DCS Communication Software for the ALICE TPC Front-end Electronics

Matthias Richter

Department of Physics and Technology, University of Bergen, Norway
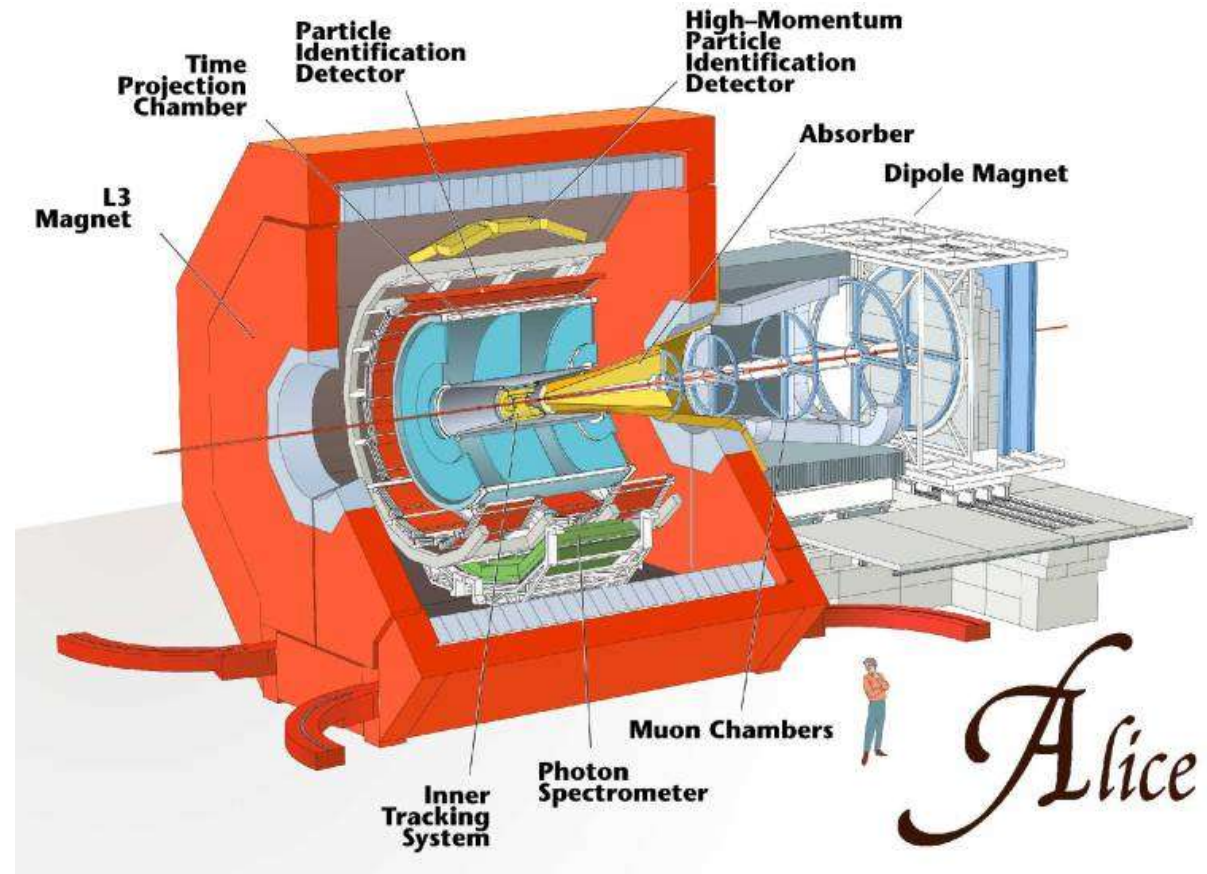
for the ALICE Collaboration

11th Workshop on Electronics for LHC and future Experiments,
Heidelberg, September 12-16th , 2005

Outline

- TPC Front-end Electronics

- Control system architecture

- Communication Software
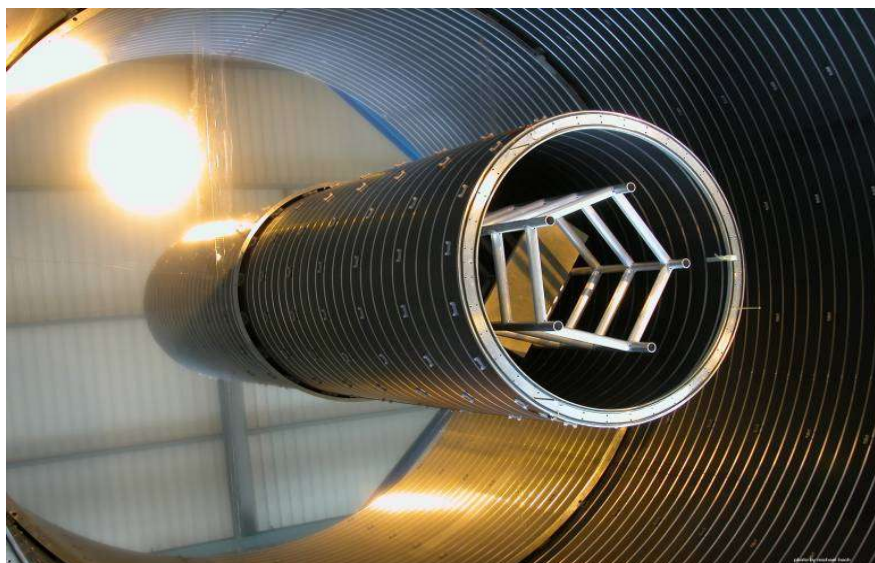
- Integration tests

- Summary

Time Projection Chamber

- main tracking detector

- detection of charged particles by ionization of the gas volume

- provides particle id and momentum

- 2-dimensional read-out at end-caps, drift time gives 3rd coordinate
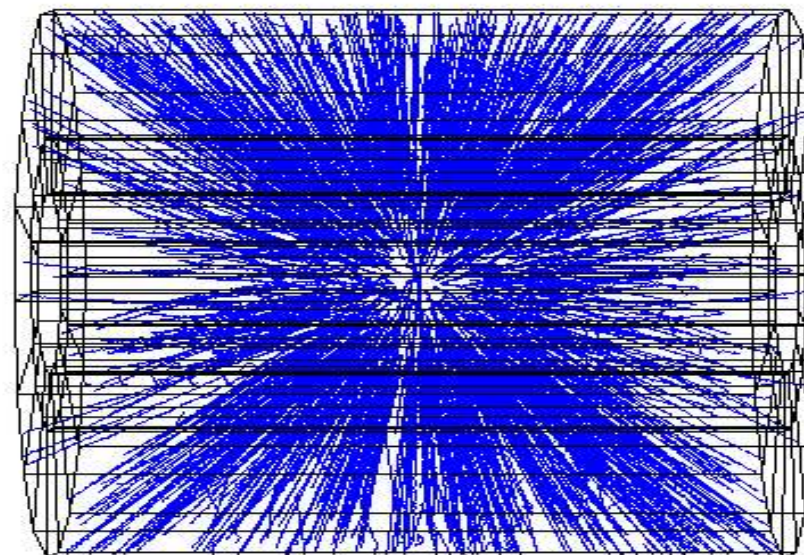
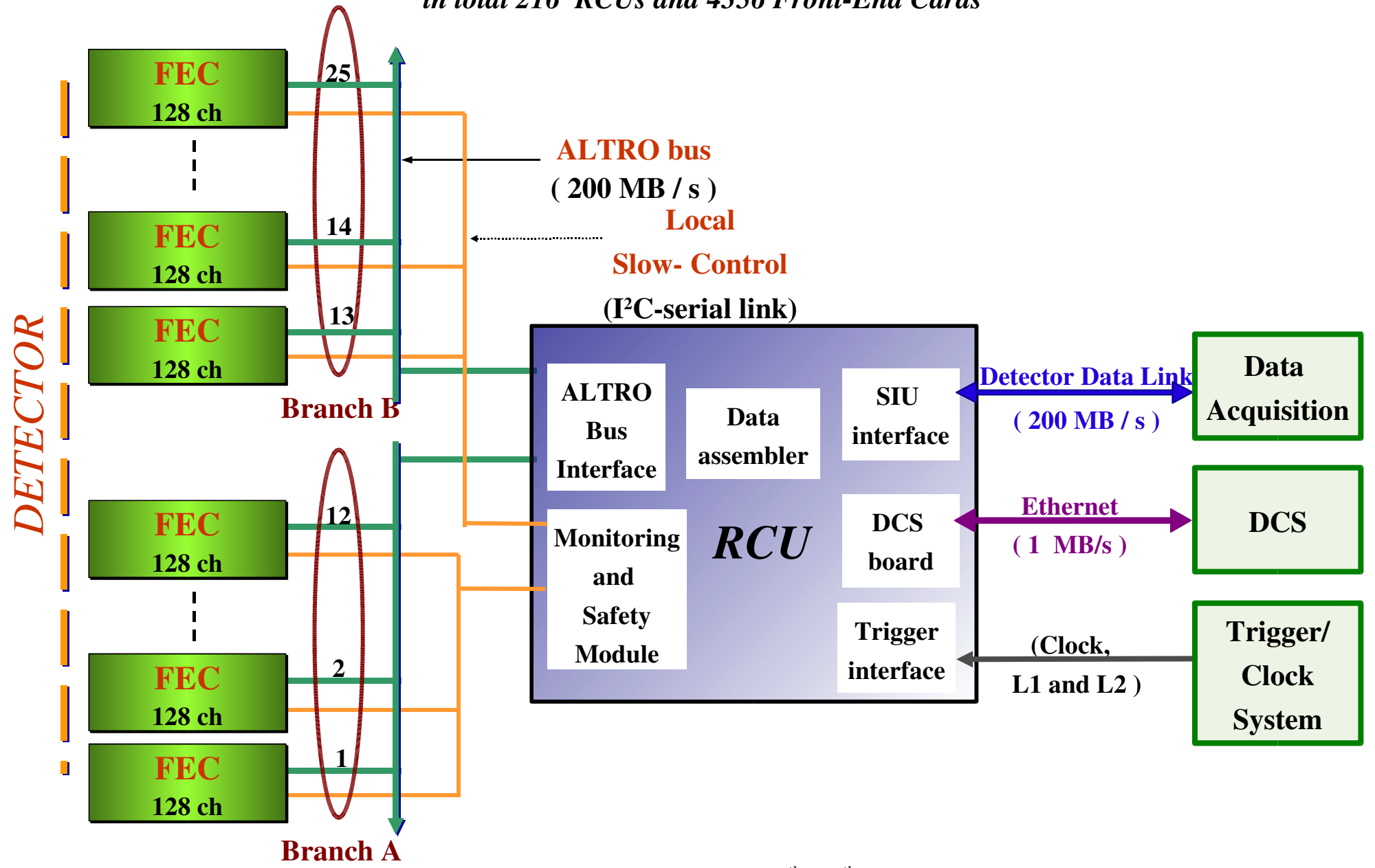

simulated low multiplicity event

TPC Field cage

- 2 x 18 sectors

- 4356 Front-End Cards, serving roughly 560000 channels

- designed for $dN_{ch}/d\eta=8000$: **20000** tracks

TPC Front-end Electronics

36 TPC Sectors, served by 6 readout subsystems, Readout Control Unit (RCU)
in total 216 RCUs and 4356 Front-End Cards

FEC 128 ch

25

14

13

Branch B

12

2

1

Branch A

DETECTOR

ALTRO bus
(200 MB / s)
Local
Slow- Control
(I²C-serial link)

ALTRO Bus Interface

Data assembler

SIU interface

Monitoring and Safety Module

RCU

DCS board

Trigger interface

Detector Data Link
(200 MB / s)

Data Acquisition

Ethernet
(1 MB/s)

DCS

(Clock, L1 and L2)

Trigger/ Clock System

Tasks of the Control system

- the presented system is a sub-system of the overall Detector Control System, which steers detector properties, e.g. voltages and cooling system

- preventing the system from data-flow interruptions

- **configuration**, **monitoring** and **controlling** the Front-End electronics

- upload of firmware to certain FPGA circuits in the system

- act upon certain conditions which may occur during operation, e.g. one FEC exceeds the temperature limit and has to be switched off

- detection and correction of errors caused by radiation damage

- access to the devices during experiment operation

- low rate event monitoring for development and debugging

Hardware Components
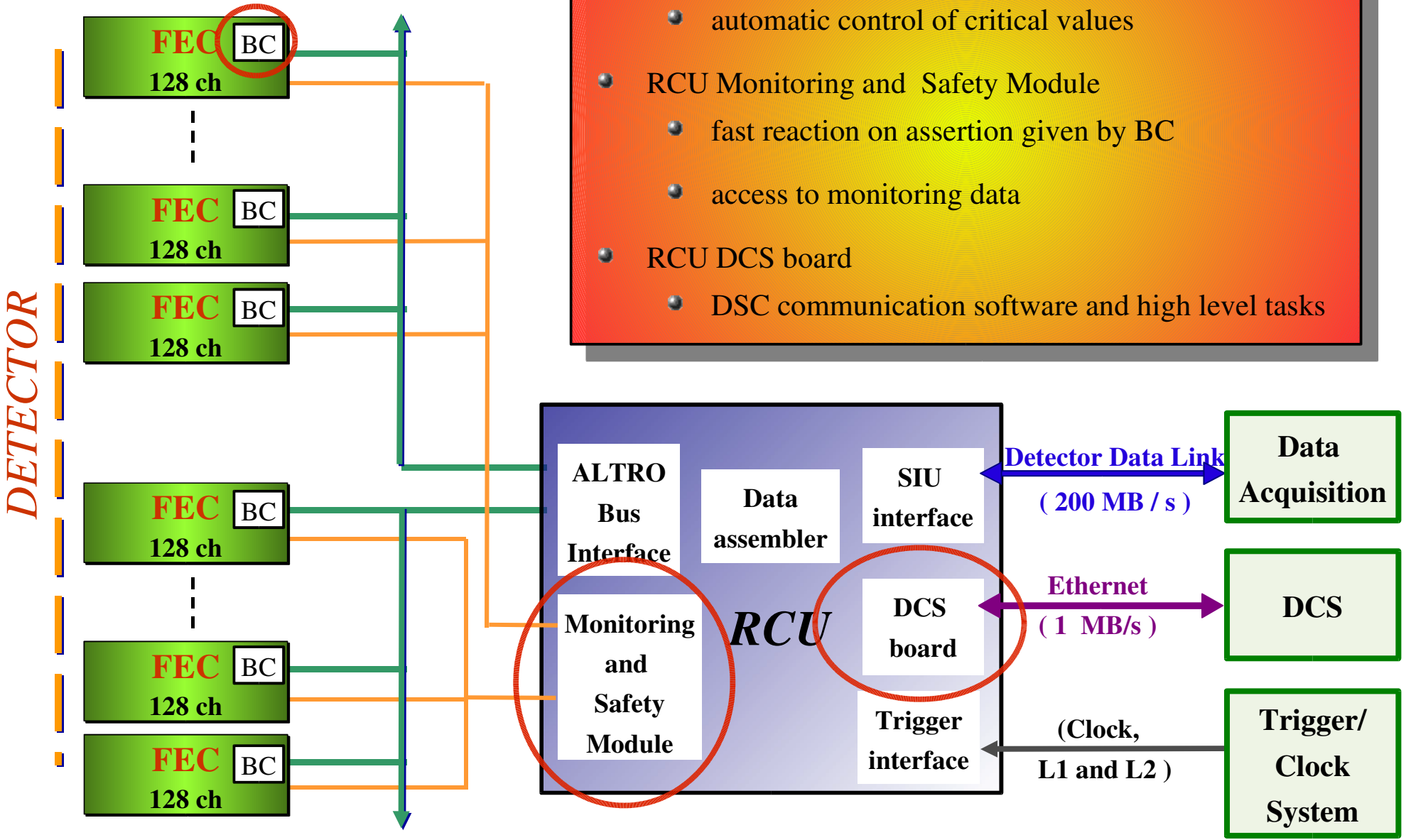
FEC Board Controller (BC)
- automatic control of critical values

RCU Monitoring and Safety Module
- fast reaction on assertion given by BC
- access to monitoring data

RCU DCS board
- DSC communication software and high level tasks

DETECTOR

FEC BC
128 ch

FEC BC
128 ch

FEC BC
128 ch

FEC BC
128 ch

FEC BC
128 ch

FEC BC
128 ch

RCU

ALTRO Bus Interface

Data assembler

SIU interface

Monitoring and Safety Module

DCS board

Trigger interface

Detector Data Link (200 MB / s)

Ethernet (1 MB/s)

(Clock, L1 and L2)

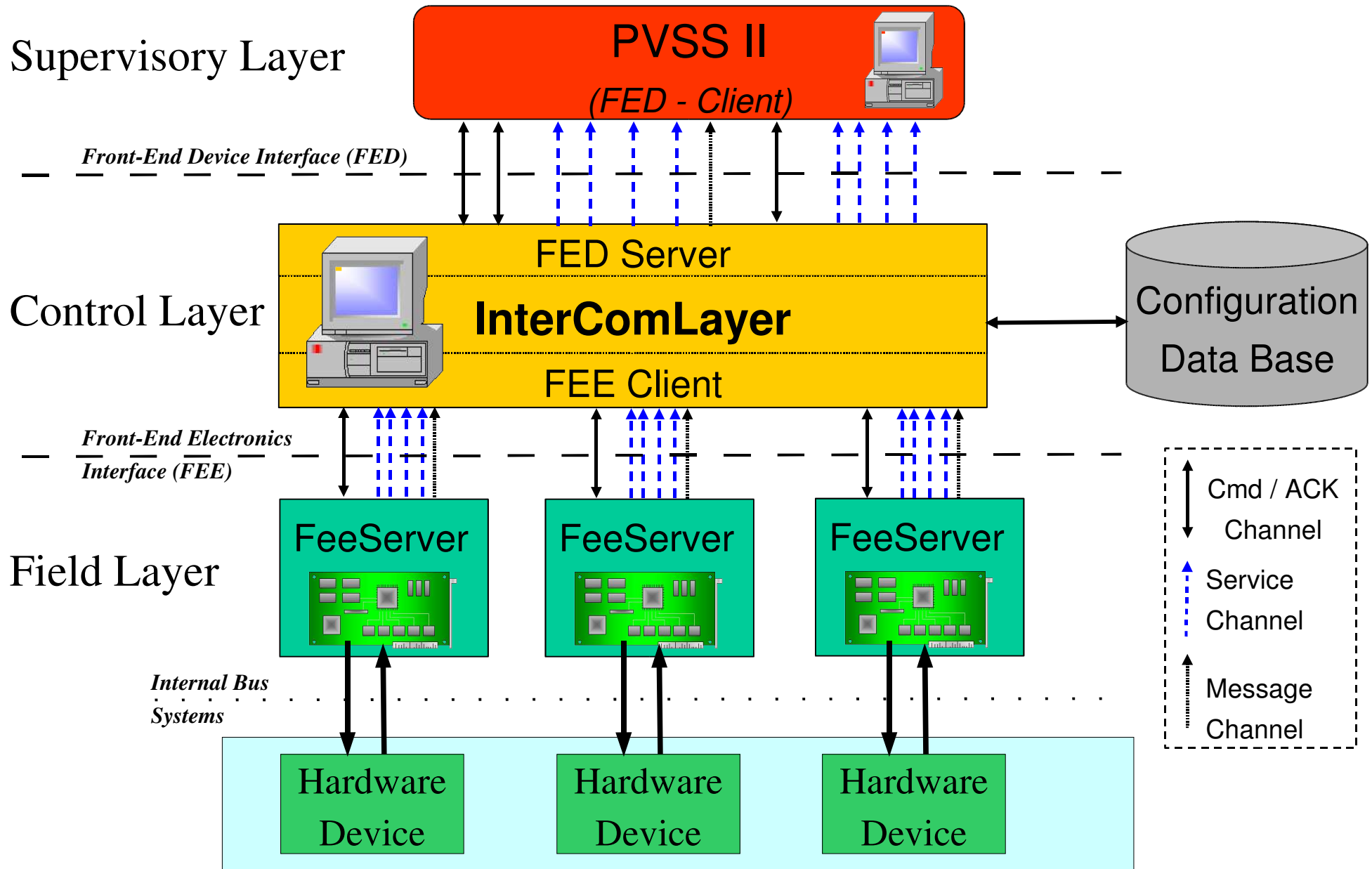Data Acquisition

DCS

Trigger/ Clock System

DCS board embedded computer

- single board computer used in several detectors in ALICE

- combines LINUX operating system with a Programmable Logic Device (PLD)

- device drivers impose abstraction layer between hardware and software

- provides low rate data readout path for debugging and monitoring purpose
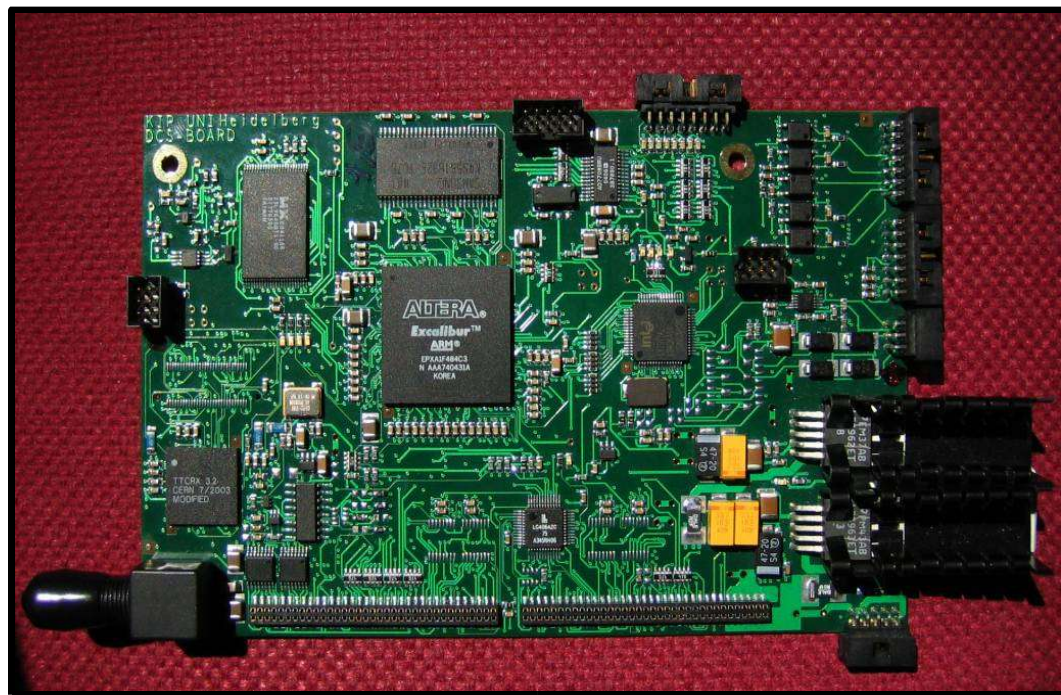
- Altera EPXA1 FPGA with
  - *32bit ARM* processor
  - 100k *PLD*
- 8 MB Flash RAM (radiation tolerant)
- 32 MB SDRAM
- Ethernet interface
- JTAG connector
- Analog to Digital Converter

Software architecture

Supervisory Layer

PVSS II
(FED - Client)

Front-End Device Interface (FED)

Control Layer

FED Server

InterComLayer

FEE Client

Configuration Data Base

Front-End Electronics Interface (FEE)

Field Layer

FeeServer

FeeServer

FeeServer

Internal Bus Systems

Hardware Device

Hardware Device

Hardware Device

Cmd / ACK Channel
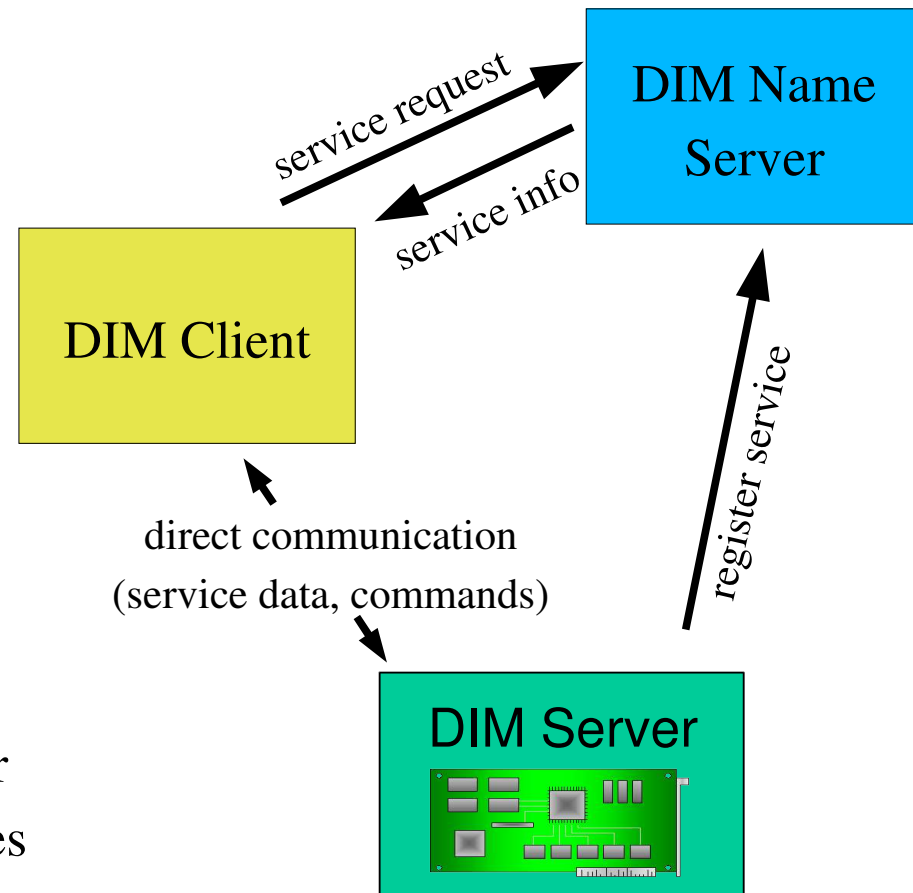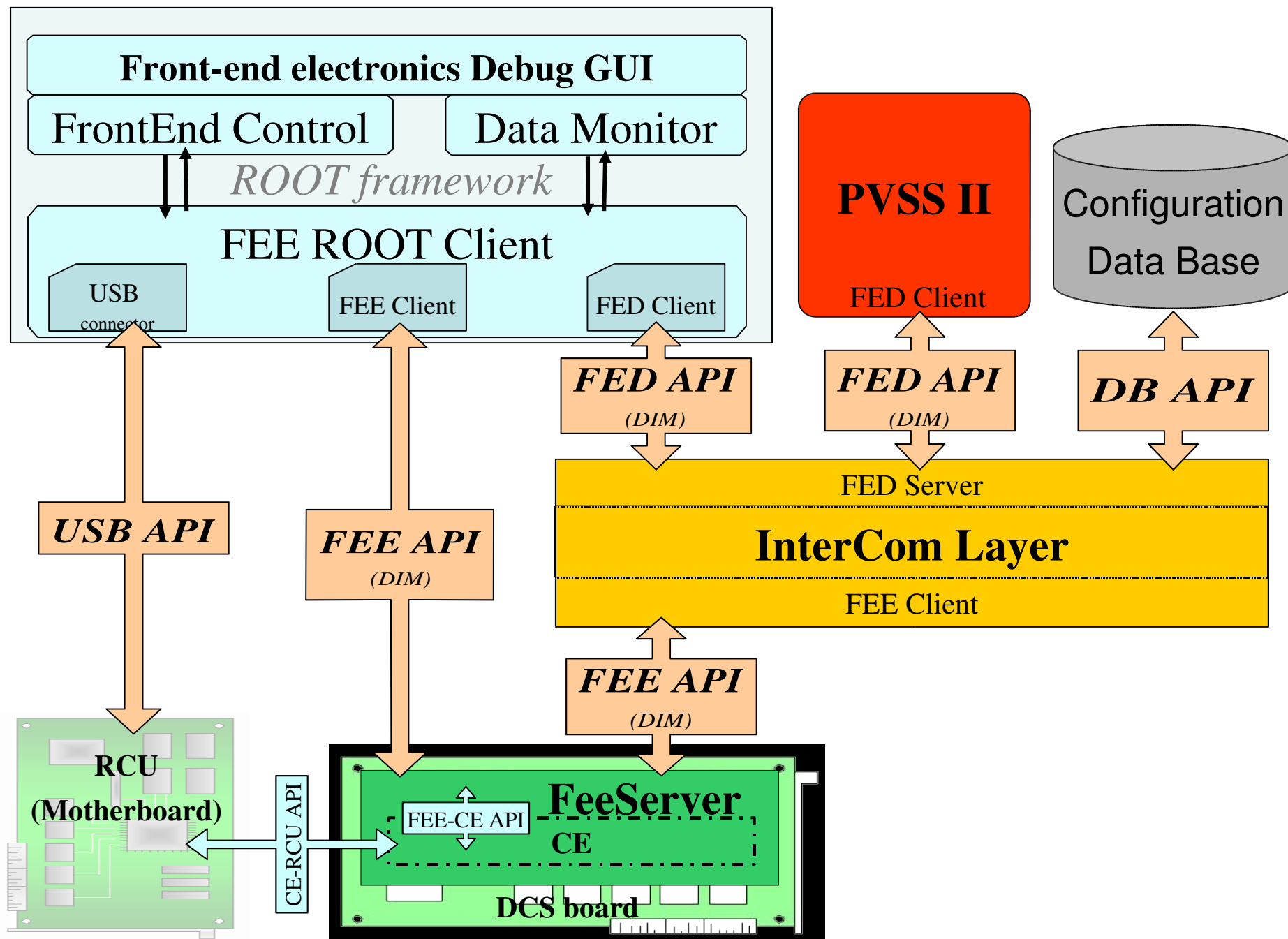
Service Channel

Message Channel

Working principle

- communication between functional layers based on Client-Server principle via *DIM* framework (Distributed Information Management System)

- a FEE subsystem is controlled by a Front-End-Electronics-Server (FeeServer)

  - running on the DCS board embedded computer

  - consists of a detector independent core and the ControlEngine(CE) carrying out device specific tasks

  - collects and publishes monitoring data

  - configures the Front-end electronics

- InterComLayer connects to many FeeServers

  - collects, filters and transports monitoring data

  - sends configuration data

  - connects to configuration data base

Communication Protocol DIM

- open source communication framework developed at CERN

- provides network transparent inter-process communication for distributed and heterogeneous environments

- Server – Client architecture

  - Server publishes services

  - Clients can subscribe to any service in the system

  - Server accepts commands sent by a client

- connection details are hidden from the user

- framework takes care of the byte ordering and handling of complex data structures

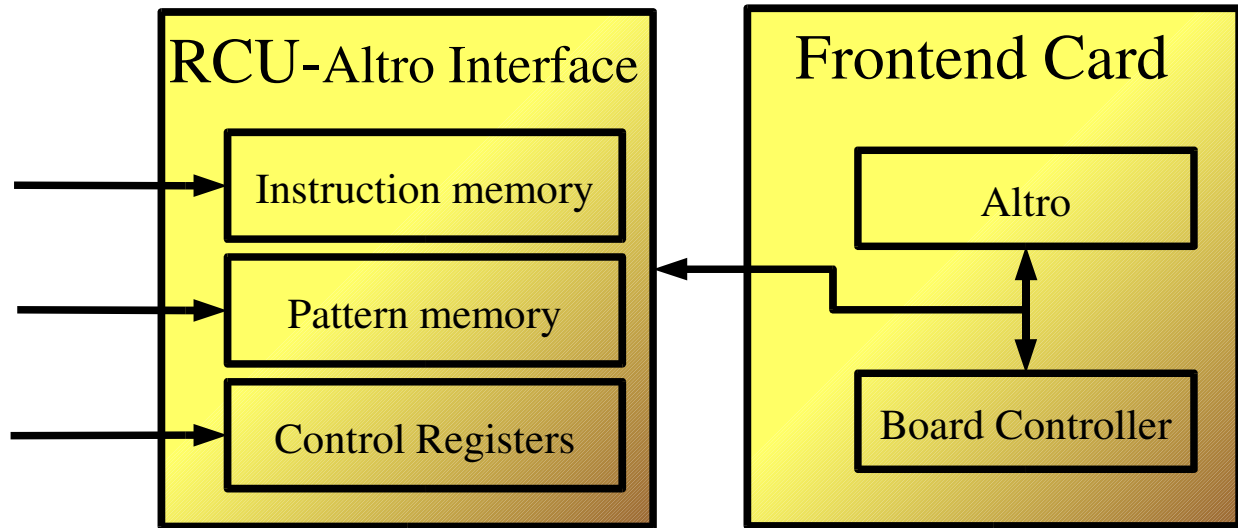- dedicated DIM Name Server takes control over all the running clients, servers and their services

DIM Name Server

service request

service info

DIM Client

register service

direct communication
(service data, commands)
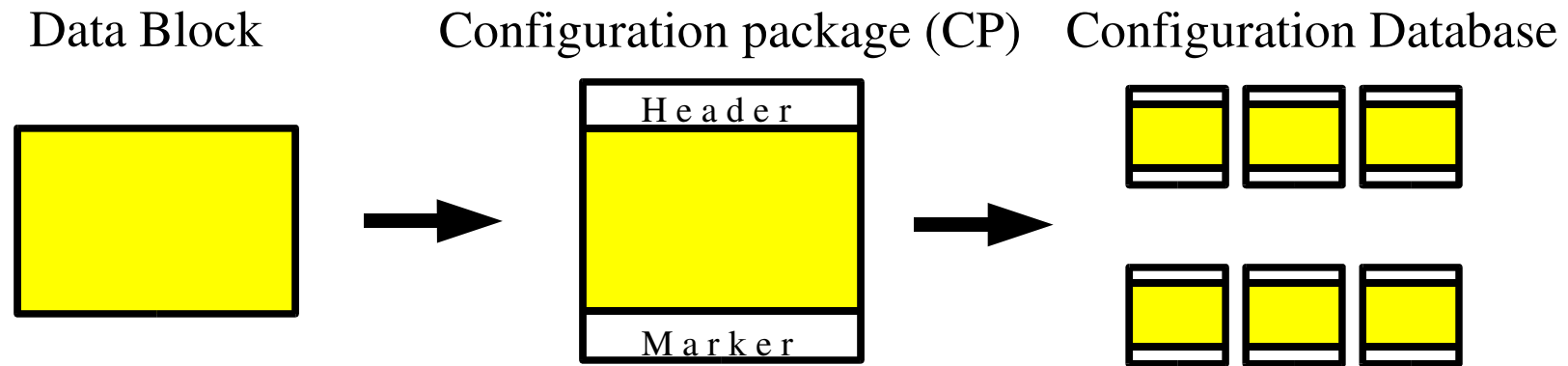
DIM Server

Development tools and APIs

FEE Configuration

Basic Sequence:

- write command sequence to instruction memory
- write extra data to pattern memory
- issue 'execute command'
- read result and status

RCU-Altro Interface
- Instruction memory
- Pattern memory
- Control Registers

Frontend Card
- Altro
- Board Controller

Content and format of configuration data is unknown to com. framework

Data Block   Configuration package (CP)   Configuration Database

Header

Marker

Data base API / Archiving

- The data is organized in programming blocks, the Configuration Database archives them and builds an index list

- an RCU configuration is defined by the **Package Descriptor (PD)**, which is a list or sequence of **Configuration Packages (CP)**

- a whole TPC configuration is a list of PDs for all RCUs, typical size 200kByte per RCU

- InterCom layer gets a full sequence from the Configuration Database/File structure and sends it to the DCS board where it is interpreted

TPC conf. #xxx   Configuration Database   Datastream
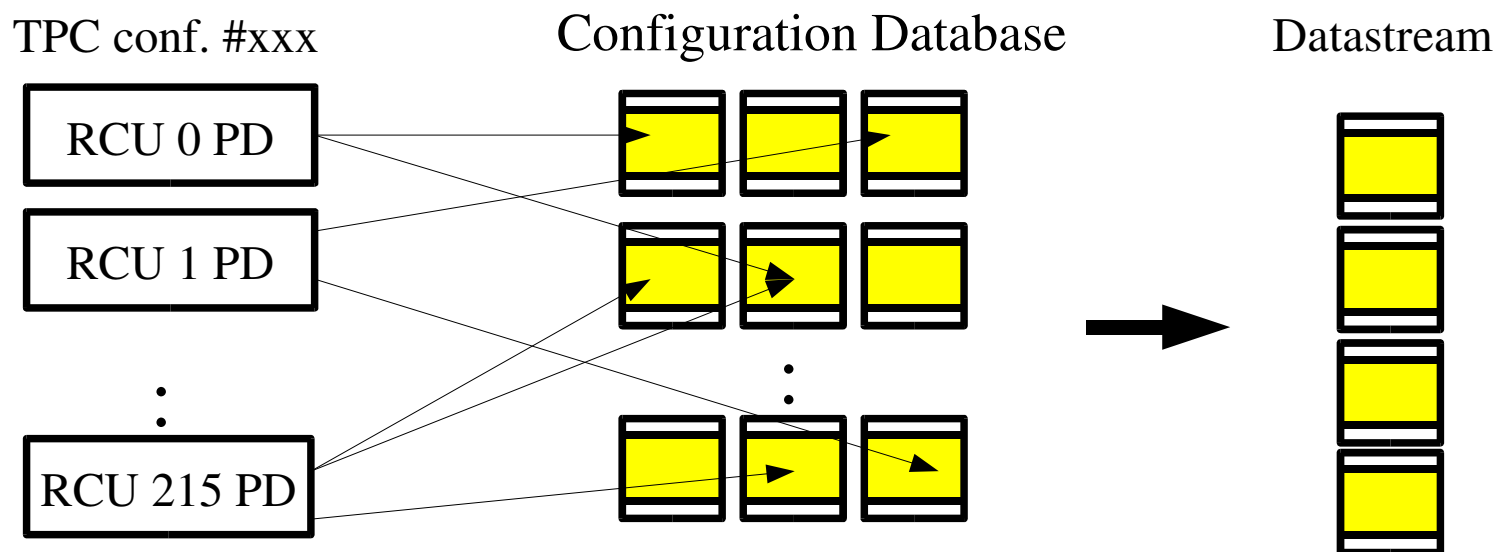
Integration tests

- several prototypes of different scale have been tested

- recent system test of realistic scale, excluding Supervisory Layer

- 1 RCU board with 1 DCS board and 9 Front-End Cards attached

- FeeServer published 10 data points per FEC and a few memory locations on the RCU

- InterComLayer running on a PC

- stable running of InterComLayer and FeeServer during several days
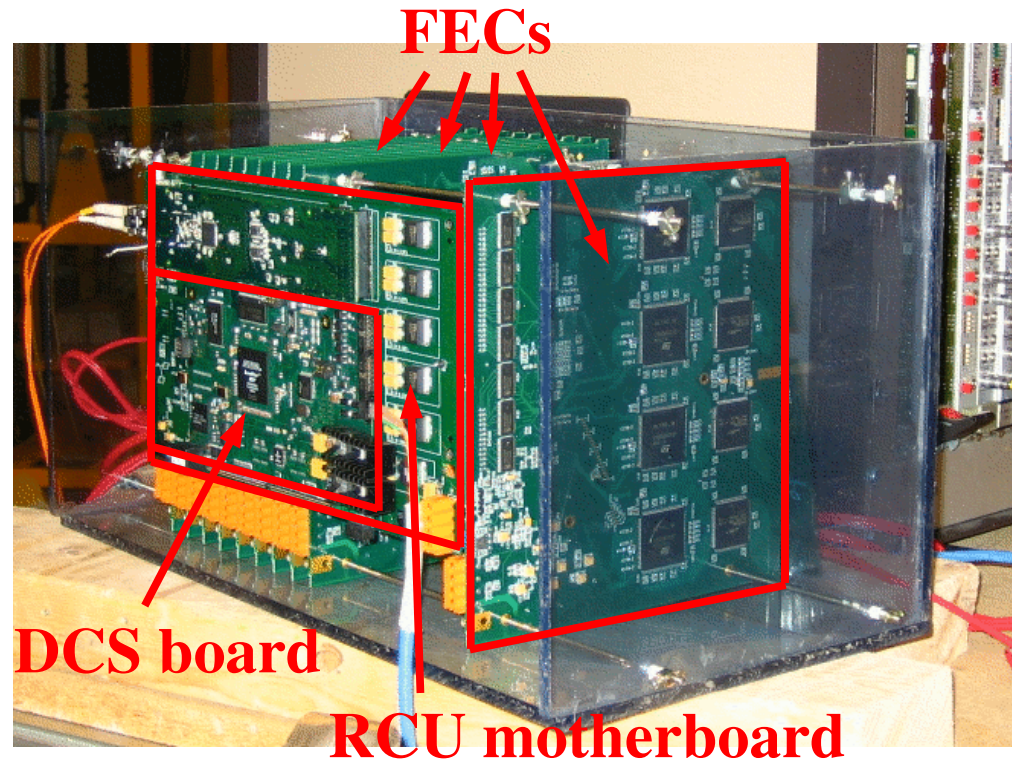
System prototype during irradiation test at TSL/Uppsala

  - satisfying results; but we still need more experience to cope with the challenge of the final system

Summary

- complete configuration and monitoring of the Front-end electronics carried out by the Control System

- Linux operating system on embedded computers combined with PLD is chief cause for modularity and flexibility

- abstraction layers and well-defined interfaces increase structure and testability

- flexible and module-based system with easily reconfigurable Soft- and Firmware

- integration tests with satisfying results

- ongoing work on user-friendly GUI
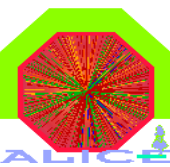
Acknowledgments

J. Alme, S. Bablok, D. Larsen, D. Röhrich, K. Ullaland

Department of Physics and Technology, University of Bergen, Norway

K. Røed

Faculty of Engeneering, Bergen University College, Norway

T. Krawutschke

Institute of Communication Engineering, University of Applied Sciences Cologne, Germany

R. Keidel, Ch. Kofler

Center for Technology Transfer and Telecommunications, University of Applied Science Worms, Germany

U. Frankenfeld

GSI, Gesellschaft für Schwerionenforschung, Darmstadt, Germany

T. Alt, D. Gottschalk, V. Lindenstruth, H. Tilsner

Kirchhoff Institute of Physics, University of Heidelberg, Germany

R. Campagnolo, C. González Gutiérrez, B. Mota, L. Musa

CERN, European Organization for Nuclear Research, Geneva, Switzerland