$\bar{P}$ANDA Detector Control System

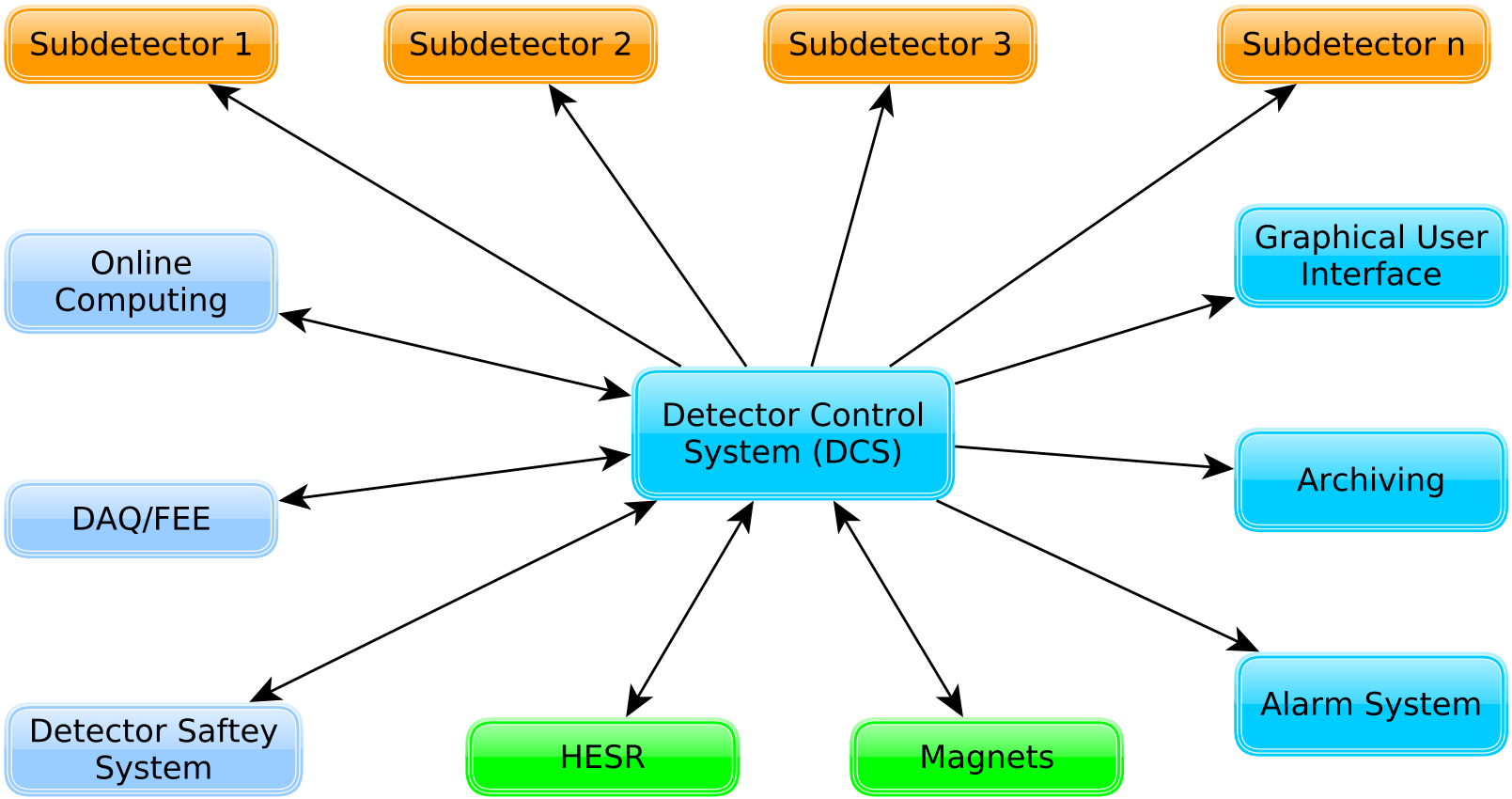
$\bar{P}$ANDA – Thailand Detectors Meeting

Tobias Triffterer

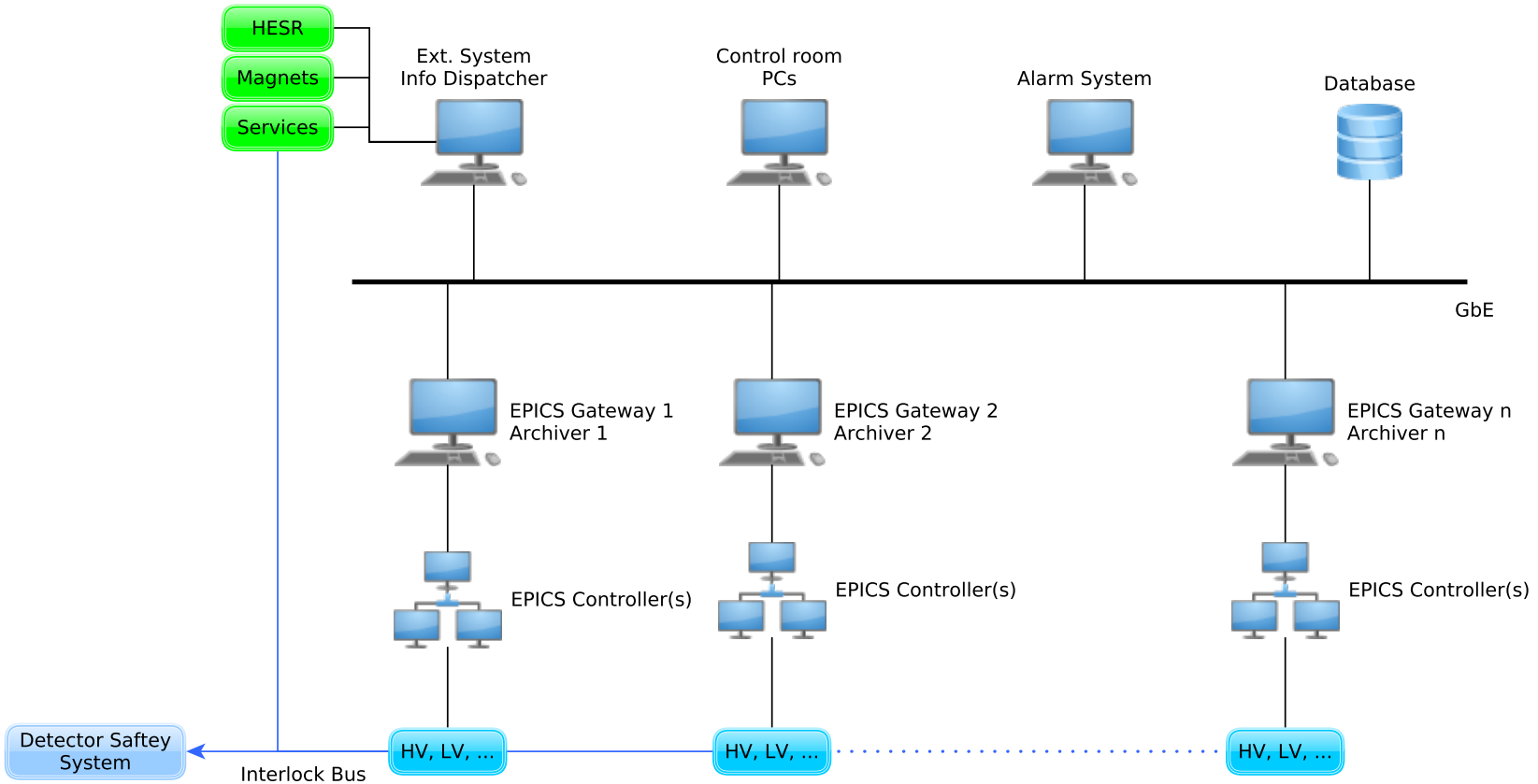
Ruhr-Universität Bochum – Experimentalphysik I

16th January 2019

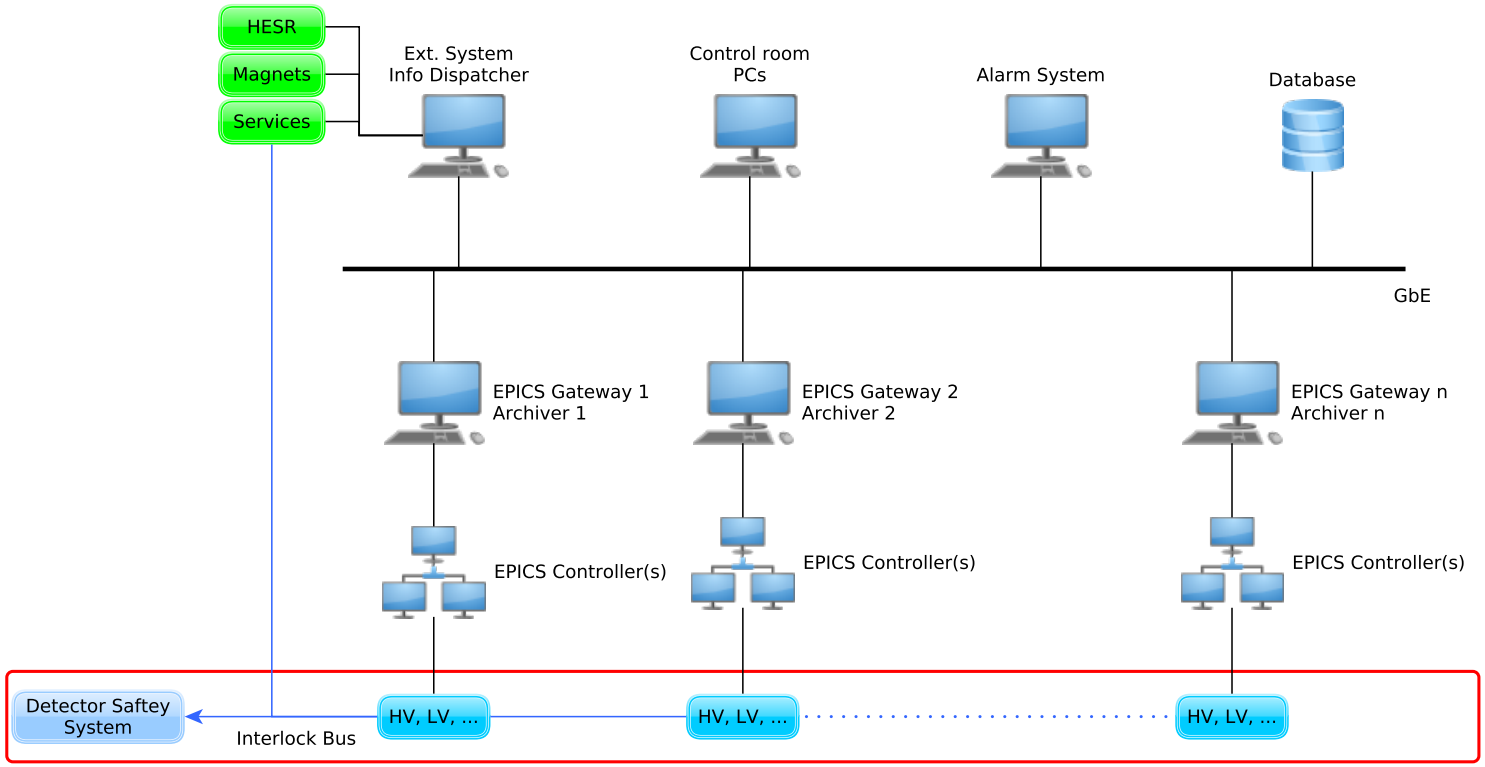
Detector Control System Centralized View



DCS Overview



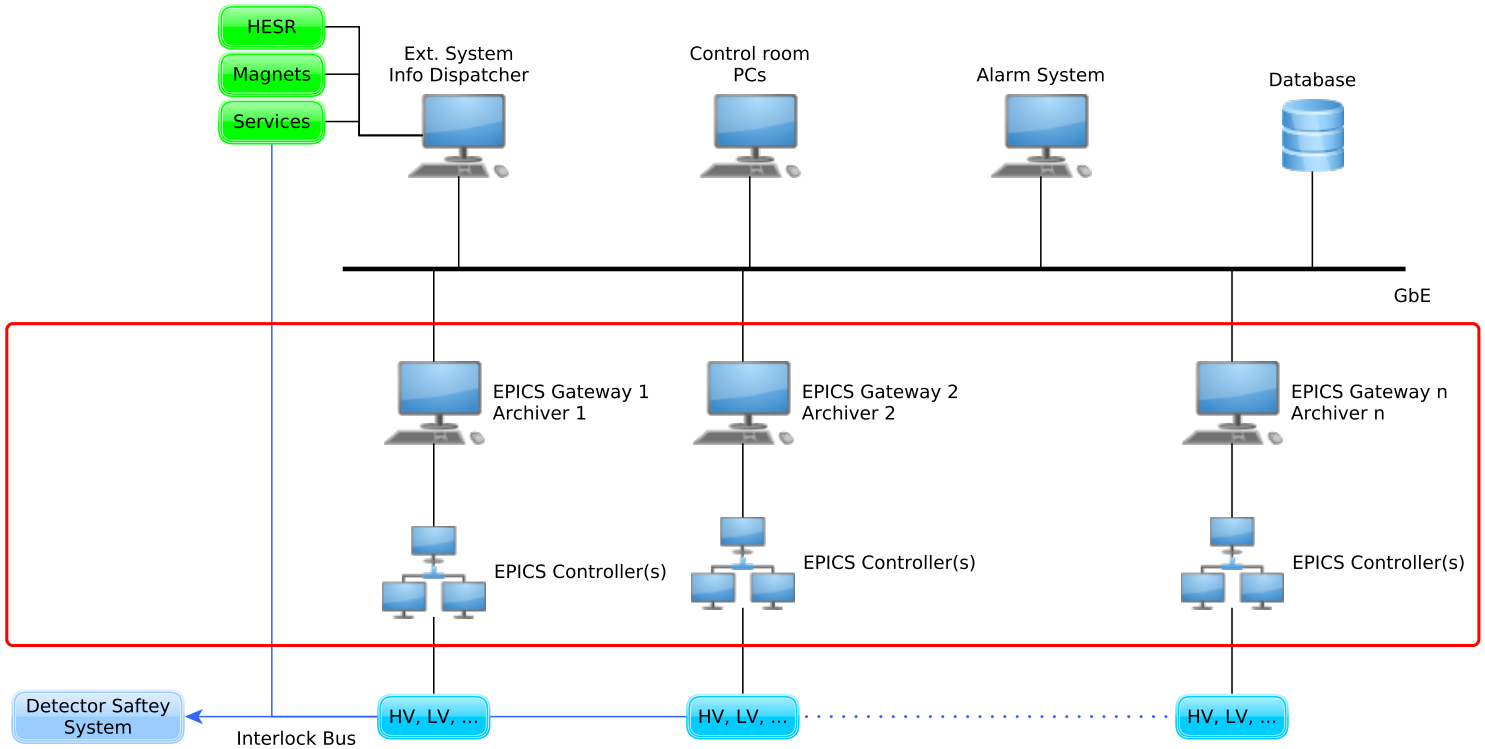
DCS Overview



Field Layer (FL):

- Temperature monitoring, power supplies, valves,...
- Every device that is monitored or controlled
- Detector Safety System (e.g. Interlocks)

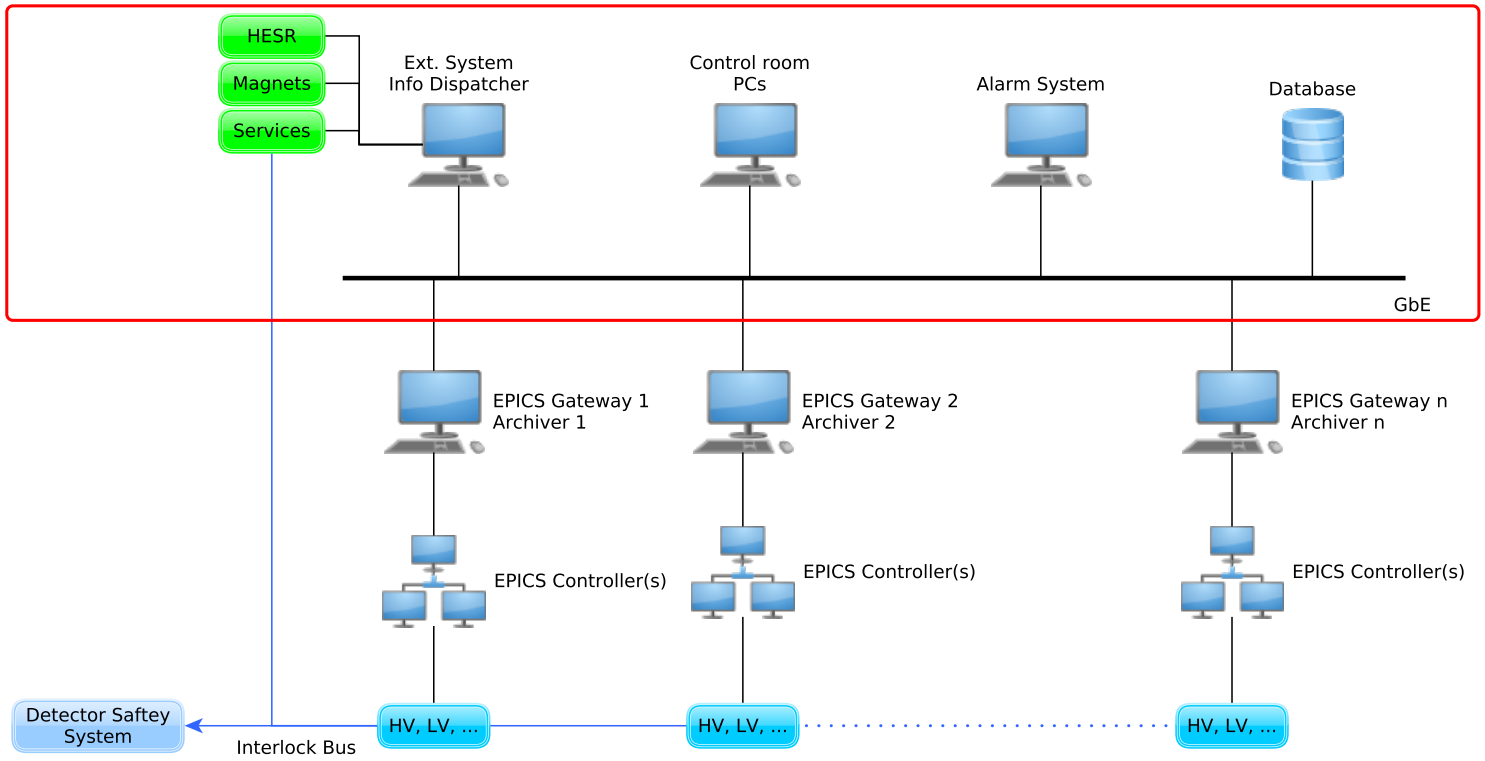
DCS Overview



Control Layer (CL):

- Input/Output controller communicating with devices in FL
- Archiver for data collection
- Gateway to Supervisory Layer

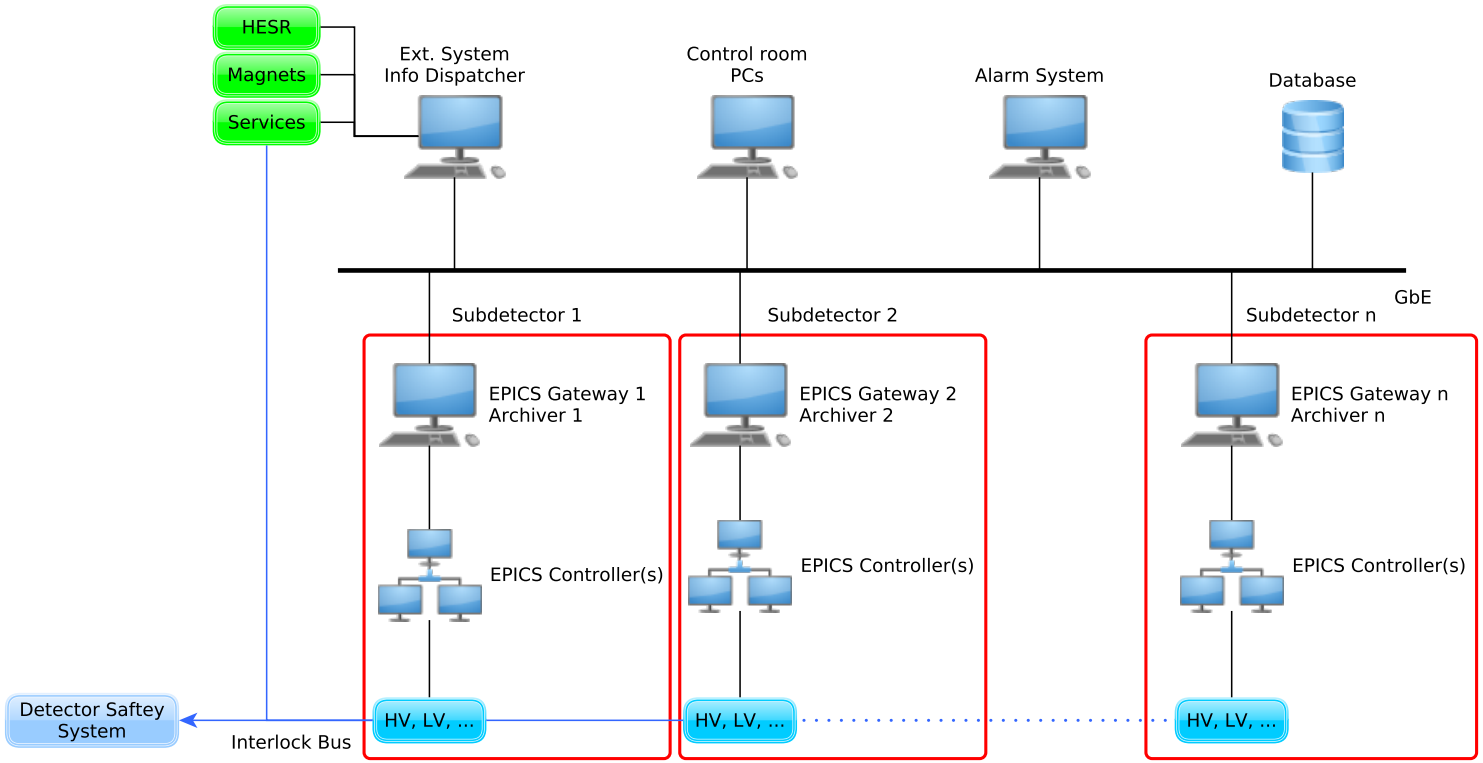
DCS Overview



Supervisory Layer (SL):

- Databases for data storage
- Graphical user interfaces
- Interface to “external” systems and experiment control

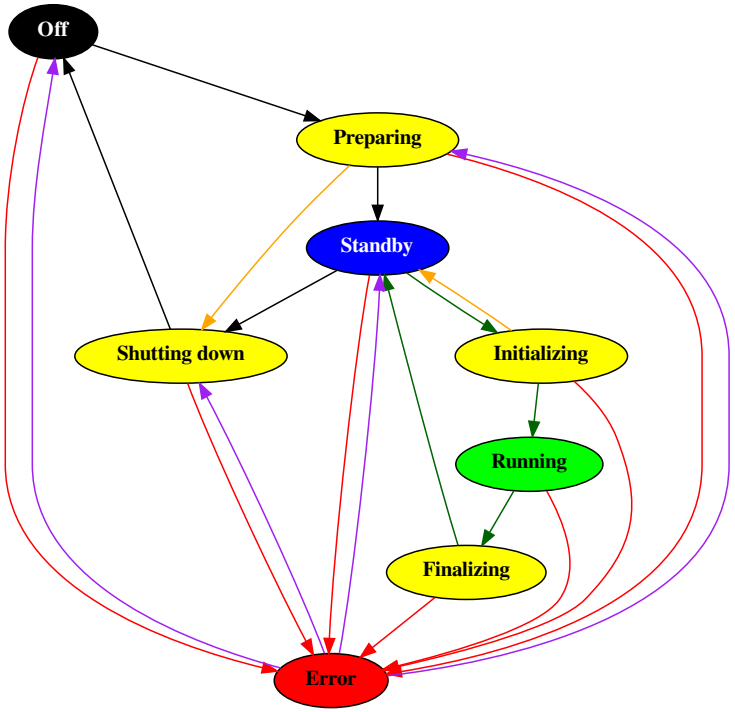
DCS Overview

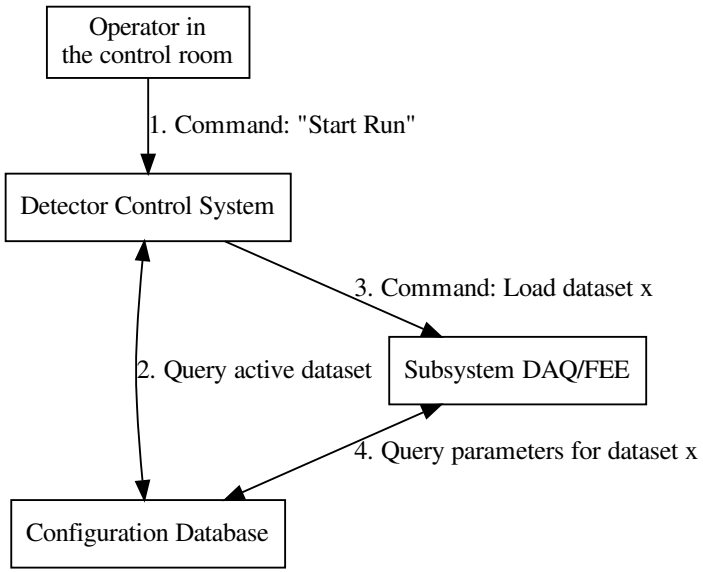


EPICS - **E**xperimental **P**hysics and **I**ndustrial **C**ontrol **S**ystem

- Decentralized architecture
- Freely scalable
- Allows “partitioning” ⇒ each subdetector has its own DCS

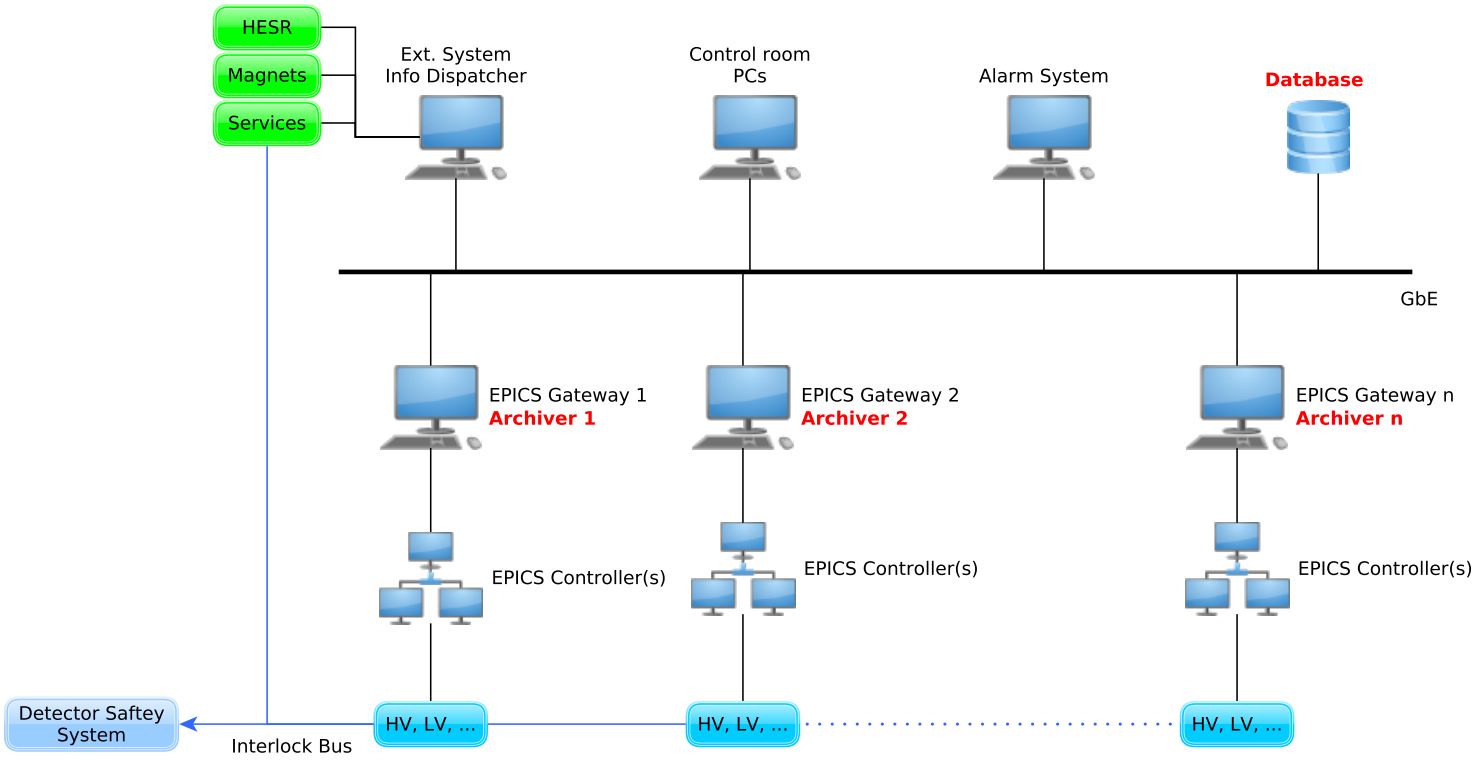
Finite State Machine and Front-End Configuration





- Each subdetector needs to perform defined actions
- Coordination with other parts of $\overline{\text{P}}$ANDA needed
- More information in $\overline{\text{P}}$ANDA DCS TDR draft

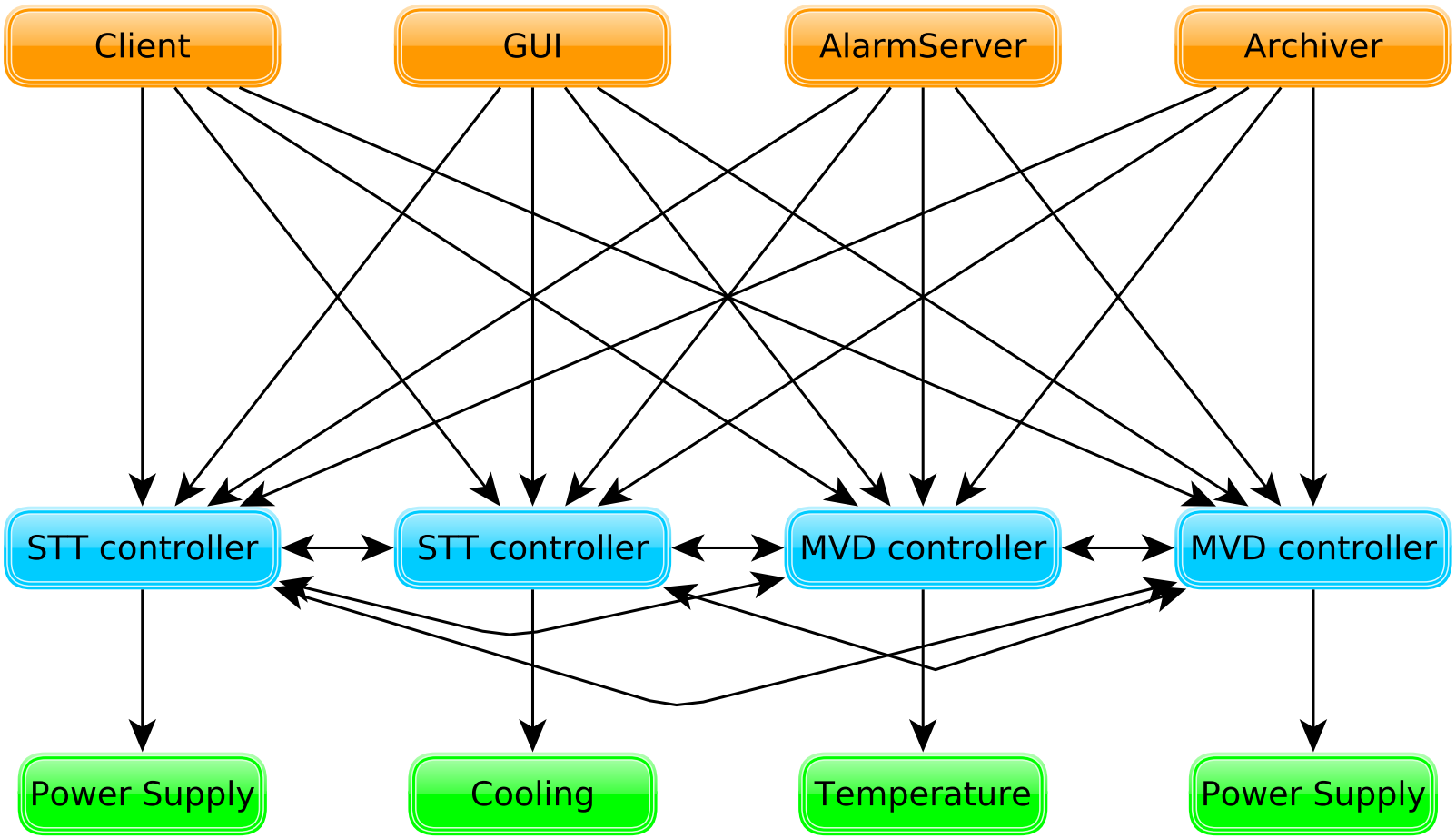
Archiving Slow Control Data



- Each subdetector has its own archiver engine (in CL)
- One common database as storage (in SL)

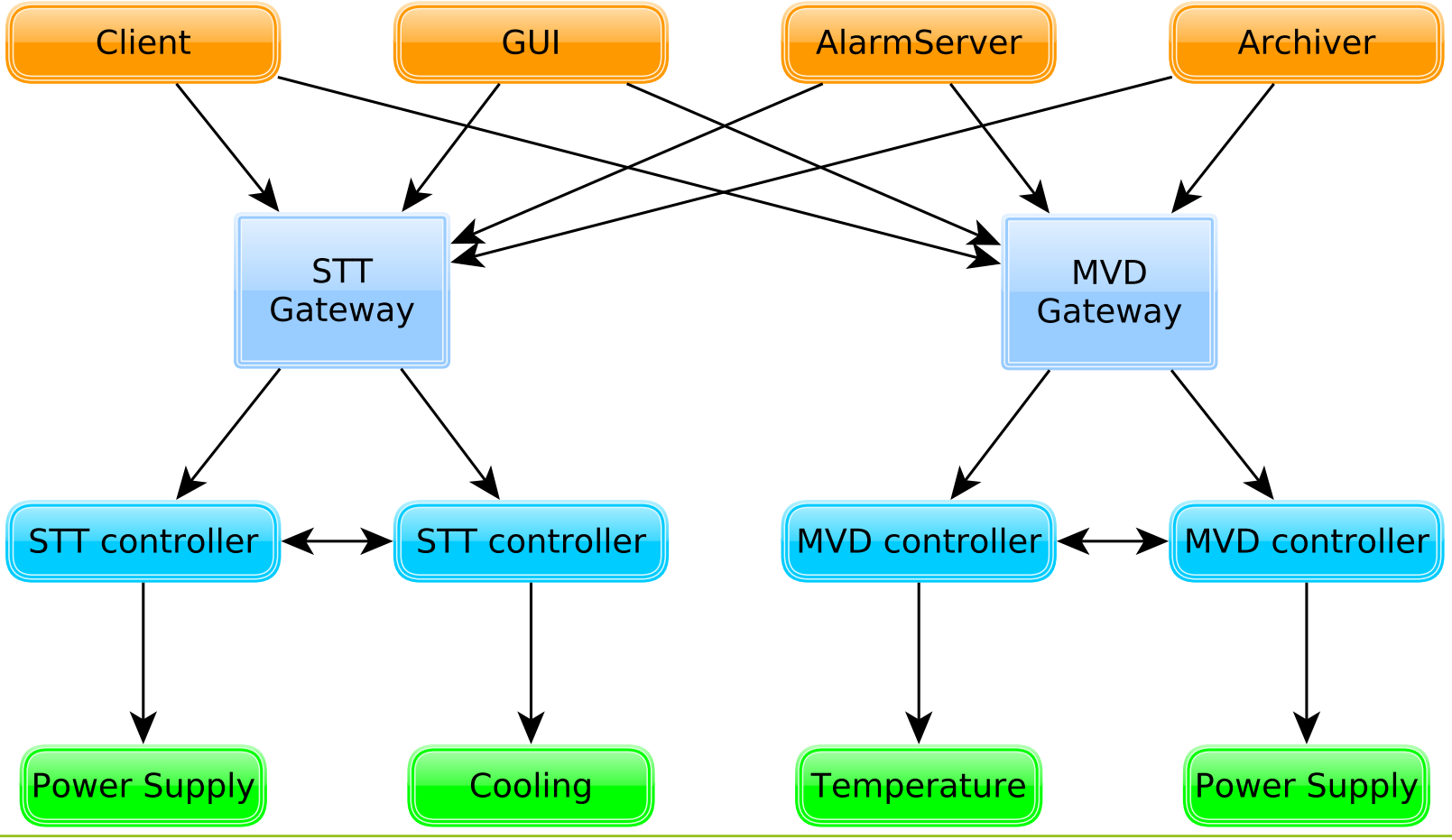
EPICS Communication Protocol

Each Client is connected to each Server
Arrows indicate direction of data queries



EPICS Communication Protocol

- Using gateways to separate subdetectors from global DCS
- Uni-directional connection, but reverse gateways possible
- Gateways not necessary in lab test setup for a single detector



EPICS Databases

- Each EPICS IOC has a database of records
- Database basically tells EPICS what to do
- Database stored in a text file with custom syntax
- Each record has a name that has to be unique
  - $\Rightarrow$ Naming convention see DCS TDR draft
- Each record has a type (analog input, binary output, etc.) and several fields (value, device connection, alarm thresholds, etc.)
- Record name + field name $\Rightarrow$ process variable (PV)
- Usually one record per device function (readout and control)
- Records can reference other PVs, also from other IOCs

EPICS Device Support

- "Device Support" is like a device driver for EPICS
- EPICS core and device support interact through defined interface
- For many devices, pre-built solution available
  - StreamDevice for everything controllable via serial interface
  - devSNMP for devices understanding SNMP
  - devModbus for devices using the Modbus protocol

$\Rightarrow$ No programming work, just writing configuration files

- If nothing available, writing custom device support necessary
- EPICS easily extensible and properly documented:
  `https://epics.anl.gov/base/R3-15/6-docs/AppDevGuide/AppDevGuide.html`

- Integrate new hardware into the $\overline{\text{P}}$ANDA DCS:
  1. Find/write device support for the hardware
  2. Write EPICS configuration for the hardware

Organization of $\overline{\text{P}}$ANDA DCS

- $\overline{\text{P}}$ANDA DCS Group: Each subdetector and their DCS managers
- $\overline{\text{P}}$ANDA DCS Core Group: F. Feldbauer, T. Triffterer, A. Belias

- Each subdetector is responsible for its DCS partition
- DCS Core Group offers support (tutorials, lists of supported hardware, ...)

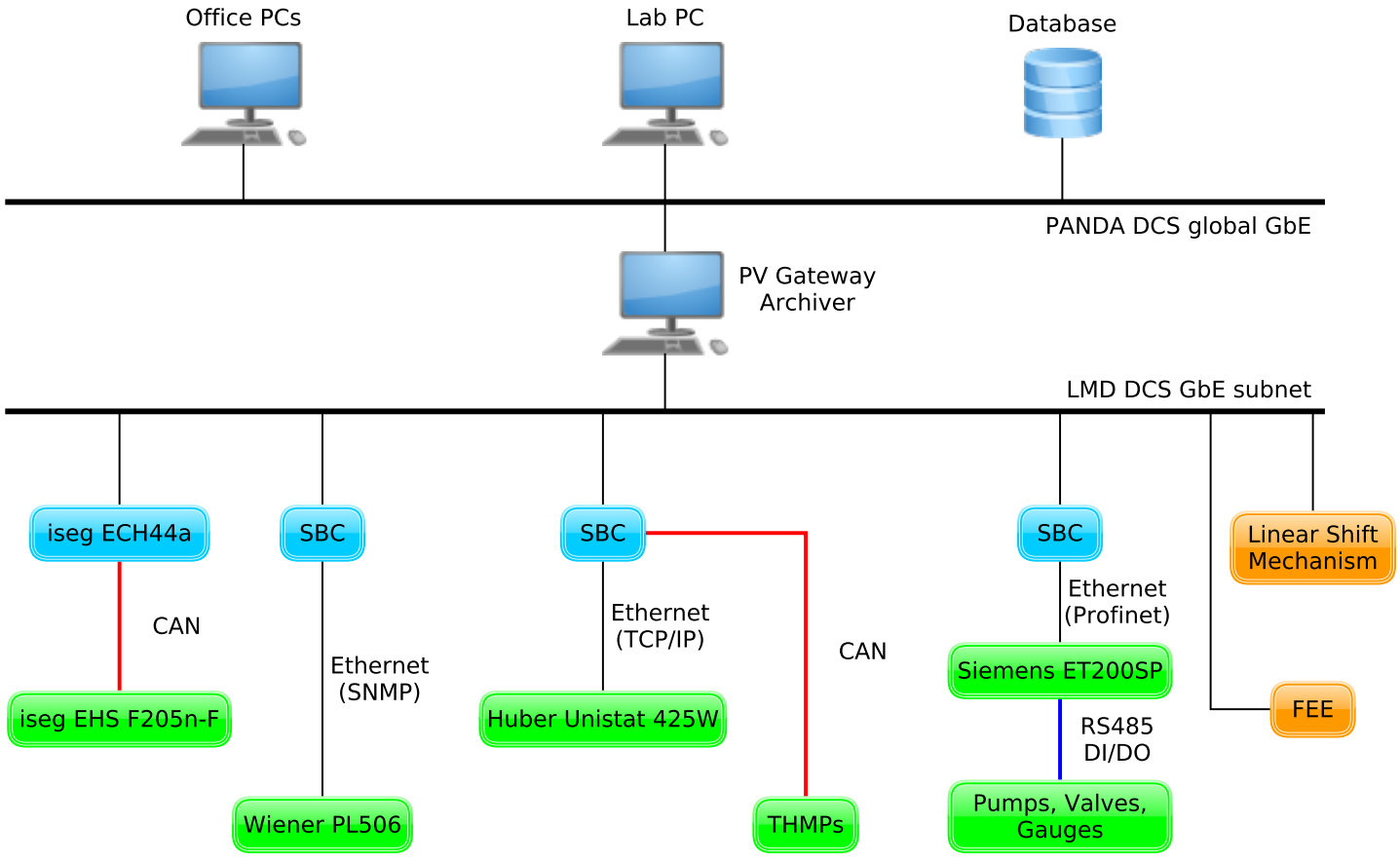
- Mailing list: panda-dcs@gsi.de
- Write to florian@ep1.rub.de to be added to the list

The End

Thank You for Your Attention!

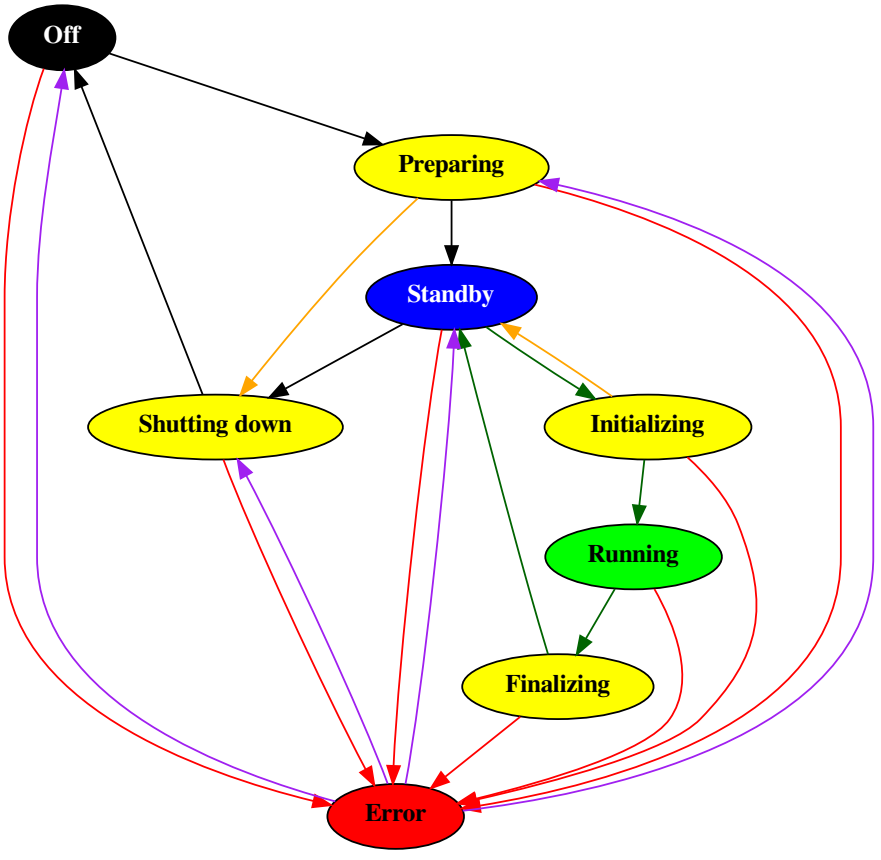
Backup Slides

Example: Luminosity Detector DCS partition



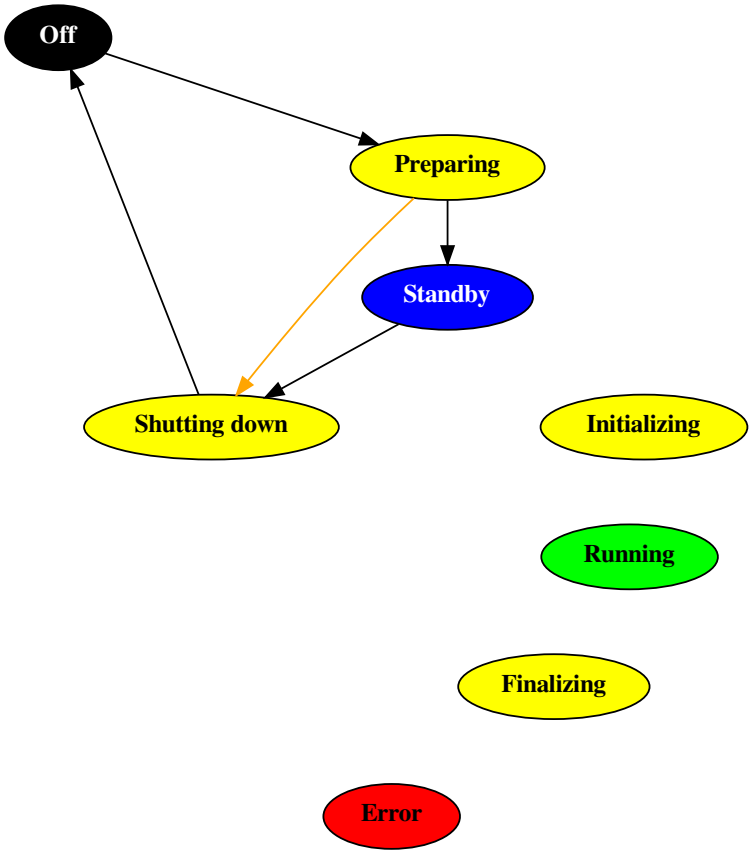
- IOCs running on Single Board Computer (SBC)
- Linear Shift Mechanism and FEE not yet implemented

Finite State Machine



Each subdetector needs to perfom defined actions
One state machine for global DCS (SL) and one for each subdetector (CL)

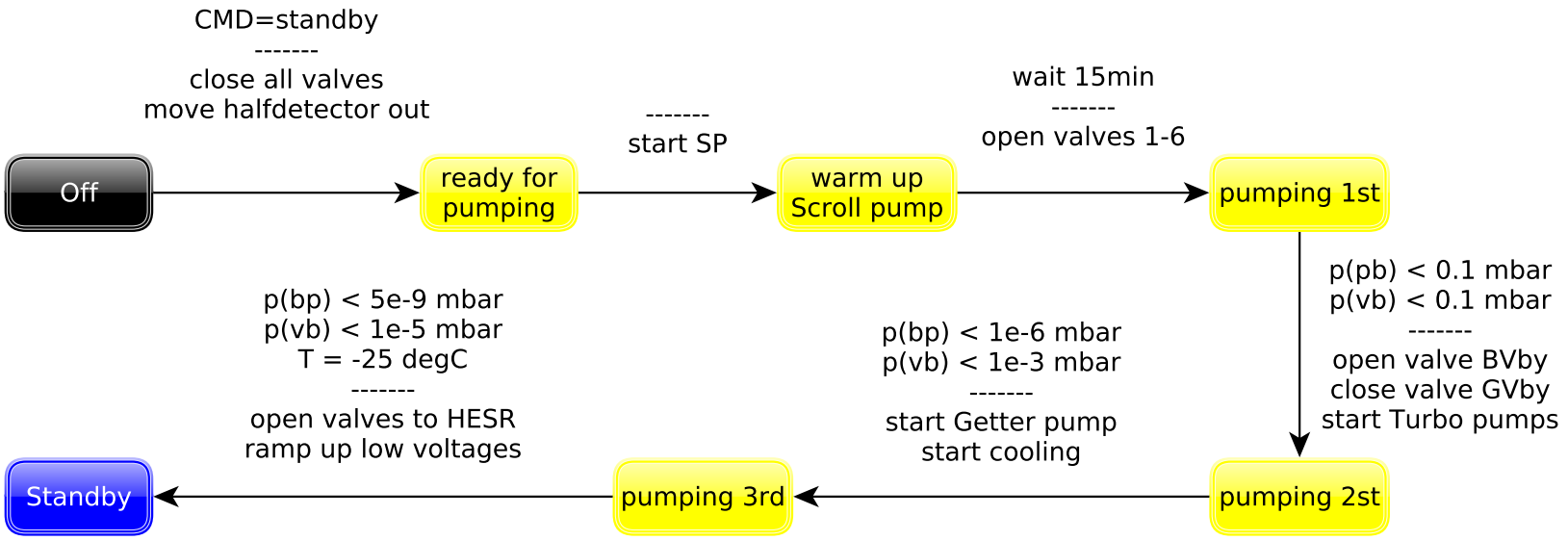
Finite State Machine



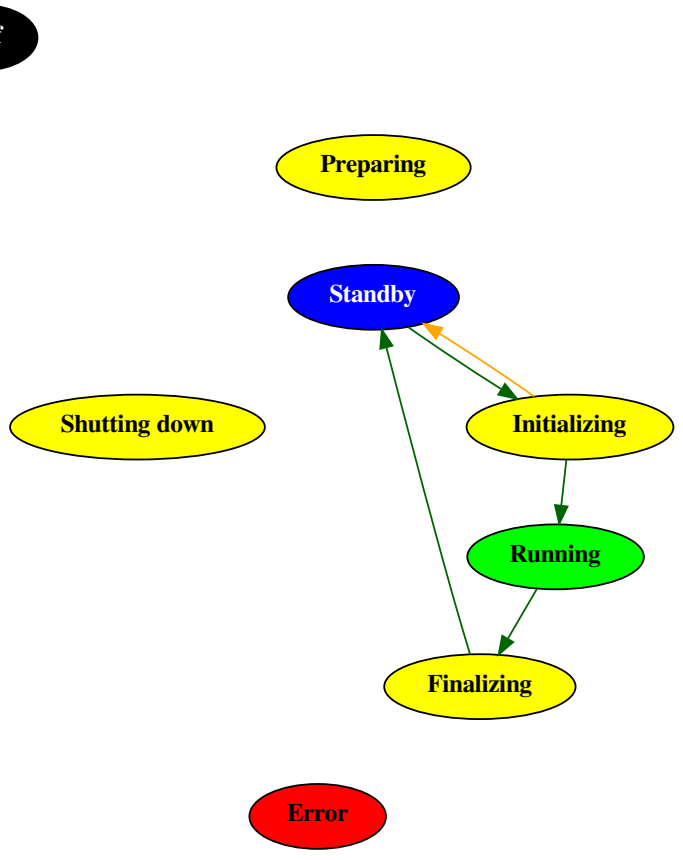
Start/End a run period (e.g. after maintainance)
Off $\rightarrow$ Preparing $\rightarrow$ Standby
Standby $\rightarrow$ Shutting down $\rightarrow$ Off

Finite State Machine

Example: Starting procedure of the Luminosity Detector

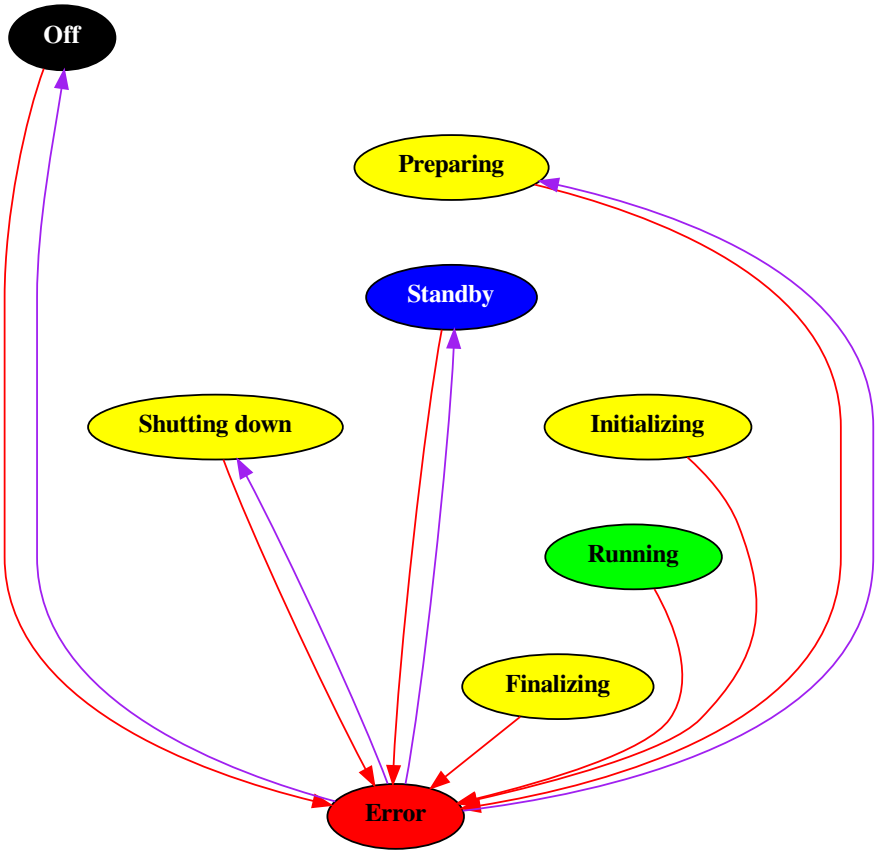


Finite State Machine



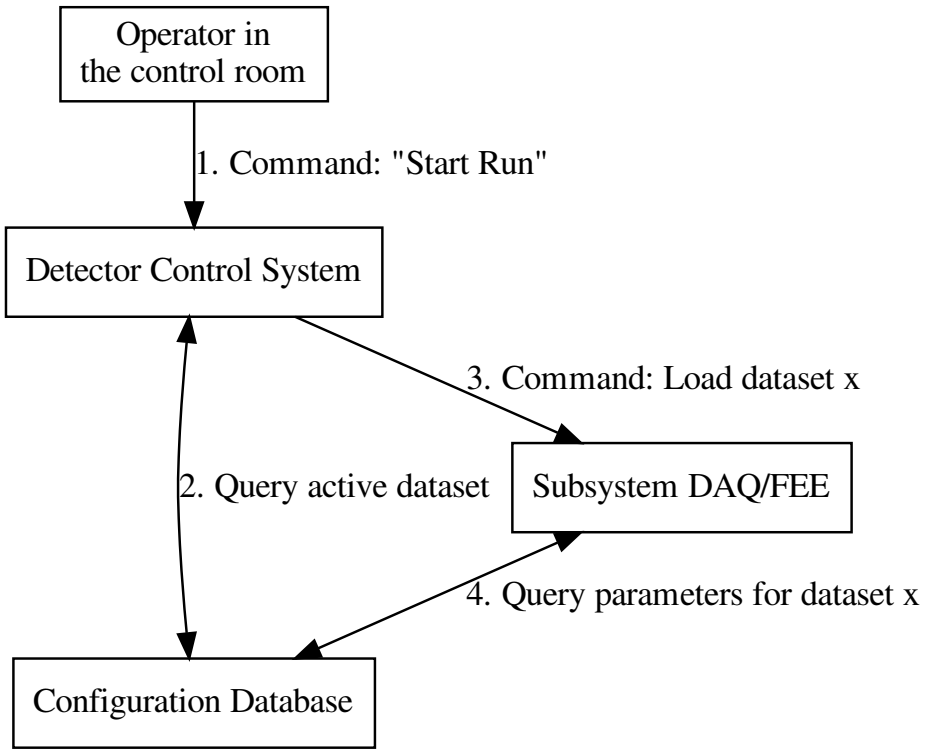
Typical procedure for data taking
Standby → Initializing → Running → Finalizing → Standby

Finite State Machine



In case of a problem Error state can be entered from any other state
After solving problem return to non-data-taking states

DCS-DAQ/FEE Interface



- DCS and DAQ/FEE configuration parameters stored in central database
- Configuration datasets get unique ID
- FEE configuration via SODAnet (not through EPICS)
- Shared responsibility between FEE and DCS groups