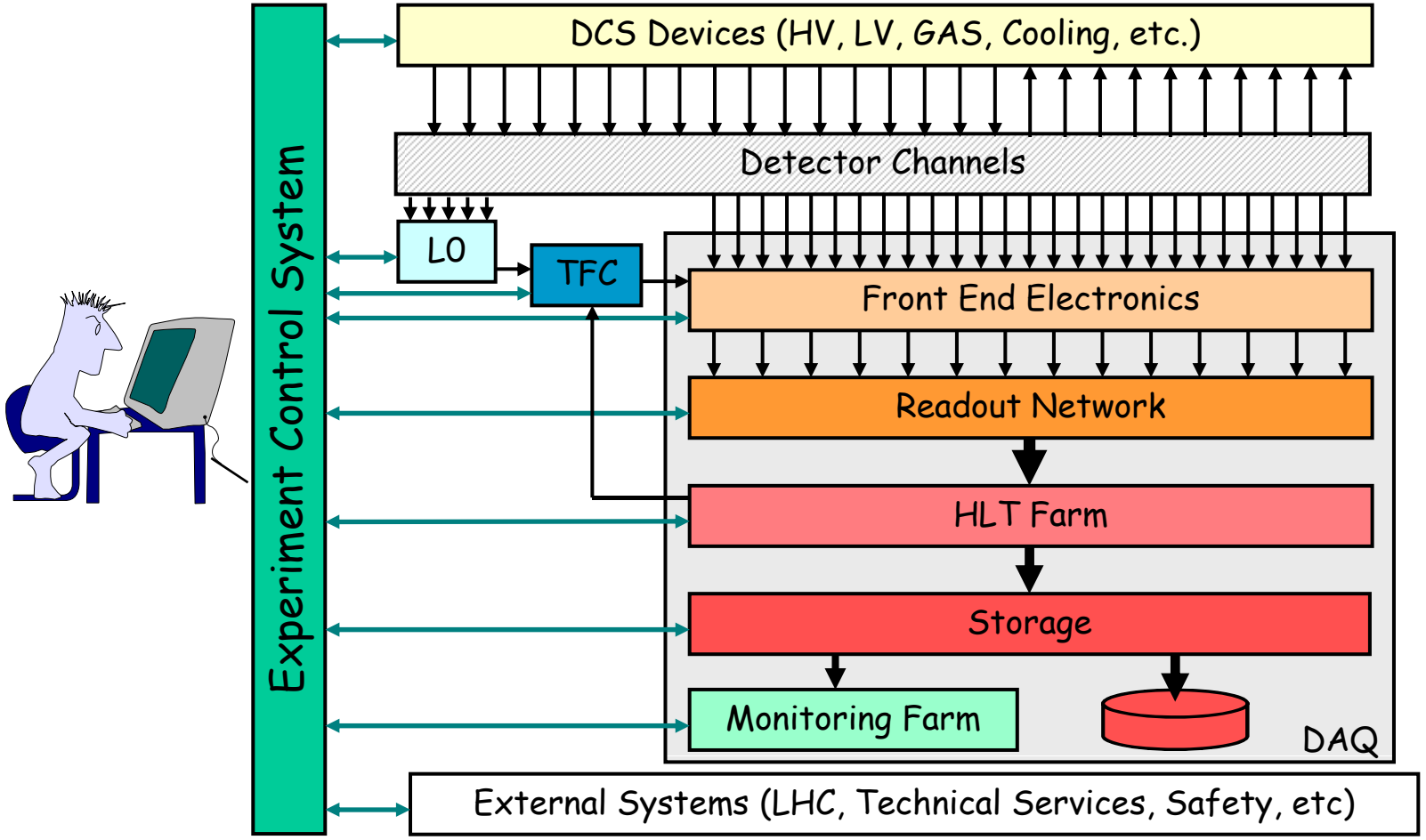
The LHCb Run Control

An Integrated and Homogeneous Control System

Clara Gaspar, March 2009

The Experiment Control System

- Is in charge of the Control and Monitoring of all parts of the experiment



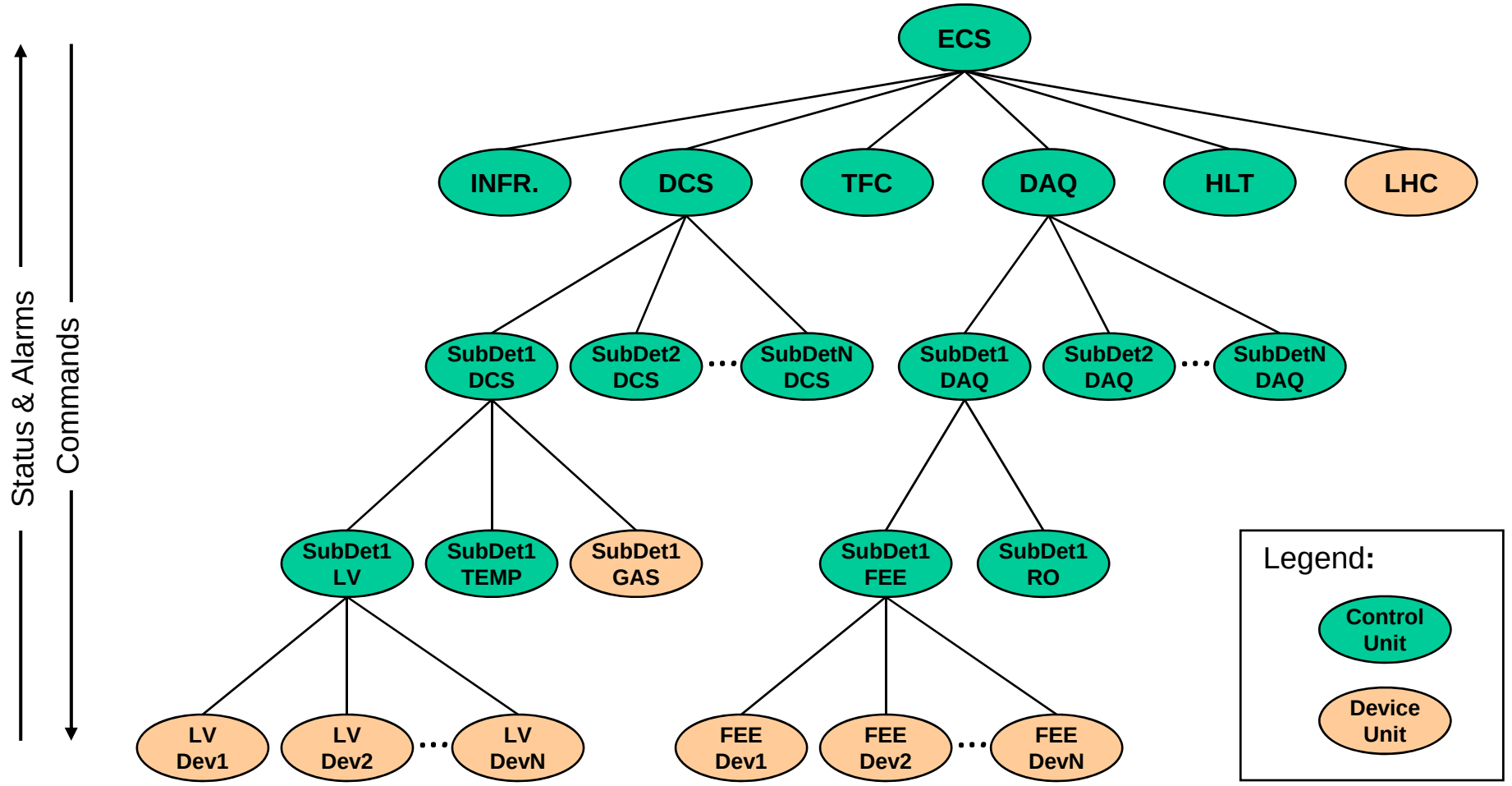
Some Requirements

- Large number of devices/IO channels
  - ➡ Need for Distributed Hierarchical Control
    - De-composition in Systems, sub-systems, … , Devices
    - Local decision capabilities in sub-systems
- Large number of independent teams and very different operation modes
  - ➡ Need for Partitioning Capabilities (concurrent usage)
- High Complexity & Non-expert Operators
  - ➡ Need for Full Automation of:
    - Standard Procedures
    - Error Recovery Procedures
  - ➡ And for Intuitive User Interfaces

Design Steps

- **In order to achieve an integrated System:**
  - Promoted HW Standardization
    (so that common components could be re-used)
    - Ex.: Mainly two control interfaces to all LHCb electronics
      - Credit Card sized PCs (CCPC) for non-radiation zones
      - A serial protocol (SPECS) for electronics in radiation areas
  - Defined an Architecture
    - That could fit all areas and all aspects of the monitoring and control of the full experiment
  - Provided a Framework
    - An integrated collection of guidelines, tools and components that allowed the development of each sub-system coherently in view of its integration in the complete system

Generic SW Architecture



The Control Framework

- **The JCOP* Framework is based on:**
  - SCADA System - PVSSII for:
    - Device Description (Run-time Database)
    - Device Access (OPC, Profibus, drivers)
    - Alarm Handling (Generation, Filtering, Masking, etc)
    - Archiving, Logging, Scripting, Trending
    - User Interface Builder
    - Alarm Display, Access Control, etc.
  - SMI++ providing:
    - Abstract behavior modeling (Finite State Machines)
    - Automation & Error Recovery (Rule based system)

Device Units

Control Units

* - The Joint COntrols Project (between the 4 LHC exp. and the CERN Control Group)

Device Units

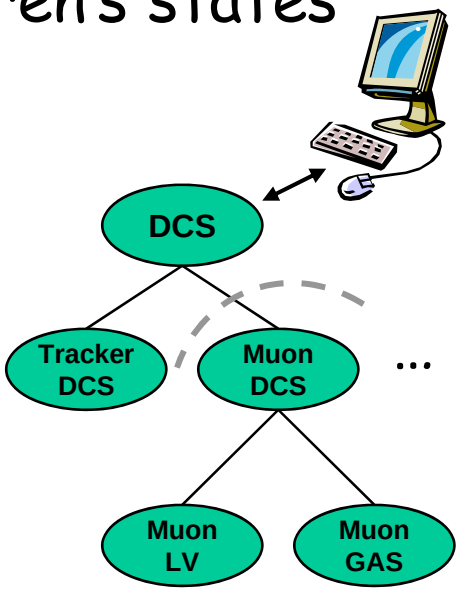
Device Unit

- **Provide access to "real" devices:**
  - The Framework provides (among others):
    - "Plug and play" modules for commonly used equipment. For example:
      - CAEN or Wiener power supplies (via OPC)
      - LHCb CCPC and SPECS based electronics (via DIM)
    - A protocol (DIM) for interfacing "home made" devices. For example:
      - Hardware devices like a calibration source
      - Software devices like the Trigger processes (based on LHCb's offline framework – GAUDI)
    - Each device is modeled as a Finite State Machine

Hierarchical control

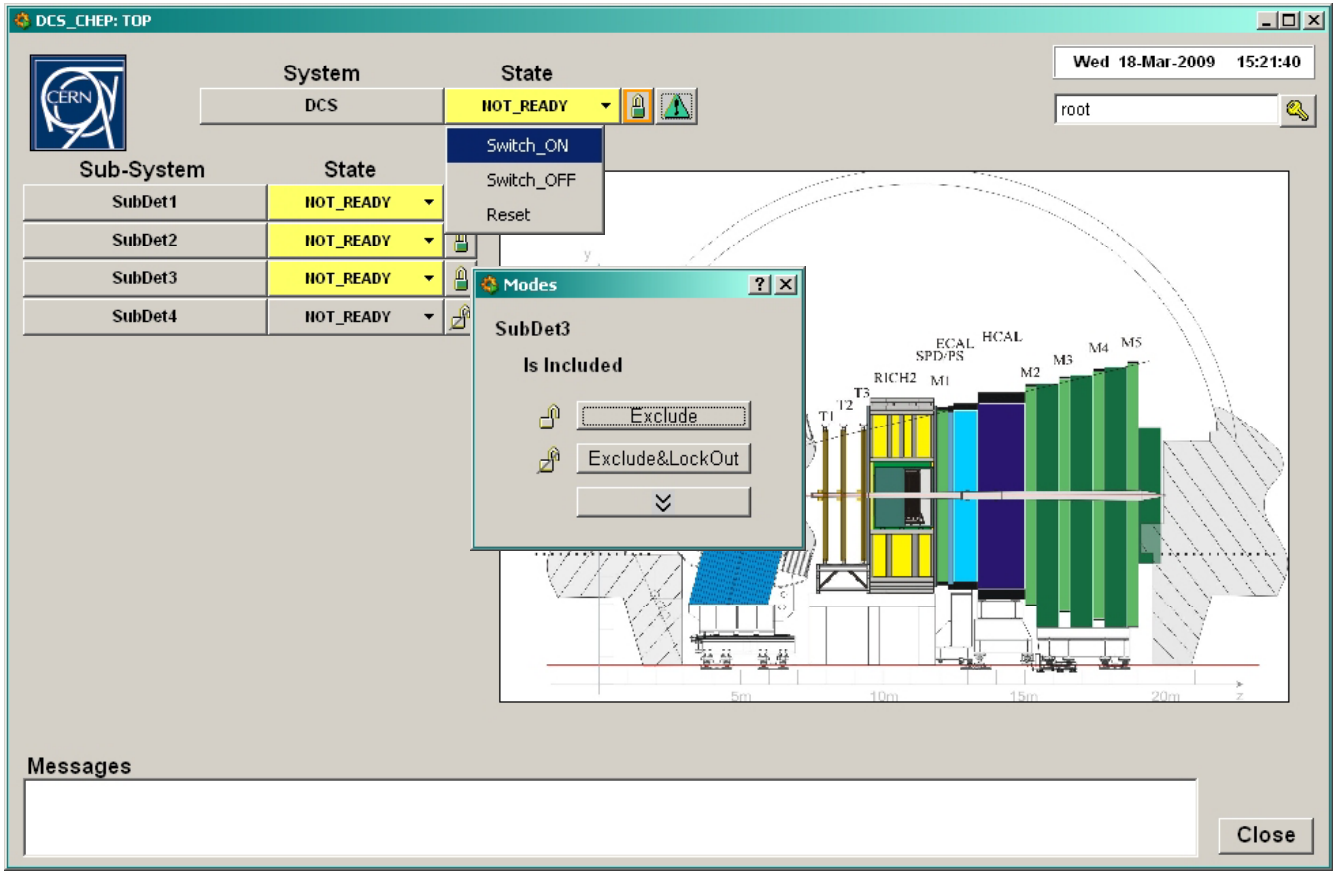
Control Unit

- **Each Control Unit:**
  - Is defined as one or more Finite State Machines
  - Can implement rules based on its children's states
  - In general it is able to:
    - Summarize information (for the above levels)
    - "Expand" actions (to the lower levels)
    - Implement specific behaviour & Take local decisions
      - Sequence & Automate operations
      - Recover errors
    - Include/Exclude children (i.e. partitioning)
      - Excluded nodes can run is stand-alone
    - User Interfacing
      - Present information and receive commands



Control Unit Run-Time

- **Dynamically generated operation panels**
  (Uniform look and feel)



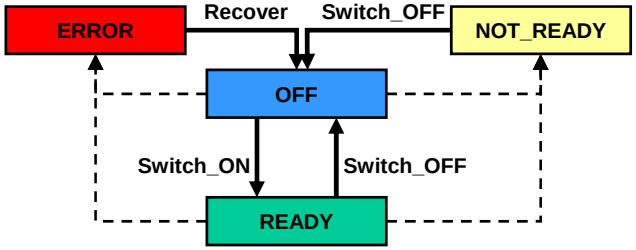
- Configurable User Panels and Logos
- "Embedded" standard partitioning rules:
  - **Take**
  - **Include**
  - **Exclude**
  - **Etc.**

Operation Domains

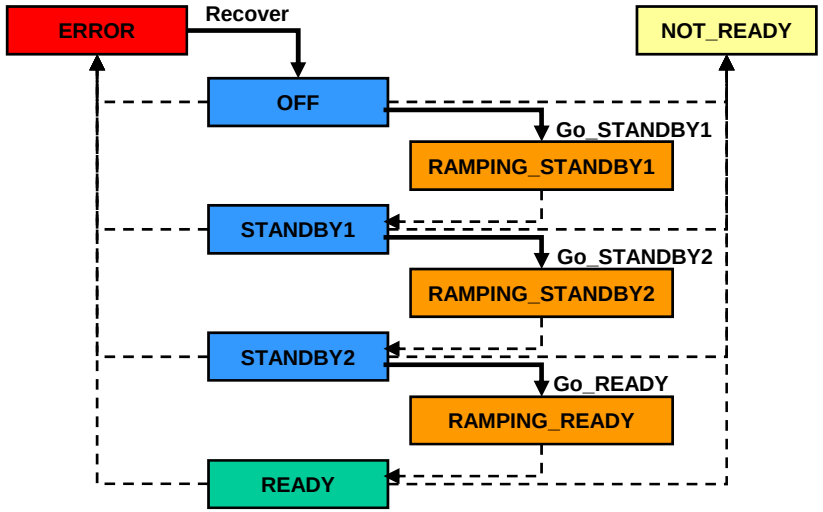
- **Three Domains have been defined:**
  - DCS
    - For equipment which operation and stability is normally related to a complete running period Example: GAS, Cooling, Low Voltages, etc.
  - HV
    - For equipment which operation is normally related to the Machine state. Example: High Voltages
  - DAQ
    - For equipment which operation is related to a RUN Example: Readout electronics, High Level Trigger processes, etc.

FSM Templates

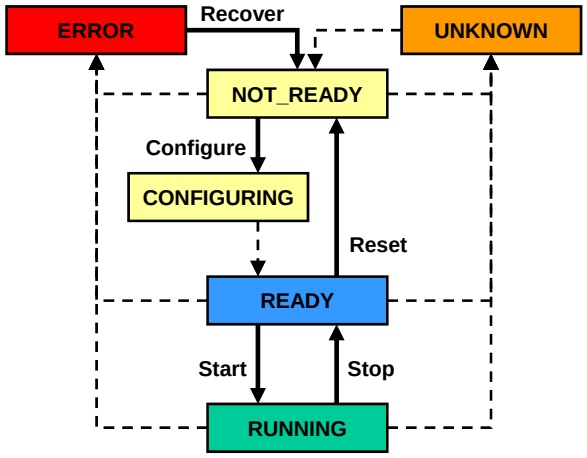
DCS Domain



HV Domain



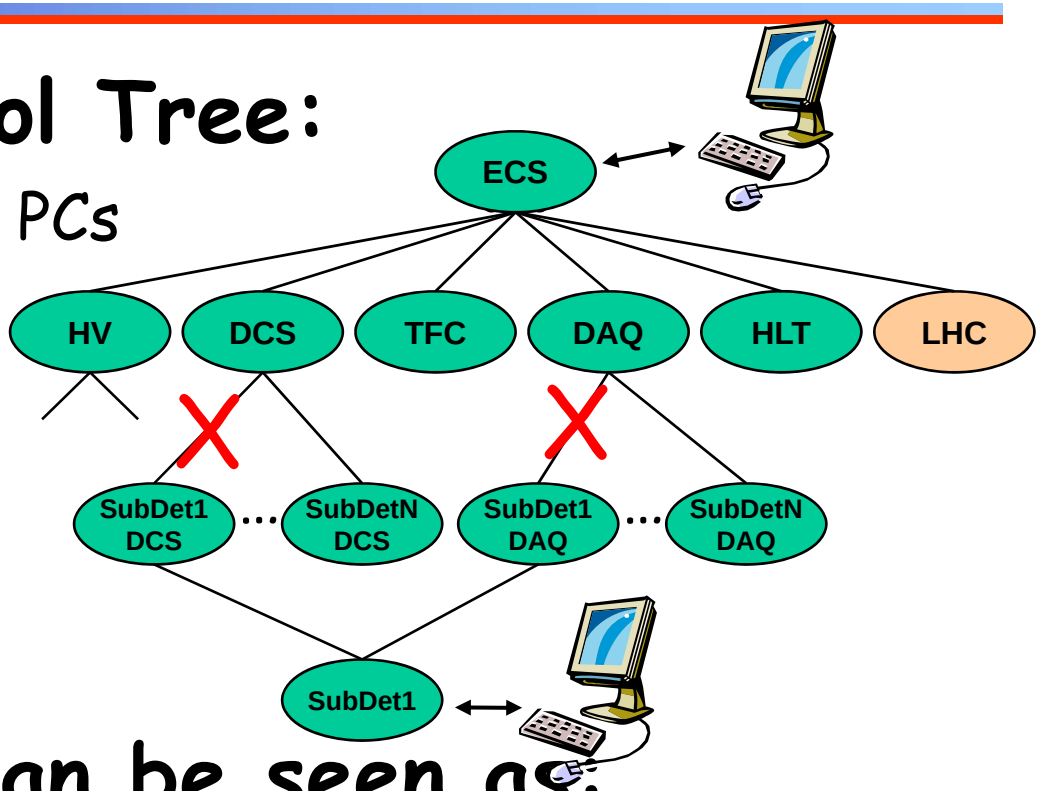
DAQ Domain



- All Devices and Sub-Systems have been implemented using one of these templates

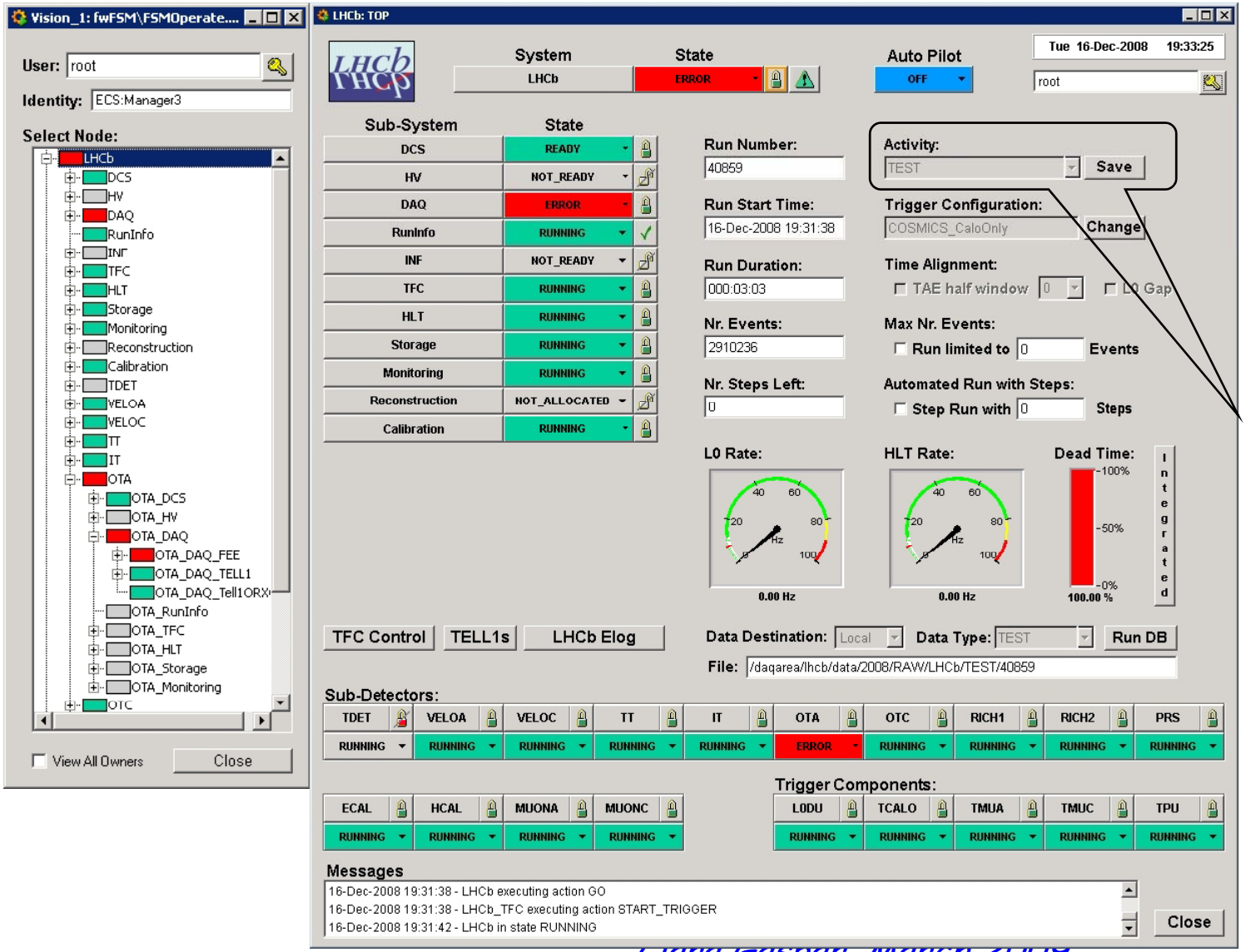
ECS: Run Control

- **Size of the Control Tree:**
  - Distributed over ~150 PCs
    - ~100 Linux (50 for the HLT)
    - ~ 50 Windows
  - >2000 Control Units
  - >30000 Device Units



- **The Run Control can be seen as:**
  - The Root node of the tree
  - ➡ If the tree is partitioned there can be several Run Controls.

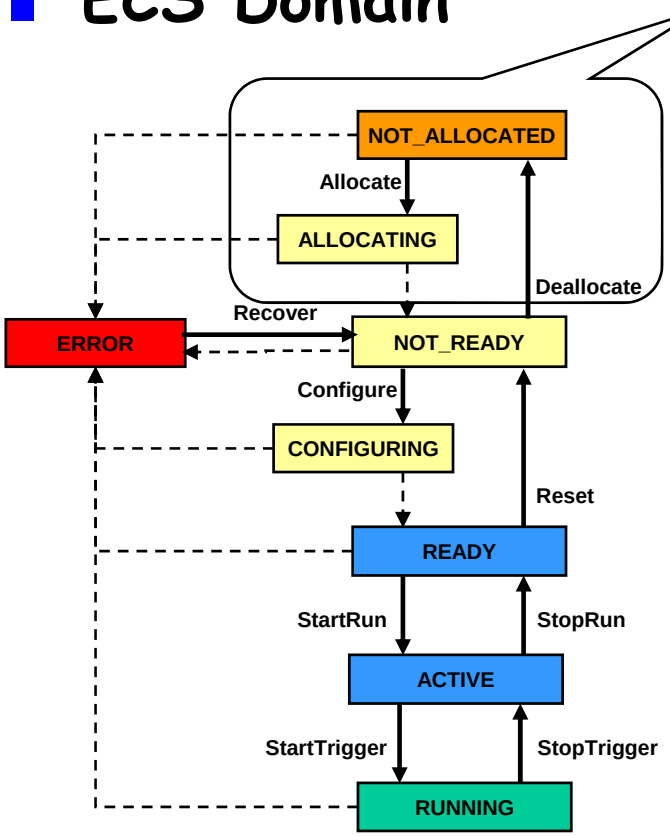
Run Control



- **Matrix**
  - Domain
  - X
  - Sub-Detector
- **Activity**
  - Used for Configuring all Sub-Systems

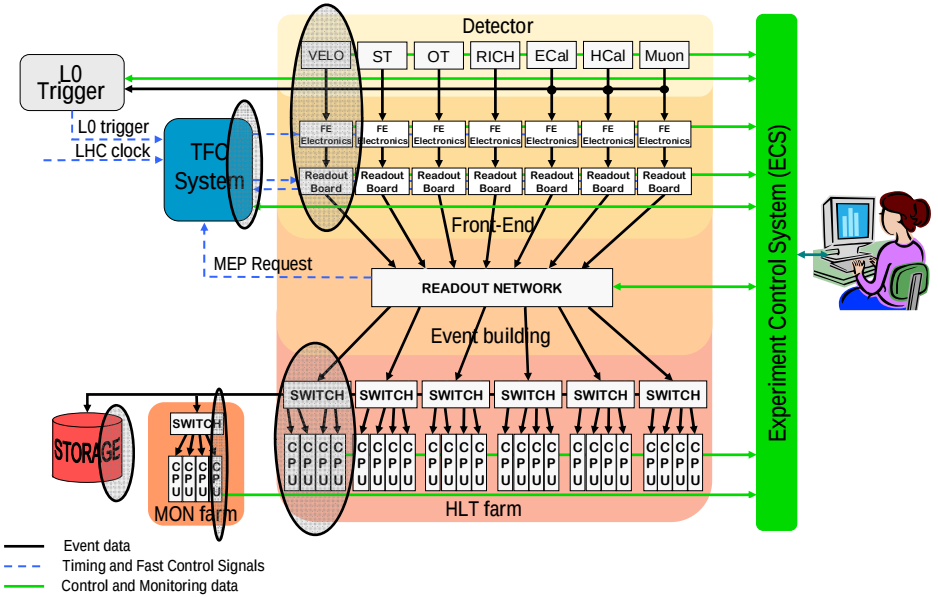
Partitioning

ECS Domain

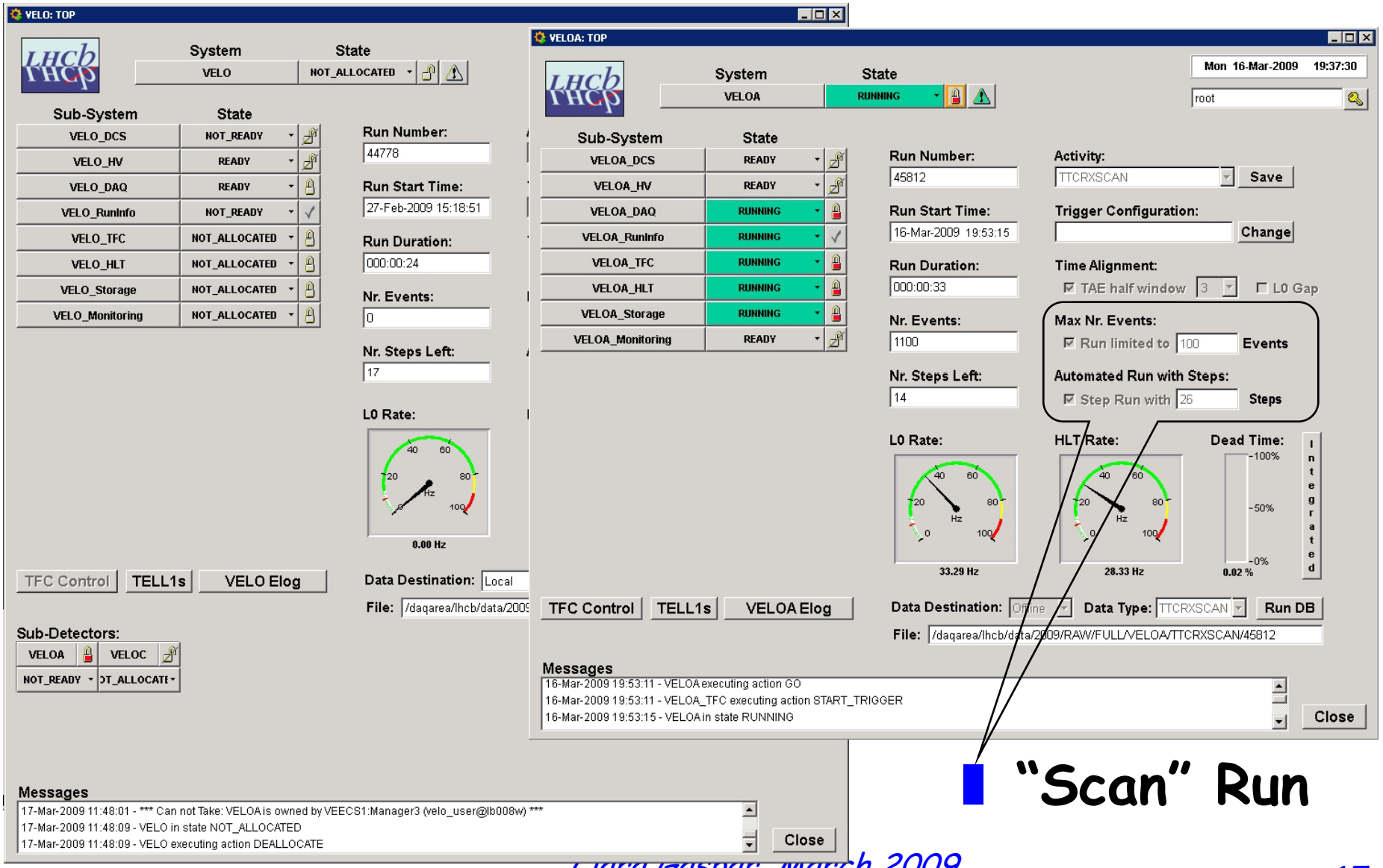


Creating a Partition

- Allocate = Get a "slice" of:
  - Timing & Fast Control (TFC)
  - High Level Trigger Farm (HLT)
  - Storage System
  - Monitoring Farm



Sub-Detector Run Control



Conclusions

- **LHCb has designed and implemented a coherent and homogeneous control system**
- **The Run Control allows to:**
  - Configure, Monitor and Operate the Full Experiment
  - Run any combination of sub-detectors in parallel in standalone
  - Can be completely automated (when we understand the machine)
- **Some of its main features:**
  - Sequencing, Automation, Error recovery, Partitioning
  - ➡ Come from the usage of SMI++ (integrated with PVSS)
    (Poster no: 540, Board no: Thursday 103)
- **It's being used daily for all sub-detector tests and global activities**