H2LC
The Hitchhiker's guide to LSA Core

Rule #1 Don't panic

Introduction

- **LSA control system helps OP managing accelerators settings.**
  - But it is also providing advanced features and interfaces to ease operation
- **Some features are included from the beginning, others are still being developed.**

References

- **LSA Project Homepage**
  - http://ab-project-lsa.web.cern.ch/ab-project-lsa/
- **LSA work in progress for LHC**
  - http://proj-lhc-software-analysis.web.cern.ch/proj-lhc-software-analysis/
- **LSA Wiki page**
  - http://wikis/display/LSA/Home
- **FESA Home page**
  - http://project-fesa.web.cern.ch/project-fesa/development/main29.htm
- **JAPC Wiki page**
  - http://controls-wiki.web.cern.ch/controls-wiki/japc/
- **MCS page**
  - http://bdidev1.cern.ch/bdisoft/operational/abbdilhctb_wiki/Ctrl/MCS
- **RBA Wiki page**
  - http://controls-wiki.web.cern.ch/controls-wiki/COGroup/RBA/RBA
- **RBAC Wiki page**
  - http://wikis/display/LAFS/Role-Based+Access+Control
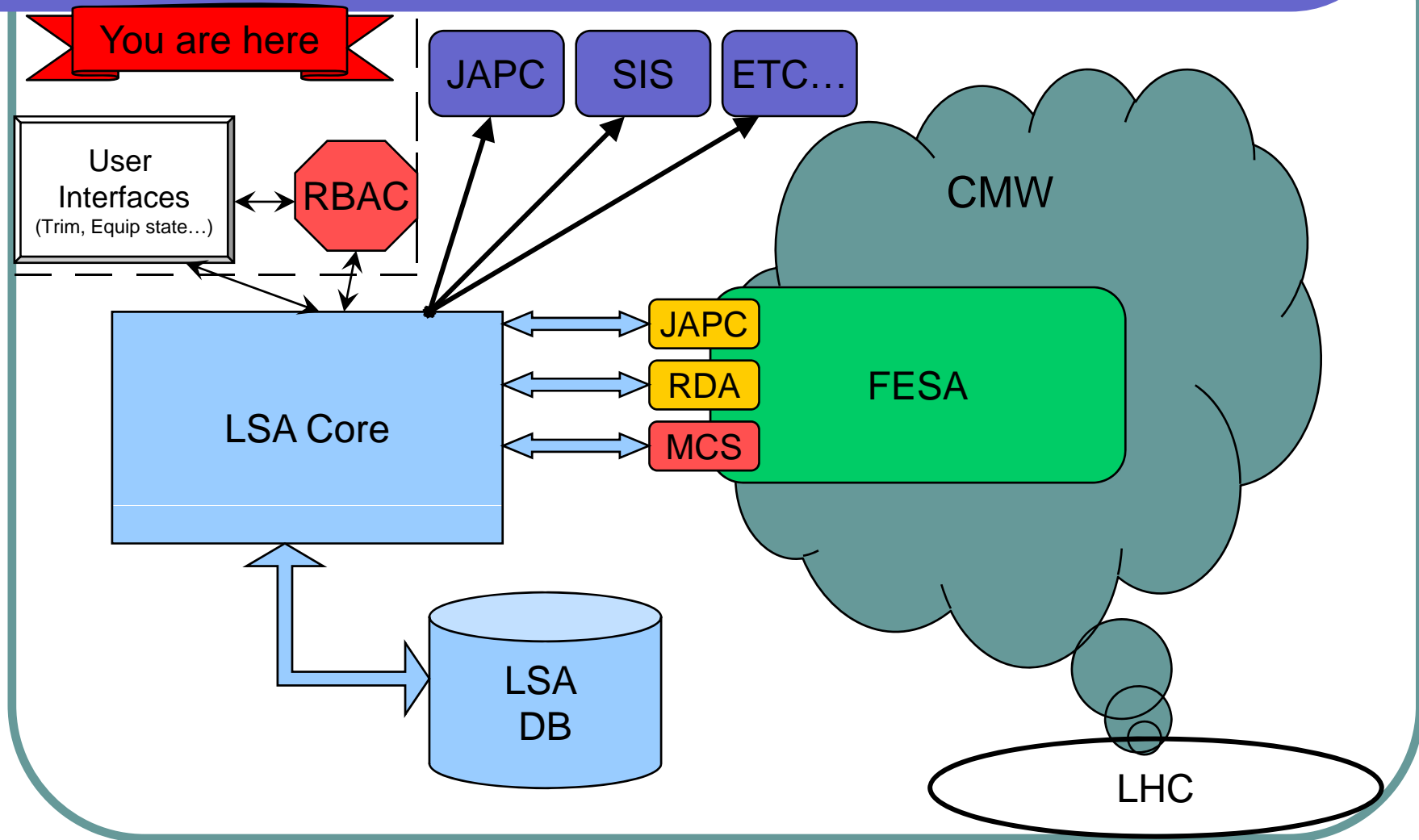
Overview

- LSA Core infrastructure
- New features for LHC
  - Settings
  - User groups
  - RBAC
  - MCS
- Pending issues

Overview

- **LSA Core infrastructure**

- New features for LHC
  - Settings
  - User groups
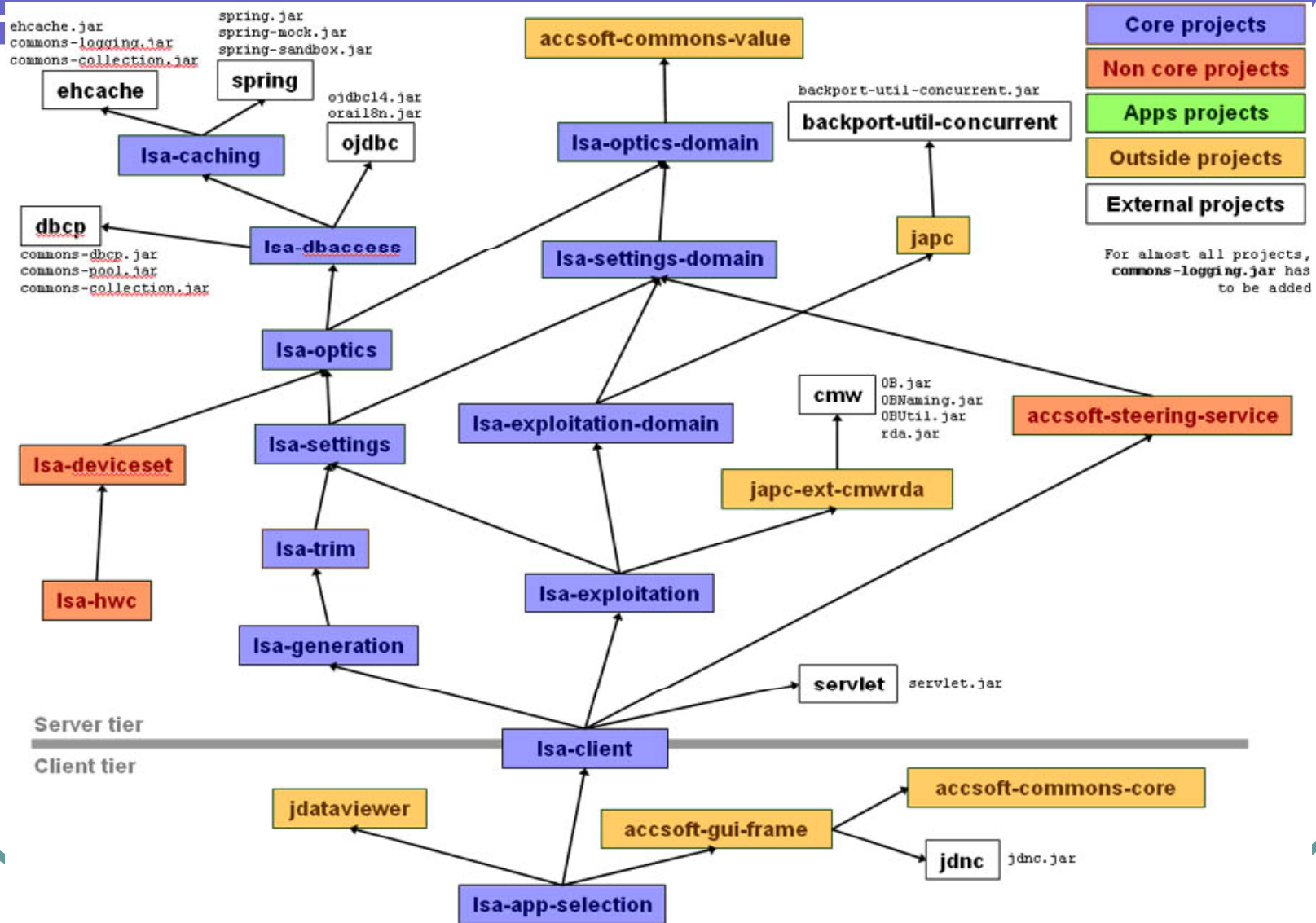 - RBAC
  - MCS

- Pending issues

LSA Core Infrastructure
LSA Core within Control System

You are here

JAPC   SIS   ETC…

User Interfaces
(Trim, Equip state…)

RBAC

CMW

JAPC
RDA
MCS

FESA

LSA Core

LSA DB

LHC

LHC Machine Checkout - Individual
systems tests - LSA Core

LSA Core Infrastructure
Detailed Schema

LSA Infrastructure
"Rules"

Parameter type

Simple Example: SPS/Momentum

On SFTPRO cycle

Momentum

Incorporation rule

9000 ms

0 ms

Make rule

9000 ms

Link rule

0 ms

IREF

12000 ms

Definitions :

● Make rule : Algorithms to convert a source parameter setting into its dependent parameter settings. E.g. conversion of Momentum (P) into current sent in the main bending magnets.

● Incorporation Rule : When we trim end points of a function, we have to ensure the continuity between point of the same settings. Incorporation rules are here to ensure this role. (NB: in case of PDOT or BDOT functions, they also ensure that the integrated sum remains at 0 value)

● Link rule : Generate link settings on the lower level (most often current level) between one cycle and its following. As high level parameters have settings only when beam is in the machine, link rules complete missing part of the functions between cycles.
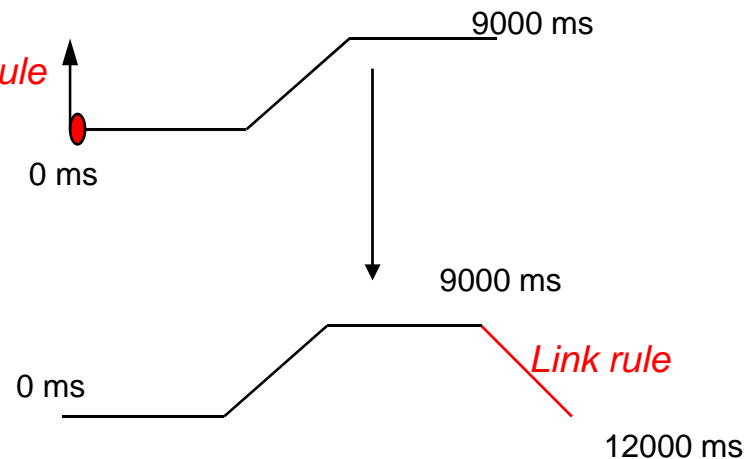
LSA Core Infrastructure
Summary

- LSA Core is well integrated within control system…
    - Use of CMW tools (FESA, JAPC)
    - Manages transactions with database but also with hardware.
    - Provides access to settings via JAPC

- … And is a good base to develop interface on top of it (you just need to depend on lsa-exploitation package and you get all Core functionalities)

Overview

- LSA Core infrastructure

- New features for LHC
  - Settings
  - User groups
  - RBAC
  - MCS

- Pending issues

New features for LHC
Settings

For more details, report to Lasse's talk

- LSA Settings infrastructure is already quite known from SPS experience (generation, trim, makerules…)

- However due to LHC being a collider and not a cycling machine, new functionalities have been added
  - Actual settings
  - Fidel (reproduce the logic of auto-trim but correction is done automatically)
  - Management of non-multiplexed settings (⇔ non-PPM)

- Remaining tasks :
  - Implementation is ongoing well, many tests have been performed.
  - Tests by OP should follow ASAP.

New Features for LHC
Users groups

- **Another layer for parameters tree**
  - Requested by equipment specialists ➜ to help them retrieving easily their own parameters.
  - Has already been transparently implemented in SPS.
  - Parameters are now filtered per user group, system, type and, finally, per name.

- **Interesting idea but :**
  - This could reduce machine status visibility (by applying this kind of filter, operator could miss important information)
  - It could get confusing to retrieve one selves between User groups, RBAC, MCS…

- **Remaining tasks :**
  - OP should validate this new functionality and provide feedback about its use.

New features for LHC
Role Based Access Control (RBAC)

- Another way of filtering rights, parameters and settings
  - For applications integrating RBAC, a login window should appear (using NICE accounts)
  - Depending on the rights granted to your account, you shall be authorized to perform certain types of actions (view, edit a setting…)

- Remaining tasks
  - Development is achieved and ready to be used
  - No user application using RBAC so far
  - Need to be tested in operational conditions

New features for LHC
Management of Critical Settings (MCS)

- Purpose : Only identified specialist can change a critical setting from a dedicated application
  - Need authentication checking
  - Should forbid changes coming from other CO interface (like FESA Navigator)
- Each setting has its own signature attached
  - The signature is encrypted via an algorithm taking into account setting's value and a public key.
  - On the front end level, a change of setting can only be accepted if the new value and its signature matches
- Remaining tasks
  - On LSA Core side, implementation is complete
  - Is Critical Systems list complete yet?
  - On User Interface side, everything has to be done
    - User identification using RBAC?
    - Same interface than for other settings?
    - If yes how do we manage parameters space?

Overview

- LSA Core infrastructure
- New features for LHC
  - Settings
  - User groups
  - RBAC
  - MCS
- Pending issues

LHC Machine Checkout - Individual
systems tests - LSA Core

Pending Issues

- BLM Threshold implementation (see Laurette and Fabio's talk)
  - Parameter space within LSA not defined yet
  - OP Use cases to be defined
  - User interface to be written
- Settings Copy
  - Should it take into account user groups?
  - Should we automatically copy Critical Settings?
  - How to handle non-multiplexed parameters?
  - Actual settings integration.
  - User interface to be developed.
- Integration with timing?

Conclusions

- LSA Core is meant to meet OP requirements
  - A large pool of LSA developers is in OP
  - We can already take benefit from SPS experience
  - LSA Core is flexible enough to include new functionalities and improve the existing ones
  - Building new interfaces using LSA requires some investment but, as you know, "good operation software is made by operation"
- Most of the new features is almost complete and can be tested
  - Core functionalities are implemented
  - We could either try and use existing GUIs or develop our owns
- We should also stay in contact with new functionalities
  - BLM Thresholds edition (do we agree with what we've been proposed?)
  - Settings copy for LHC, specifications would be welcome
  - A review of "access to settings, who do what and what can we see?" would be interesting.