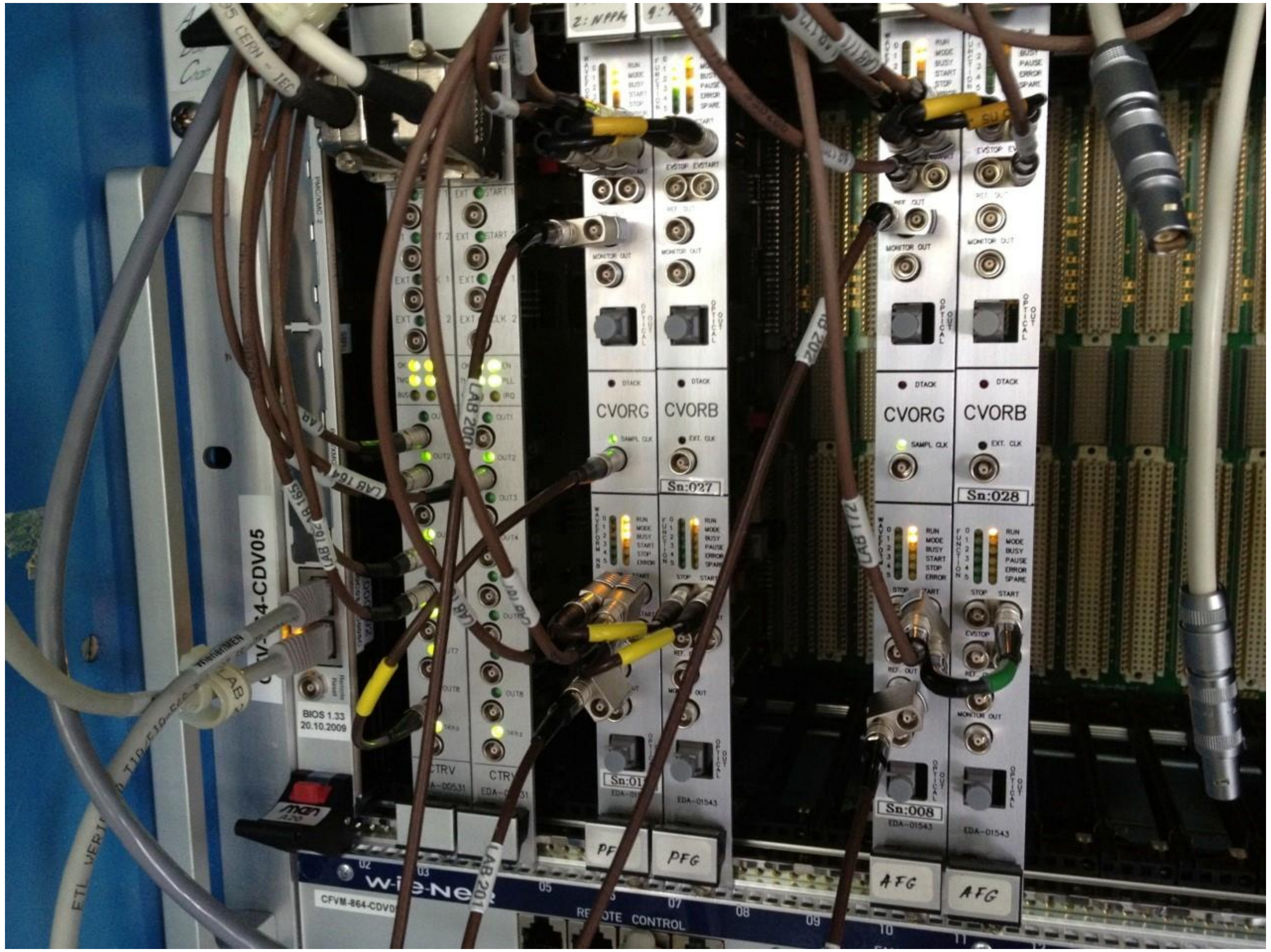
Resource Tier

- **Open enclosures**
  - Easy access
  - Better cooling and power available
  - But expensive
- **Closed enclosures**
  - Possible for simple functions (e.g. fieldbus control)
  - Cost effective; when deployed in big number, e.g. LHC power converter control gateways







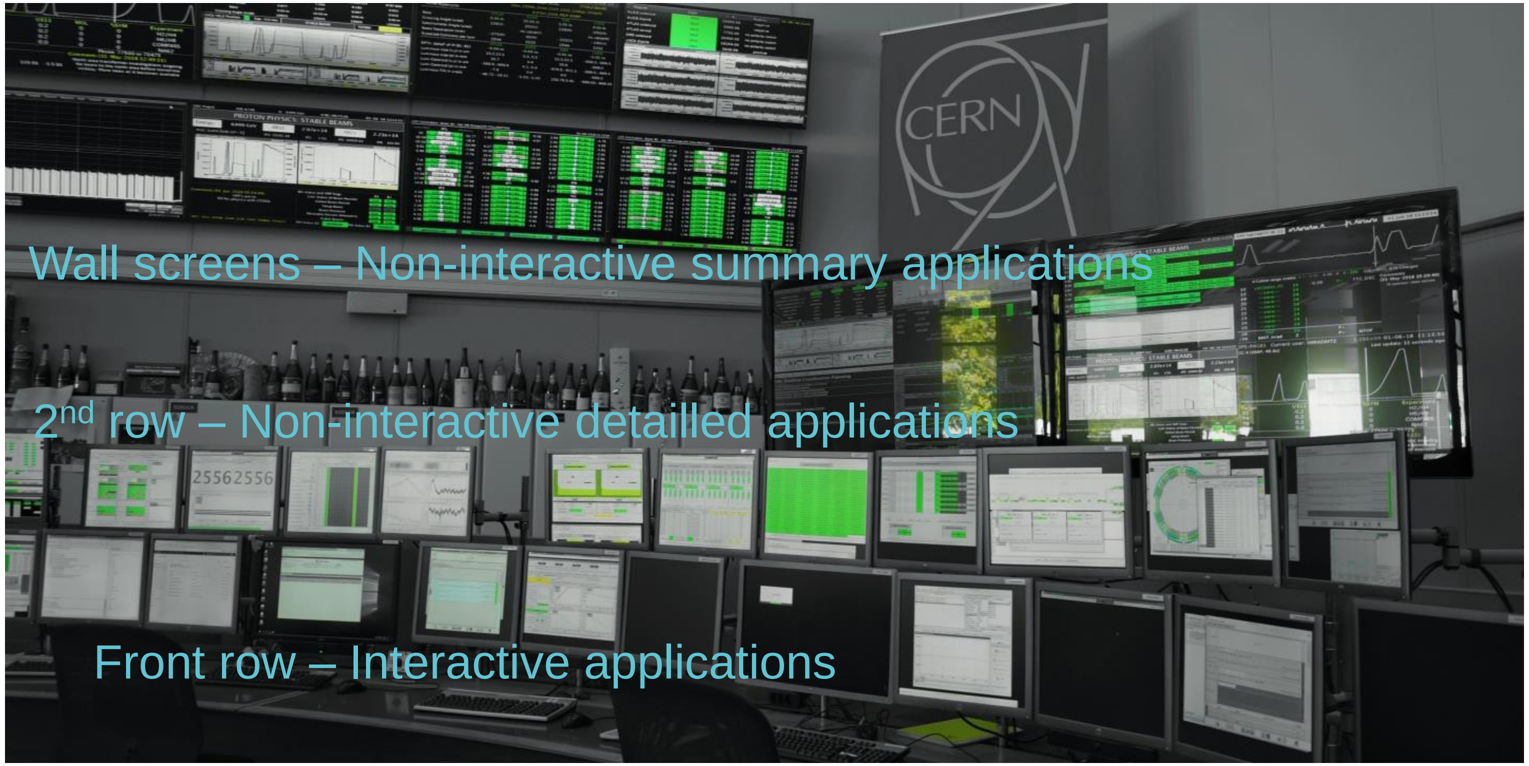


High-Level Hardware Architecture

Middle Tier

- IT-Computer-centre type of hardware
- High-density
- Highly available (redundancy and hot-swap)

High-Level Hardware Architecture



Control Room Computers

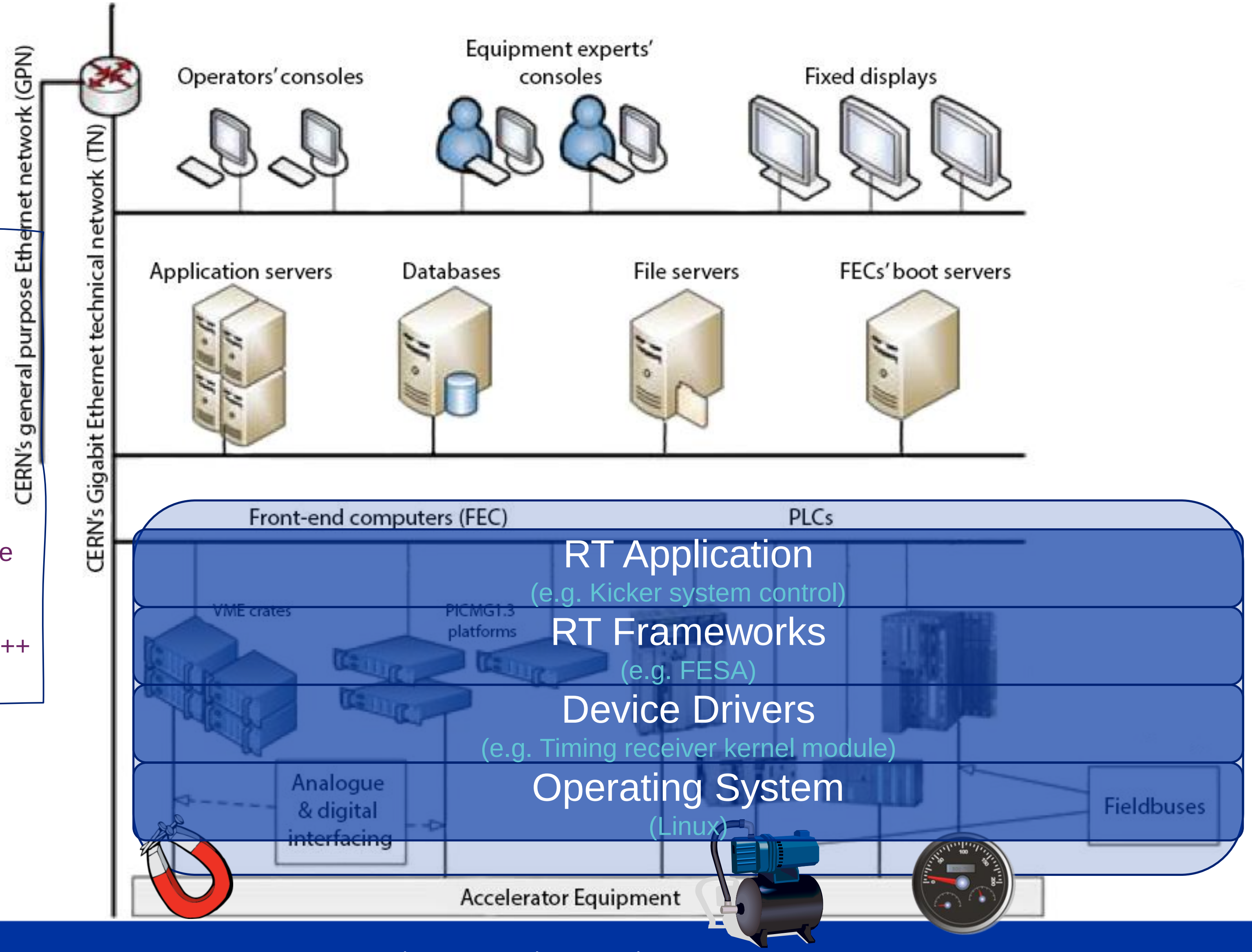
- As much as possible COTS desktop PCs but MTBF requirements might be difficult to satisfy
- Users expect modern reactive GUIs
- Several layers of screens to have as much data as possible available

High-Level Software Architecture

Front-end Tier

Real-time control and acquisition

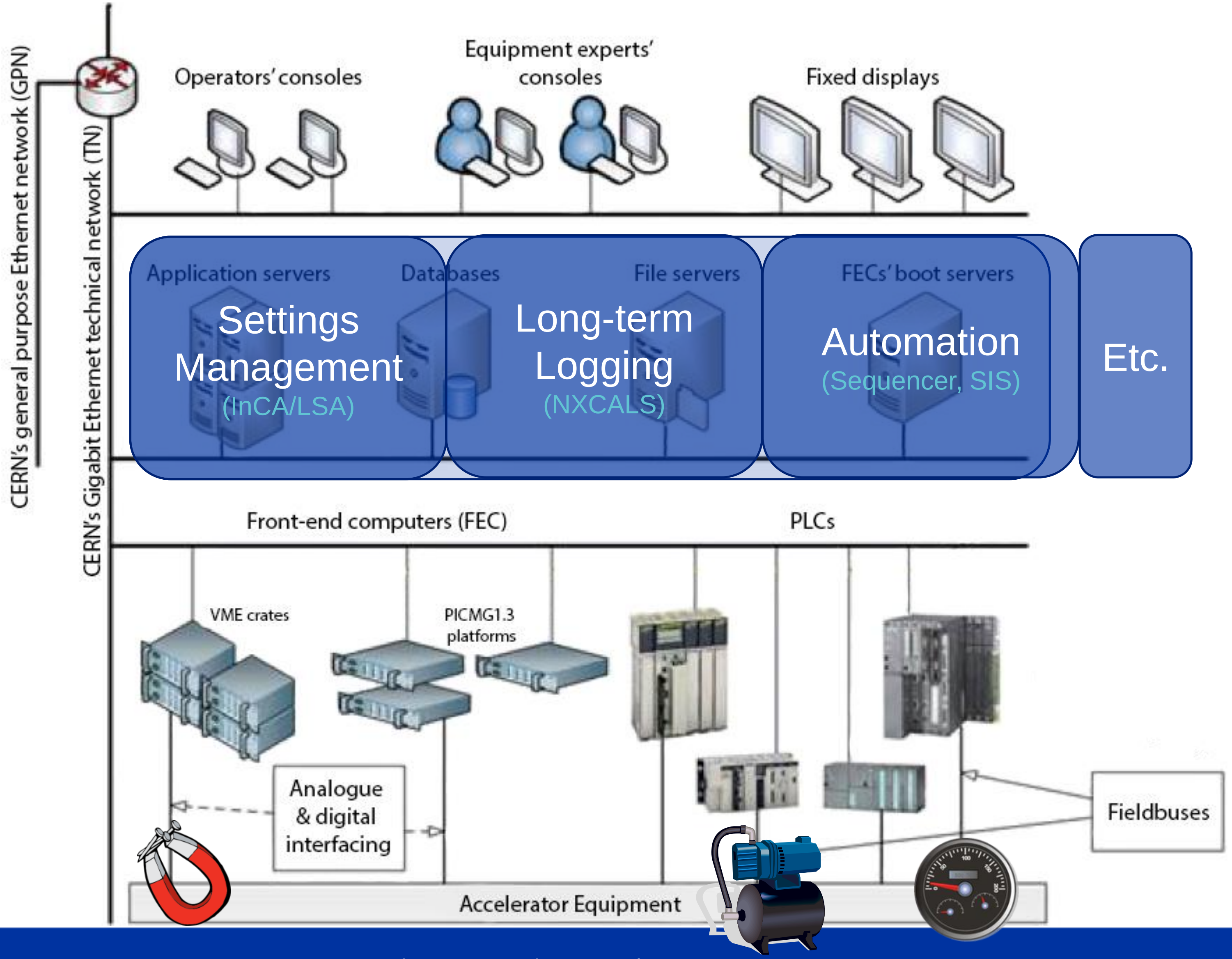
- Limited, local scope
- Fast reaction possible (interrupts)
- Limited computing power (compared to other tiers)
- Equipment processing to provides a high-level view of the hardware
- Real-time (RT) applications relies on frameworks, which capture the recuring aspects (react to events, publish new data, etc.) E.g. Front-End Software Architecture (FESA)
- Based on technologies closed to the hardware (C for drivers, C++ for RT, etc.)



High-Level Software Architecture

Business Tier
General purpose services & Specific business logic

- Broader scope; able to coordinate the entire accelerator
- Powerful computers
- Less reactive (network) and at a higher-level of abstraction
- Based on technologies that are better suited for high-level business logic (e.g. Java)



High-Level Software Architecture

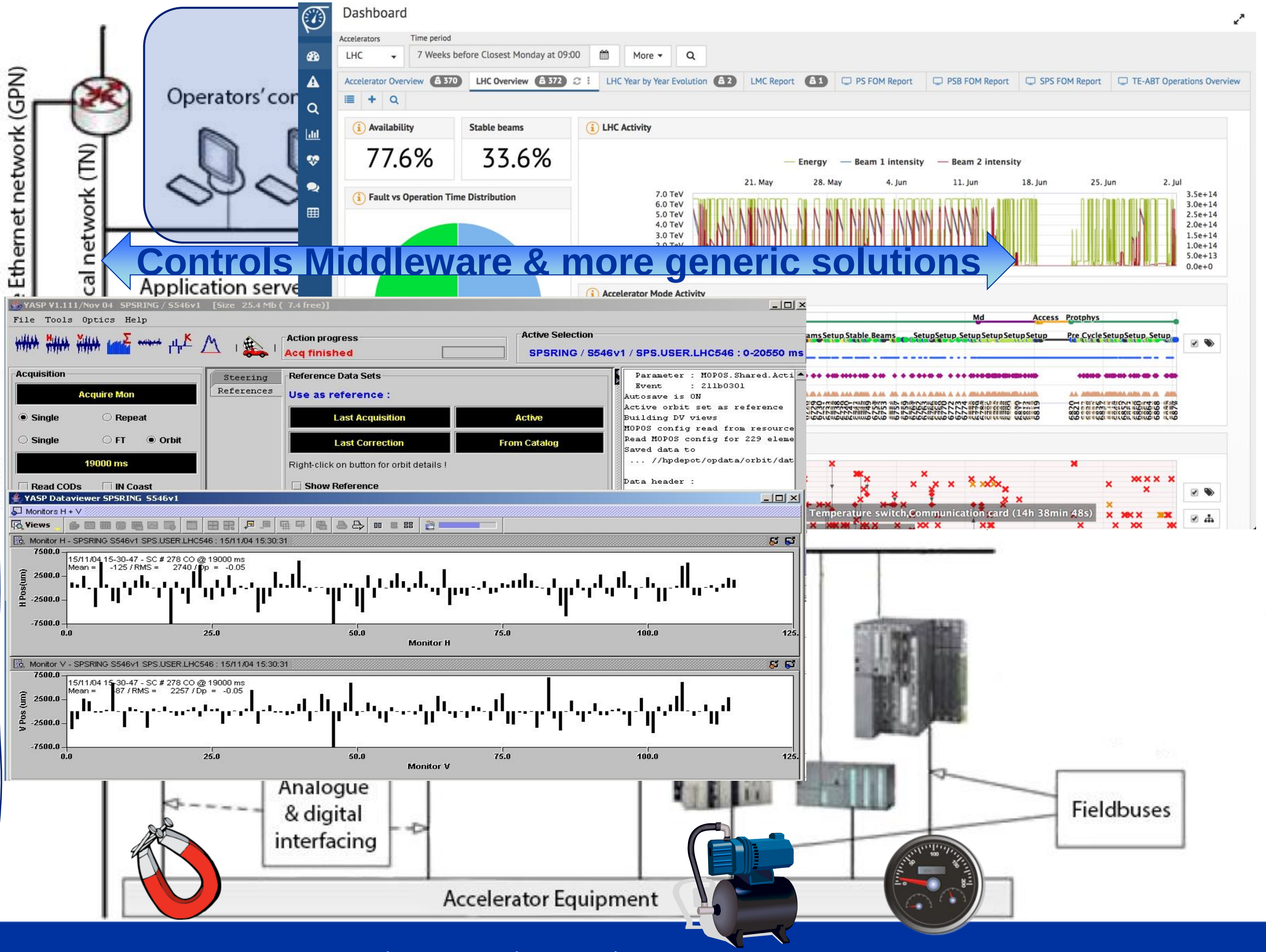
Presentation Tier
Graphical applications

Different technologies available
- Java Swing, Java FX
- Qt, PyQt
- Web ecosystem (Angular, View.js, etc.)

Keywords:
Graphical User Interface (GUI)
Command-line interface (CLI)

Communication

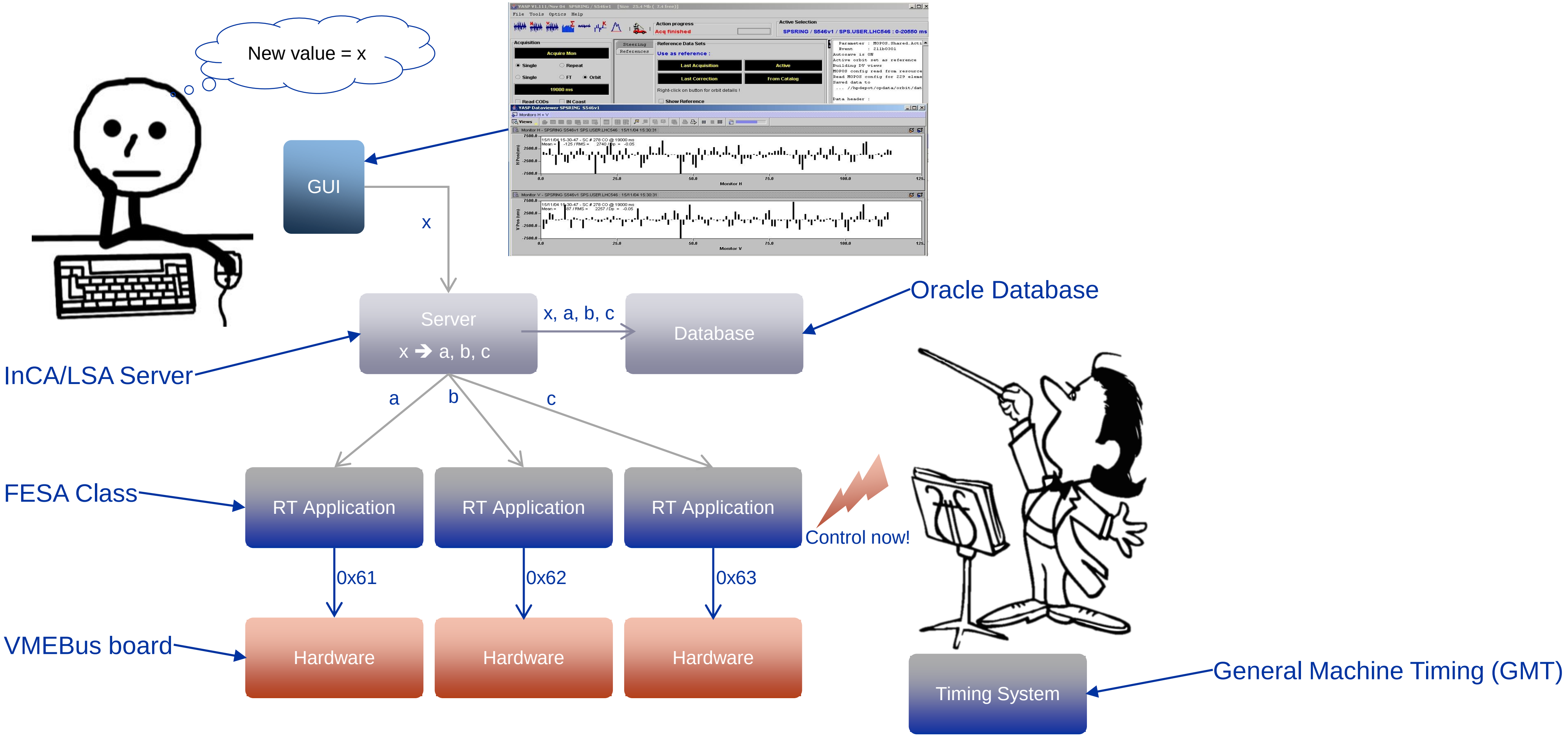
- Accelerator-specific protocols for the lower layers
  - Controls Middleware (CMW)
- Potentially, more generic technologies for the higher layers
  - RMI/JMS
  - REST API
  - gRPC
  - …



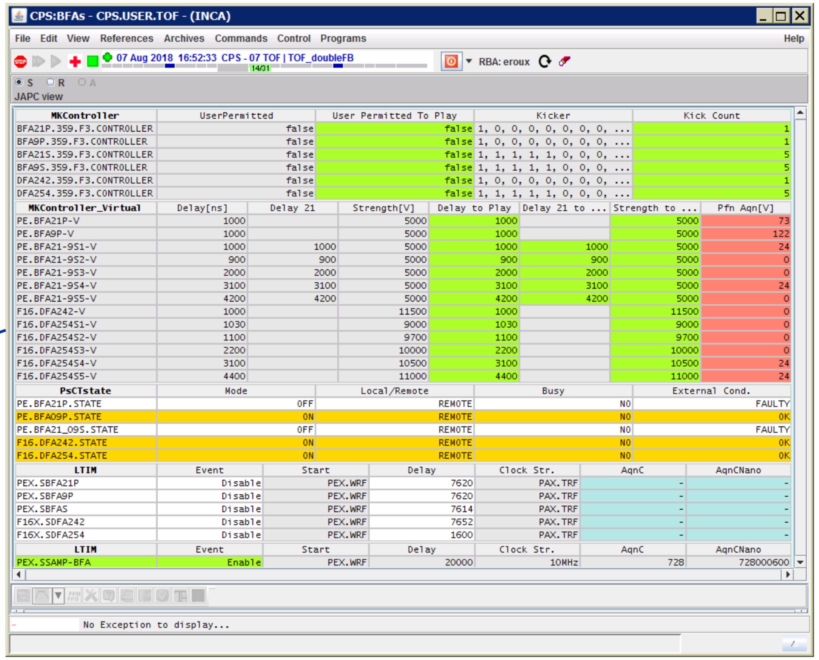
A few examples

And the key components used…

Example 1 – Control



Example 2 – Monitoring



CPS:BFAs - CPS.USER.TOF - (INCA)

File Edit View References Archives Commands Control Programs Help

07 Aug 2018 16:52:33 CPS - 07 TOF | TOF_doubleFB 14/31 RBA: eroux

S R A

JAPC view

MKController	UserPermitted	User Permitted To Play	Kicker	Kick Count
BFA21P.359.F3.CONTROLLER	false	false	1, 0, 0, 0, 0, 0, 0, 0, ...	1
BFA9P.359.F3.CONTROLLER	false	false	1, 0, 0, 0, 0, 0, 0, 0, ...	1
BFA21S.359.F3.CONTROLLER	false	false	1, 1, 1, 1, 1, 0, 0, 0, ...	5
BFA9S.359.F3.CONTROLLER	false	false	1, 1, 1, 1, 1, 0, 0, 0, ...	5
DFA242.359.F3.CONTROLLER	false	false	1, 0, 0, 0, 0, 0, 0, 0, ...	1
DFA254.359.F3.CONTROLLER	false	false	1, 1, 1, 1, 1, 0, 0, 0, ...	5

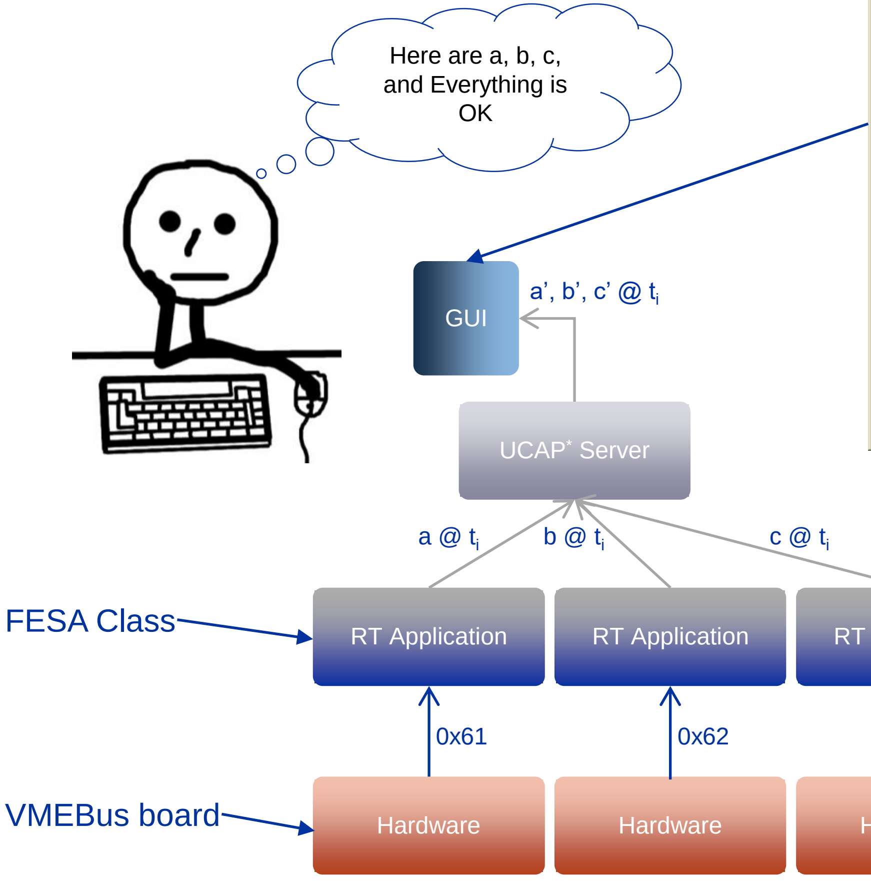
MKController_Virtual	Delay[ns]	Delay 21	Strength[V]	Delay to Play	Delay 21 to ...	Strength to ...	Pfn Aqn[V]
PE.BFA21P-V	1000		5000	1000		5000	73
PE.BFA9P-V	1000		5000	1000		5000	122
PE.BFA21-9S1-V	1000	1000	5000	1000	1000	5000	24
PE.BFA21-9S2-V	900	900	5000	900	900	5000	0
PE.BFA21-9S3-V	2000	2000	5000	2000	2000	5000	0
PE.BFA21-9S4-V	3100	3100	5000	3100	3100	5000	24
PE.BFA21-9S5-V	4200	4200	5000	4200	4200	5000	0
F16.DFA242-V	1000		11500	1000		11500	0
F16.DFA254S1-V	1030		9000	1030		9000	0
F16.DFA254S2-V	1100		9700	1100		9700	0
F16.DFA254S3-V	2200		10000	2200		10000	0
F16.DFA254S4-V	3100		10500	3100		10500	24
F16.DFA254S5-V	4400		11000	4400		11000	24

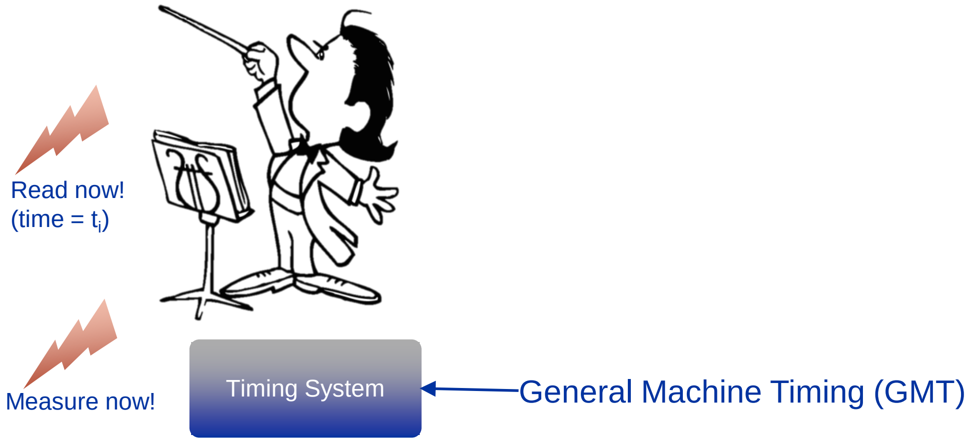
PsCTstate	Mode	Local/Remote	Busy	External Cond.
PE.BFA21P.STATE	OFF	REMOTE	NO	FAULTY
PE.BFA09P.STATE	ON	REMOTE	NO	OK
PE.BFA21_09S.STATE	OFF	REMOTE	NO	FAULTY
F16.DFA242.STATE	ON	REMOTE	NO	OK
F16.DFA254.STATE	ON	REMOTE	NO	OK

LTIM	Event	Start	Delay	Clock Str.	AqnC	AqnCNano
PEX.SBFA21P	Disable	PEX.WRF	7620	PAX.TRF	-	-
PEX.SBFA9P	Disable	PEX.WRF	7620	PAX.TRF	-	-
PEX.SBFAS	Disable	PEX.WRF	7614	PAX.TRF	-	-
F16X.SDFA242	Disable	PEX.WRF	7652	PAX.TRF	-	-
F16X.SDFA254	Disable	PEX.WRF	1600	PAX.TRF	-	-

LTIM	Event	Start	Delay	Clock Str.	AqnC	AqnCNano
PEX.SSAMP-BFA	Enable	PEX.WRF	20000	10MHz	728	728000600

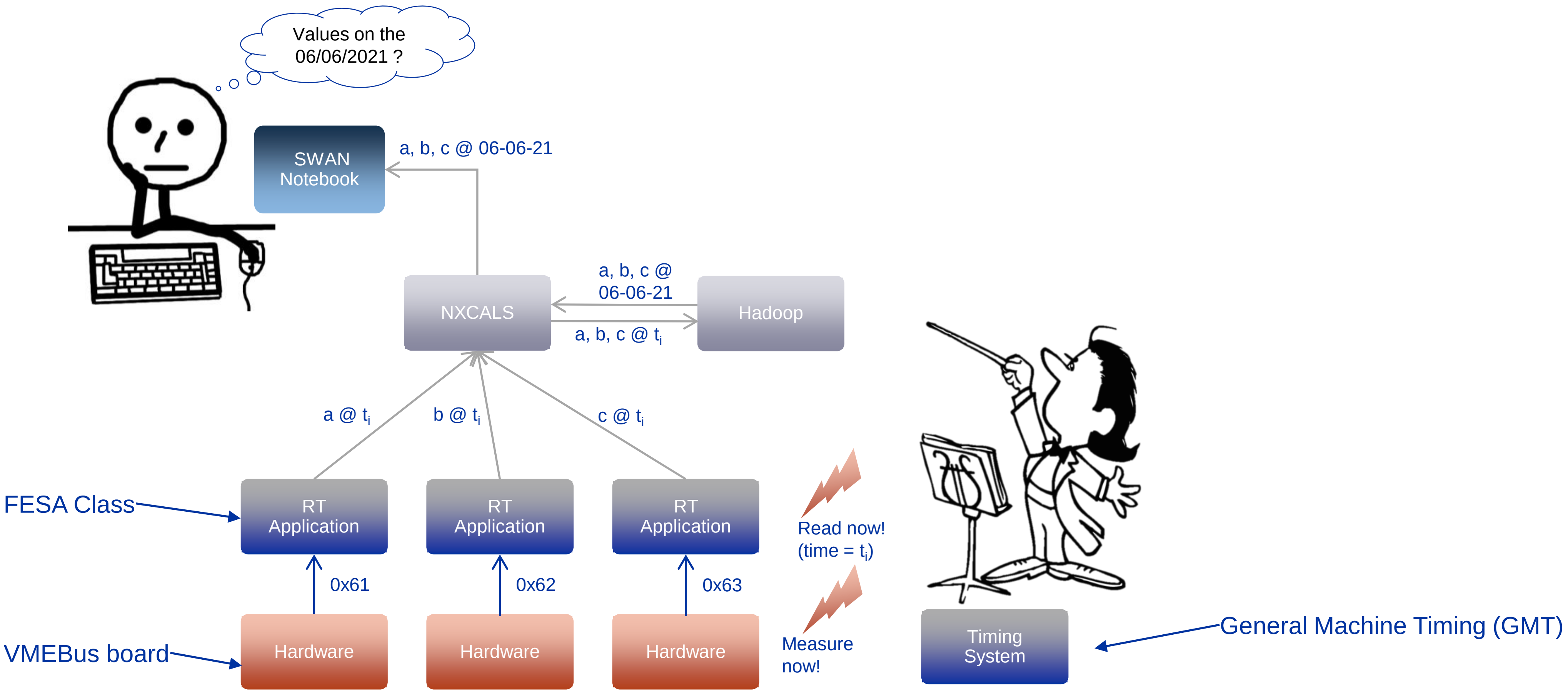
No Exception to display...





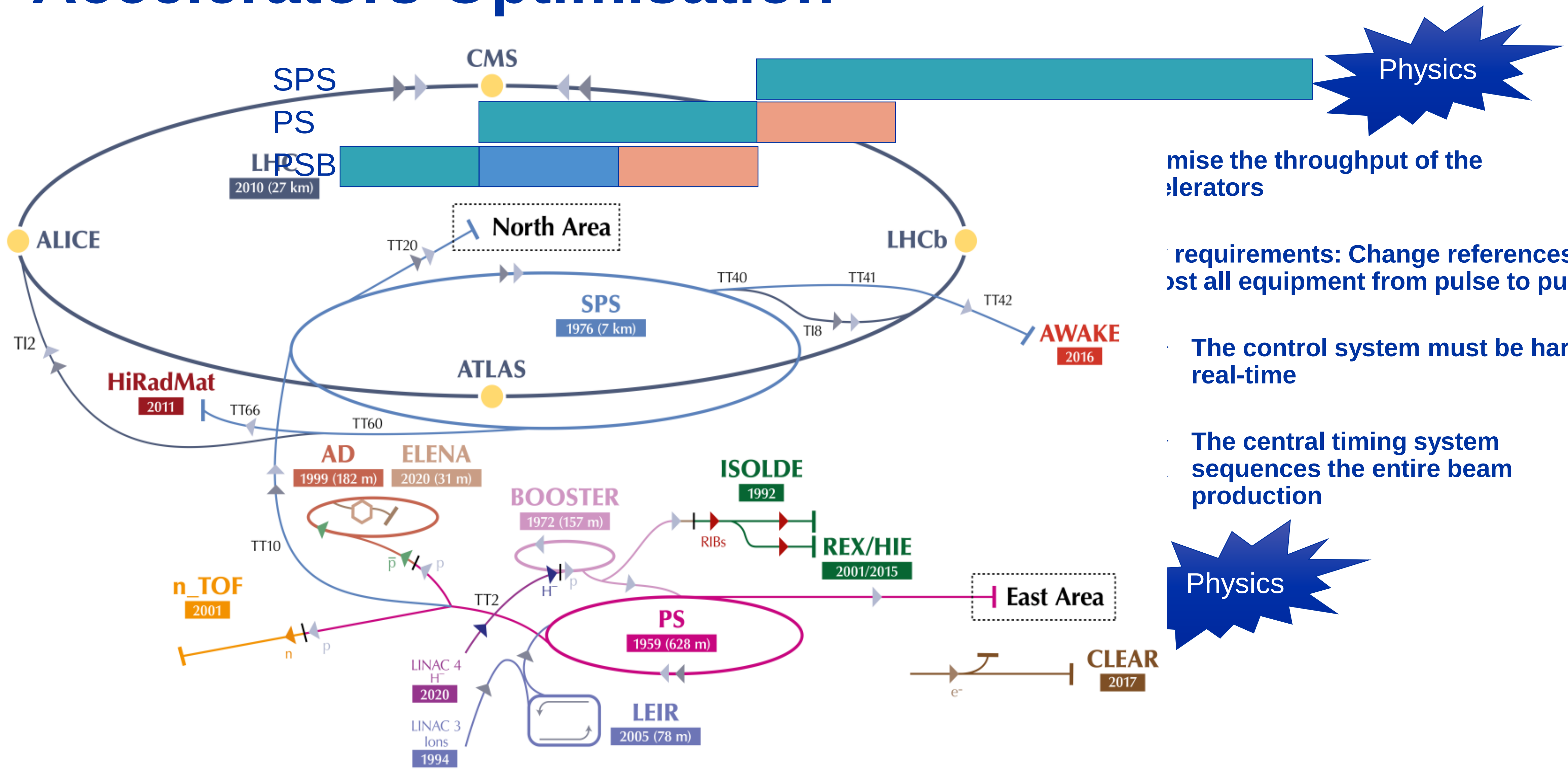
*UCAP: Unified Controls Acquisition & Processing framework

Example 3 – Logging



*SWAN: Service for Web based ANalysis

Accelerators Optimisation



mise the throughput of the
elerators

requirements: Change references of
ost all equipment from pulse to pulse

- The control system must be hard real-time
- The central timing system sequences the entire beam production

Program for Tomorrow…

- **Review the examples but looking at the technical implementation details**
  - Which technologies
  - Workflows
  - Design choices
  - Etc.

- **If possible, review**
  - The high-level architecture (get used to the names)
  - The examples
  - The timing system

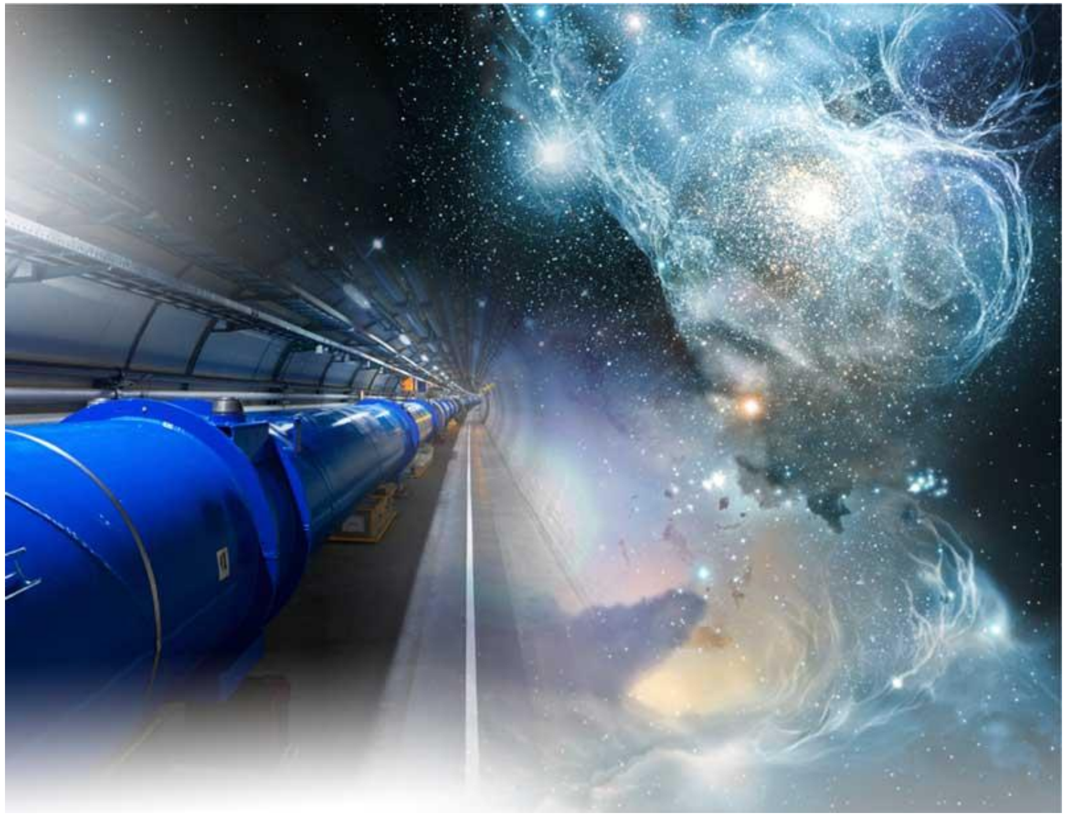
Want to know more?

CERN Beams Department (https://beams.cern/)

Introduction to BE-CO Control System, 2019 Edition, S. Deghaye & E. Fortescue, CERN, 2020. (https://cds.cern.ch/record/2748122)

Tango Controls (https://www.tango-controls.org/)

EPICS (https://epics-controls.org/)



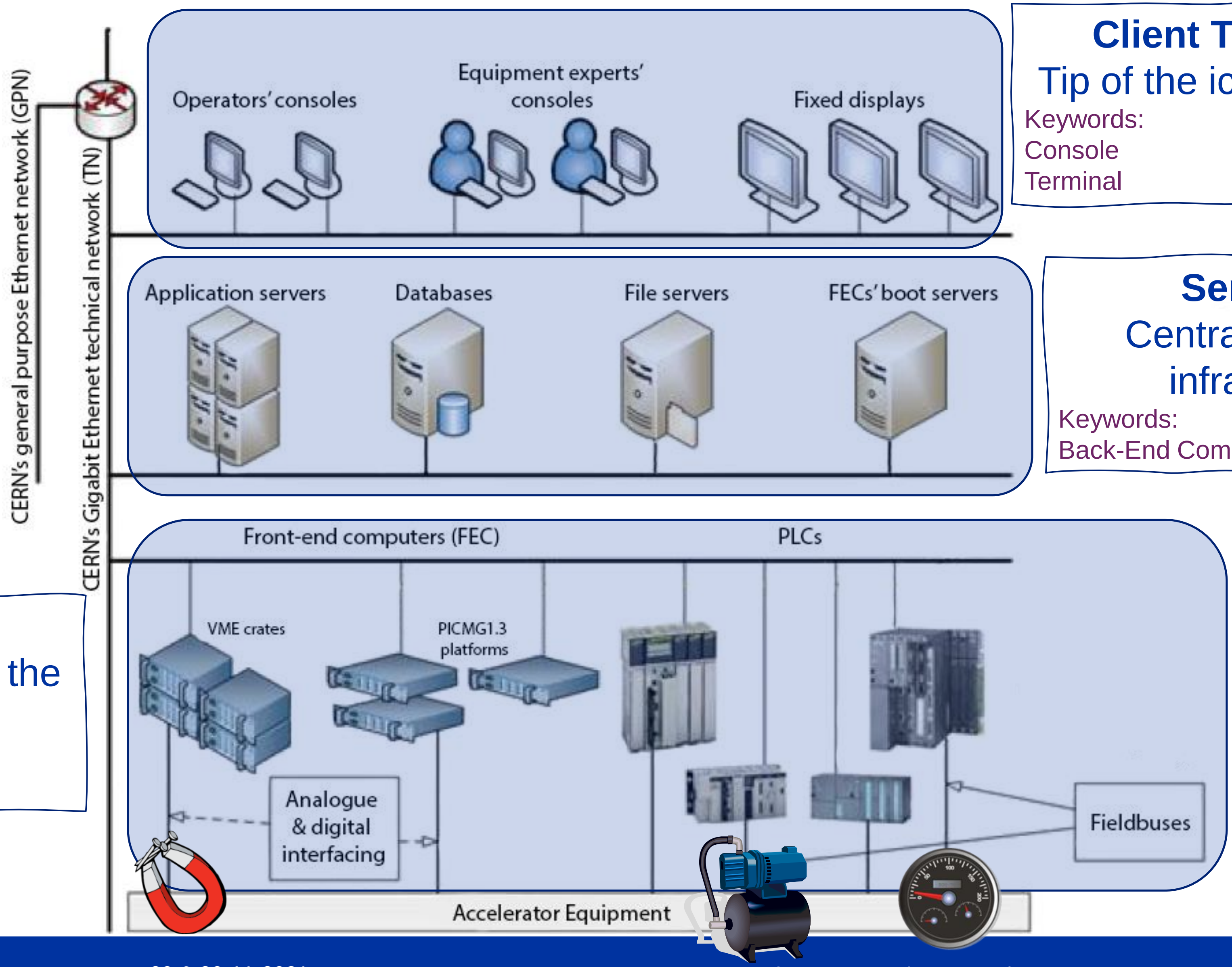
Day 2

Agenda

1. What is an accelerator made of?
2. Control System Requirements
3. Implementation Philosophy
4. A bit of History
5. Hardware & Software Architecture
6. CERN examples & Key Components
7. Detailed implementations and concepts

High-Level Hardware Architecture

Recap



Client Tier
Tip of the iceberg
Keywords:
Console
Terminal

Server Tier
Central computing infrastructure
Keywords:
Back-End Computer (BEC)

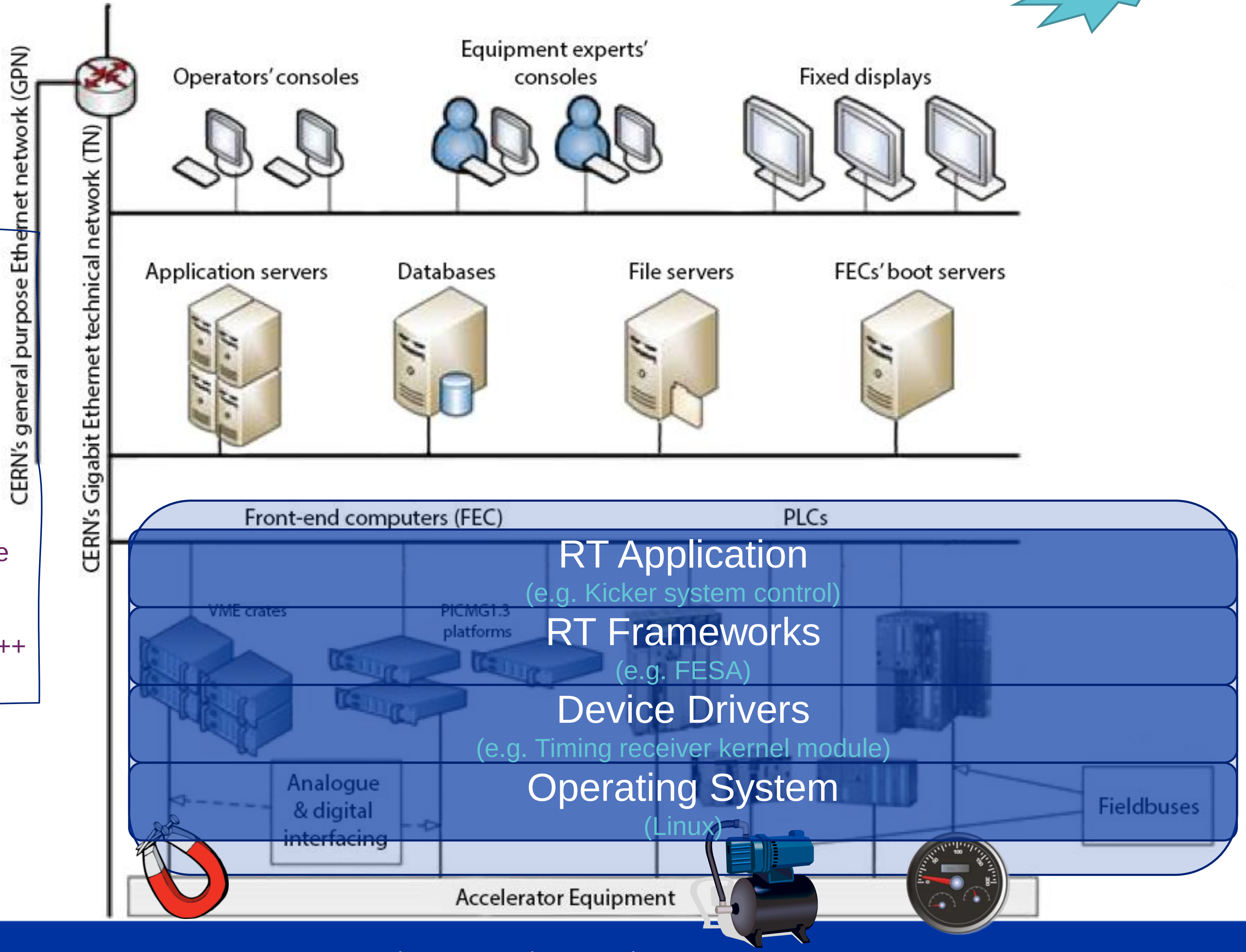
Resource Tier
Electronics close to the accelerator
Keywords:
Front-End Computer (FEC)

High-Level Software Architecture

Recap

Front-end Tier
Real-time control and acquisition

- Limited, local scope
- Fast reaction possible (interrupts)
- Limited computing power (compared to other tiers)
- Equipment processing to provides a high-level view of the hardware
- Real-time (RT) applications relies on frameworks, which capture the recuring aspects (react to events, publish new data, etc.) E.g. Front-End Software Architecture (FESA)
- Based on technologies closed to the hardware (C for drivers, C++ for RT, etc.)

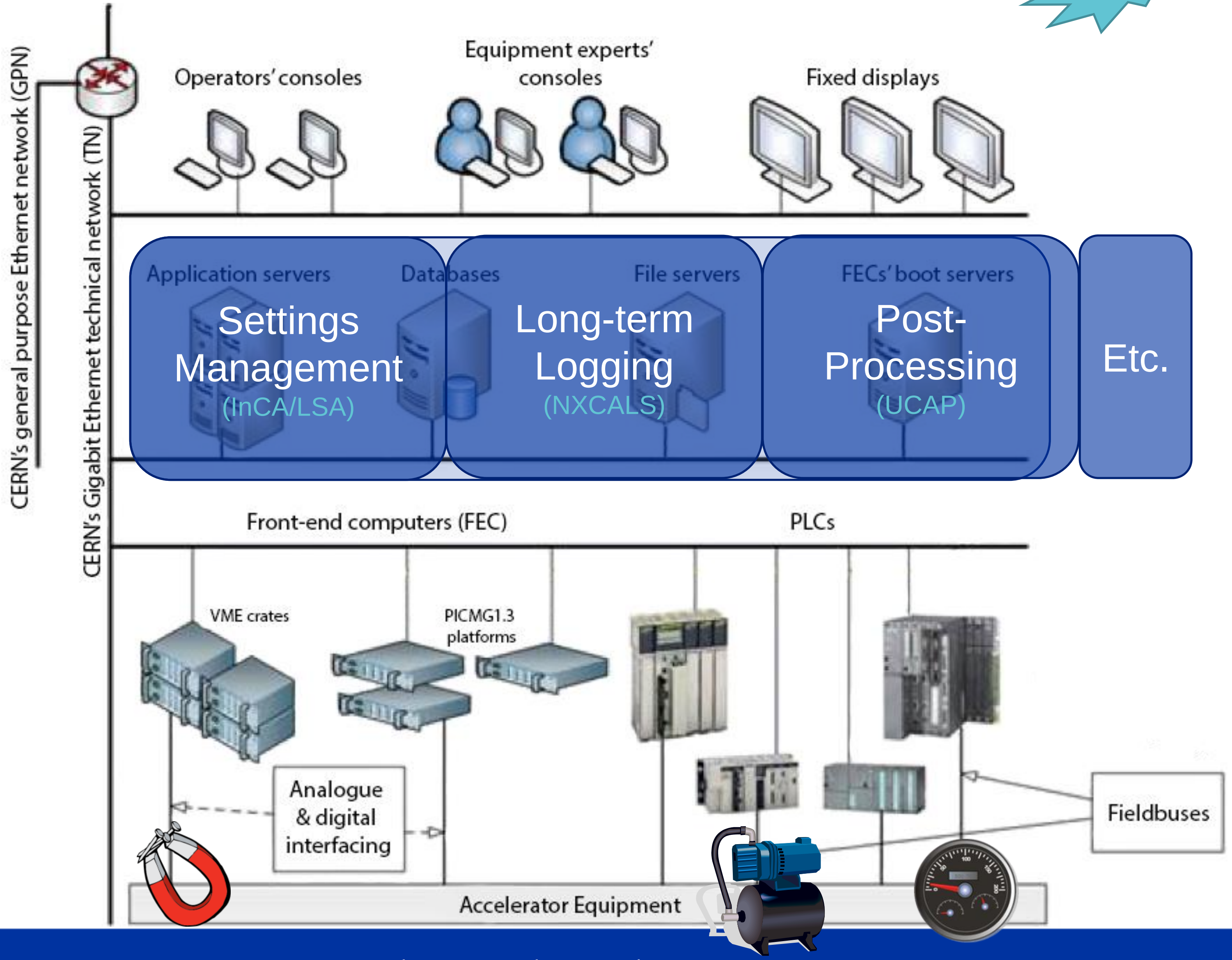


High-Level Software Architecture

Recap

Business Tier
General purpose services & Specific business logic

- Broader scope; able to coordinate the entire accelerator
- Powerful computers
- Less reactive (network) and at a higher-level of abstraction
- Based on technologies that are better suited for high-level business logic (e.g. Java)



High-Level Software Architecture

Recap

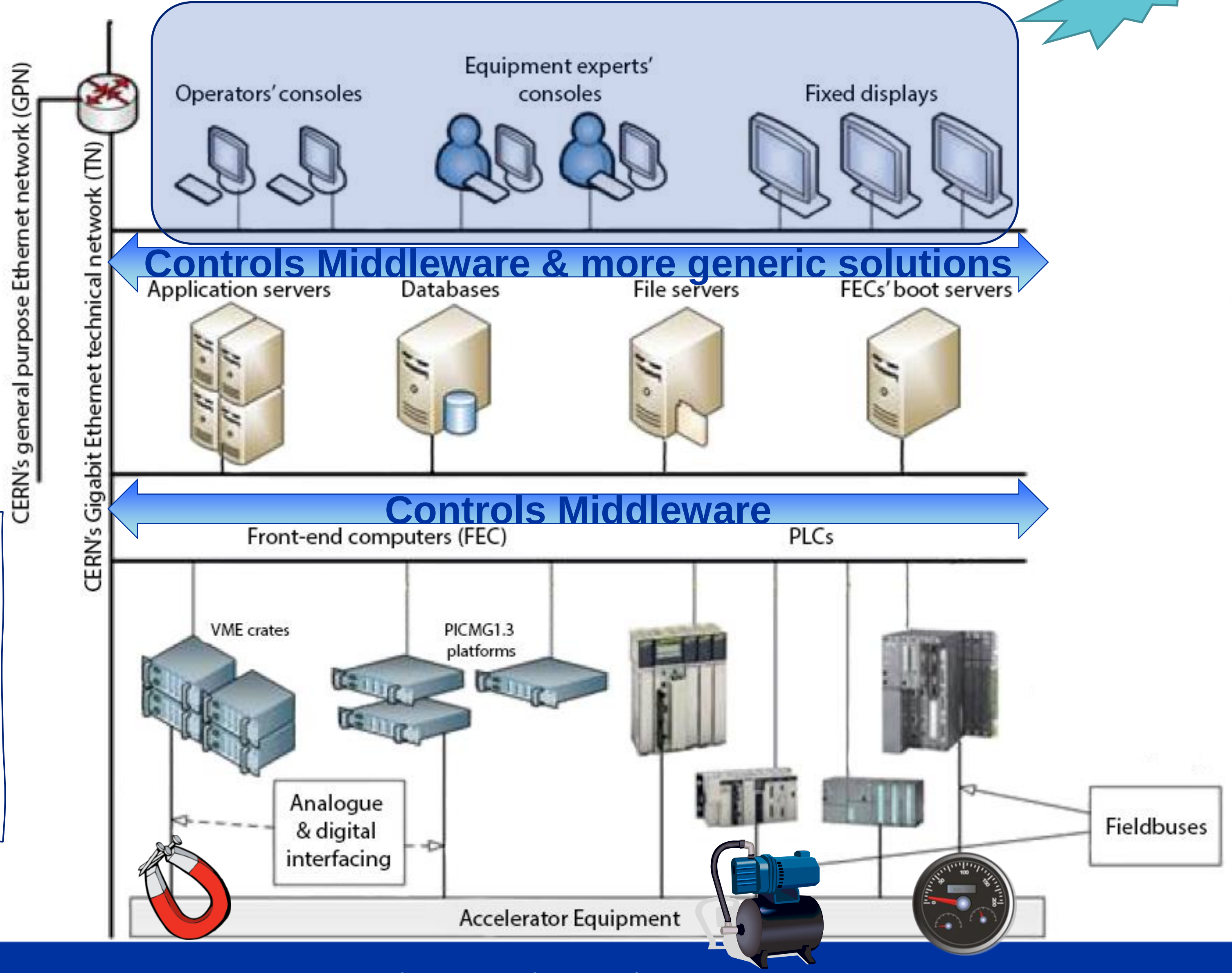
Presentation Tier
Graphical applications

Different technologies available
- Java Swing, Java FX
- Qt, PyQt
- Web ecosystem (Angular, View.js, etc.)

Keywords:
Graphical User Interface (GUI)
Command-line interface (CLI)

Communication

- Accelerator-specific protocols for the lower layers
  - Controls Middleware (CMW)
- Potentially, more generic technologies for the higher layers
  - RMI/JMS
  - REST API
  - gRPC
  - …

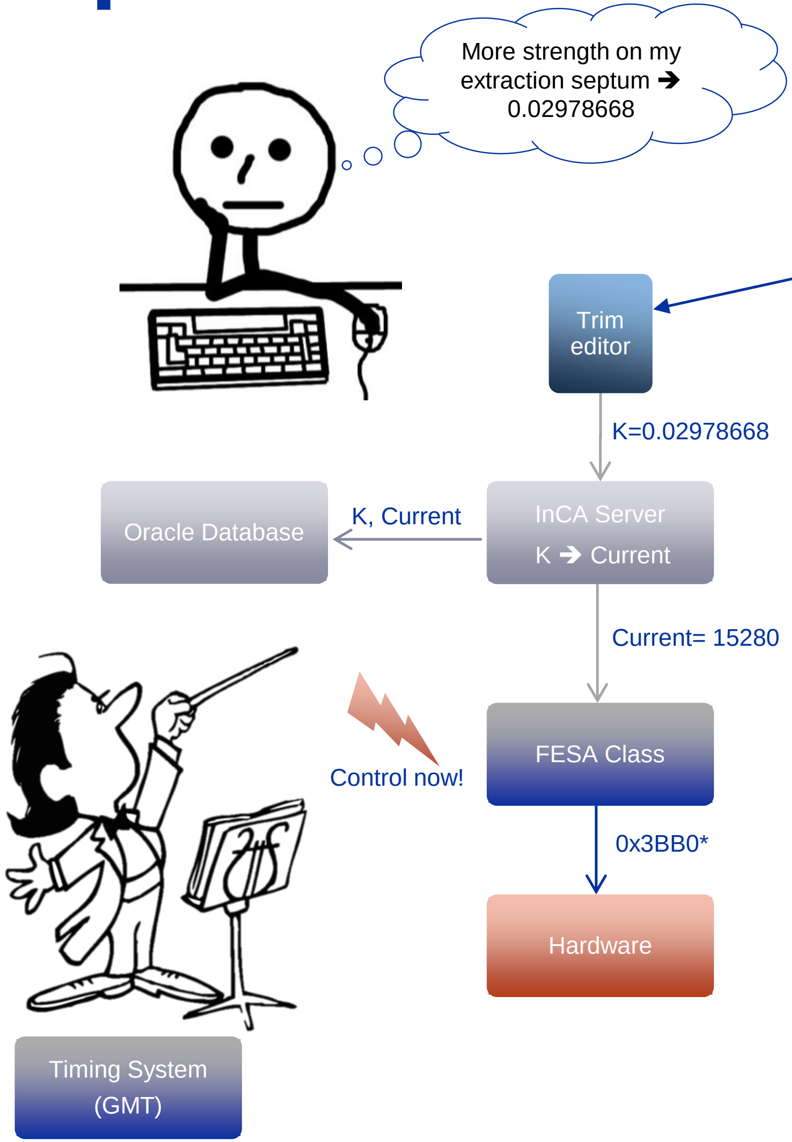


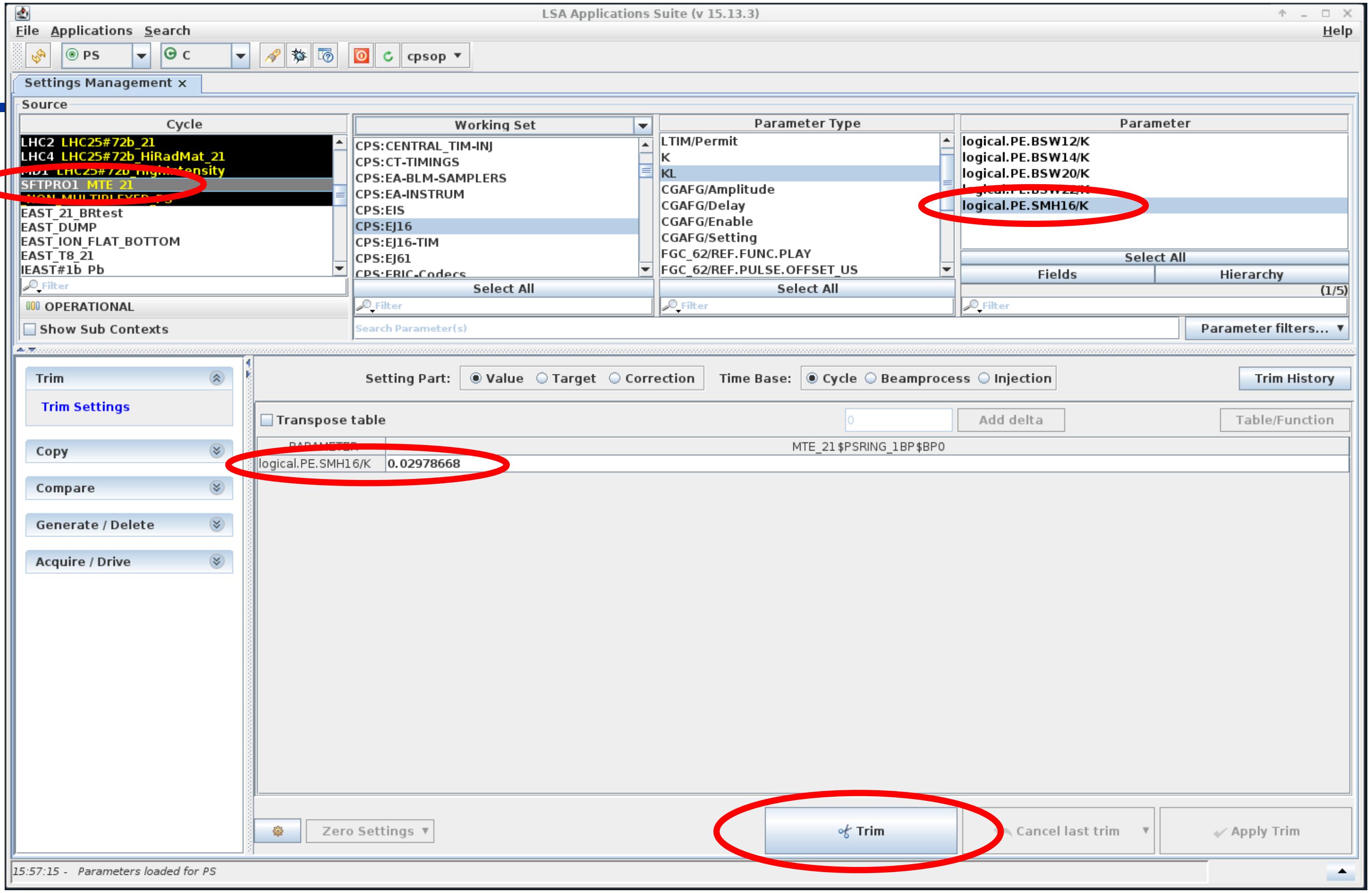
Use Case 1: Control

- **Who:** PS operator
- **What:** Change the strength of the extraction septum towards the SPS for the Fixed Target beam
- **Involved controls components:**
  - InCA/LSA (Setting management)
  - CMW (Controls Middleware)
  - FESA (Real-time hardware control)
  - Timing (Synchronisation)

In all the examples, we assume all the configuration is already done and we focus on the run-time aspects

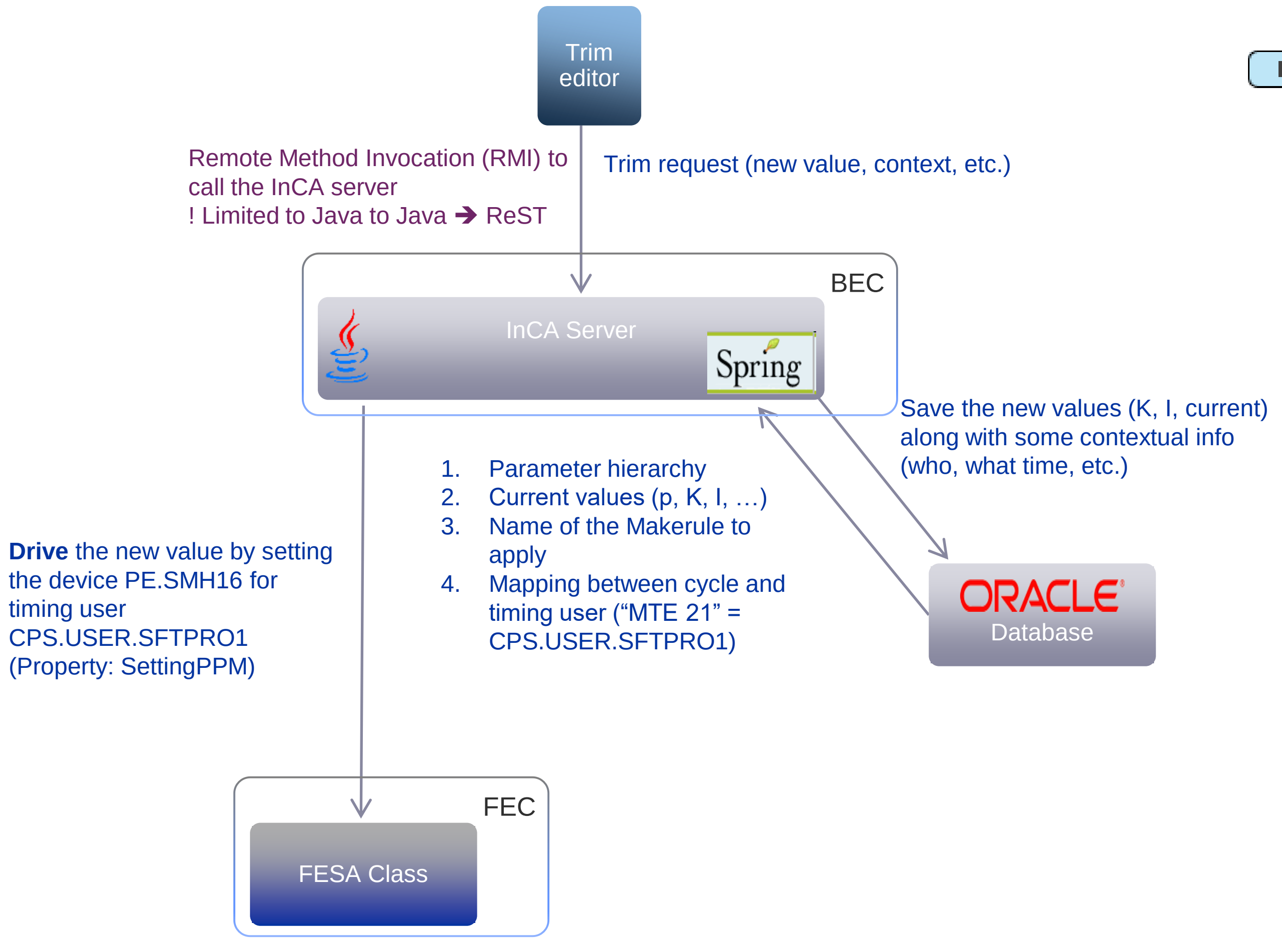
Example 1 – Control –

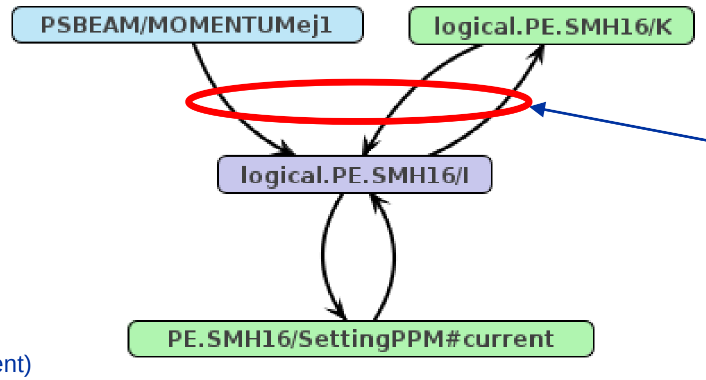




*not a real value

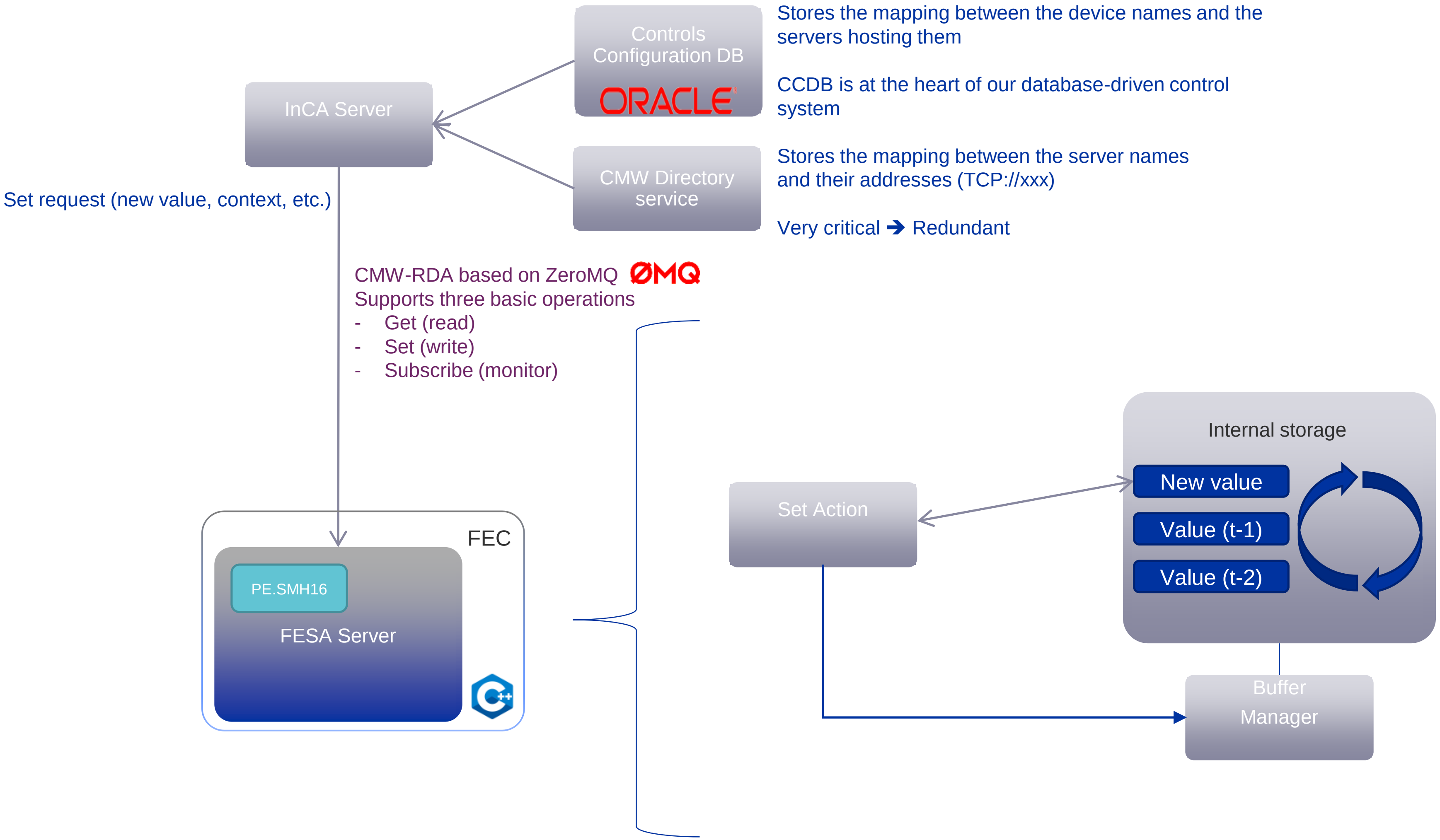
Example 1 – Control - Communication





Makerule: Java class to compute parameters based on other parameters, hardware characteristics (calibration curve) and cycle specific values

Example 1 – Control – Low-level



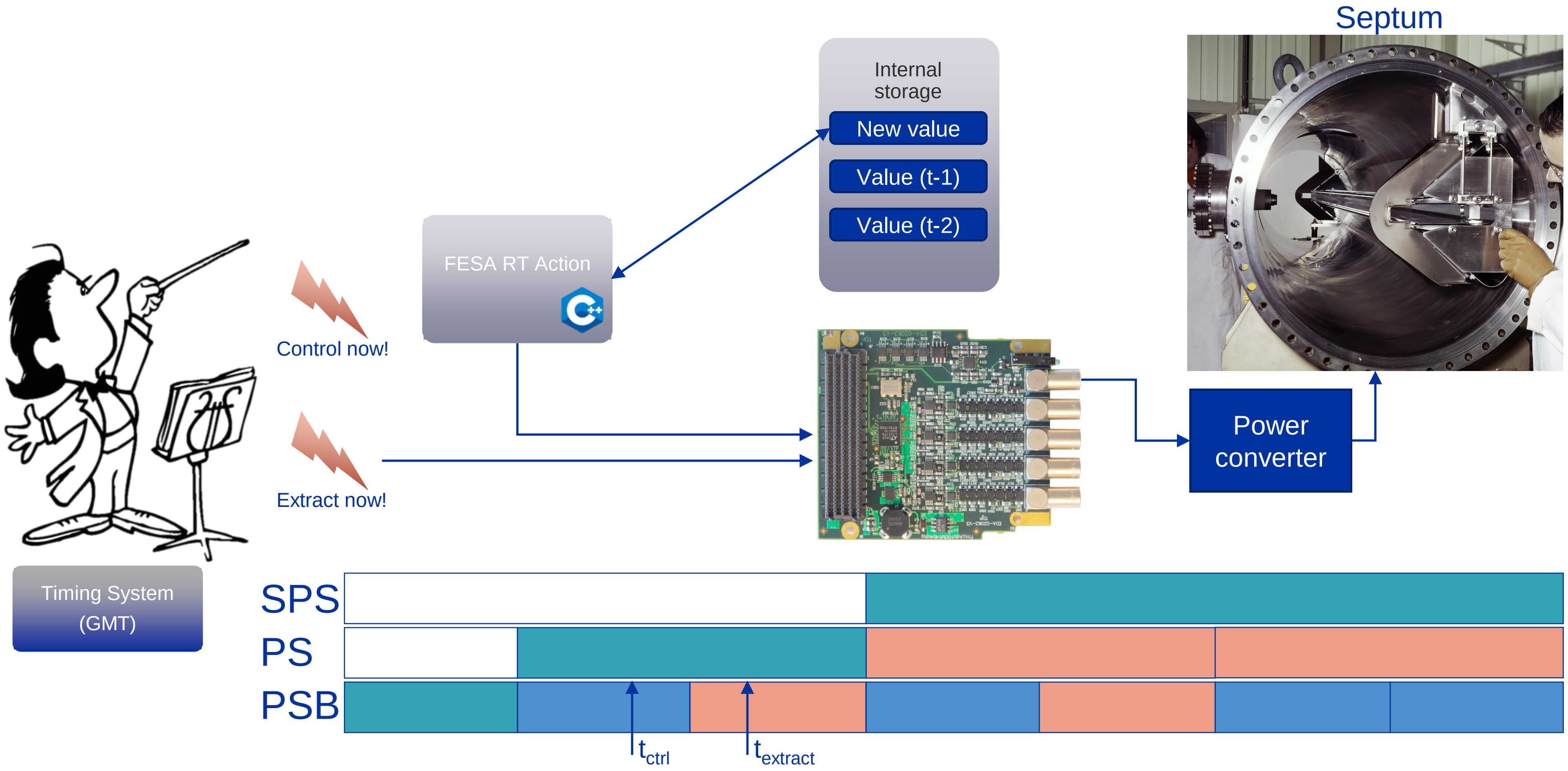
Stores the mapping between the device names and the servers hosting them

CCDB is at the heart of our database-driven control system

Stores the mapping between the server names and their addresses (TCP://xxx)

Very critical ➔ Redundant

Example 1 – Control – Beam production

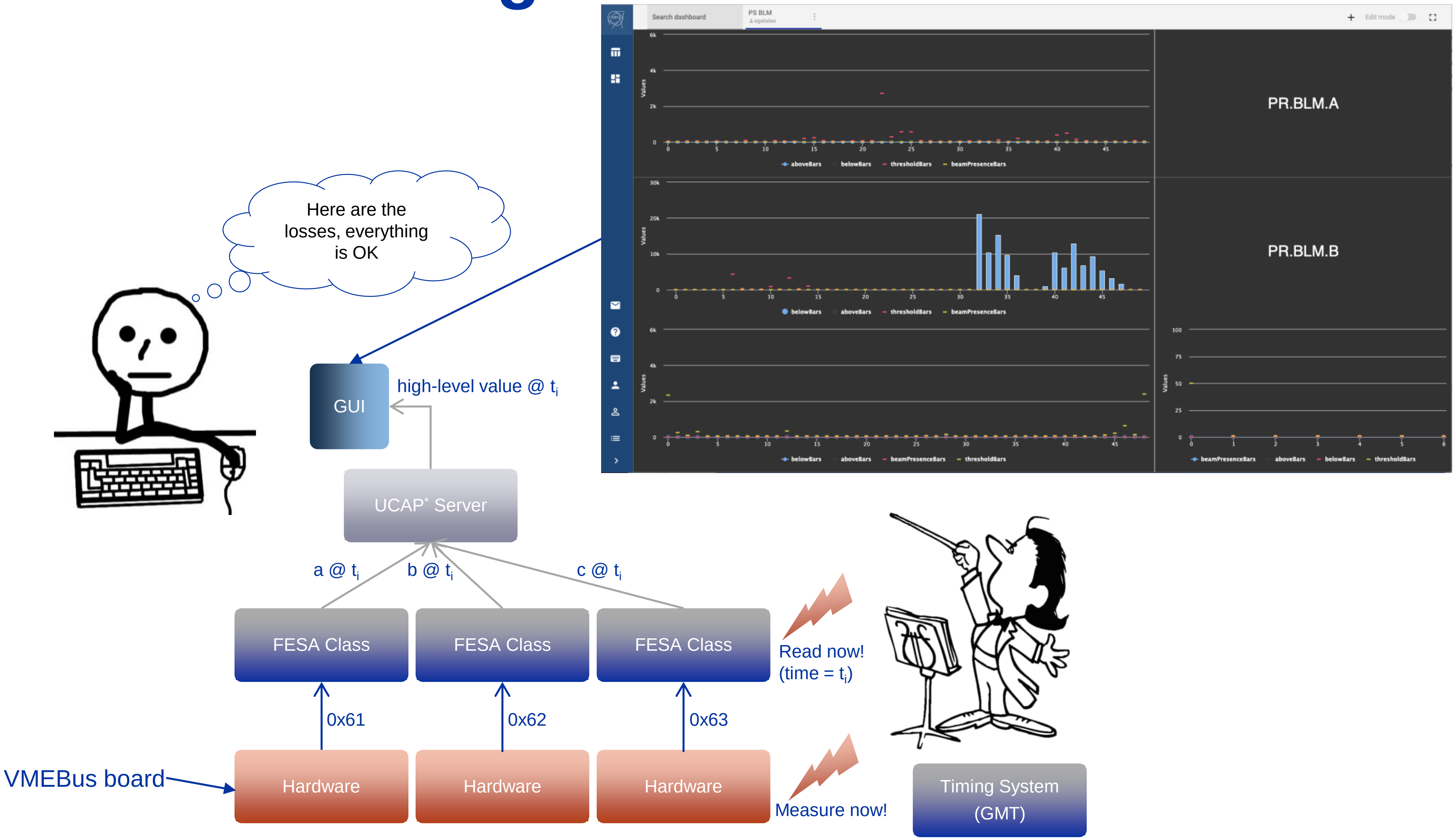


Use Case 2: Acquisition

- **Who:** PS operator
- **What:** Keep an eye on acquisition values of many control devices. The low-level data needs post-processing and will be displayed as a graph in a web page
- **Involved controls components:**
  - Timing (Synchronisation)
  - FESA (Real-time hardware control)
  - CMW (Controls Middleware)
  - UCAP (Unified Controls Acquisition & Processing)
  - WRAP (Web Rapid Application Platform)

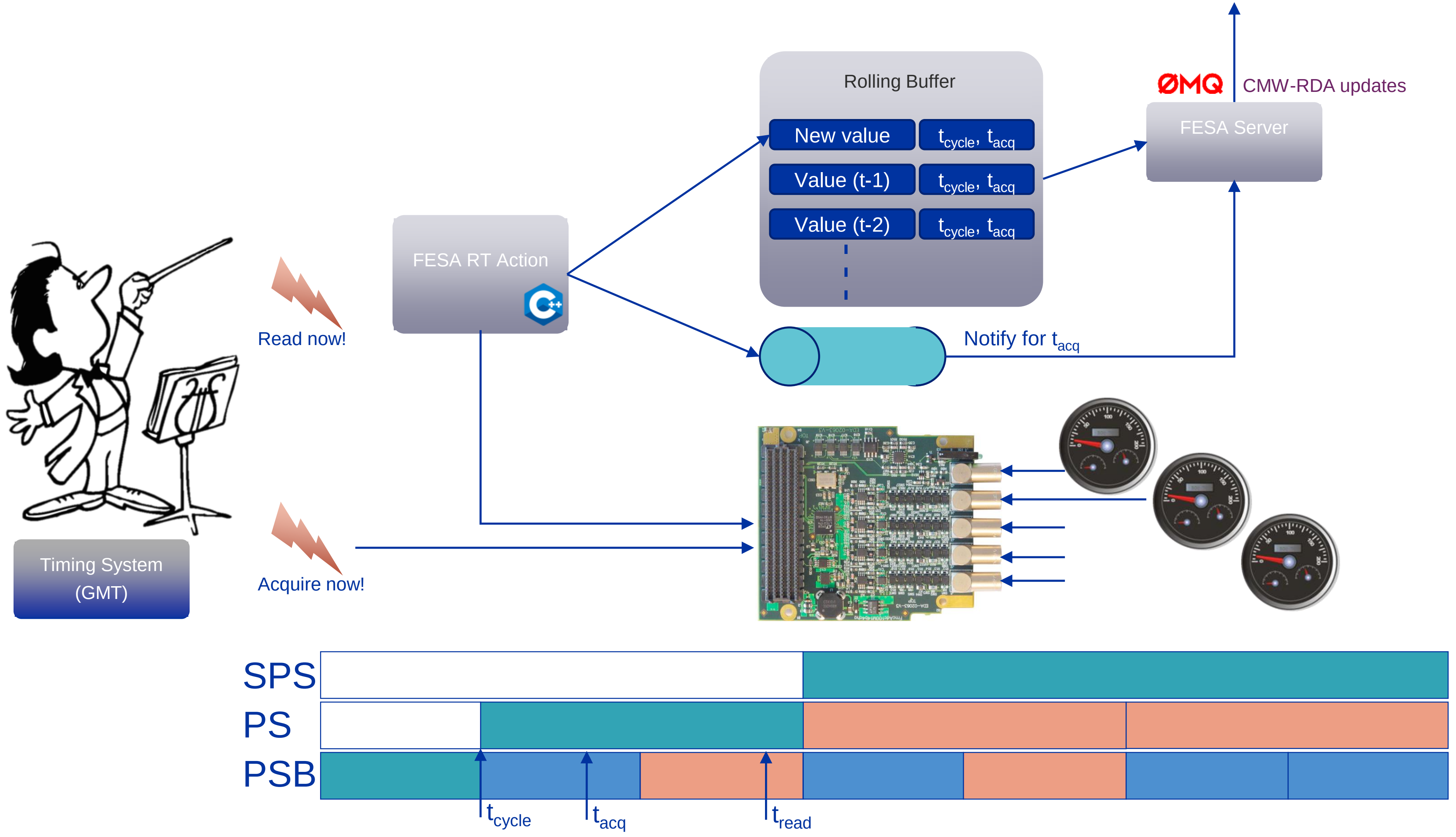
In all the examples, we assume all the configuration is already done and we focus on the run-time aspects

Example 2 – Monitoring

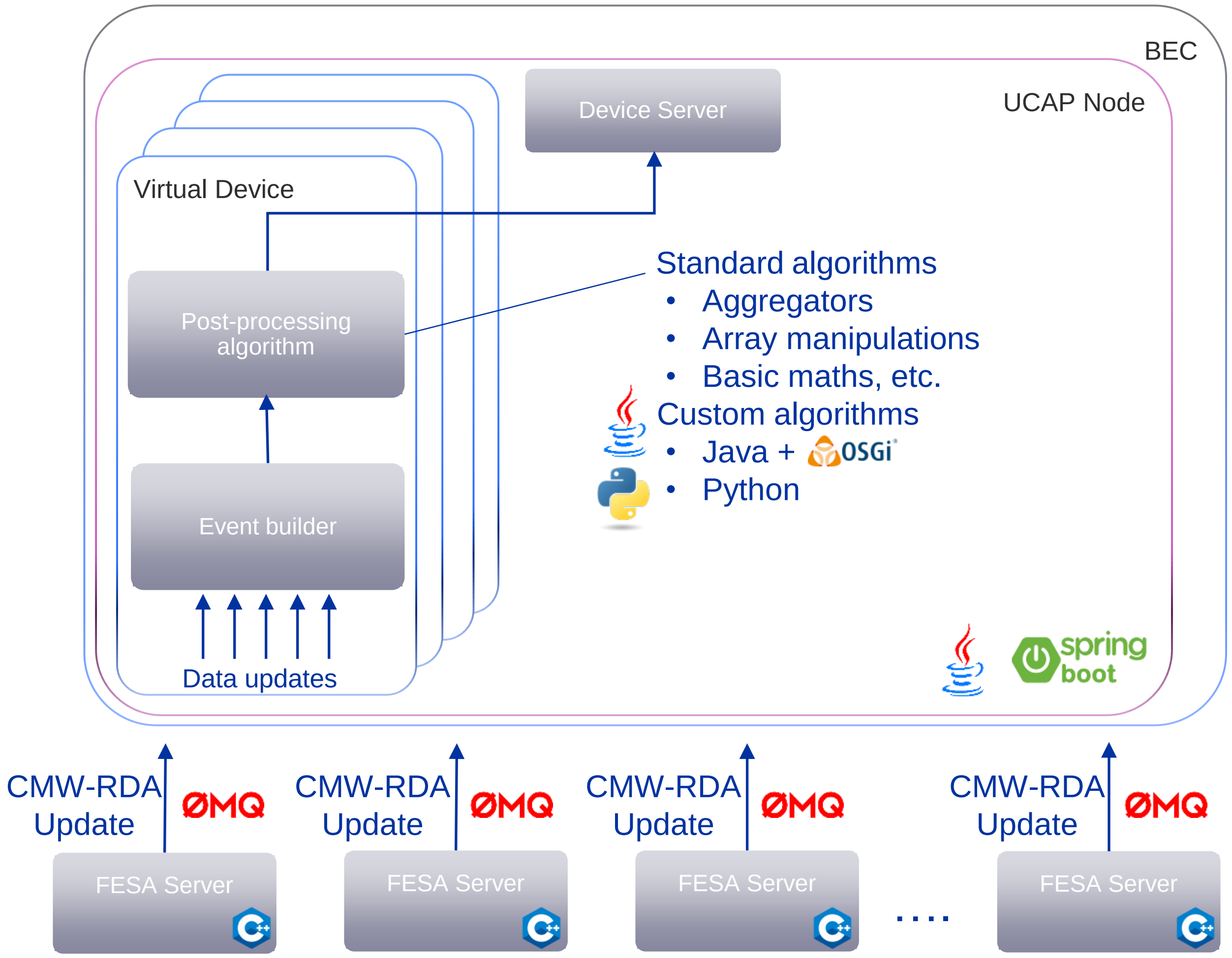


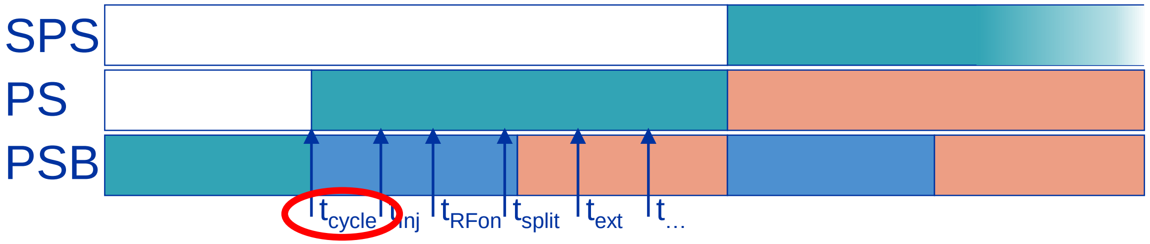
*UCAP: Unified Controls Acquisition & Processing framework

Example 2 – Monitoring – Low-level acquisition



Example 2 – Monitoring – Post-processing



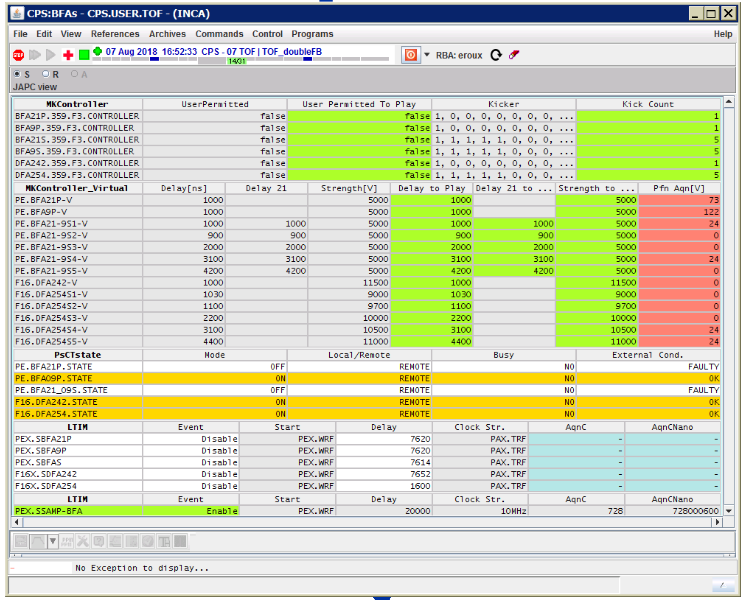


- How to group the incoming data?
  ➔ Start Cycle timestamp (AKA cyclestamp)
- When to trigger the post-processing?
  ➔ Once all the data is there or after a time-out
- How long to wait for late comers?
  ➔ Configurable time-out
- What to do if no data is published?
  ➔ Out-of-the-box monitoring

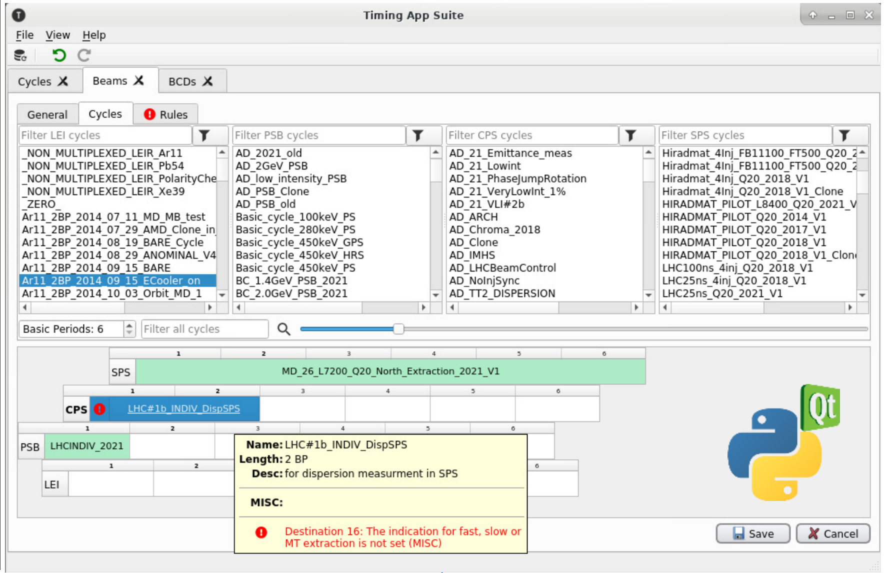
Grafana Prometheus

Example 2 – Monitoring – Diagnostics

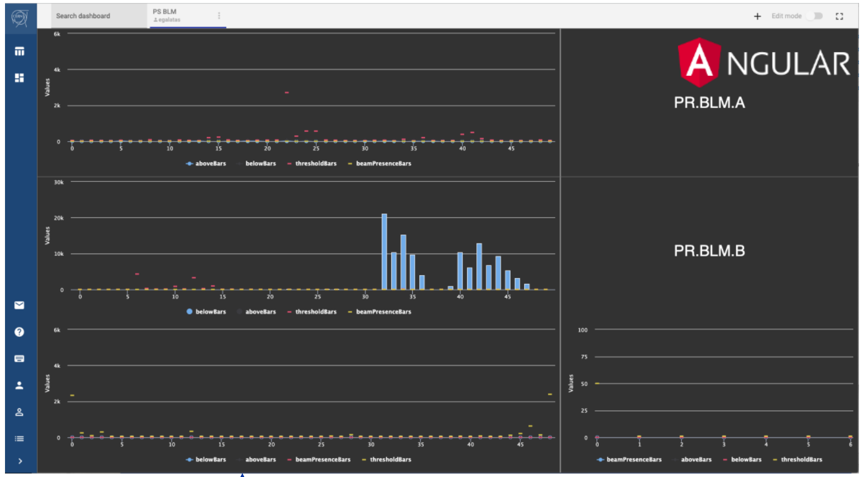
Example 2 – Monitoring – Graphical User Interface



SWING



WRAP



Simplified flow

REST for command/response
HTTP/2 push events for live updates

Controls Configuration DB
ORACLE

Stores the configurations of the web dashboards

BEC

Web back-end server

spring boot

UCAP Node

Device Server

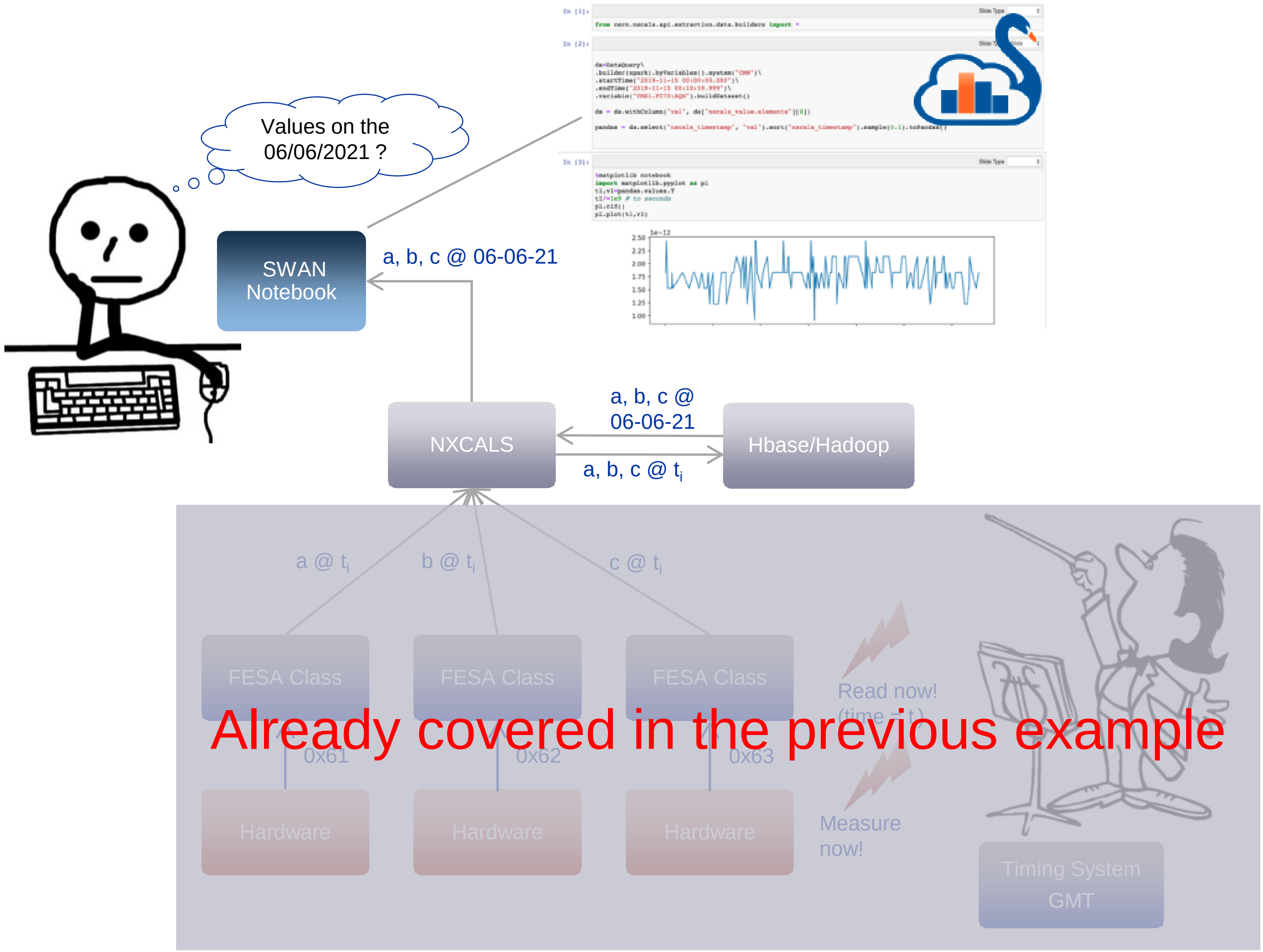
CMW-RDA updates
ØMQ

Use Case 3: Logging

- **Who:** Accelerator physicist
- **What:** Store acquisition values of many control devices long-term and perform analysis
- **Involved controls components:**
  - Timing (Synchronisation)
  - FESA (Real-time hardware control)
  - CMW (Controls Middleware)
  - NXCALS (Data Logging)
  - SWAN (Web-based analysis tool)

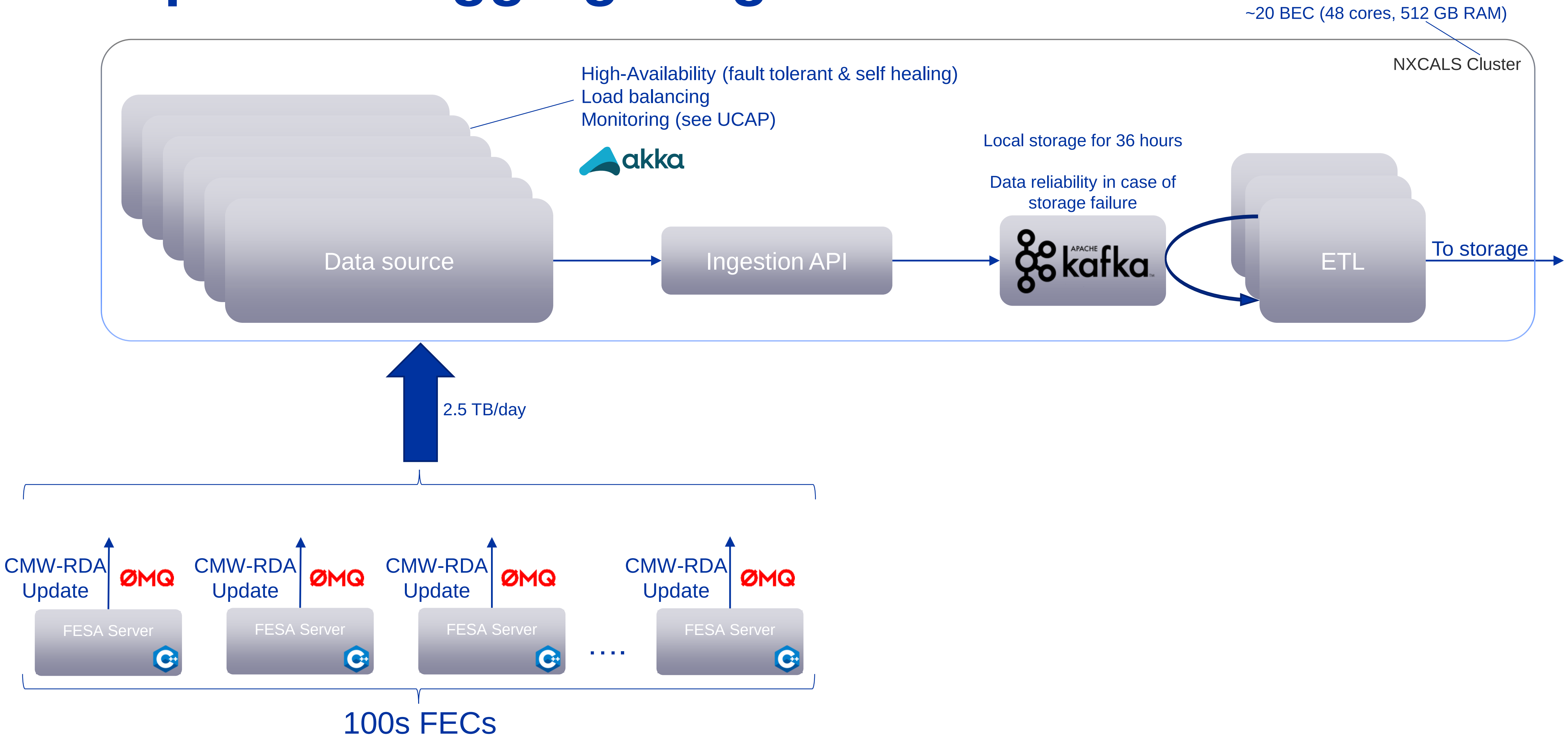
In all the examples, we assume all the configuration is already done and we focus on the run-time aspects

Example 3 – Logging

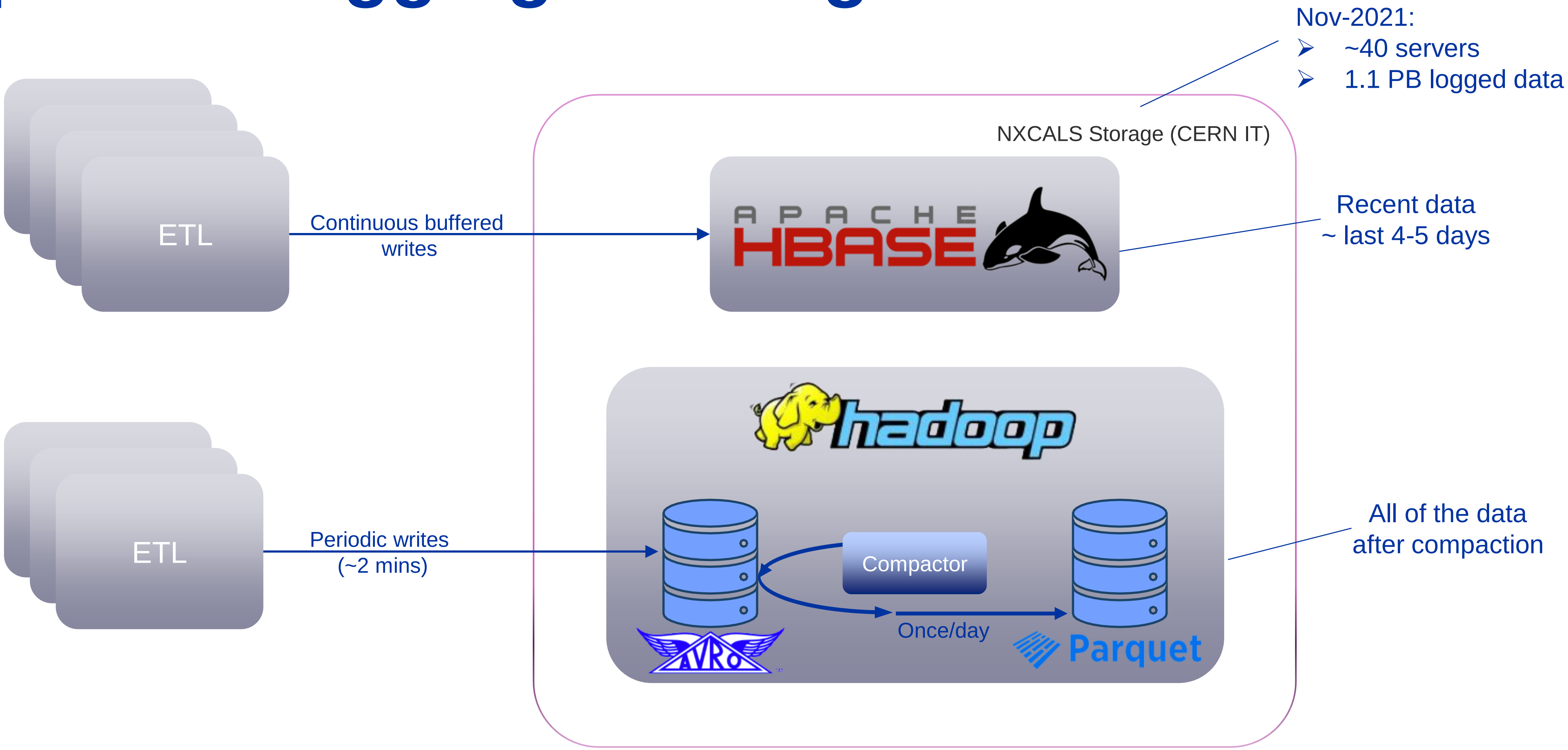


*SWAN: Service for Web-based ANalysis

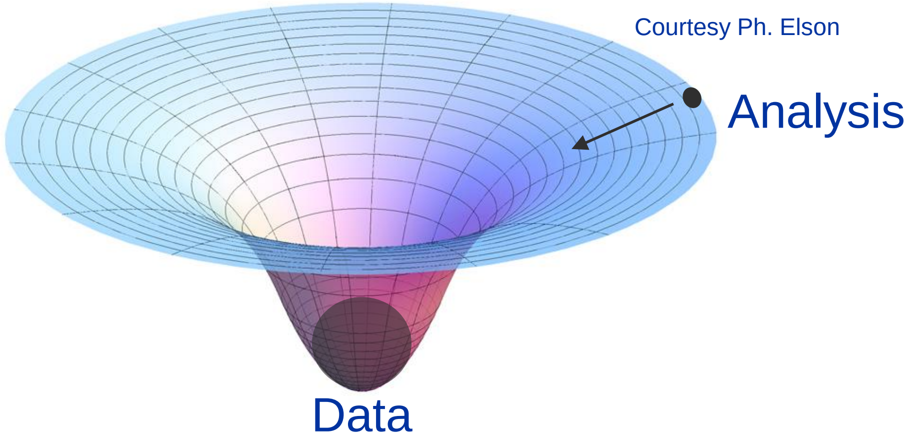
Example 3 – Logging - Ingestion

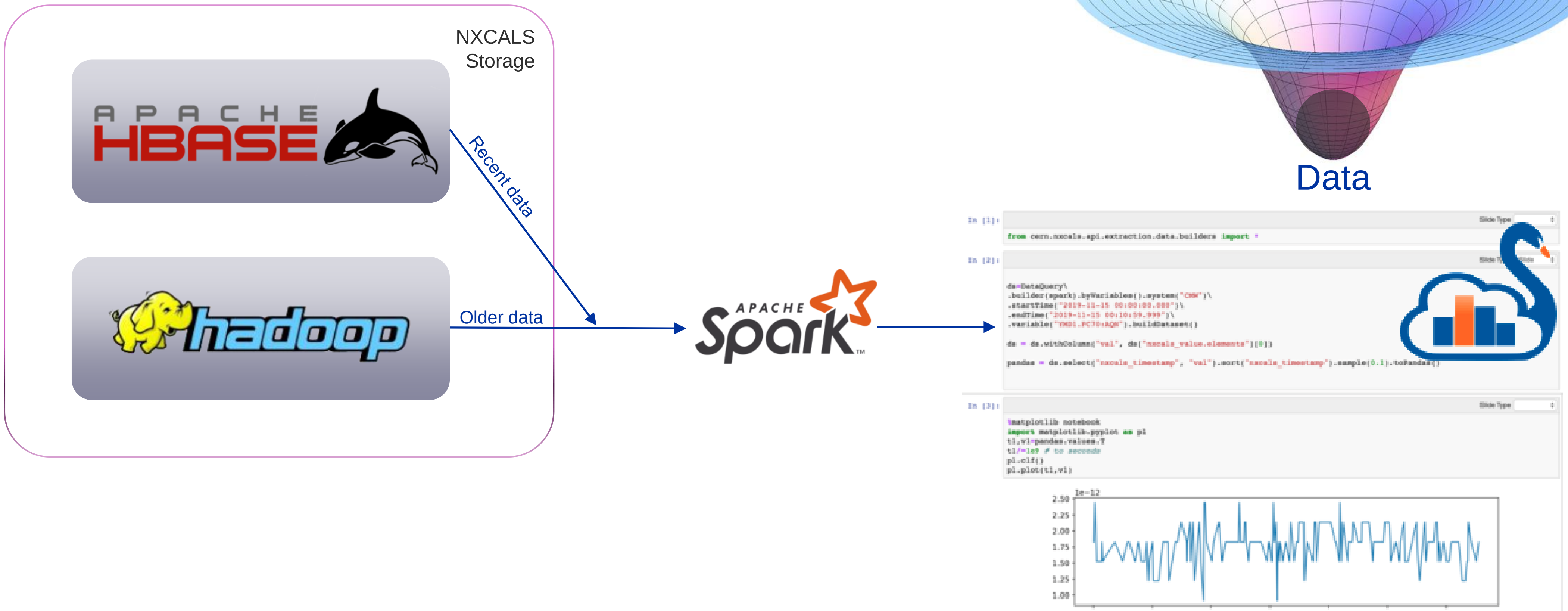


Example 3 – Logging - Storage



Example 3 – Logging - extraction





Want to know more?

CERN Beams Department (https://beams.cern/)

Introduction to BE-CO Control System, 2019 Edition, S. Deghaye & E. Fortescue, CERN, 2020. (https://cds.cern.ch/record/2748122)

Tango Controls (https://www.tango-controls.org/)

EPICS (https://epics-controls.org/)