

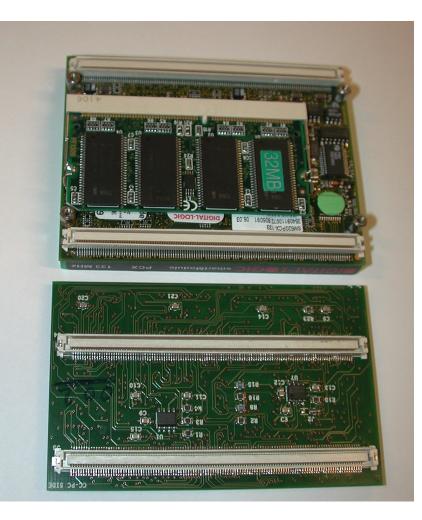
The Credit Card PC:

### The Server:

- New Release
- Outlook

### Tools (PVSS):

- a brief introduction





# Introduction (I):

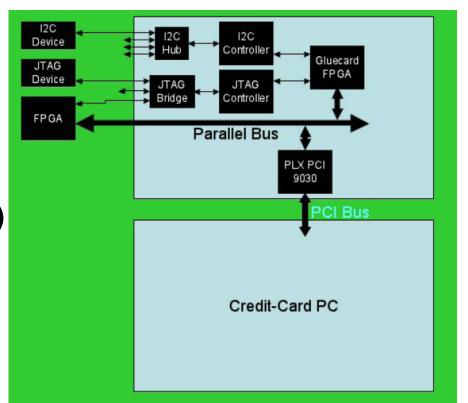
**Plx/PCI** bridge to gluecard

- 3 bus-types:
  - parallel bus (8/16/32 bits)
  - I2C bus (4 distinct lines)
  - JTAG (3 chains)

### CCPC is used on:

- Tell1 board
- UKL1 board (16 bit local bus)
- TFC boards
- OPG and others (L0)

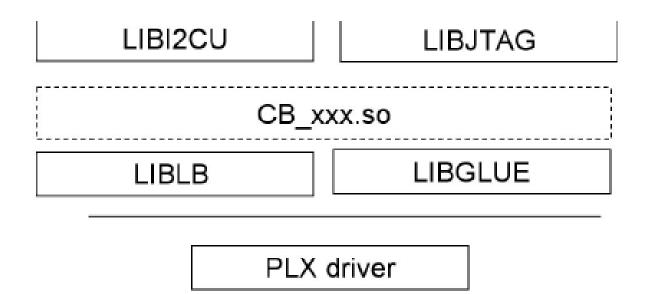
Different Hardware but same server!





Introduction (II):

#### The software layers on which the server is based on:



### These libraries are very reliable and well tested. Currently running under SLC4.



More Information:

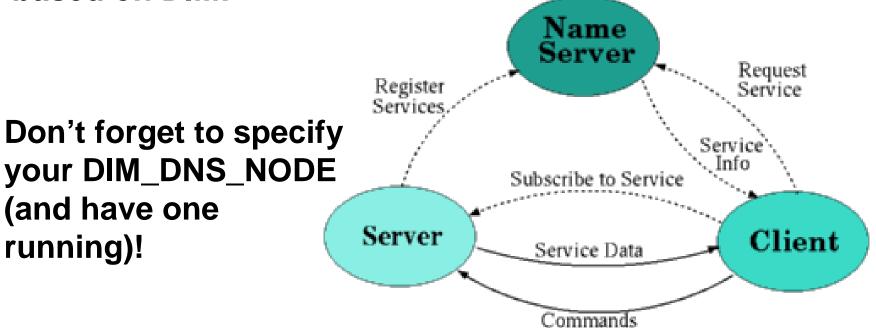
#### Niko Neufeld: The CCPC Software developers guide

#### http://lhcb-daq.web.cern.ch/lhcbdaq/ccpc/development/guide.pdf



Introduction (III):

Communication between server and clients based on DIM:



The PVSS clients are presented later in more detail ...



More Information:

#### Clara Gaspar: Distributed Information Management System

http://dim.web.cern.ch/dim



Server (I):

- In order to have similar framework functions on the PVSS client-side for both SPECS & CCPC the interface has changed.

- The way 'dpSet' is implemented in PVSS seems to affect the performance:

The new interface tries to optimize this.

- Traffic is kept as low as possible: All register information is stored in lists on the serverside. On the other hand, registers are just updated on the client-side if they have really changed.



<u>Server (II):</u>

- A simple interface shall be provided on the userside:

try as much as possible to hide complexity of the different hardware types.

(for the benefit of the user and the pain of the developer  ${\sf J}$  )

- Future release will add functionalities whithout breaking existing PVSS interfaces. (user code is safe)

- Existing SPECS panels adapted for the CCPC. (Ricardo)



Server (III):

Future tasks:

- Model the whole Tell1 board (using the new tools) as PVSS datapoint types.

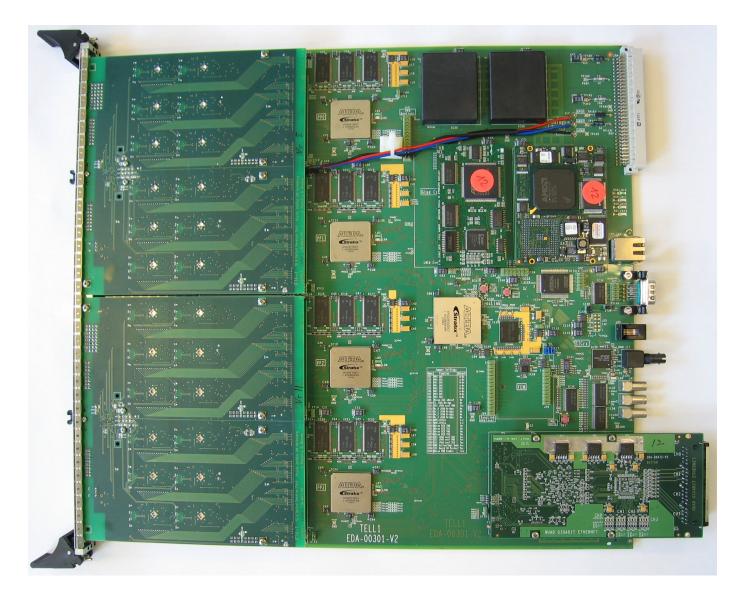
- Defining a higher level framework for the Tell1 to be used by finite state machines or modelling macroscopic functions (e.g. exploiting the powerful PHY chip on the GBE-card, etc).

- Monitoring of registers bundled to further improve performance.

- Connection to configuration database. Up- and Download of default values. (defining PVSS recipies)



<u>Use-case Tell1 board (I):</u>





# Use-case Tell1 board (II):

Creating registertypes with FwHw-tool (e.g. I2C):

Similar to the specssystem. (as in tutorial yesterday)

Parameters to be defined are however different! I2C,LBUS,GBE provided so far.

	Hardwares Config	
	Hardwares (type: test)	Common settings (hardware: mytest)
	mytest	pc pctell07
n_1: Hw Tool el ? dware types-	mytest Create Remove Clone	
est	Subscribe Unsubscribe	Cancel
	Registers (hardware: mytest)	Specific settings (register: myregister)
	myregister (type: I2C)	bus 0
		I <sup>2</sup> C address 0x50
		I <sup>2</sup> C sub address 0x0
		Size (bytes)
		type 0
		Refresh rate (seconds) 1
		Data change 0
Create	Show registers of type (All)	Save Cancel
Configure		Close
		Close

🙀 Visia

<u>File P</u>ar

Hai



# Use-case Tell1 board (III):

R/W or Monitoring (e.g. I2C): again similar to the tutorial yesterday!

Or not so similar: TTCrx has nonstandard I2C interface! The ID-EEPROM allows just a 16 Byte write access (pagesize)! Beetle offers 8bit shift registers ....

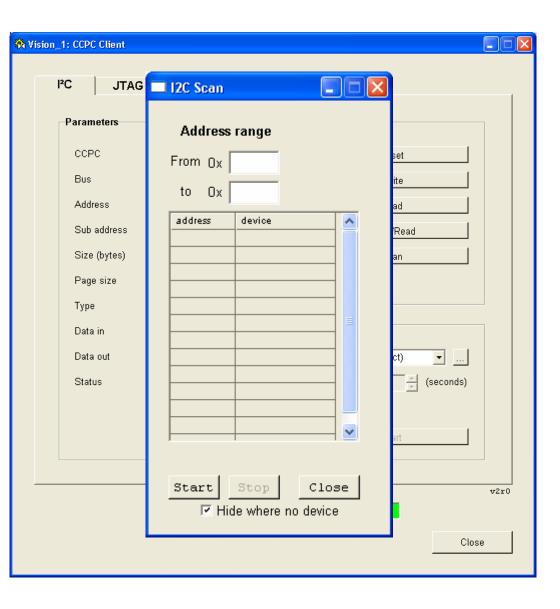
			ADVANCED
Parameters			Commands
CCPC	pctell07	•	Reset
Bus	0	•	Write
Address	0x50		Read
Sub address	0x0		Write/Read
Size (bytes)	1		Scan
Page size			
Туре	Separated (TTCrx)	•	
Data in	(Select) Separated (TTCrx) Combined (Others)		Monitoring
Data out	Combined (Others) Shift register (Beetle)		Registers (Select) 💌
Status			Refresh 🛛 📩 (seconds)
			🗖 Data change
			Start
		ıt is running	v2r

Stefan Koestner



### Use-case Tell1 board (IV):

I2C-scan: scan the addresses of existing (sending acknowledge) devices.





## Use-case Tell1 board (VI):

Monitoring: select either on change and/or a predefined polling rate.

-Re

	🖗 Vision_1: CCPC Client	
	I <sup>2</sup> C JTAG	The 'test' register has the following parameters:
	Parameters	CCPC pctell07
	CCPC	Bus
ange	Bus	Address 0x50
d	Address	Sub address 0x0
u	Sub address	Size (bytes)
isters Config		Page size
		Type Separated (TTCn)
egisters		Close
test		Combined (Others) Shift register (Beetle) Registers (Select)
		Refresh (seconds)
		🗖 Data change
		Start
test		v2r0
Create	love Close	Close



## Use-case Tell1 board (VII):

LBus/GBE: on the user-side they behave similar.

Internally however GBE is a 16 bit bus, where the low word is stored in a different memory block than the high word. (which is already treated at low level)

PC	JTAG	LBUS	GBE	ADVANCED
Parame	ters			Commands
CCPC	Γ	pctell07	•	Reset
Addres	s [	Dx4200000		Write
Туре	Γ	32 bits	•	Read
Size (v	vords)	1		Write/Read
Data in		feedbabe		Memory
Data o	ut [	FEEDBABE		
Status	Γ	OK (Write/Read)		
				Monitoring
				Registers (Select) 💌
				Refresh 🛛 📩 (seconds)
				🗖 Data change
				Start
				v
		[21:32:03] CCPC Cli	ent is runnin	ig.



# Use-case Tell1 board (VIII):

JTAG: Basic JTAG Operations are supported.

Integration into the generic FwHw-tool (for defining registers) possible. Not really required for the Tell1 board.

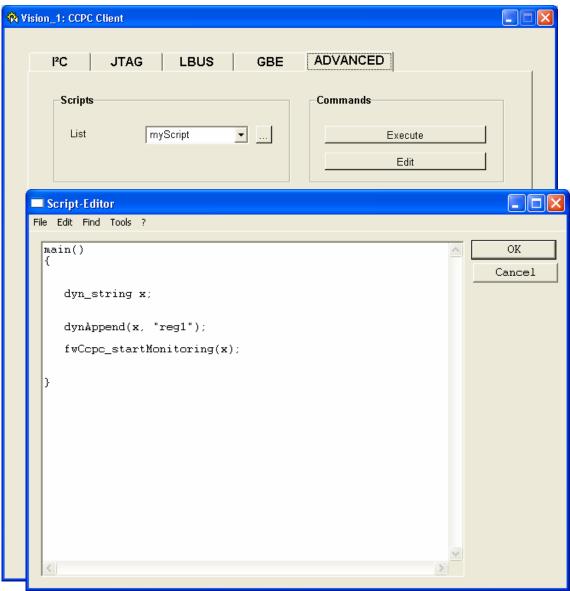
I²C	JTAG	LBUS	GBE	ADVANCED	
Parame	eters			Commands	
ССРС	:	pctell07	•	Reset	
Chain		0	•	JTAG Reset	
Size (	(bits)	16		JTAG Idle	
Data i	n	cafe		TAP Reset	
Data	out			IR Scan	
Statu	•			DR Scan	
				Scan	
					v2:
		[21:32:03] CCPC (	Client is running	g.	



## Use-case Tell1 board (IX):

scripting: little scripts can be run using the framework functions.

commandline tools: can be launched from PVSS as well. (soon)





# Use-case Tell1 board (X):

download of FPGA firmware:

supported files are .pof and .jbc (not .jam).

With jumper setting FPGAs can be programmed directly without EPC16!

ion_1: CCPC Cli	ent 💶 🗖
I²C	JTAG LBUS GBE ADVANCED
-Scripts List	Commands       myScript        Edit
-FPGA Pro	gramming Commands
CCPC Chain File	pctell07     Send       0     Image: C:/TELL1_1s25_orx_1
System	Open  C:\CcpcClient\
	Look jn: System (C:) OEMDRVS OEMDRVS Program Files pvss TELL1_1s25_orx_BER_v61.pof SpecsServer tmp
	File name:     TELL1_1s25_orx_BER_v61.pof
	Files of type: *pof; *jam  Cancel



Summary:

We have a running system which we want to distribute after some additional testing.(~2 weeks) Comprises:

- I2C,LBus,GBE for register creation tool (FwHw)
- I2C,LBus,GBE,JTAG on test panels
- firmware download from PVSS
- scripting

Users should be able to use it while we take care of some (time consuming) details and improve it further (transparent). If you wish some special services on the server (which I have not thought of) please tell me!

### **Little Demo:** Or how to solve communication-problems with DIM

