



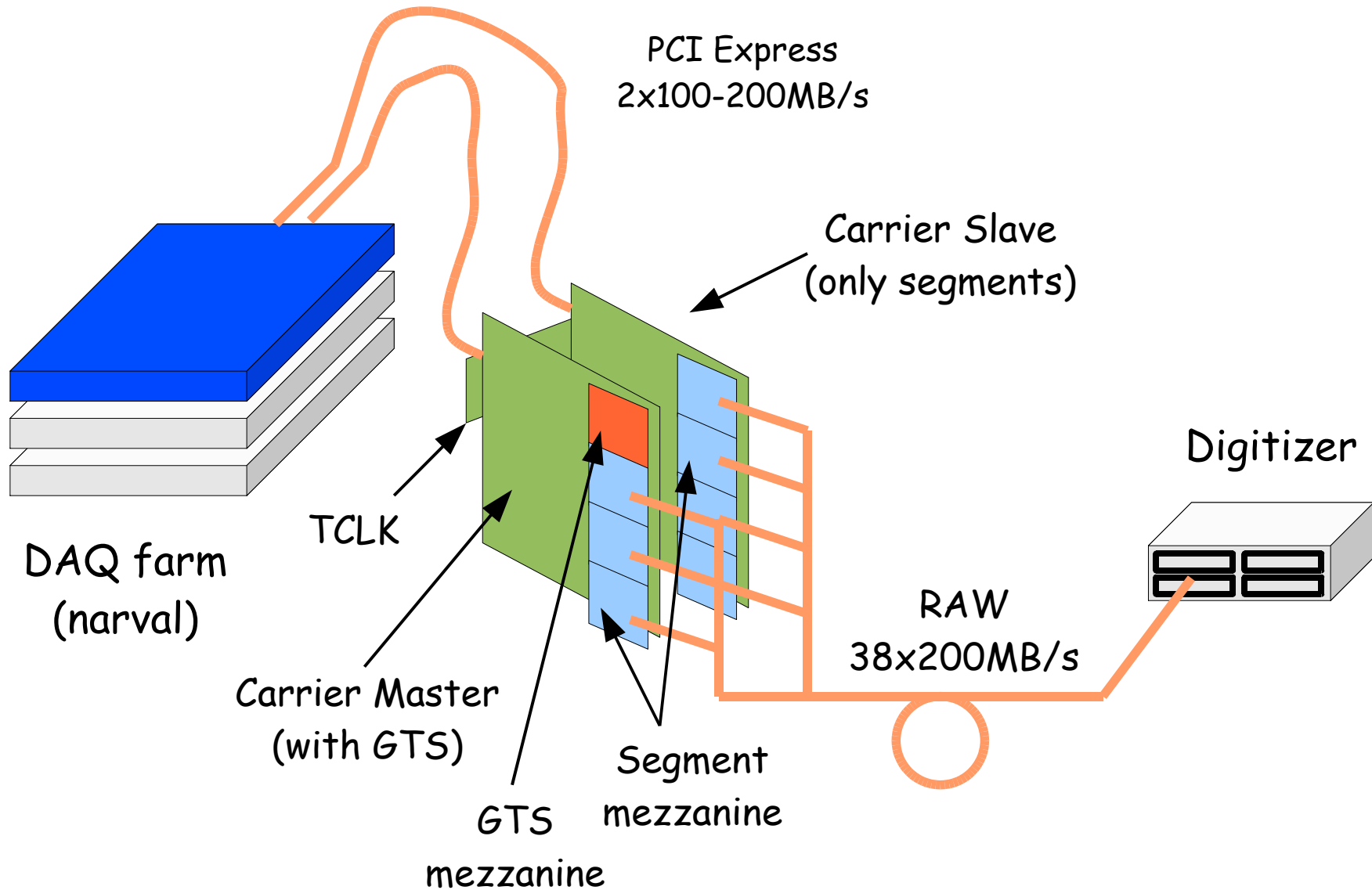
8th AGATA week
Köln, March 31th – April 1st 2009



CONTROLLING THE ATCA CARRIERS AND THE MEZZANINES

Damiano Bortolato
INFN Padova

General overview (one crystal)



Carrier Overview

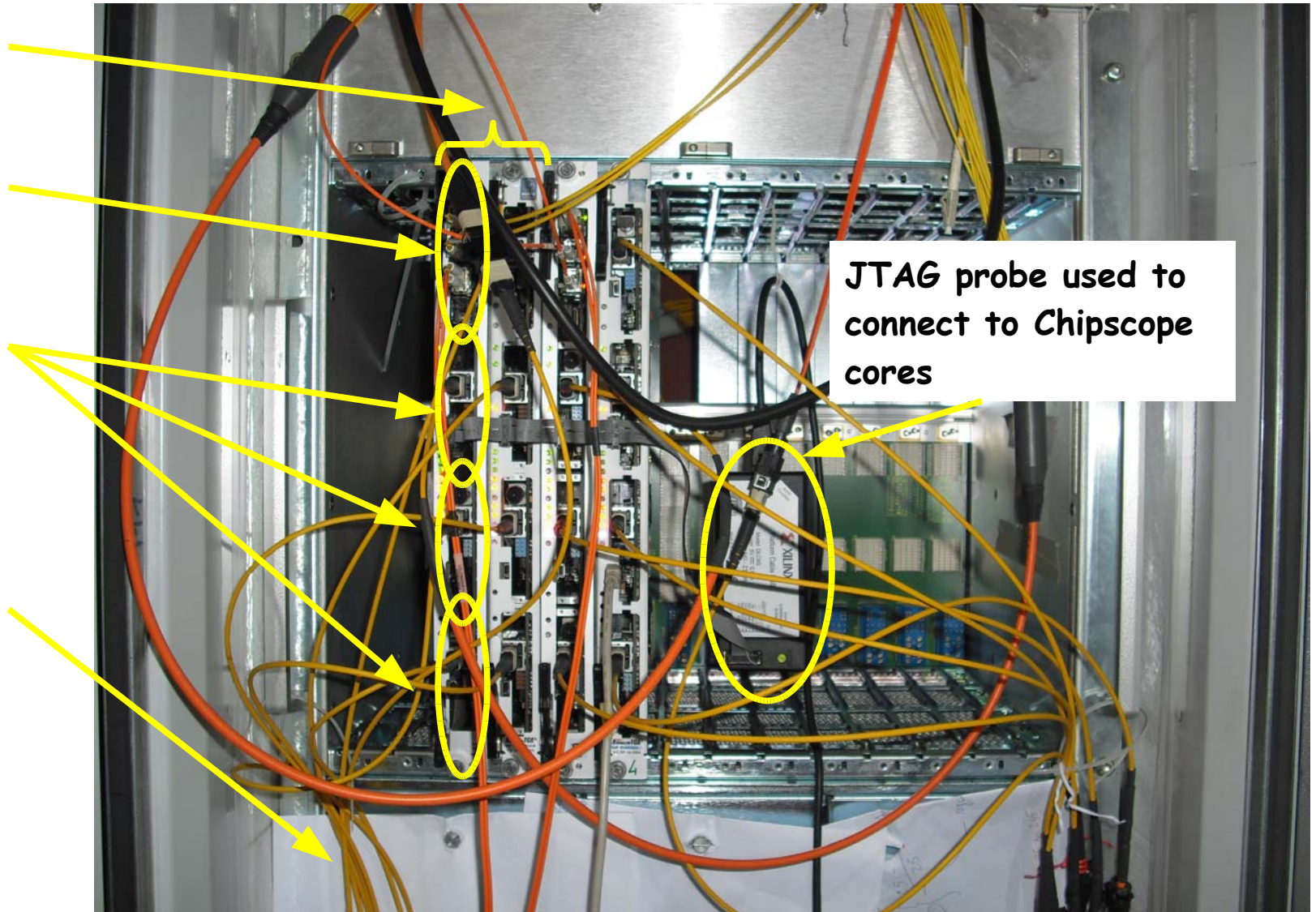
ATCA crates

**One crystal
carriers M-S**

**GTS
mezzanine**

**Segment
mezzanines**

**Digitizer
connections**



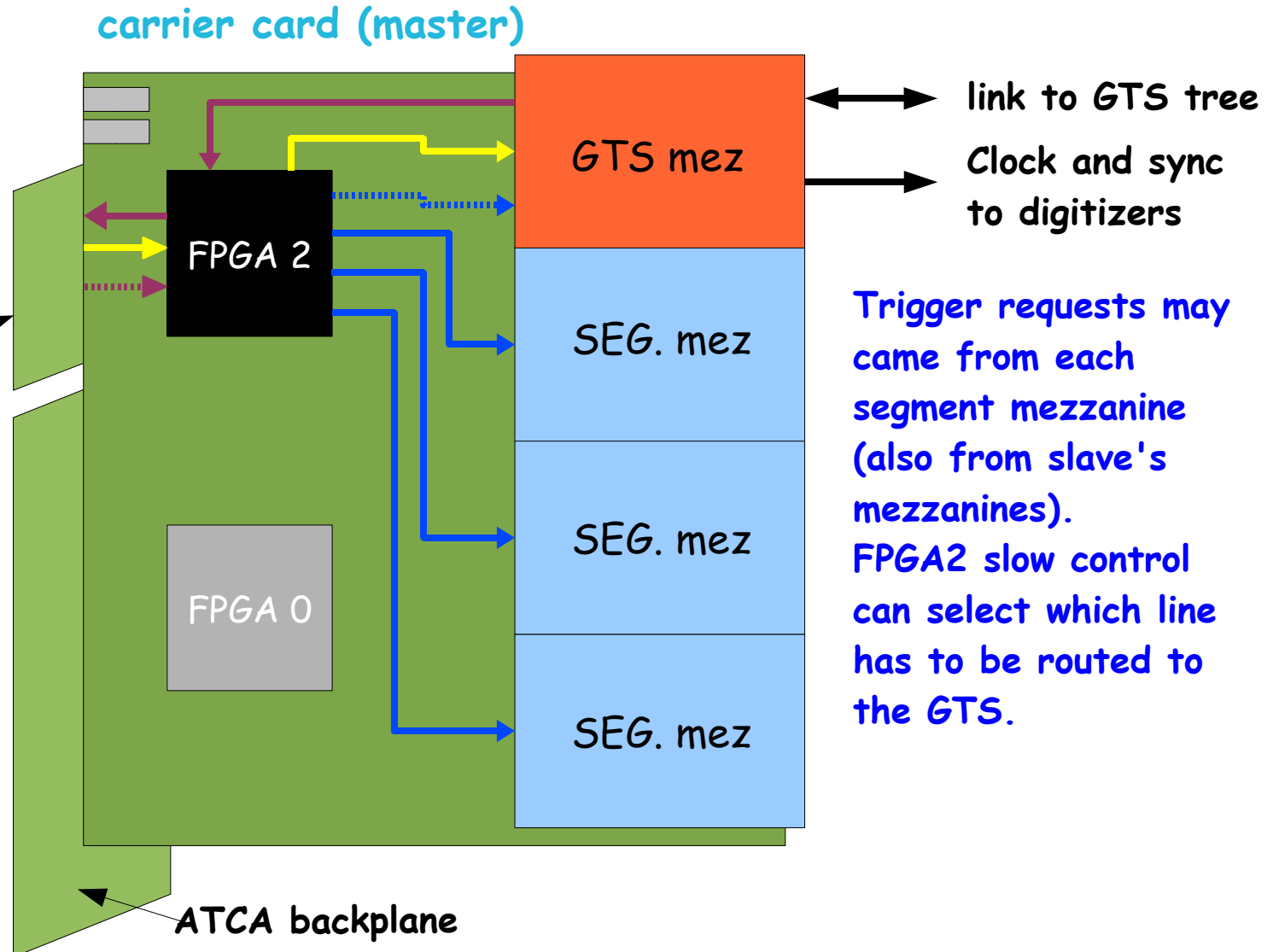
Carrier Overview

trigger distribution

FPGA2 distributes the trigger bus transitions to all the mezzanine slots equalizing the latencies. All mezzanines (master and slave) receive the trigger bus signals simultaneously.

TCLK board by IPN (Ch. Oziol)

Slave card is connected via the TCLK board through 6 LVDS serial channels @800 Mb/s each. There are 2 different version for FPGA2 firmware: master and slave. (dashed links are active only for slave board)



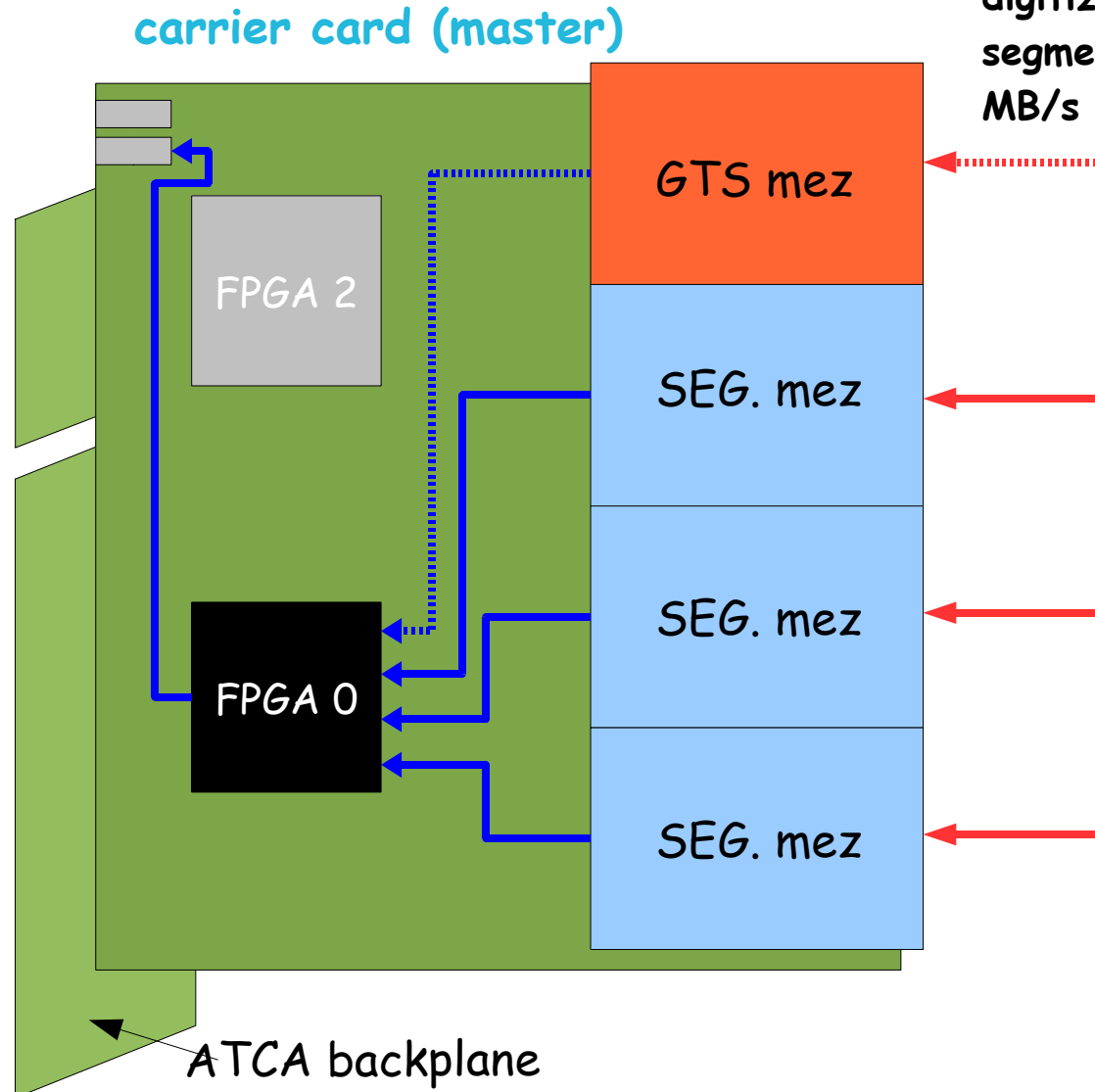
Trigger requests may come from each segment mezzanine (also from slave's mezzanines). FPGA2 slow control can select which line has to be routed to the GTS.

Carrier Overview data flow

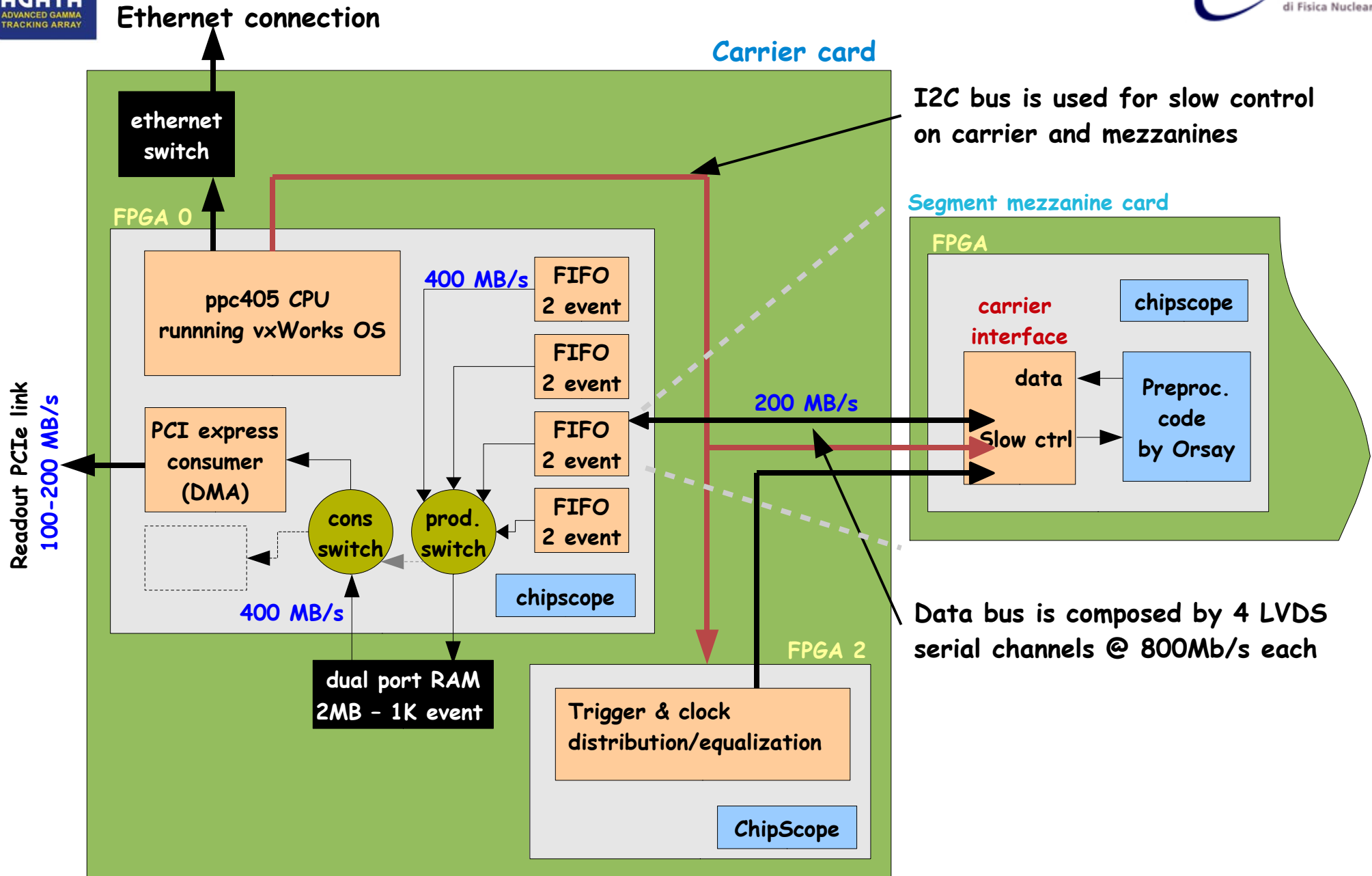
One DAQ machine reads out the data from master and slave (one crystal)

Preprocessed data comes out from the segment mezzanines formatted as packets. Each packet contains energy and timing information as well as a snapshot of 160 samples for each of the 6 channels around the trigger event. The length of the packet is 2048 bytes so the amount of data to read out one crystal is 14 KiB/event.

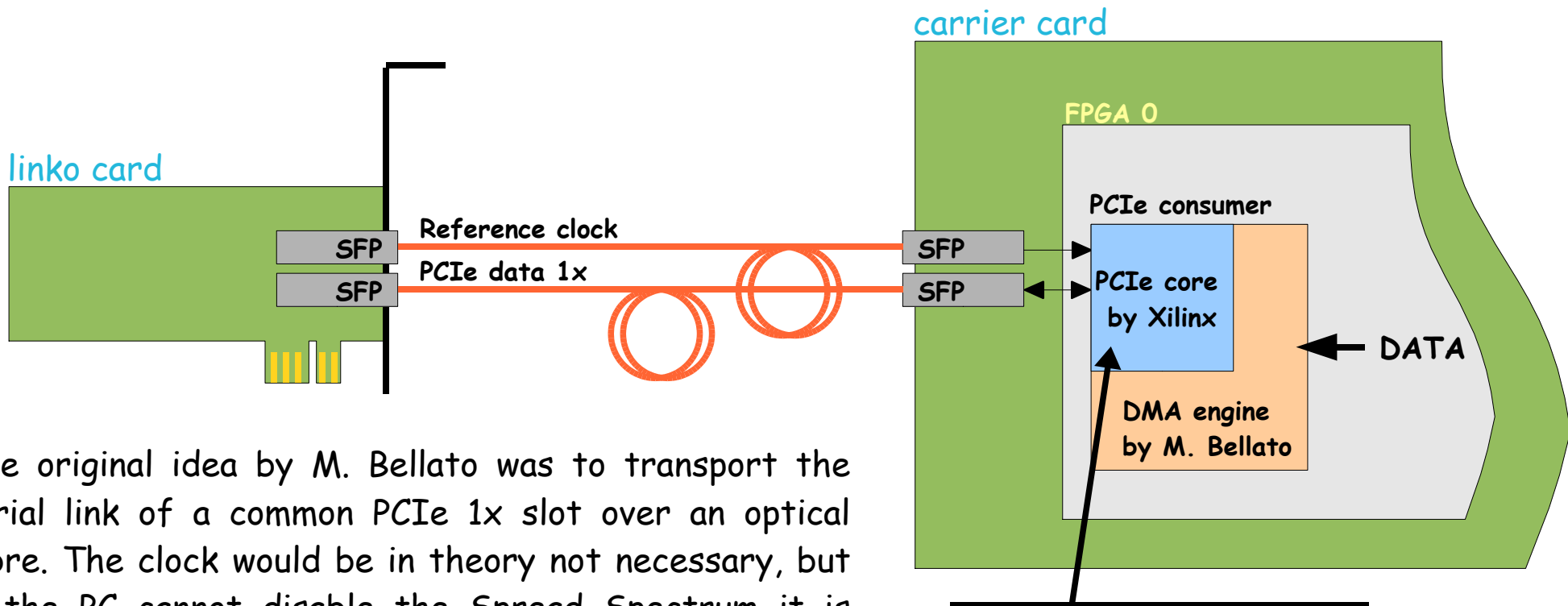
Raw data comes from digitizers 6 channel per segment mezzanine @200 MB/s



Firmware



PCI Express readout



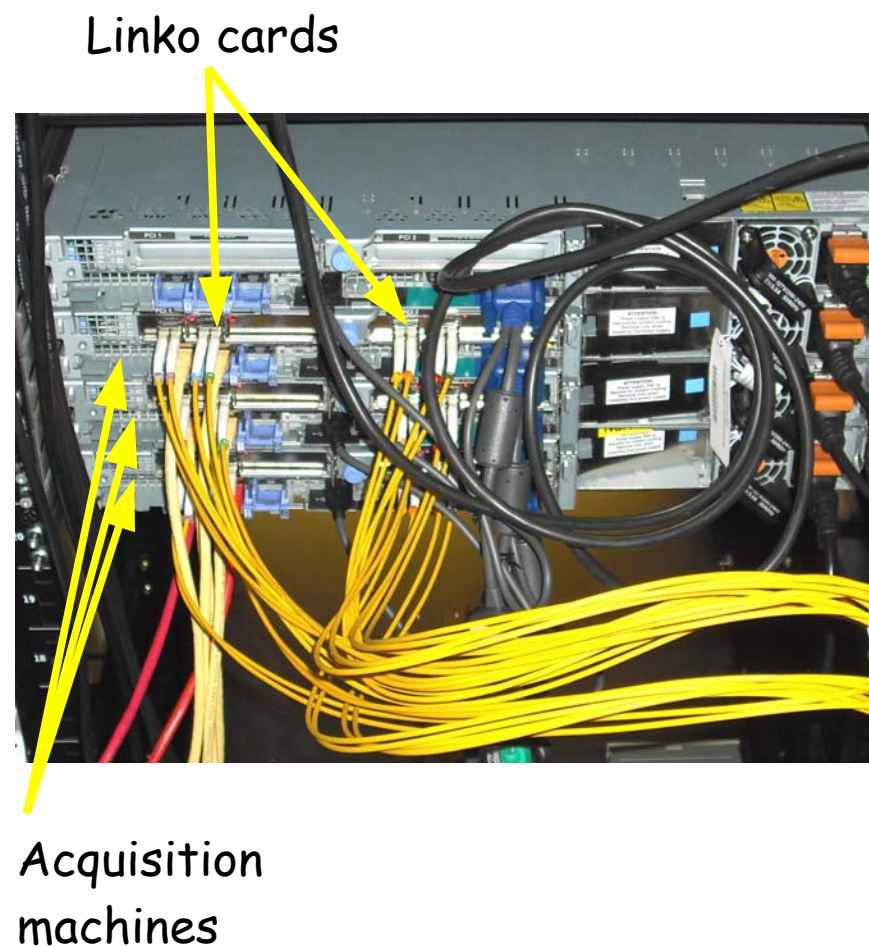
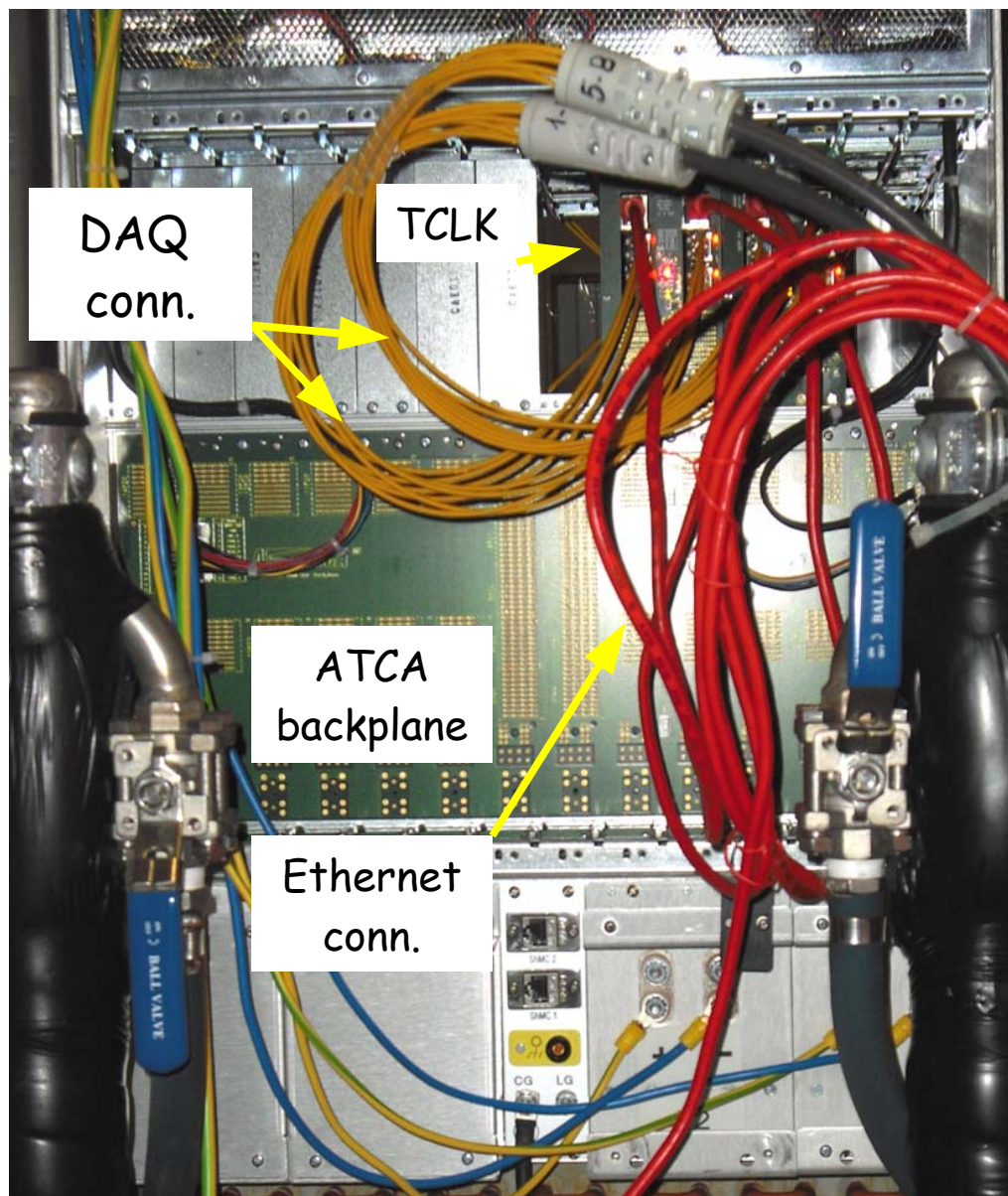
The original idea by M. Bellato was to transport the serial link of a common PCIe 1x slot over an optical fibre. The clock would be in theory not necessary, but if the PC cannot disable the Spread Spectrum it is needed. Some tests performed Padova with 2 connections on the same machine showed a DMA rate of ~400 MB/s (200MiB block).

The software driver has been developed by A. Triossi and D. Bortolato. It has been ported to the 64 Linux platform by X. Grave.

Devices are seen as `/dev/xdev0` and `/dev/xdev1`

During the PC startup the BIOS detects and configures PCI BARs properly. If the carrier is power-cycled the BIOS configuration is lost: there is the need for a configuration restoring mechanism in order to avoid a PC reboot.

PCIe readout

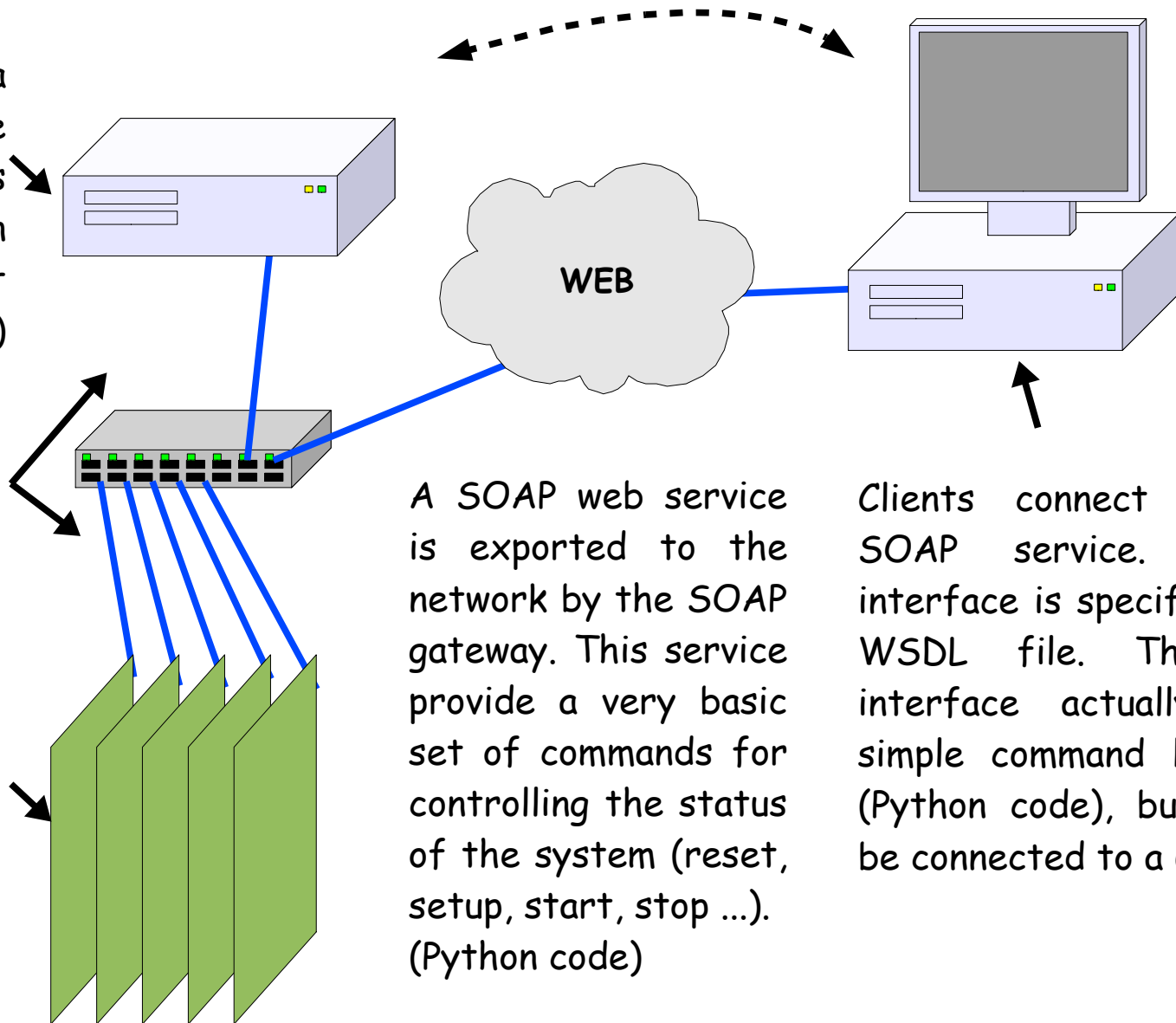


Carrier run/slow control

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)

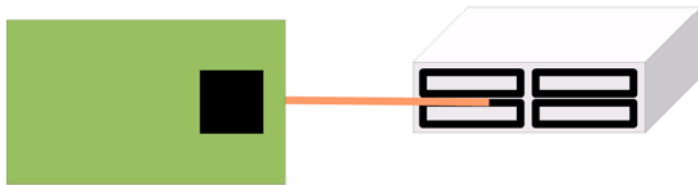


A SOAP web service is exported to the network by the SOAP gateway. This service provide a very basic set of commands for controlling the status of the system (reset, setup, start, stop ...). (Python code)

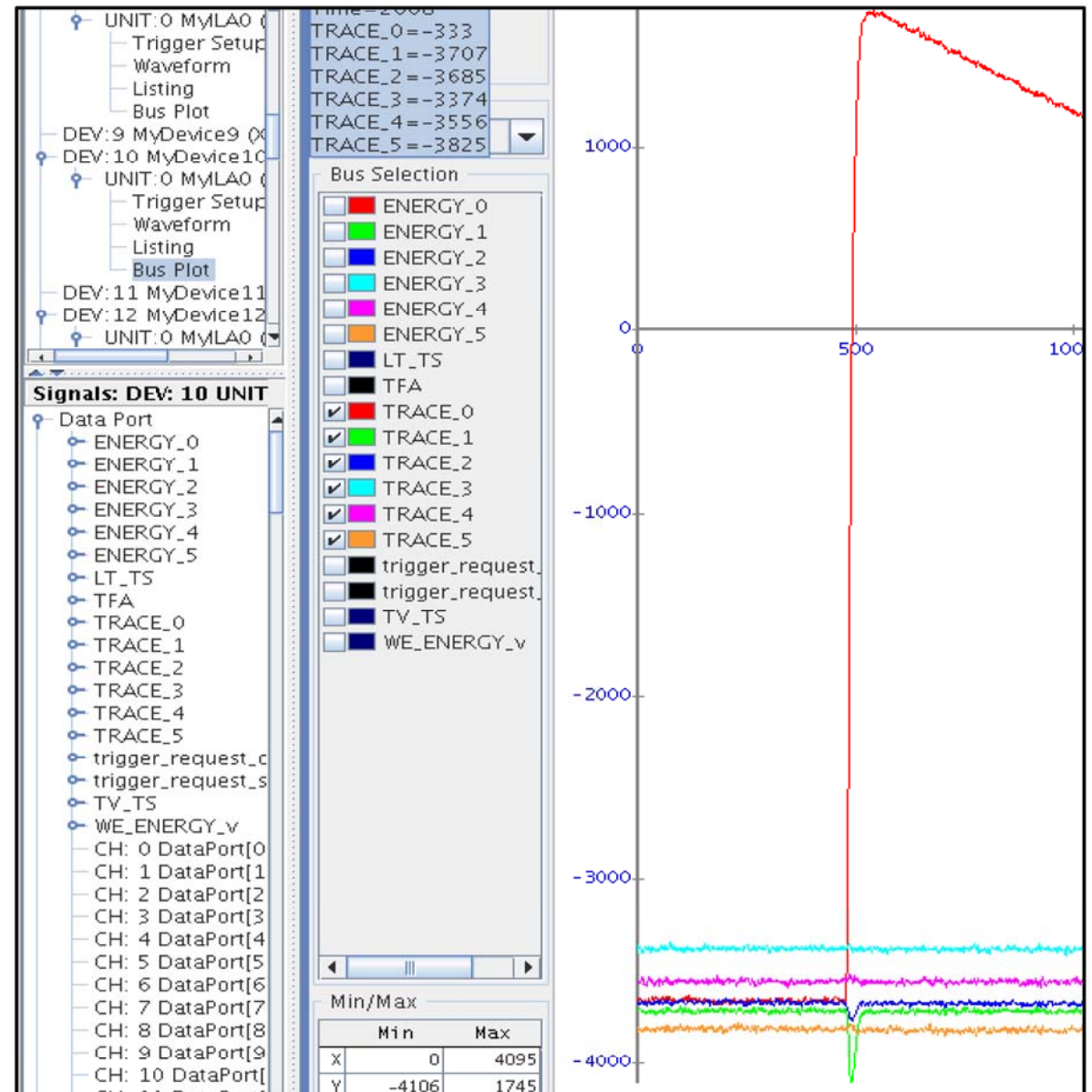
Clients connect to the SOAP service. The interface is specified by a WSDL file. The user interface actually is a simple command line tool (Python code), but it will be connected to a GUI.

Diagnostic channel alignment

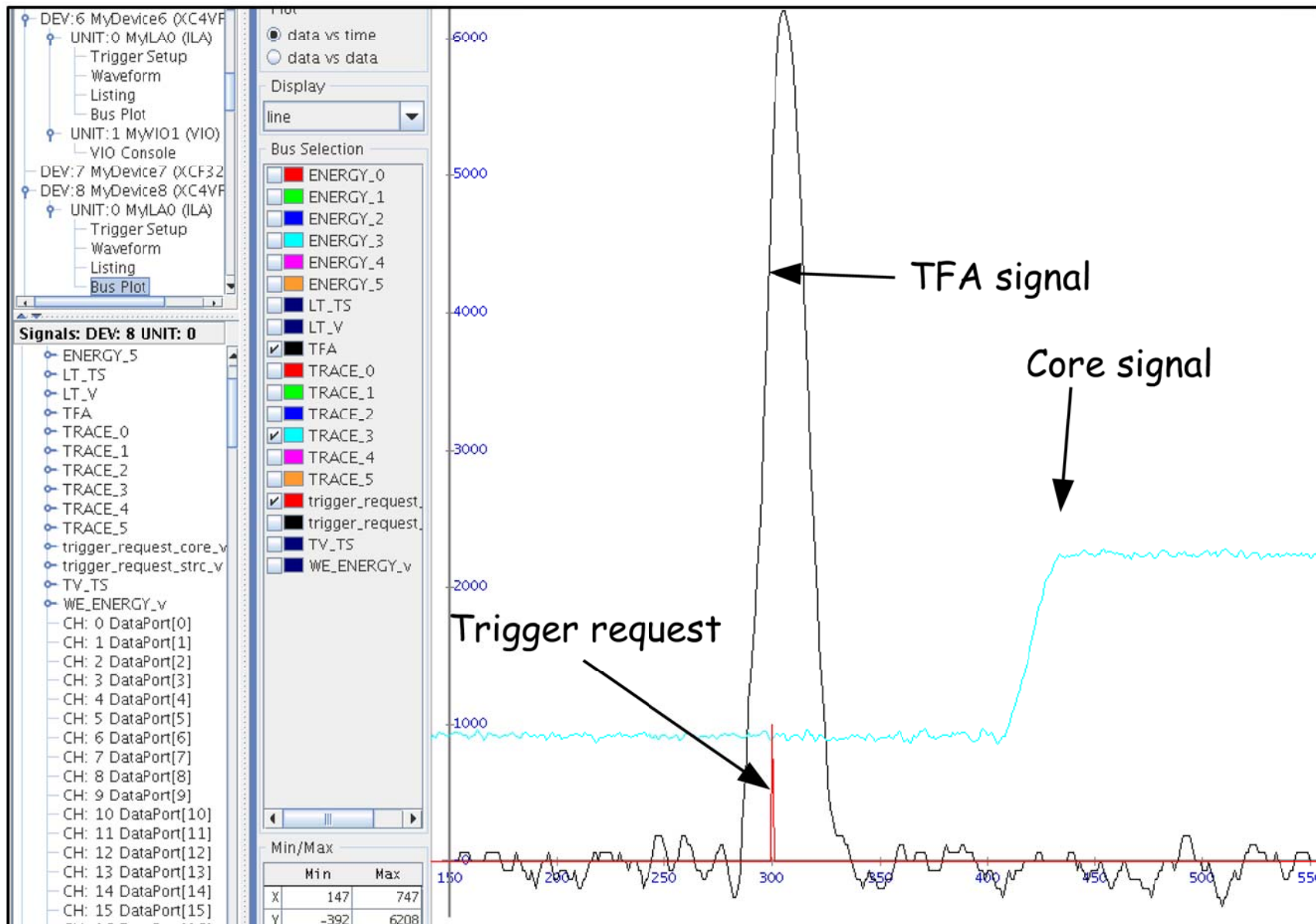
Each mezzanine has a ChipScope core on the main FPGA, which is used as a digital scope to monitor the channels alignment.



At present, the only way that we have to check if data channels between digitizer and segment mezzanines is good, is to look one by one to the traces received by the mezzanines, exploiting a VHDL based digital scope provided by Xilinx : ChipScope.



Trigger on the core





Set Up Procedure



Currently, the set up procedure involves the manual synchronization of several operations that have to be done on different subsystems. This is an example of what are the main steps (see also Samantha's manual):

- Switch on everything (digitizers ATCA crates GTS crate...)
- Start the GTS calibration procedure and wait for it to finish the first step.
- Launch the carriers management daemon and start the carrier setup.
- Wait for carrier and GTS procedures to finish.
- Send a setup 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.



To do list



Software & Firmware:

- Need to replace the chipscope manual monitor with an automatic mechanism.
- On the carrier is possible to add some second order checks and monitors as well as some histogrammers, counters, ...
- Improvements in the PCIe driver (mmap, interrupts, IOCTLs...)

Slow/Run control:

- Currently each subsystem is a black box where everything is considered working or not working (no topology information between subsystems). This means that is not possible to reconfigure/recover only a part of the system.
- Need for SC supervisors in order to coordinate subsystems with each other.
- Need to avoid DAQ rebooting the machines after every PCIe core reset.