bmb+f

Großgeräte der physikalischen Grundlagenforschung

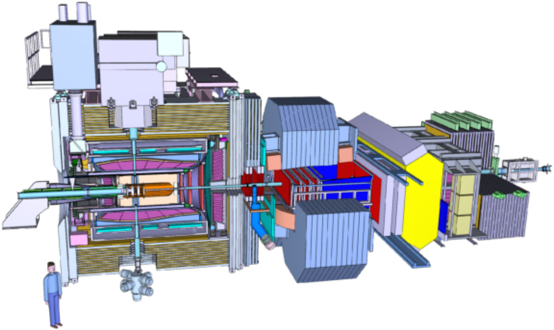
Structure of the $\overline{\text{P}}$ANDA Detector Control System

Tobias Triffterer

Ruhr-Universität Bochum – Institut für Experimentalphysik I

HK 20.7

The $\overline{\text{P}}$ANDA Experiment



- Key experiment at FAIR, located at High Energy Storage Ring (HESR)
- Fixed target experiment, antiproton beam
- $1.5\,\text{GeV}/c < p_{\bar{p}} < 15\,\text{GeV}/c$

Detector Control System (DCS)

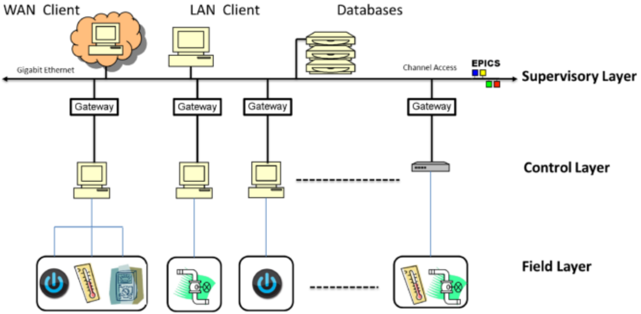
- One of the main components in the operation of a detector
- Purpose:
  - Ensure safe, correct and efficient operation
  - Contribute to collection of high-quality data
- Tasks:
  - **Monitor** the status of the detector (temperature, voltage, current, pressure etc.)
  - **Control** the front-ends and services of the detector (power supplies, chillers, heaters, valves, pumps etc.)
  - **Archive** all sensor data and commands for further use
    $\Rightarrow$ see HK 52.35
  - **Inform** the shift crew in case any parameter exceeds its threshold or any device malfunctions

Challenges for the $\overline{\mathrm{P}}$ANDA DCS

- $\overline{\mathrm{P}}$ANDA consists of 30 detector subsystems
- Subsystems are built and tested by different universities/institutes around the world, but must fit together once shipped to Darmstadt
- Subsystems must be able to run autonomously for testing/maintenance and collectively for beamtime

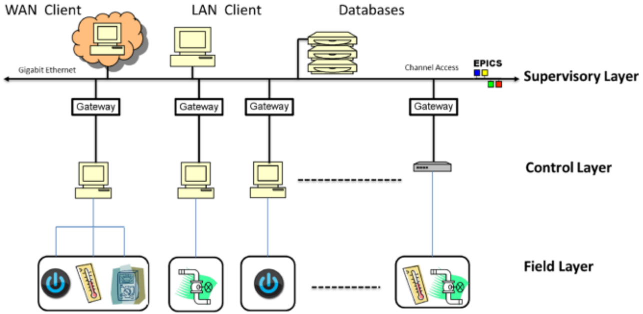
⇒ The $\overline{\mathrm{P}}$ANDA DCS must be scalable, distributed and extensible

Structure of the $\overline{\text{P}}$ANDA DCS



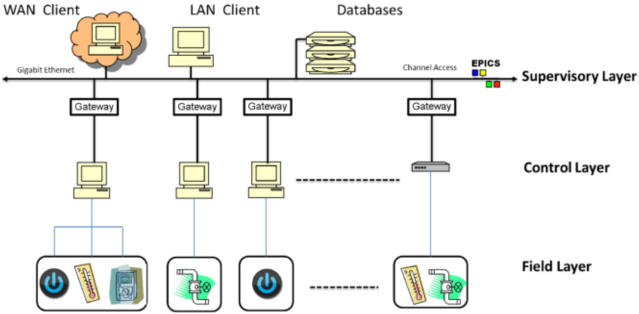
- $\overline{\text{P}}$ANDA DCS split intro three layers
  - Supervisory layer
  - Control layer
  - Field layer

Structure of the $\overline{\text{P}}$ANDA DCS



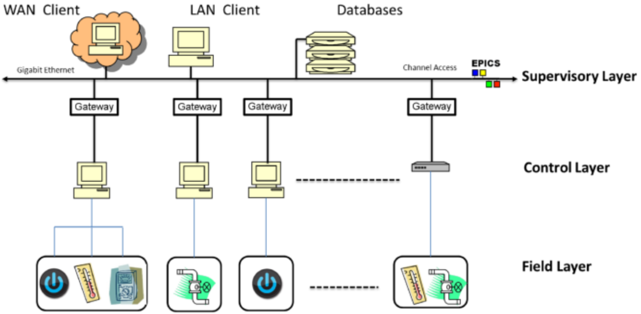
- Field layer:
  - Front-end devices, sensors, actuators, power supplies

Structure of the $\overline{\text{P}}$ANDA DCS



- Control layer:
  - Separate network for every subsystem of $\overline{\text{P}}$ANDA
  - Computers to control and operate devices on the field layer
  - Computers used: Normal PCs, single-board computers (e. g. Raspberry Pi), dedicated real-time hardware

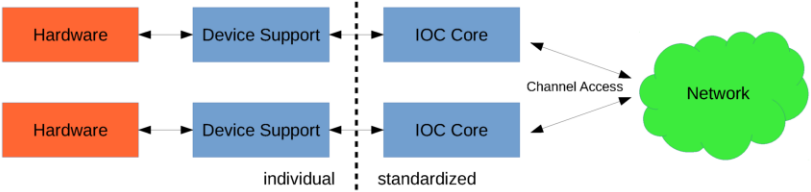
Structure of the $\overline{\text{P}}$ANDA DCS



- Supervisory layer:
  - Common system for all of $\overline{\text{P}}$ANDA
  - Graphical interface for shift crew
  - Central databases for archiving

Central software of the DCS: EPICS

- **E**xperimental **P**hysics and **I**ndustrial **C**ontrol **S**ystem
- Toolset to create individual control system
- IOCs (Input/Output Controller) building blocks of EPICS-based DCS
  - IOCs use "Device Support" to monitor/control hardware
  - IOCs communicate via custom protocol (Channel Access, abbrev. CA) to other IOCs and clients on the network
- Standardized access to all devices via CA
- Number of IOCs on the network virtually unlimited

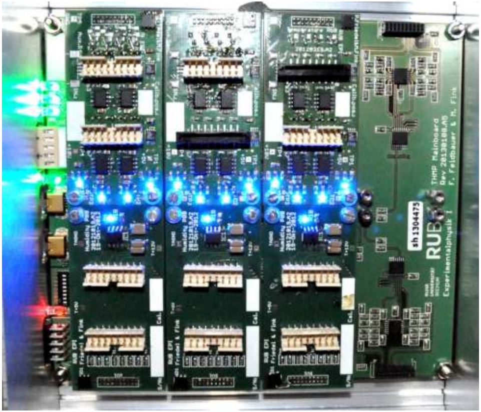


Device Support Examples

- Device Support for many devices available from the EPICS Community as open source software

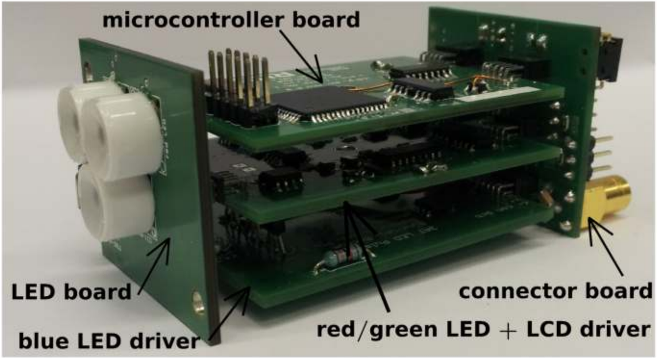
- For several devices, we programmed a custom EPICS device support
- Especially for devices created within $\overline{\text{P}}$ANDA

Device Support Examples



- $\text{THM}\overline{\text{P}}$: **T**emperature and **H**umidity **M**onitoring Board for $\overline{\textbf{P}}$ANDA

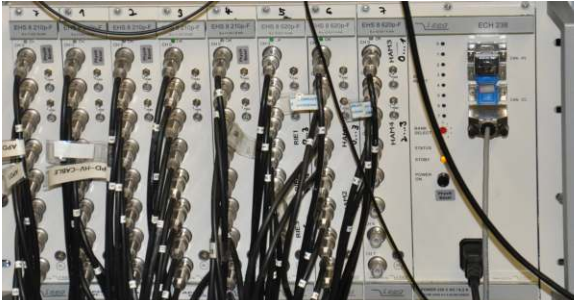
Device Support Examples



- LED Pulser for the $\overline{\text{P}}$ANDA electromagnetic calorimeter (EMC)

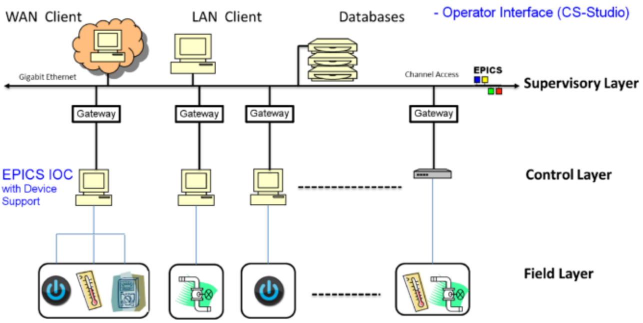
⇒ Test the readout chain of the EMC

Device Support Examples



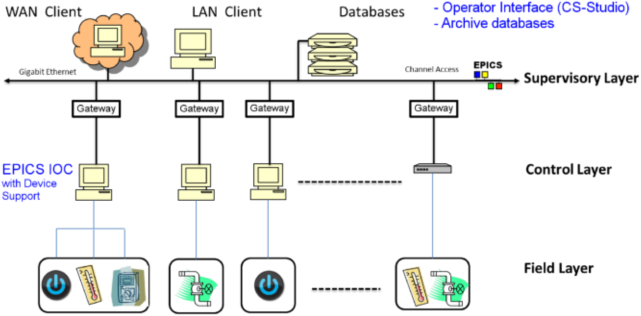
- Device Support for High Voltage power supplies from iseg Spezialelektronik GmbH
- Written by $\overline{\text{P}}$ANDA member Florian Feldbauer
- iseg now ships their devices with built-in EPICS support based on Florian’s work

EPICS Clients



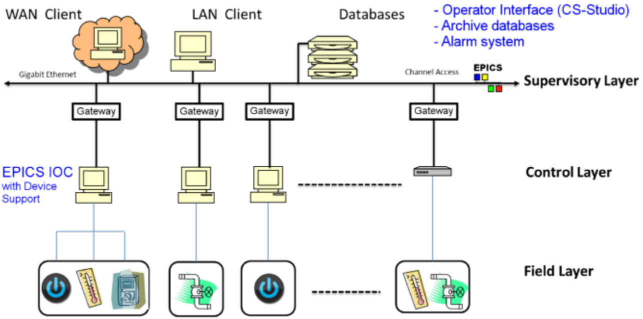
- Control System Studio (CS-Studio)
  - Graphical interface for operators
  - Based on Eclipse RCP

EPICS Clients



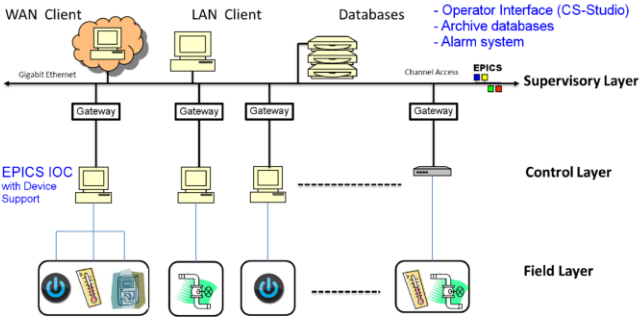
- Archive system
  - Archive all parameters in a database
  - Software to be used not yet decided
  - Options: CS-Studio Archive Engine, Cassandra PV Archiver, etc.

EPICS Clients



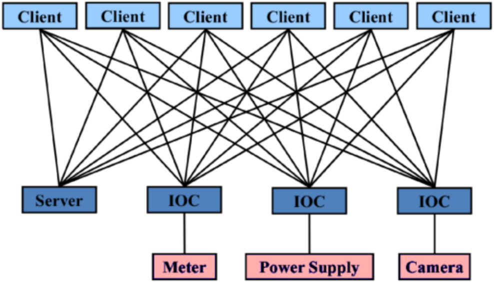
- Alarm system
  - BEAST (part of the CS-Studio project)
  - Alert shift crew if parameter exceeds threshold
  - Inform subsystem expert automatically

EPICS Clients



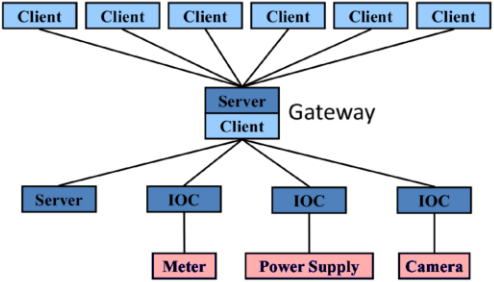
- Extensible: New software can easily be added using the EPICS CA library

Network Structure and Encapsulation



- Problem:
  - Structure network into proper subdivisions (subnets)
  - Protect against shift crew errors and attacks from outside

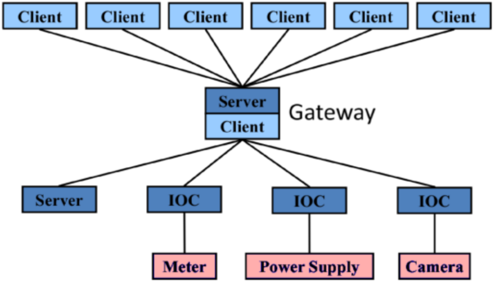
Network Structure and Encapsulation



- Separate subnet for every $\overline{\text{P}}$ANDA subsystem
- Additional subnet for the supervisory layer
- CA gateway regulates data flow accross subsystem boundaries and enforces access control

⇒ Provide proper encapsulation of each subsystem

Network Structure and Encapsulation



- Autonomous tests and maintenance of a subsystem
- A subsystem will not accept a potentially dangerous command from the supervisory layer (e. g. shift crew)
- Subsystem experts can access computers inside the subnet of the subsystem via VPN+SSH

Summary

- $\overline{\mathrm{P}}$ANDA DCS based on EPICS and CS-Studio
- $\overline{\mathrm{P}}$ANDA DCS split into layers and subnets

⇒ Divide complex problem into a set of easier problems that can be solved separately

- Components built at 64 universities/institutes from 19 countries must work together as a distributed but coherent control system.

Thank you for your attention!