Quality Assessment for the NA61/SHINE experiment at CERN

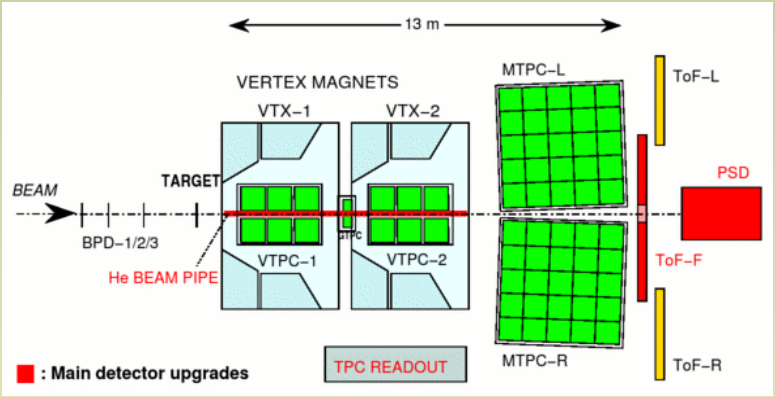
Alexander Sadovsky
sadovsky@inr.ru
Institute for Nuclear Research RAS
for the NA61/SHINE Collaboration
prepared
for
XXXI INTERNATIONAL WORKSHOP
“Neutrino physics at accelerators”
DLNP, JINR, Dubna,
January 27-29, 2009

- NA61 upgrade
- Offline (DST) QA
- Online QA
- Web-interface
- Bookkeeping

NA61/SHINE upgrade

NA61/SHINE experimental setup

- 4 large volume TPCs + 1 gap TPC (63 subdetectors)
- 3 Time-of-Flight detectors
- 6 Beam Position Detectors
- 1 Projectile Spectator Detector



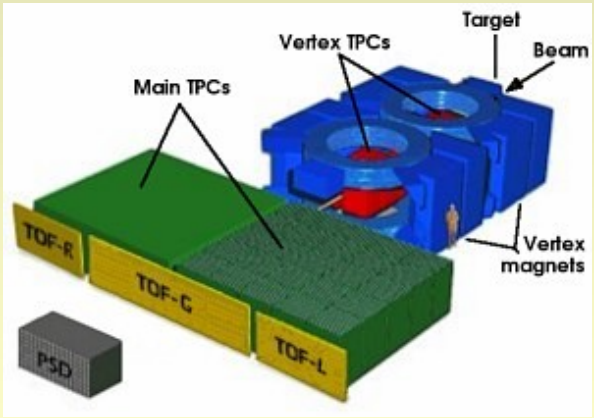
Experimental runs

- 2007 /physics/ ~600 data files (p+C)

  TDC readout & DAQ-upgrade *(TPC event rate~100 Hz i.e. NA49x10)*
- 2008 /DAQ-test/ ~100 data files (LHC incident)
- 2009 /physics/ ~10000 data files are expected (at least)

Problem to keep under control:

- Detector control plots: [hist]x[file] ≈ [100]x[10000]
- Possible only with semiautomatic tools
- Quick view-access to any (prepared in advance) QA plot
- Finally, DST file validation is needed!



Offline DST QA

h**istogram **history for DST QA was designed to allow for:

- help during DST preparation (DST="Data Summary Tape")
- time stability control
- expert decisions on a standard data representation view

It assumes:

- that experts provide a set of ~ (1...12) QA plots for each detector
- these histograms are saved per each DST file to a corresponding ROOT-histogram file
- GUI program (*hhistory*) is used to control long term detector stability
- decision is taken based on (logbook entries & expert opinions) for a limited set of files
- semiautomatic control in case of numerous amount of DST files with help of stability control

Long term aims:

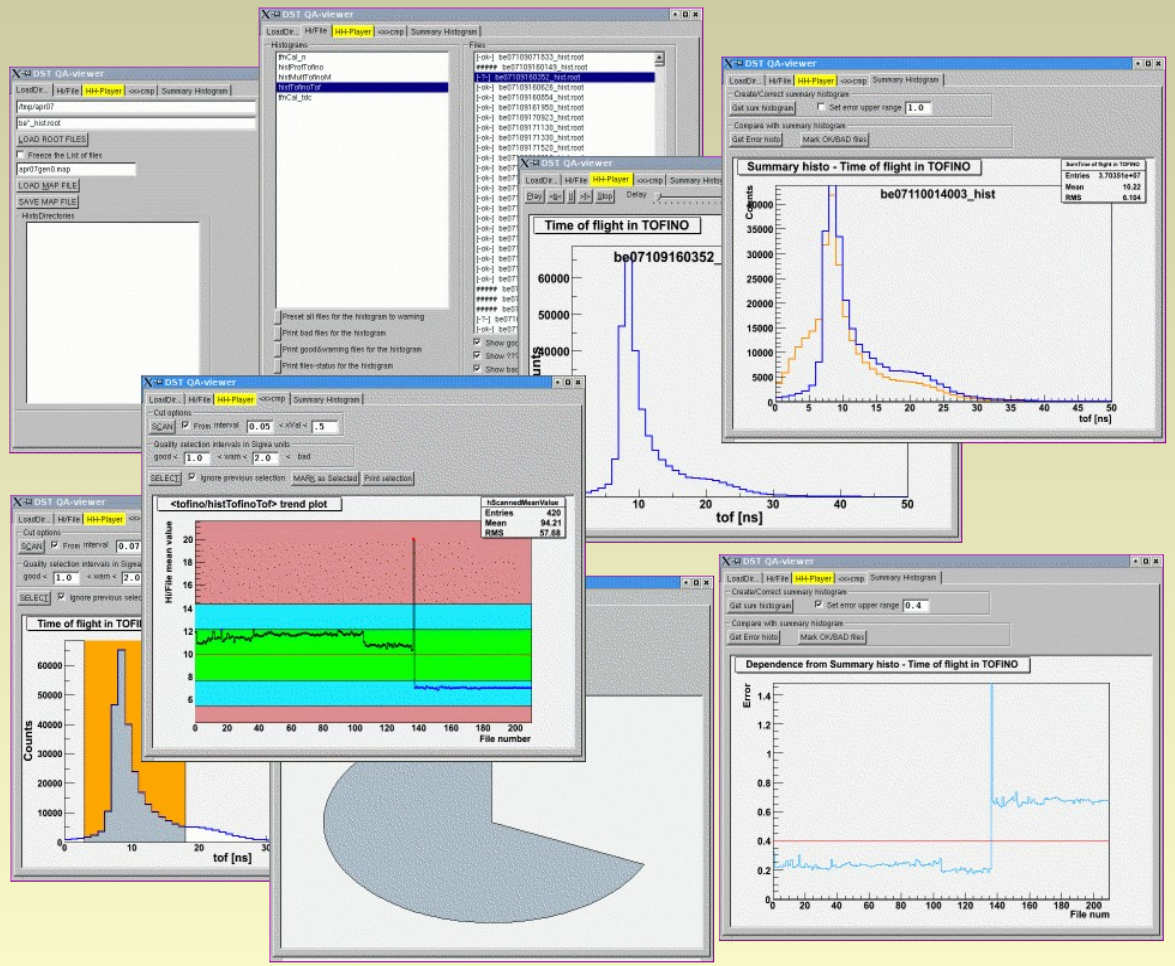
- incorporate DST-QA into the WEB-server with tree-like navigation (~bookkeeping)
- quick view-access to any (prepared in advance) QA plot
- DST file validation

“hhistory” for offline-QA

What is it for?

+ light-weight GUI application
+ Linux/ROOT based
+ general purpose tool
+ histogram player
+ semiautomatic
+ <x> comparator
+ 1-dm shape control
+ 2-dm shape control
+ pattern ~ expert selectable
+ collaborating usage
+ all kinds of printouts
+ since 2003 (HADES@GSI)

Helps in understanding of what was happening during past experiment

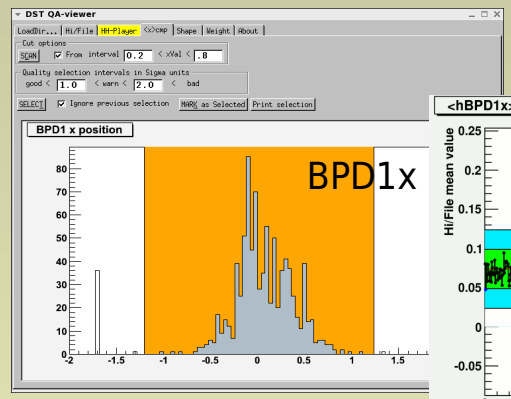


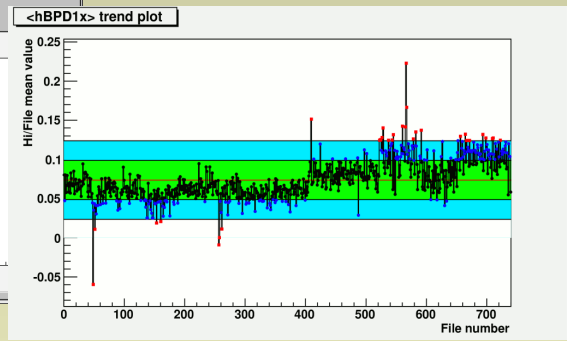
2007-2008 developments: in collaboration with T.Solovieva /diploma thesis/

Examples of work

BPD example

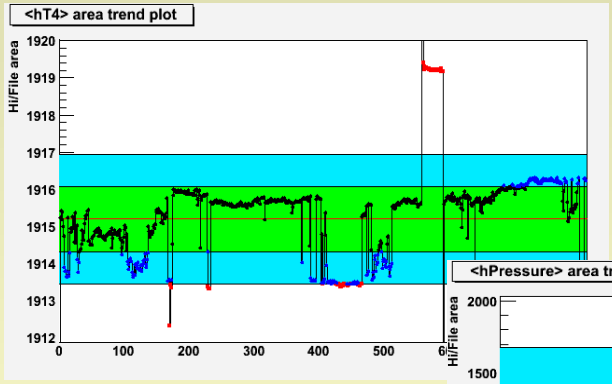
peak mean value distribution
and its trend plot

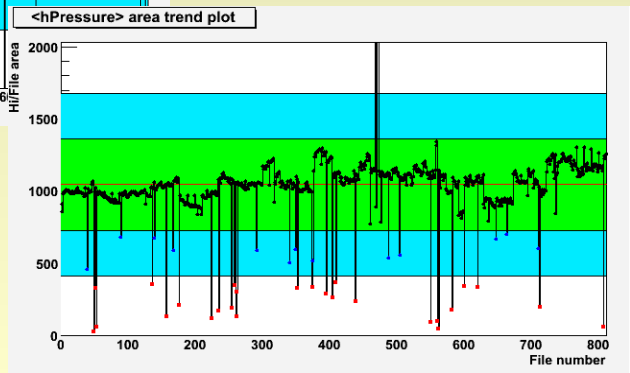


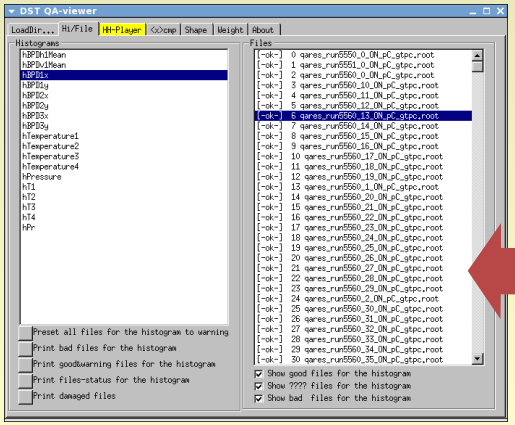


Pressure and temperature

examples of trend plots
as seen from 2007'exp data







Results

report is stored
in the file list
also as a text file

hhistory for online-QA

Online? – rather - fast off-line as a solution for online quality control
- Not to be mixed up with slow control
- Created for shift crew, to help in realizing of drastic changes of experimental data flow
- New DAQ: each file after 10-20s (delay up to ~3-5 minutes for full chain DAQ→histogr→Online-QA)
- May involve deeper level of data analysis compared to “real online-control”

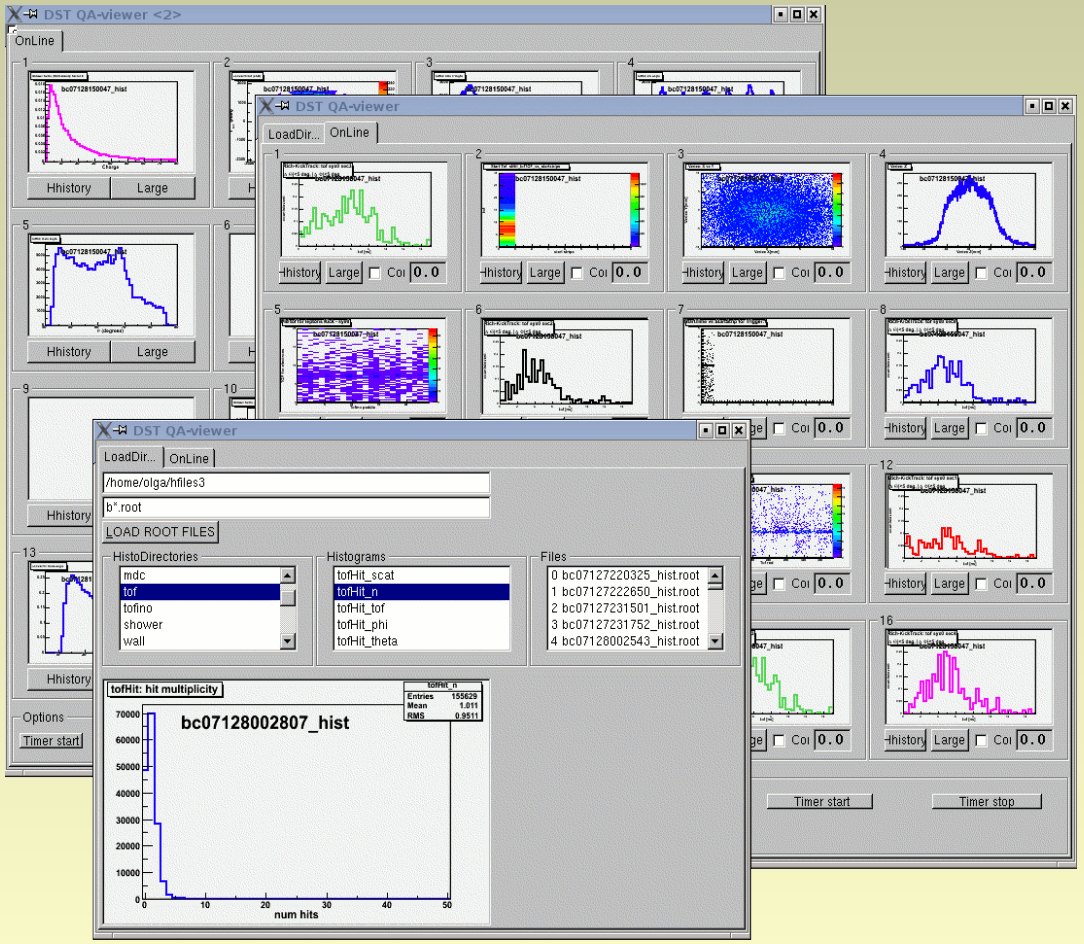
Histogram history GUI-application was extended to allow for:
- Display the last available DAQ-file status (pre-analyzed into a set of control histograms)
- Quick look up to the histogram status reverse in time (file)
- Creation of trend plot for any 1-dm histogram “on-fly” by user request
  - Mean value comparator
  - Kolmogorov-Smirnov test (1-dm and 2-dm)
  - Chi^2 test
- Specify multiple layouts (4x4) and display them
  - in built-in window or
  - in pop-up windows

Status on Oct-2008:
- Beta tests with 2007-data in DAQ→QA imitated mode
- Version 3.14: several bugfixes
- Installed at runna61@na61pc002
- Unfortunately was not really tested in real experiment

From offline- to online- QA

Expansion towards online-QA

+ for NA61/SHINE
+ online option (fast offline)
+ data control for shift crew
+ histogram shape history
+ latency < (1..5) min depending on DAQ-QA flow
+ multiple online windows
+ page layout editor
+ interaction with offline module at any time

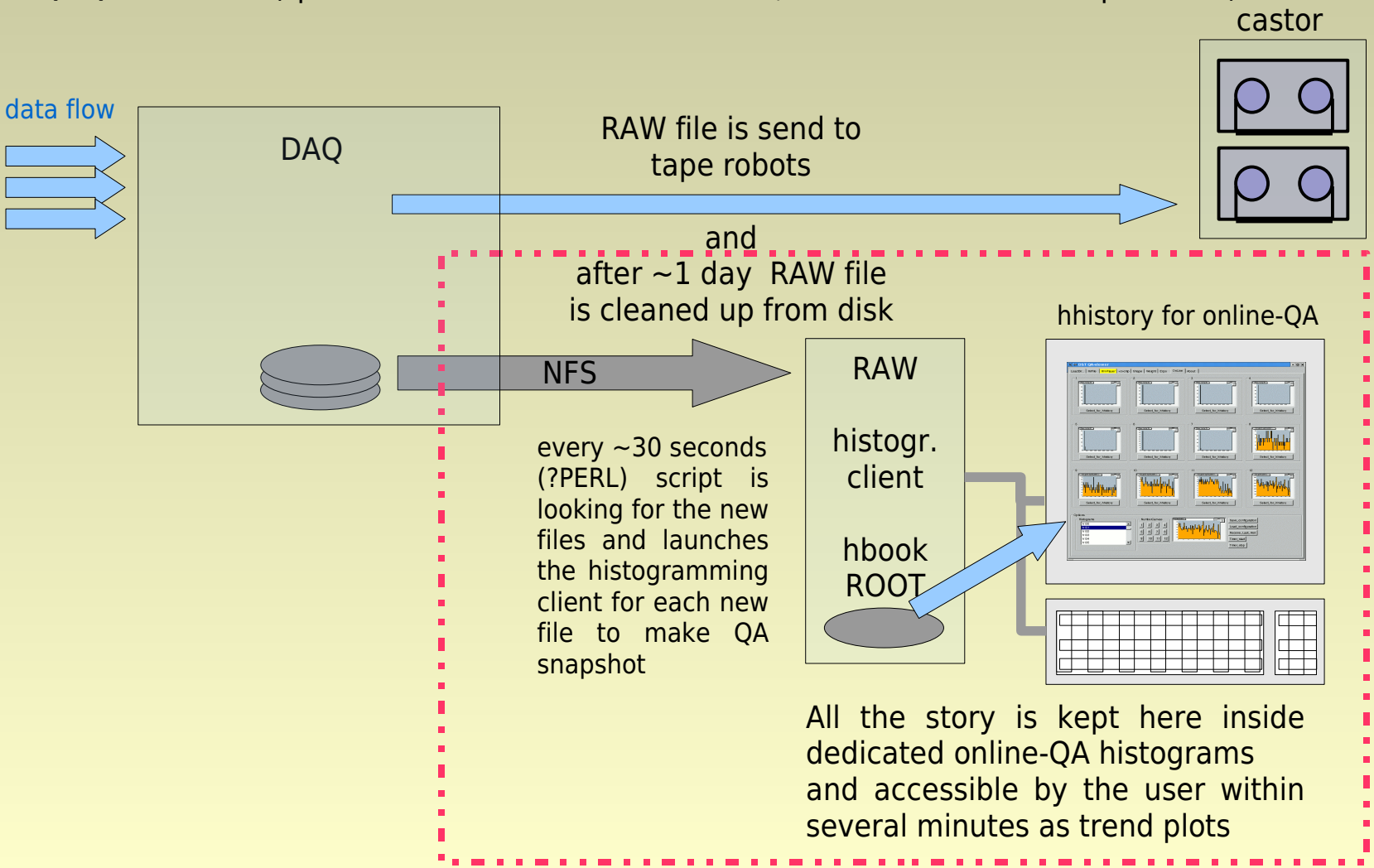


Combines offline problem recognition power for just in time application during data flow at experimental run

2008-2009 developments: in collaboration with O.Busygina /diploma thesis/

Online QA functionality*

DAQ→QA data flow (*priv. communication with A.Laszlo, to be tested in real experiment)



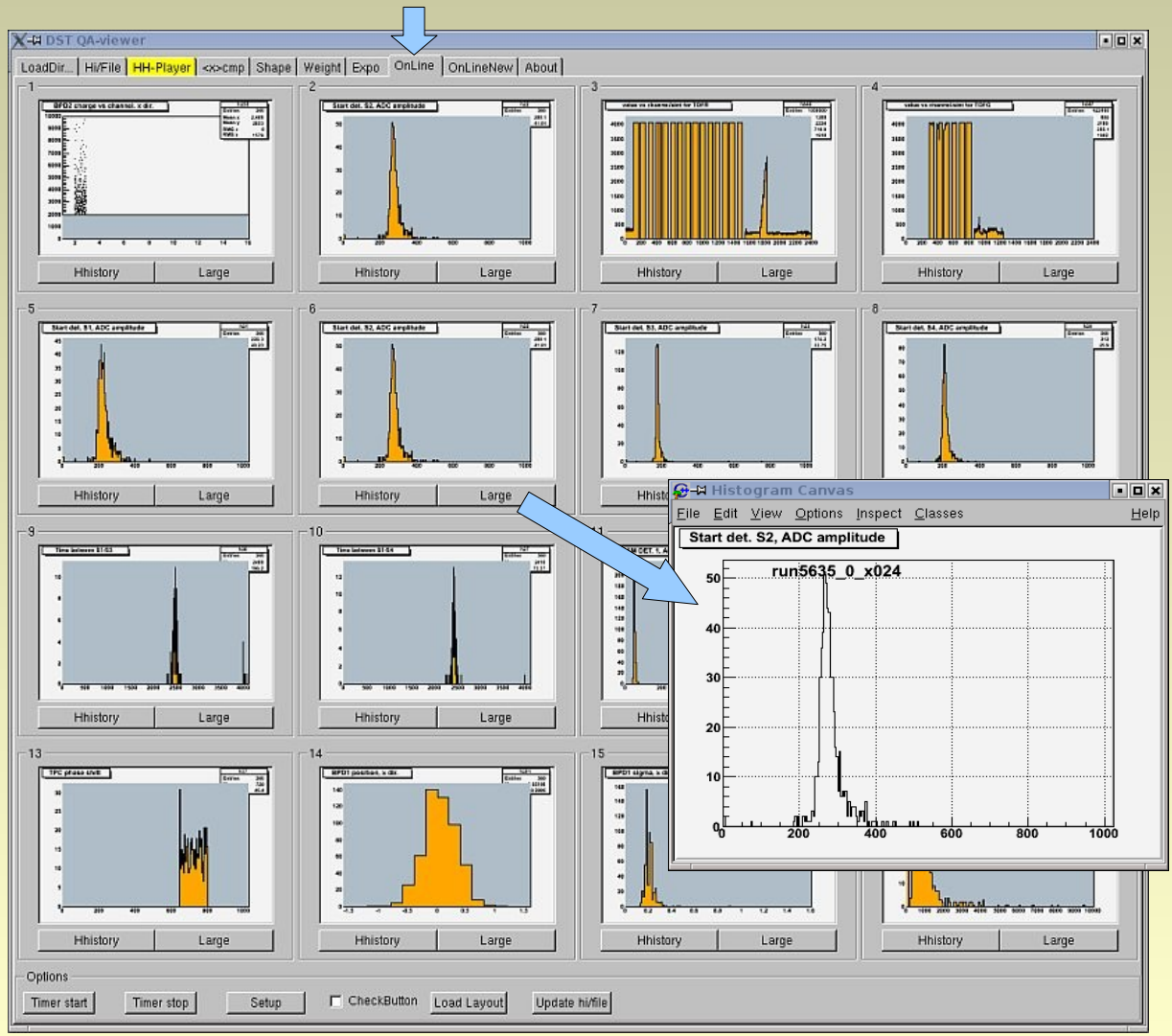
Online QA extension

Prototype v3.14:

supports for:

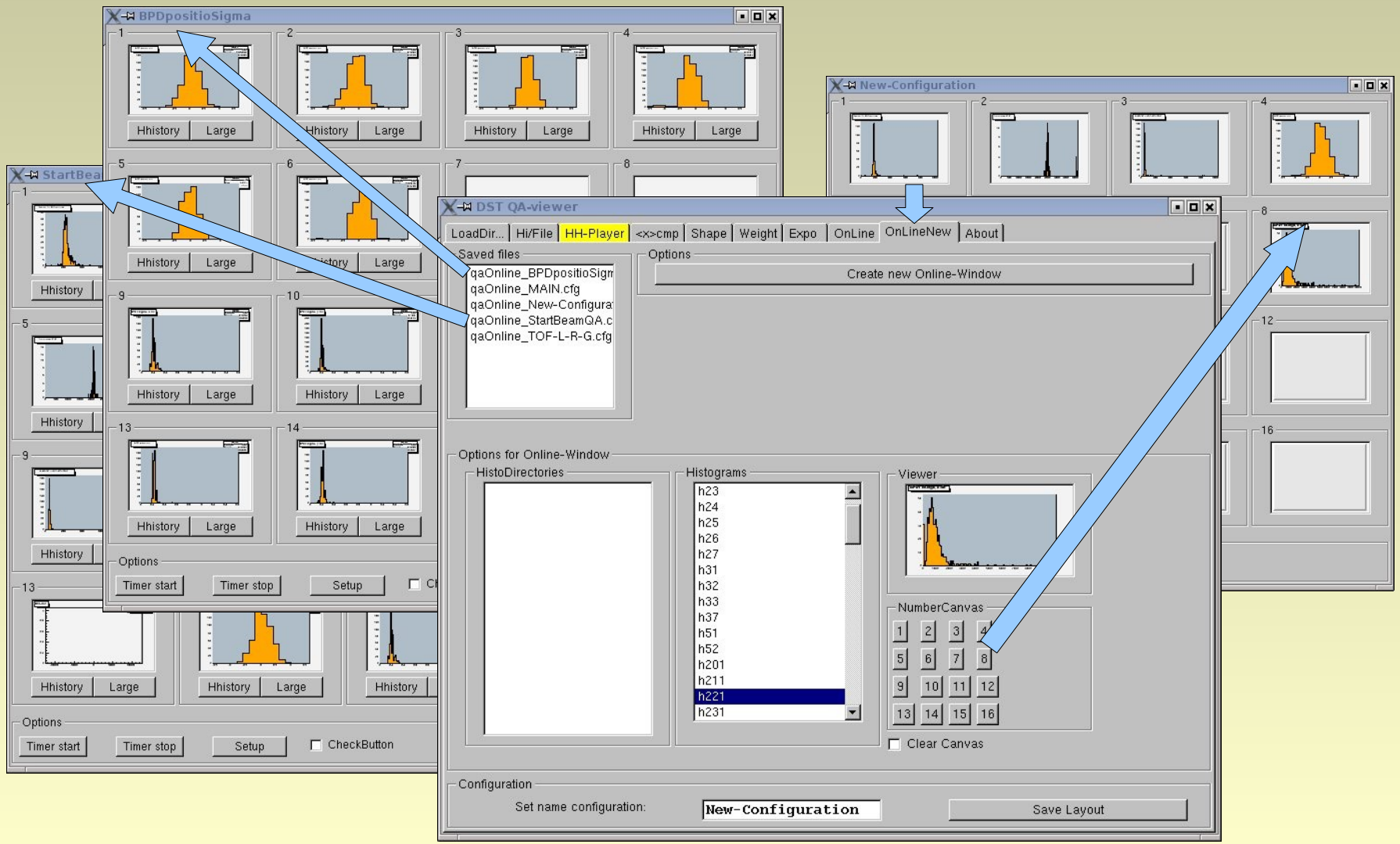
- edit/load layout (16 hists),
- GUI histogram selection,
- show enlarged view,
- saving configuration and
- online update (from the latest ROOT file)

Allows “history” investigation of for an individually-selected plot.



Diploma project:
Olga Busygina

New layout / open window



Improving NA61 web-site

NA61/SHINE web site
https://na61.web.cern.ch/na61/xc/

What is it for?

Small redesign of the overall view:
- larger fonts
- less freedom for picture positioning

Easy import of simple HTML texts from anybody
- e.g. *~/user/public/www/somefile.html*

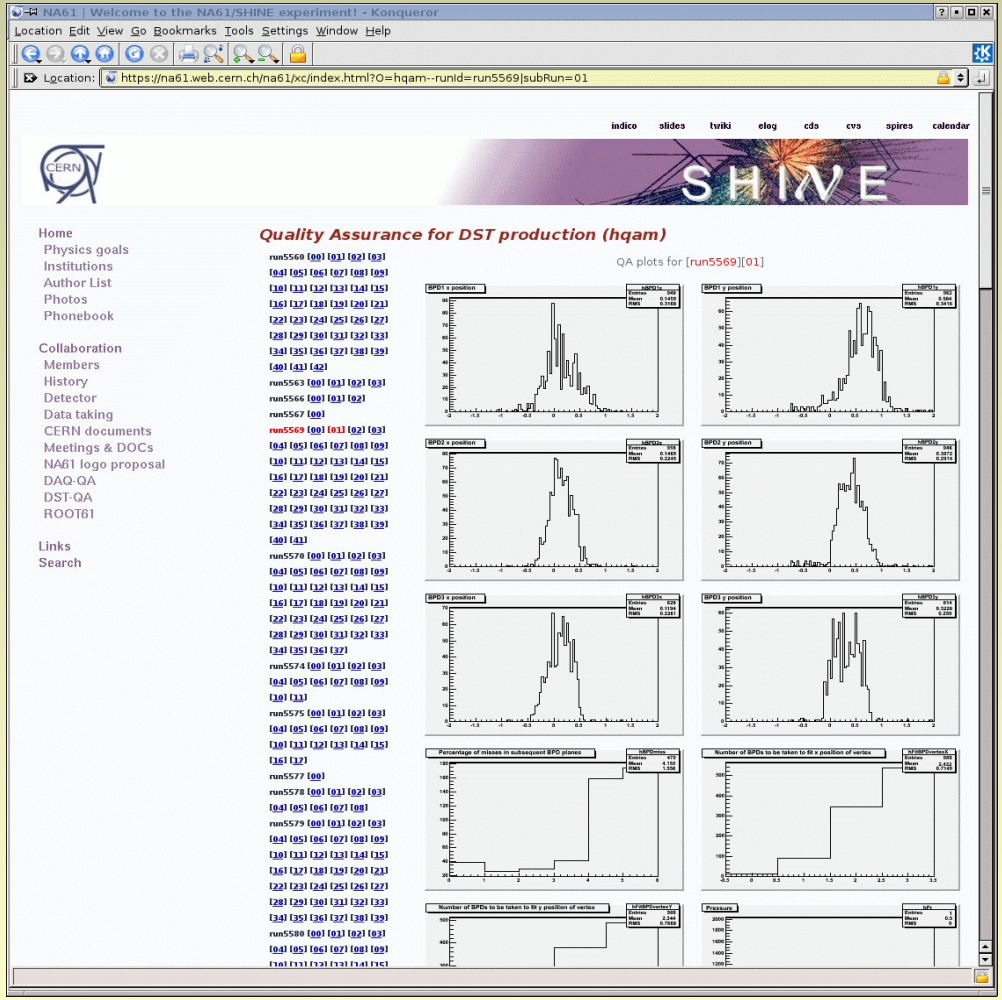
Run auxiliary CGI scripts on CERN server (for all kind of work which is suitable for a web-server)
- \+ keep all links in one place (not redoing standard tools like: TWiki, indico, ...)
- \+ online documentation generation
- \+ bookkeeping organization
- \+ possibility to organize centralized parameter storage
- \+ nearly online/offline monitoring for run/DAQ/DST status from outside
- \+ requests from external databases if necessary
- \+ possibility of simplified distributed documentation
- − not for multiuser work at a low cost (rather use dedicated systems TWiki, CVS/CVN etc.)
- − not for online/remote/slow control (due to possible network problems)
- − not for CPU-intensive jobs

Offline-QA web interface

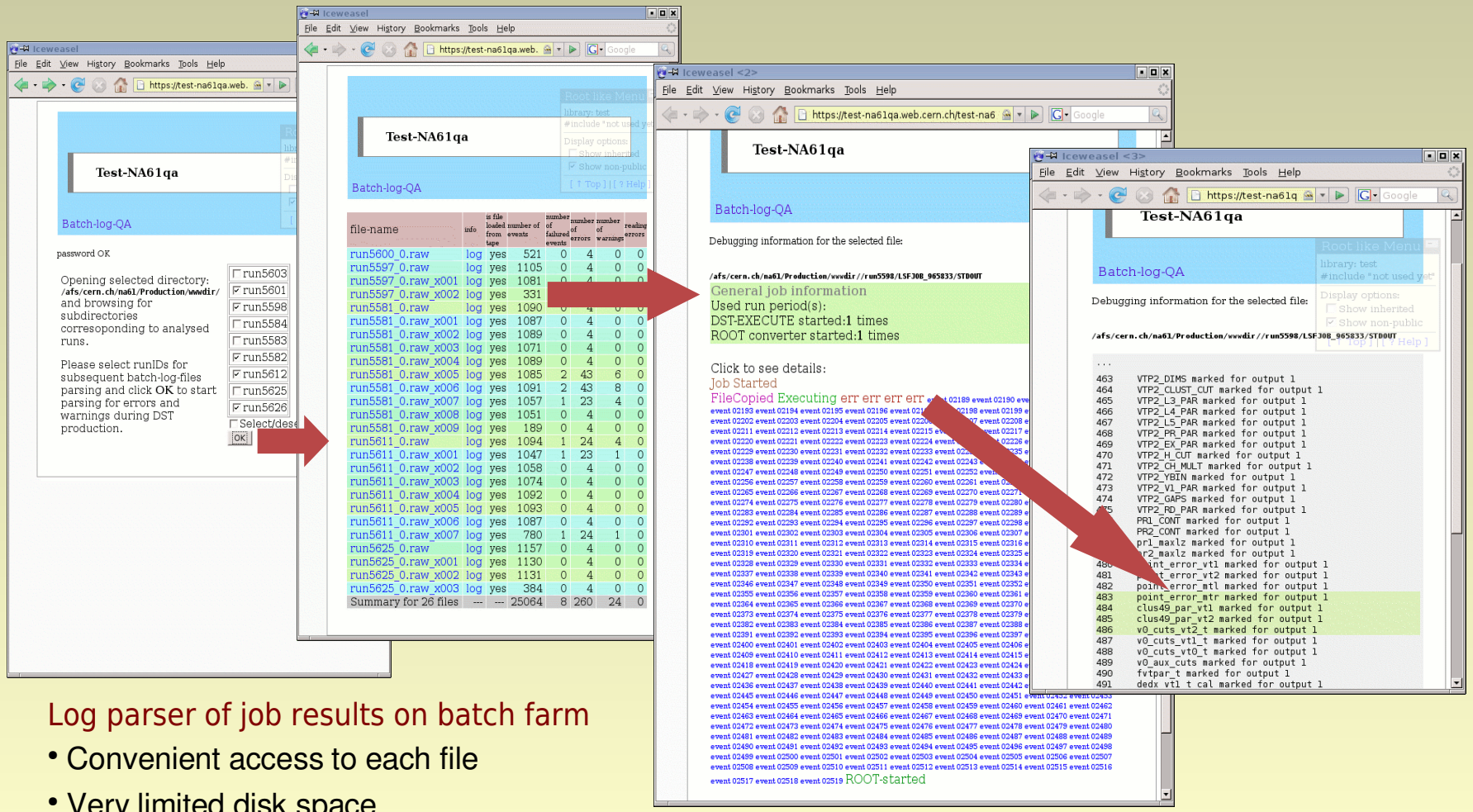
NA61/SHINE website development

- CERN hosted under UNIX
  - /-/ storage limit to 800Mb
  - needs external disk capacity DB?
- self made CMS
- employs file system, CGI/HTML/CSS
- Best suited for QA access
  - from outside CERN
- Offline-QA
- DST-batch-jobs log-QA parser
- ? Bookkeeping
  - /requires ~ 10-50Gb disk space/

Shall it be read-open outside the collaboration?



Log parser for batch jobs



Log parser of job results on batch farm

- Convenient access to each file
- Very limited disk space
- Relatively slow for simultaneous parsing of many files on default CERN's web-server computer

Bookkeeping

Task(s) from physics to data:

It is good to remember conditions which were used while producing the Data Summary Tapes (DST) as well as for simulation.

It is convenient to have possibility to cheek the data taking conditions and performance of certain detectors for any beam time starting from “physics” and ending by the Quality plots, calibration parameters etc.

Problems:

Computations are done on various computers Possibly at different research centers (GRID)

On the early stage calibration- (and also simulation-) parameters might be stored/created not at central repository.

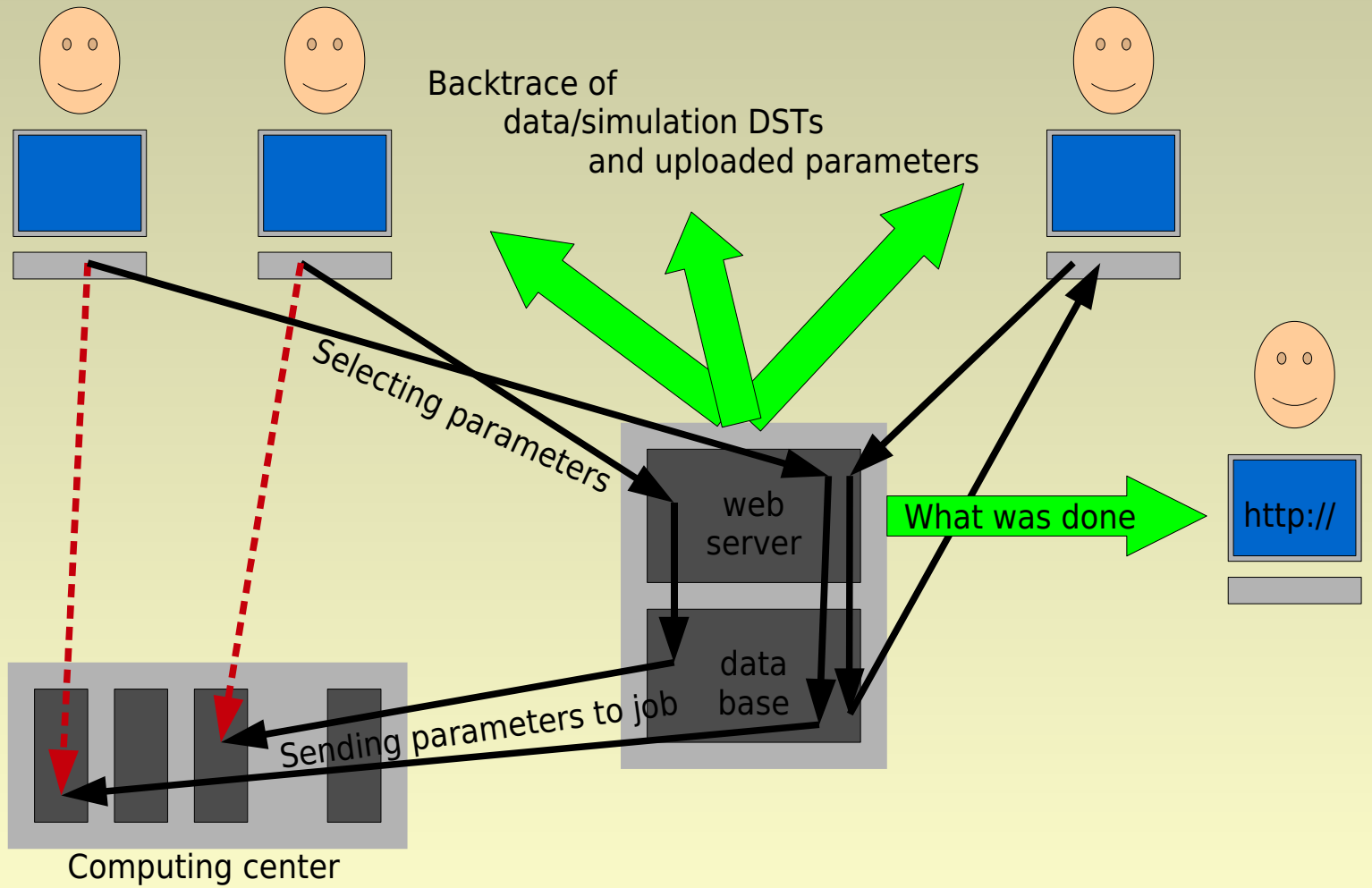
It's hard to convince people to commit detailed documentation.

Possible solution:

We can organize central gate responsible for the user interface between a standard computation task and the user. As soon this is done we have possibility to trace what jobs and under which parameter conditions were computed.

This can be realized as a web-form page which is creating certain start-up script which is used as a 1-st step of any analysis. If this is done conveniently this will also be profitable for the end user to use that form instead of programming start-up script from scratch every time.

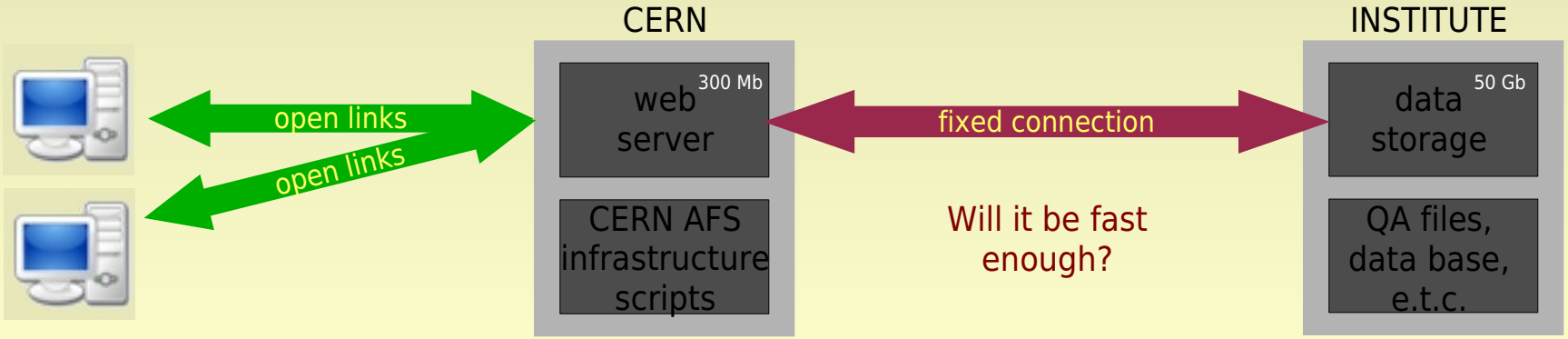
Bookkeeping proposal



Web server limitations

- CERN rules:
  - Own AFS quota < 800Mb
  - Access to other users within AFS if directory has proper permission
  - No access to tape robots
  - Own PC located in the office is not secured against AC power loss.
  - Own PC (web+data base server) can not be located at comp.center.

- Possible solution (to be tested):
  - Remote computing center for data storage (also for the data base)
  - Web server information exchange using ip↔ip connection
  - bookkeeping test to be done.



Current status

- DST batch-log-parser:
  shell and web supported (still some problems with permissions)
  demonstrated possibility of work, to be tuned into common interface/DB.

- Offline DST QA:
  more histograms to be added by detector experts.

- Online QA:
  integration with offline QA, bugfixing,
  real test to be done in 2009.

- Web-site/QA:
  prototipes are ready,
  external data-storage is under consideration,
  bookkeeping test to be done.

BACKUP SLIDES

NA61 data taking plans

2009: August 12 to November 16

p+C at 31 GeV/c	3 weeks (T2K)
π+C at 158, 300 GeV/c	2 weeks (C-R)
p+p at 6 energies	6 weeks (SIM)
p+p at 158 GeV/c	2 weeks (high pT)

2010:

p+p at 158 GeV/c	11 weeks (high pT)

2011:

30+30 at 6 energies	6 weeks (SIM)
p+Pb at 158 GeV/c	6 weeks (high pT)

2012:

10+10 at 6 energies	6 weeks (SIM)
p+Pb at 6 energies	6 weeks (SIM)

2013:

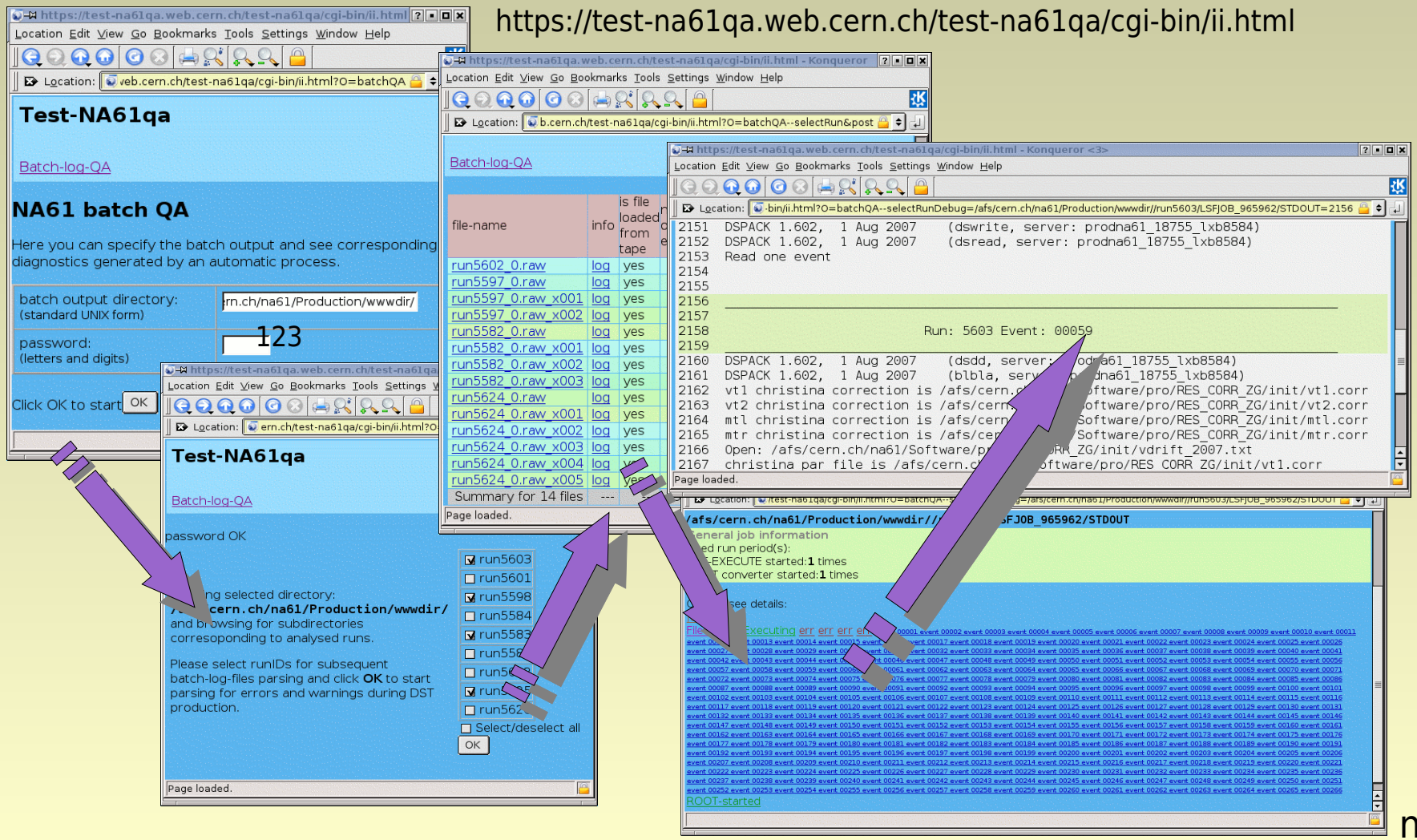
100+100 at 6 energies	6 weeks (SIM)

6 energies are: 10, 20, 30, 40, 80, 158 GeV/c.

Expected data yeld per year ~ 12 weeks ~ 25000 files (of 1Gb size) per year!
Strong demands to QA!

Log-file parser for DST prod.

https://test-na61qa.web.cern.ch/test-na61qa/cgi-bin/ii.html



histogramming client

Since *hhistory* does nothing but displaying and analysing histograms from ROOT-files located as a certain directory one needs a client which is creating and filing such histograms.

Currently *check_raw_zoltan* -client is used to create control online-QA histograms
- + was already used before
- + works
- − does not include control plots for some detectors (TPCs?)
- − needs some modifications when data structure changes (new DAQ)
- − written in C with support for HBOOK/PAW, plots has to be converted to ROOT format

? May be we shall stick to DSPACK related version also for online-QA
- + no need in two independent conversions of RAW data into the ready-to-use format
- + will almost merge with offline-QA and will need less manpower to support
- − must be slower, but the cost is CPU which is in the end cheaper compared to bug hunting
- − it can be, that some important low-level information is not transmitted into DSPACK (?)