LHC Software Architecture – Evolution Toward LHC Beam Commissioning

17/10/2007

ICALEPCS 2007 - Knoxville
Grzegorz Kruk on behalf of the LSA Team

Agenda

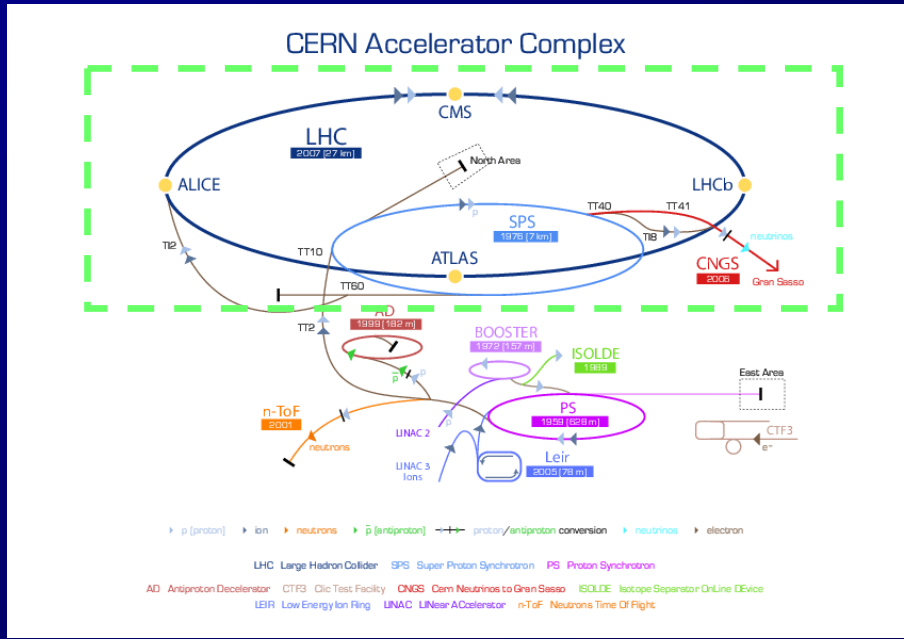
- Project essentials
- Challenges
- LSA scope & key concepts
- Implementation
- Recent developments
- Summary

Agenda

- Project essentials
- Challenges
- LSA scope & key concepts
- Implementation
- Recent developments
- Summary

Project essentials

- **Mission:**
  Provide homogenous software to operate the SPS, its transfer lines and the LHC
  - Note: high level application software



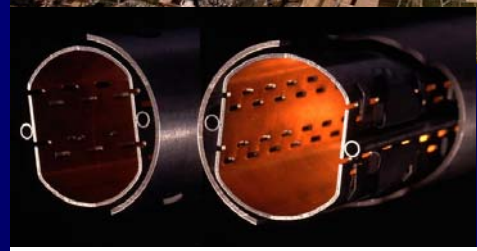
- Project shared between controls and operations
- Entirely written in Java

Agenda

- Project essentials
- Challenges
- LSA scope & key concepts
- Implementation
- Recent developments
- Summary

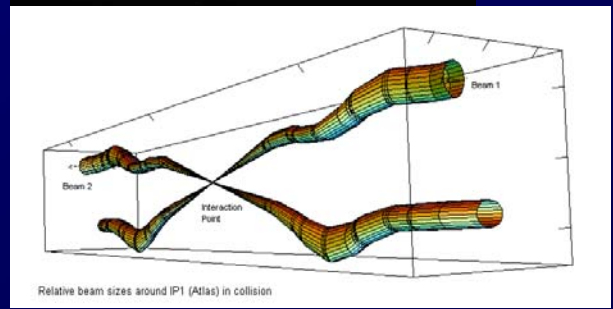
LHC Challenge
What will be LSA used for?

- Accelerate 2 beams to a very high energy around 27 km long ring



- In two pipes of a few cm diameter

- Squeeze them down to a size smaller than the diameter of a hair



- Get them collide

- Keep them colliding for at least 10 hours

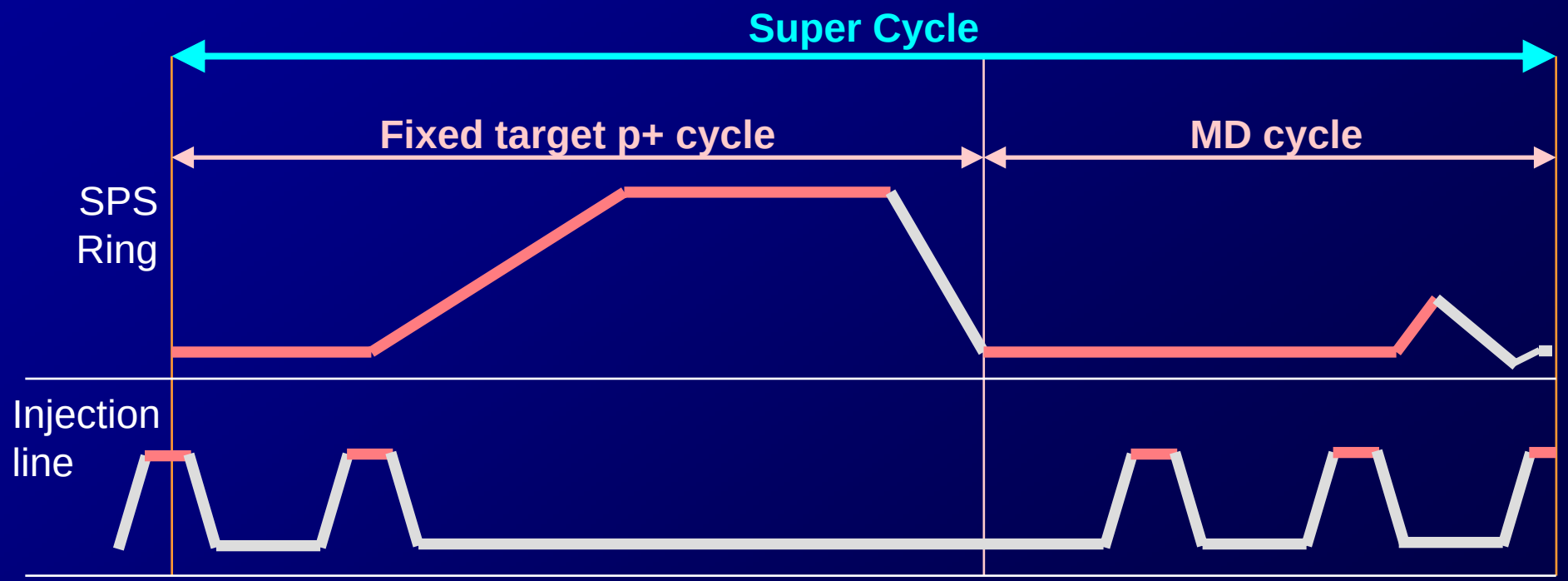
Machine diversity challenge

- Different accelerators: cycling and non-cycling
- Different hardware types and front-ends
- Different philosophy of operating these machines
- Different users (operators)

Cycling vs. non-cycling machines

SPS is a cycling machine

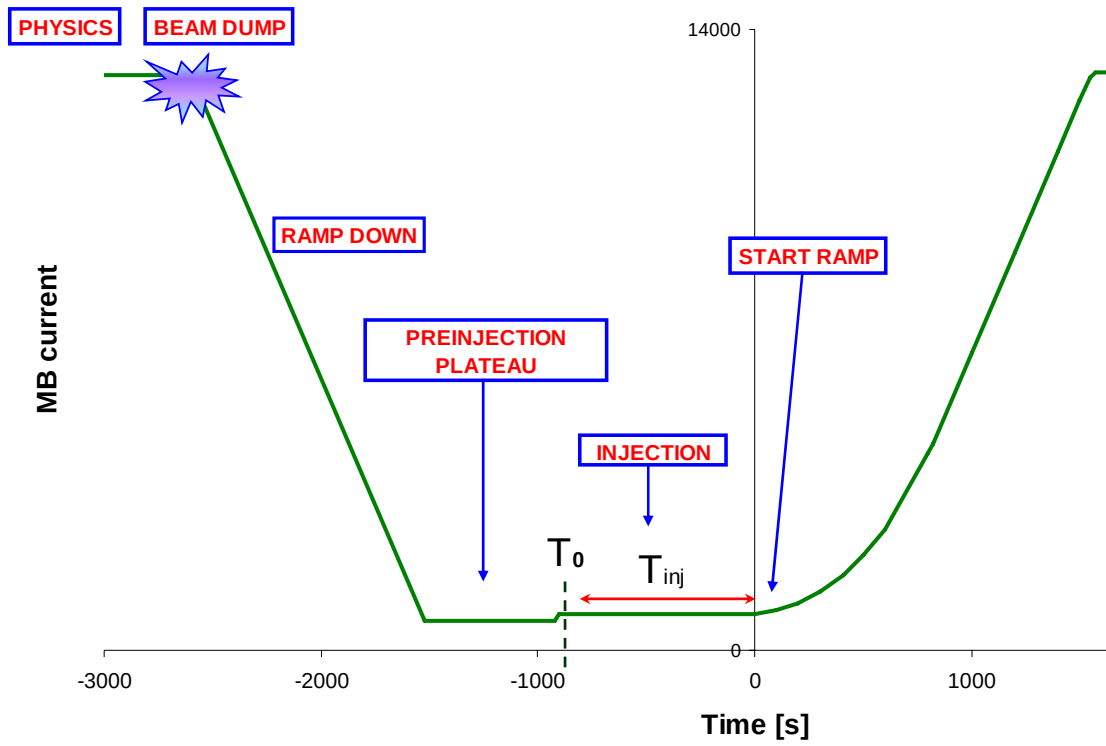
- Sequence of cycles (super cycle) is played repeatedly in a synchronous way
- Length of all cycles is fixed



Cycling vs. non-cycling machines

LHC is different – there are no cycles

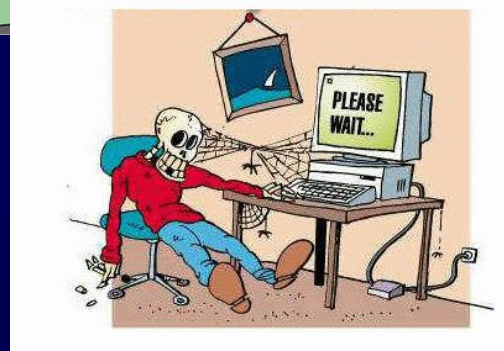
- Sequence of processes (i.e. injection, ramp, squeeze, physics) executed asynchronously
- Length of some of these processes is unknown in advance e.g. physics



Ramp down	≈ 18 Mins
Pre-Injection Plateau	15 Mins
Injection	≈ 15 Mins
Ramp	≈ 28 Mins
Squeeze	< 5 Mins
Prepare Physics	≈ 10 Mins
Physics	10 - 20 Hrs

And the system has to be..

- Reliable
  - 100% availability when there is a beam in the machine
- Secure
  - From unauthorized access
  - Against unwanted actions
- User friendly
  - Large amount of devices and data



Agenda

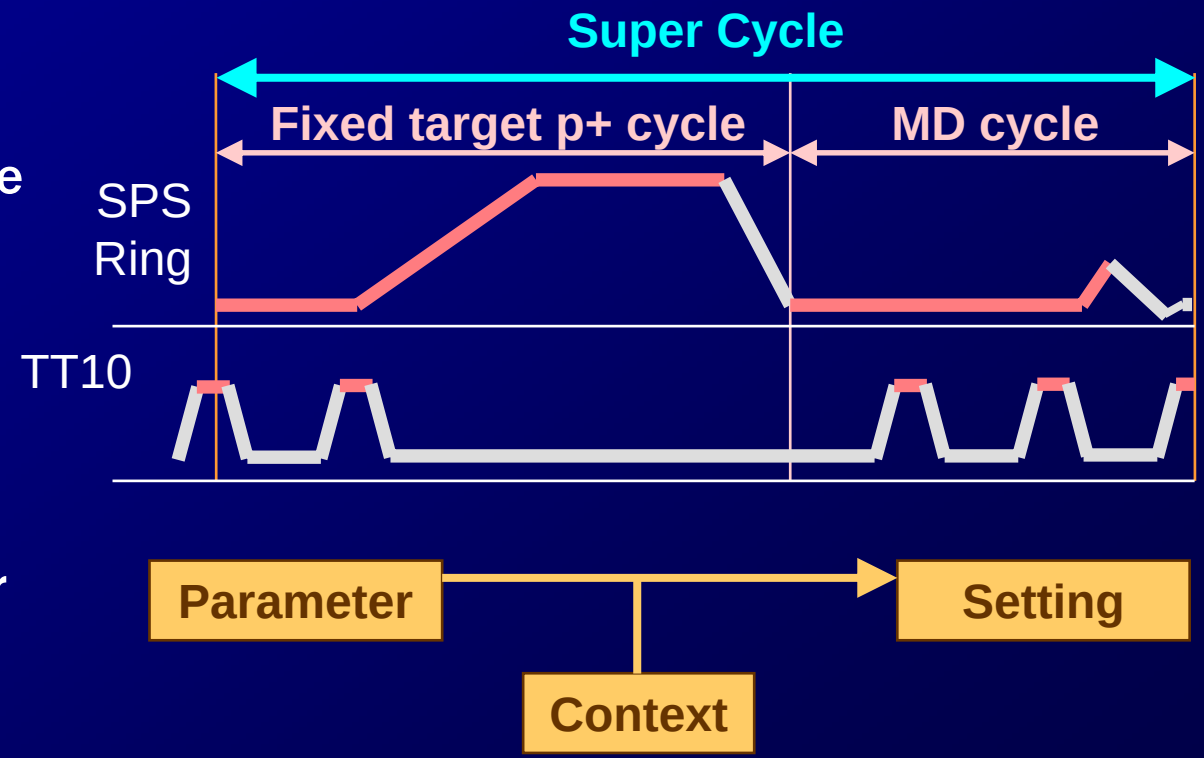
- Project essentials
- Challenges
- LSA scope & key concepts
- Implementation
- Recent developments
- Summary

What is covered by LSA?

- Optics
  - Information about all devices
  - Machine layout
  - Twiss parameters
  - ..
- Settings generation
  - Generation of initial settings based on optics
- Settings management & trim
  - Management of values for all parameters
  - Coherent modifications
  - History of changes and rollback
- Hardware exploitation
  - Equipment control
  - Sending settings to the hardware
- Equipment & beam measurements

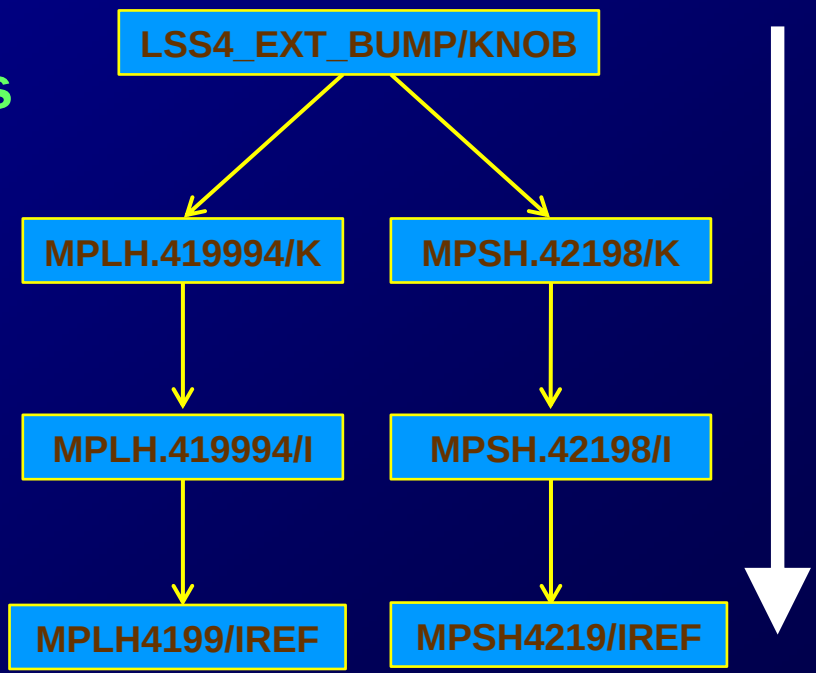
Key Concepts

- Parameter
  - Settable or measurable entity on a device (real or virtual)
  - e.g. LHCBEAM/QPH, MPLH.41994/K, MPLH4199/IREF
- Context
  - Cycle in a Super Cycle
  - Beam process (LHC)
- Setting
  - Value of a parameter for a given context

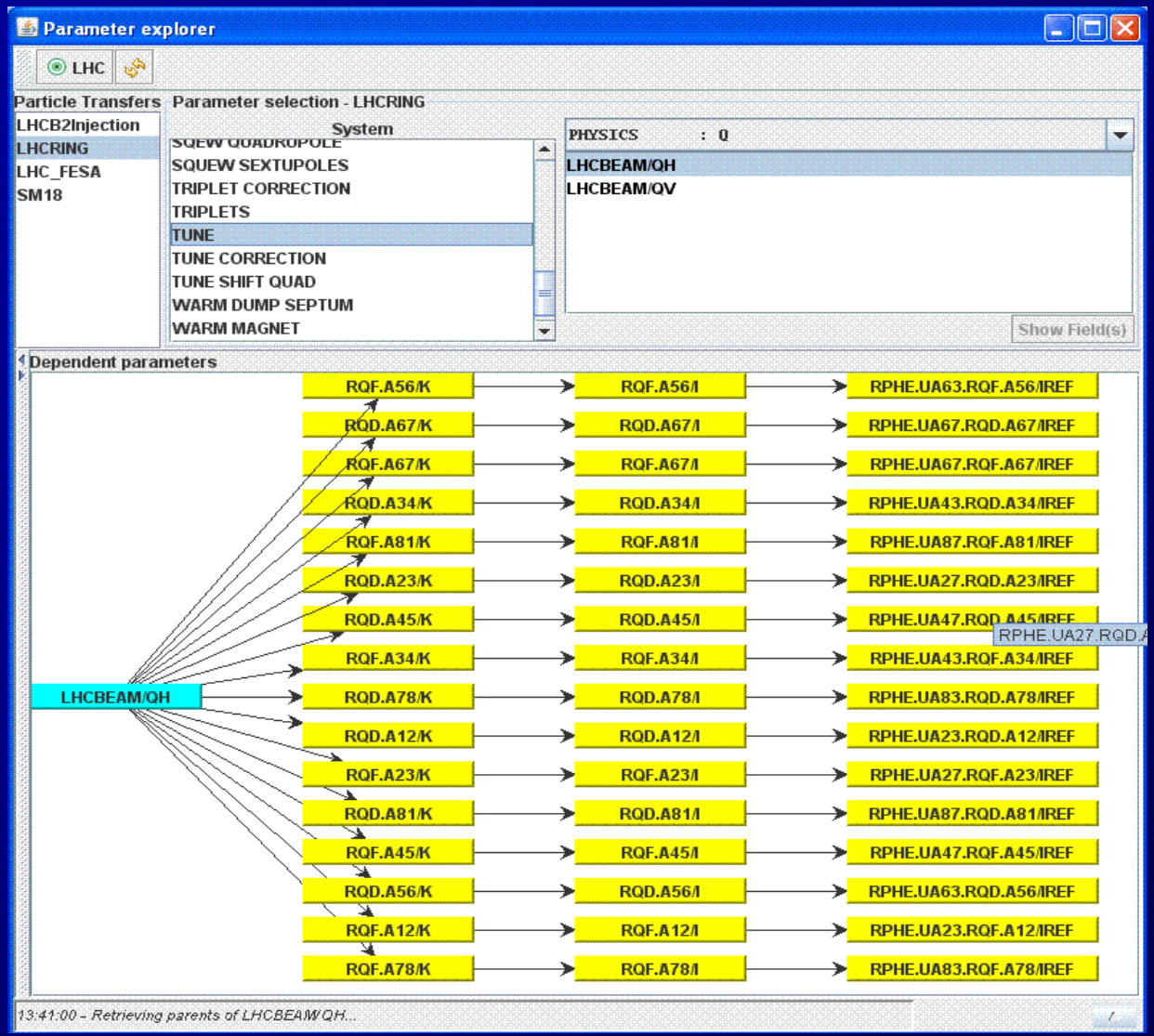


Parameters Space

- Parameters are organized in hierarchies
- Each hierarchy describes relations between parameters
  - Change of a parameter affects all its dependant parameters
- Roots → usually physics parameters
  - e.g. momentum, tune, chromaticity,...
- Leaves → hardware parameters
  - e.g. reference current on power converters



LHC Parameters Space



Domain model

- The domain model is quite complex
  - > 100 domain objects
  - ~350K lines of code
  - Currently ~40 GUI applications using services provided by LSA
- It still evolves...
- Without a good architecture it would be very difficult to handle that complexity...

Agenda

- Project essentials
- Challenges
- LSA scope & key concepts
- Implementation
- Recent developments
- Summary

Architecture - 3-tier approach

- We wanted to deploy the system in 3 physical layers due to:
  - Central access to the database and to the hardware
  - Central security
  - Caching
  - Reduced network traffic
  - Reduced load on client consoles
  - Scalability
  - Ease of web development
- With a minimal cost of 3-tier architectures
  - Complexity of programming
  - Testing & debugging
  - Deployment
- Plus we needed support for standard services
  - Transactions, remote access,...

tier, tire or tyre ??

Spring Framework

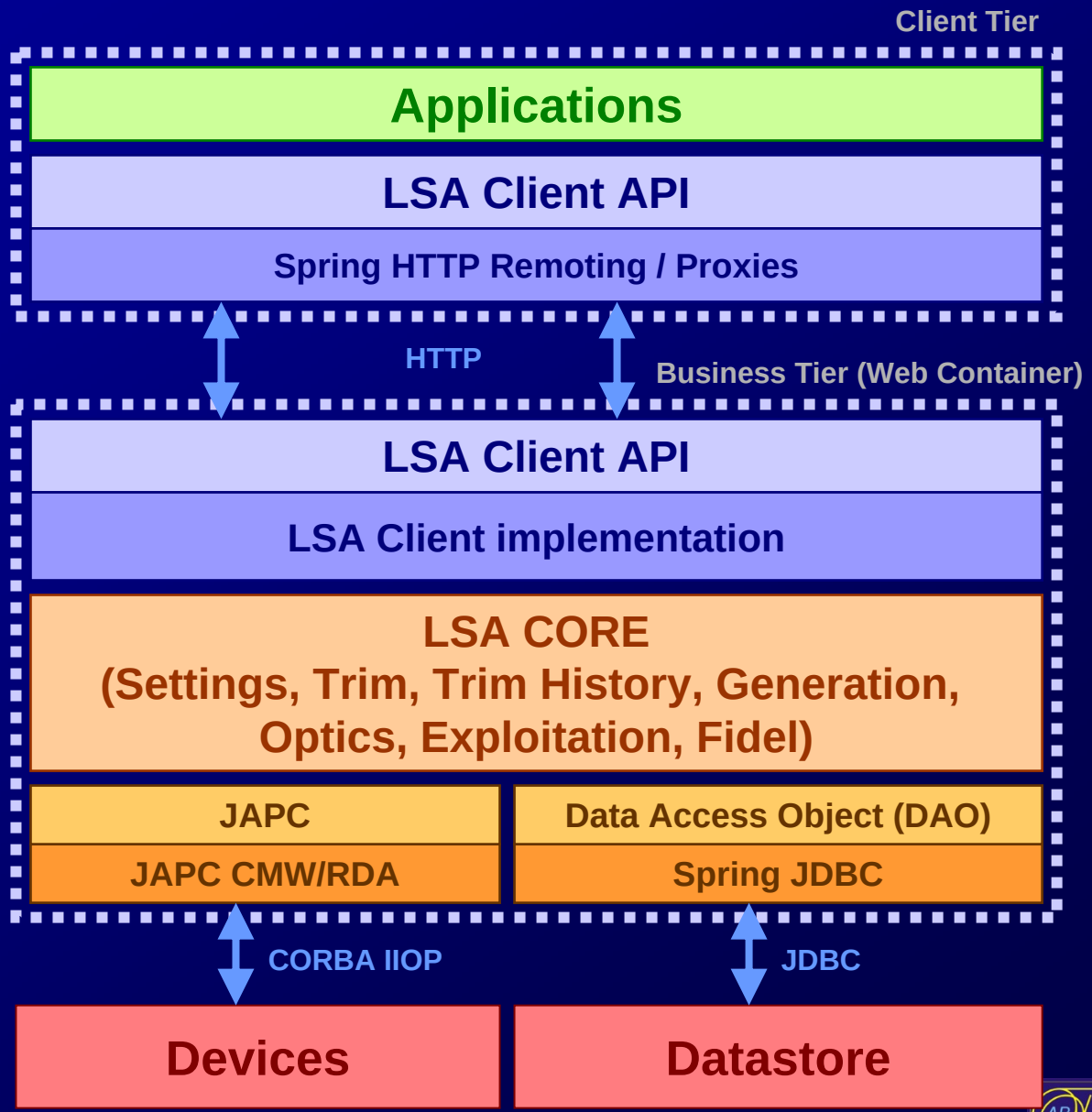
- Leading lightweight container
  - Alternative to Enterprise Java Beans (EJB)
- Plain Java Object (POJO) programming model
  - None or minimum dependency on the framework
- All standard services provided
  - Components orchestration, transactions, remoting, security, …
- Seamless deployment in 2- and 3-tier mode
- Integration with many 3rd party products
- Very little effort to maintain the infrastructure

What we use from Spring

- XML based configuration (wiring)
  - Extremely simplified in Spring 2.0
- Database access
  - Spring JDBC abstraction layer
- Transactions management
- HTTP based remoting
- Testing framework
  - Excellent to test Data Access Objects
- Caching
  - Home made mechanism
  - Based on Spring AOP – method call interception
  - Uses ehcache
  - Annotation based configuration

Architecture

- Modular
- Layered
- Distributed

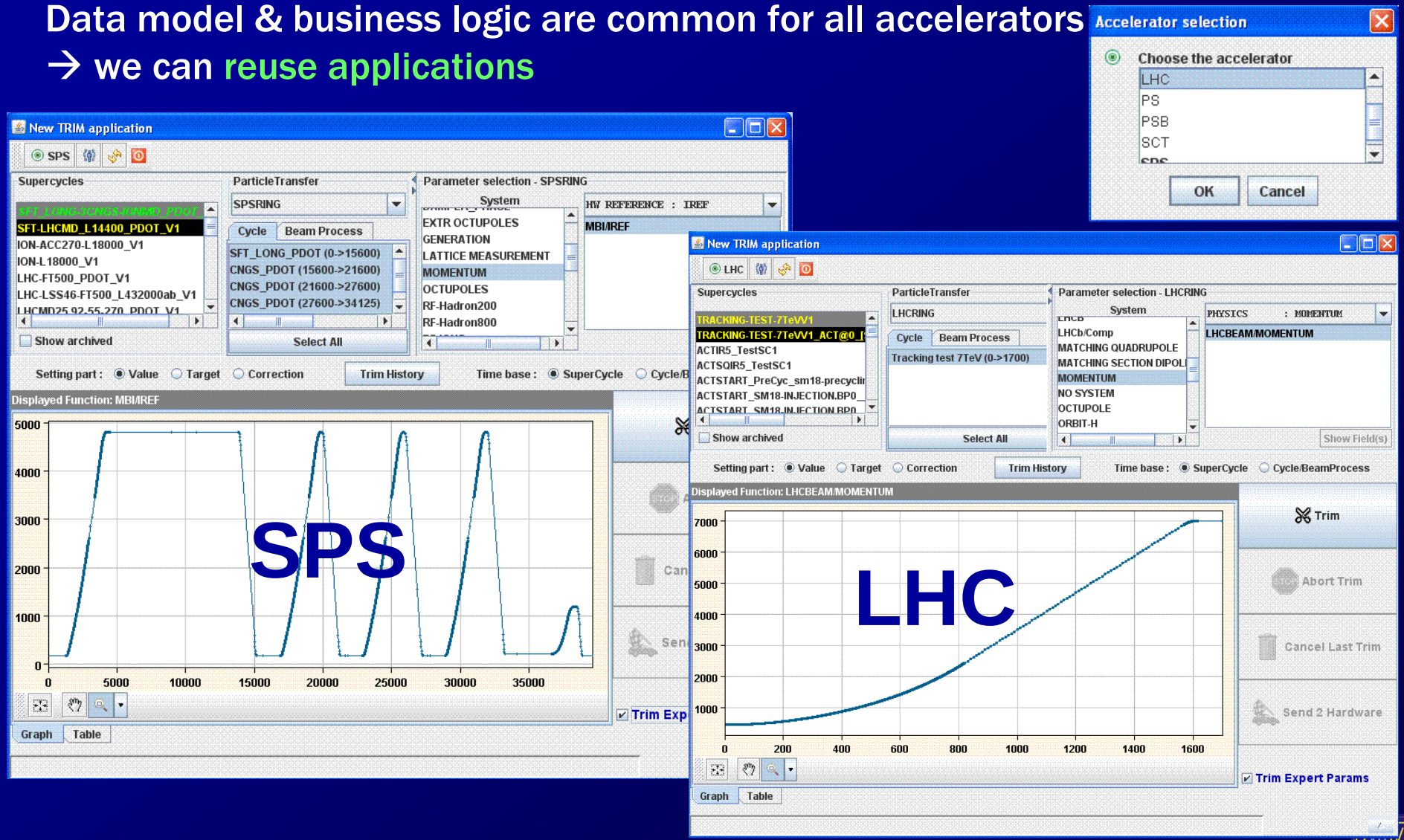


Data model

- The system is highly data-driven
- Single model (database schema) for all machines
  - SPS, LEIR, LHC,...
- Result of several iterations
- Rationalized but nevertheless quite complex
  - ~170 tables

Generic Applications

Data model & business logic are common for all accelerators
→ we can reuse applications

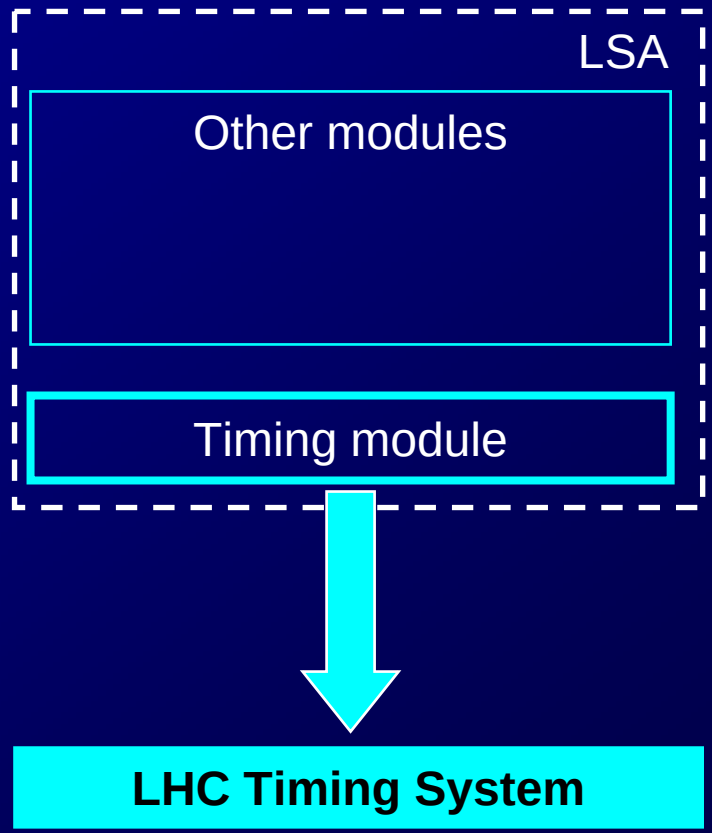


Agenda

- Project essentials
- Challenges
- LSA scope & key concepts
- Implementation
- Recent developments
- Summary

LHC Timing

- All LHC processes (e.g. injection, ramp,...) will be synchronized and triggered using timing events
  - Sent by the LHC Timing System [*see J.Lewis' talk on Friday*]
- LSA provides service to manage these events
  - Creation, modification
  - Loading to and unloading from the Timing System



LHC is dangerous

Airbus A380 at 700 kph

Energy stored in the LHC magnet system[1]

LHC Beam energy[2]

Aircraft carrier at 11 knots

We have to be very careful…

- **RBAC**
- **MCS**
- **MAD**

(1) R.Schmidt „Status of the LHC accelerator“, November 2005
(2) M.Lamont, “LHC Collimators review”, 30 June 2004

Role Based Access Control

- Created in the frame of the LHC at FermiLab Software (LAFS) collaboration

- Helps to protect:
  - Against unauthorized access to the equipment i.e. sending settings by inappropriate people
  - From doing bad things at bad time
  - Functionality reserved for specified groups of users (experts)
- When sending settings to the hardware - verification of credentials is done by the middleware (CMW)
- Seamless integration in LSA thanks to the Spring remoting
  - No need to modify the API

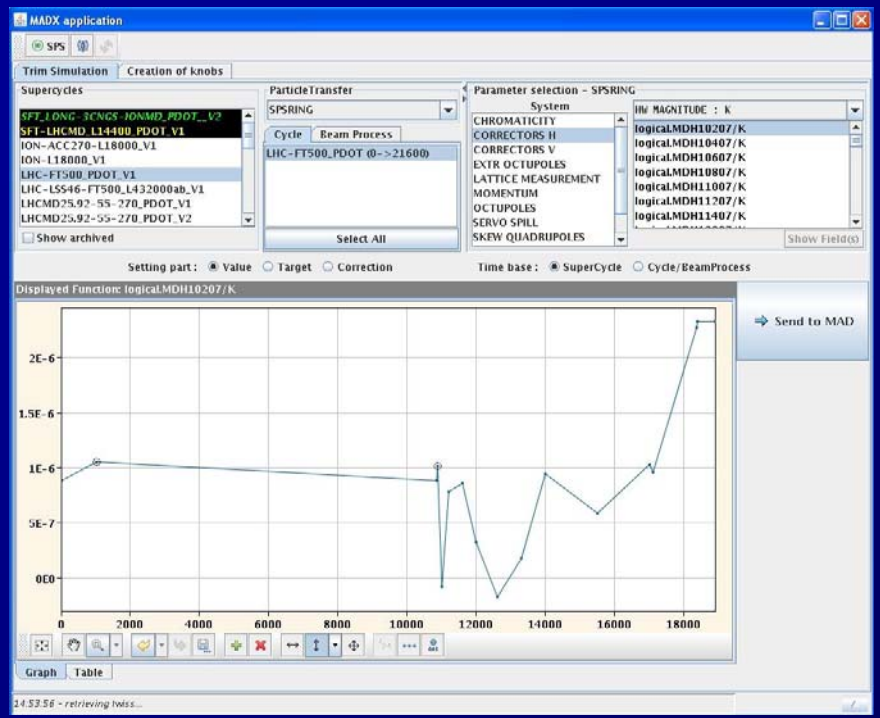
Management of Machine Critical Settings (MCS)

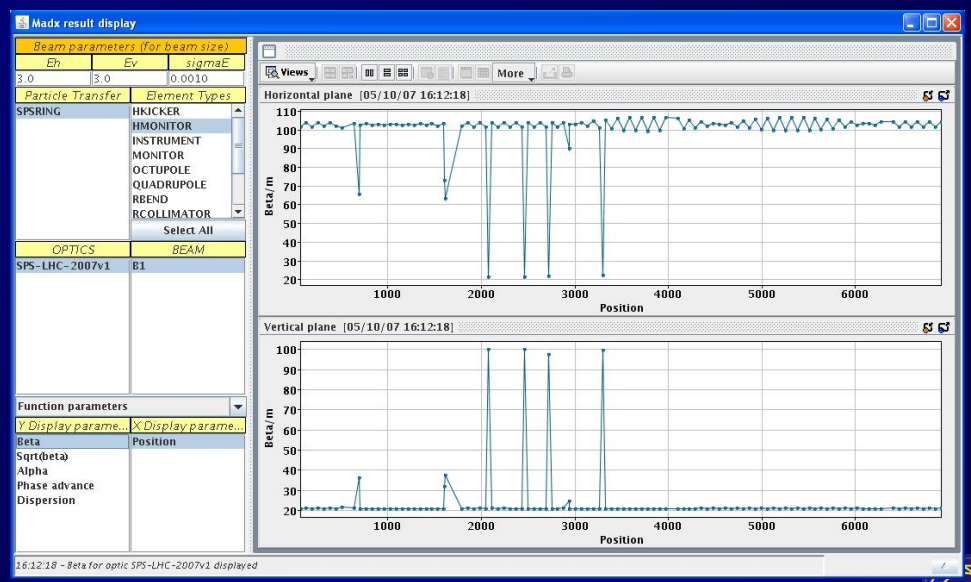
- Aimed for the most critical and potentially dangerous devices/settings
- Complementary to the RBAC
  - Second layer of security
- Based on a digital signature scheme
  - To ensure data integrity
- Verified on the front-end level (FESA framework – *next talk*)

V. Kain "Management of Critical Settings and Parameters for LHC Machine Protection Equipment", Functional Specification, CERN, 2006

Methodical Accelerator Design (MAD)

- **Modeling and simulation tool** for particle accelerators and beam lines
- Simulation of settings changes before applying them to the hardware





Agenda

- Project essentials
- Challenges
- LSA scope & key concepts
- Implementation
- Recent developments
- Summary

Summary

- Stable and reliable architecture
  - ✓ We can concentrate on the domain which is complex enough
- For all accelerators we have the same:
  - Data model
  - Core logic
  - Applications

Good data model → Commonalities

- System used currently for the SPS, its transfer lines, the LEIR and the LHC hardware commissioning
- Crucial functionality for the LHC in place