Databases for operations of the Injectors – overview, dependencies and strategy for smooth upgrades of the data-driven controls system

LHC Injectors and Experimental Facilities Committee 2011 Workshop

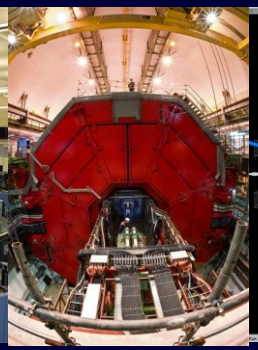
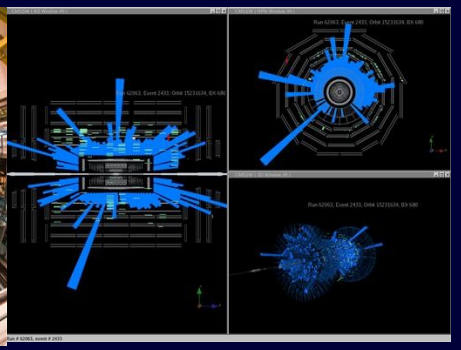
Zory Zaharieva
on behalf of the DM team

Beams Department
Controls Group
Data Management Section

Outline

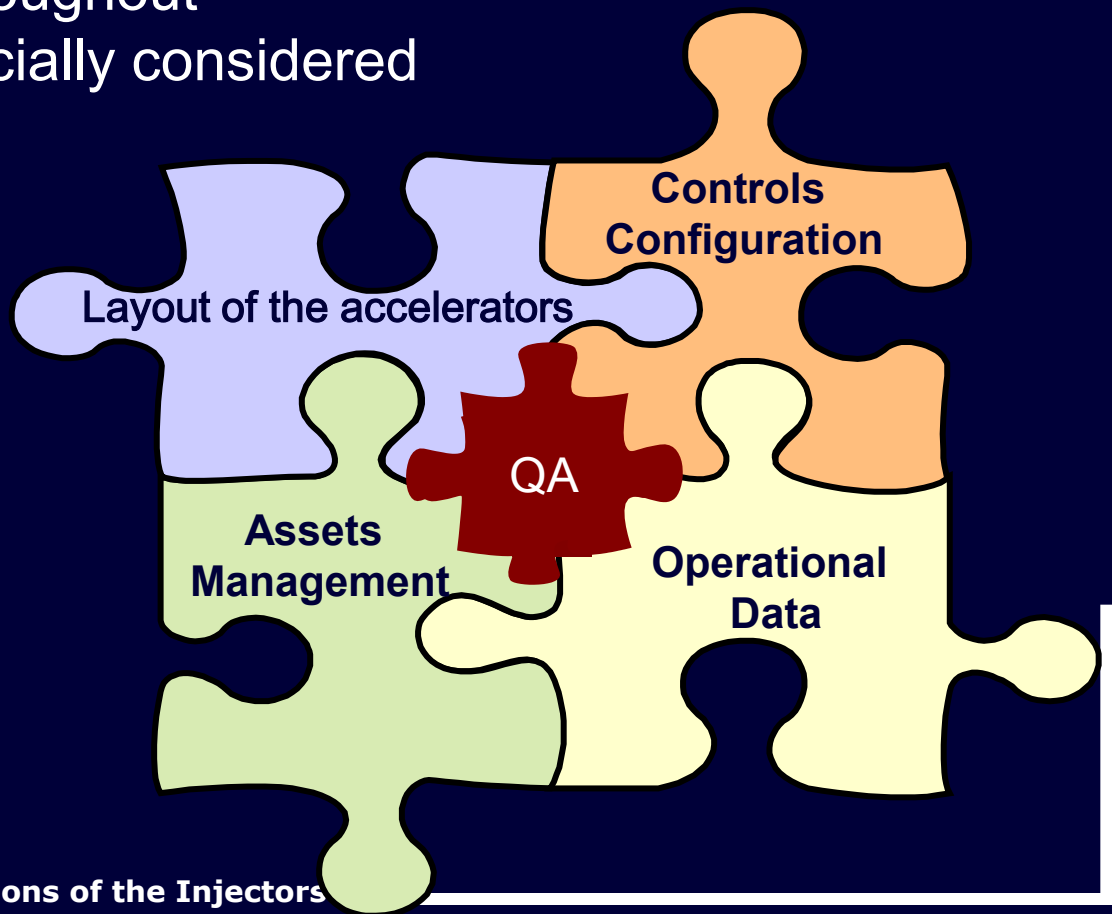
- Overview of the main data domains for the Injectors
- State and on-going developments
- Dependencies between the different databases
- Strategy for smooth upgrades of the data-driven controls system
- Conclusion



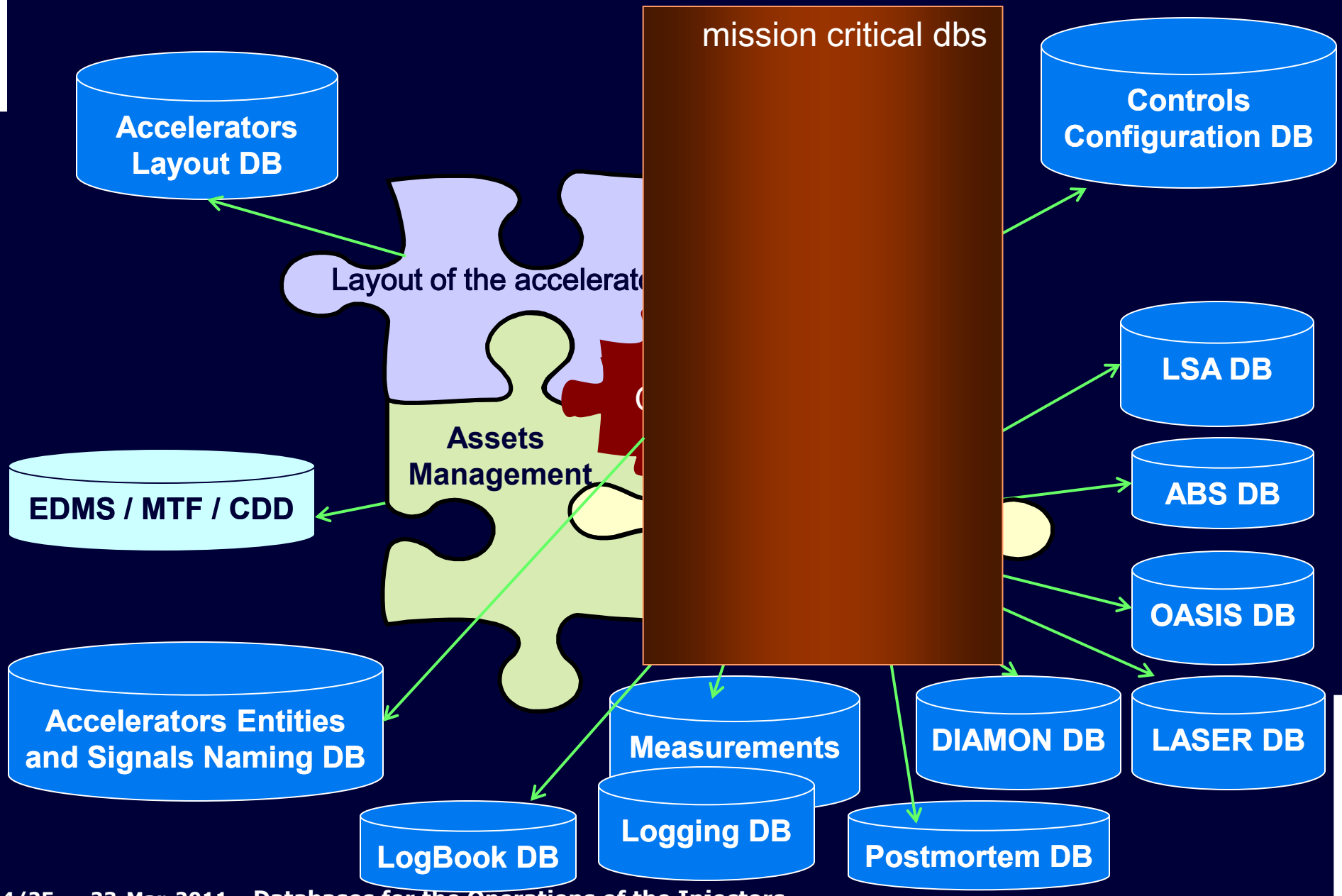




Data Domains for the Injectors

- Logical break-down of the data
- Easier to organize and manage each individual area
- Integration of the data throughout the domains must be specially considered
  - Common understanding
  - Good communication



List of the Databases in the Data Domains



Implementation Strategy

- DB technology
  - Oracle Databases for all technical data
- Data-driven applications and APIs
  - Java, J2EE
  - Oracle technology applications stack
- Reliable database services
  - On-line usage of database services for the accelerators control

Outline

- Overview of the main data domains for the Injectors
- State and on-going developments
- Dependencies between the different databases
- Strategy for smooth upgrades of data-driven controls system
- Conclusion

Layout DB

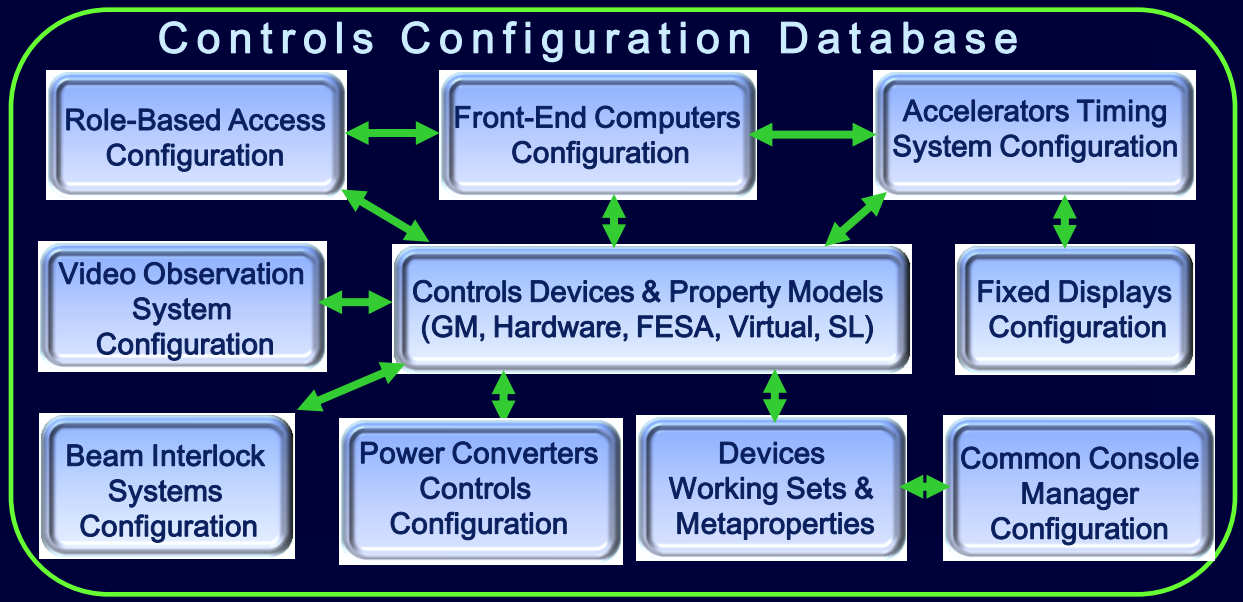
- Information scope – data about the positions (slots) of the installed equipment
  - Beam line equipment
  - Assembly break-down structures of slots
  - Electrical circuits - connectivity between power converters and their loads (magnets, RF cavities, …)
  - Layout of the electronics for the controls system
  - Hub to other data repositories
    - Design info, drawings, pictures (EDMS);
      Asset (MTF);
      Equipment databases (NORMA, TE-EPC ALIM DB)
- Aim to describe the complete accelerator complex

Layout DB recent work

- Work on capturing data since Sep 2010
  - Linac4 (priority 1) - capturing data for the electrical circuits including those related to the accelerating cavities
  - Work on the installed equipment in the Transfer Lines towards the Booster [LT, LTB, LBE, LBS, BI] (priority 2); driving factor LINAC4
  - Establishing the Booster electrical circuits (priority 3)
  - Work on TT10 (priority 4) – investigation into the existing sources of data (MAD files, drawings, reality)
  - Started the feasibility study to link Layout data to the 'Cablotheque' DB
- Data capture and validation
  - Is challenging & resource intensive – reverse engineering from legacy sources
  - Needs prioritizing
- Internal work on the Layout db structures – refactoring the db model due to the expansion in functionality during the last 7 years

Controls Configuration DB

- The heart of the Controls System providing configuration data for all accelerators - service with 25 years of history
- Contains data for the complete Controls System topology
  - Main users: FESA, CMW, RBAC, Computers Configuration (~3000), Drivers Gen, Accelerators Timing System, Common Console Manager, FixedDisplays, Power Converters Controls ...
  - 5 device-property models (GM, FESA, SL, Hardware, Virtual)
  - Total of 1400+ classes & 75000+ devices
  - For the Injectors
    800+ classes &
    34000+ devices

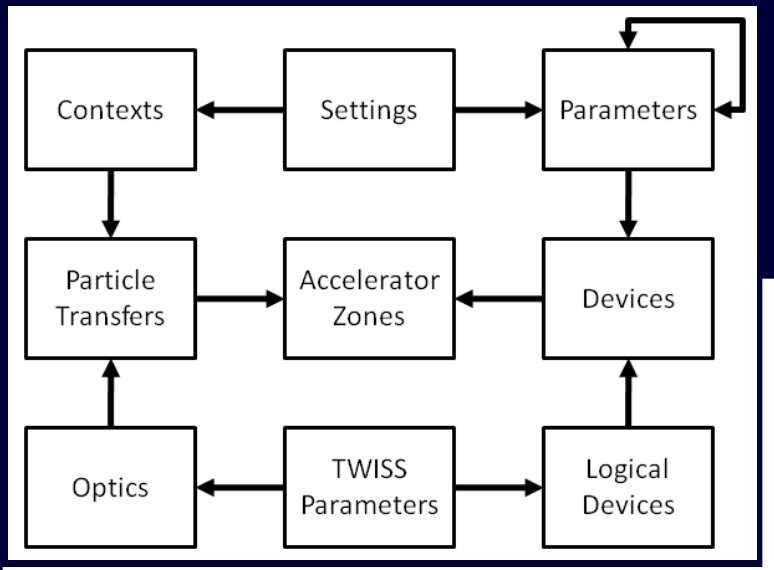


Controls Configuration DB

- New db developments for:
  - FESA 3.0 – major effort to restructure the data management part
    - Structured handling of XML files (used only for data exchange)
    - Complete integration of FESA data into the CCDB relational db model
    - Easier to handle common data management tasks (FESA 2.x)
  - Renovation projects – InCA and ACCOR
  - Improvements in the Front-End Computers Configurations
  - Renovation of the Accelerators Timing Configuration
- Continuous improvements of:
  - Interactive GUIs – 12 data editing applications (200+ forms), reporting tools (150+ reports)
  - Java & PL/SQL APIs
- Request to OP & Equipment experts – maintain the data!

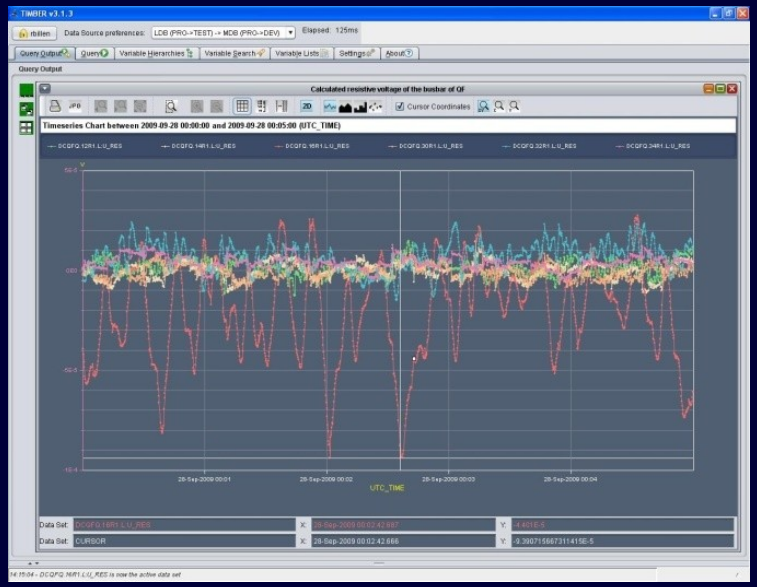
Operational Domain – LSA DB

- High level controls and settings management for the accelerators
- Introducing the LSA DB into the PS Complex – renovation project
  - PS in 2010, PSB in 2011
  - Integrate and validate the existing data into the LSA model
  - Extend and adapt the model to cater for the specifics of the Injectors
- PS Complex Automated Beam Steering DB (ABS)
  - Migrating ABS data to LSA DB (YASP)
  - Difficult to model the PS machine
  - Phasing out the ABS DB in 2013



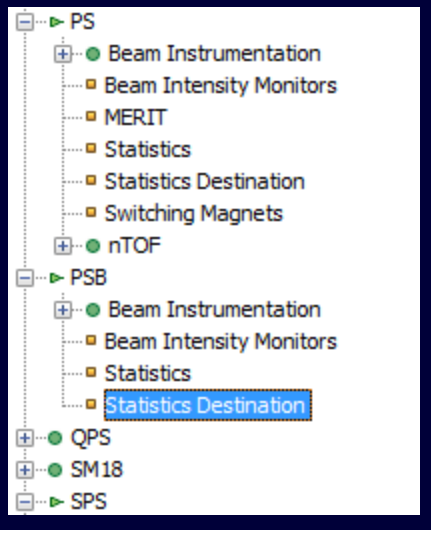
Operational Domain - Measurements and Logging

- Stores time-series data on beam and equipment measurements



- Functionality boost and new developments
  - Java API for clients
  - Internal db improvements
  - GUI – Timber v.4

- Provides some tailored services for the Injectors
  - Statistics – based on expressed OP requirements



- Request to data providers and end-users - check the stored data periodically to ensure correct data has arrived

Operational Data Domain – Diamon and LASER

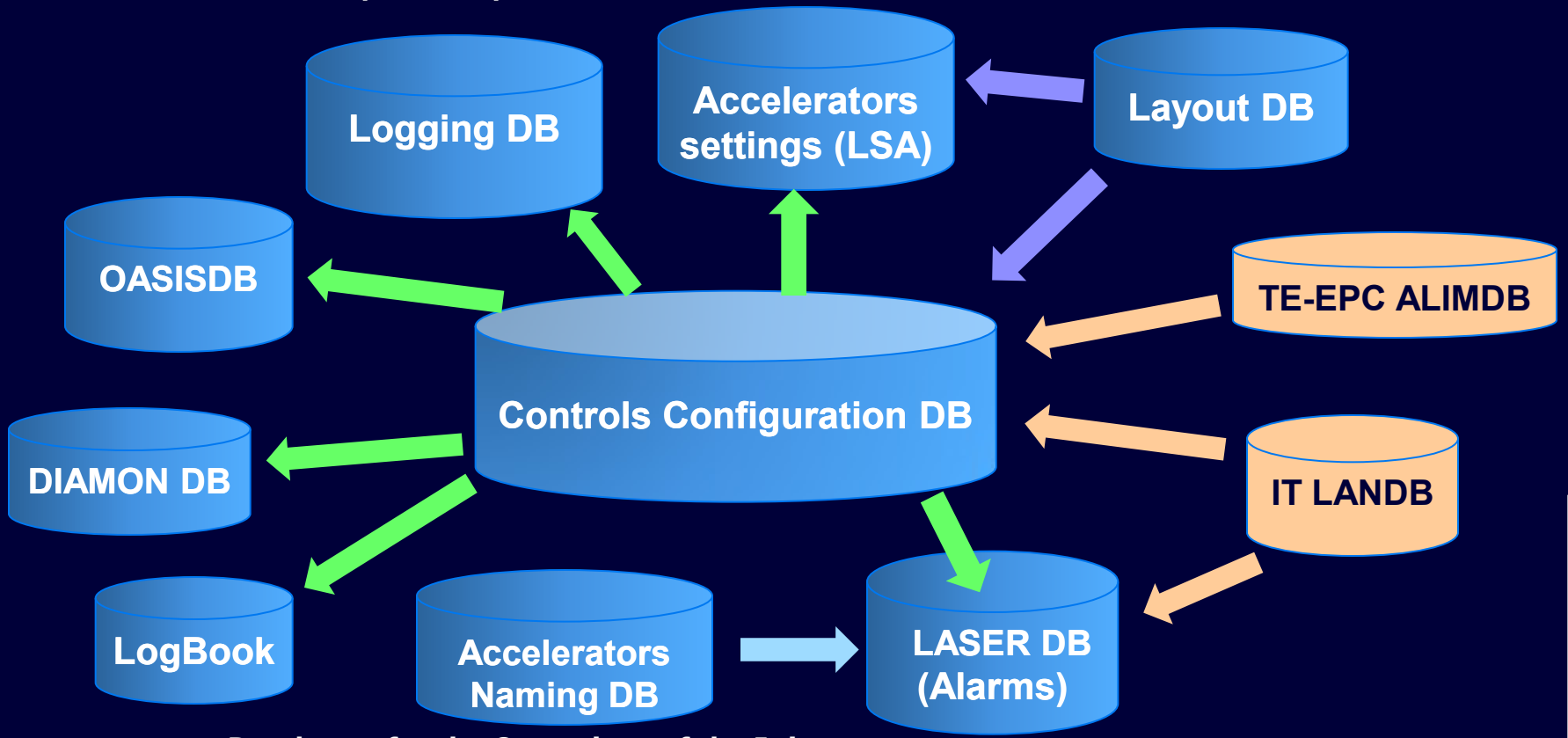
- LASER (Alarms) - capturing, storing and notification of anomalies
- Diamon - diagnostics and monitoring of the Controls infrastructure
- Continuous efforts in improving the configuration data set
  - Streamlining the data flow into the Alarms database from the different providers of alarms configuration data
  - Enhancing the data set propagated from CCDB to Diamon
  - Development of suite of Data Management Tools for LASER – give users the possibility to explore their data and maintain it
- Work on new db models for Laser and Diamon - triggered by upcoming changes in both systems

Outline

- Overview of the main data domains for the Injectors
- State and on-going developments
- Interdependencies between the different databases
- Strategy for smooth upgrades of data-driven controls system
- Conclusion

Interdependencies between the databases

- Data is maintained only in one place
  - Ensuring single source of consistent data
  - Examples: Controls configuration data in Controls Configuration DB, data for positions of installed components in Layout DB
- Data propagation from one domain to another for the purpose of the accelerator complex operation

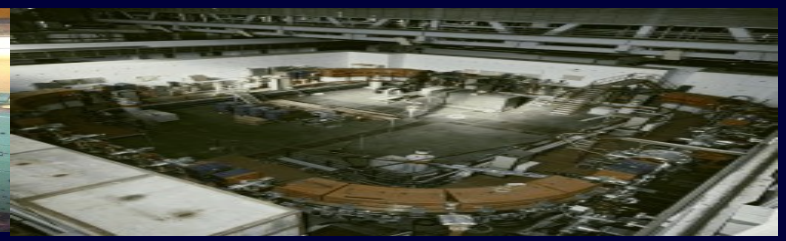


Responsibility

- DM team responsible for
  - Database structures
  - GUIs to modify / visualize the data
  - Initial data loading
- Data Owners – equipment experts, CO experts and Operators
  - Responsible for the data – to define data and keep it up-to-date
- A database is only as good as the correctness of the data it contains and how closely it represents reality

Outline

- Overview of the main data domains for the Injectors
- State and on-going developments
- Interdependencies between the different databases
- Strategy for smooth upgrades of data-driven controls system
- Conclusion



Types of Upgrade from DM point of view

- Upgrades of the database systems and related interfaces
  - Structural / functional changes to the db schema or interfaces
  - Under the responsibility of the DM team
- Changes to the data stored in the different databases
  - Under the responsibility of the data owners

Upgrades of the Database Systems

- Strategy for smooth upgrades
  - Involve end-users right from the start, throughout the design and development process
  - Provide 4 separate environments for development, unit and functional testing, integration testing (TestBed), production
  - Analyze the impact of a change and try to apply only backward compatible changes
  - Clear communication on scheduled intervention and their impact
  - Coordinate the upgrades with impacted clients
- The strategy is working well
- Proposal to OP & Equipment experts to verify the list of link people, responsible for the different applications and systems each year

Changes to the data stored in the databases

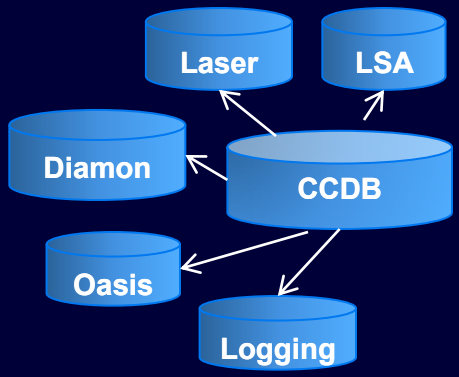
- In a data-driven Controls System
  - The data in the different databases represents components and their properties as seen by the Controls System
  - Example: FESA or GM class and its properties
- The components of the Controls System need to be upgraded regularly (Front-End Computers, Device Classes, etc), therefore there is a need for regular data changes

Safe propagation of data changes

- Challenges in a distributed db environment
  - A data change in one database could have an impact in other areas (non-backward compatible data changes)

Example: A controls device is renamed in the Controls Configuration DB

- is this device used in LSA by Operations
- are there alarm definitions in LASER
- are there signals to be logged for it in the Logging DB
- are there equipment relations in Oasis



- Need to know the data dependencies between the different systems
- Need to ensure a coherent set of data throughout all distributed databases – the Controls system needs a coherent set of data

Strategy for safe propagation of data changes

- Changes of data should be addressed in a structured and systematic way
  - Knowledge of data dependencies of Operations systems
  - Analyze the impact of a data change
  - Authorize only backward compatible changes (if possible)
  - Non-backward compatible changes allowed with coordination and follow-up
- Proposed solution: restrict (delay) data modifications for non-backward compatible changes until necessary actions have been taken in the dependent systems
- Clearly define the visibility (usage) of the data for Operations systems
  - Defining public (operations) and private (equipment experts) data for the Configuration of the Controls system

Conclusion

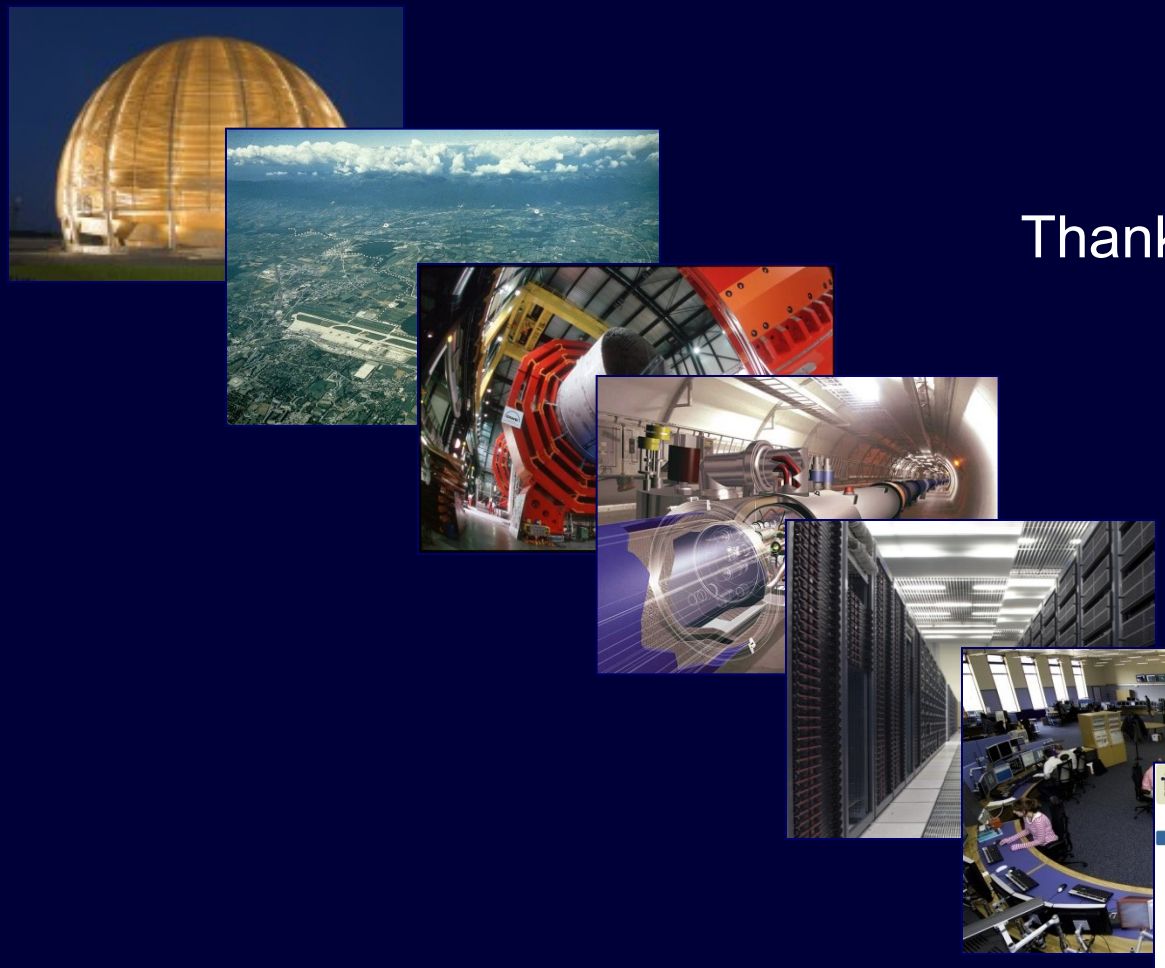
- The databases for the Injectors Controls and Operation are a fundamental service upon which different systems are built

- Continuous effort is being put into rationalizing, improving, federating and developing new functionality in the existing databases and their interfaces
  - Years of history for some of the domains and accelerators – more difficult to impose certain QA rules

Conclusion

- Data management requires the involvement of the data owners and data users
  - Being proactive, maintaining the data up-to-date, cleaning-up obsolete data
- To ensure smooth upgrades, based on data changes of the data-driven control system procedures should be followed
  - In the Layout domain - Engineering Change Requests
  - In the Controls Configuration domain - notify dependent systems, restrict (delay) data modifications for non-backward compatible changes
- Communication is an important aspect of the smooth upgrades
  - Data owners
  - Data users
  - DM team

Questions



Thank you for your attention!