AGATA DAQ Slow Control

Frédéric Saillant - GANIL

AGATA DAQ Slow Control

What is to be (slow-)controlled ?

- AGATA Digitizers
- AGATA GTS Trigger Processor
- AGATA ATCA carrier boards
- AGATA GTS mezzanines
- AGATA Core & Segment mezzanines
- Ancillary electronics

AGATA DAQ Slow Control

- First meeting June 10th 2008 at INFN Padova

- Discussion on Slow Control was late since :
  - electronics teams overloaded with preprocessing
  - setting up the DAQ box was the priority of the DAQ WG

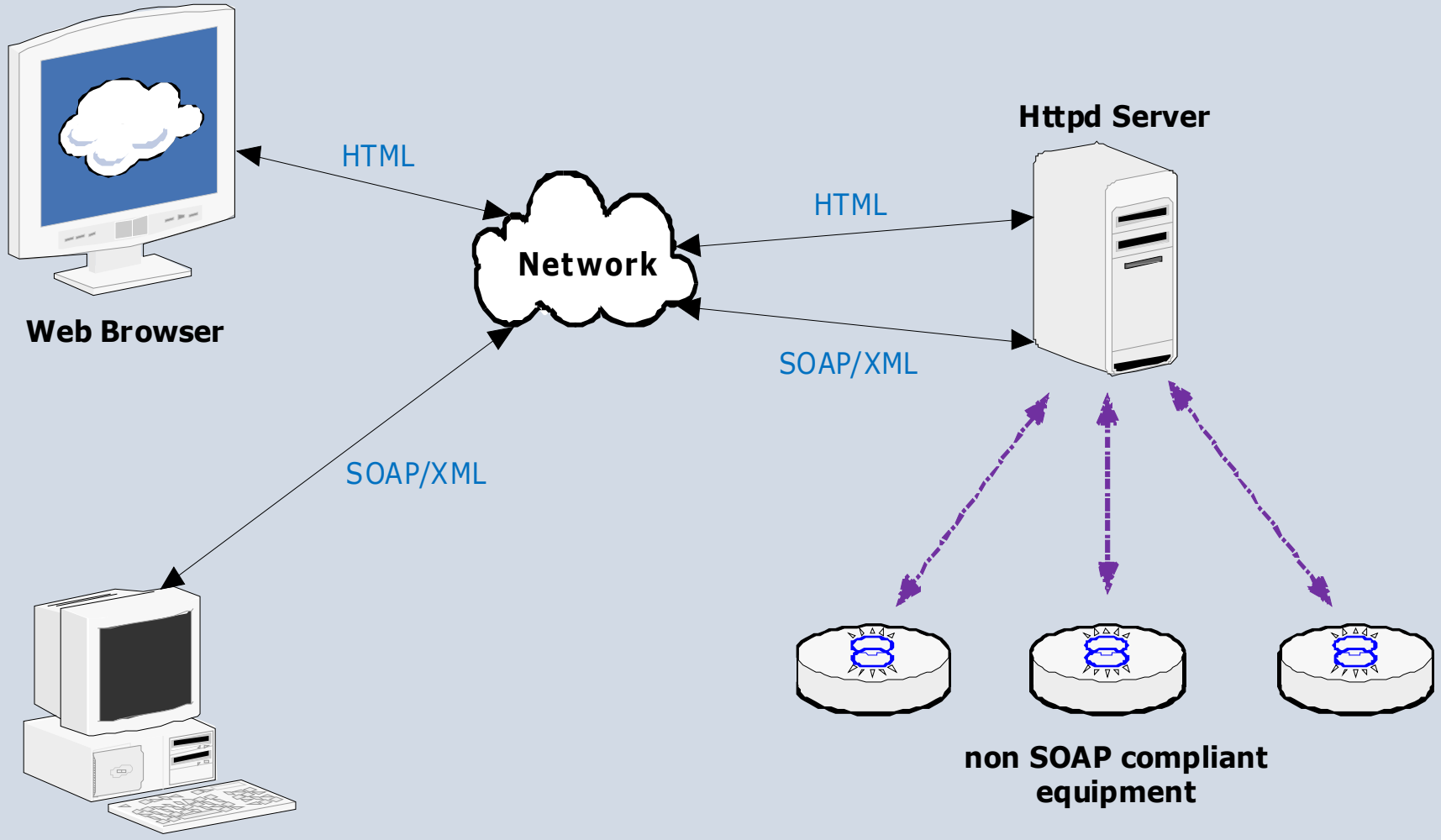
- First milestone is to operate one TC in October 2008

Digitizer Slow Control

- developped by Vic Pucknell (STFC Daresbury, UK)
- two Graphical User Interfaces :
  - tcl/tk GUI with SOAP/XML communications
  - standard Web browser with HTML and Javascript
- two 1U « pizza boxes » purchased by Daresbury for digitizer Slow Control, one sent to Legnaro

AGATA Digitizer Slow Control

SOAP/XML gateway



Digitizer Slow Control (cont'd)

- special tcl/tk GUI for digitizer diagnostics, with scripting facility
- temperature monitoring in tcl/tk GUI, with « histogram dowload » facility for external display
- to do :
  - detector based GUI
  - continuous temperature monitoring
  - implement « Run Control » WSDL interface
  - implement save/restore
  - pulser control
  - Time Over Threshold setup

Core and Segment mezzanines

- ENX software developped by CSNSM and IPN Orsay
- written in Ada
- runs under Linux on the Virtex/PowerPC and DAQ pizza boxes
- SOAP/XML communications
- tree structure :
  1. low-level ENX server on each mezzanine
  2. mid-level ENX server for each crystal
  3. only one top ENX server for the whole system

Core and Segment mezzanines (cont'd)

- register based generic interface implemented
- no graphical user interface yet since lack of manpower
- agreement to implement Vic's generic « Run Control » WSDL interface
- save/init/reset/start/stop commands to be implemented by mid-october
- monitoring :
  - first implementation mid-july
  - will evolve with time, need parameters from electronics team

ATCA carrier and GTS mezzanine

- software developped by INFN Padova and Legnaro
- VxWorks 5.4 implemented in Virtex/PowerPC for ATCA carrier card and GTS mezzanine
- access to registers through UDP/IP, with keyboard commands and script files
- no GUI yet, none foreseen for first periods of demonstrator
- switch to VxWorks 6.6 and SOAP/XML in the future but no manpower now in Italy for this
- many parameters to setup for the carrier but mostly fixed
- many things to monitor on the carrier

GTS Trigger Processor

- software developped by Luciano Berti (Legnaro)
- Trigger Processor has a programmable part one needs to compile
- steps to setup Trigger Processor :
  1. Partition the system
  2. Set coincidences
  3. Compile
  4. Load
  5. Run

GTS Trigger Processor (cont'd)

- Luciano is needed to setup experiment, despite fixed configuration at the beginning
- no GUI (no manpower)
- timing part of GTS :
  - about 50 steps needed to setup
  - alignment with a pulser
  - test the system with fake trigger request
- a trigger pattern is needed for Event Builder, Xavier and Luciano will discuss the implementation

Ancillary electronics

- Legnaro ancillary electronics based on VME
- existing control system : MIDAS (Daresbury)

Network architecture

- one software server per piece of electronics
- one C class for slow control network
- MAC and IP adresses assignment :
  - digitizers already have serial number and MAC address
  - assignment needed for carriers and mezzanines
  - AGATA database should contain serial numbers and MAC addresses

Requirements for GUI developments

- would be better to have one common GUI
- some parameters should be hidden (eg digitizer shutdown)
- no permissions/passwords on parameters
- main high-level window, then access to more advanced ones
- one big « Setup everything » button
- application will evolve with experience gained using the system
- should operate using detector logic rather than horizontal layer logic

Data Quality Monitor (DQM)

- checks that each piece supposed to give data does indeed
- checks coincidences
- should alert Run Control which stops experiment in case of corrupted data
- DQM working group should be setup
- people from Legnaro should be involved
- DQM WG should define procedures to check the system and to perform calibration
- manpower ?

Save/Restore policy

- database vs file :
  - DB is more general
  - DB needs more work
  - DB needs to be installed in all places where system is used
  - if we start with file there will be little chance to move to DB
- Vic's proposal : text file implementation with self-describing path name
- save actions should be logged in Run Control logger and electronic log book