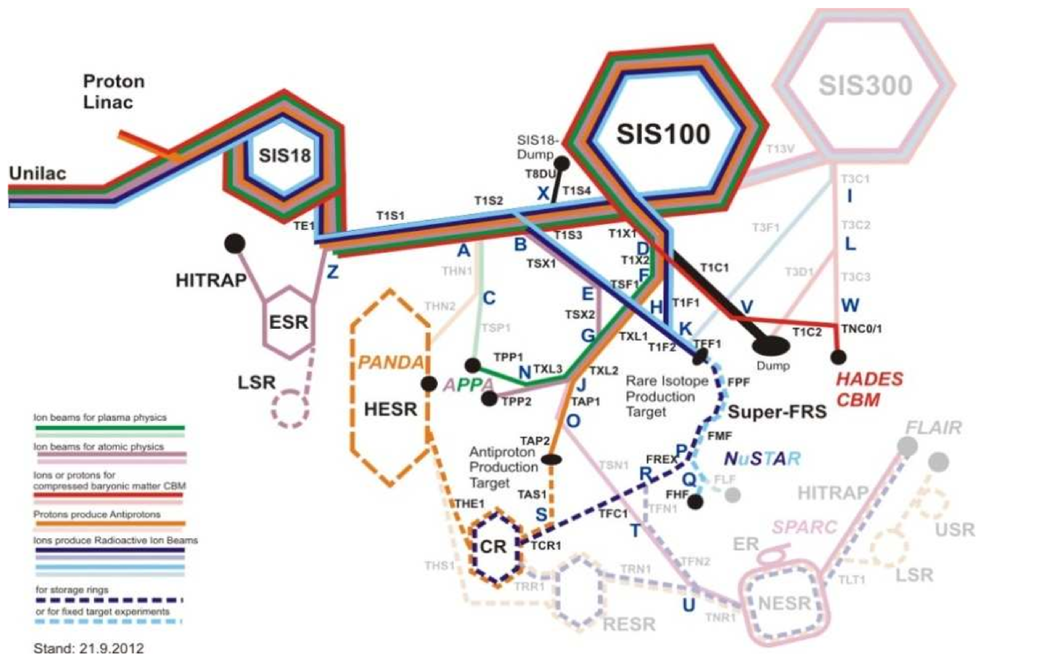
The FAIR Control System - System Architecture and First Implementations

Ralf Huhmann, Ralph C. Bär, Dietrich Hans Beck, Jutta Fitzek,
Günther Fröhlich, Ludwig Hechler, Udo Krause, Matthias Thieme
GSI Helmholtz Centre for Heavy Ion Research, Darmstadt,Germany

Abstract

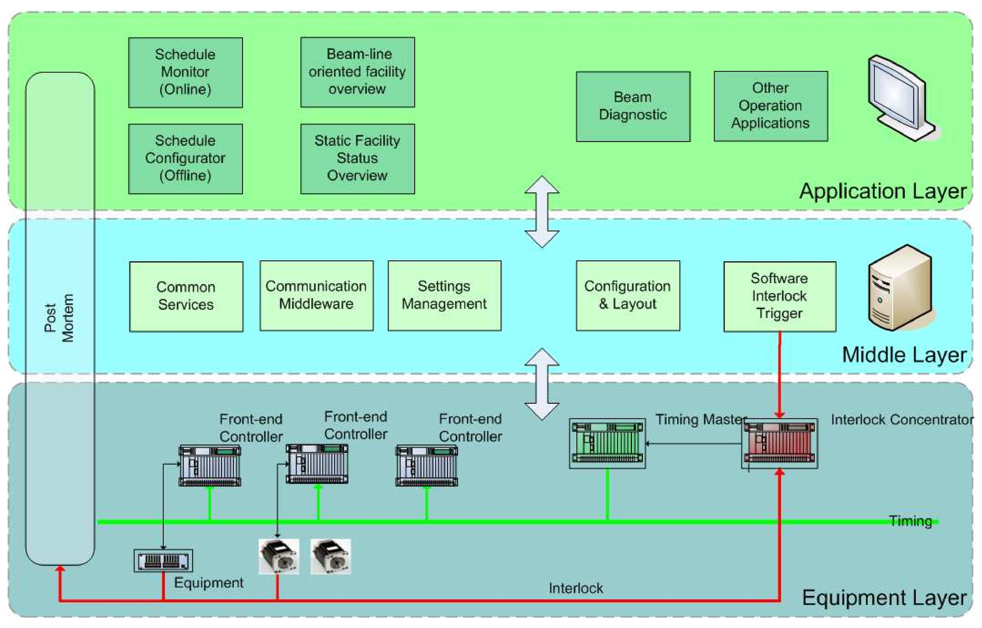
The paper presents the architecture of the control system for the Facility for Antiproton and Ion Research (FAIR) currently under development. The FAIR control system comprises the full electronics, hardware, and software to control, commission, and operate the FAIR accelerator complex for multiplexed beams. It takes advantage of collaborations with CERN in using proven framework solutions like FESA, LSA, White Rabbit, etc. The equipment layer consists of equipment interfaces, embedded system controllers, and software representations of the equipment (FESA). A dedicated real time network based on White Rabbit is used to synchronize and trigger actions on equipment level. The middle layer provides service functionality both to the equipment layer and the application layer through the TCP/IP control system network. LSA is used for settings management. The application layer combines the applications for operators as GUI applications or command line tools typically written in Java. For validation of concepts already in 2014 FAIR's proton injector at CEA/France and CRYRING at GSI will be commissioned with reduced functionality of the proposed FAIR control system stack.

System Overview



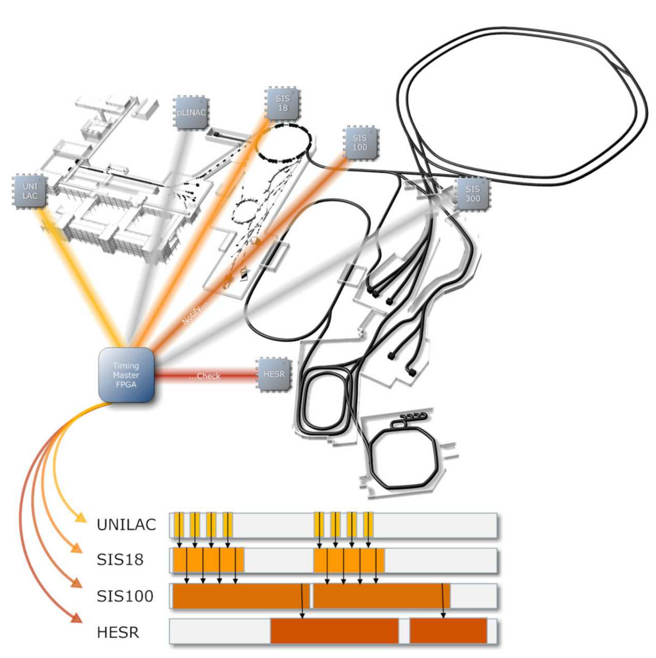
FAIR parallel beam operation © P. Schütt, GSI, Darmstadt.

- Research areas: Nuclear Structure, Astrophysics, Antiproton Physics, High Energy Nuclear Physics, Atomic and Plasma Physics.
- ≈ 4000 devices producing heavy ion and proton beams
- Parallel beam production for serving up to 4 experiments simultaneously
- Multiplexing of upstream accelerators by switching between predefined settings
- Dedicated Timing System based on White Rabbit network for synchronized device action and coherent data settings and data acquistion



Control System Overview

Timing System



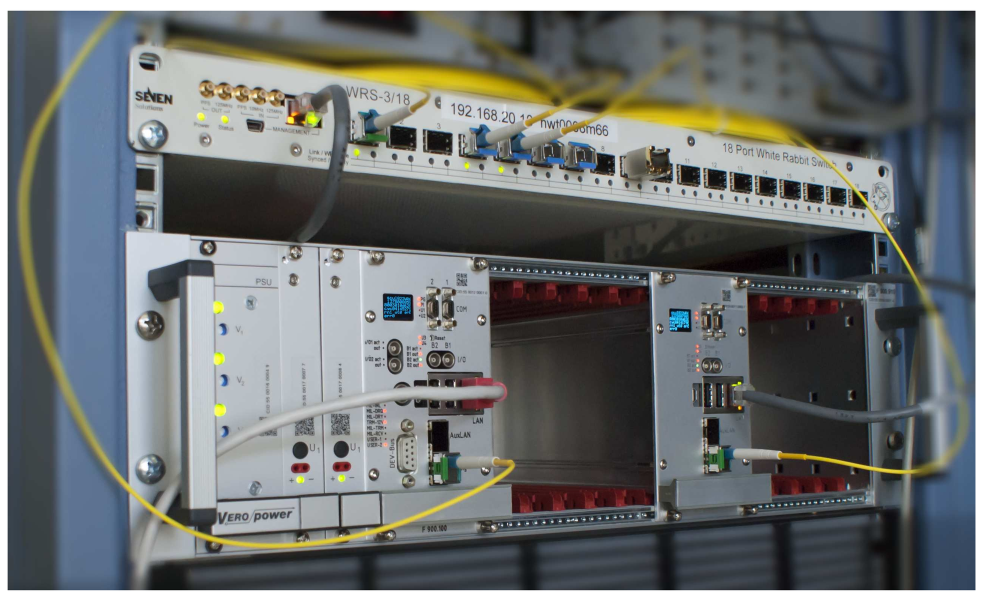
Timing schedule ©M. Kreider, GSI, Darmstadt.

- Separated timing network, central timing master (TM), facility wide (≈ 2000) distributed timing receivers
- By using White Rabbit an absolute time synchronization in nanosecond range, jitter in low picoseconds range is achieved
- TM dispatches timing event messages in advance which carry absolute execution time, event type, and data index.

When executed on timing receiver site, event data is used to select the pre-supplied settings for the equipment.

- FESA real time actions (precision $\approx 100\mu s$), direct equipment control via FPGA soft CPU (precision $\approx 1\mu s$), and digital signal generation for nanosecond precision.

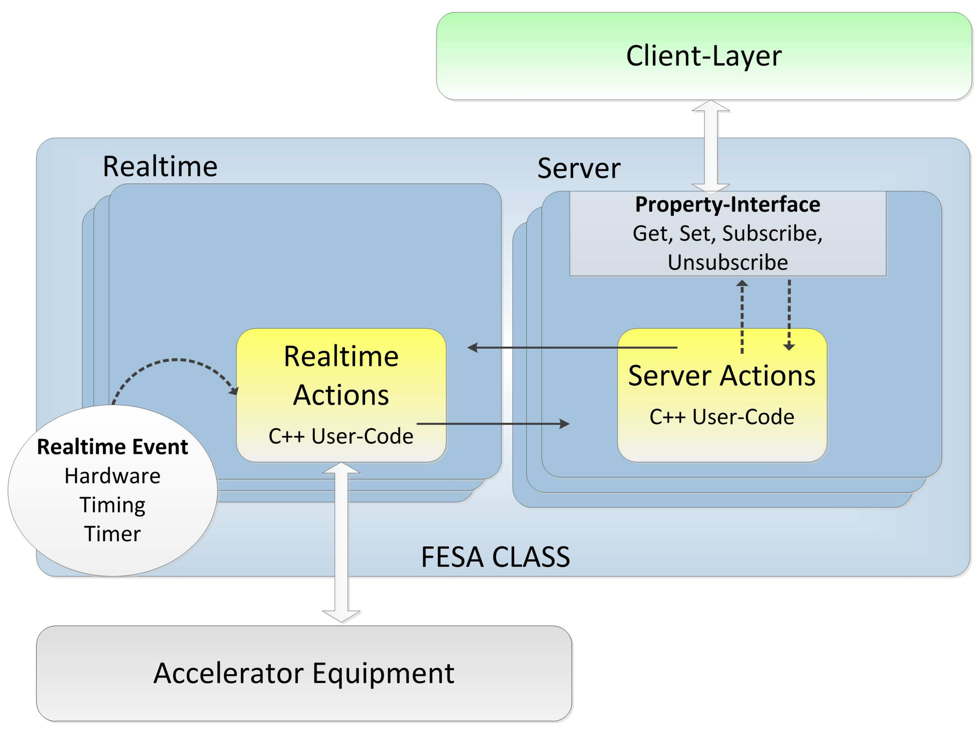
Equipment Controller (SCU)



Scalable Controller Unit (SCU) and White Rabbit Switch

- SCU combines a x86 COM Express Board and ™Altera Arria II FPGA
- Proprietary parallel bus interface (SCU bus) with up to 12 slave boards which connect to equipment
- FPGA hosts the SCU bus master and a Timing Receiver
- x86 board hosts Linux with real time patch to run FESA classes

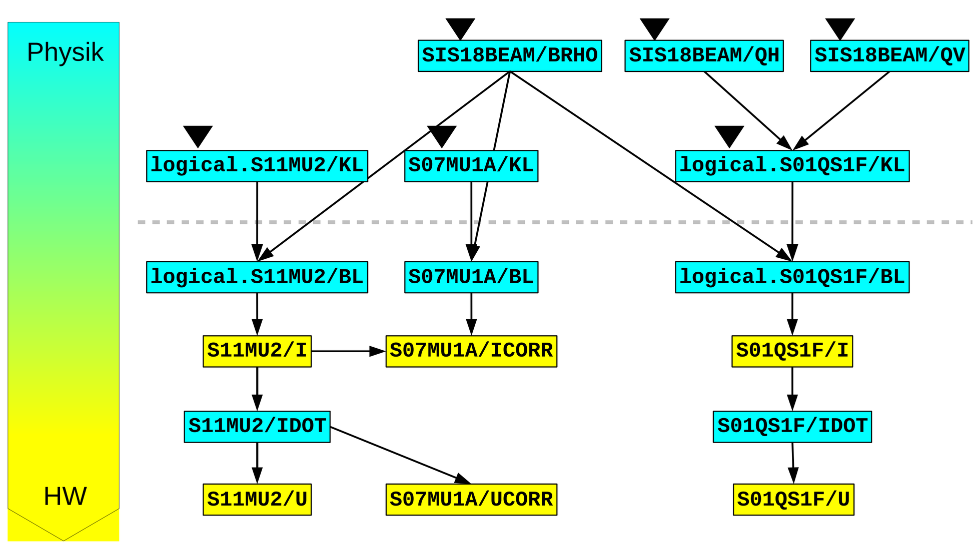
Frontend Software (FESA)



FESA Class

- FESA is a framework developed by CERN to build frontend controller software
- An Eclipse plugin models abstract device objects with process variables presented as properties; the device model and user specific code (Realtime and Server part) builds the FESA class; toolchain links FESA core libraries to one or more FESA classes and builds x86 executable for Linux
- At runtime the server part provides an uniform interface to upper layers via the object-property model
- Multiplexed settings for time multiplexed operation
- At runtime the real time actions are triggered e.g. by timing events, the actual beam specific setting data is selected based on information supplied by the timing event and sent to the equipment

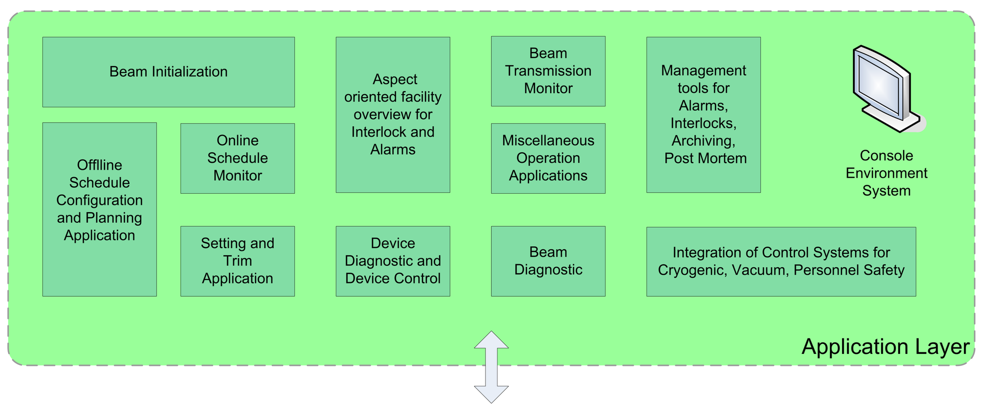
Settings Management (LSA)



LSA parameter hierarchy

- LSA is a framework developed by CERN
- Based on a physics model for accelerator optics (twiss, layout) and relations between parameters and between accelerators, generation rules for parameters are coded in Java
- LSA-server provides offline generation of machine settings, sending these settings to all involved devices, and programming the schedule of the timing master
- Generation rules create timing constraints (e.g. ramp-curve) and equipment's data settings (e.g. field) for all devices from adjustable physics parameters (e. g. beam energy)
- For FAIR the framework is extended to model the overall schedule of all accelerators

Operation Software

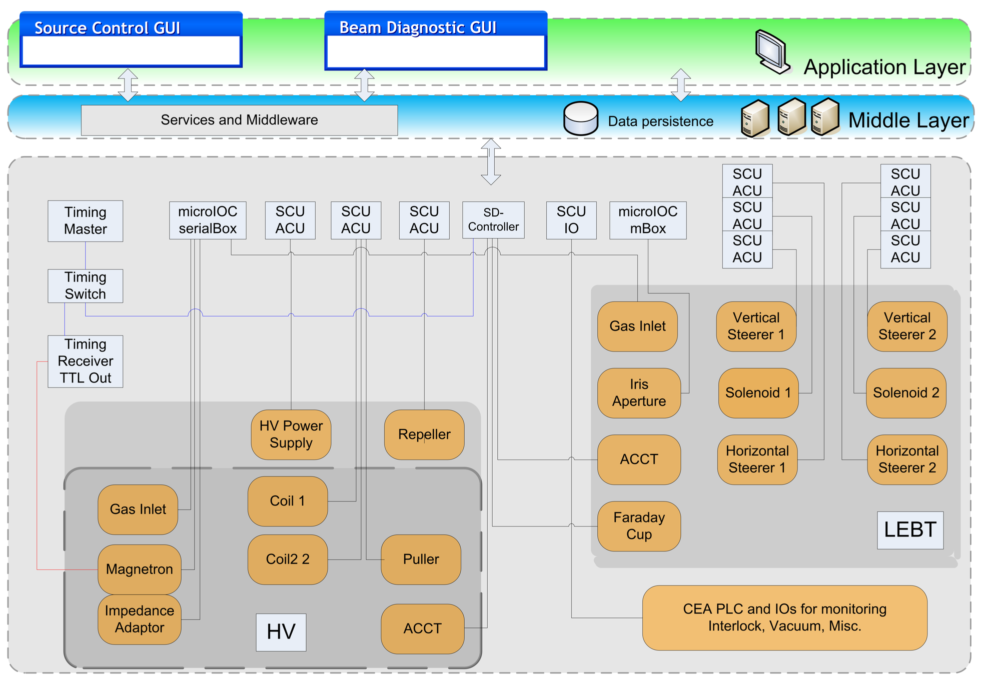


Functional Blocks of Operation Software

- Applications provide an integrated working environment
- Setting, optimizing, monitoring for parallel beam operation
- Generic applications to adapt to beam production chains, accelerator areas, and operation modes
- LSA Client API and JAPC for device communication

First Implementations

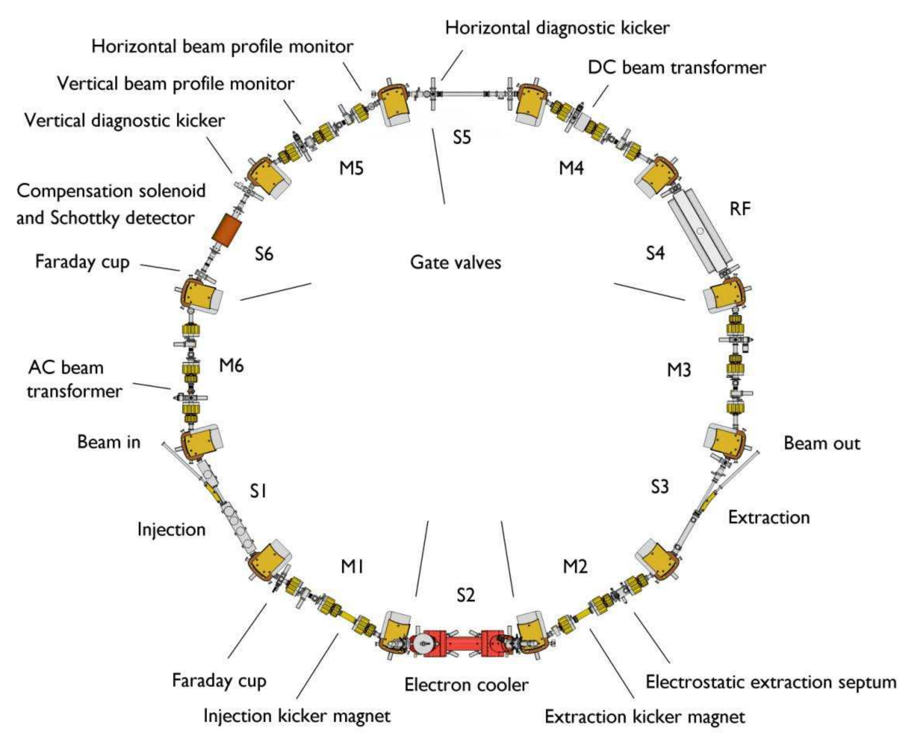
Proton Injector at CEA/Saclay (France)



Overview of Mini Control System for FAIR's Proton Injector (CEA/Saclay)

- Proton Injector is commissioned in 2014 at CEA/Saclay
- FAIR standard solution for power supplies: SCU with proprietary slave boards
- All injector equipment is modeled as FESA devices
- FAIR timing system for magnetron's pulsed mode

CRYRING at GSI



CRYRING at GSI

- CRYRING is Swedish inkind contribution to FAIR
- Legacy standalone low energy experimental storage ring of Manne Siegbahn Laboratory (MSL)
- Transferred to GSI/FAIR in 2013, injection from ESR, commissioning is scheduled for 2014
- FAIR control system stack: SCU, Timing System, FESA, LSA
- CRYRING is used as test bed for hardware and software as well as new concepts for multiplexed beam operation