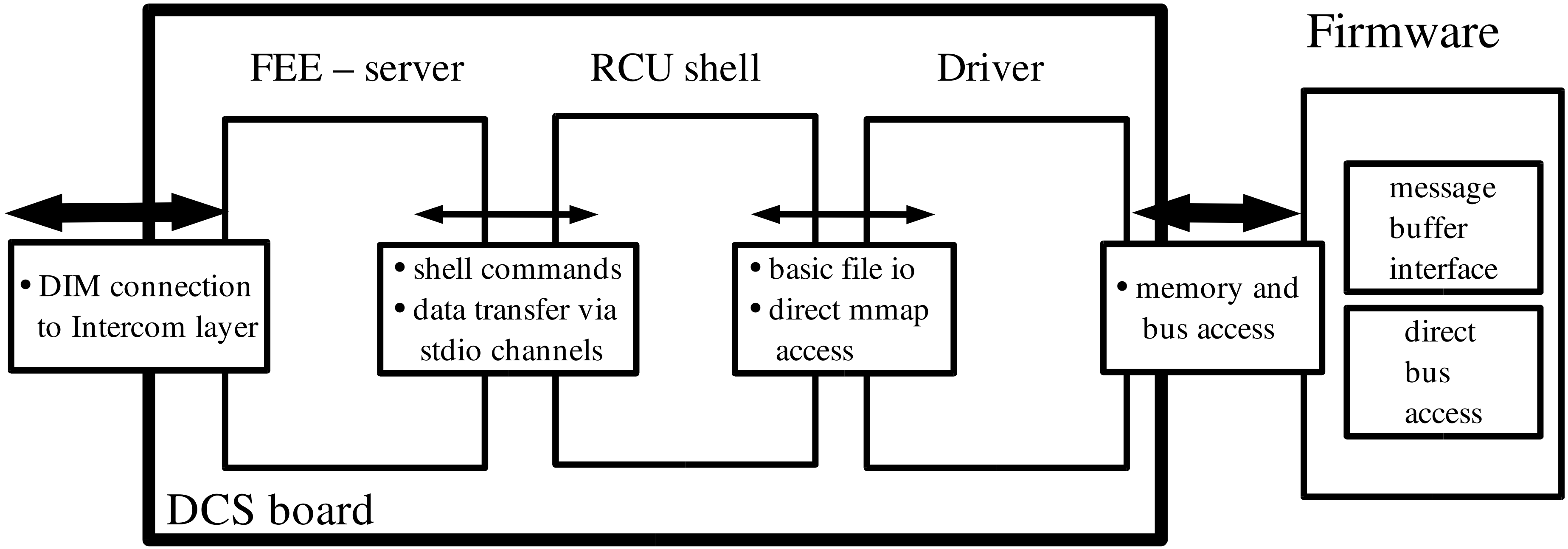
DCS board software status and concept

Matthias Richter

Department of Physics and Technology, University of Bergen, Norway

ALICE TPC FEE workshop, CERN Jan. 13th - 14th 2005

Components



- modularized: 3 components which communicate through well defined interfaces
- from the software's point of view the firmware appears as a memory mapped interface, the driver translates this into a "user friendly address space"
- changing the firmware requires adjustment of driver, all other components unchanged

Driver

- communication interface to RCU memory: DCS message buffer interface
- RCU FPGA/Flash configuration: direct bus access

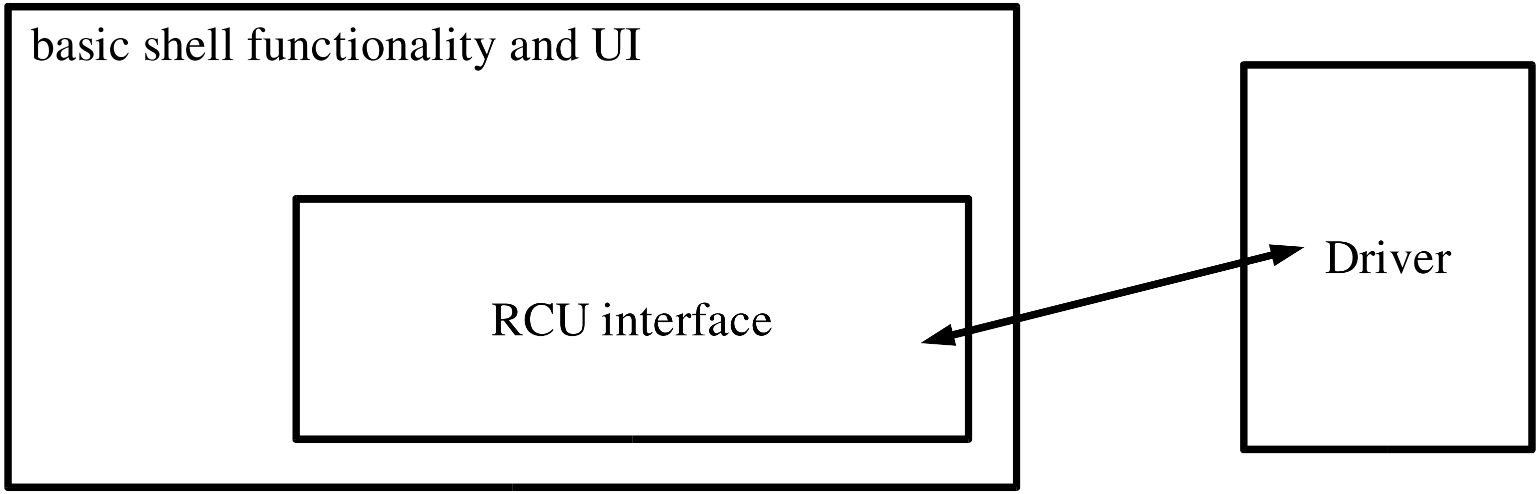
Devices:

/dev/rcu/msgbuf	message buffer interface
/dev/rcu/fpga	fpga configuration
/dev/rcu/flash	general flash memory access
/dev/rcu/flash0	flash memory bank 0
/dev/rcu/flash1	flash memory bank 1
/dev/rcu/flash2	flash memory bank 2
/dev/rcu/flash3	flash memory bank 3

everything covered by one driver/kernel module with driver lock facility
→ handling of concurrenting processes

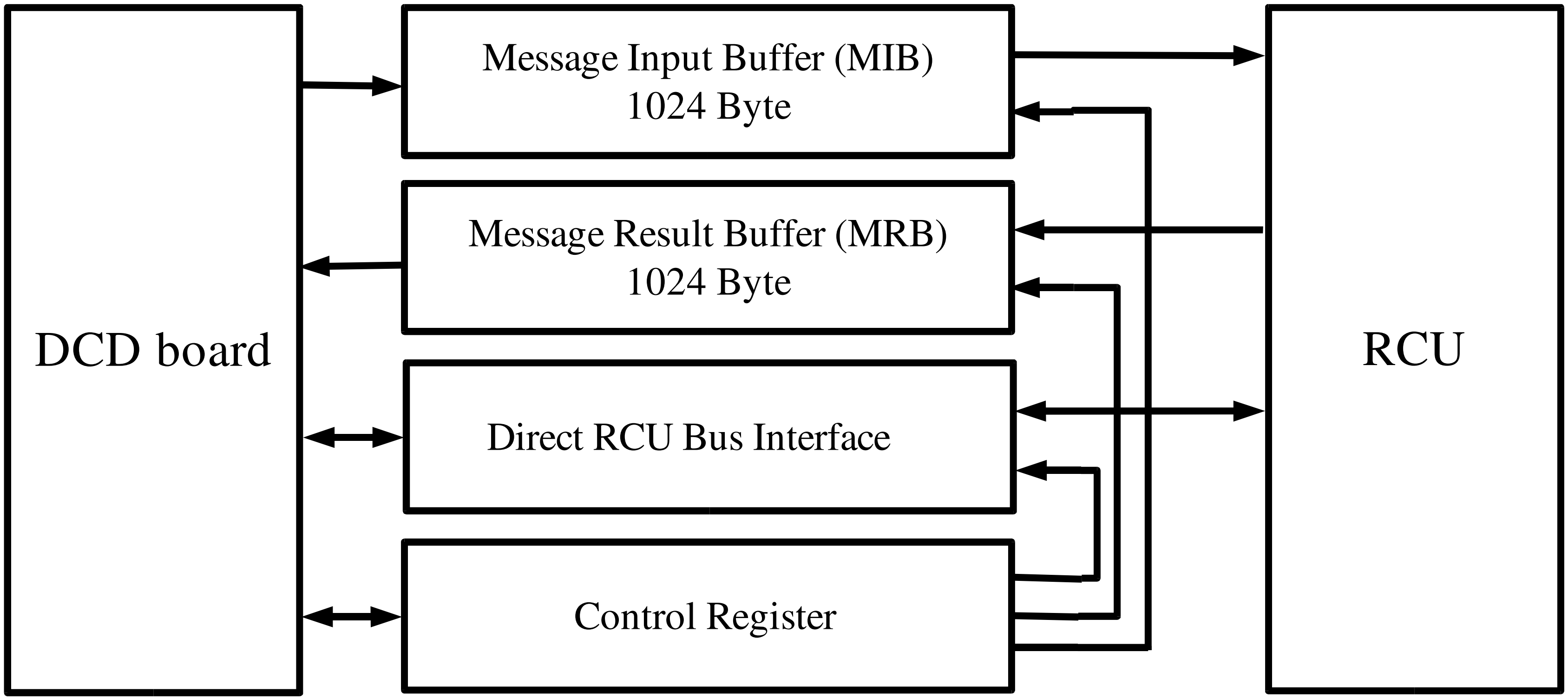
RCU shell

- sendRCUcommand program
- provides basic read/write access to RCU memory
- batch scripts
- debugging tools
- shell scripts via program arguments



Interface to RCU

Two interface types: Message Buffer Interface and Direct Bus Access



Control Register: Data Flow Control

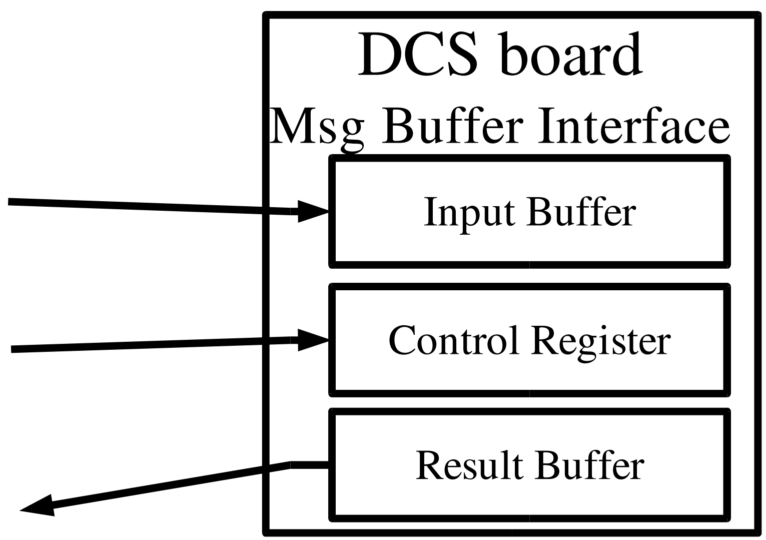
Bit 7	Start Command	Bit 6	MIB multiplexer, read enable
Bit 0	Ready	Bit 5	Direct mode select

Message Buffer Interface

- the interface moves some of the complexity to the firmware
- decouples cpu from the communication task
- block by block transfer
- 2 Buffers for communication and one Contol Register

Basic Sequence:

- write command sequence to MIB
- set 'execute' flag
- wait for the 'ready' flag
- read result and status



Data exchange format

31 … 16	15 … 6	5 … 0
Block number	Number of Words	Command Id
Command word #1		
Command word #2		
Command word #3		
Command word #4		
…		
…		
…		
Command word #n		
Marker	Checksum	
Block number	Number of Words	Command Id
Command word #1		
Command word #2		
…		
…		
…		
Command word #n		
Marker	Checksum	
End marker word		

information word:

- Bit 0-5: Command identifier
- Bit 6-15: Number of 32 bit data words in command block, (excluding information word and marker word)
- Bit 16-31: Block number, flags tbs

marker word:

- Bit 0-15: Checksum
- Bit 16-31:Marker word id AA55

end marker word:

- Bit 0-15: not used
- Bit 16-31:Endmarker word id DD33

Command identifiers:

Identifier	Command
000001	Single read
000010	Single write
000011	Multiple read
000100	Multiple write
000101	Random read
000110	Random write

Instruction format

Instruction format

- single read
  - address word
- single write
  - address word
  - data
- multiple read
  - address word
  - number of data words to read
- multiple write
  - address word
  - number of data words
  - data word 1
  - :
  - data word N
- random read
  - address word 1
  - address word 2
  - :
  - address word N
- random write
  - address word 1
  - data word 1
  - address word 2
  - data word 2
  - :
  - address word N
  - data word N

Examples:

single read from address 0x7000

0x00010041	Information word
0x00007000	address 0x7000
0xAA550000	Block marker without Checksum
0xDD330000	End marker

multiple write
4 words starting at address 0x6800

0x00010184	Information word
0x00006800	address 0x6800
0x00000004	4 words to write
0x0000AFFE	data word 1
0x0000D00F	data word 2
0x00001234	data word 3
0x00005678	data word 4
0xAA550000	Block marker without Checksum
0xDD330000	End marker

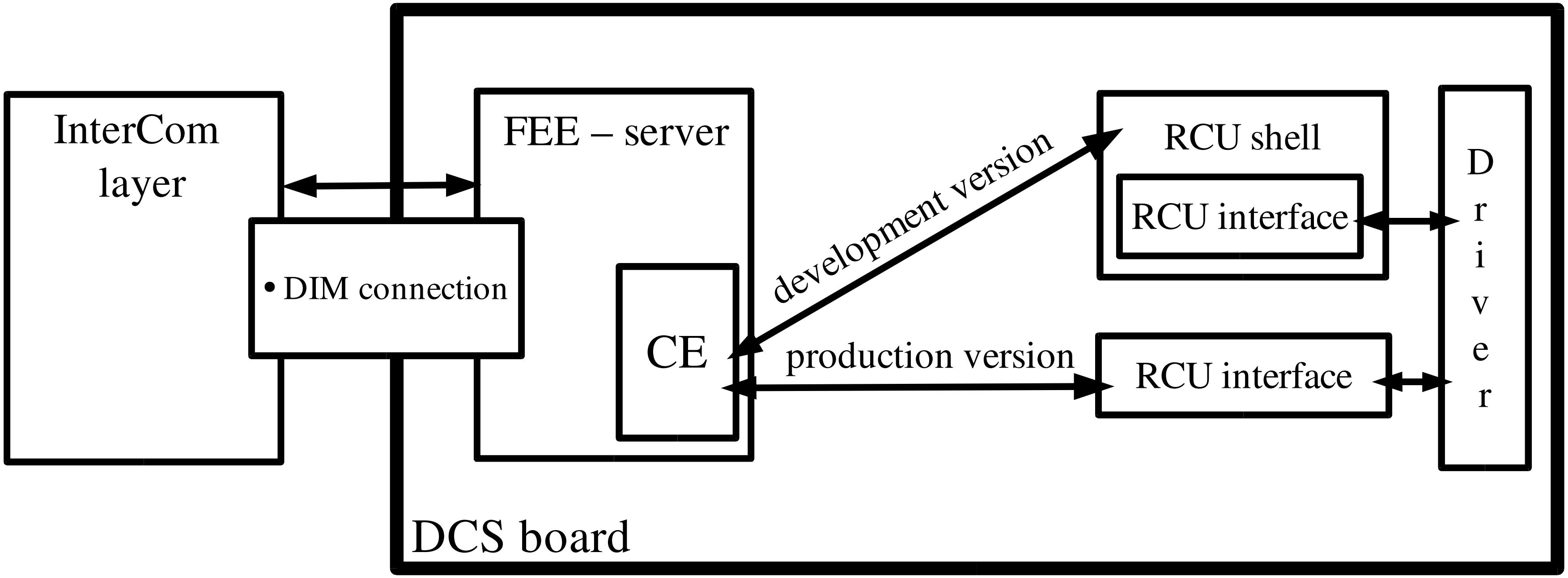
The FrontEnd Electronic server

The FEE server is the gateway to the central Detector Control System via DIM channels

Main tasks:

- Configuration of FEE
- Monitoring
- Controlling and Publishing of states
- Configuration Watchdog/Selftest

Connection to FEE-server



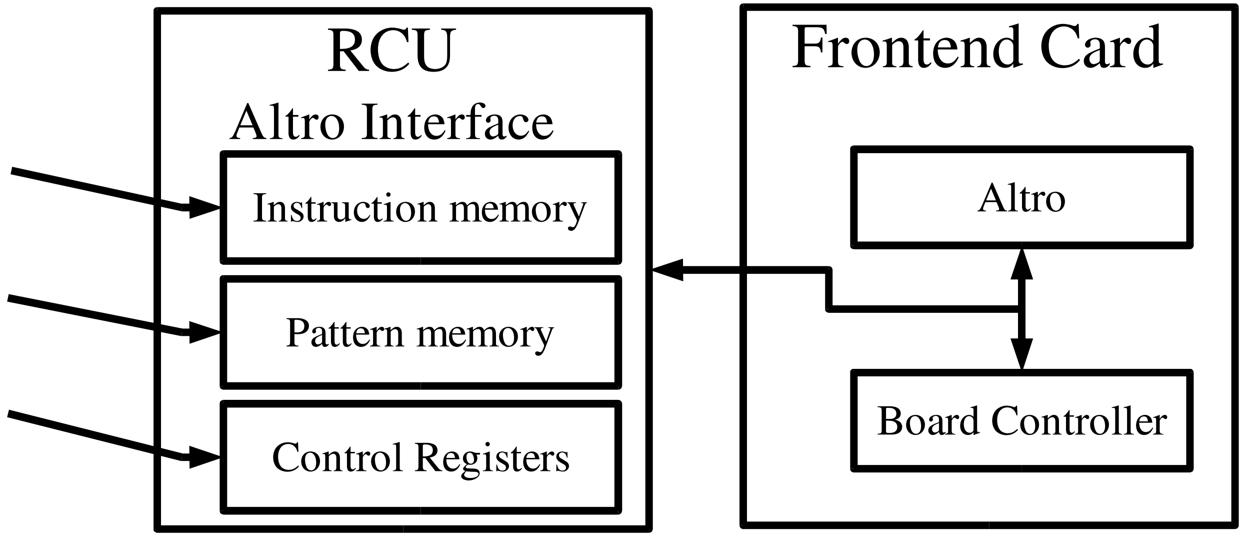
Control Engine (CE)

- development version invokes RCU shell to access RCU memory
- production version implements RCU interface directly
- controls specific deamons and watchdogs
- reads monitoring values from the Slow Control

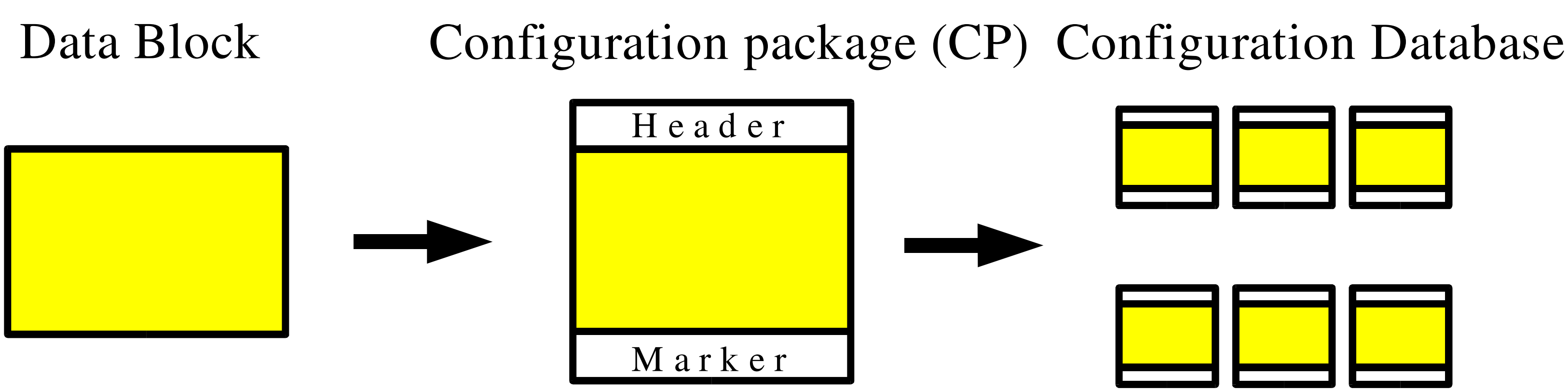
FEE Configuration

Basic Sequence:

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



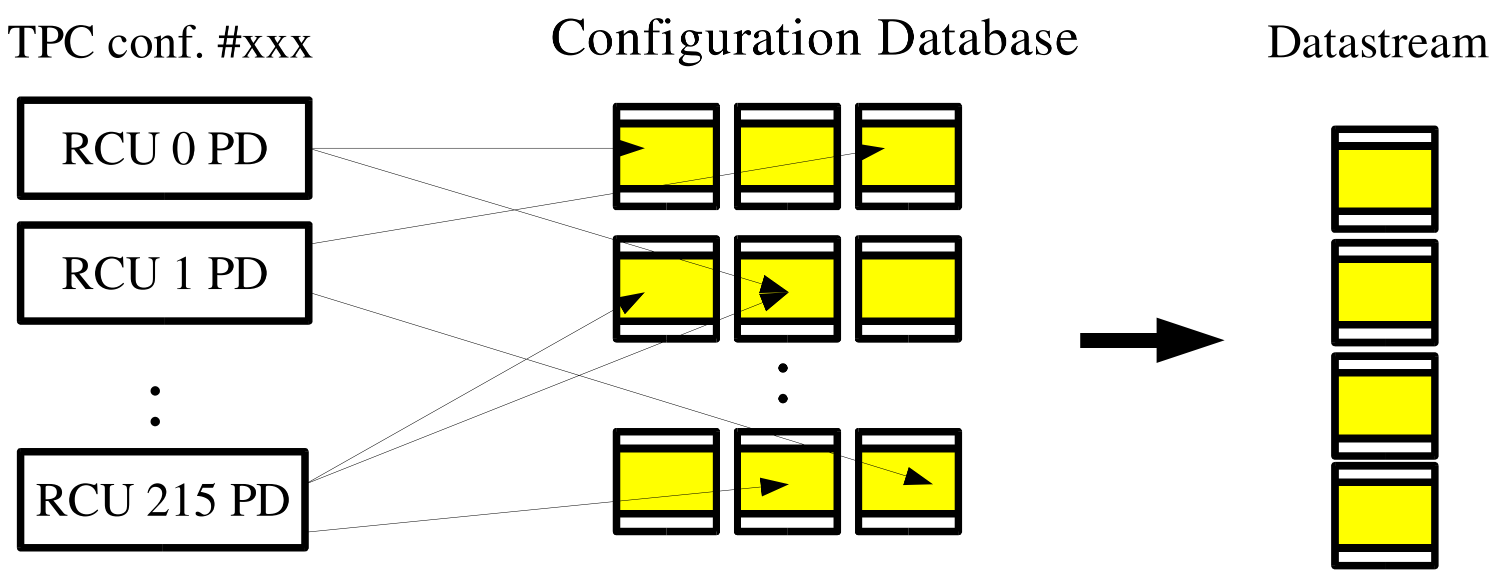
Creation of configuration data:



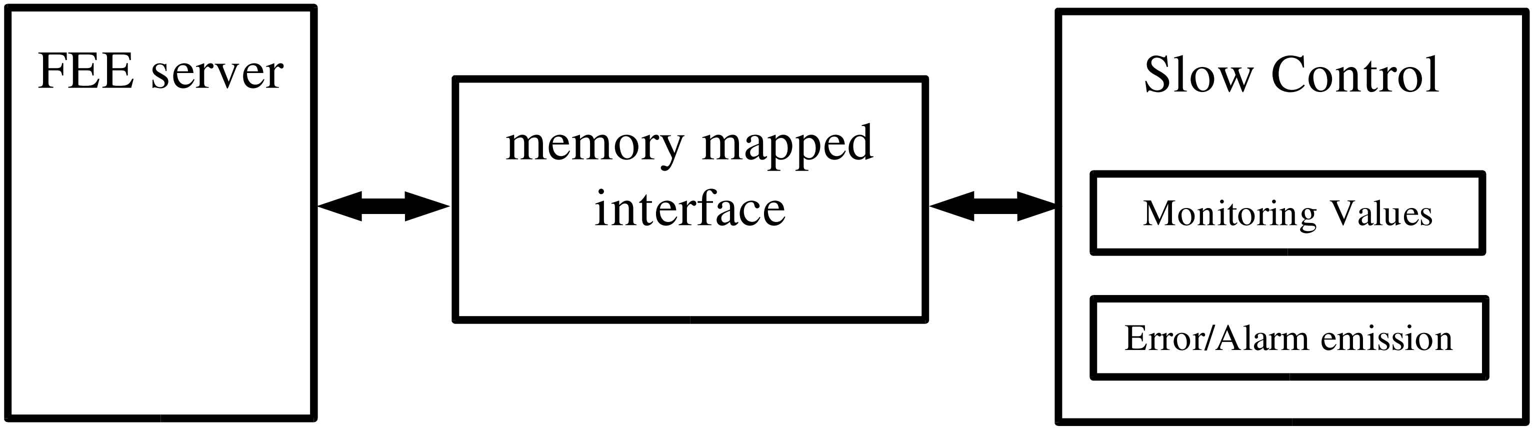
- Configuration package is stored in message buffer data format

Archiving of Configuration Data

- The data is organized in programming blocks, the Configuration Database archives them and builds a index list
- an RCU configuration is defined by the Package Describer (PD), which is a list or sequence of Configuration Packages (CP)
- a whole TPC configuration is a list of PDs for each RCU
- InterCom layer gets a full sequence from the Configuration Database/File structure and sends it to the DCS board



Monitoring and slow control



- access to slow control via memory mapped interface, by writing the right magic numbers to the interface the FEE server can get the desired values
- FEE server polls the monitoring values und publishes them to the DCS
- critical alarms are handled as close as possible to the source, i.e. on the RCU/FEC; a notification or error message is sent to the central DCS
- possible errors/alarms: temperature, high voltage, low voltage, HV current, LV current, missing clock, missing triggers, corrupted FEC configuration

Status and outlook

- all modules full functional and have to be connected
- code optimization: direct memory access
- driver lock has to be fully implemented
- specification for 'Slow control' tasks on the DCS board have to be discussed