9th AGATA week

Legnaro January 20th – 22nd 2010

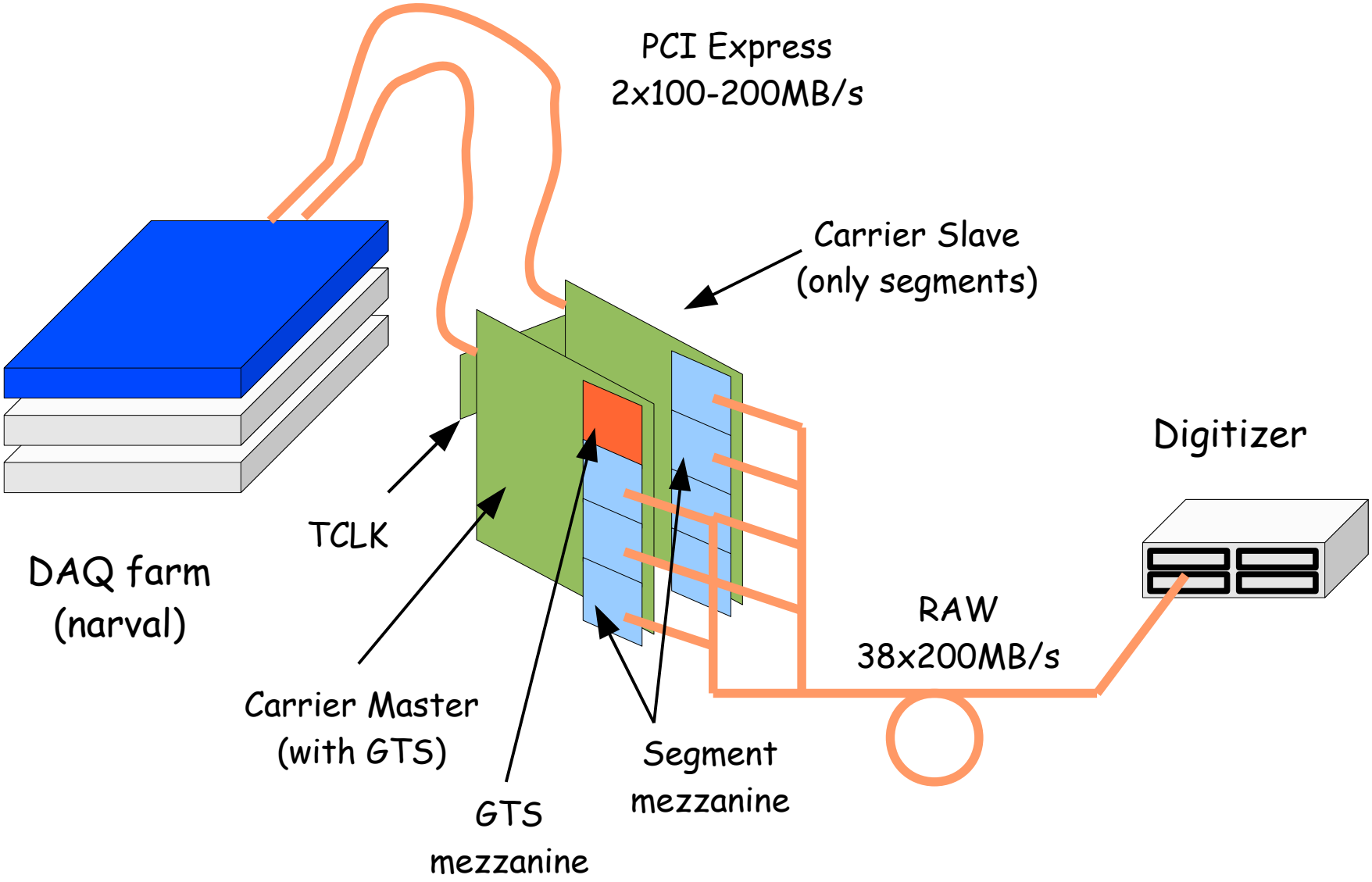
CONTROL AND READ OUT OF ATCA ELECTRONICS

Damiano Bortolato - INFN - Padova

Outline

- System overview
- Slow control status
- Software & firmware improvements
- To do list

General overview
(one crystal)



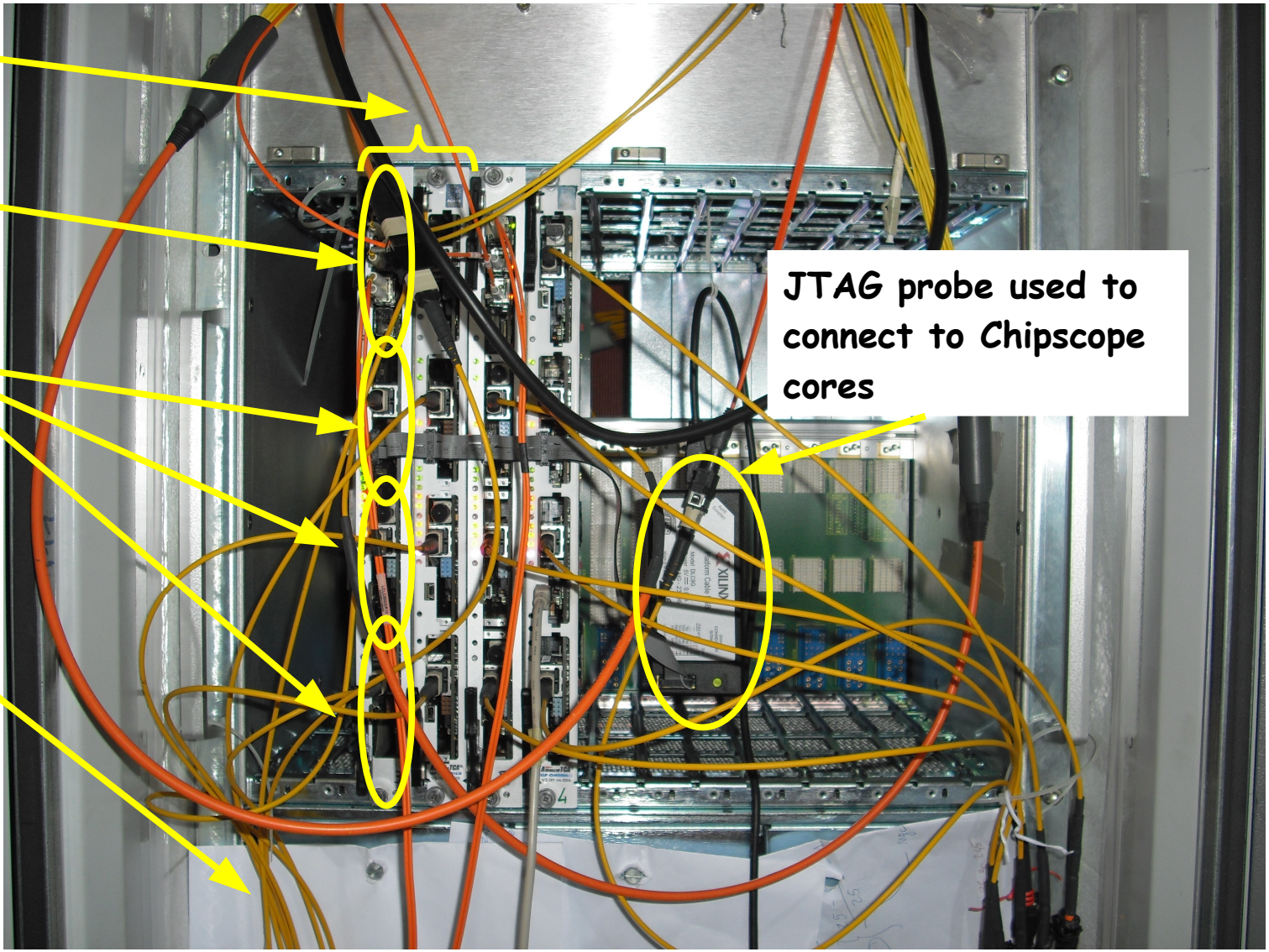
Carrier Overview
crates ATCA

One crystal carriers M-S

GTS mezzanine

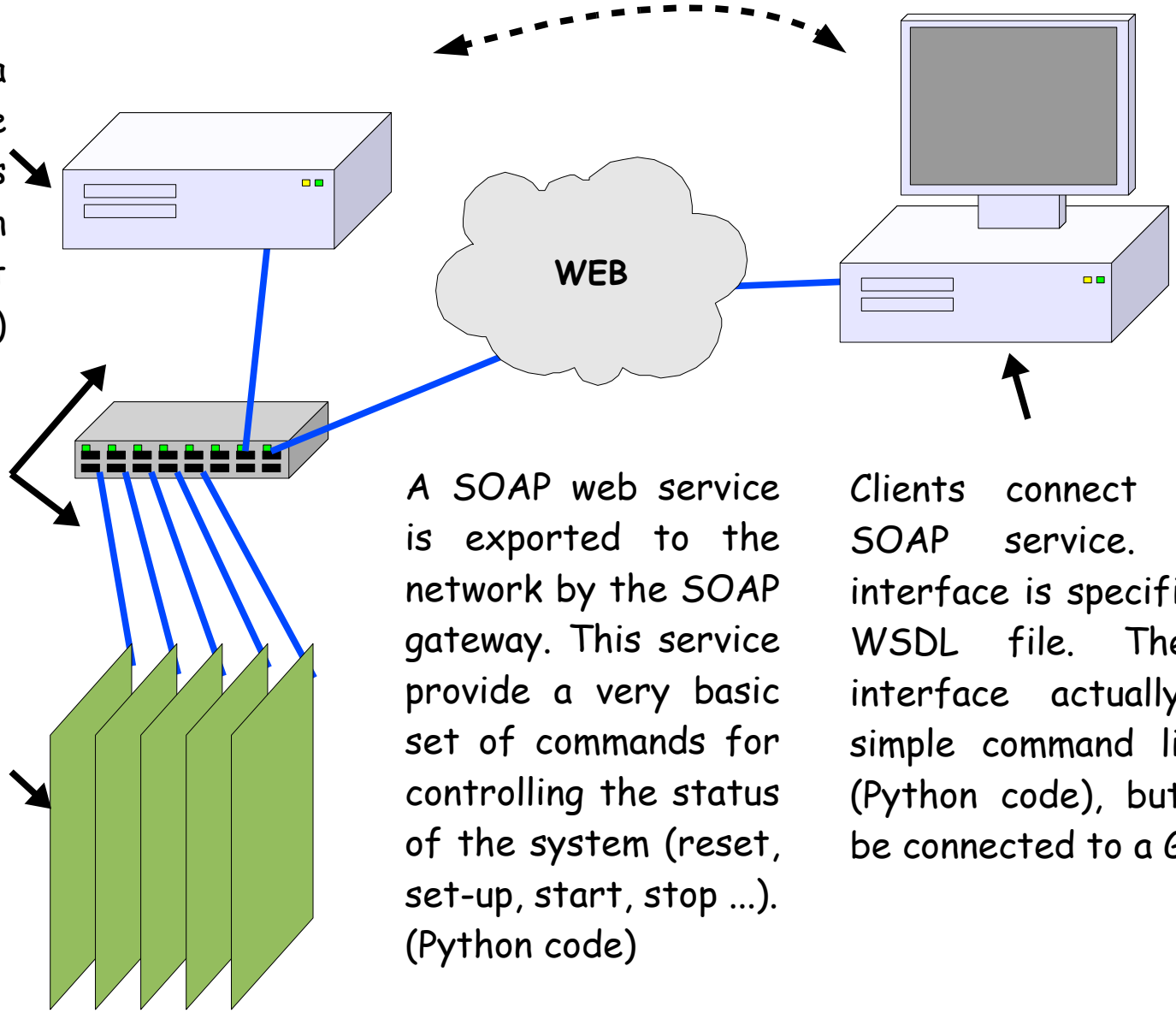
Segment mezzanines

Digitizer connections



Carrier run/slow control

The control host is a SOAP gateway for the carrier system, It runs the controlling daemon and listen for client connection. (Python code)



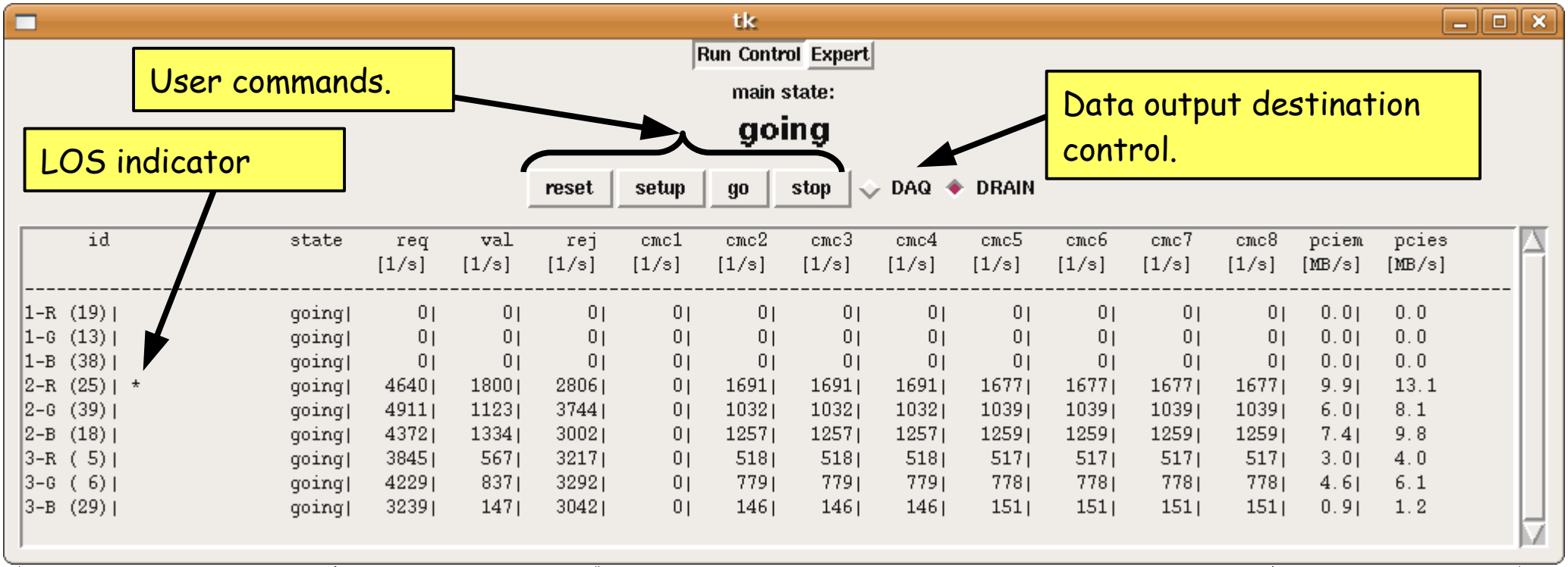
Carrier cards are connected through an Ethernet connection.

Carrier cards are controlled via a telnet session. The control daemon sends commands to each card and parse their output.
(C code)

A SOAP web service is exported to the network by the SOAP gateway. This service provide a very basic set of commands for controlling the status of the system (reset, set-up, start, stop ...). (Python code)

Clients connect to the SOAP service. The interface is specified by a WSDL file. The user interface actually is a simple command line tool (Python code), but it will be connected to a GUI.

Slow Control
current GUI



User commands.

LOS indicator

Data output destination control.

id		state	req [1/s]	val [1/s]	rej [1/s]	cmc1 [1/s]	cmc2 [1/s]	cmc3 [1/s]	cmc4 [1/s]	cmc5 [1/s]	cmc6 [1/s]	cmc7 [1/s]	cmc8 [1/s]	pciem [MB/s]	pcies [MB/s]
1-R (19)		going	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
1-G (13)		going	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
1-B (38)		going	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
2-R (25)	*	going	4640	1800	2806	0	1691	1691	1691	1677	1677	1677	1677	9.9	13.1
2-G (39)		going	4911	1123	3744	0	1032	1032	1032	1039	1039	1039	1039	6.0	8.1
2-B (18)		going	4372	1334	3002	0	1257	1257	1257	1259	1259	1259	1259	7.4	9.8
3-R (5)		going	3845	567	3217	0	518	518	518	517	517	517	517	3.0	4.0
3-G (6)		going	4229	837	3292	0	779	779	779	778	778	778	778	4.6	6.1
3-B (29)		going	3239	147	3042	0	146	146	146	151	151	151	151	0.9	1.2

Crystals list and status.

Trigger rates. (seen in the carrier)

Per mezzanine event rates. (seen in the carrier)

Readout data rate

Set Up Procedure

previous

- Switch on everything (digitizers ATCA crates GTS crate...)
- Start GTS calibration procedure and wait for it to finish the first step.
- Launch carriers management daemon and start carrier set-up.
- Wait for carrier and GTS procedures to finish.
- Send set-up command to digitizers through their own control interface, and wait for it to finish.
- Connect the Chipscope and check manually the status of the channels of each mezzanine.
- Power on or reboot DAQ machines.
- Start narval.

current

- Switch on everything (digitizers ATCA crates GTS crate...)
- Launch "start.sh" & give "reset" "setup" command to carrier GUI.
- Send setup command to digitizers through their own control interface, and wait for it to finish.

  (no more needed)
- Reload kern. modules on DAQ nodes.
- Start narval.

Firmware Improvements

- Full PCI Express License available (periodic reboots are no more needed)
- Rate counters for each segment mezzanine on carrier FPGA0
- Long trace capture on mezzanines FPGA
- JTAG chain access through PCI Express device.
- Trigger activity signalling on carrier front panel leds.

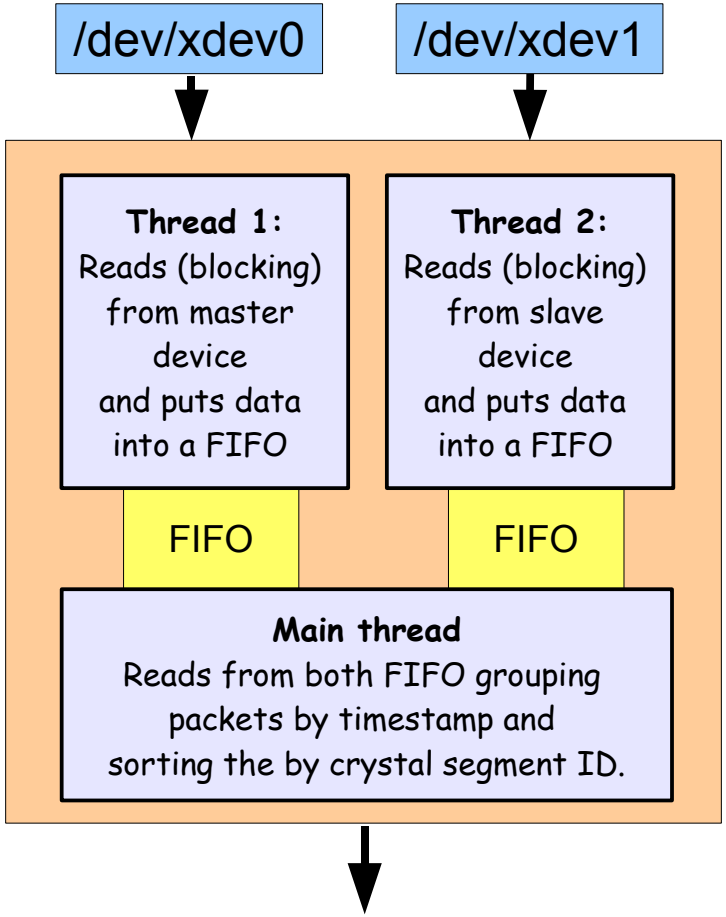
Read Out
kernel module improvements

- Switched from poll DMA mode to interrupt driven DMA.
- Added non-blocking read support
- Added poll() and select() support
- Added fake parallel port (parport) device to access the JTAG chain on HW.

Improved throughput performance

Read Out
user space: narval actor

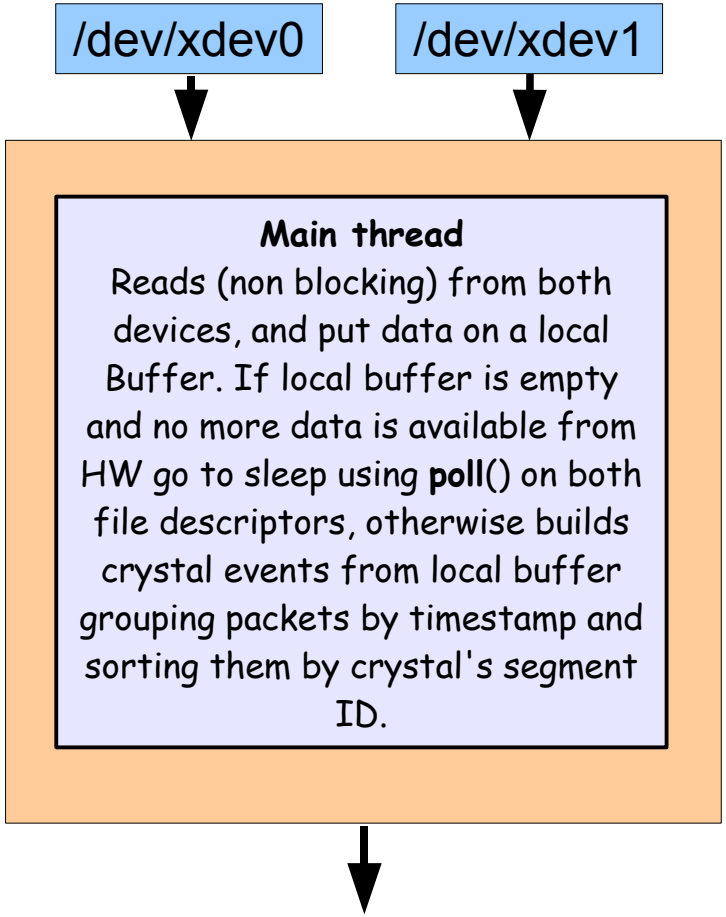
Current implementation



Crystal events to Narval

Rate Max: 120 MB/s ~= 9KHz

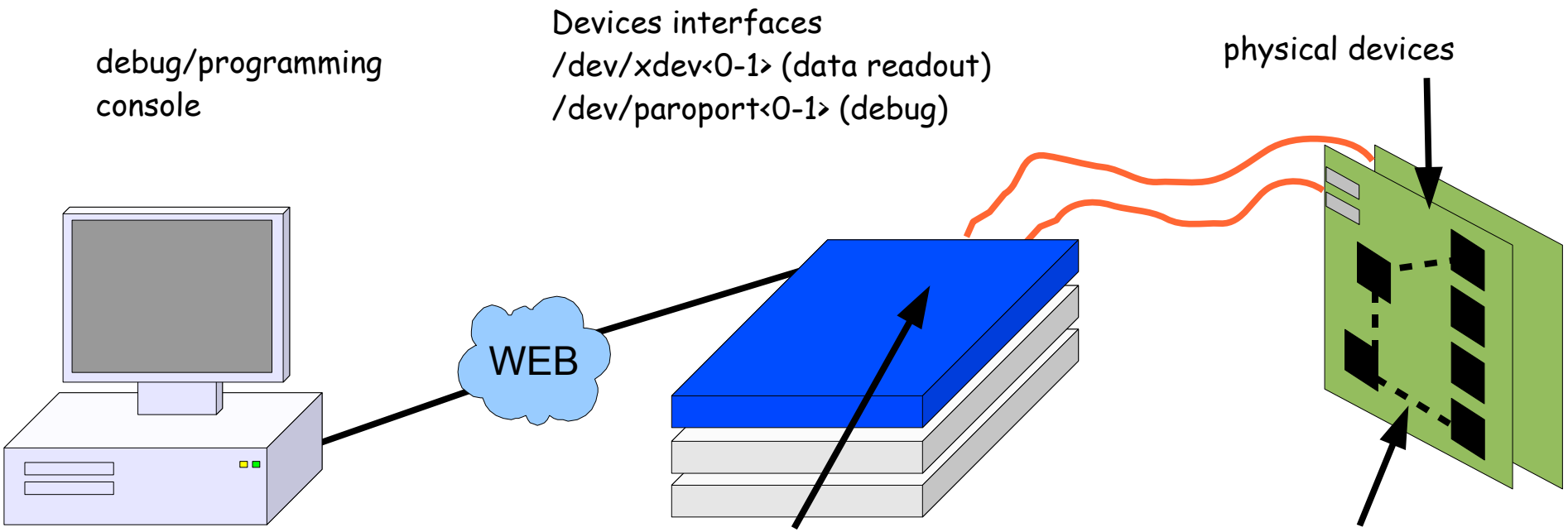
Test implementation



Crystal events to Narval

Rate Max: 315 MB/s ~= 23KHz

Diagnostics & Debug
JTAG (IEEE 1149.1) scalability



The debug console can now access to the JTAG chain to update the firmware or to debug a specific carrier though Xilinx Chipscope analyzer.
Eprom burning is slower if compared with last Xilinx Platform USB probe, but it can be done in parallel for all the carriers and without attaching any physical probe to the HW.

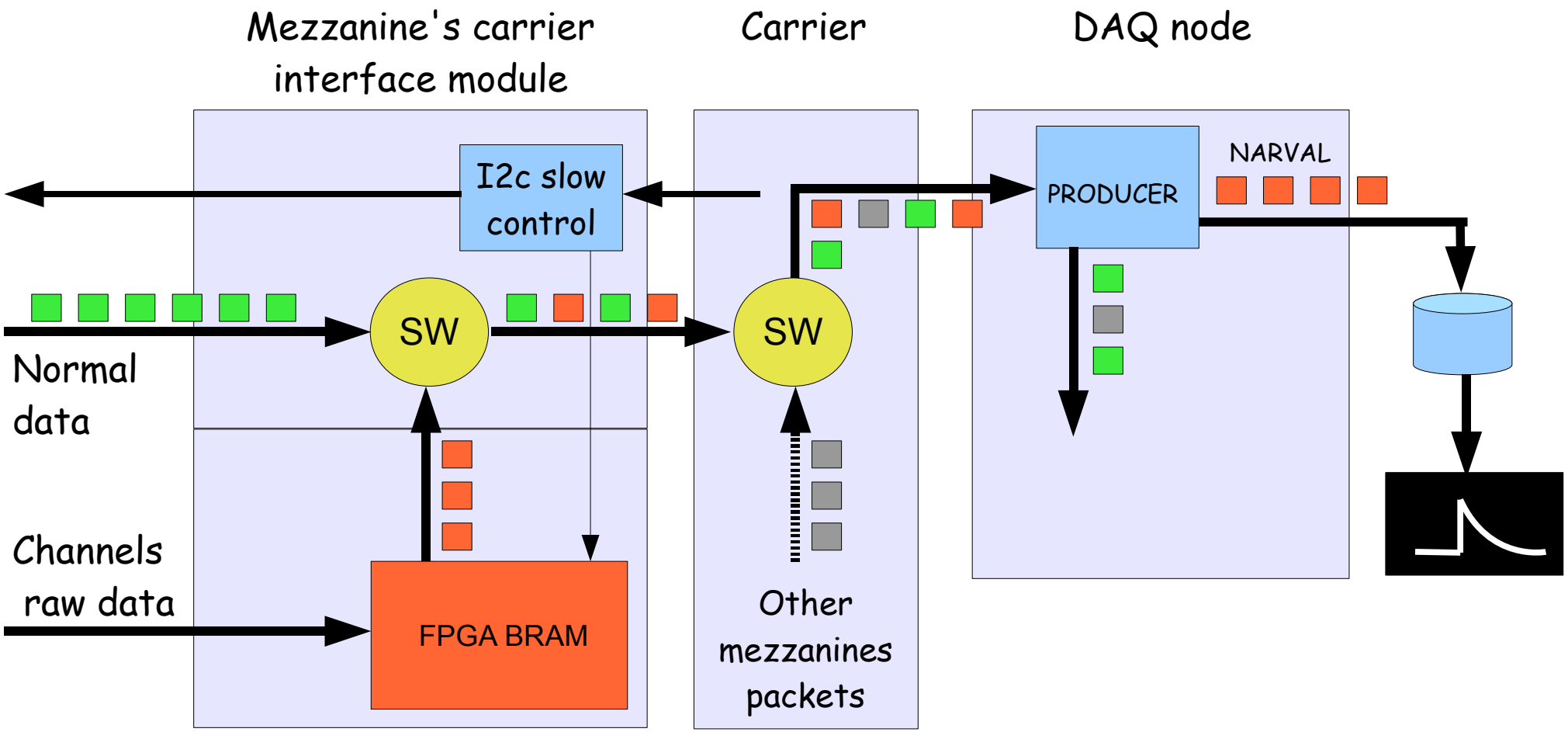
Our Linux kernel module (xdev.ko) creates a fake parallel port device (/dev/parport)
The fake parallel port registers are on the same PCI Express core used for reading out the data.
From the software point of view, xdev and parport devices are completely independent each other, both devices can be used at the same time, without troubles.

JTAG chain is accessed through the carrier's FPGA0 firmware. A piece of VHDL emulates a Xilinx Parallel Cable III JTAG probe.

To Do

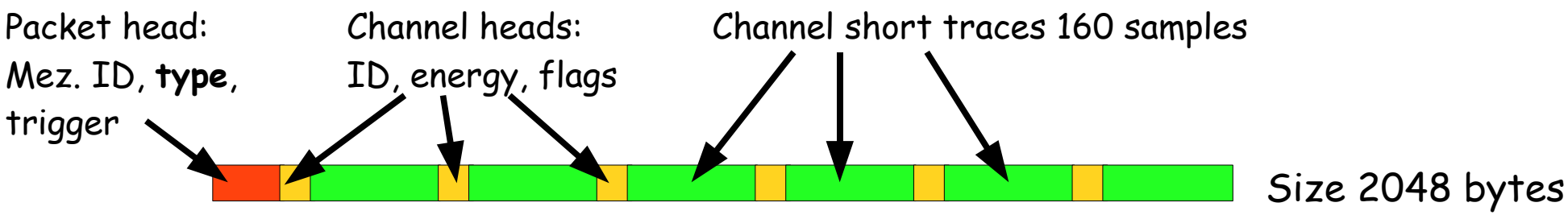
- Completing the new GUI development
- Integration with Global Slow control system
- Preparing and setting up the HW for the next clusters (still under production)
- Repairing malfunctioning HW

Long traces
global view

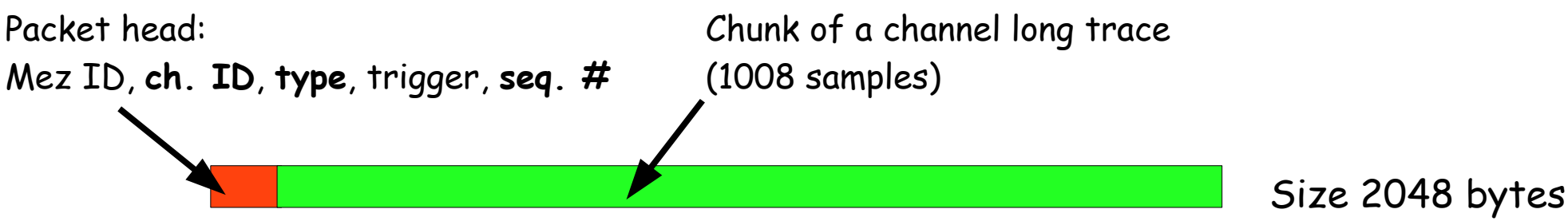


Long traces
data formats

Acquisition data format: each mezzanine sends to carrier 1 packet per event each of them containing 6 short traces



Long trace data format: each mezzanine sends to carrier several packets each of them containing a chunk of long trace



Because the carrier just transports packets of data of fixed size without looking into them, the two data format can be transparently transported to the readout system.