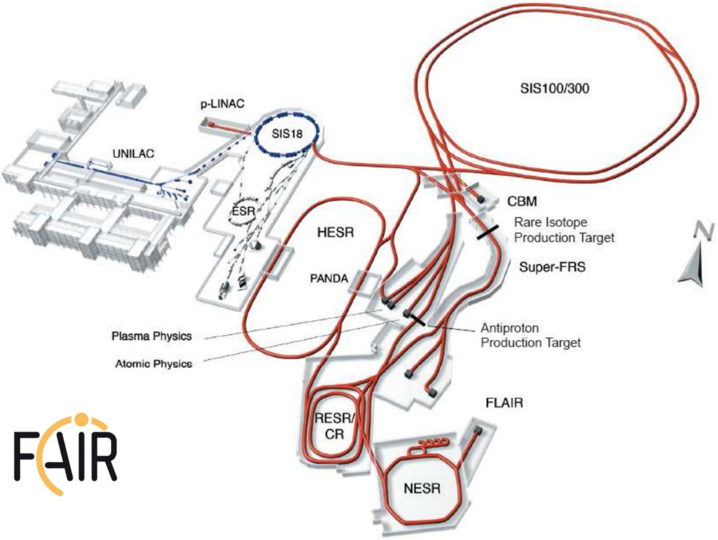
The Detector Control of the $\overline{\text{P}}$ANDA Experiment

Florian Feldbauer
on behalf of the $\overline{\text{P}}$ANDA Collaboration

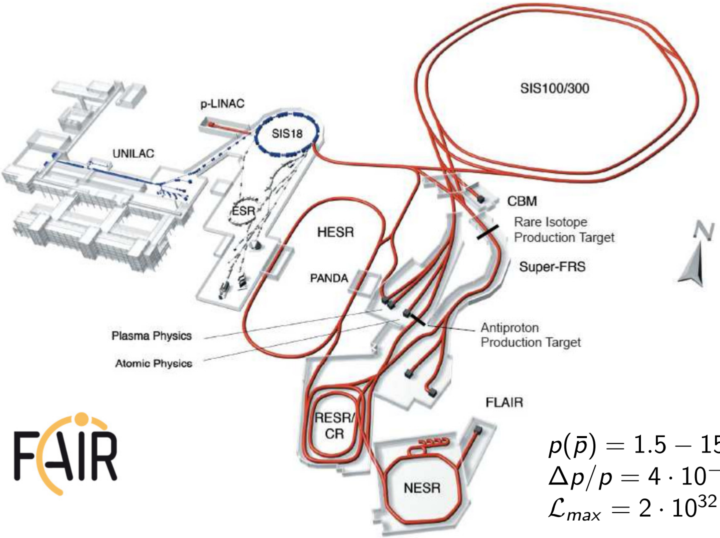
Helmholtz-Institut Mainz
Johannes Gutenberg-Universität Mainz

Instrumentation for Colliding Beam Physics
BINP, Novosibirsk
March 01, 2014

FAIR - Facility for Antiproton and Ion Research

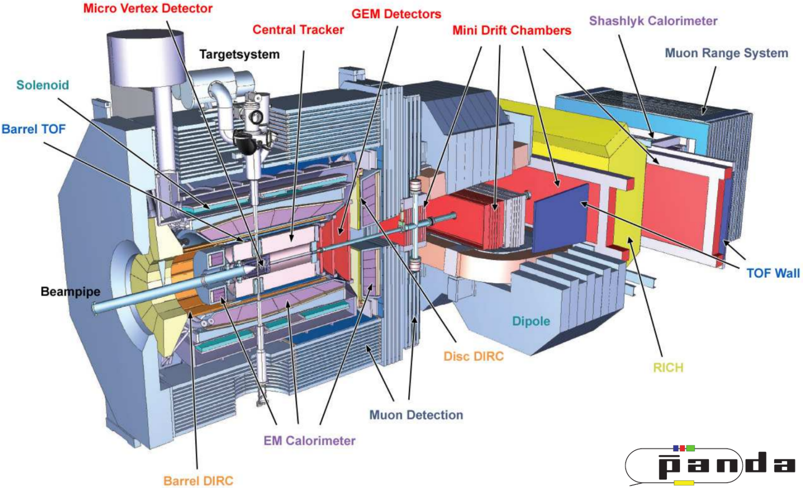


FAIR - Facility for Antiproton and Ion Research

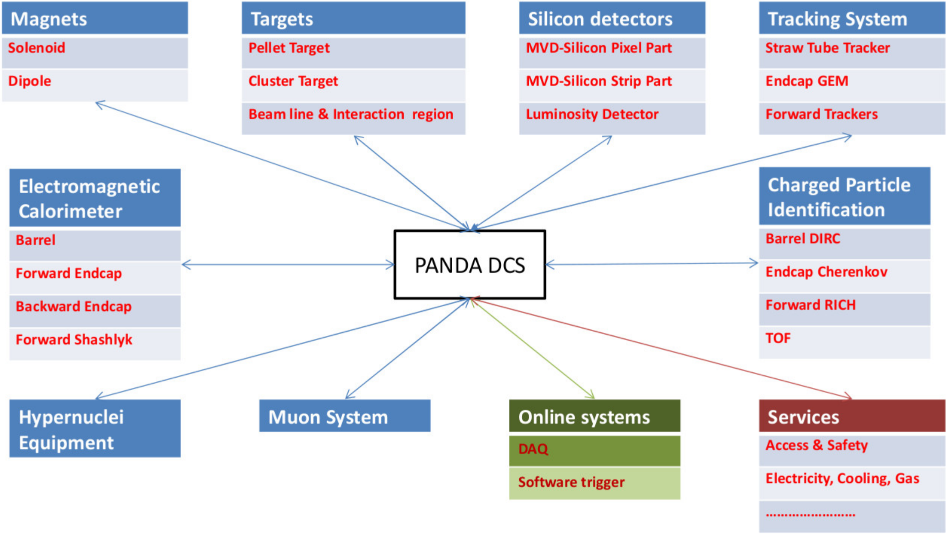


$$p(\bar{p}) = 1.5 - 15\,\text{GeV}/c$$
$$\Delta p/p = 4 \cdot 10^{-5}$$
$$\mathcal{L}_{max} = 2 \cdot 10^{32}\,\text{cm}^{-2}\text{s}^{-1}$$

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



$\overline{\mathrm{P}}$ANDA DCS Centralized View



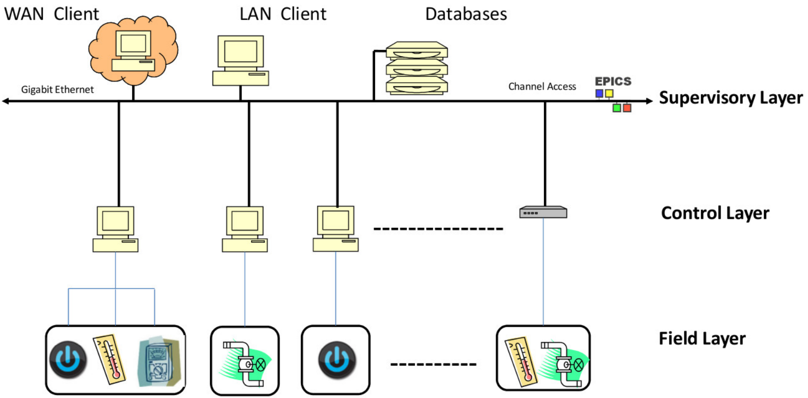
Requirements of $\overline{\text{P}}$ANDA DCS

(Some) Requirements of $\overline{\text{P}}$ANDA DCS:

- Scalable, modular
- Autonomous operation of each sub-detector (calibration, physics runs, maintenance)
- Archiving
- Alarm handling
- Non-expert operation
- Graphical UI

16 sub-detectors, 2 magnets, targets, beam
$\Rightarrow$ order of $2 \cdot 10^4$ "slow" channels expected

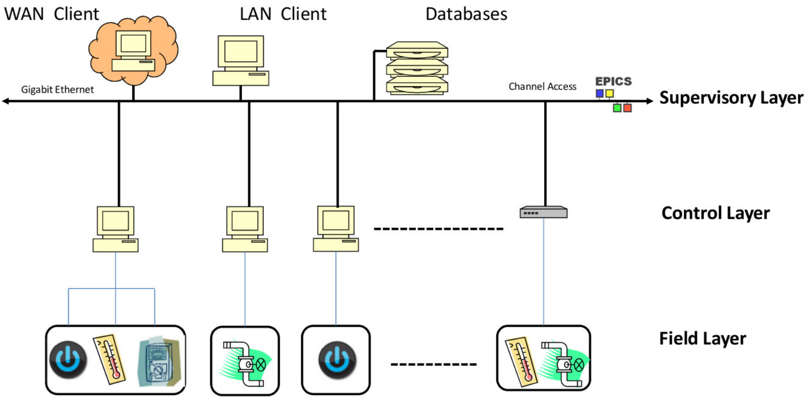
P̄ANDA DCS Overview



Field Layer (FL):

- Temperature monitoring, power supplies, valves,...
- Every device that is monitored or controlled

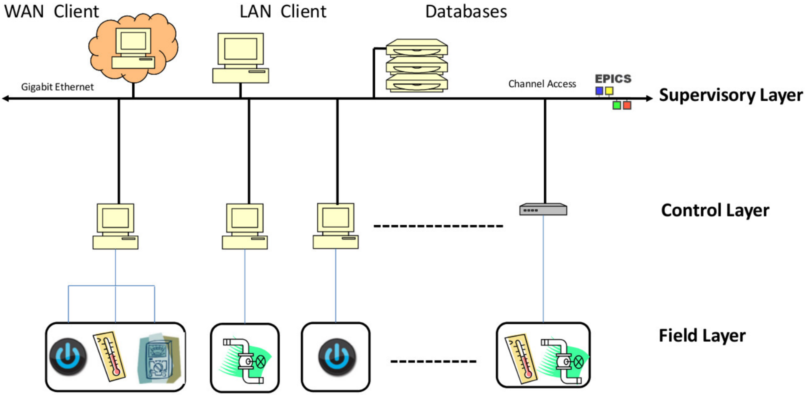
$\overline{\text{P}}$ANDA DCS Overview



Control Layer (CL):

- Input/Output controller communicating with devices in FL
- Used protocols RS232, RS485, TCP/IP, SNMP, CAN bus, ...
- Communication with Supervisory Layer via Ethernet

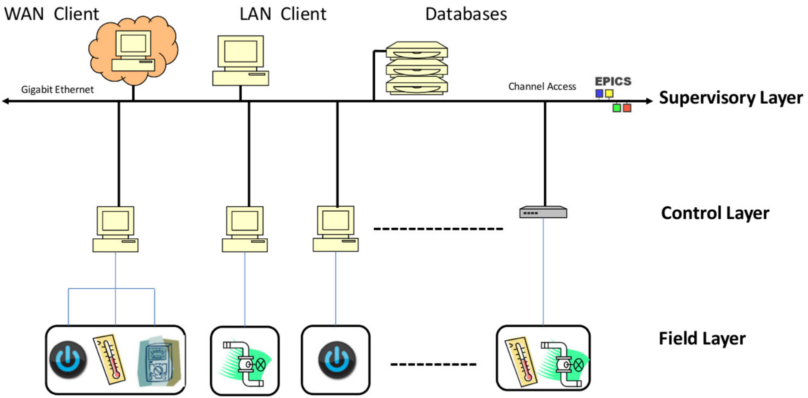
$\overline{\text{P}}$ANDA DCS Overview



Supervisory Layer (SL):

- Databases for data storage
- LAN Clients for graphical user interfaces

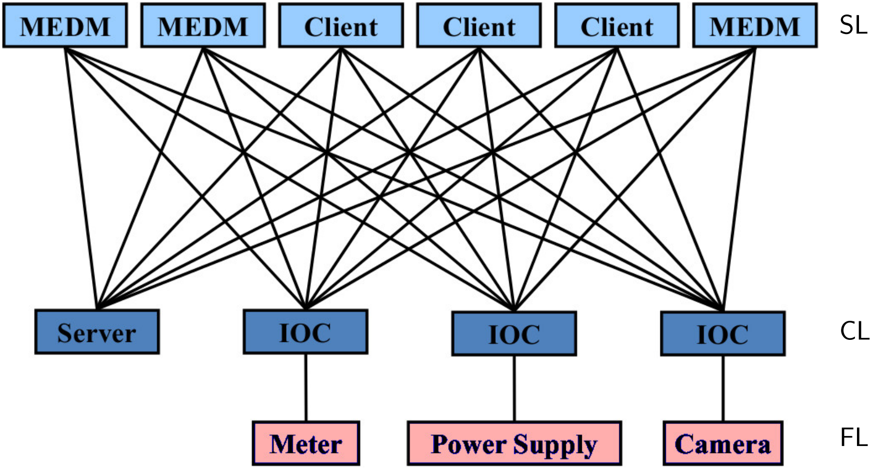
$\overline{\text{P}}$ANDA DCS Overview



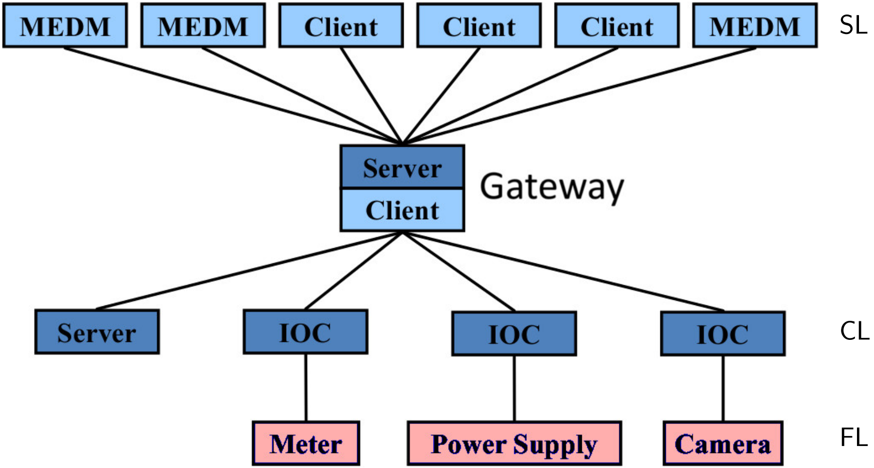
EPICS - **E**xperimental **P**hysics and **I**ndustrial **C**ontrol **S**ystem

- Network protocol based on UDP and TCP ("Channel Access")
- Decentralized architecture
- Freely scalable

EPICS Channel Access

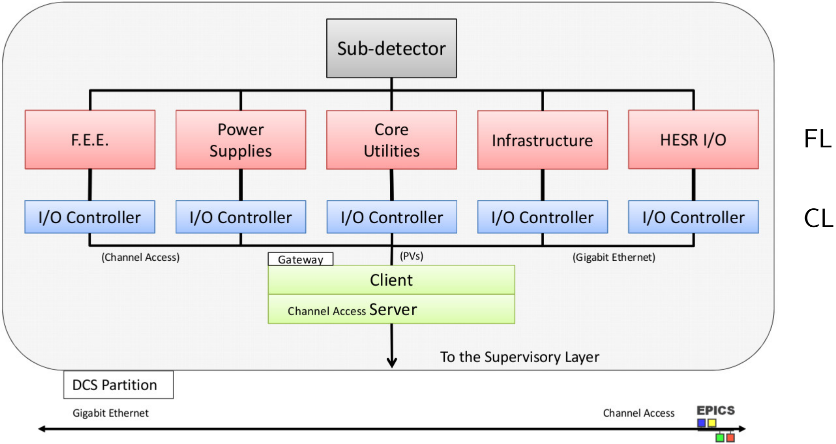


EPICS Channel Access



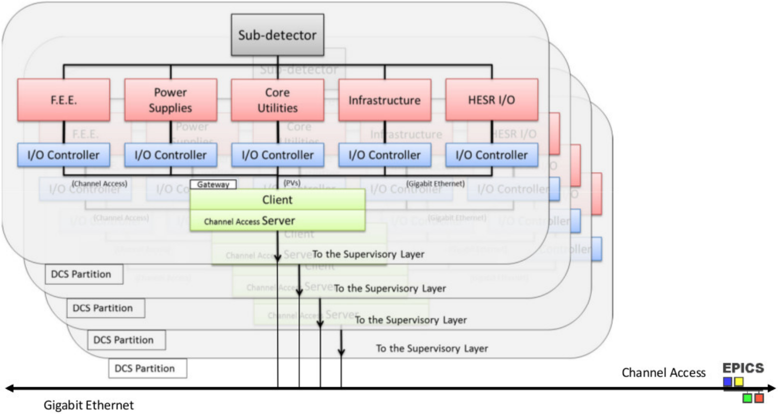
P̄ANDA DCS Architecture - Sub-detector

P̄ANDA DCS partitioning: Each sub-detector has it's own DCS Partition



$\overline{\text{P}}$ANDA DCS Architecture - Modularity

All partitions communicating with each other and supervisory layer via Gigabit Ethernet



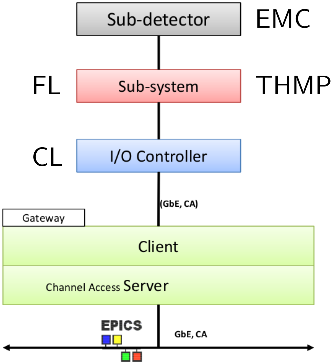
$\overline{\text{P}}$ANDA DCS Architecture - Sub-System

Example of sub-system

- Many sub-detectors of $\overline{\text{P}}$ANDA need temperature monitoring
- $PbWO_4$ scintillating crystals cooled down to $-25\,^{\circ}C$
- Light yield strongly depend on temperature (4%/K)

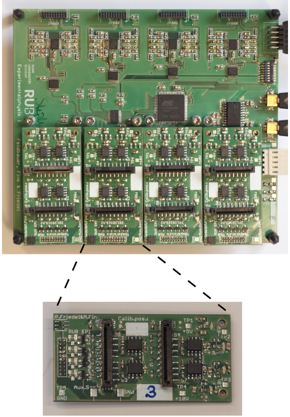
$\Rightarrow$ Temperature measurement with precision $\leq 0.05\,K$ over wide range needed

Example:



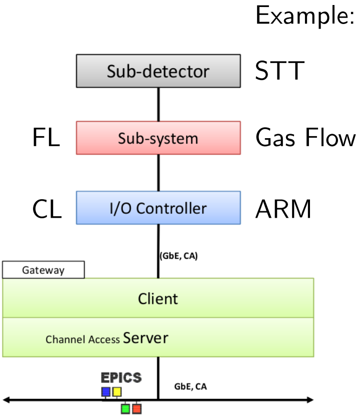
Temperature and **H**umidity **M**onitoring Board for **$\overline{\mathrm{P}}$**ANDA (THMP)

- Developed for $\overline{\mathrm{P}}$ANDA EMC by F. Feldbauer, M. Fink, and P. Friedel (Bochum University)
- Modular read out system for temperature, humidity, pressure, ...
- Mainboard with 8 piggyback boards
- 8 channels per piggyback board $\Rightarrow$ 64 channels per THM$\overline{\mathrm{P}}$
- 14 bit, 8 channel ADC
- Read out via CAN bus
- Temperature measurement:
  - Using PT100 resistance temperature sensors with 4 wire measurement
  - Constant current source with 1 mA
  - Measurement range $-50\,^{\circ}\mathrm{C} - +50\,^{\circ}\mathrm{C}$

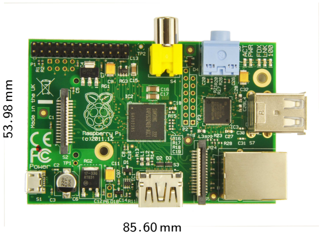


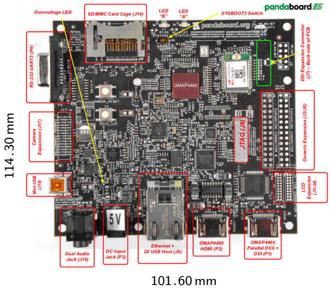
I/O Controller (IOC)

- Any device (PC, micro-controller board, FPGA board etc.) able to manage the I/O of the sub-system
- Usage of IOCs running on embedded Linux devices
- ARM Development Boards currently used:
  - ARMv6: Raspberry Pi Computer
  - ARMv7: PandaBoard ES



Linux Ready ARM IOC candidates





Raspberry Pi Computer

- ARM CPU, 700 MHz
- 512 MB RAM
- 10/100 Ethernet
- 2x USB 2.0, GPIO expansion header

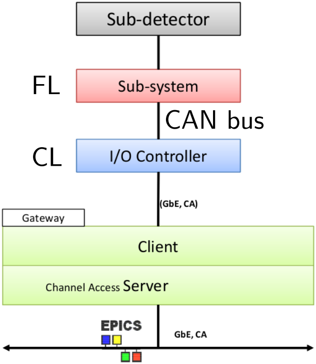
PandaBoard ES

- Dual-core ARM CPU, 1.2 GHz
- 1 GB DDR2 RAM
- 10/100 Ethernet, 802.11 b/g/n Wi-Fi
- 3x USB 2.0, RS-232, GPIO expansion header

Requirements for CAN Bus Interface

CAN bus as main interface to FL

- high data throughput needed
- Availability of hardware
- Easy maintainability of software
- Reliability
- Costs should be as low as possible
- Little space required
- Shielding
- Galvanic insulation



Available CAN Bus Interfaces

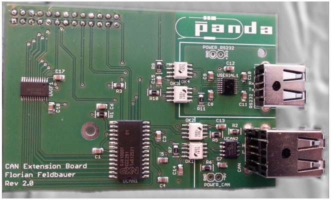
CAN bus interfaces available from Kvaser and Peak Systems:

high data throughput	+
very expensive ($\gtrsim$200 €)	–
Need PC for read out	
$\Rightarrow$ expensive	–
$\Rightarrow$ needs lots of space	–
Driver support from company?	
$\Rightarrow$ No easy maintainability of software!	–

- All CAN interfaces from Kvaser and Peak Systems use SJA1000 stand-alone CAN Controller
- Parallel interface with 8 mulitplexed address/data lines, 5 control lines

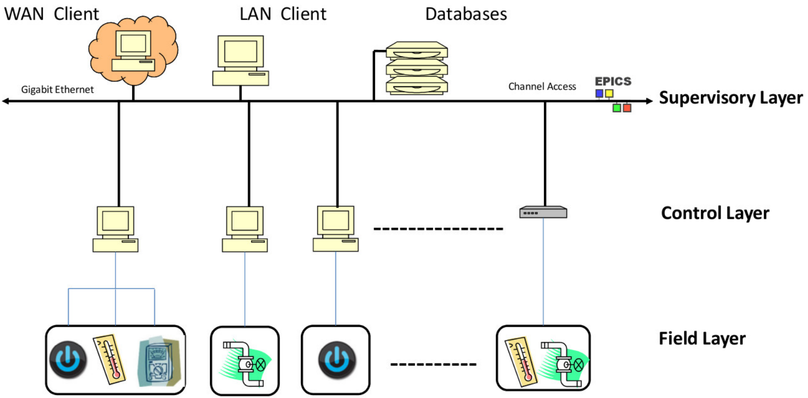
Are there other solutions?

- Idea: Connect SJA1000 directly to an embedded Linux device
- Extension board connected to GPIOs of Raspberry Pi computer



- Kernel module based on open-source linux driver from Peak Systems
- Data throughput/performance $\sim$ 1000 CAN frames/s sending/receiving at 125 kbit/s

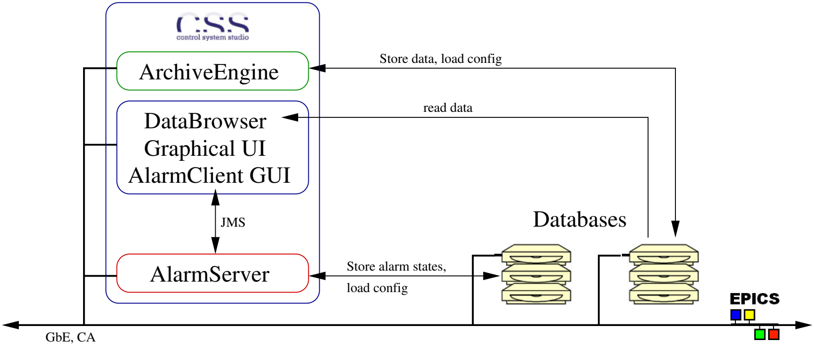
$\overline{\text{P}}$ANDA DCS Supervisory Layer



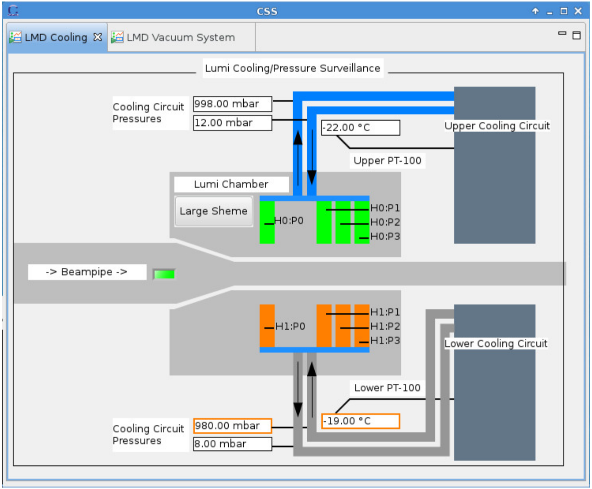
SL: $\overline{\text{P}}$ANDA specific version of Control System Studio (cs-studio)

- Collaboration between DESY, SNS, CLS, BNL, ITER, ...
- Toolkit based on Java and Eclipse RCP
- Modular infrastructure

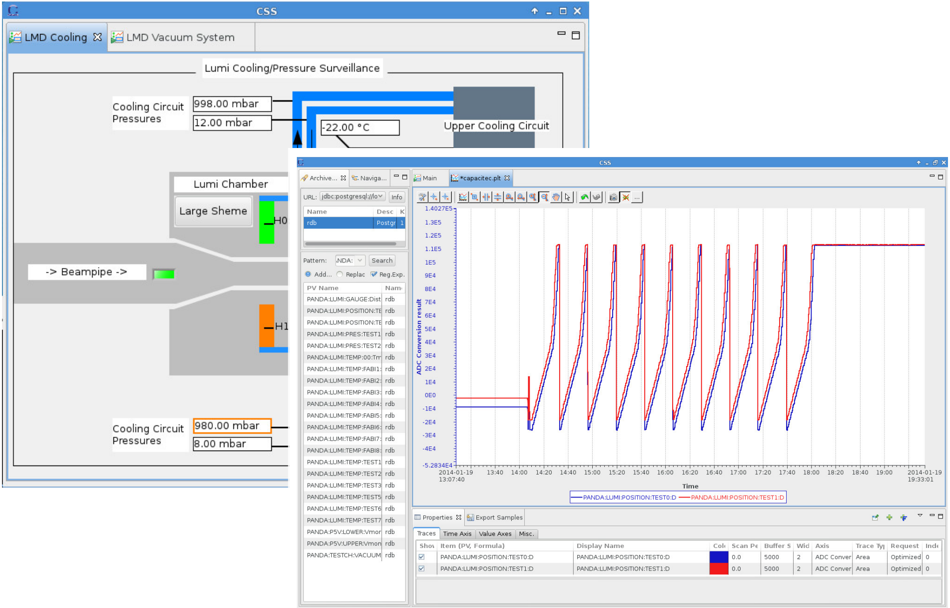
$\overline{\text{P}}$ANDA DCS Supervisory Layer



PANDA DCS Supervisory Layer



PANDA DCS Supervisory Layer



Summary

- $\overline{\text{P}}$ANDA DCS based on EPICS and cs-studio
- Modularized architecture
- I/O Controller running on embedded Linux devices
- EPICS CA Gateway to reduce network traffic
- THMP for temperature measurement with high precision
- High performance, low cost CAN bus interface for Raspberry Pi

BACKUP

Ultra-Thin PT100 Sensors

- Using polyimide foil coated with copper
- Etching traces with 1 mm pitch on polyimide foil as cable
- Using platinum wire with ø 25 μm
- Coating copper pads of cable with silver/gold
- Silver-plated conductor adhesive used to connect platinum wire to cable
- Using self-adhesive polyimide foil for insulation
- $\Rightarrow$ 70 μm thick cable/sensor