



FESA and LSA at GSI

Jutta Fitzek

Agenda

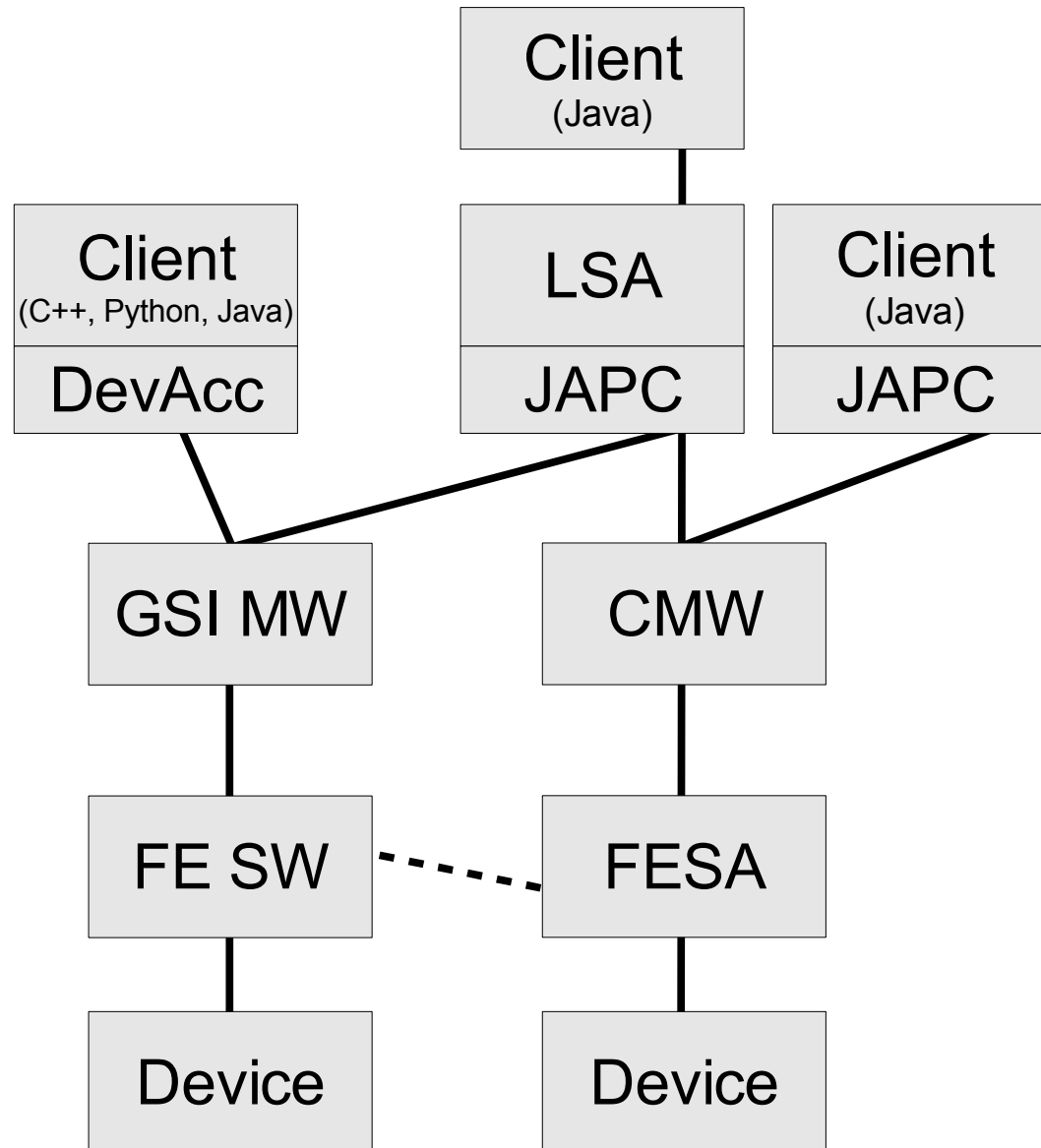
Current Software Stack at GSI

What is FESA? (framework at frontend level)

What is LSA? (framework for settings management)

Architecture

GSI – different ways of accessing devices



Agenda

Current Software Stack at GSI

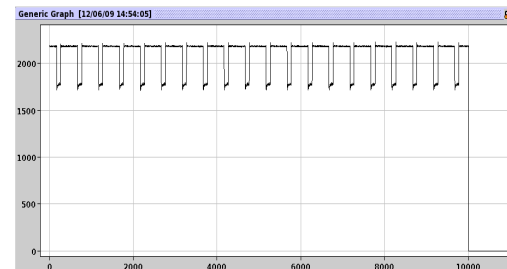
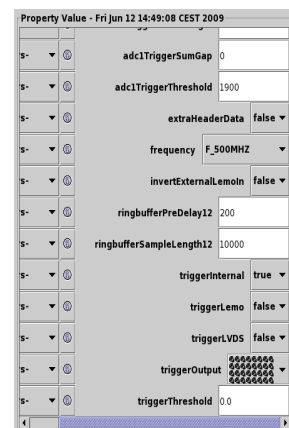
What is FESA? (framework at frontend level)

What is LSA? (framework for settings management)

What is FESA?

K. Höppner - © GSI 2009

- core component of the CERN control system at Frontend level
- integrates realtime and non-RT actions on one CPU
- easy design and deployment of FESA device classes with FESA shell and code generation tools
- supports multiplexed operation of accelerators
- well developed framework, broad equipment support
- supports Intel and PowerPC architecture, easy integration of upcoming FAIR controller assumed



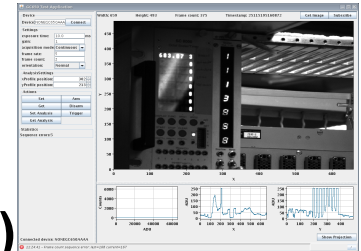
H. Bräuning - © GSI 2009

FESA at GSI



- **FESA 2.10**

- **prototype installation, several SD use cases running (BPM, Ablass 2, Prosilica CCD Camera, Readout of a digitizing VME ADC, first version for PLCs)**
- **enhanced by the beam diagnostics group: GSI timing converter**
- **FESA class development is done using linux machines from the blade cluster (developing 32bit FESA classes on a 64bit machine)**



- **FESA 3.0**

- **major refactoring of the FESA framework**
- **enhanced class modelling features (aggregation, composition)**
- **currently under development, expected to be ready spring 2010**
- **modular concept, support of “lab packages” (→ better integration)**
- **the FESA version to be supported for production usage at GSI**
- **development of FESA classes will also take place on the linux cluster**
- **support of a “FESA development package” planned**

Agenda

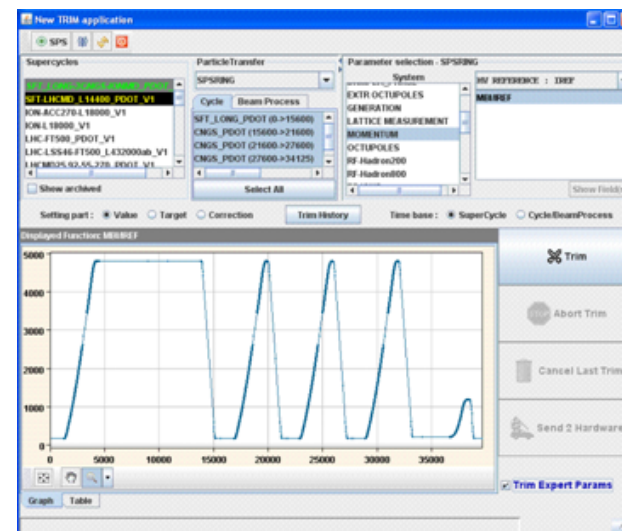
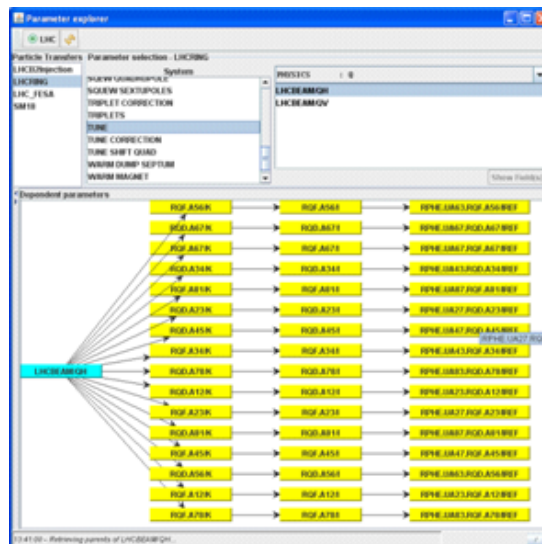
Current Software Stack at GSI

What is **FESA**? (framework at frontend level)

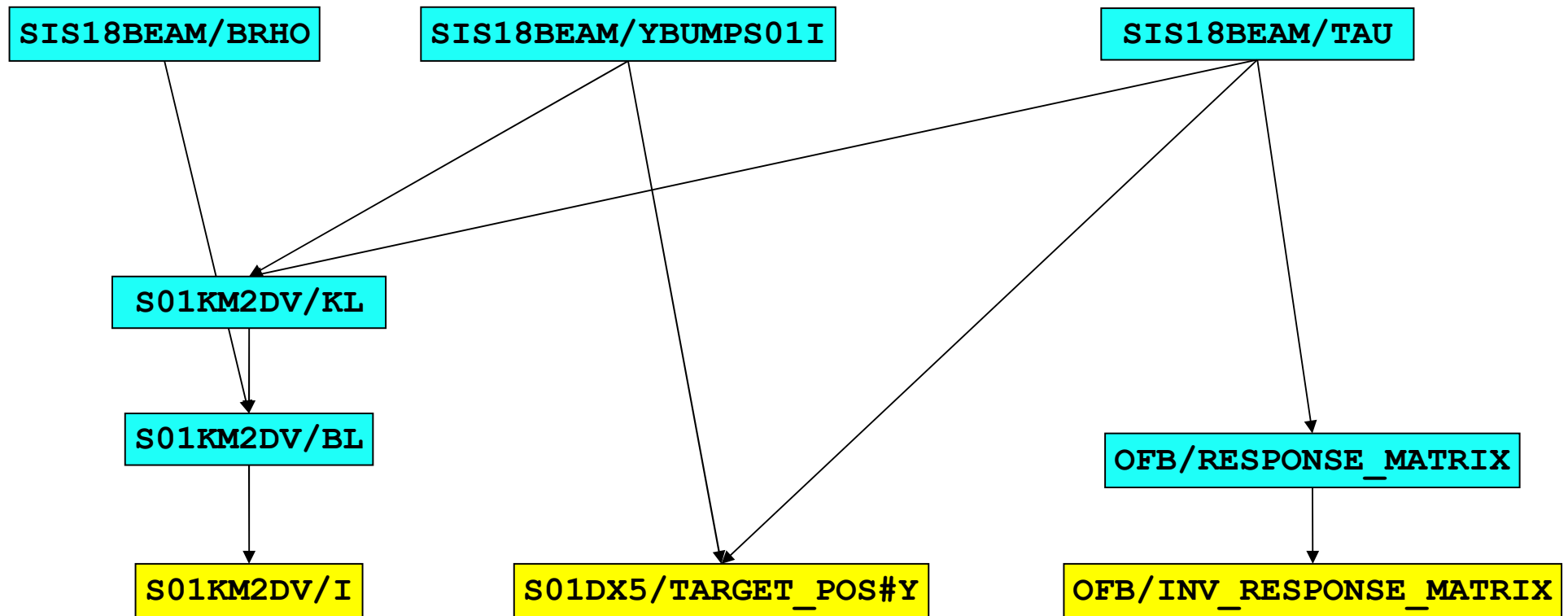
What is LSA? (framework for settings management)

What is LSA?

- LSA – core component of the CERN control system at operating level
 - highly data driven, DB is the master and contains all needed information about optics, devices, cycles etc.
 - ONE DB for all accelerators
 - parameters are organized in hierarchies (from physics to hardware)
 - consistent settings generation and management for all levels
 - devices are accessed using an abstraction layer called JAPC, that hides the middleware
 - with LSA, machines are operated on physics level in a consistent way



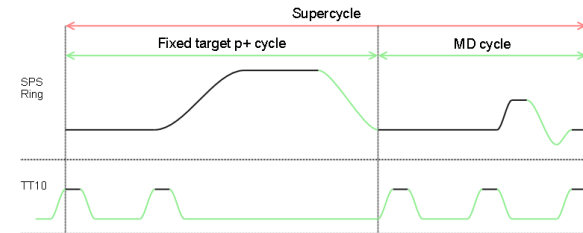
LSA: example BPM hierarchy



LSA: “Context” and “Setting”

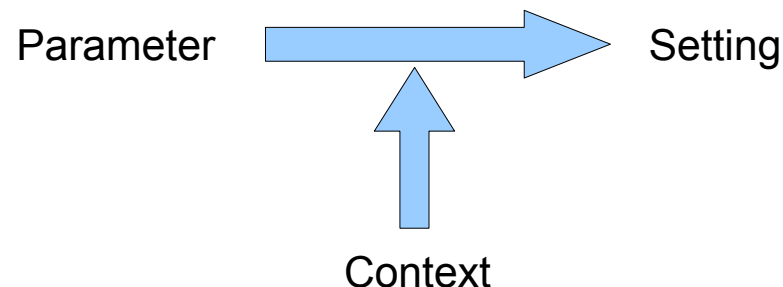
Context in LSA represents something, that can be executed in the machine, i.e. BeamProcess, Cycle, SuperCycle.

Contexts can be part of each other, e.g. a BeamProcess in a Cycle.



LHC Software Architecture - G.Kruk - © CERN 2007

Setting is a scalar/function for a parameter depending on a context



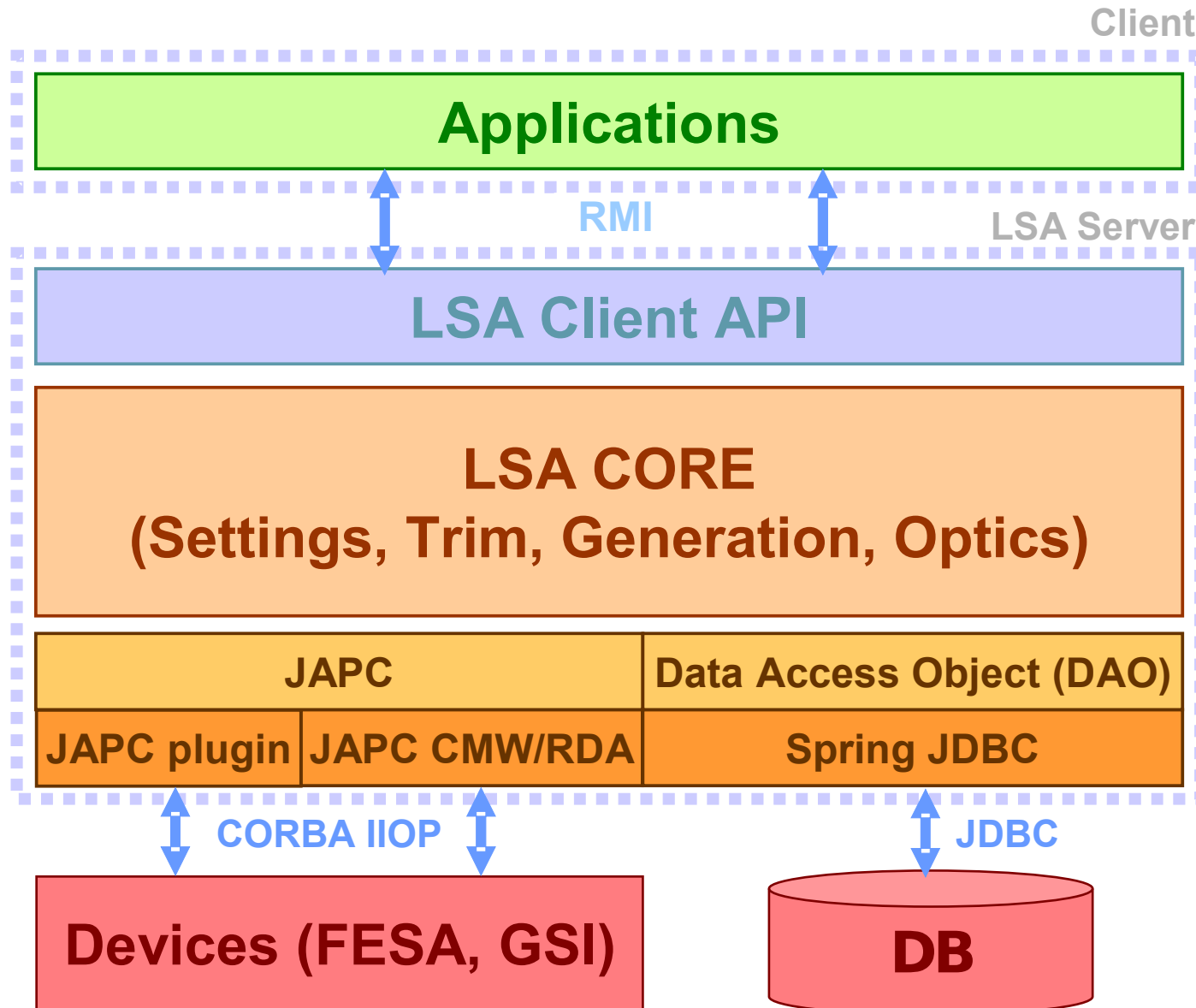
A setting consists of target and correction value

- Target values are usually calculated using the accelerator model
- Correction values are applied by operators or calculated by applications

Settings are stored for ALL levels of the parameter hierarchy

Architecture

LSA



LSA: Benefits

- **Supporting realistic model of the world:
trimming and measuring on those levels that are relevant for physics**
- **Big and complex machines, resulting in complex parameter space:
Mapping to few high level parameters makes it manageable**
 - **CERN Collimators: trimming one Nsigma, 96 functions are calculated**
- **The LSA Trim mechanism allows:**
 - **coherent modification of settings**
 - **propagation down the parameter hierarchy using MakeRules**
 - **trims are archived and can be reverted**
- **Totally data driven, same model for all accelerators**

LSA @ GSI

- LSA is at the moment used in a prototype state to model the SIS18
- LSA is being enhanced in collaboration with CERN, to make some parts of it more flexible
- Will be the central part for settings management within the future control system

