DATA ANALYSIS

EGAN Workshop, Liverpool, December 5-9, 2011

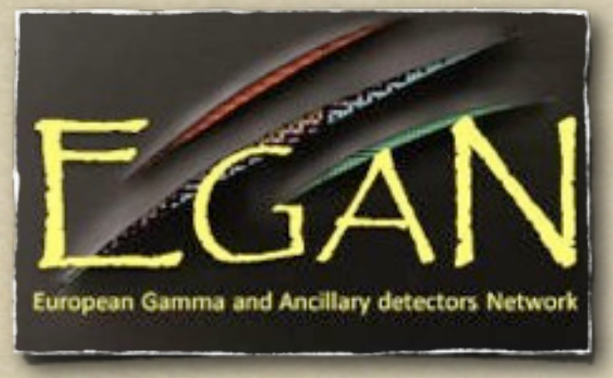
O. Stézowski

DATA ANALYSIS

EGAN Workshop, Liverpool, December 5-9, 2011

+ EGAN Workshop, GSI, December 3-7, 2012

O. Stézowski

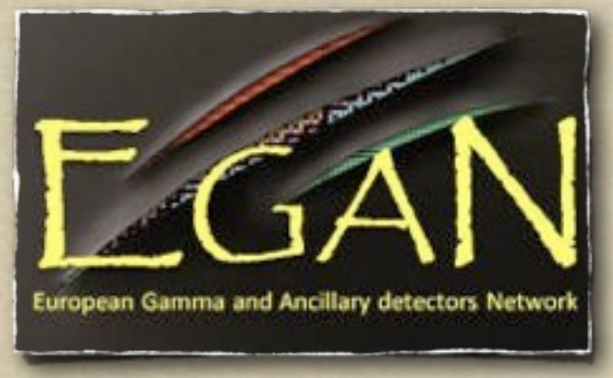


DATA ANALYSIS

EGAN Workshop, Liverpool, December 5-9, 2011

+ EGAN Workshop, GSI, December 3-7, 2012

O. Stézowski



DEBUGGING DATA ANALYSIS

EGAN Workshop, Liverpool, December 5-9, 2011

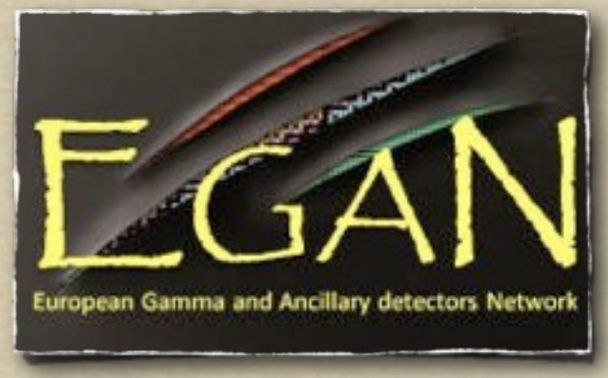
+ EGAN Workshop, GSI, December 3-7, 2012

P. Stézowski

beginning GSI

➠

Legnaro



DEBUGGING DATA ANALYSIS

EGAN Workshop, Liverpool, December 5-9, 2011

+ EGAN Workshop, GSI, December 3-7, 2012

O. Stézowski

Goals

- *Get a general (clear) view of what means AGATA data analysis / data processing*
- *Be able to play with data*
- *Be able to become an actor*

Overview

Basic elements

What already exists

How to extend

Overview

Complexity

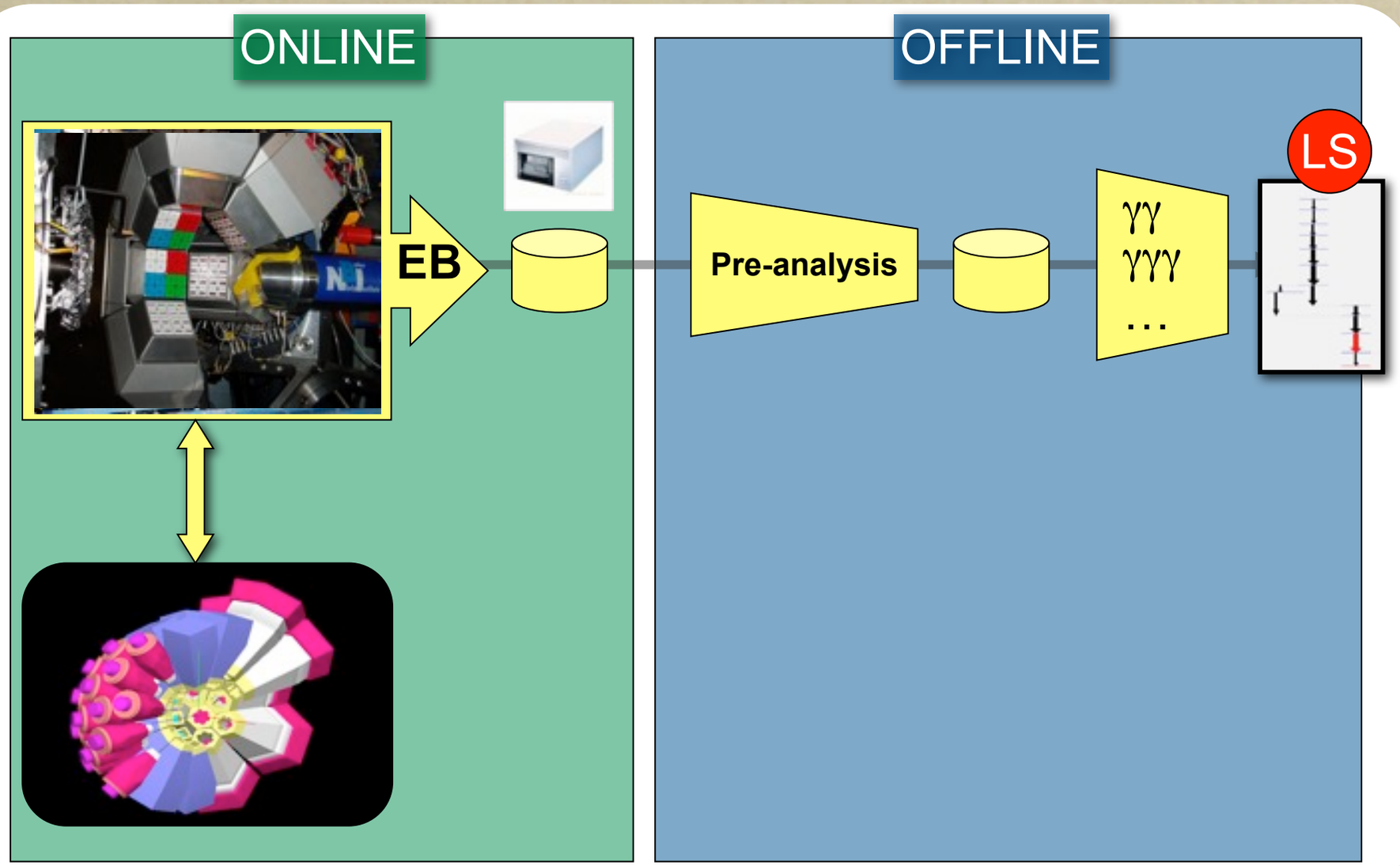
Basic elements

Narval philosophy

Organisation

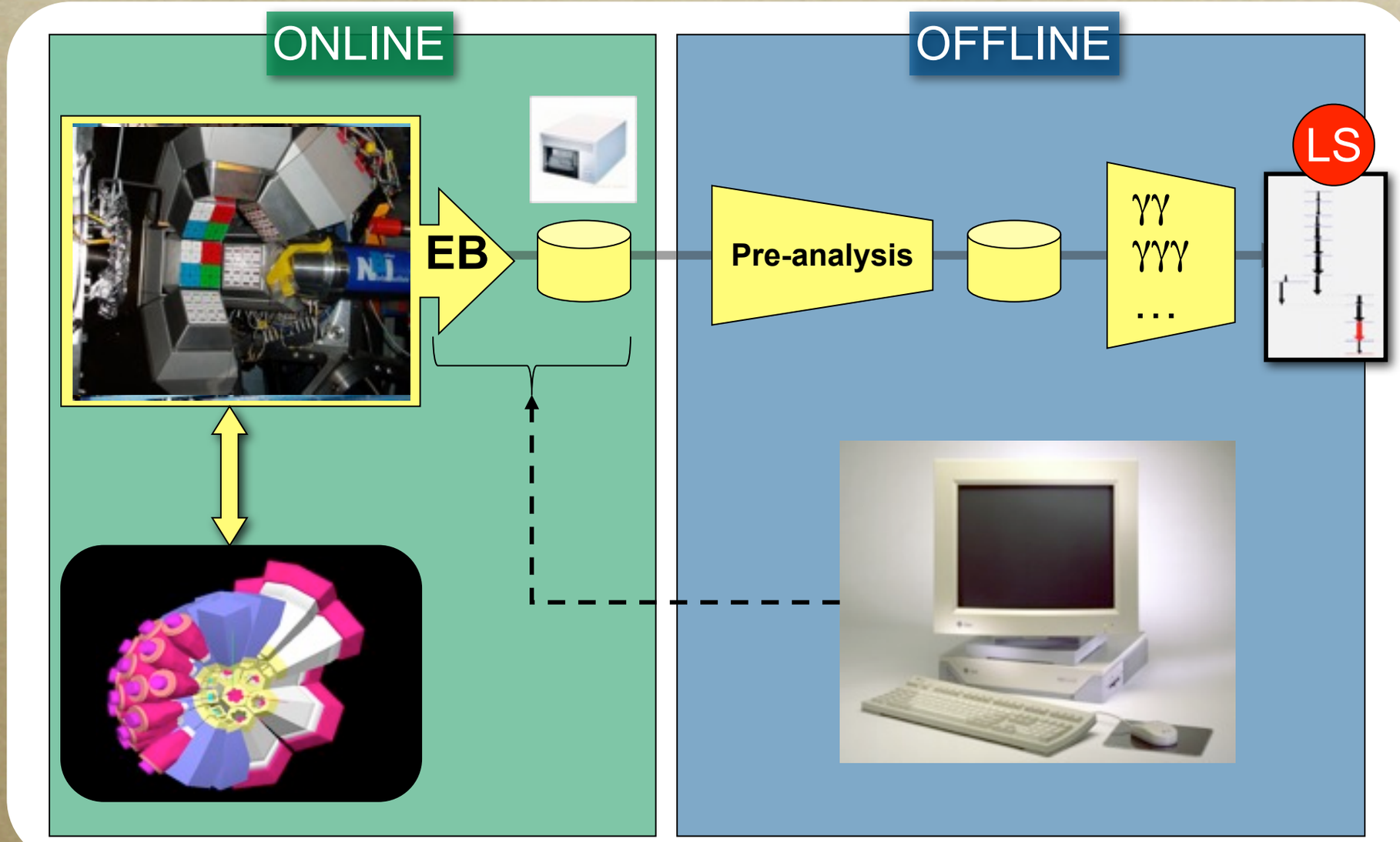
Complex environment

From …



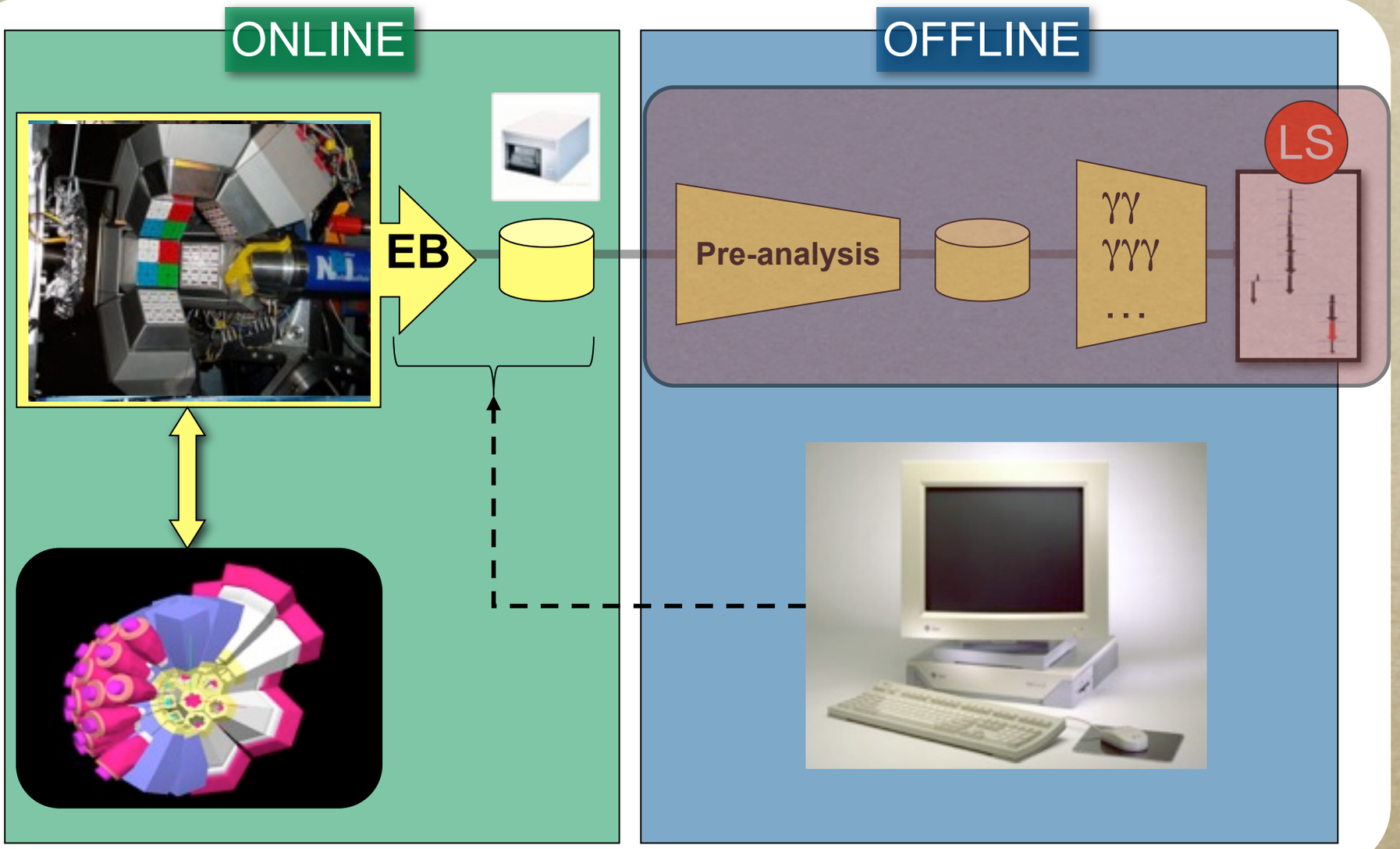
Complex environment

From …

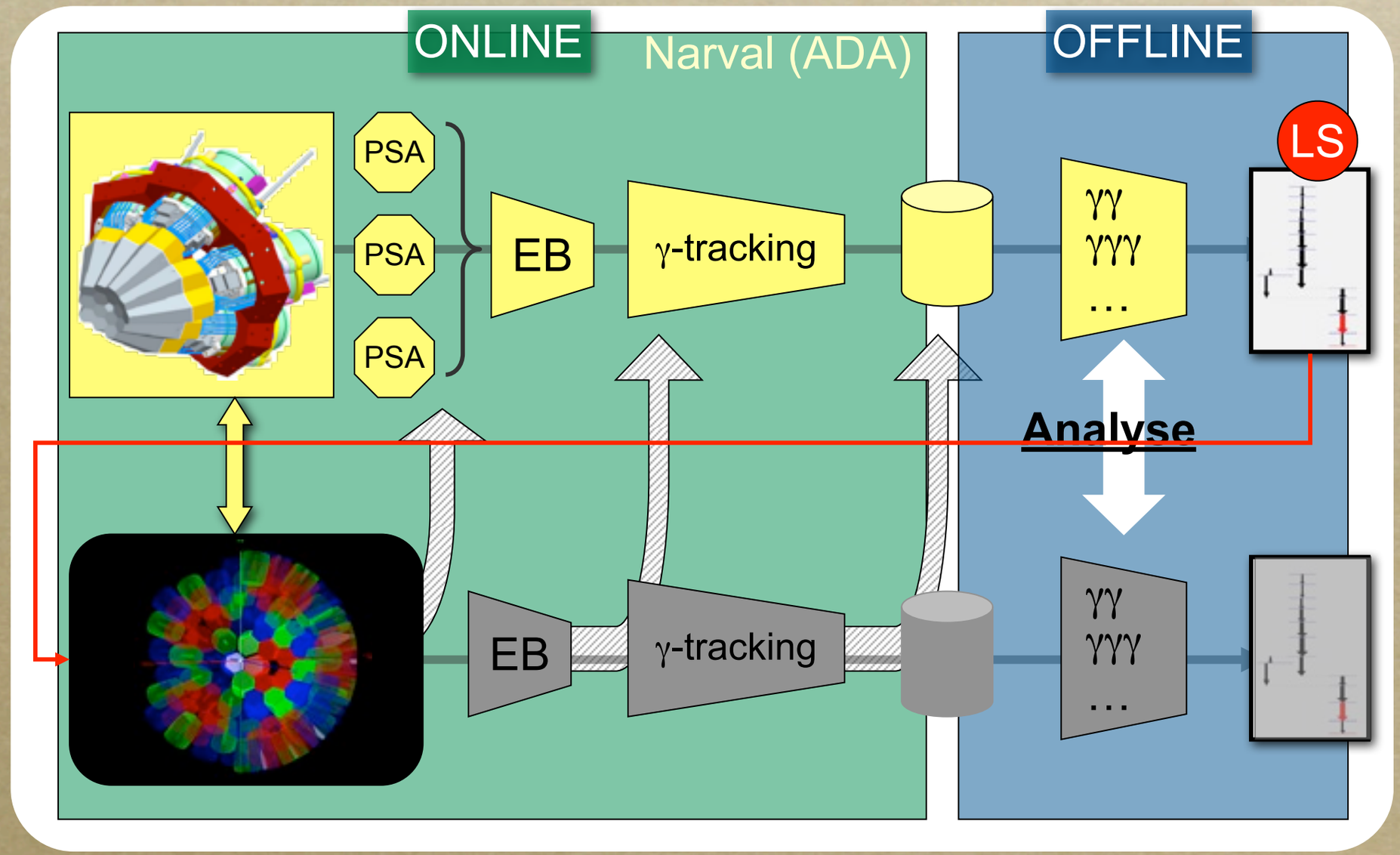


Complex environment

From …

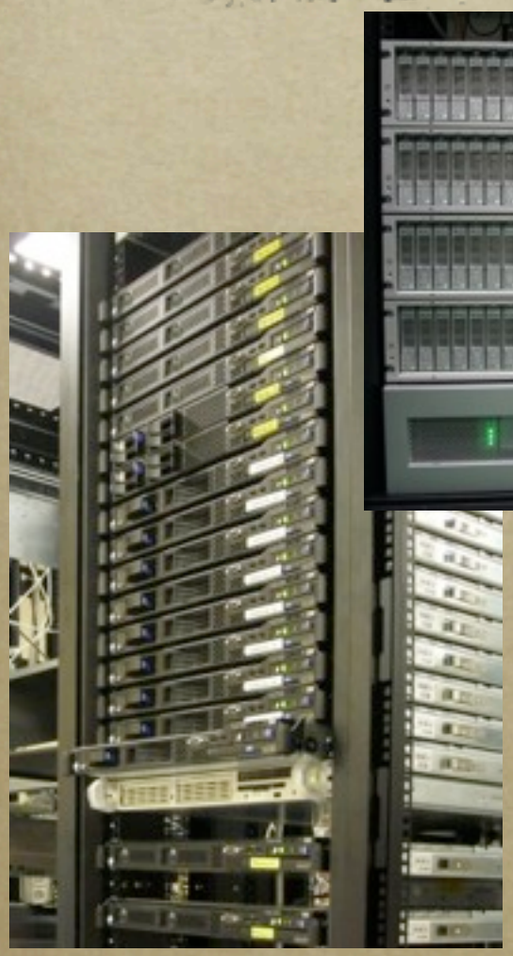


Complex environment

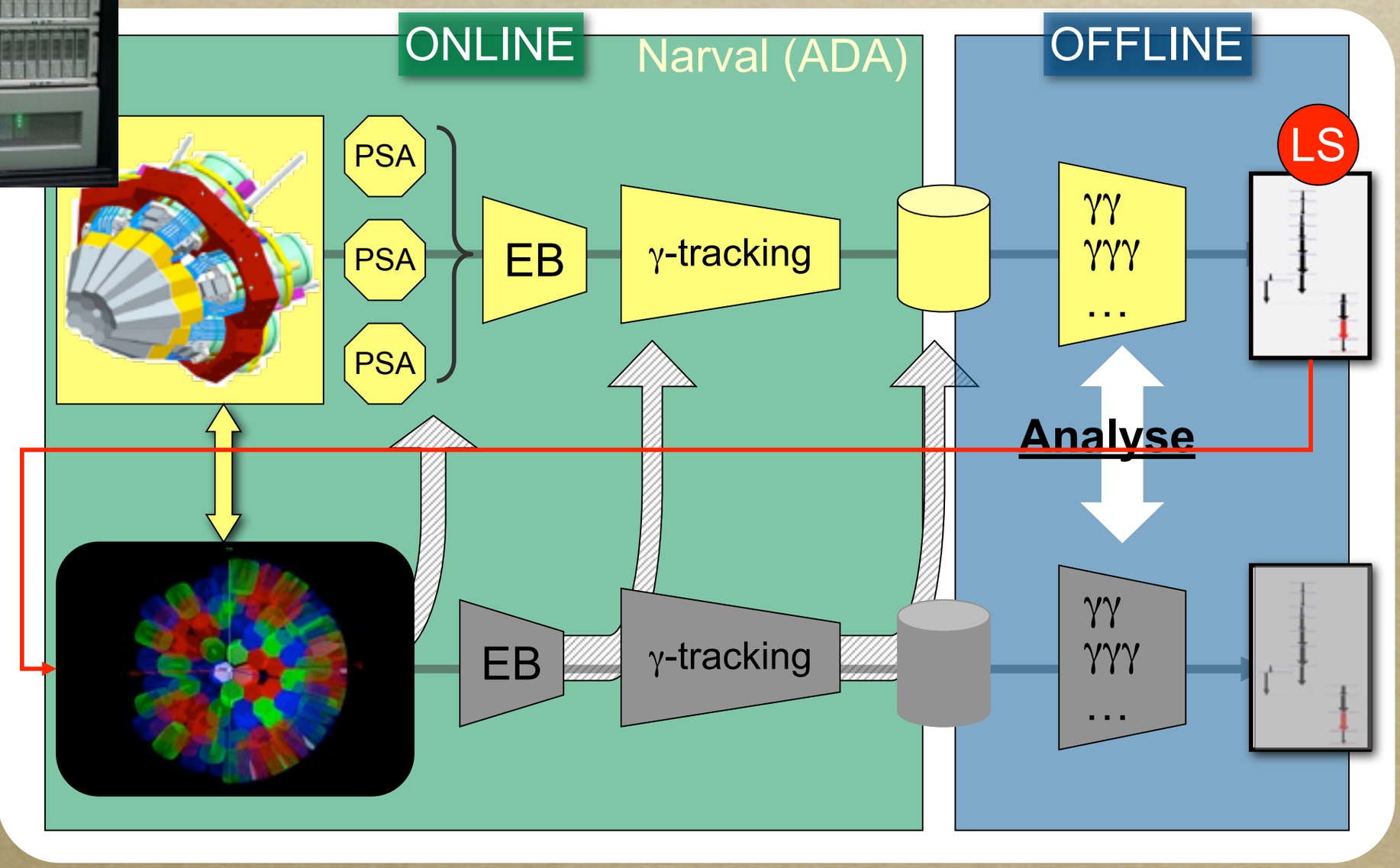


to …

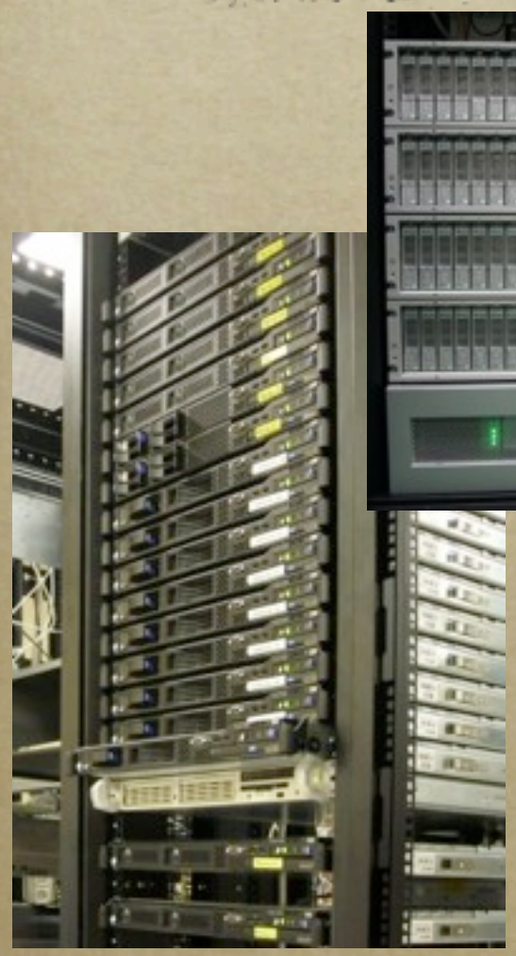
Complex environment

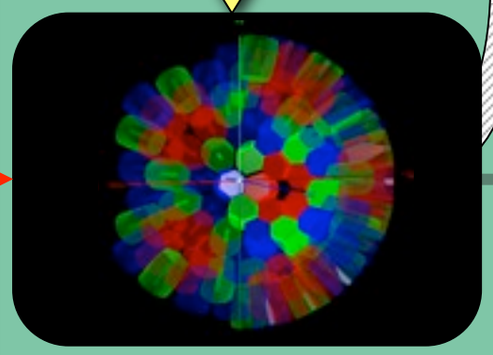


to ...



Complex environment





to …

Complex environment

- *many 'builders' involved in software*

  (building software packages is like building a bridge)

- *need of coordinations*

  shared files, doc (user's guide, code), tutorial, bugtracker, elog ...

  [model : G4, ROOT]

- *different roles :*

  users : use the existing bricks

  developers : create new bricks

- *several packages to be assembled*

  interfaces between different parts

  dependencies (compatibilities between the different parts)

Complex environment

Narval

ROOT

femul

Watchers

ADF

Emulator

AGAPRO

Tracking

GammaWare

PSA

PRESPEC

Grid

PRISMA

GO4

How they are linked, dependant, used ???

Overview

Complexity

Basic elements

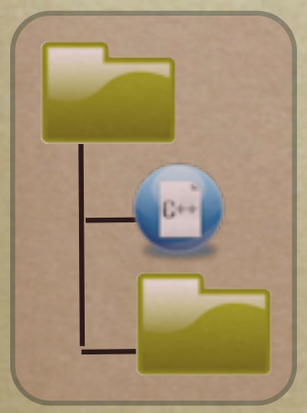
Narval philosophy

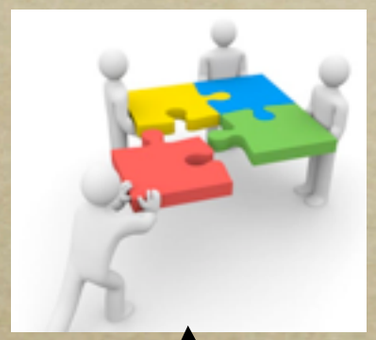
Organisation

Work with svn*

**subversion*

Repository, on a server





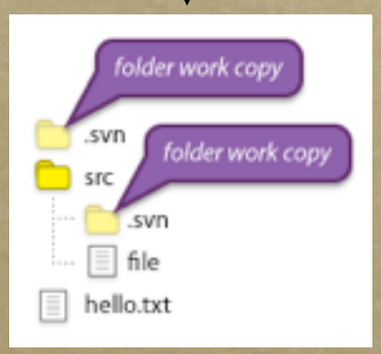
can get any version, any time !

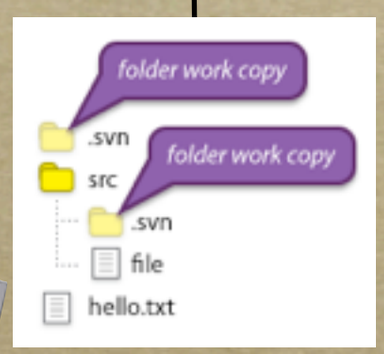
main develoment line : trunk

0

n

freezed : tag, release





....

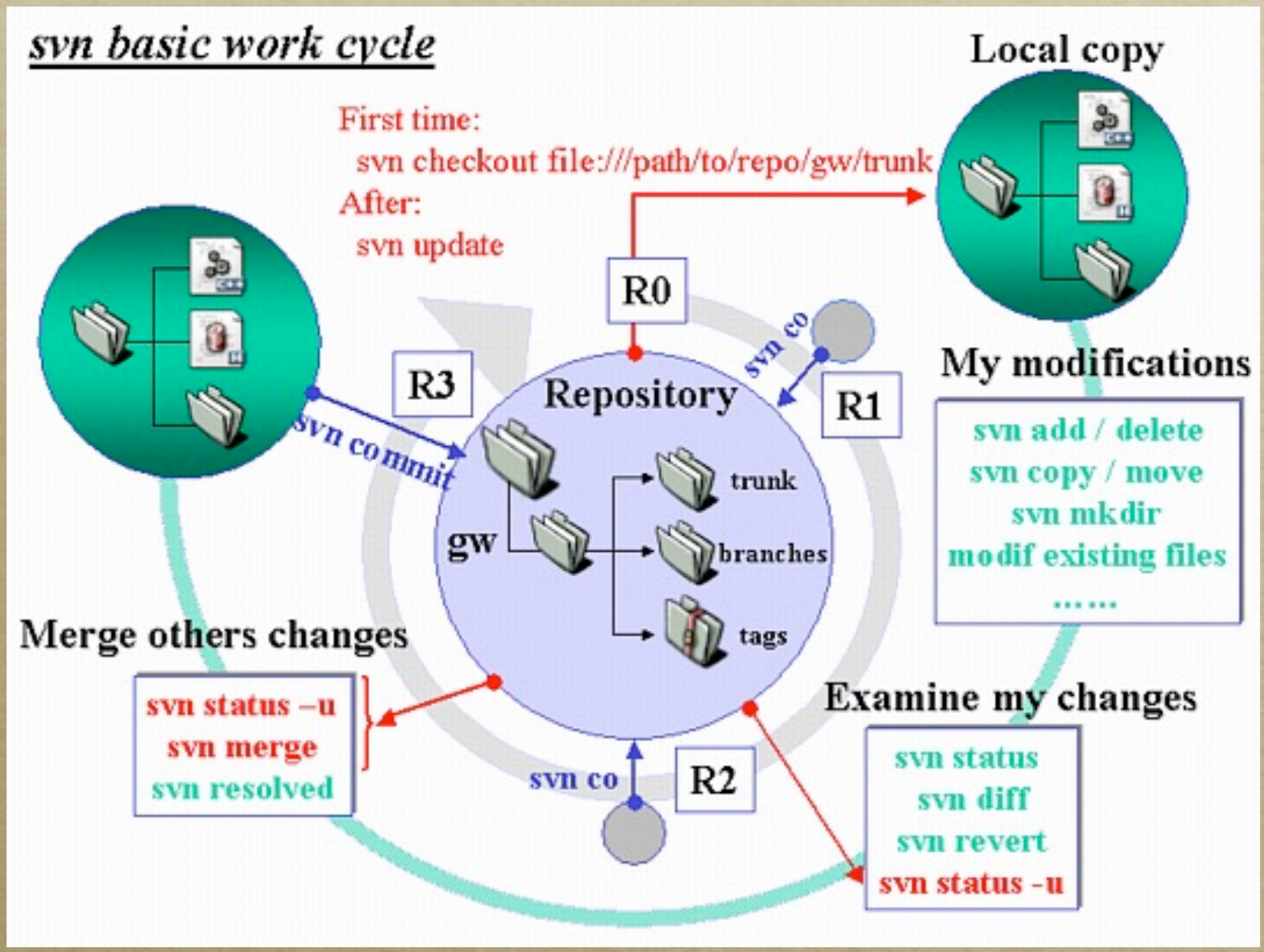
....

work on a local copy

synchronize the local copy with the repository

Work with svn*

**subversion*



svn cycle with the main commands

Work with svn

[svn] / gammaware / trunk / src / adf / CrystalFrame.h Repository: svn Go

ViewVC

Log of /gammaware/trunk/src/adf/CrystalFrame.h

Using XCode

Compare 1163 vs. Local — CrystalFrame.h

Debug | root5.24D_dev | x86_64

Overview Tasks Ungrouped Project

Compare 1163 vs. Local — CrystalFrame.h

<No selected symbol>

```
   //! to get individual segment
   /*!
   */
   virtual GeSegment  *GetSegment(UShort_t) = 0;

   //! to get each core
   /*!
   */
   virtual GeCore     *GetCore(UShort_t) = 0;
};

//! General interface for a CrystalFrame
/*!
   A crystal frame gives some global data, 36 segments signals and 2
*/
class ACrystalFrame : public AgataDataFrame< CrystalInterface >
{
protected:
   ACrystalFrame(const Key *akey):
      AgataDataFrame< CrystalInterface >(akey) {;}
public:
   virtual ~ACrystalFrame()
      {;}
};

// typedef ProxyDataFrame<ACrystalFrame,CrystalInterface> CrystalFra

} // namespace ADF
#endif
```

```
idual segment
   */
   virtual GeSegment  *GetSegment(UShort_t) = 0;

   //! to get each core
   /*!
   */
   virtual GeCore     *GetCore(UShort_t) = 0;
};

//! General interface for a CrystalFrame
/*!
   A crystal frame gives some global data, 36 segments signals and 2
*/
class ACrystalFrame : public AgataDataFrame< CrystalInterface >
{
protected:
   ACrystalFrame(Key *akey):
      AgataDataFrame< CrystalInterface >(akey) {;}
public:
   virtual ~ACrystalFrame()
      {;}
};

// typedef ProxyDataFrame<ACrystalFrame,CrystalInterface> CrystalFra

} // namespace ADF
#endif
```

Work with svn

[svn] / gammaware / trunk / src / adf / CrystalFrame.h Repository: svn Go

Diff of /gammaware/trunk/src/adf/CrystalFrame.h

Parent Directory | Revision Log | Patch

#	revision 1163, Tue Feb 9 09:08:56 2010 UTC Line 14	revision 1635, Fri Mar 18 12:56:46 2011 UTC Line 14
14	* GNU General Public License for more details. *	* GNU General Public License for more details. *
15	* *	* *
16	* You should have received a copy of the GNU General Public License *	* You should have received a copy of the GNU General Public License *
17	* aLong_t with this program; if not, write to the *	* along with this program; if not, write to the *
18	* Free Software Foundation, Inc., *	* Free Software Foundation, Inc., *
19	* 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. *	* 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. *
20	********/	********/

#	Line 137	Line 137
137	class ACrystalFrame : public AgataDataFrame< CrystalInterface >	class ACrystalFrame : public AgataDataFrame< CrystalInterface >
138	{	{
139	protected:	protected:
140	ACrystalFrame(const Key *akey):	ACrystalFrame(Key *akey):
141	AgataDataFrame< CrystalInterface >(akey) {;}	AgataDataFrame< CrystalInterface >(akey) {;}
142	public:	public:
143	virtual ~ACrystalFrame()	virtual ~ACrystalFrame()

Colored Diff Show

Legend:
Removed from v.1163
changed lines
Added in v.1635

Using ViewVC

IN2P3 SVN/CVS Admins. ViewVC Help

Powered by ViewVC 1.0.7

Work with svn

❶ *Release versions: not that used ... so far (GSI ?)*

lack of policy (root, any 6 months with objectives)
at least be aware of the version (release) you are using

❷ *Mercurial (Hg) also used @ GSI*

(different but same philosophy)

❸ *Good practice in case of problems/bugs, give :*

the distribution (which linux, mac), environment
version of the codes, output etc ...

☛ *not only : it does not work ...*

Some 'svn' servers

AGAPRO (=narval emulator)

svn://gamma01.lnl.infn.it/agata
svn://gamma01.lnl.infn.it/agata/tags/GSI/121001_rudolph/

tag version ...
... real start for tag !?

GammaWare

svn+ssh://anonsvn@anonsvn.in2p3.fr/agata/gammaware/trunk

➥ *ADF only*

svn+ssh://anonsvn@anonsvn.in2p3.fr/agata/gammaware/trunk/src/adf

➥ *watchers directly*

svn+ssh://anonsvn@anonsvn.in2p3.fr/agata/gammaware/trunk/demos/adf

Prespec

ssh://agatagsi@lxi038.gsi.de:22//u/agatagsi/new_prespec_Go4

Other useful sites

DAQ team

https://forge.in2p3.fr/projects/agata
https://forge.in2p3.fr/projects/narval

γware and co

http://agata.in2p3.fr/
https://svn.in2p3.fr/agata/docs/trunk/

Gw User's guide + cookbook !

html doc of the code

http://www.ipnl.in2p3.fr/gammaware/doc/html/

ELOG : http://agata-0.lnl.infn.it:8989/ ☛ http://lxagata0.gsi.de:8989/

Fill it, important !!!

http://root.cern.ch/drupal/

https://savannah.cern.ch/bugs/?func=additem&group=savroot

Other useful sites

GammaWare Head Version for release 0.9

Search

Gw::GCondHandler
Gw::GCondition
Gw::GeantLMOF
Gw::GEM
Gw::GLSPlayer
Gw::GSPlayer
Gw::H1Calibrator
Gw::HistoConverter
Gw::HistoDB
Gw::InfoData
Gw::Level
Gw::LevelEditor
Gw::LevelScheme
Gw::Link
Gw::LogCollector
Gw::LogMessage
Gw::LoopControl
Gw::LoopOnTasks
Gw::LSaxis
Gw::MatPlayer
Gw::Measure< Data_T >
Gw::NuclearLevel
Gw::OStreamCollector
Gw::PadManager

AutoSave
DoCanvas
Exec
GetLastSaveArg1
GetLastSaveArg2
GetLastSaveArg3
GetLastSaveArg4
LoopControl
Save
SetAutoTime
Zero
~LoopControl

```
void LoopControl::Save ( const Char_t *  main,
                         const Char_t *  tag,
                         const Char_t *  option,
                         Int_t           autotime
                       )
```

Save the spectra in files.

- main_file is the name of the root file to save spectra
- tag_file is the name of the root file to save tagged spectra
- option has the following format :
  - ; means save spactra in the current TDirectory
  - (means open a new file
  -) means close the file
  - ch means works with a chain i.e. sequence of names with the same pattern.
  - option to Zero spectra is given using {opt1:opt2}.
- autotime : call automatically this command again after autotime seconds. Off if set to 0.

Here are some examples.

- if you have open your watcher is a root directory and would like just to save there the current histograms : "" "0" ";" 0
- to save in the current root dir, open a new one and move all hitograms there : "my_new.root" "0" ";(" 0
- to dump histograms in a root file (open, save and close the file) : "my_dump" "0" "(;)" 0
- to dump histograms in a chain of root files (open, save and close the file) : "my_dump" "0" "ch(;)" 0
  it looks for existing my_dumpXXXX.root files in the current dir XXXX = [0, ...[. It takes the first non existing one.
- to open a new files that becomes the mother file of all watchers, and save histo there : "my_dump" "0" "(;" 0
- to set spectra of the pool to zero : "my_dump" "0" "(; {pool}" 0
- to set all spectra to zero and change the binning : "my_dump" "0" "(; {pool:100 0 100}" 0

Definition at line 1459 of file Watchers.cpp.

Fill it, important !!!

http://root.cern.ch/drupal/

https://savannah.cern.ch/bugs/?func=additem&group=savroot

Other useful sites

DAQ team

https://forge.in2p3.fr/projects/agata
https://forge.in2p3.fr/projects/narval

γware and co

http://agata.in2p3.fr/
https://svn.in2p3.fr/agata/docs/trunk/

Gw User's guide + cookbook !

html doc of the code

http://www.ipnl.in2p3.fr/gammaware/doc/html/

ELOG : http://agata-0.lnl.infn.it:8989/ ☛ http://lxagata0.gsi.de:8989/

Fill it, important !!!

http://root.cern.ch/drupal/

https://savannah.cern.ch/bugs/?func=additem&group=savroot

installation of the code

[system/user]

required root privileges

apt-get, rpm, dmg etc ...

boost

skstream

Grid, never admin

(except if we are reached to buy a virtual box on each site)

sometime better to install on the user side

For other AGATA packages, nothing like this

configure
make
make install

[user]

- *narval emulator : make, make install*
- *PRISMA : make, make install*
- *Gammaware : configure, make, make install*
- *ake install*
- *ake install*

python script
AgataLegnaro.py
to help
loading, compiling, installing
(see practical sessions)

Should be used widely in the future

installation of the code

[system/user]

required root privileges

apt-get, rpm, dmg etc ...

boost

skstream

Grid, never admin

(except if we are reached to buy a virtual box on each site)

sometime better to install on the user side

For other AGATA packages, nothing like this

configure

NEW :

- *same tool to configure / compile / install : cmake*
- *a standard Geant4, ROOT*
- *still python script [AgataSoftware.py] to help*

AgataSoftware.py

Future :

conf

clean

Usage : python AgataSoftware.py --options=opt actions

keywords for the list of actions :
- ** load*
- ** compile*
- ** install*
- ** all = load + compile + install*

see practical session

possible options are [be careful, the order does matter !] :
- ** -h : print help*
- ** --rev : for svn servers, in case trunk is required, download a specific revision*
- ** --adf : follow by = (default, means trunk) or by a valid path to a tag version in the repository*
- ** --gw : follow by = (default, means trunk) or by a valid path to a tag version in the repository*
- ** --skstream : default (0.3.7) or valid version number in the file skstream-V.tar.gz on the http server*
- ** --agapro : follow by = (default, means trunk) or by a valid path to a tag version in the repository*

ex :

```
python AgataSoftware.py -h
python AgataSoftware.py --adf= --skstream= --boost=tags/release/Boost_1_46_0/ --agapro= load
python AgataSoftware.py --agapro= all
```

List of entry points for the known modules :
- *+ gw ==> svn+ssh://anonsvn@anonsvn.in2p3.fr/agata/gammaware/*
- *+ skstream ==> http://prdownloads.sourceforge.net/worldforge/*
- *+ agapro ==> svn://gamma01.lnl.infn.it/agata/*
- *+ adf ==> svn+ssh://anonsvn@anonsvn.in2p3.fr/agata/gammaware/*
- *+ oft ==> http://csngwinfo.in2p3.fr:2401/oft/*
- *+ boost ==> http://svn.boost.org/svn/boost/*

****** in case of questions/problems, agata{_{at}_}ipnl.in2p3.fr ******

Overview

Complexity

Basic elements

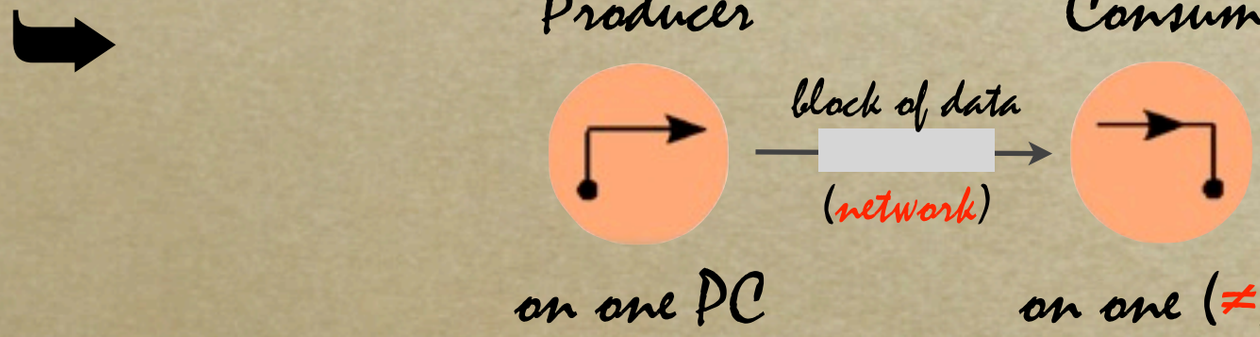
Narval philosophy

Organisation

Narval philosophy

Decomposition of processing in chain of actions

Ex - simple analysis : read data AND create spectra



Advantages { *if source of data changes, just change the producer*
the same data flow could be redirected to different consumers

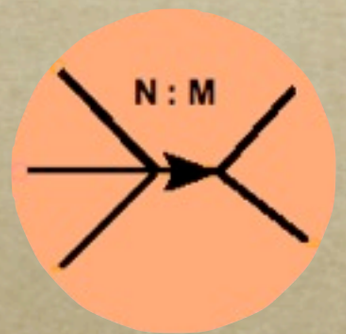
Main actors

- Producer (produced data) — *0 input, 1 output*
- Consumer (consumer data) — *1 input, 0 output*
- Filter (consume data, apply algo, produce new data) — *1 input, 1 output*

algorithms ! (tracking, PSA)

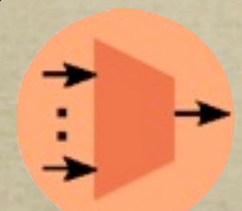
Narval philosophy

♣ *More complex actors : organisation of the data flow*



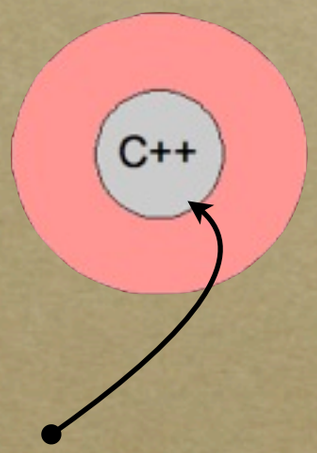
n inputs, m outputs

⇨ *Event Builder, Merger i.e. n inputs, 1 output*



♣ *Narval is written in ADA BUT can load C/C++ code*

⇨ *the library has to defined a precise interface(*)*



libActor.so

```
TrackingFilter *process_register (...)
void process_config (...)
void process_block(...)
void process_initialise(...)
void process_reset(...)
void process_start(...)
void process_stop(...)
void process_pause(...)
void process_resume(...)
```

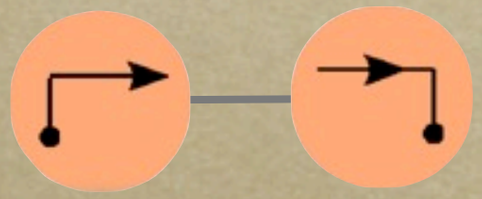
⇨ *only for producer, consumer and filter !*

(*)Agata PSA and Tracking Algorithm Integration, J. Cresswell and X. Grave, 2nd Draft

Narval philosophy

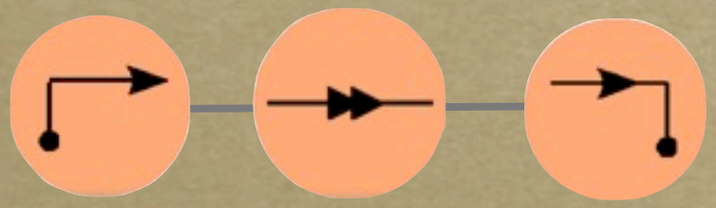
A <u>topology</u> defines how actors are connected.

some examples



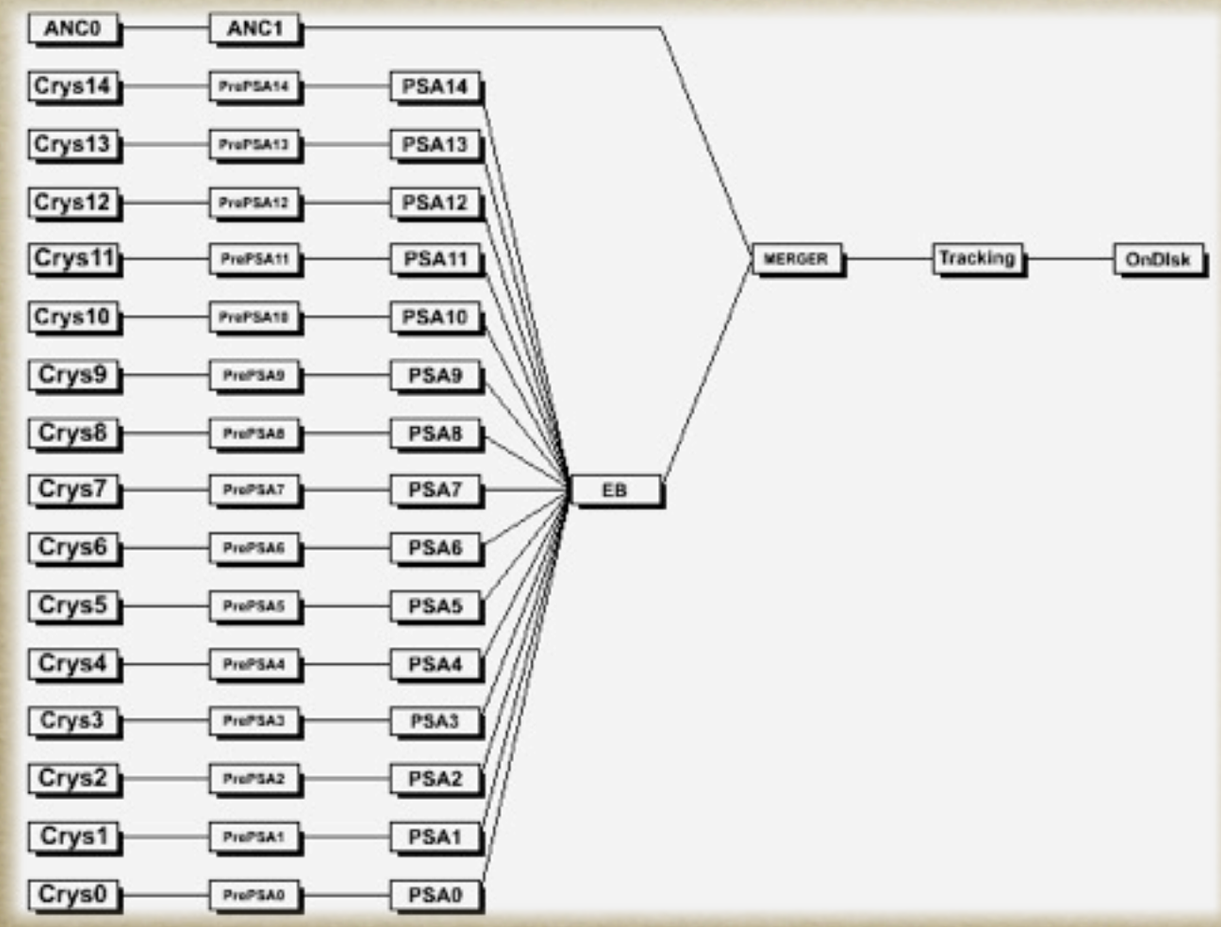
- read data, produce spectra
- read data, produce root tree

...



- read data, apply filter, save
- read data, apply algo, produce spectra

...



almost current online topology

Narval philosophy

Narval can :

- *built complex topology*
- *run actors (ADA,C++) on clusters of PC*

C++ actors can be assembled in C++ environment

- *so-called <u>emulators</u>*

Emulators are useful for :

- *debugging*
- *developing new actor*
- *'light' analysis ... i.e. does not required parallelisation*

You can also install Narval in your institute !

Actors in frameworks

agata producer, filter, consumer

EB, merger

replica

c++

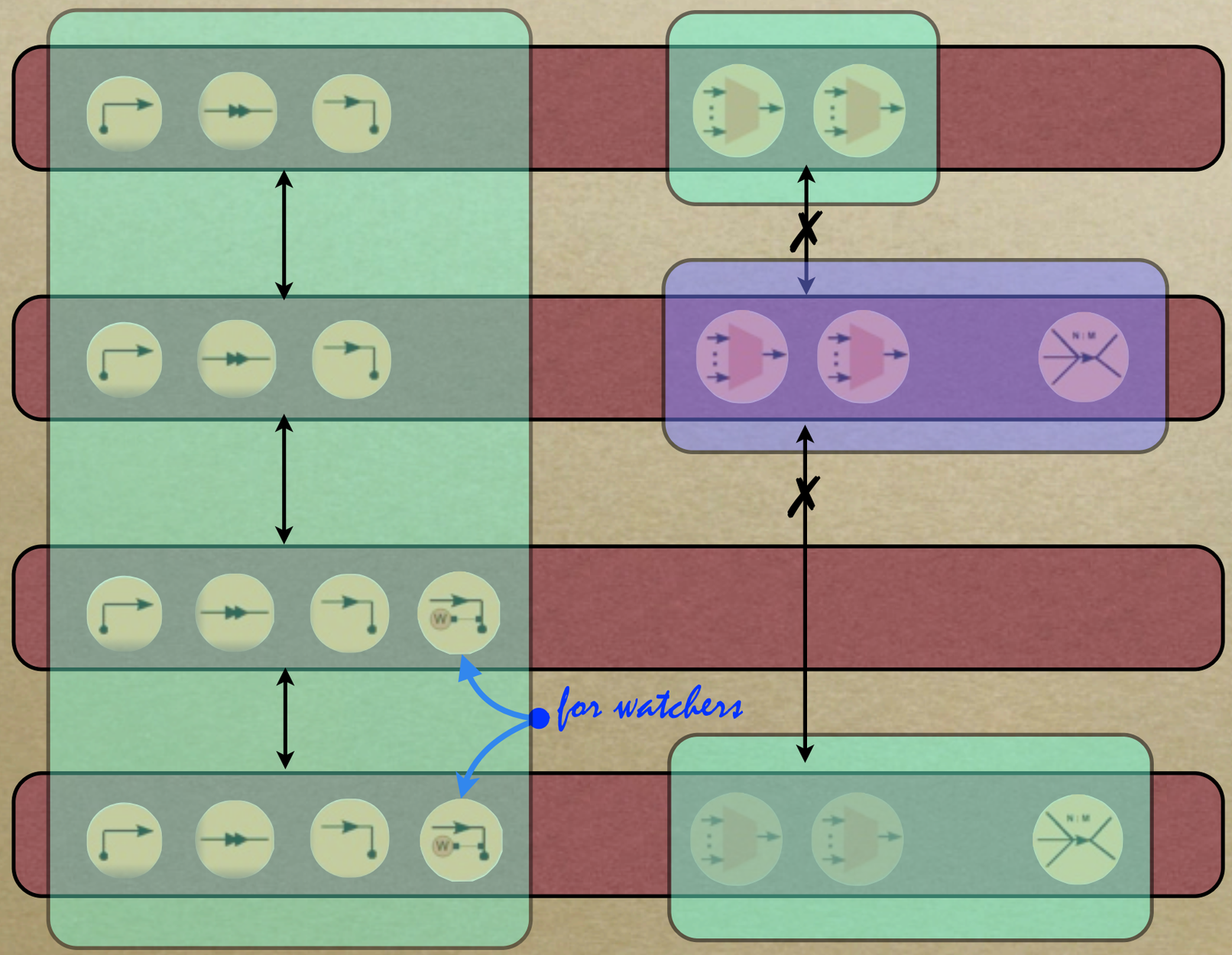
ada

femul
standalone exec

Narval
framework

ROOT[Gw]
framework

ROOT[Gw]
framework



EmulatorPC
EmulatorPFC

DEmulator
RMTEmulator ...
(BoostEmulator)
ProofEmulator ?

Actors in frameworks

agata producer, filter, consumer

EB, merger

replica

c++

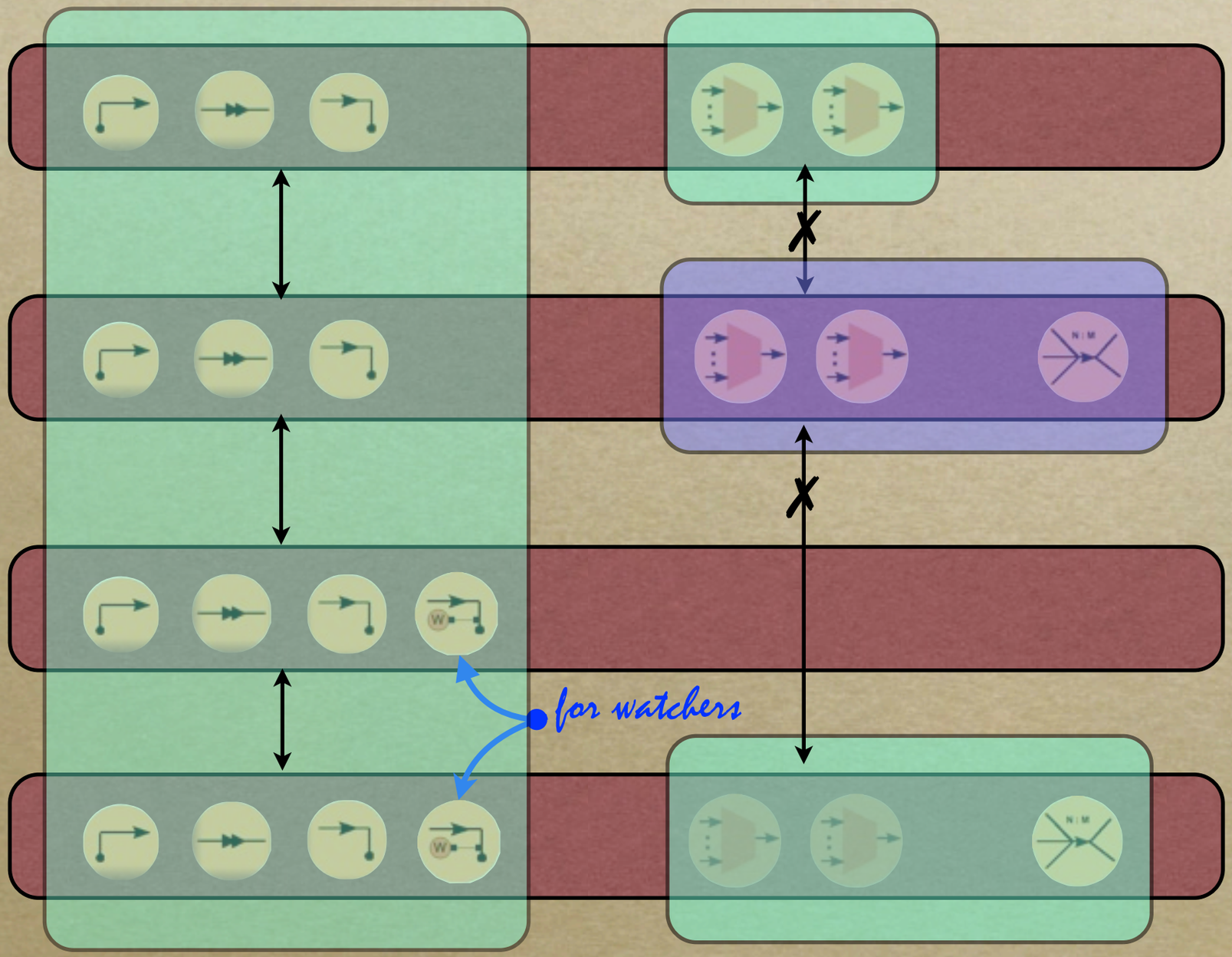
ada

femul
standalone exec

Narval
framework

ROOT[Gw]
framework

ROOT[Gw]
framework



EmulatorPC
EmulatorPFC

DEmulator
RMTEmulator ...
test
(BoostEmulator)
ProofEmulator ?

Actors in frameworks

agata producer, filter, consumer

EB, merger

replica

c++

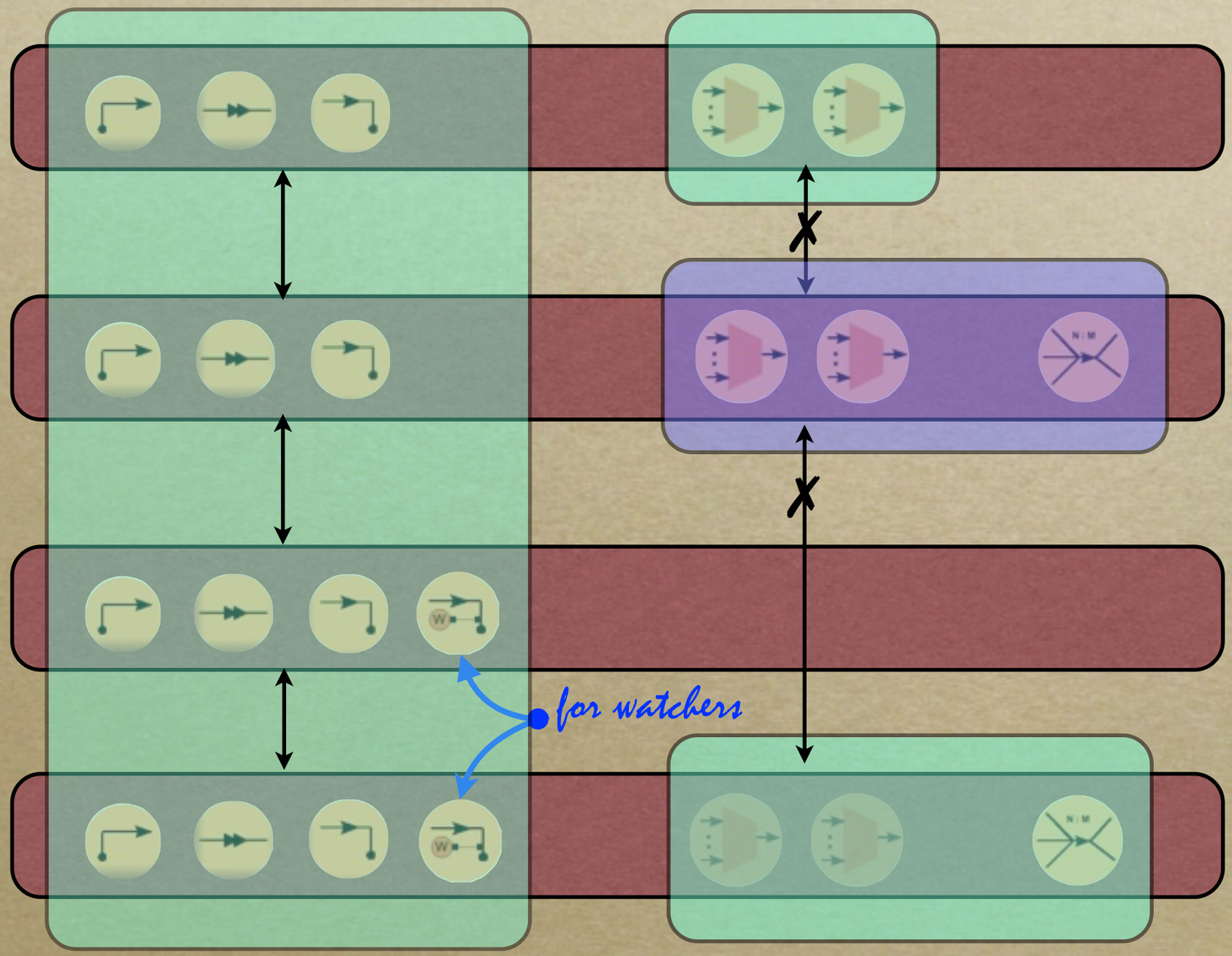
ada

femul
standalone exec

Narval
framework

ROOT[Gw]
framework

ROOT[Gw]
framework



EmulatorPC
EmulatorPFC

DEmulator
RMTEmulator ...
test

(BoostEmulator)
ProofEmulator ?
under dev

Agata shared actors

The svn called 'narval-emulator' contains

agapro

- *the actors shared between the frameworks*
  - ➥ (`BasicAFC, AncillaryProducerTCP, TrackingFilter PSAFilterGridSearch etc ...`)
- *the PRISMA library*
- *femul + femul event builder / merger*

internal

+ a copy of ADF, OFT

***internal** / external building*

- *the actors shared between the frameworks*
  - ➥ `Should be external packages !` (GSI?)
  - ➥ `Otherwise pb of synchronisation`

Cmake (AgataSoftware.py) as unique building system
[agapro, gw, adf, prespec]

Agata shared actors

The svn called 'narval-emulator' contains

agapro

- *the actors shared between the frameworks*
  - ➥ (`BasicAFC, AncillaryProducerTCP, TrackingFilter PSAFilterGridSearch etc ...`)
- *the PRISMA library*
- *femul + femul event builder / merger*

internal

+ a copy of ADF, OFT

***internal** / external building*

- *the actors shared between the frameworks*
  - ➥ `Should be external packages !` (GSI?)
  - ➥ `Otherwise pb of synchronisation`

Cmake (AgataSoftware.py) as unique building system
[agapro, gw, adf, prespec]

Agata shared actors

The svn called 'narval-emulator' contains

agapro

- *the actors shared between the frameworks*
  - ➥ (`BasicAFC, AncillaryProducerTCP, TrackingFilter PSAFilterGridSearch etc ...`)
- *the PRISMA library*
- *femul + femul event builder / merger*

internal

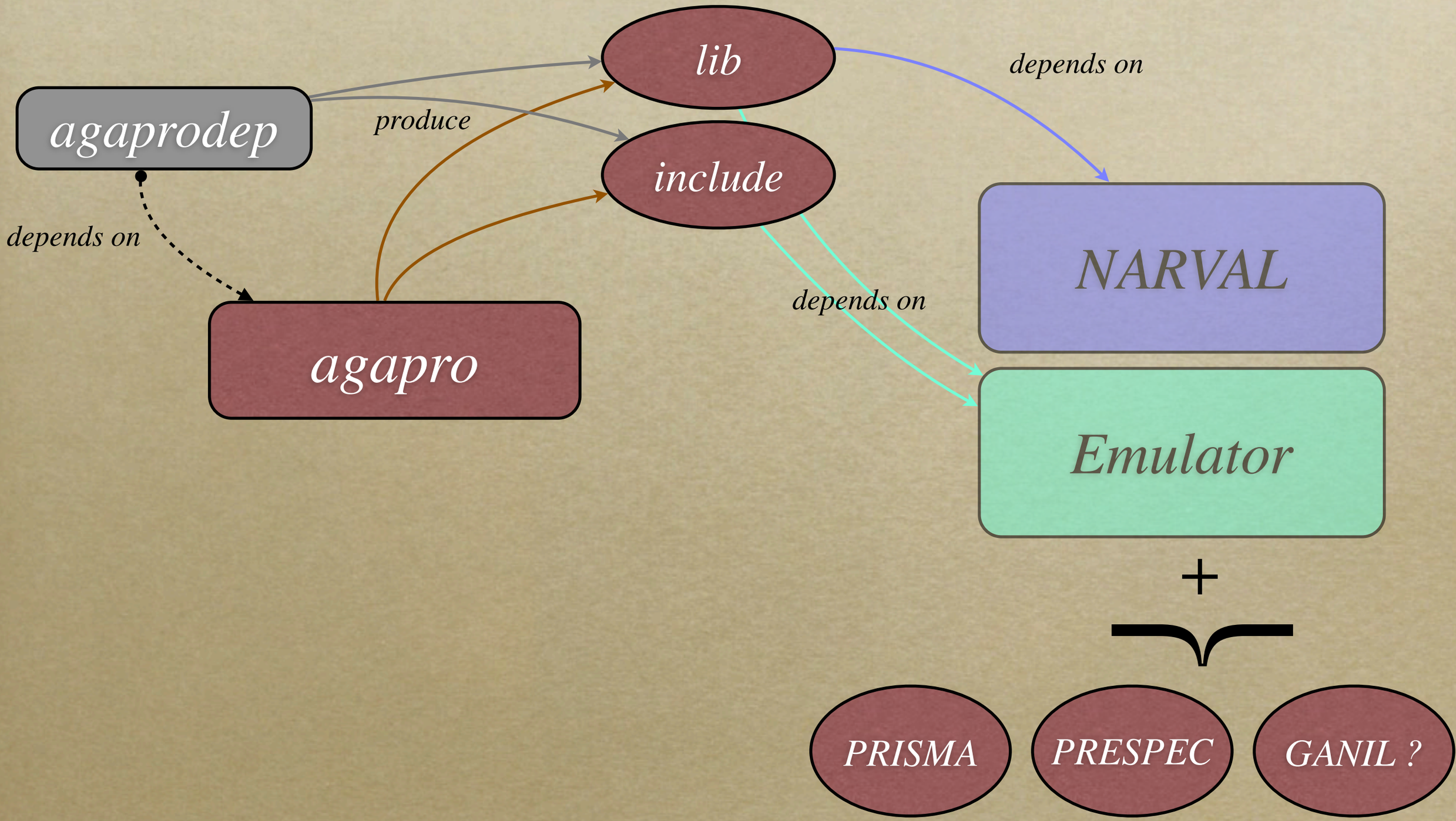
+ a copy of ADF, OFT

internal / external building

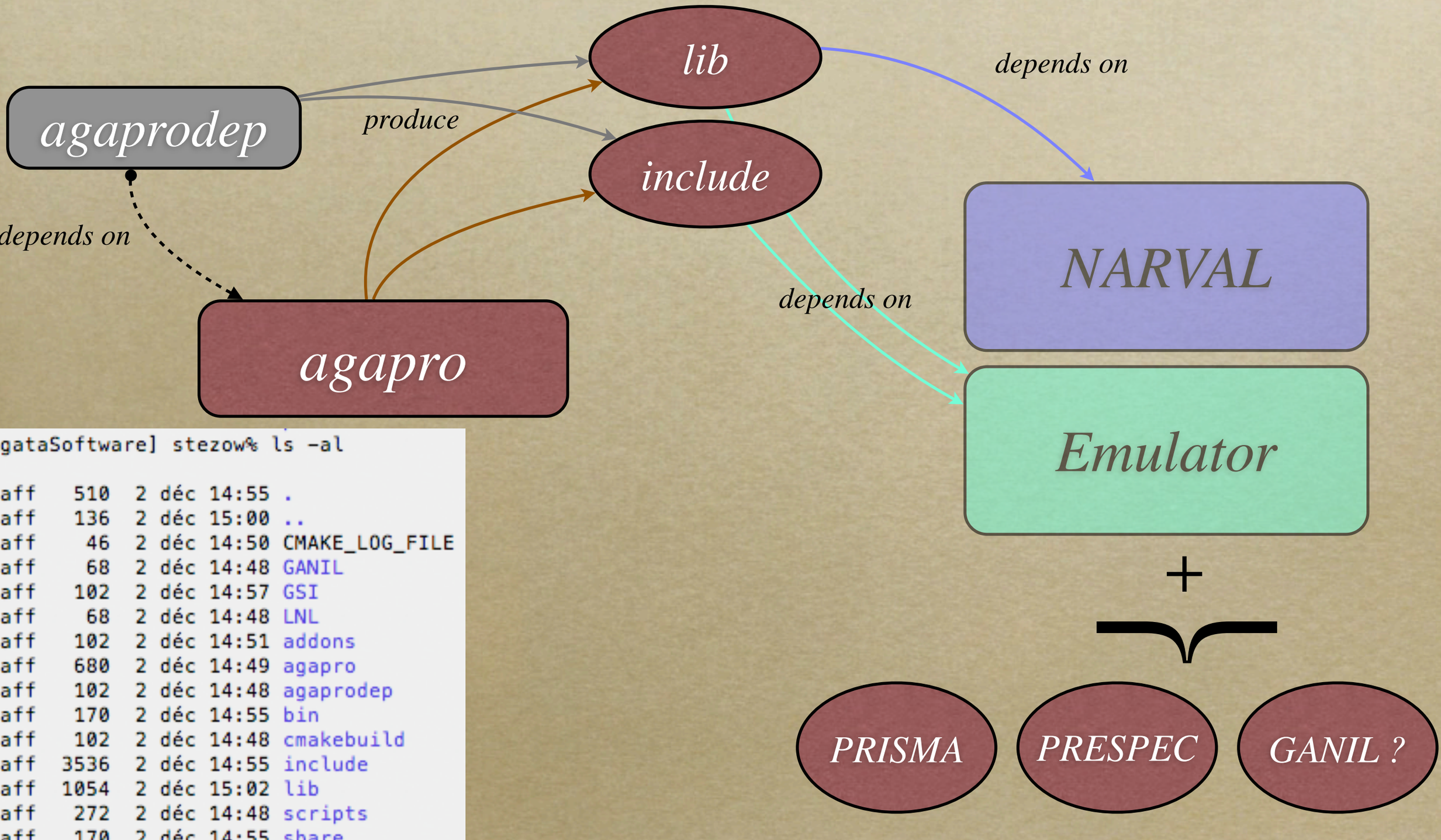
- *the actors shared between the frameworks*
  - ➥ `Should be external packages !` (GSI?)
  - ➥ `Otherwise pb of synchronisation` ✓

Cmake (AgataSoftware.py) as unique building system
[agapro, gw, adf, prespec]

AGATA Shared actors



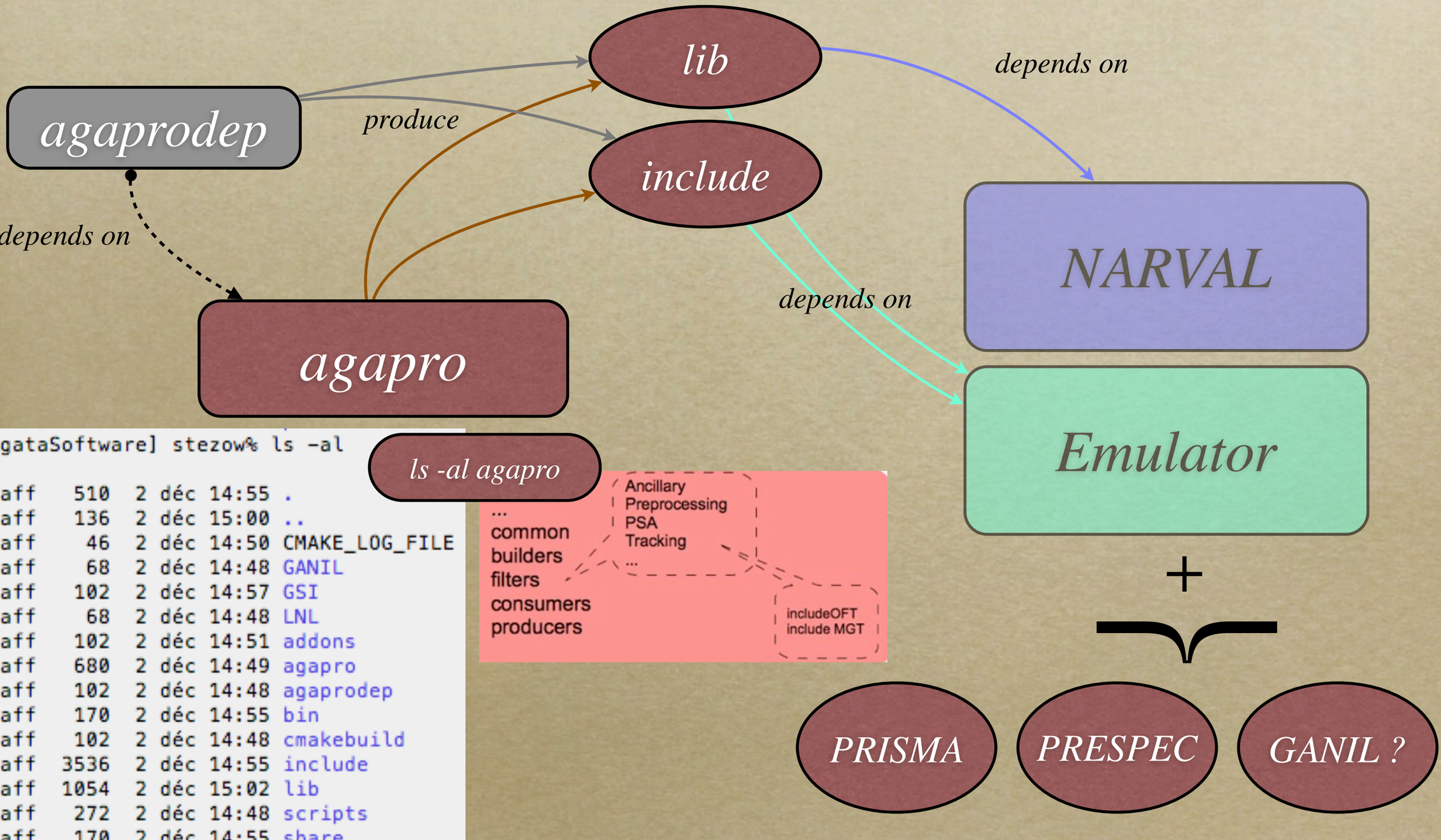
AGATA Shared actors




```
/AgataSoftware] stezow% ls -al

staff    510  2 déc 14:55 .
staff    136  2 déc 15:00 ..
staff     46  2 déc 14:50 CMAKE_LOG_FILE
staff     68  2 déc 14:48 GANIL
staff    102  2 déc 14:57 GSI
staff     68  2 déc 14:48 LNL
staff    102  2 déc 14:51 addons
staff    680  2 déc 14:49 agapro
staff    102  2 déc 14:48 agaprodep
staff    170  2 déc 14:55 bin
staff    102  2 déc 14:48 cmakebuild
staff   3536  2 déc 14:55 include
staff   1054  2 déc 15:02 lib
staff    272  2 déc 14:48 scripts
staff    170  2 déc 14:55 share
```

AGATA Shared actors




```
/AgataSoftware] stezow% ls -al

staff    510  2 déc 14:55 .
staff    136  2 déc 15:00 ..
staff     46  2 déc 14:50 CMAKE_LOG_FILE
staff     68  2 déc 14:48 GANIL
staff    102  2 déc 14:57 GSI
staff     68  2 déc 14:48 LNL
staff    102  2 déc 14:51 addons
staff    680  2 déc 14:49 agapro
staff    102  2 déc 14:48 agaprodep
staff    170  2 déc 14:55 bin
staff    102  2 déc 14:48 cmakebuild
staff   3536  2 déc 14:55 include
staff   1054  2 déc 15:02 lib
staff    272  2 déc 14:48 scripts
staff    170  2 déc 14:55 share
```

ls -al agapro

Complex environment

Narval

ROOT

femul

Watchers

ADF

Emulator

Tracking

AGAPRO

PSA

GammaWare

PRESPEC

Grid

PRISMA

GO4

I hope you have a better view now !

Overview

Basic elements

What already exists

How to extend

Overview

Actor's actions

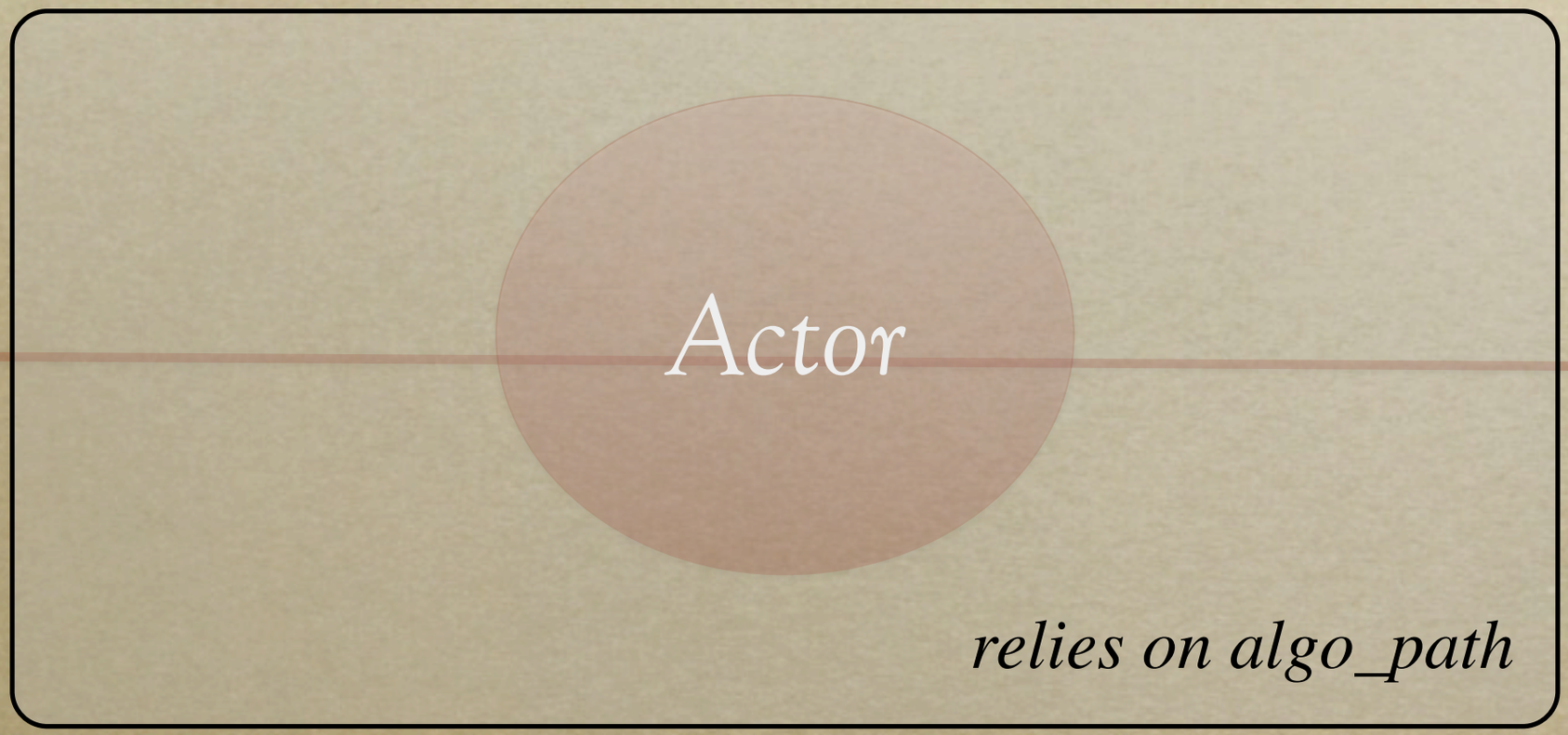
What already exists

Watchers

Data flow structure

Actor's communication

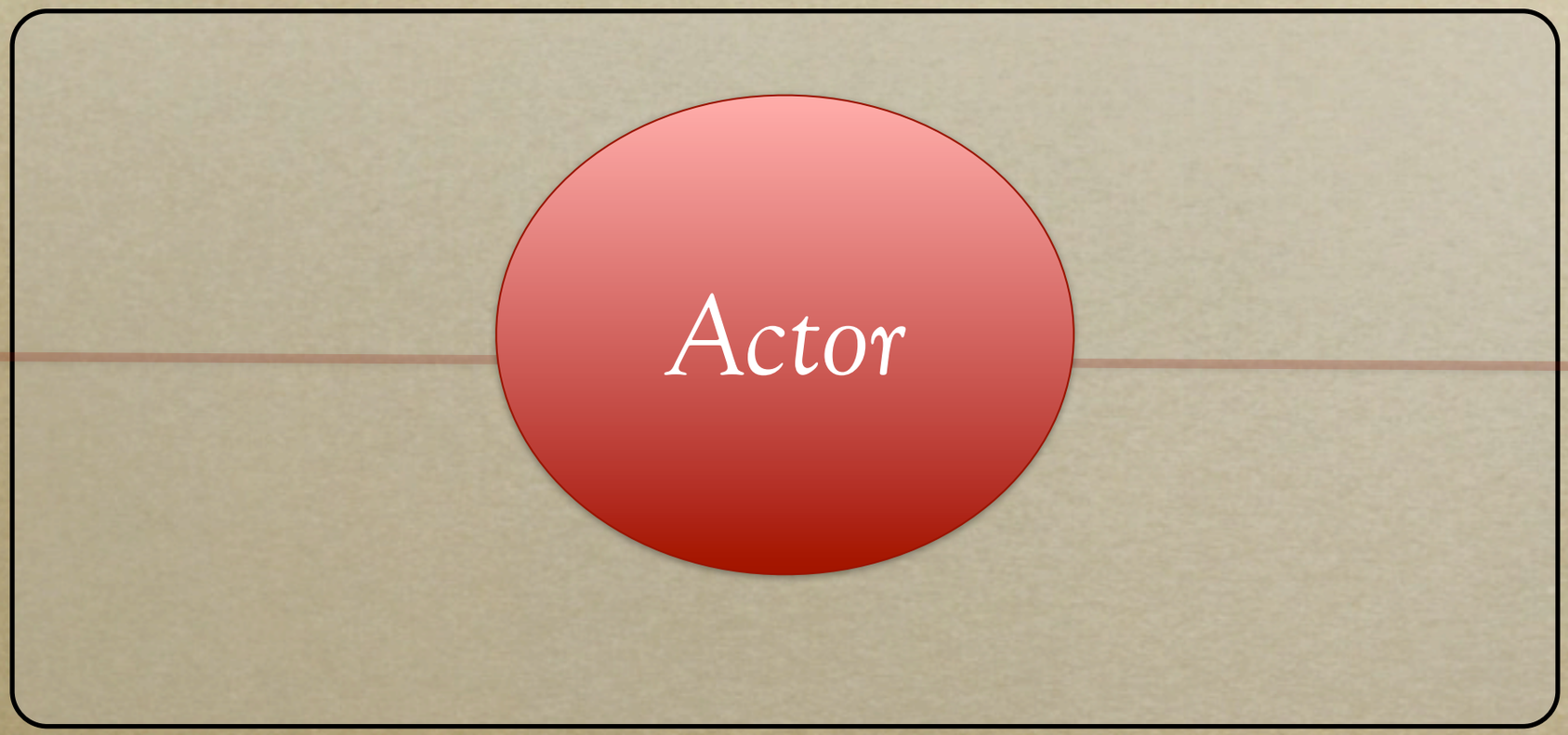
an actor is configured first ...



... means ~ the environment around is set

Actor's communication

an actor is configured first ... then created



Actor's communication

the actor is initialised (just once) using .conf files

python script ***gen_conf.py***
to help configuring all actors

for PSA algorithm

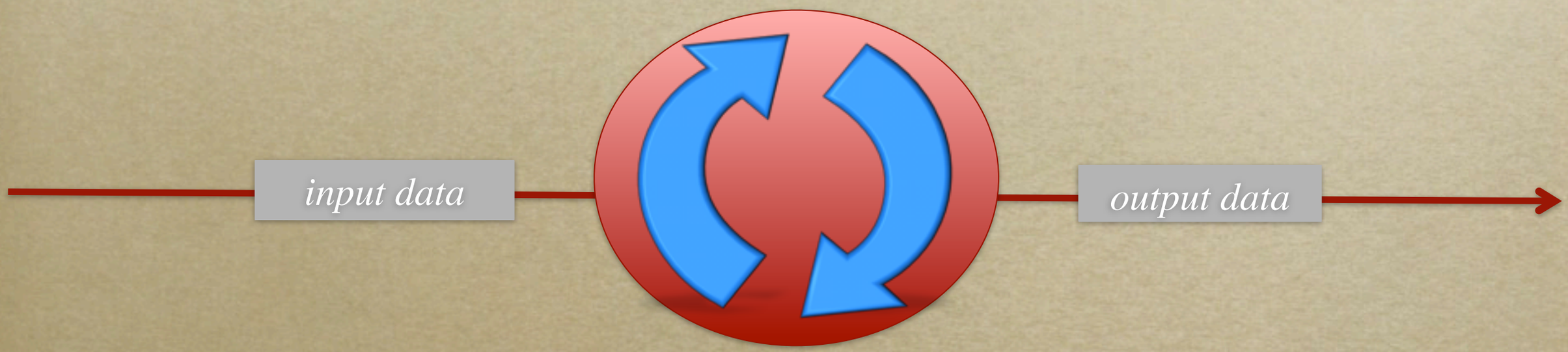
ActualClass PSAFilterGridSearch
BasisFile /agatadisks/bases/ADL/LibTrap_C002.dat
SaveDataDir /agatadisks/data/2010_week24/Replay_run100/Out/1B
EnergyGain 2
WriteTraces 1000
XtalkFile xinv_1325-1340.cal

for tracking algorithm

ActualClass TrackingFilterMGT
SaveDataDir /agatadisks/data/2010_week24/Replay_run100/Out/Global
EnergyGain 2
Ancillary
SourcePosition 0 0 -59
TimeWindowGeAnc 640 690
WriteInputHits

Actor's communication

at running time ...



the actor

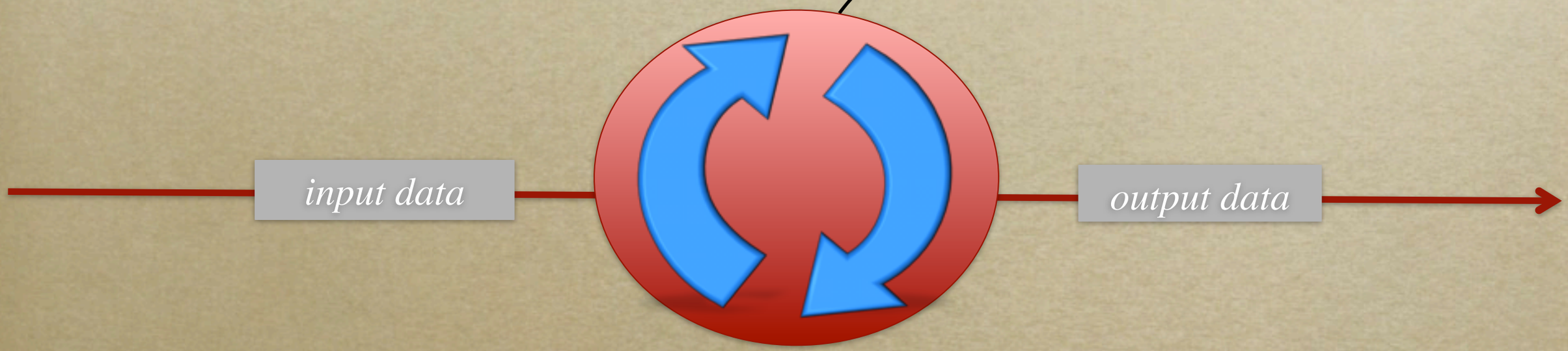
- *extracts data from the input*
- *consumes them or not*
- *produces new data*

not only data can come from input
(ex : configuration frames)

Actor's communication

at running time ...

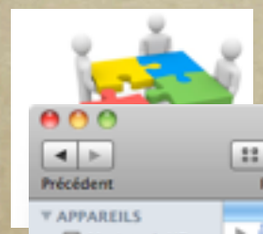
messages (printout, log system)



the actor

- *sends messages*

Actor's communication



save spectra in files

SaveDataDir /agatadisks/data/2010_week24/Replay_run100/Out/1B

Prep__1000-42-100-S__Traces.samp
Prep__2-10-16384-UI__Esum.spec
Prep__2-40-1024-UI__TT1.spec
Prep__38-16384-UI__Ener.spec
Prod__1000-42-100-S__Traces.samp
Prod__38-65536-UI__Ener.spec
Psa__1000-2-42-60-S__Traces.samp

+tkt

or spectra available with GRU

GANIL ?

actor is a server of spectra
☛ machine, port #

the actor

- *produces spectra (if activated)*

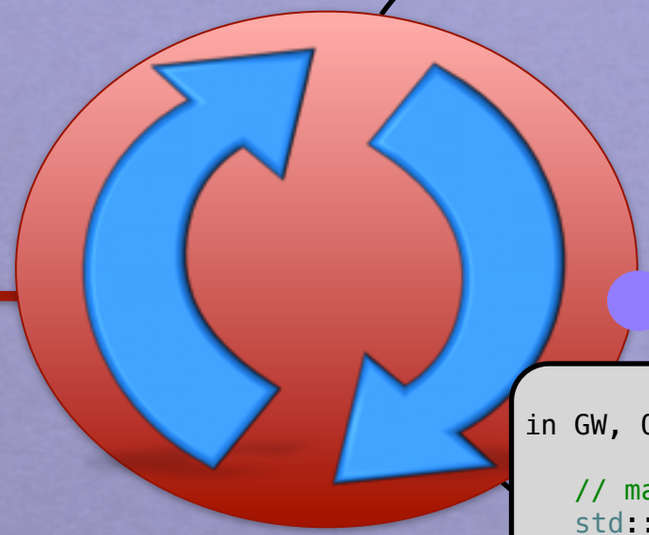
Actor's communication

at running time ...

Run control

Set/GetParameters()

Only in Narval



input data

output data

the actor can

- *have inner parameters modified*
- *send sometime copy of the output over the network*
  - ➥ *online watchers*

```
in GW, OnlineWatchers.C

   // machine on which crystal producer is running
   std::string gMachineCrys[gNumberOfCrystals] =
   {
        "10.10.1.119", "10.10.1.113", "10.10.1.114",
        "10.10.1.115", "10.10.1.116", "10.10.1.117",
        "10.10.1.109", "10.10.1.110", "10.10.1.111",
        "10.10.1.106", "10.10.1.107", "10.10.1.108",
        "10.10.1.103", "10.10.1.104", "10.10.1.105"
   };
   UInt_t gPortCrys[gNumberOfCrystals] =
   {
        9012, 9013, 9014,
        9015, 9016, 9017,
        9009, 9010, 9011,
        9006, 9007, 9008,
        9003, 9004, 9005
   };
```

Overview

Actor's actions

What already exists

Watchers

Data flow structure

Data flow structure

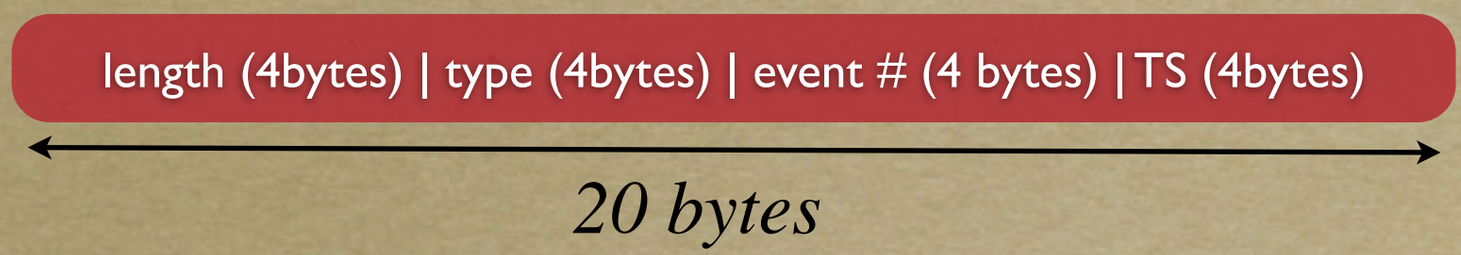
How look like the input and output buffers ?

➥ **Basic element : Frame**

*A Frame is composed of a **key** and a **content***

the key part give information on the content ☛ *always starts with frame length* ☛ *mapping possible without knowning the content*

Agata key



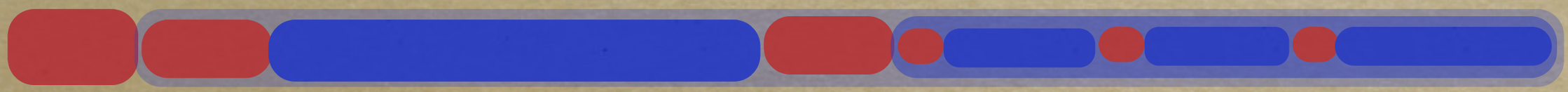
Data flow structure

There are different kind of Frames

- *data : produced (consumed by algo.) [binary]*
- *conf : to pass human readable info [ascii]*
- *meta : general data (ex: Vertex)*

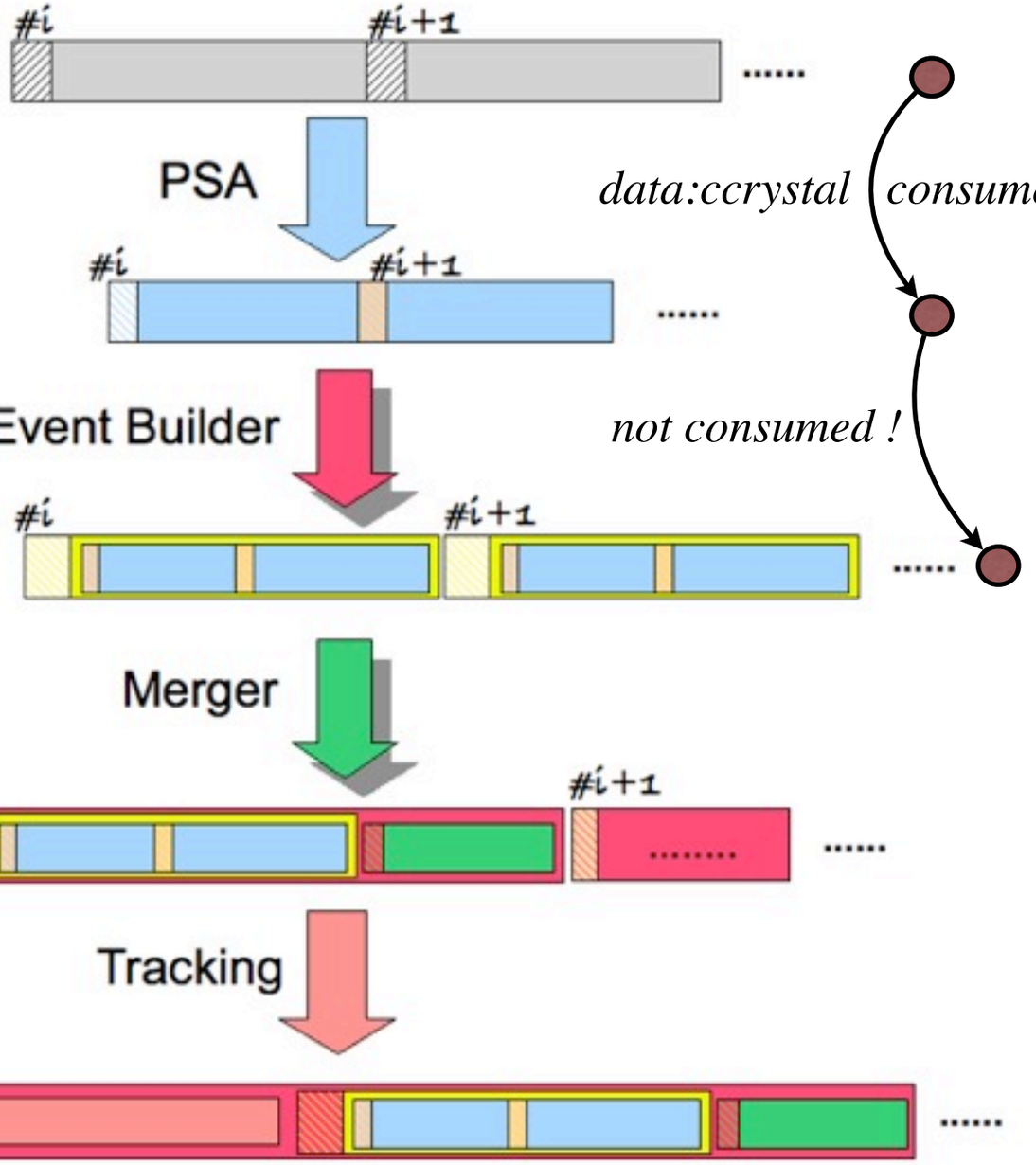
A Frame could be composed of Frames : CompositeFrame

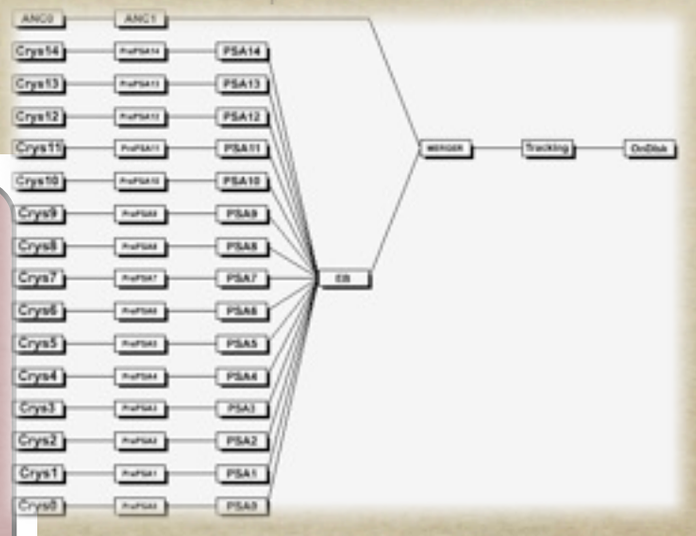
➥ *coincidences ≋ event*



Data flow structure

Frames currently in uses



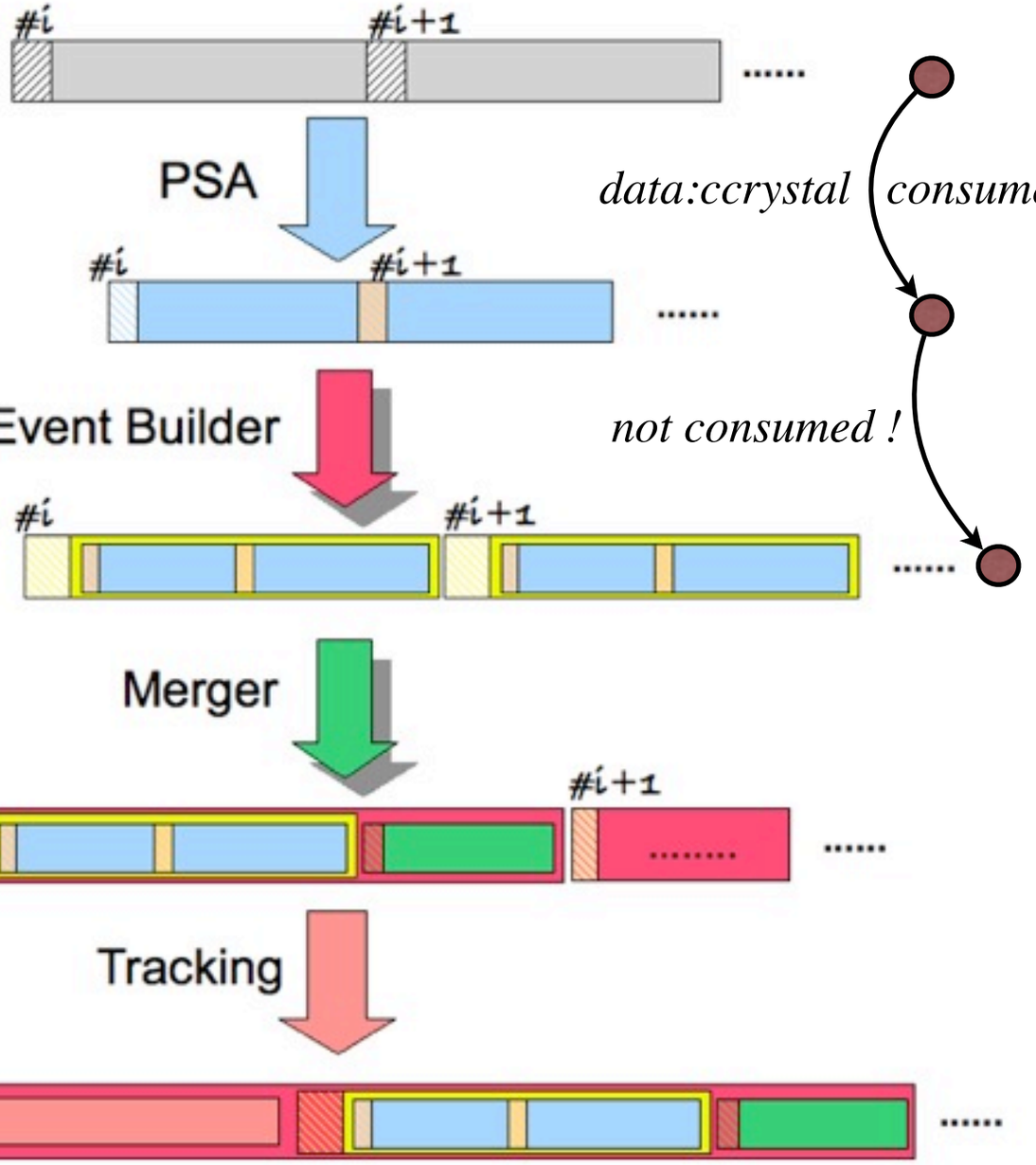


disctinction using Key[type]

to be defined @ GSI

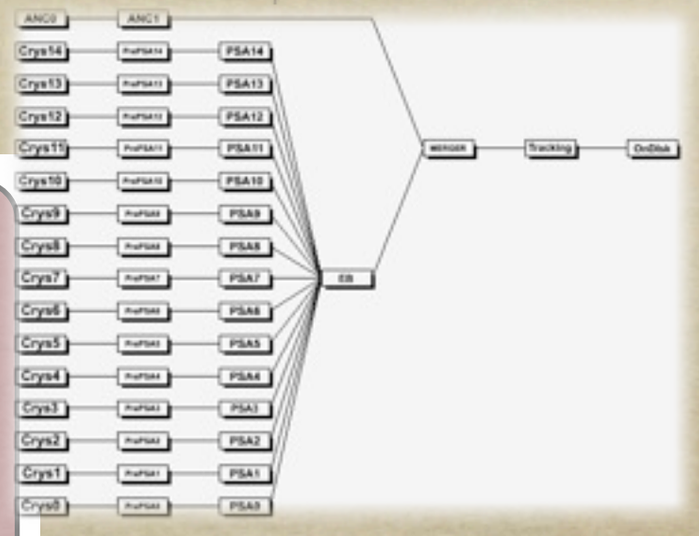
Data flow structure

Frames currently in uses



data:ccrystal *consumed !*

not consumed !



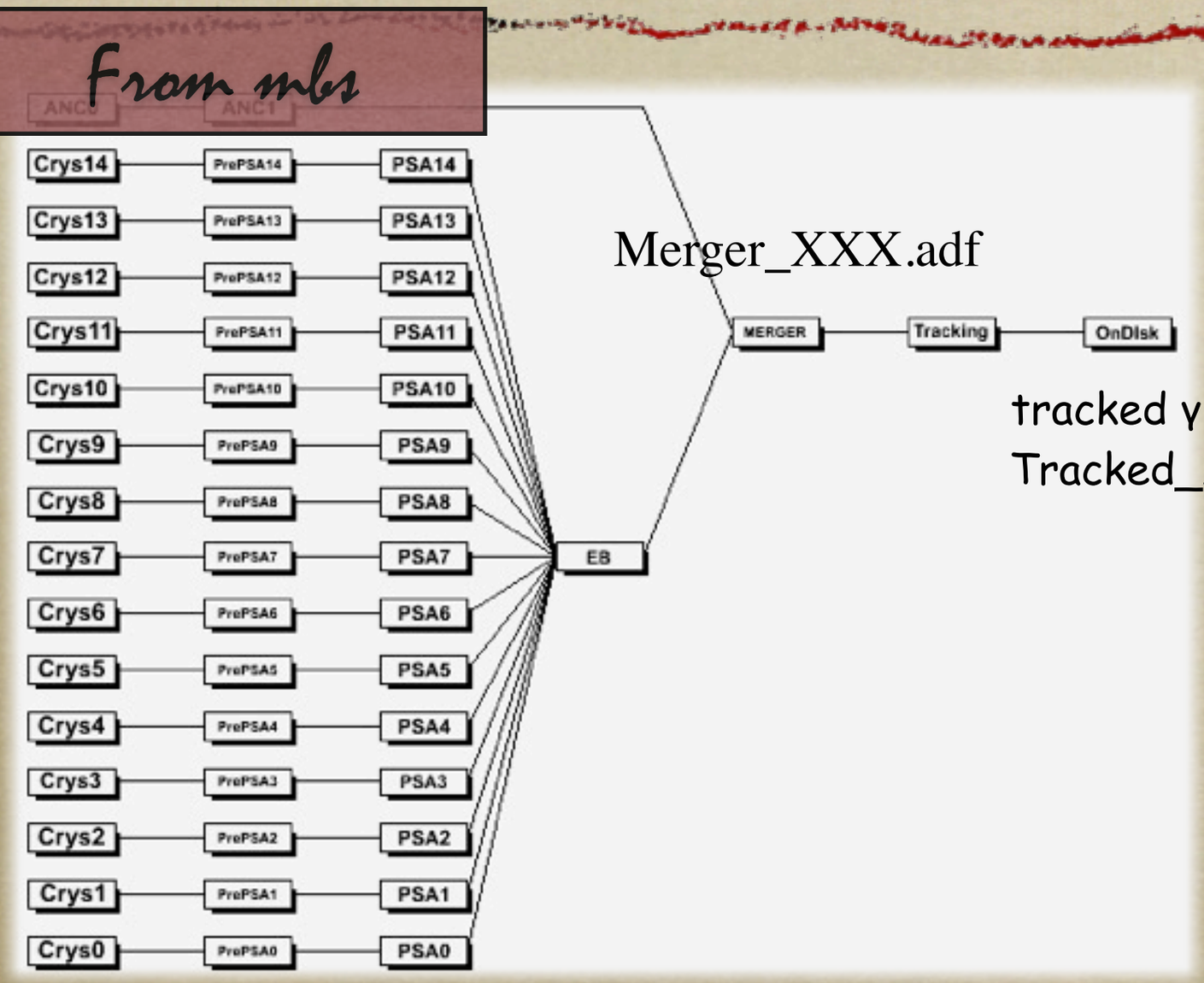
disctinction using Key[type]

to be defined @ GSI

done as data:ranc0 ≙ mbs structure

Data flow structure

From mbs



traces saved in
event_mezzdata.bdat

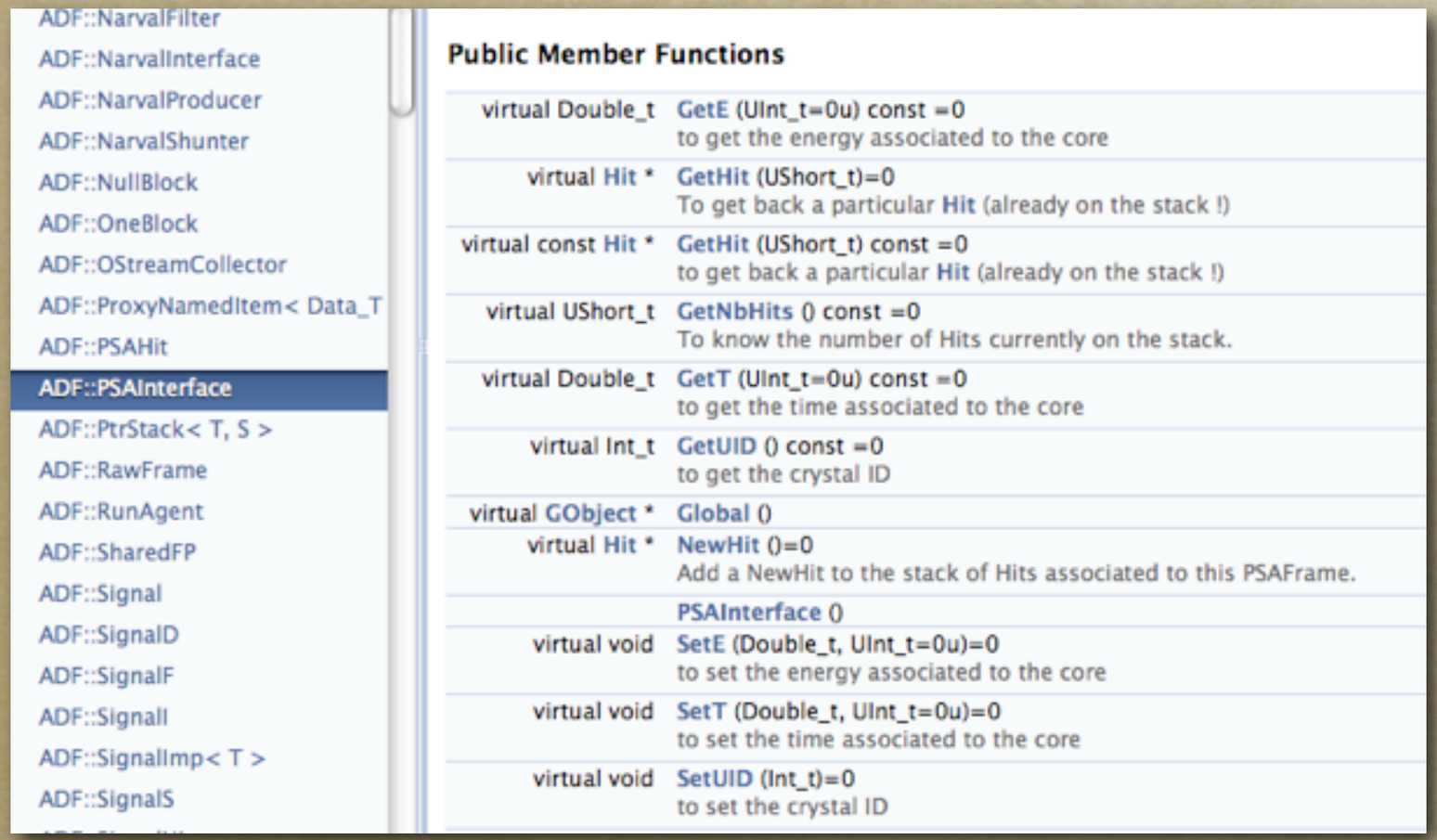
psa hits saved in
PSA_1R_XXX.adf

The Data Interface

To help the users using Frames,

➥ *data interface to access its content*

Ex : PSA Interface (list of Hits)



to access individual hits

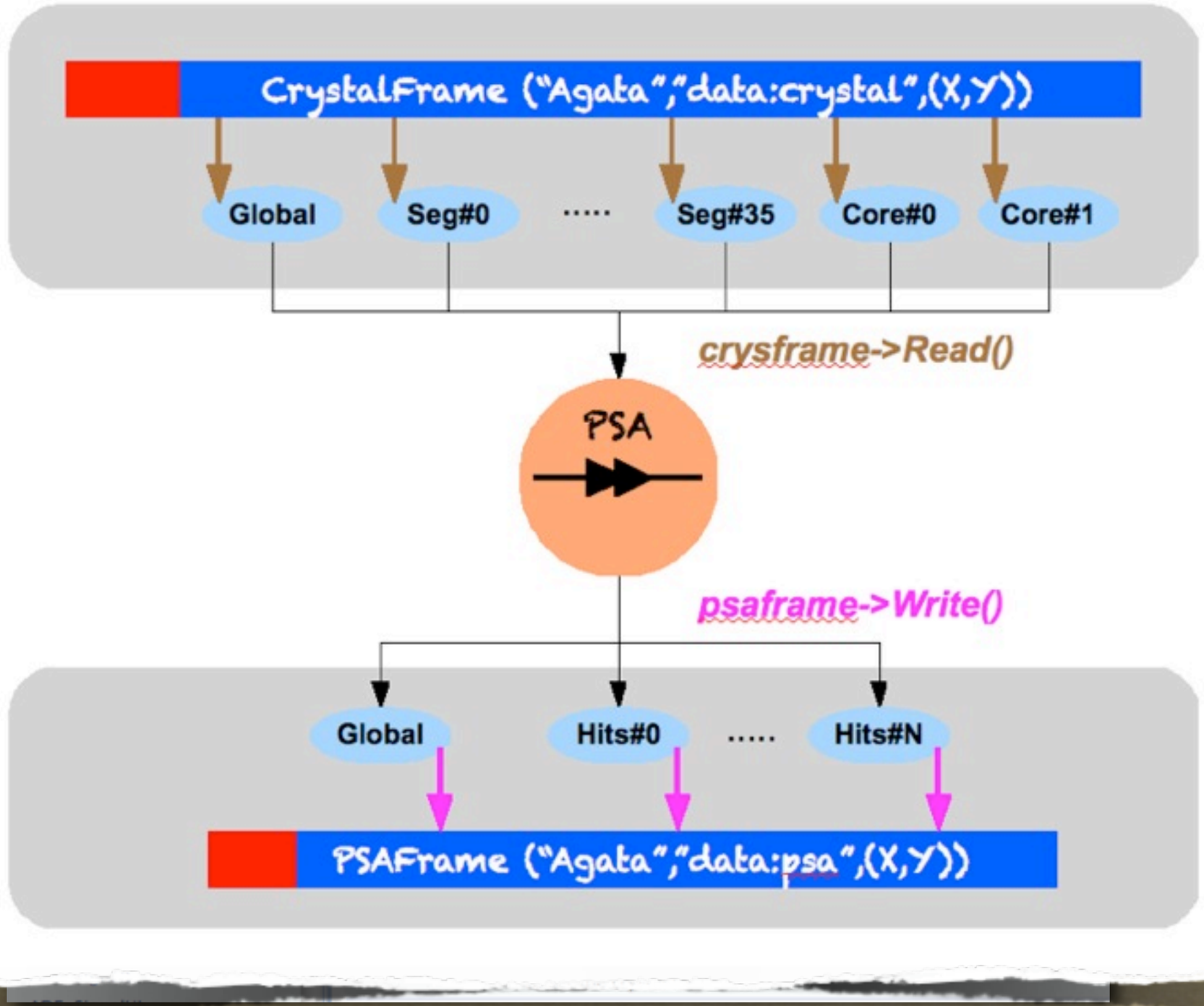
# of hits

to add a new hit

The Data Interface

To h *t*

Ex :



ess individual hits

its

d a new hit

Frame has versions

Why version numbers ?

AGATA : 10 years project

➥ *algorithms (Tracking, PSA) likely to change*

➥ *better to include this since the beginning*

small modifications - float ☛ double

Version (Major, Minor)

large modifications - error on hits

Frame has versions

the full definition of circulating frames are in ADF.conf

```
...
Frame:  Agata data:crystal 4 0  Agata data:crystal 65000 0
Frame:  Agata data:ccrystal 4 0  Agata data:ccrystal 65000 0
Frame:  Agata data:ranc0 4 0  Agata data:ranc0 65000 0
Frame:  Agata data:ranc1 4 0  Agata data:ranc1 65000 0
Frame:  Agata data:psa 4 0  Agata data:psa 65000 1
Frame:  Agata data:tracked 4 0  Agata data:tracked 65000 0
Frame:  Agata event:data 4 0  Agata event:data 4 0
Frame:  Agata event:data:psa 4 0  Agata event:data:psa 4 0
Frame:  Agata meta:vertex 4 0  Agata meta:vertex 0 1
...
```

Since GSI

Note : ADF.conf should be in the directory pointed by ADF_CONF_PATH

Frame has versions

the full definition of circulating frames are in ADF.conf

see
ADF_GSI.adf
ADF_LEG.adf
in adf sources directory

```
ata:crystal 4 0  Agata data:crystal 65000 0
ata:ccrystal 4 0  Agata data:ccrystal 65000 0
ata:ranc0 4 0  Agata data:ranc0 65000 0
Frame:  Agata data:ranc1 4 0  Agata data:ranc1 65000 0
Frame:  Agata data:psa 4 0  Agata data:psa 65000 1  0 0
Frame:  Agata data:tracked 4 0  Agata data:tracked 65000 0  0 0
Frame:  Agata event:data 4 0  Agata event:data 4 0
Frame:  Agata event:data:psa 4 0  Agata event:data:psa 4 0
Frame:  Agata meta:vertex 4 0  Agata meta:vertex 0 1
...
```

Since GSI

Note : ADF.conf should be in the directory pointed by ADF_CONF_PATH

The trigger mechanism

A Actor (analysis) just needs some specific data (Frames) to work

Ex:

1		✔	
2			✔
3	✔		✔

Tracking

prisma, prespec

Doppler correction

Idea : call actions only if condition is found

3

```
// a trigger on event with hits, tracked gammas and ancillary
AgataFrameTrigger *trig =
   AgataFrameTrigger::Build("TRACKING", "event:data event:data:psa data:tracked data:ranc1");
```

The trigger mechanism

A Ac … *rk*

- Add
- Add
- Add
- AddUtility
- AddUtility
- AddUtility
- AgataFrameTrigger
- Build
- SetOutputFrame
- SetOutputFrame
- SetOutputFrame
- SetOutputFrame
- ~AgataFrameTrigger

```
virtual SharedFP* ADF::AgataFrameTrigger::Add ( const FactoryItem &  key_item,
                                                const FactoryItem &  frame_item,
                                                bool                 iscons = true,
                                                const Char_t *       option = ""
                                              )                      [inline, virtual]
```

Add a frame to the list of required frames.

the frame is added with a flag to tell if the given frame is consumable or not i.e. if it is still on the output dataflow in case the trigger defines also an output frame. isconc true if

A FrameTrigger has a tree (stack) structure as followed :

```
MainFrame (#0)
        SubFrame (#1)
        SubFrame (#2)
        ...
```

The mainframe is mandatory, other ones (subframes inside the main one) are optional.

Options are :

Options →

- none (default) means one and exactly one such a kind of frame is expected to trig
- ! : anti coincidence means the current trigger does not contain the frame
- | : means the frame is there or not. In such case, check if the frame is there with IsIndividualFired

Reimplemented from ADF::FrameTrigger.

Definition at line 492 of file Trigger.h.

3

```
// a trigger on event with hits, tracked gammas and ancillary
AgataFrameTrigger *trig =
   AgataFrameTrigger::Build("TRACKING", "event:data event:data:psa data:tracked data:ranc1");
```

Overview

Actor's actions

What already exists

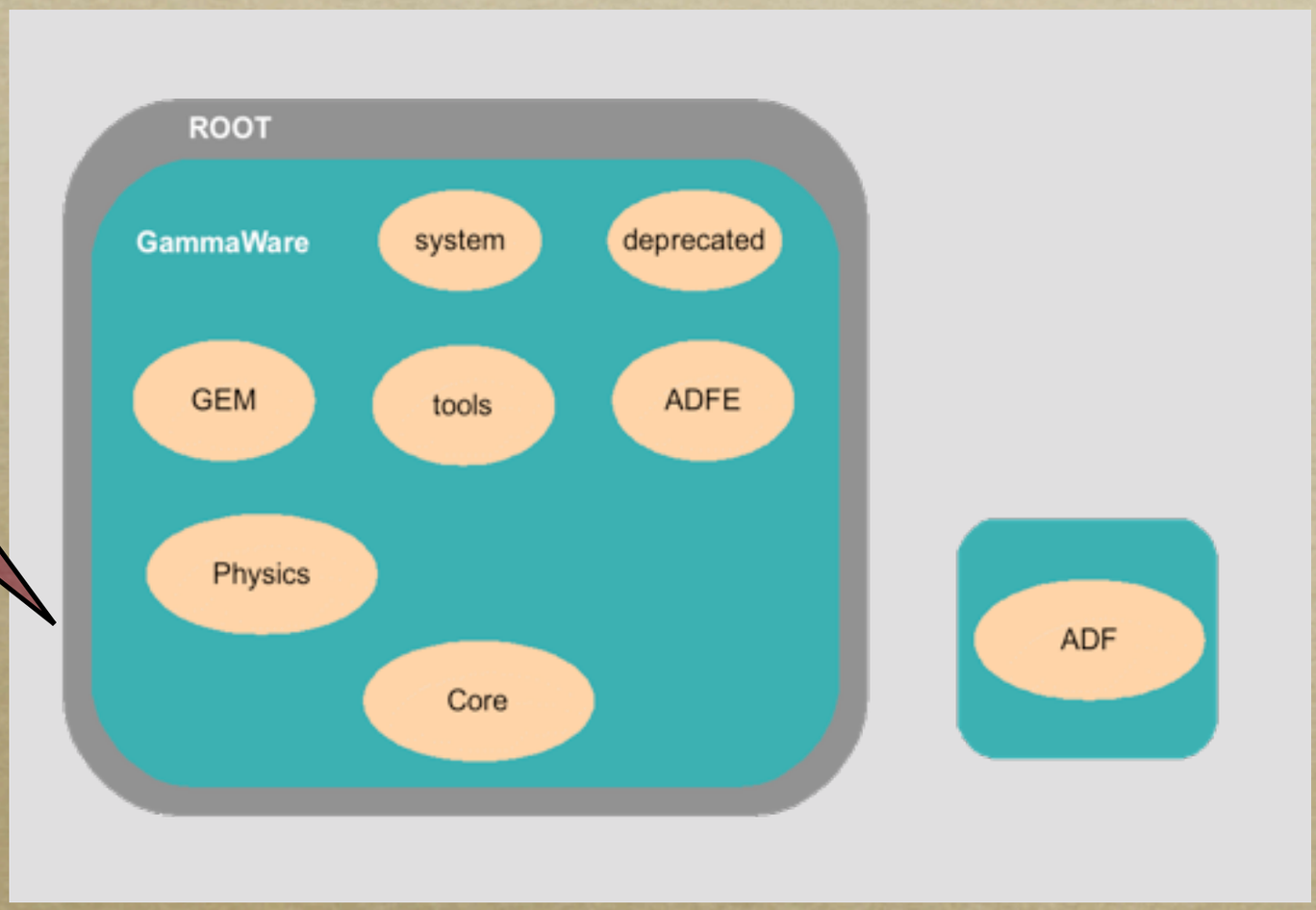
Watchers

Data flow structure

GammaWare

ROOT : huge framework for data analysis

installation
cmake
&
AgataSoftware.py

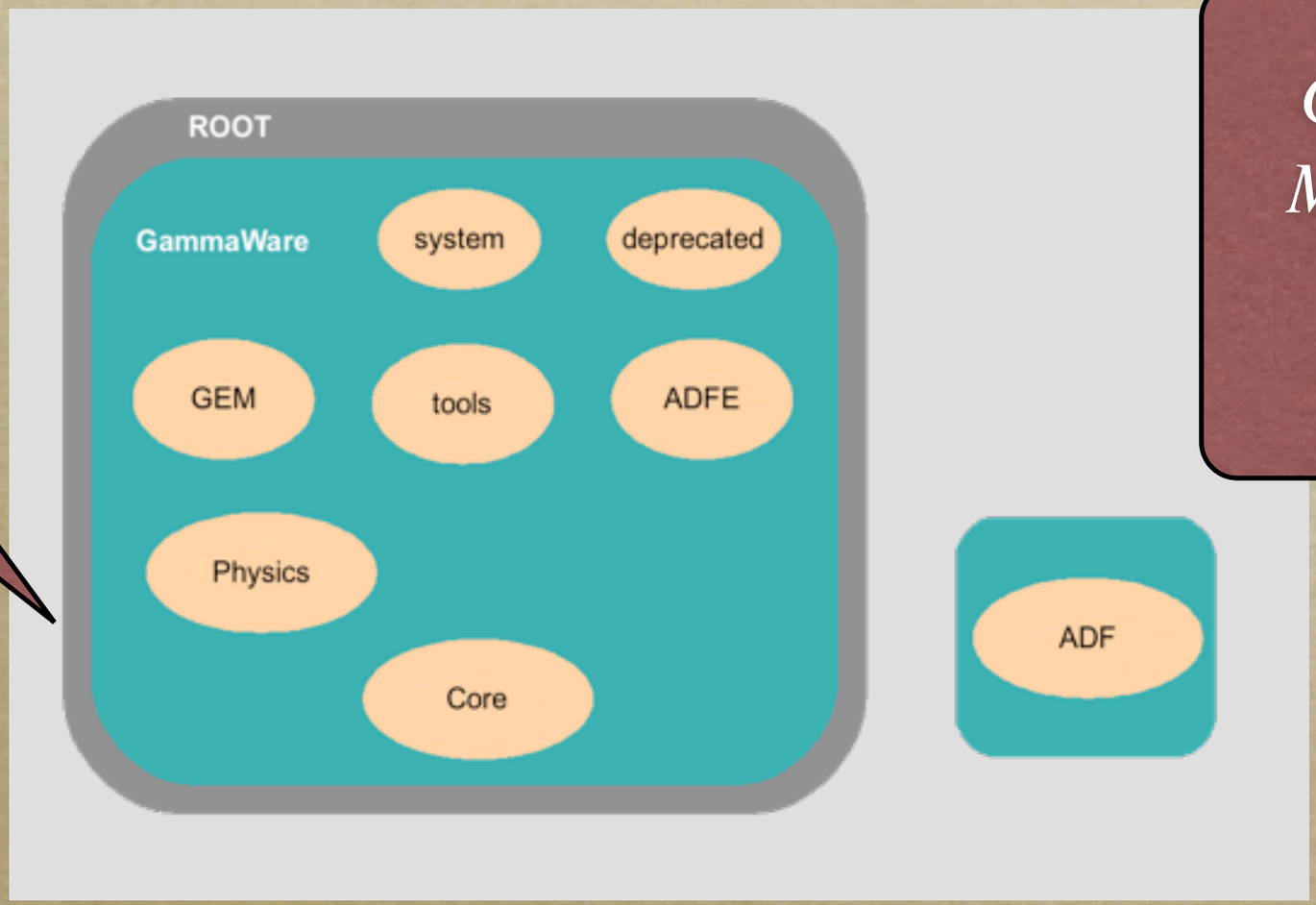


Gw: libraries to add to root γ-ray like analysis

GammaWare

ROOT : huge framework for data analysis

installation
cmake
&
AgataSoftware.py



Graphical fit
Matrix player
LS Player
...

Gw: libraries to add to root γ-ray like analysis

Watchers[Gw]

Def : task dedicated to a given Frame (Trigger)

idea : re-usable bricks

Ex of watchers :

on data:psa

task to display hits in 3D

on data:ranc1 *Legnaro*

display raw spectra

calibrate + do recoil velocity, make it available for other watchers

display recoil velocity

on data:tracked

γγ matrix

write out in a root tree

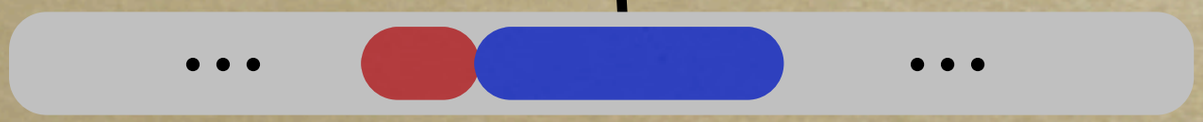
Watchers[Gw]

Watchers can be linked in actors

```
...
FrameDispatcher *fd;
...
// a trigger on event with hits, tracked gammas and ancillary
AgataFrameTrigger *trig =
     AgataFrameTrigger::Build("TRACKING", "event:data event:data:psa data:tracked data:ranc1");

// set the main trigger to coincidences between ancillary et gammas
fd->SetTrigger(trig);

// specific to tracked frame
// gamma-gamma matrix
fd->Add<Coinc2D>(GetWN("GxG",ext),"Gamma Gamma coincidence");
...
```



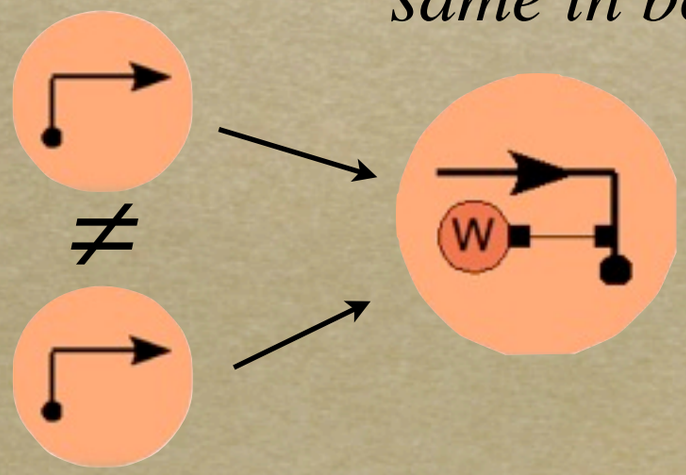
Coincidences 'for free' !

same in both cases !!

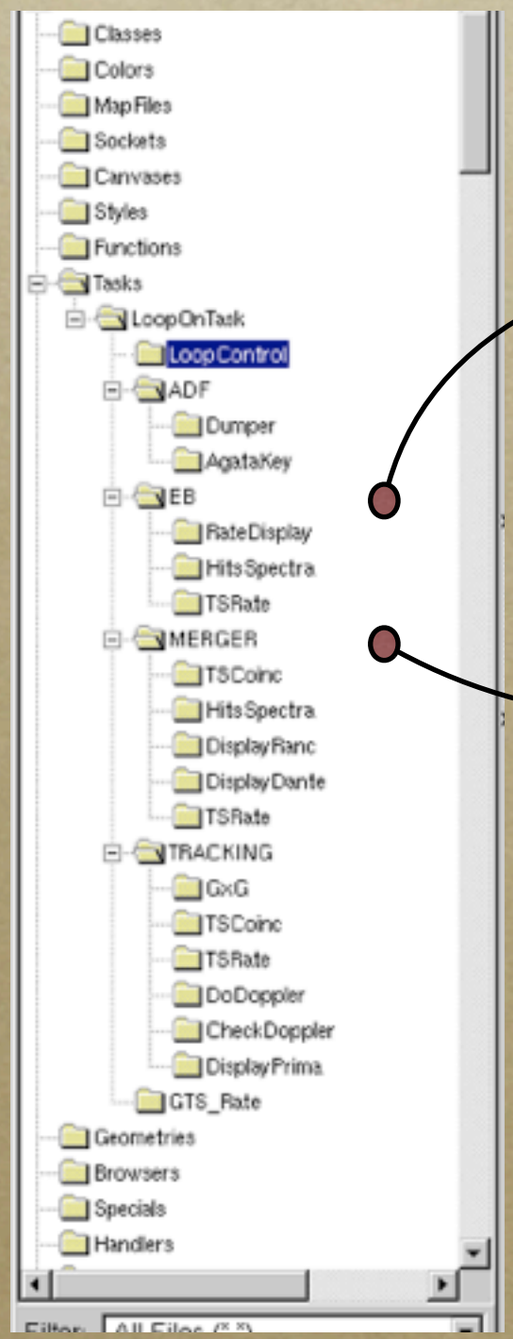
Online, get data from socket

≠

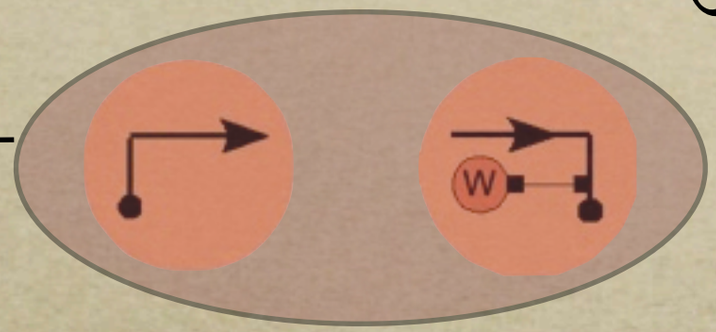
Offline, get data from .adf files

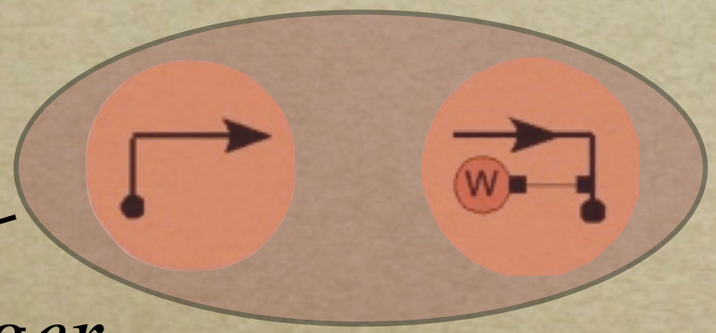


Watchers[Gw]



out of EB



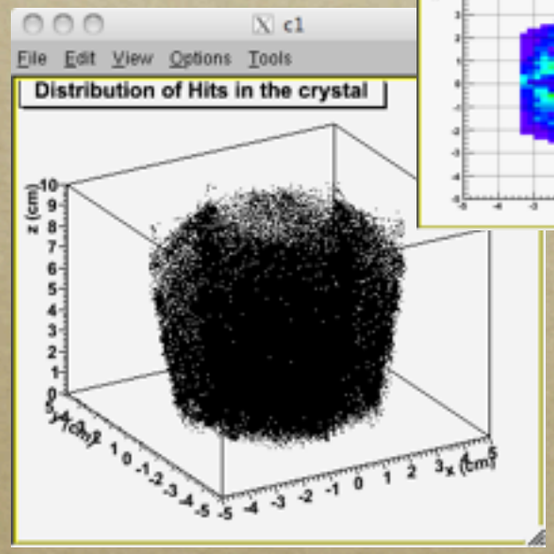


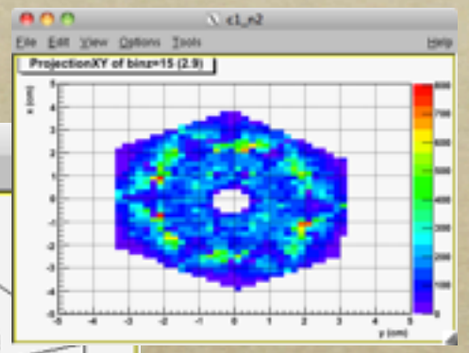
out of merger

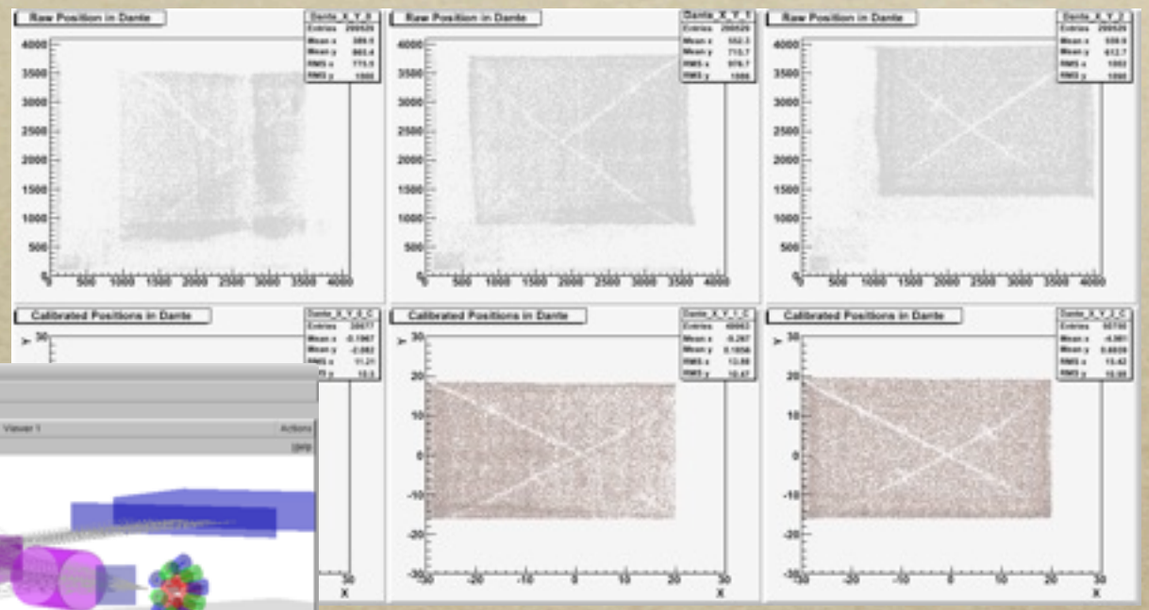
Simple emulators

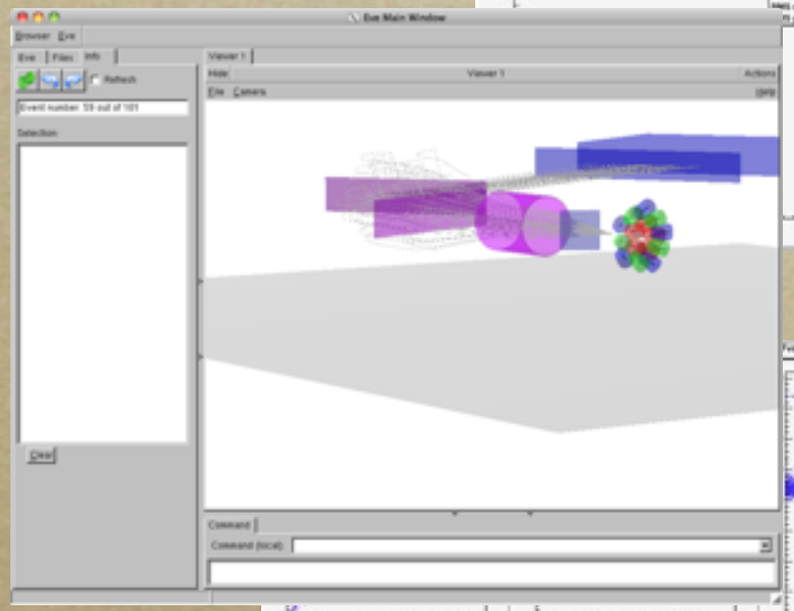
Online @ Global Level Processing

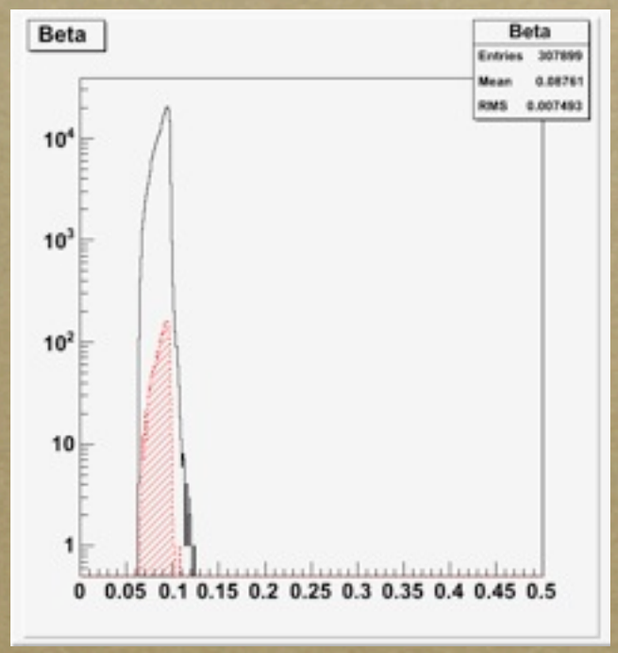
Watchers[Gw] @Legnaro

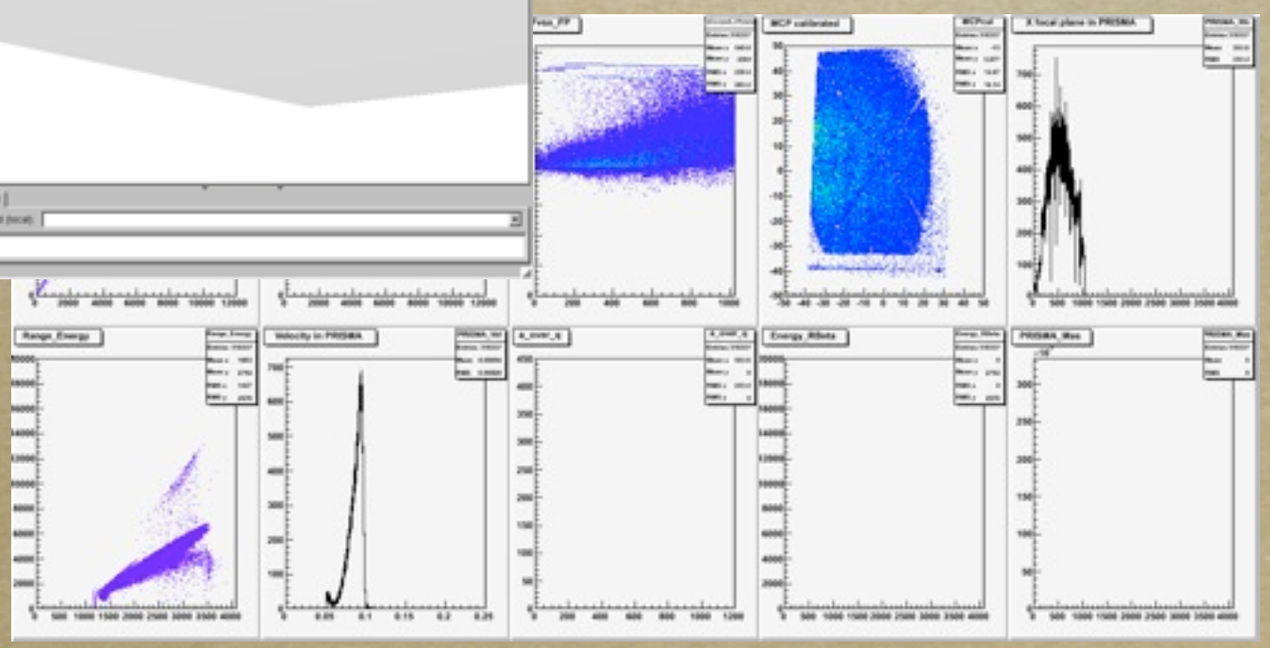






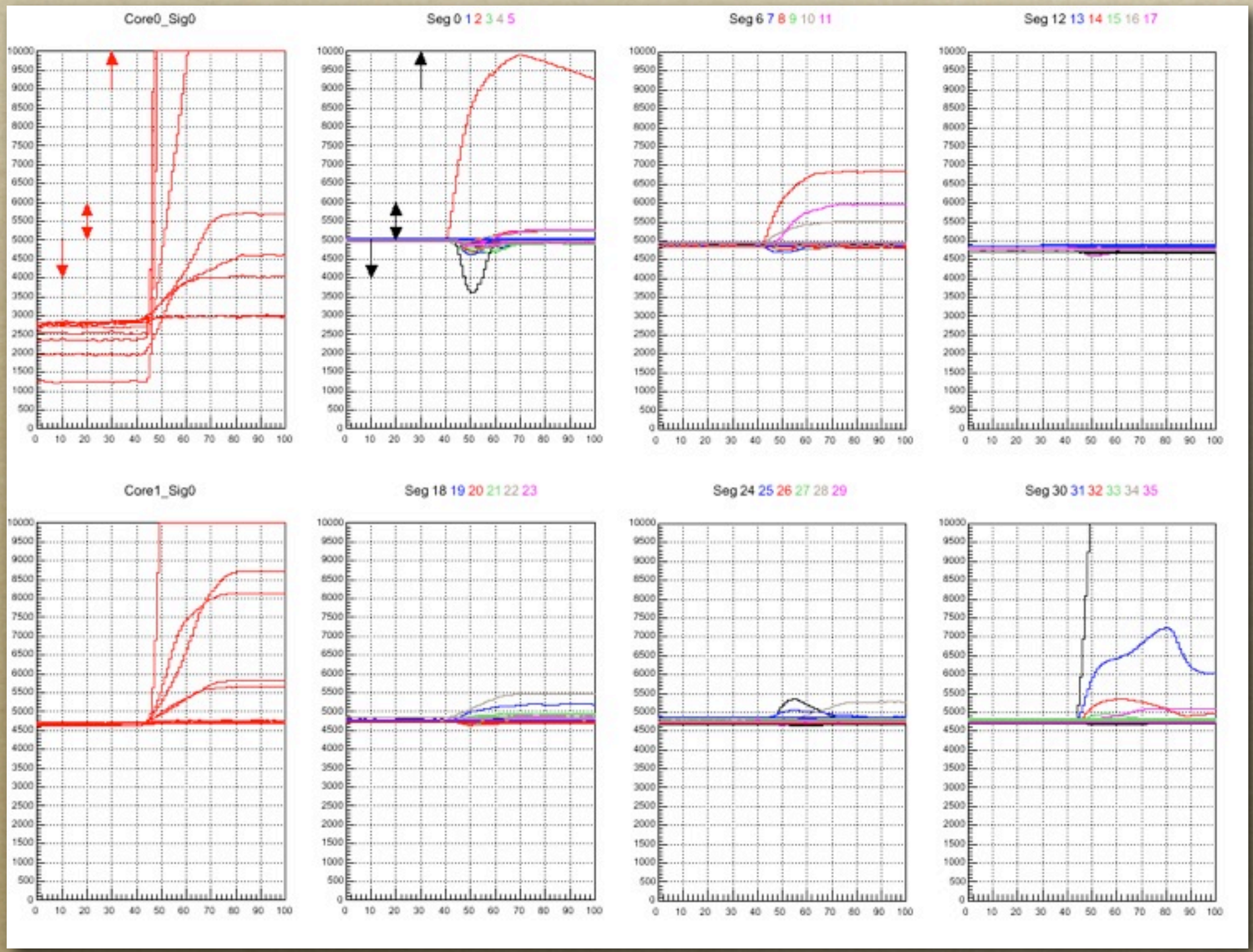






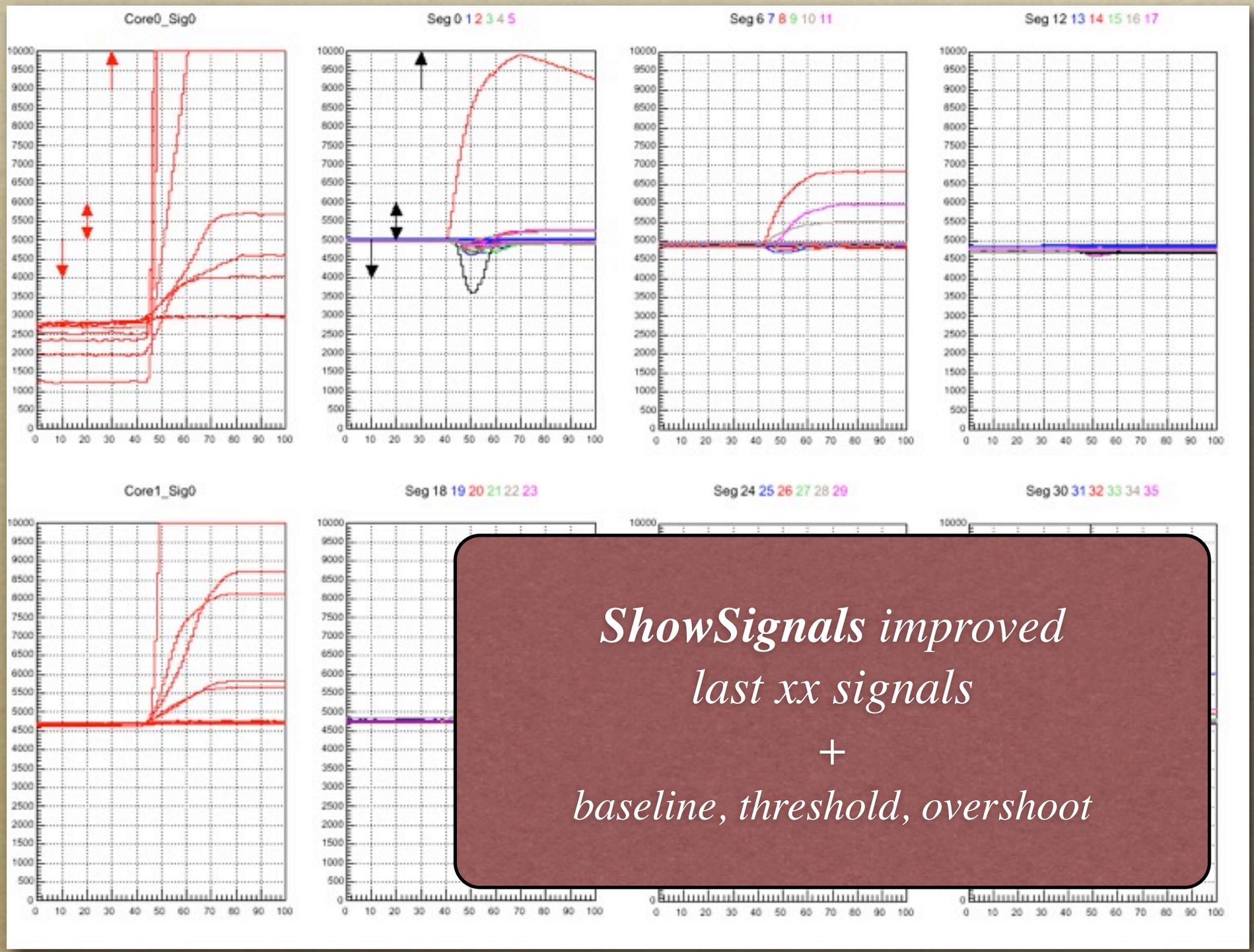
Watchers[Gw] @GSI LLP

on data:crystal / data:ccrystal



Watchers[Gw] *@GSI LLP*

on data:crystal / data:ccrystal



ShowSignals *improved*

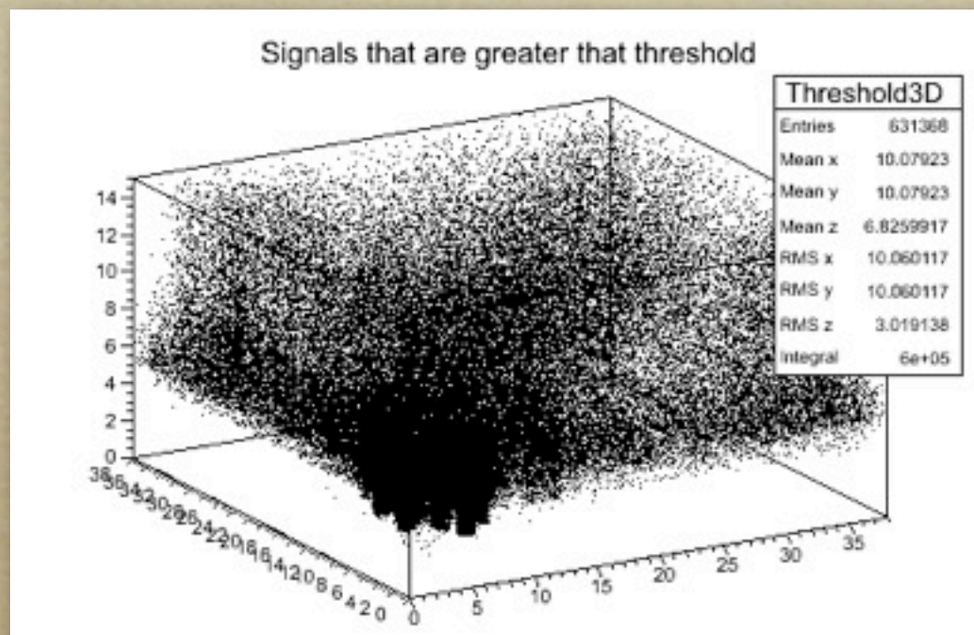
last xx signals

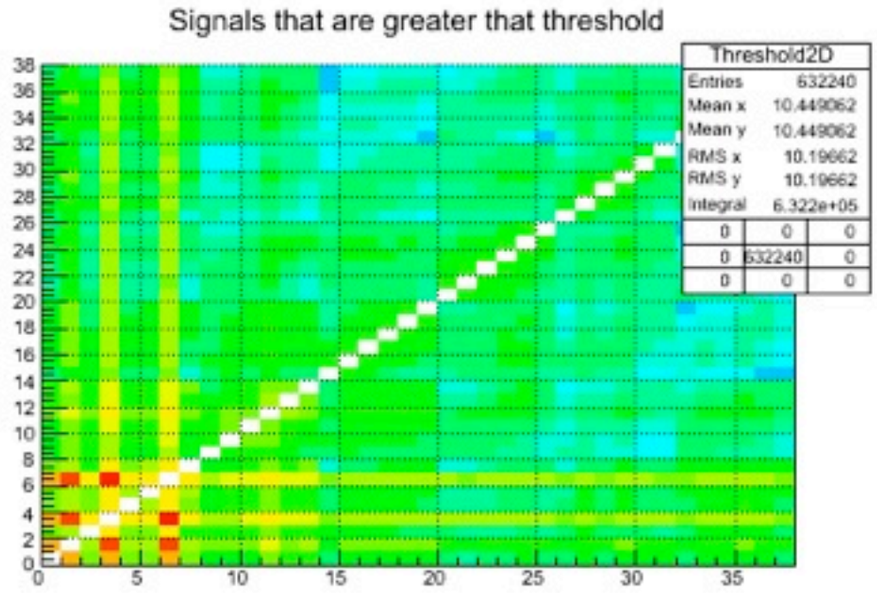
+

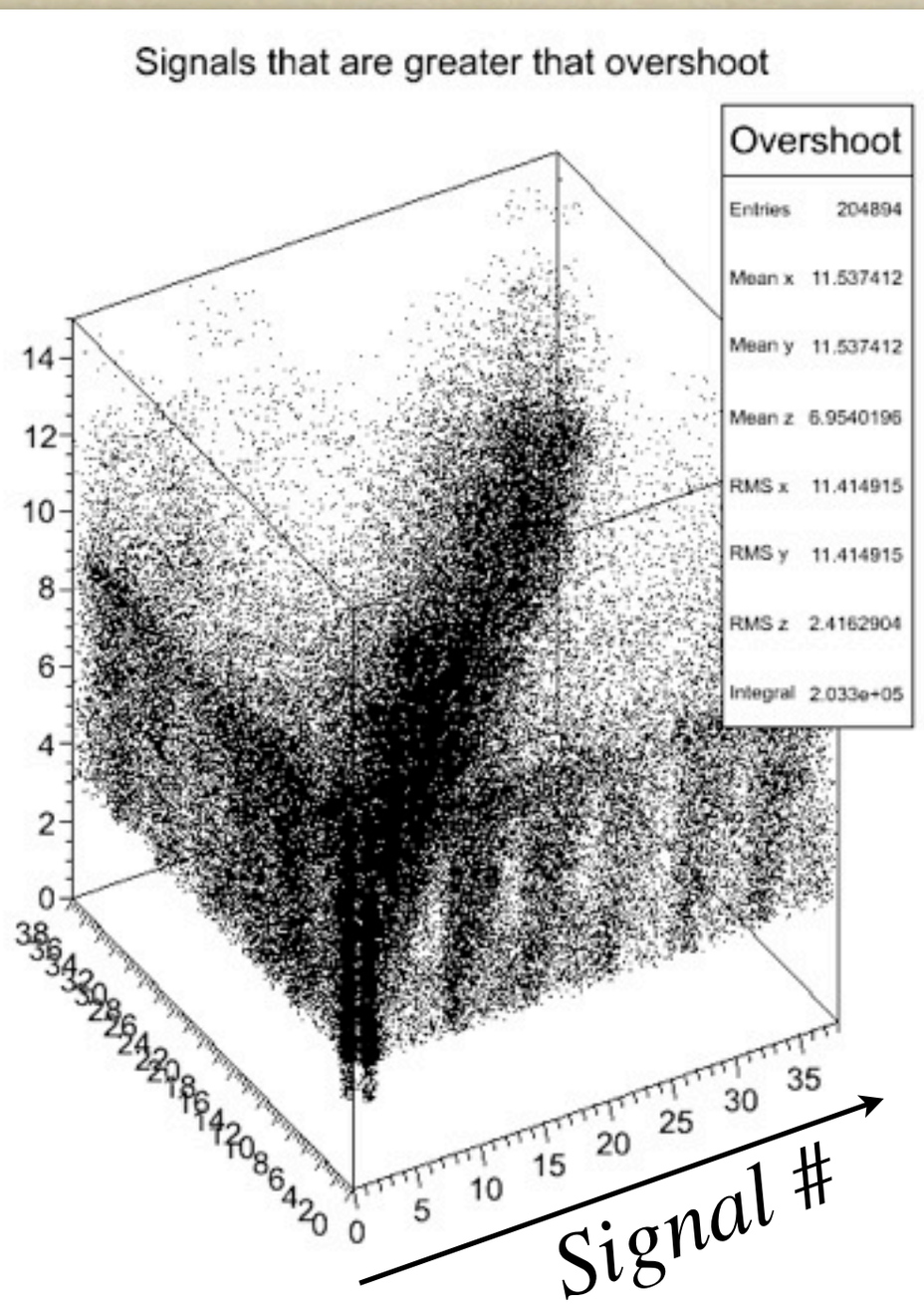
baseline, threshold, overshoot

Watchers[Gw] @GSI LLP

on data:crystal / data:ccrystal

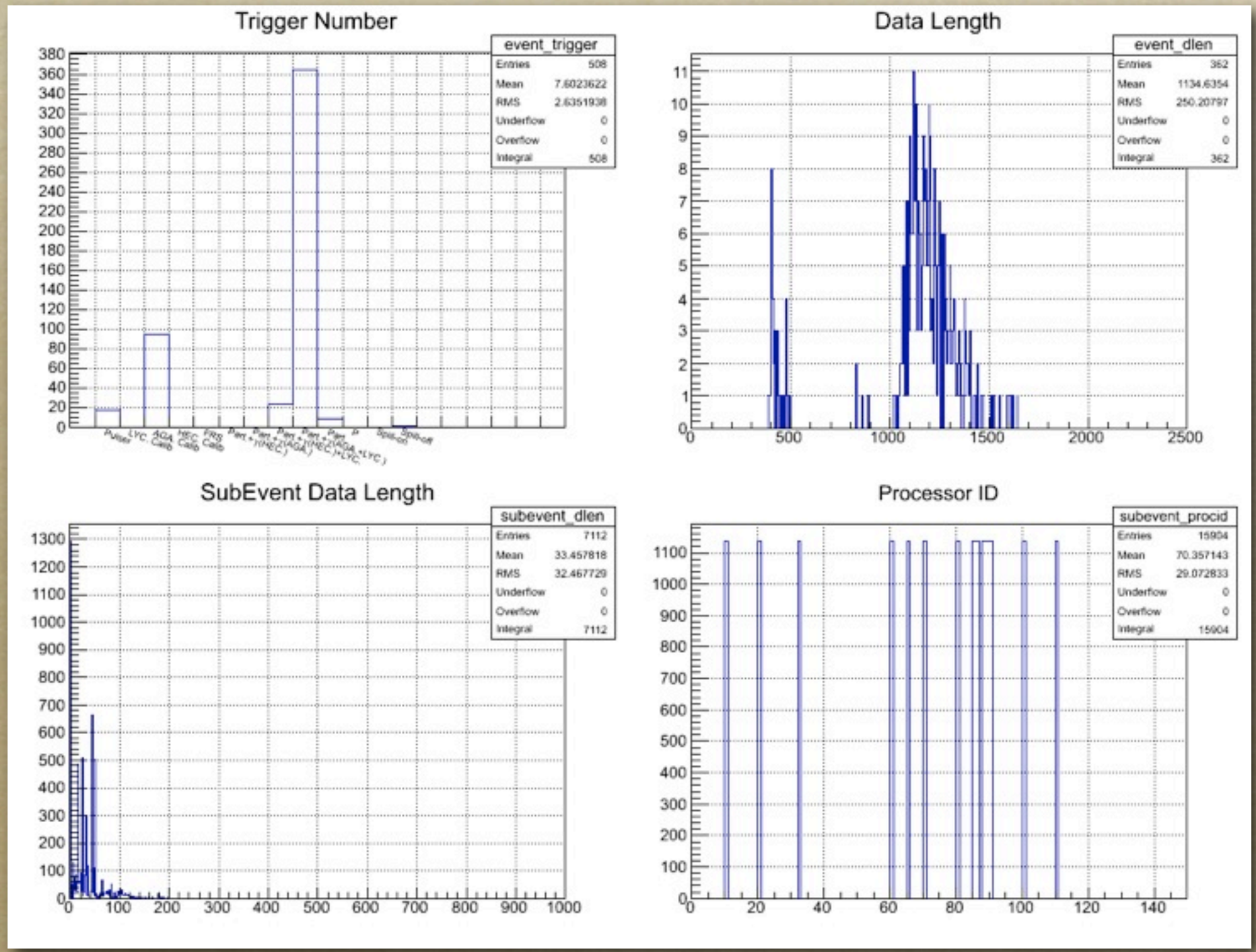






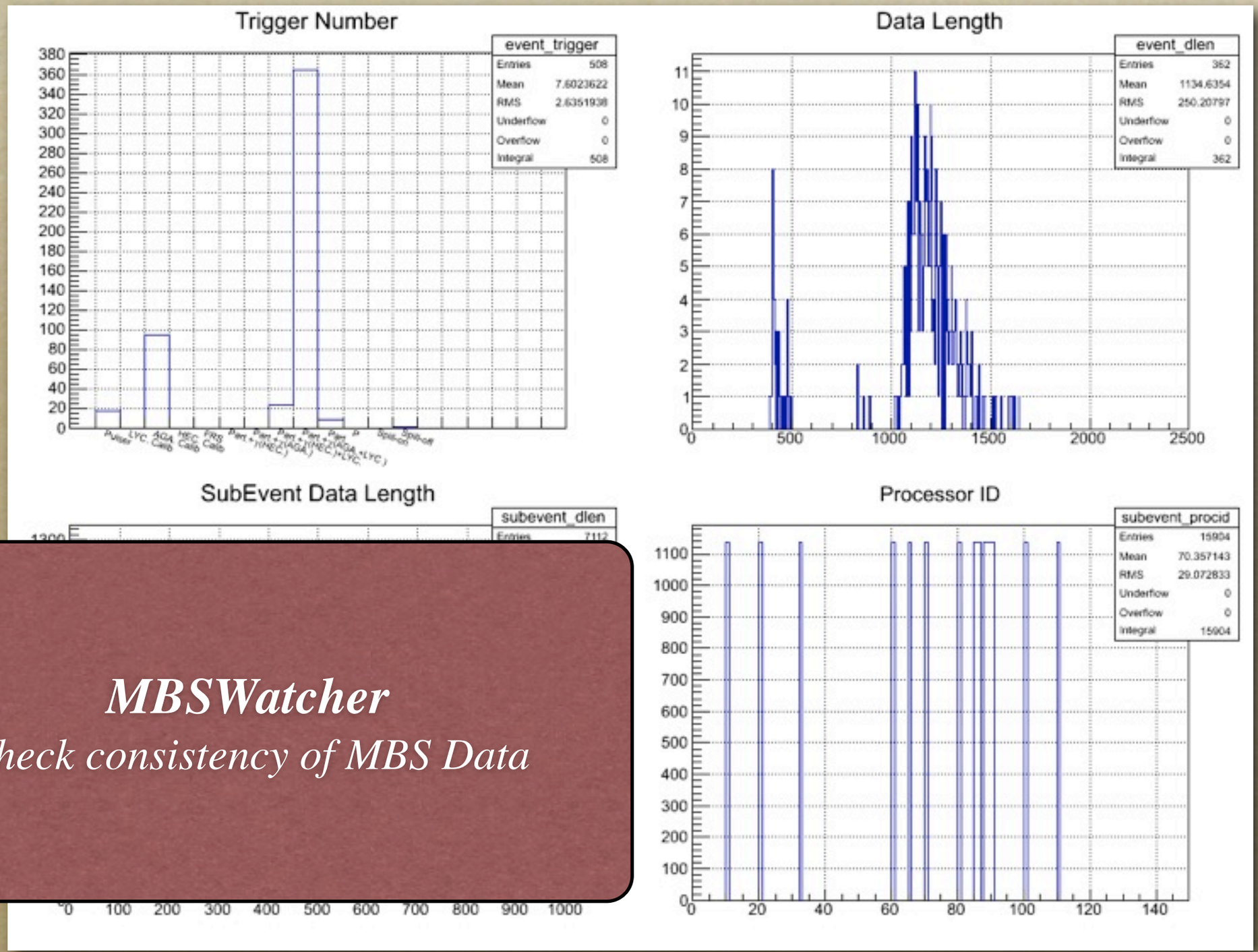
Watchers[Gw] @GSI LLP

on data:ranc0



Watchers[Gw] @GSI LLP

on data:ranc0

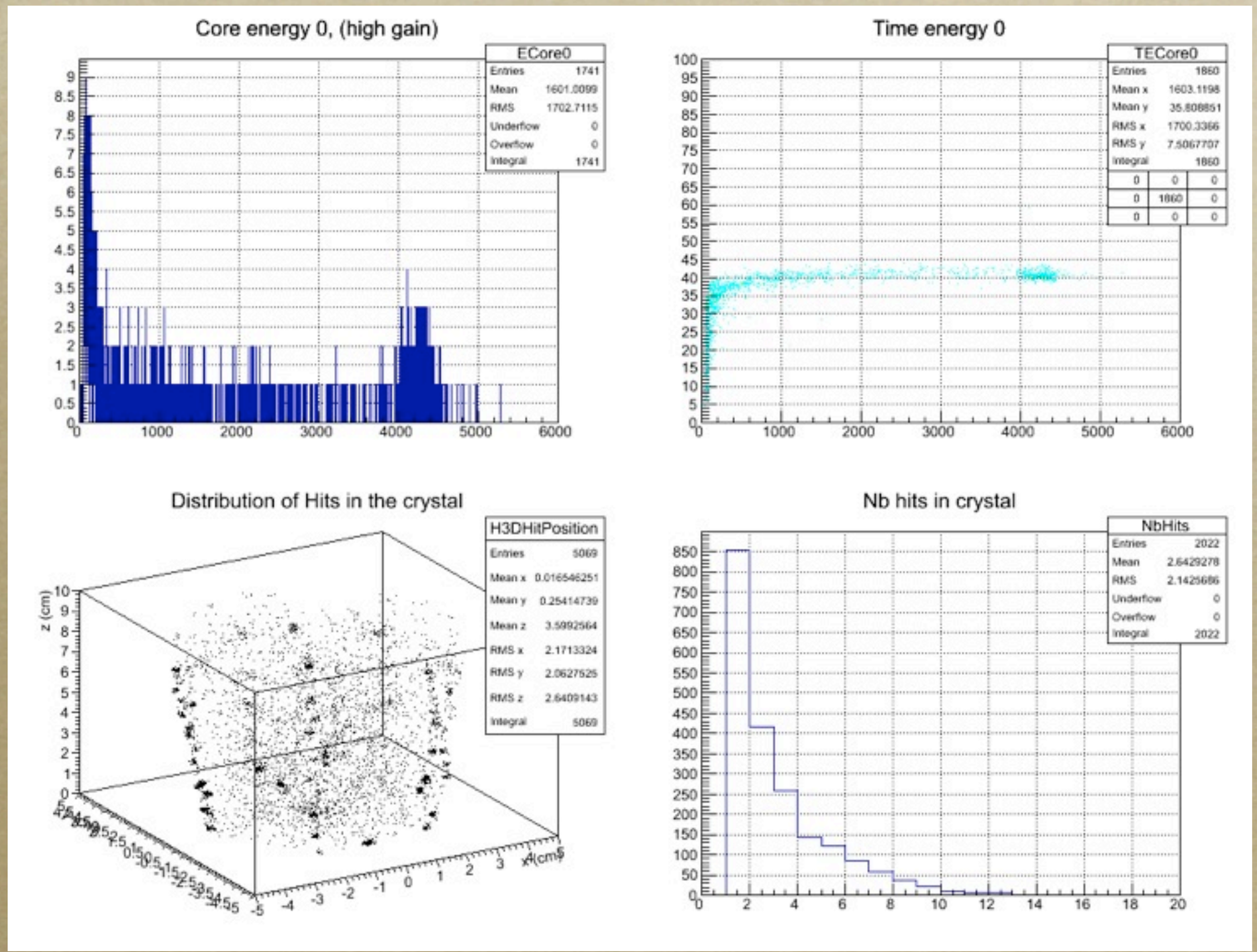


MBSWatcher

Check consistency of MBS Data

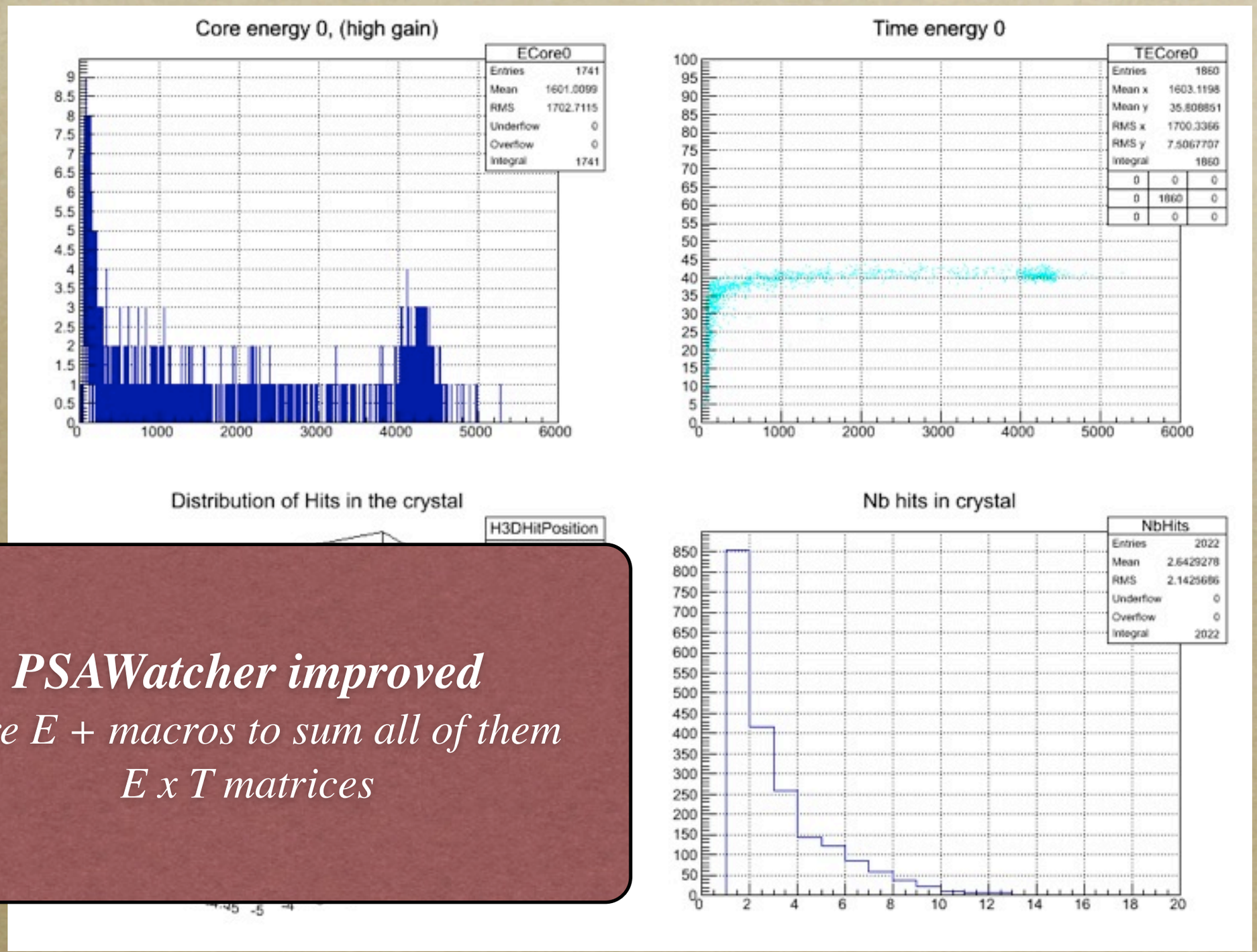
Watchers[Gw] @GSI LLP

on data:psa



Watchers[Gw] @GSI LLP

on data:psa



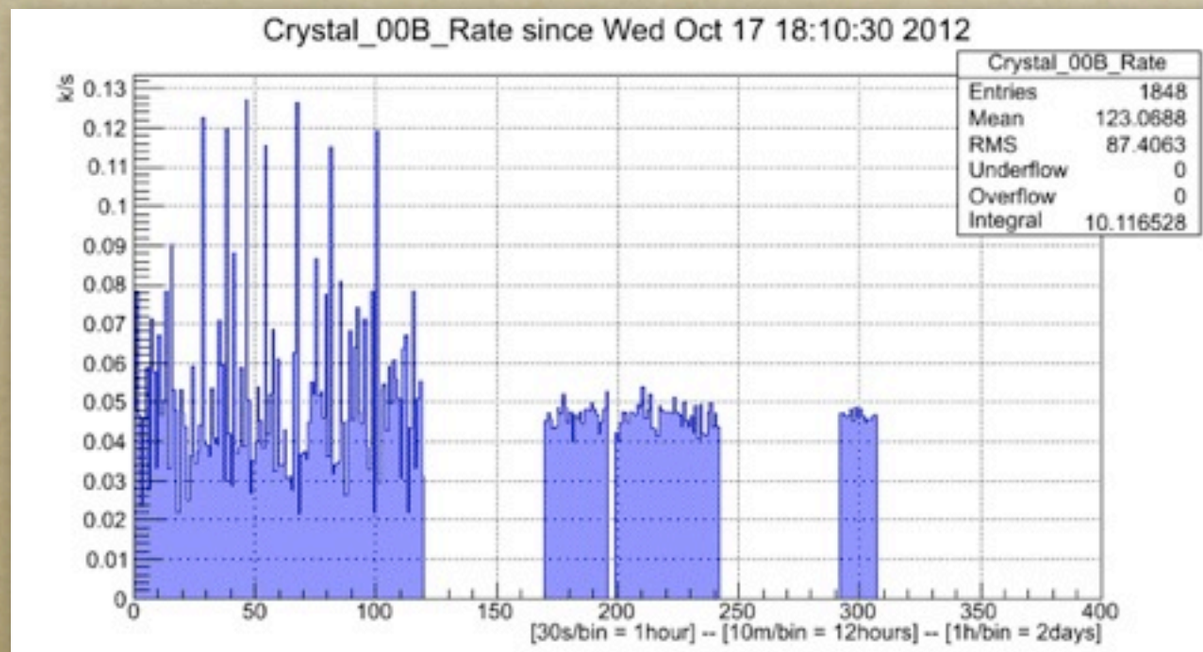
PSAWatcher improved

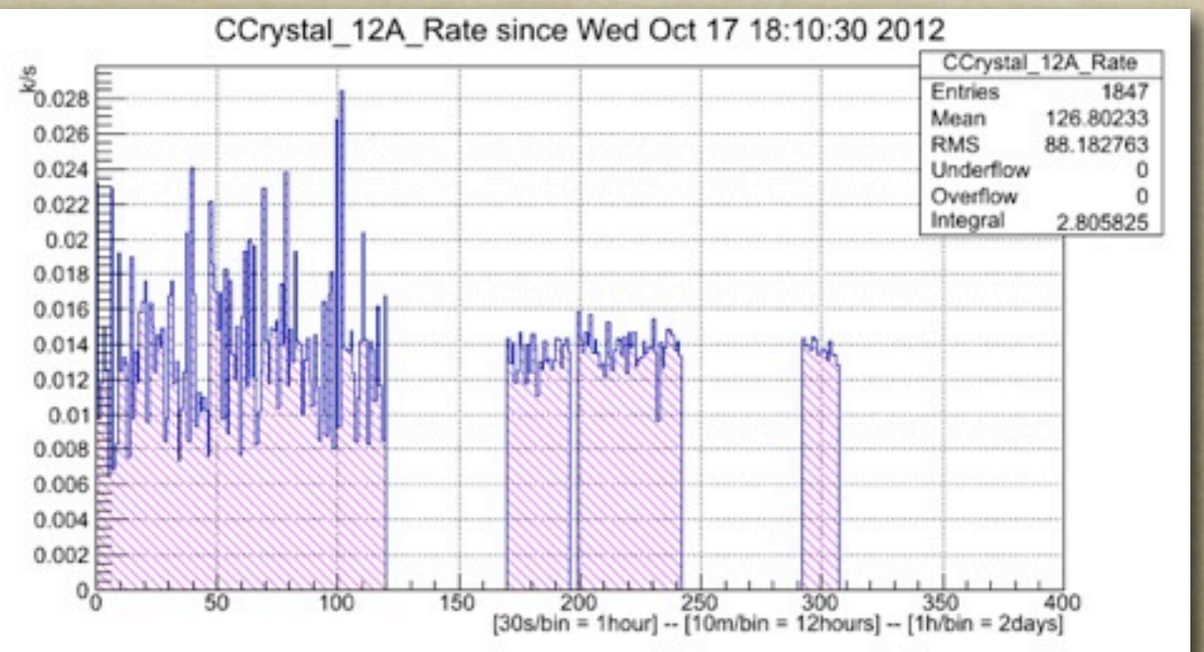
Core E + macros to sum all of them

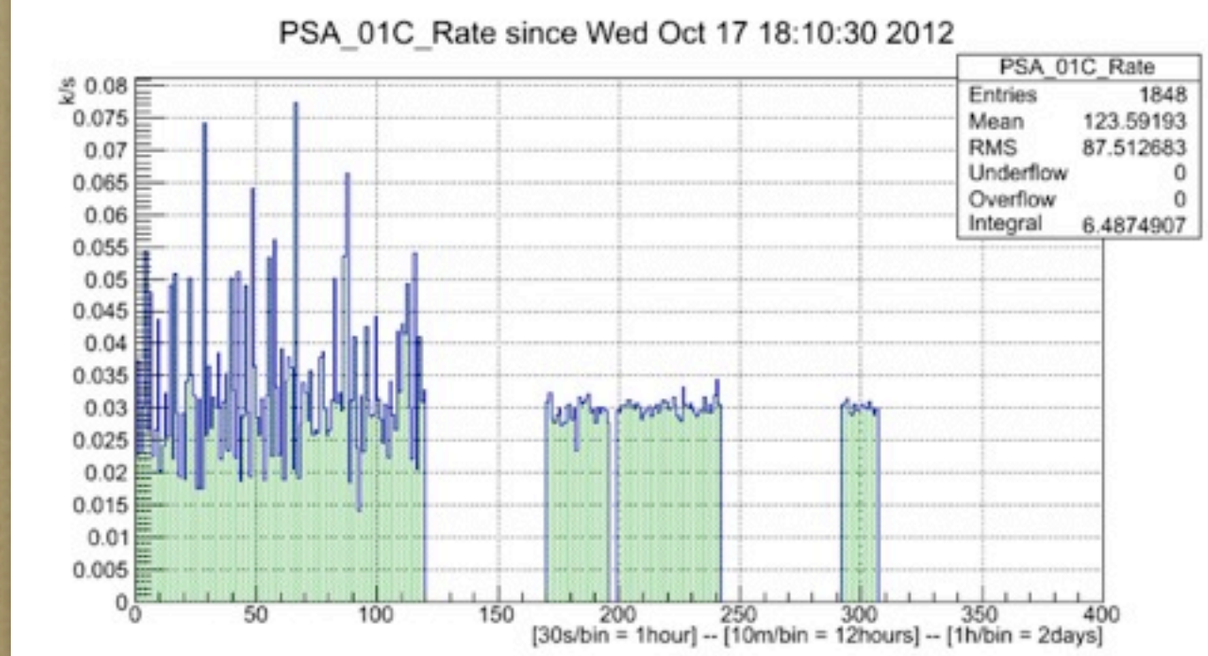
E x T matrices

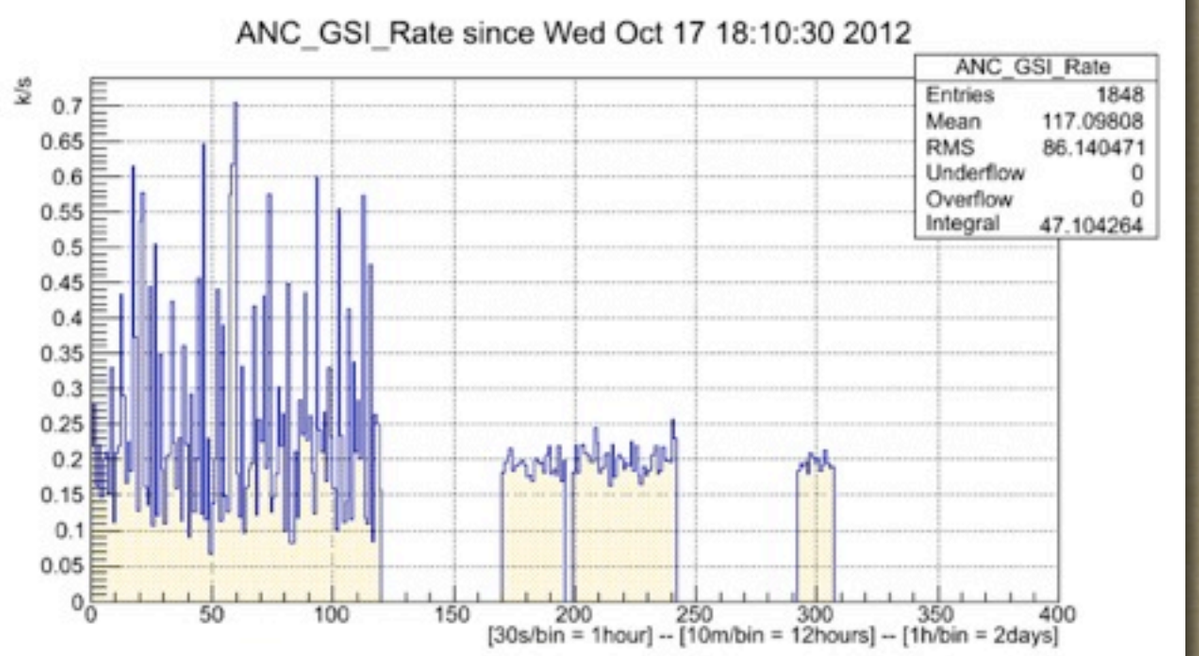
Watchers[Gw] @GSI LLP

Full monitoring of rates history, any kind of frames



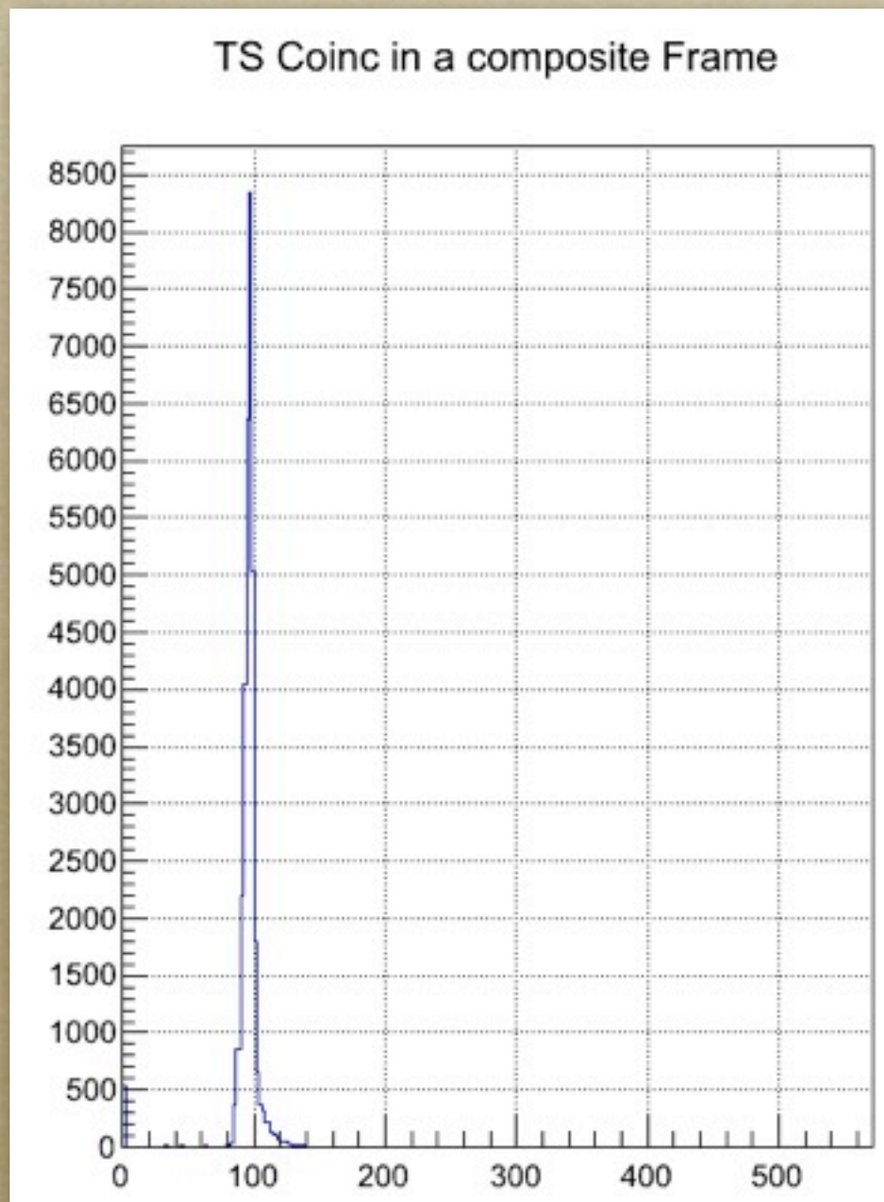


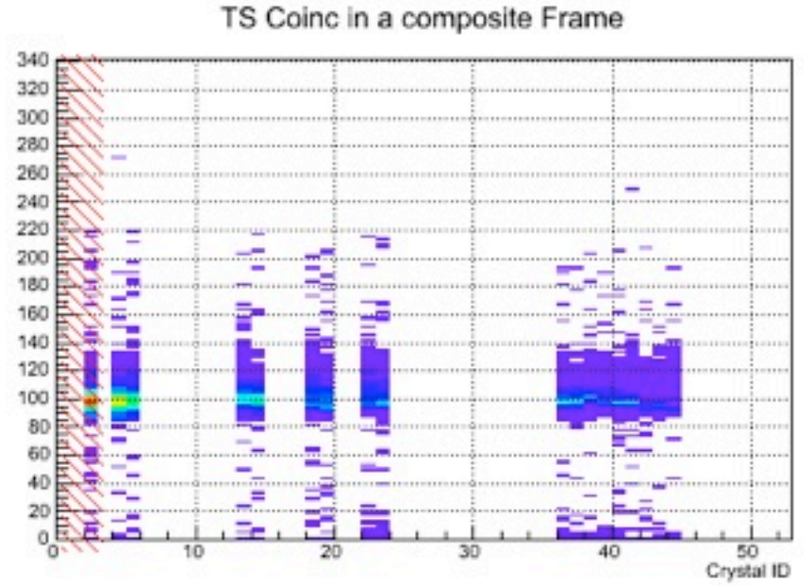


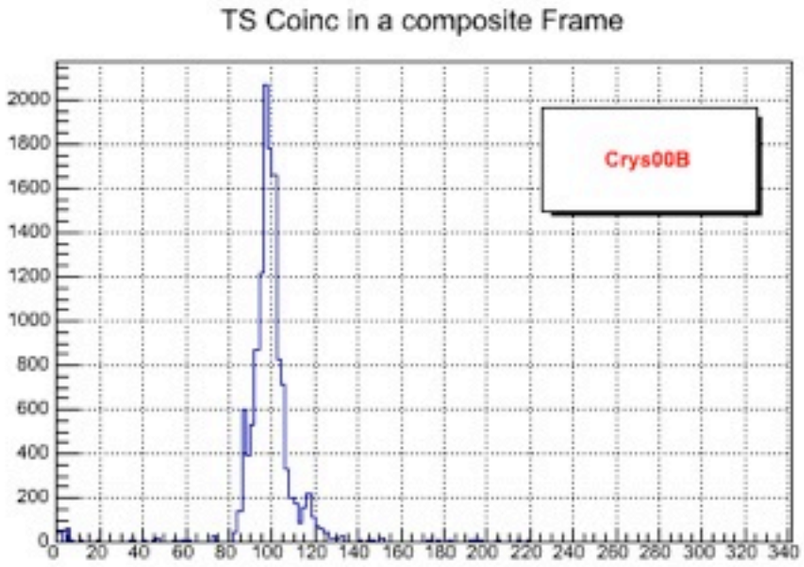


Watchers[Gw]@GSI GLP

on event:data

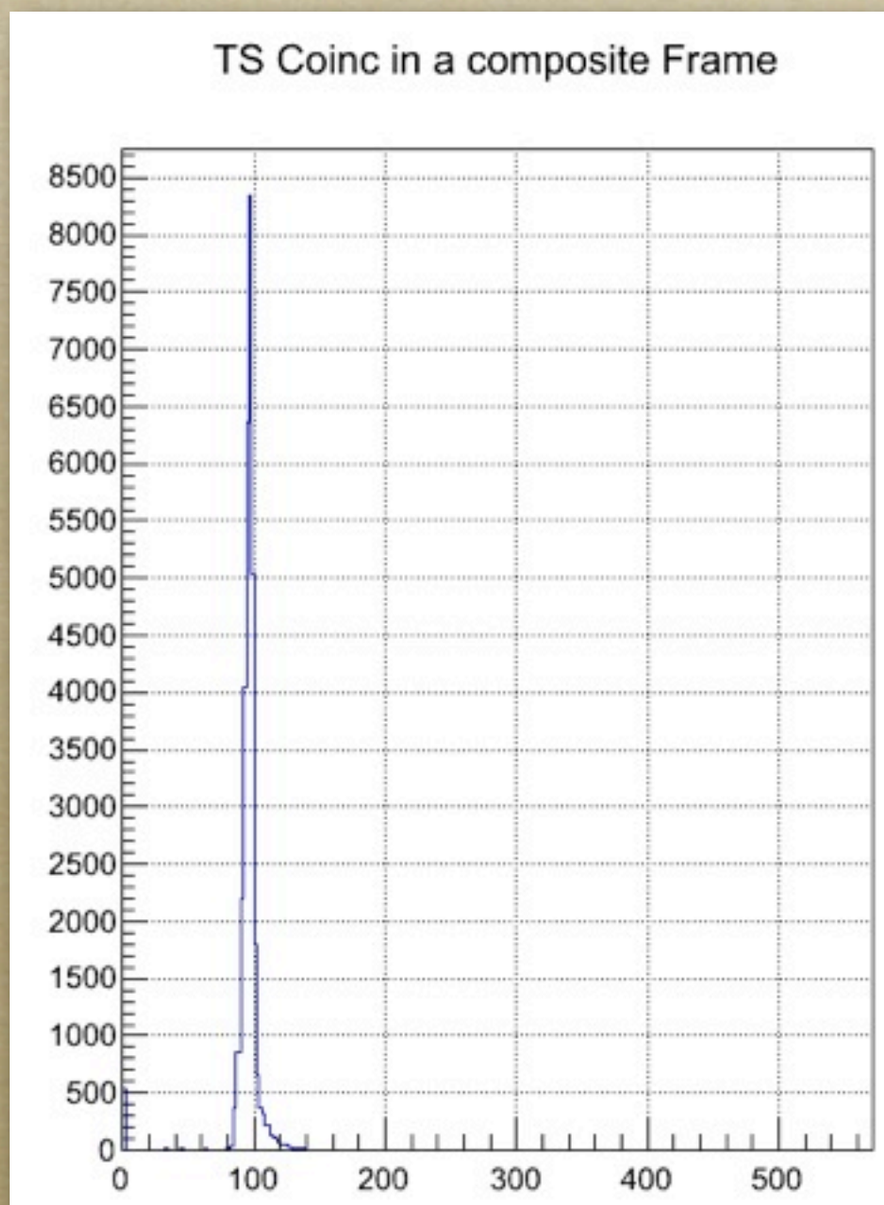


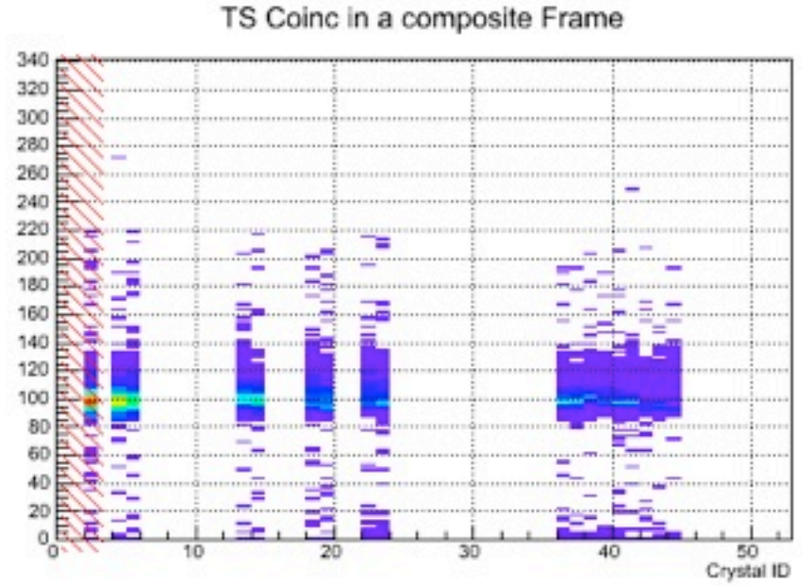


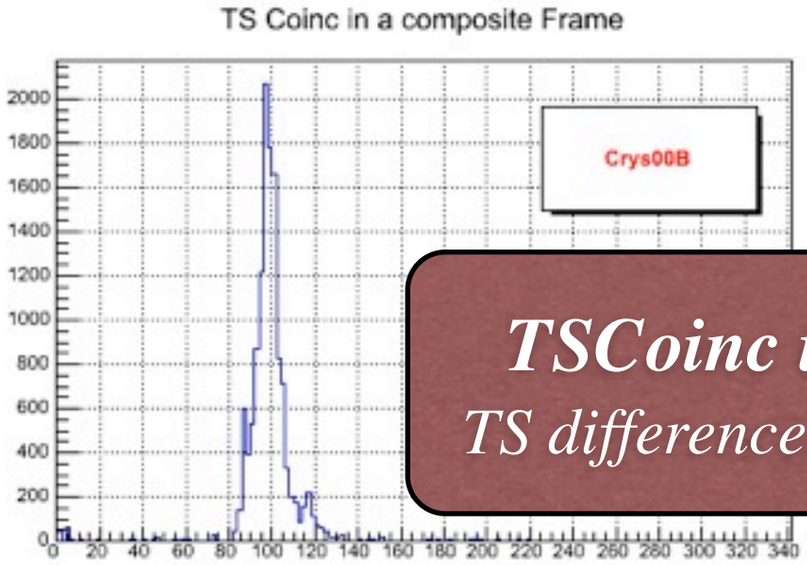


Watchers[Gw]@GSI GLP

on event:data





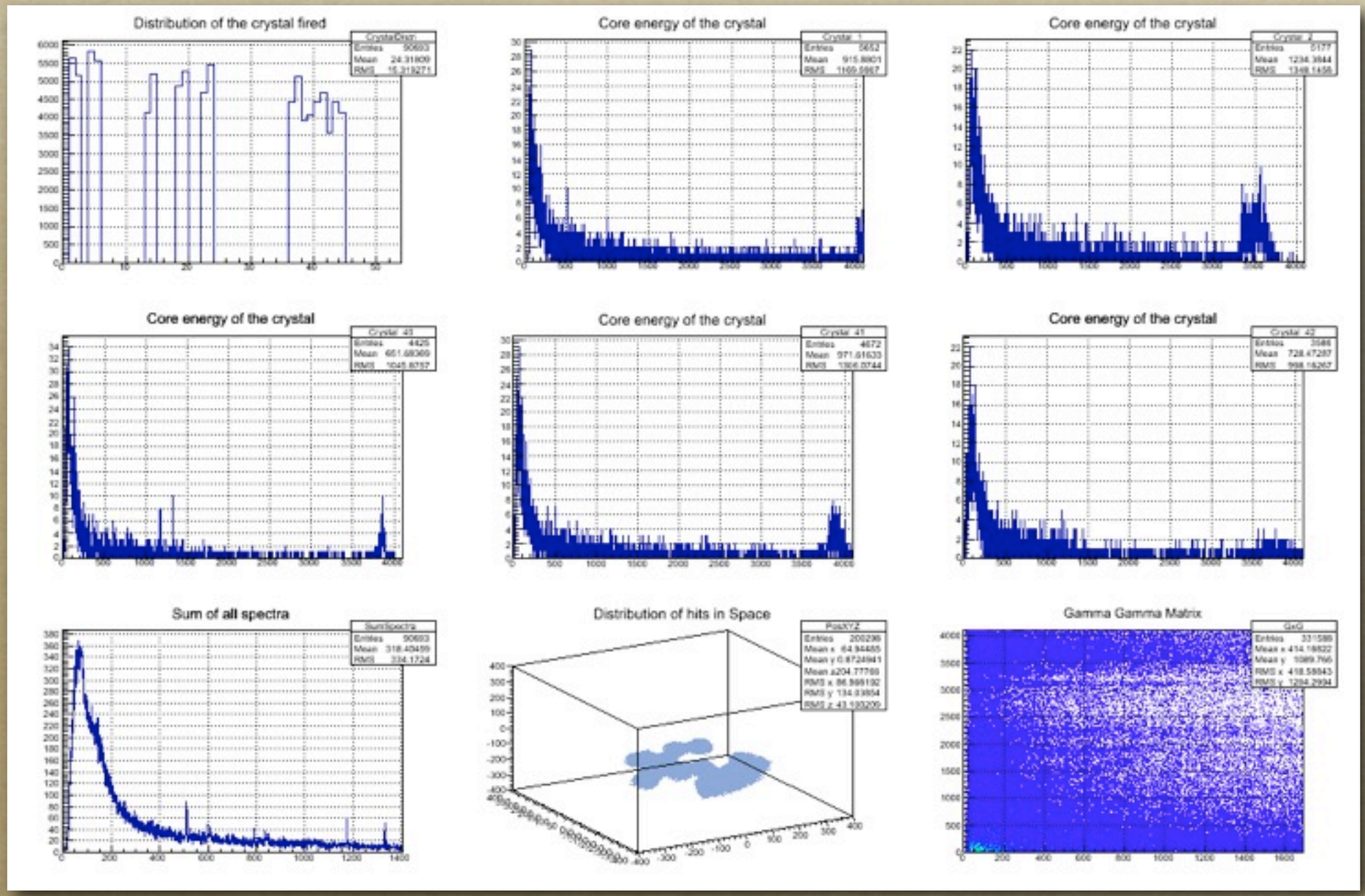


TSCoinc improved

TS differences per crystal

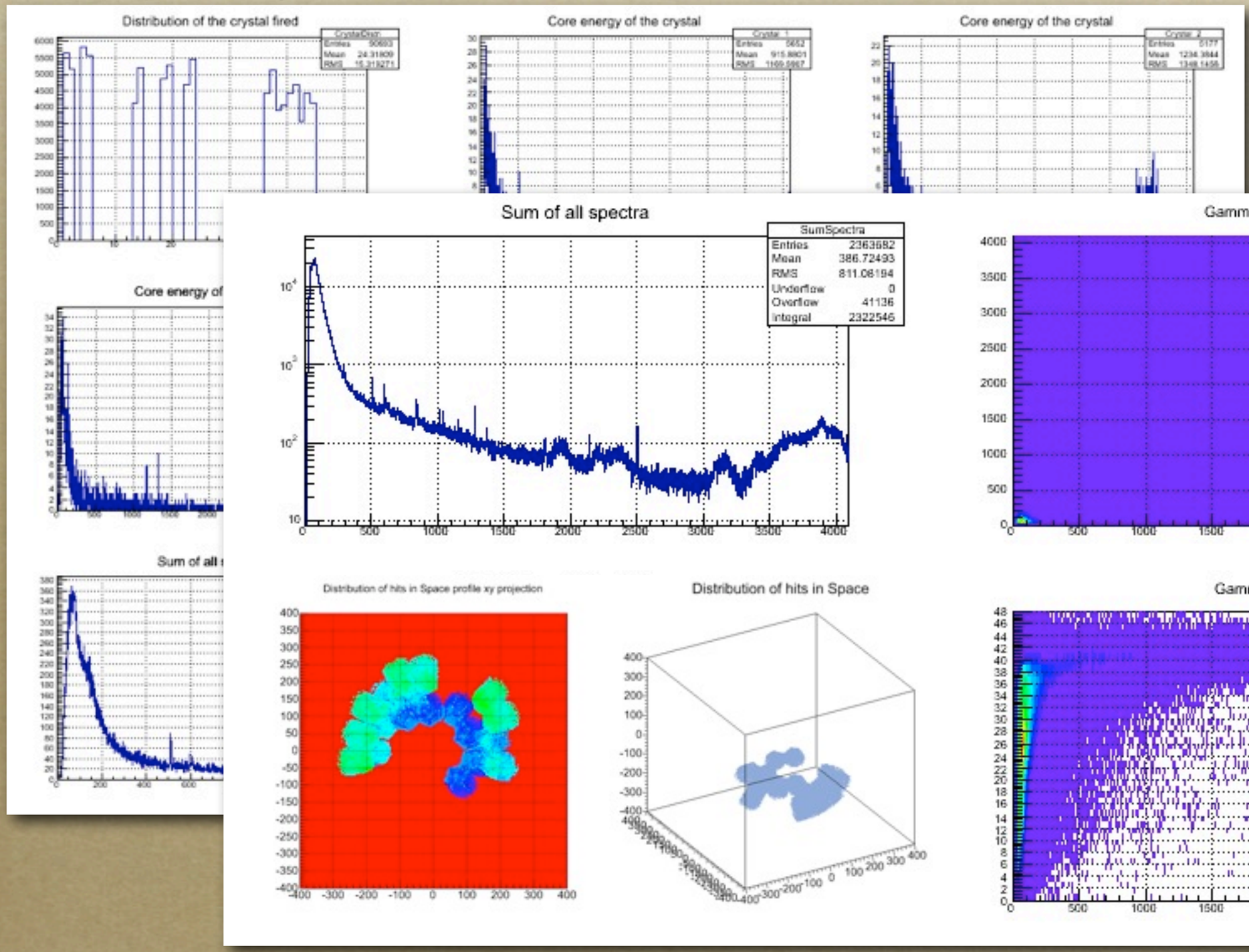
Watchers[Gw]@GSI GLP

on event:data:psa



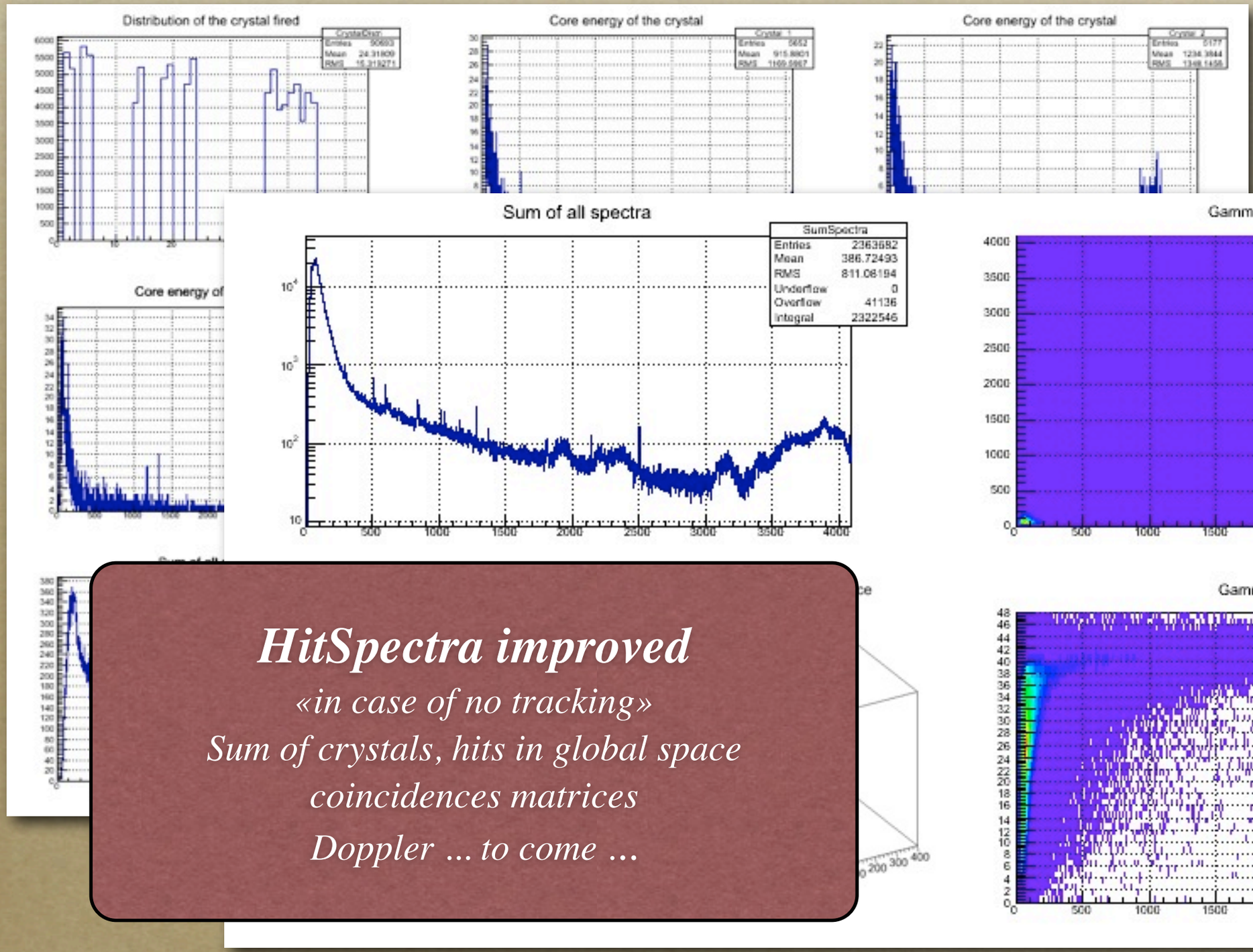
Watchers[Gw]@GSI GLP

on event:data:psa



Watchers[Gw]@GSI GLP

on event:data:psa



HitSpectra improved

«in case of no tracking»

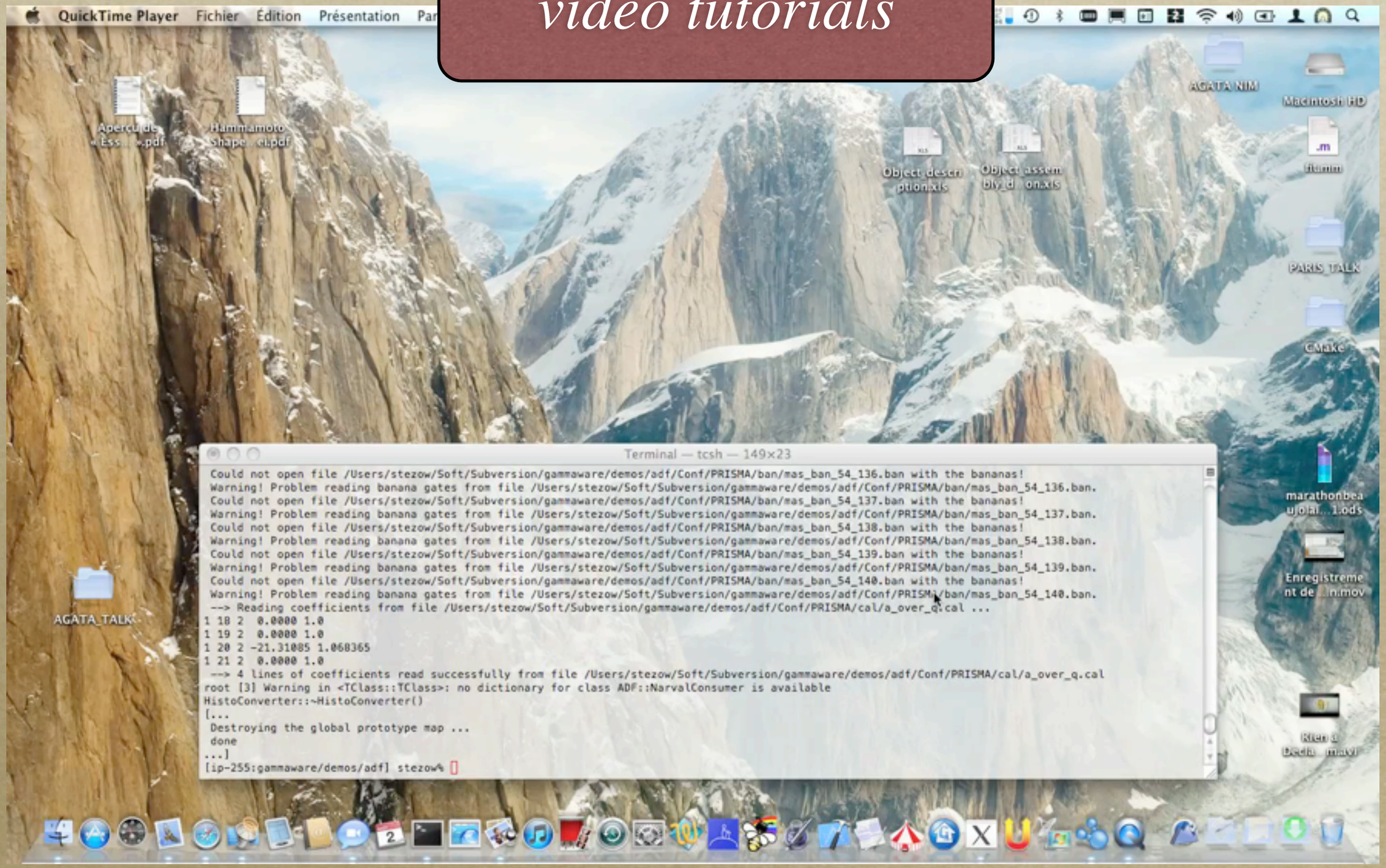
Sum of crystals, hits in global space

coincidences matrices

Doppler … to come …

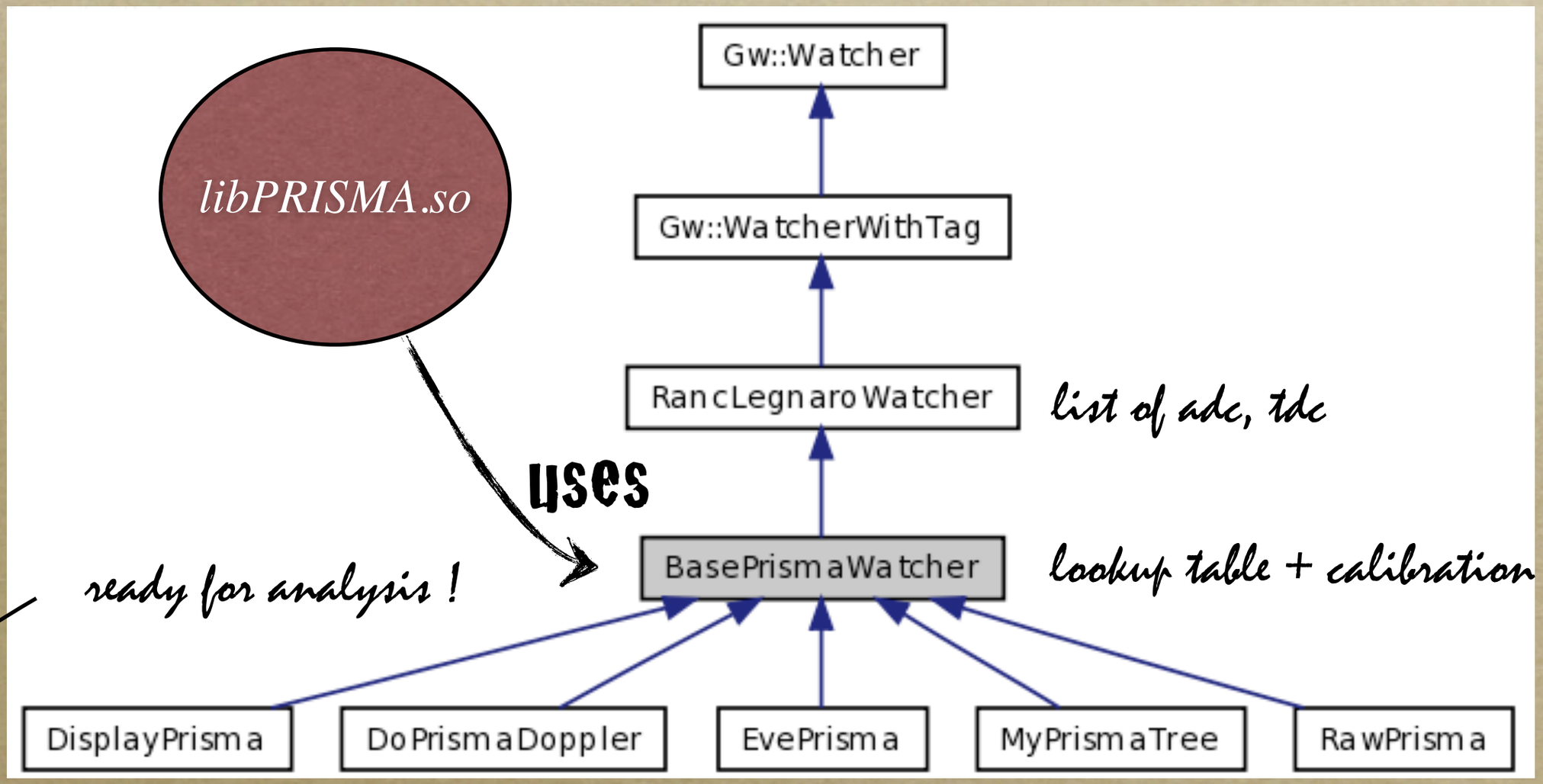
Watchers[Gw]

video tutorials



Watchers[Gw]

Inheritance : base class, service for daughter classes



Watchers[Gw]

constructor : build spectra

exec : fill spectra

```
DisplayPrisma::DisplayPrisma(const char *name, const char *title): BasePrismaWatcher(name, title)
{
   IcABvsABCD       = new TH2F("Ic_ABvsABCD", "E_DeltaE_AB", 1024,0,12000,2048,0,8000);
   AddToPool(IcABvsABCD);
   IcAvsABCD        = new TH2F("Ic_AvsABCD", "E_DeltaE_A", 1024,0,12000,2048,0,8000);
   AddToPool(IcAvsABCD);
   TOF_D            = new TH2F("ToF(ns)vsX_FP(mm)", "ToFvsx_FP", 1024,0,1024,4096,0,4096);
   AddToPool(TOF_D);
...
}

void DisplayPrisma::Exec(Option_t *option)
{
   const double  c_light = 29.97295 * cm/ns;

   // fill the array of floats and compute the PRISMA physics
   SetLastError(0u);
   BasePrismaWatcher::Exec(option);
   if ( GetLastError() > 0 )
          return;

   if ( fPM->length() > 0. )  {

          Vector3D Vel = fPM->velocity()/c_light;

          IcABvsABCD    -> Fill(fPM->ic_energy(),fPM->ic_energy_AB_DE());
          TOF_D         -> Fill(fPM->x_fp() /mm, 10*fPM->tof()/ns);
          PRISMA_MCPcal -> Fill(fPM->mcp_x()/mm , fPM->mcp_y()/mm) ;
          PRISMA_X_FP   -> Fill( fPM->x_fp() /mm) ;
          PRISMA_Vel    -> Fill( Vel.rho() ) ;
          Range_Energy->Fill(fPM->range(), fPM->ic_energy()) ;
          Vel_Theta->Fill(10.*fPM->theta_c()/degree, Vel.rho() );
          a_over_q->Fill(fPM->x_fp()/mm, fPM->a_over_q_uncal());
          Energy_RBeta->Fill(fPM->r_beta(),fPM->ic_energy());
          PRISMA_Mass->Fill(fPM->Mass());
   }
}
```

just to allow global actions (zero, save, delete ...)

Watchers[Gw]

constructor : build spectra

```
DisplayPrisma::DisplayPrisma(const char *name, const char *title): BasePrismaWatcher(name, title)
{
    fSumSpectra = MakeTH1<TH1F>("SumSpectra","Sum of all spectra",4096,0,4096);
    fGxG    = MakeTH2<TH2F>("GxG","Gamma Gamma Matrix",4096,0,4096,4096,0,4096);
    fDistri = MakeTH1<TH1F>("CrystalDistri","Distribution of the crystal fired",180,0,180);
    fPosXYZ = MakeTH3<TH3F>("PosXYZ","Distribution of hits in Space",200,-400,400,200,-400,400,200,-400,400);
    fGxT = MakeTH2<TH2F>("GxT","Gamma Time Matrix",4096,0,4096,300,0,300);

}
```

just to allow global actions (zero, save, delete ...)

exec : fill spectra

```
void DisplayPrisma::Exec(Option_t *option)
{
   const double  c_light = 29.97295 * cm/ns;

   // fill the array of floats and compute the PRISMA physics
   SetLastError(0u);
   BasePrismaWatcher::Exec(option);
   if ( GetLastError() > 0 )
           return;

   if ( fPM->length() > 0. )  {

           Vector3D Vel = fPM->velocity()/c_light;

           IcABvsABCD    -> Fill(fPM->ic_energy(),fPM->ic_energy_AB_DE());
           TOF_D         -> Fill(fPM->x_fp() /mm, 10*fPM->tof()/ns);
           PRISMA_MCPcal -> Fill(fPM->mcp_x()/mm , fPM->mcp_y()/mm) ;
           PRISMA_X_FP   -> Fill( fPM->x_fp() /mm) ;
           PRISMA_Vel    -> Fill( Vel.rho() ) ;
           Range_Energy->Fill(fPM->range(), fPM->ic_energy()) ;
           Vel_Theta->Fill(10.*fPM->theta_c()/degree, Vel.rho() );
           a_over_q->Fill(fPM->x_fp()/mm, fPM->a_over_q_uncal());
           Energy_RBeta->Fill(fPM->r_beta(),fPM->ic_energy());
           PRISMA_Mass->Fill(fPM->Mass());
   }
}
```

Overview

Basic elements

no standard analysis

➥ *always new spectrum, cuts*

Improvement of existing actors

New actors (GSI) !

What already exists

How to extend

Why ?

Overview

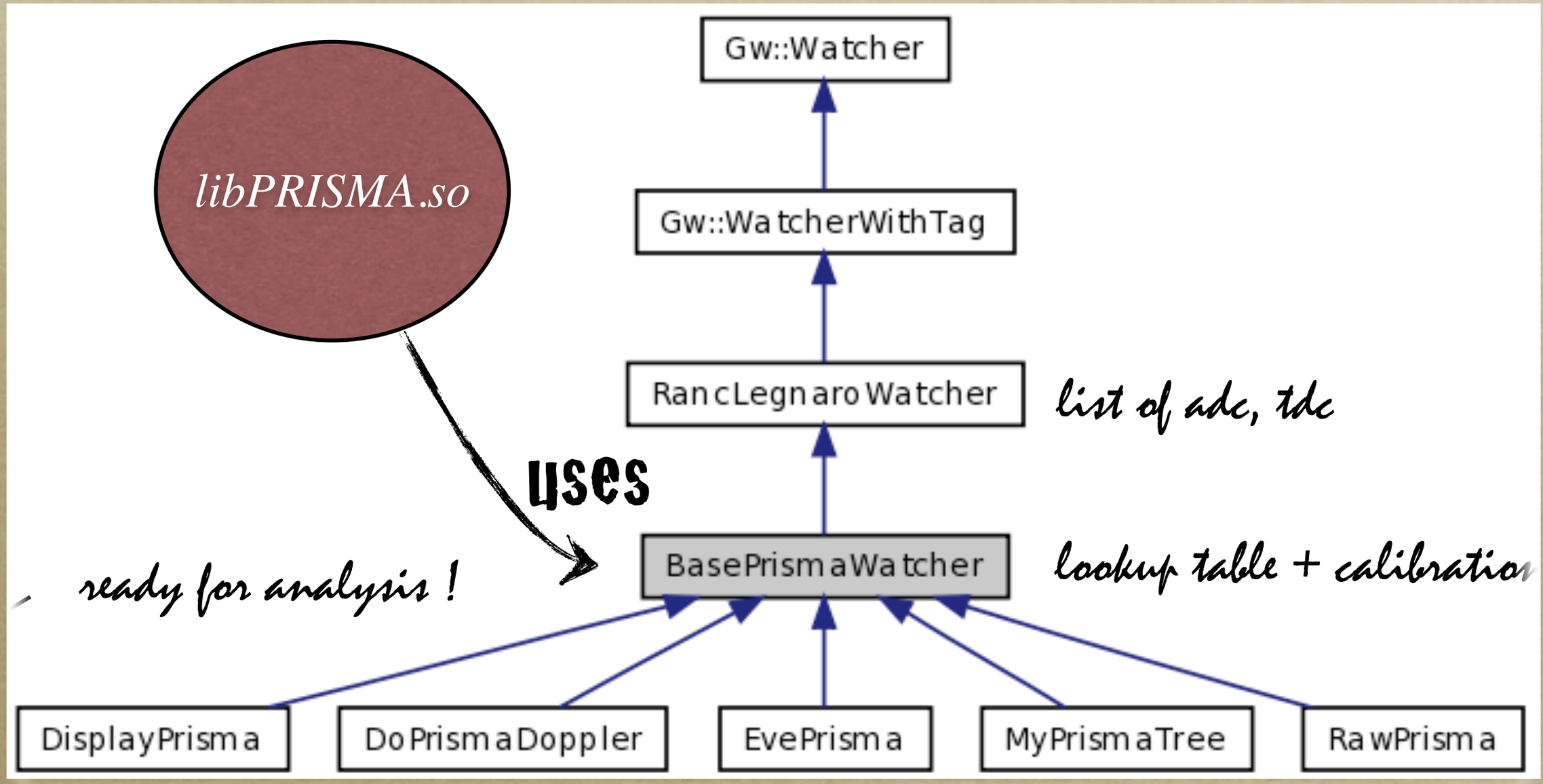
New Watcher

How to extend

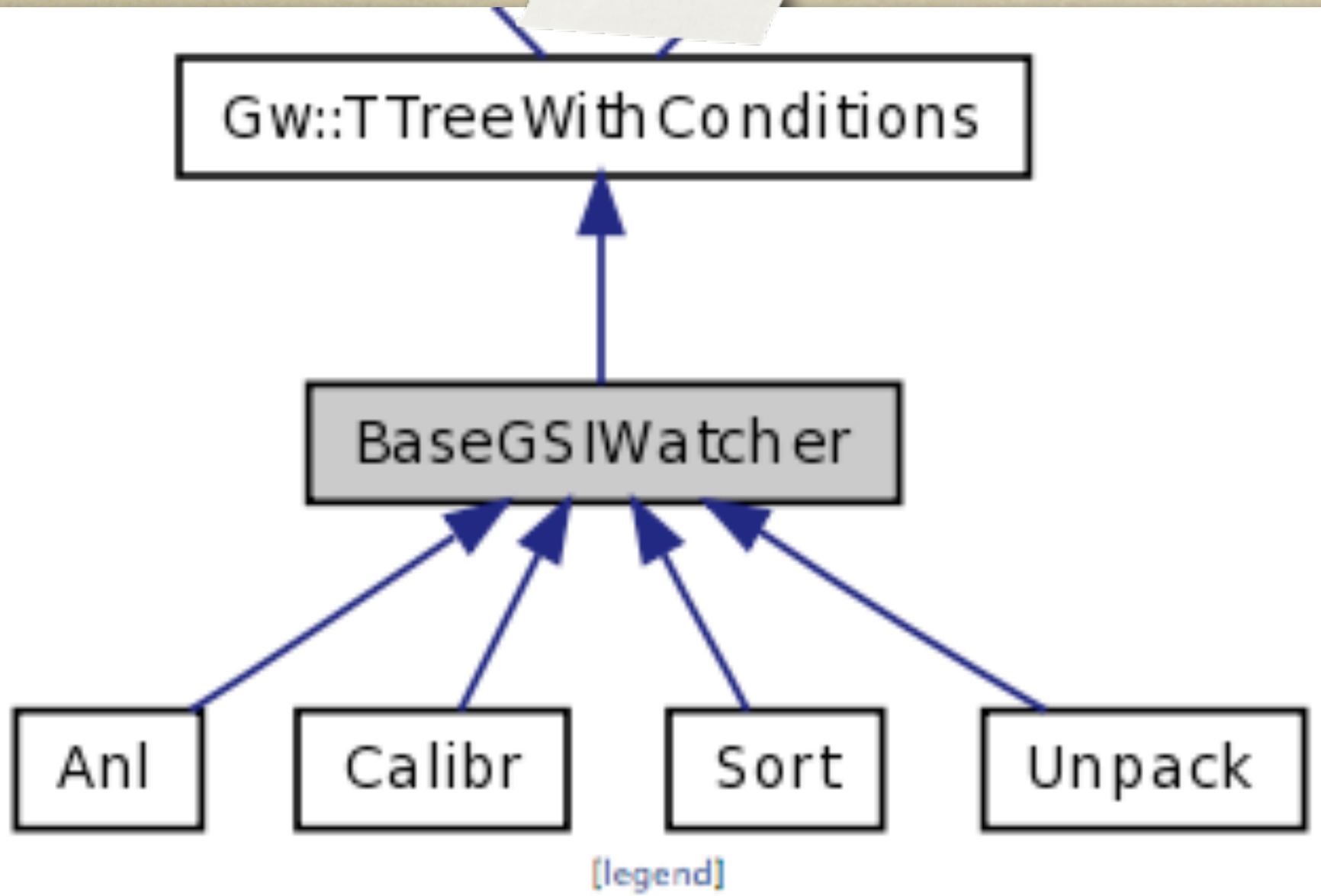
About replay

New Actor

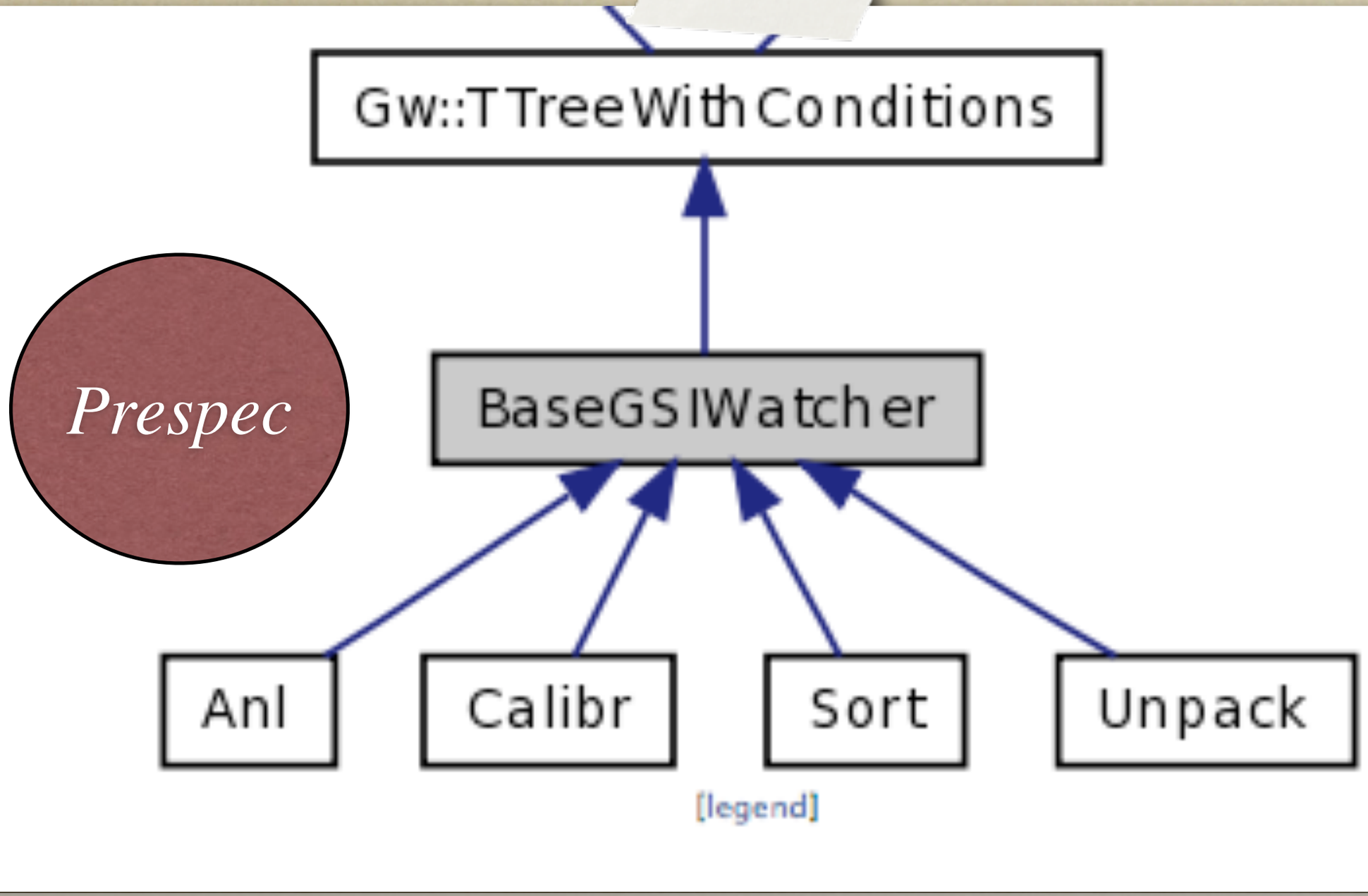
Watchers[Gw] @GSI



Watchers[Gw] @GSI



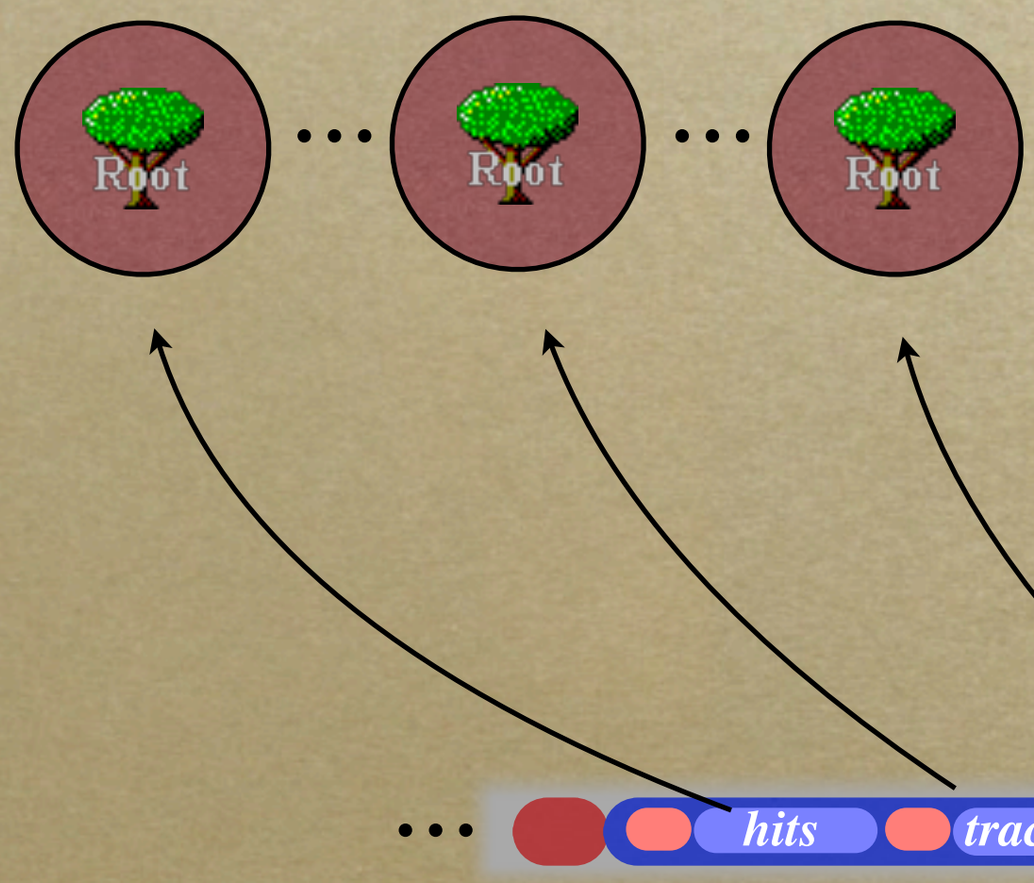
Watchers[Gw] @GSI



Watchers[Gw]

Watchers could be chained, build ROOT Tree

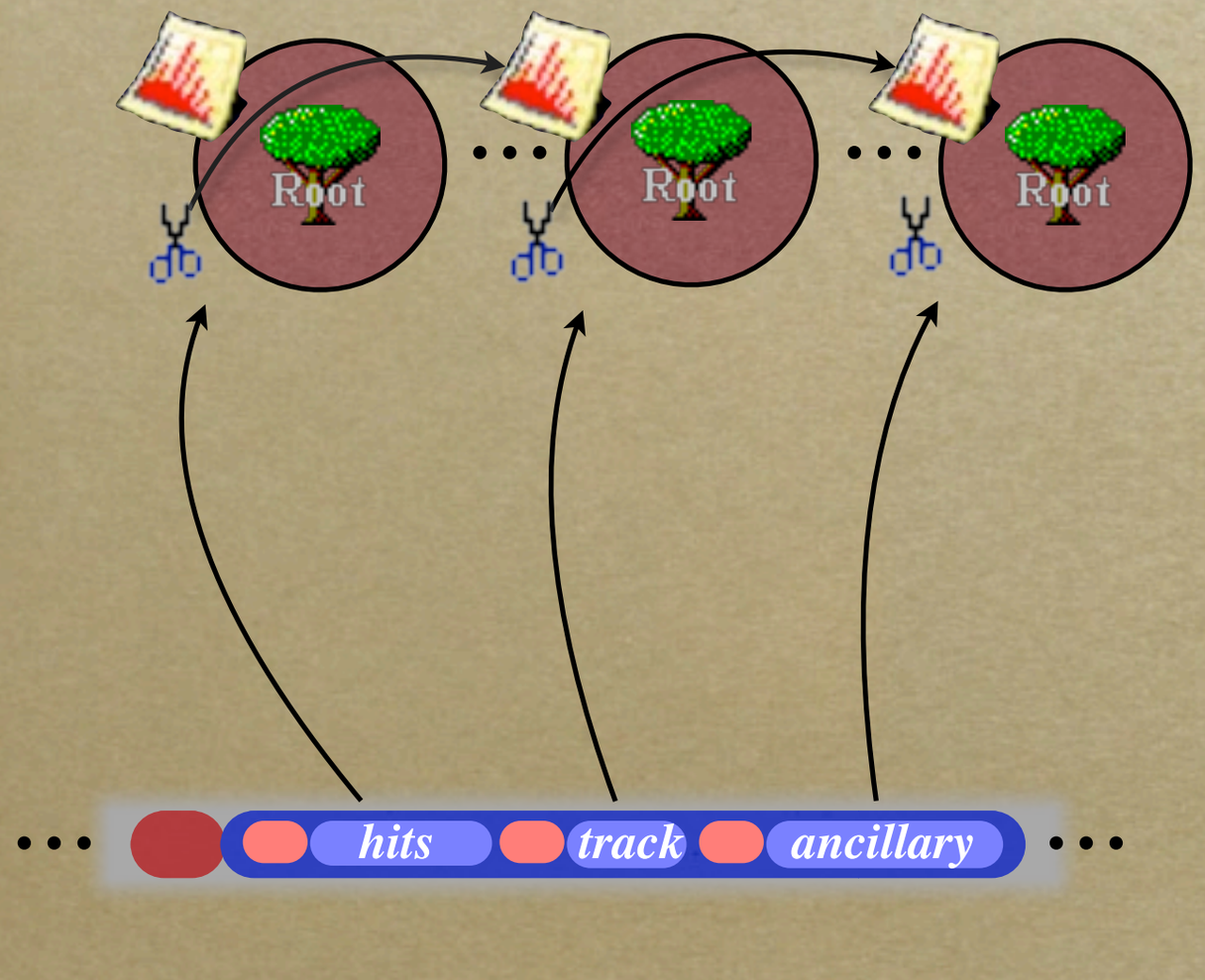
Watcher1 *Watcher2* *Watcher3*



Already the case @ Legnaro

Watchers[Gw] @GSI

Watchers could be chained, build ROOT Tree



<u>*NEW*</u> :

- *spectra*
- *conditions*

'à la GO4' Graph. cond. 1D/2D
*Expression 'à la root' x*y > 10*

- *event list*

E# : 0 1 2 3 4 5 ...
w1 : 1 0 1 0 0 1 ...
w2 : 0 0 1 0 1 1 ...
w3 : 1 0 1 0 1 0 ...

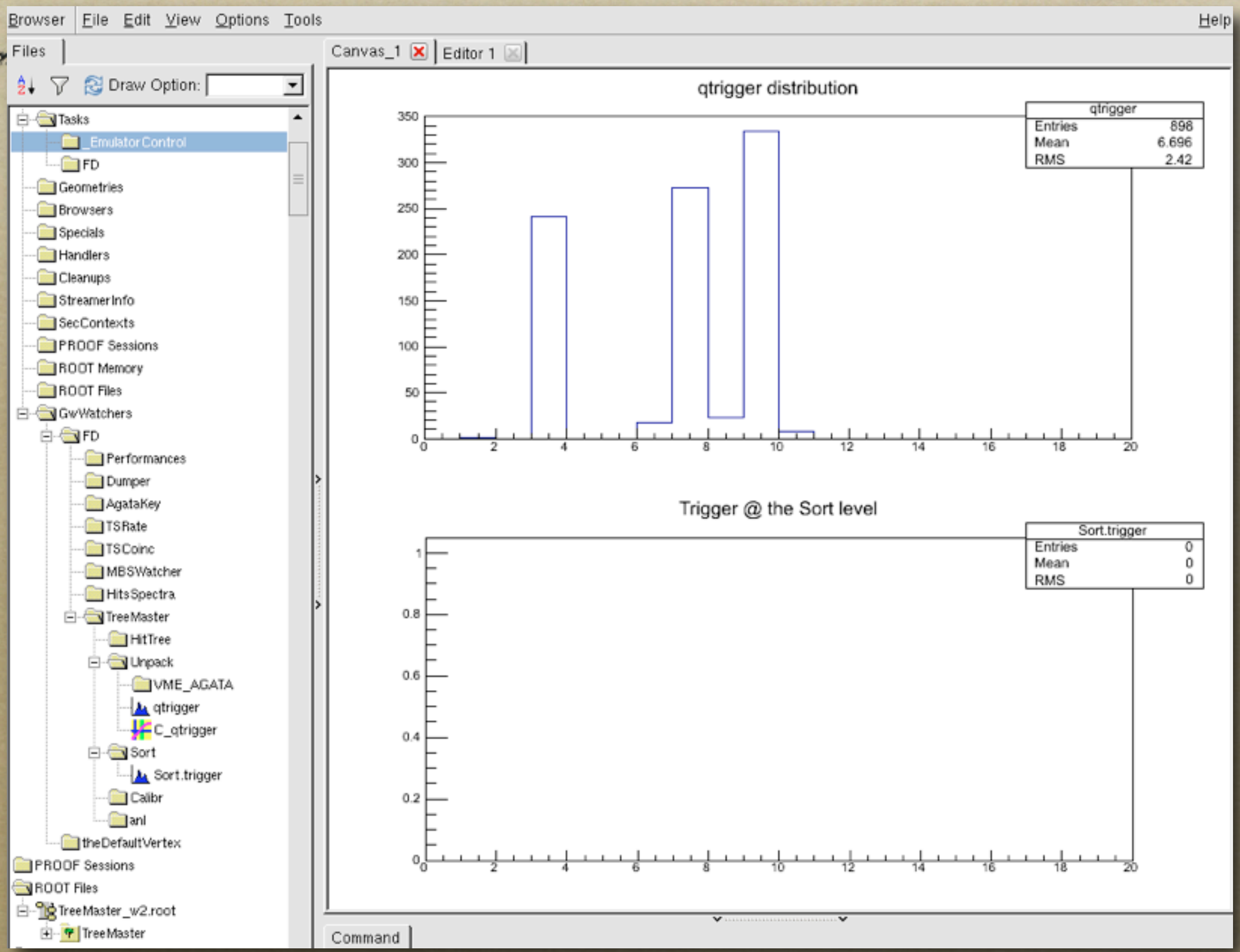
➥ ***<u>Speed up next step analysis !</u>***

Watchers[Gw] @GSI

Control

Spectra & conditions

Trees

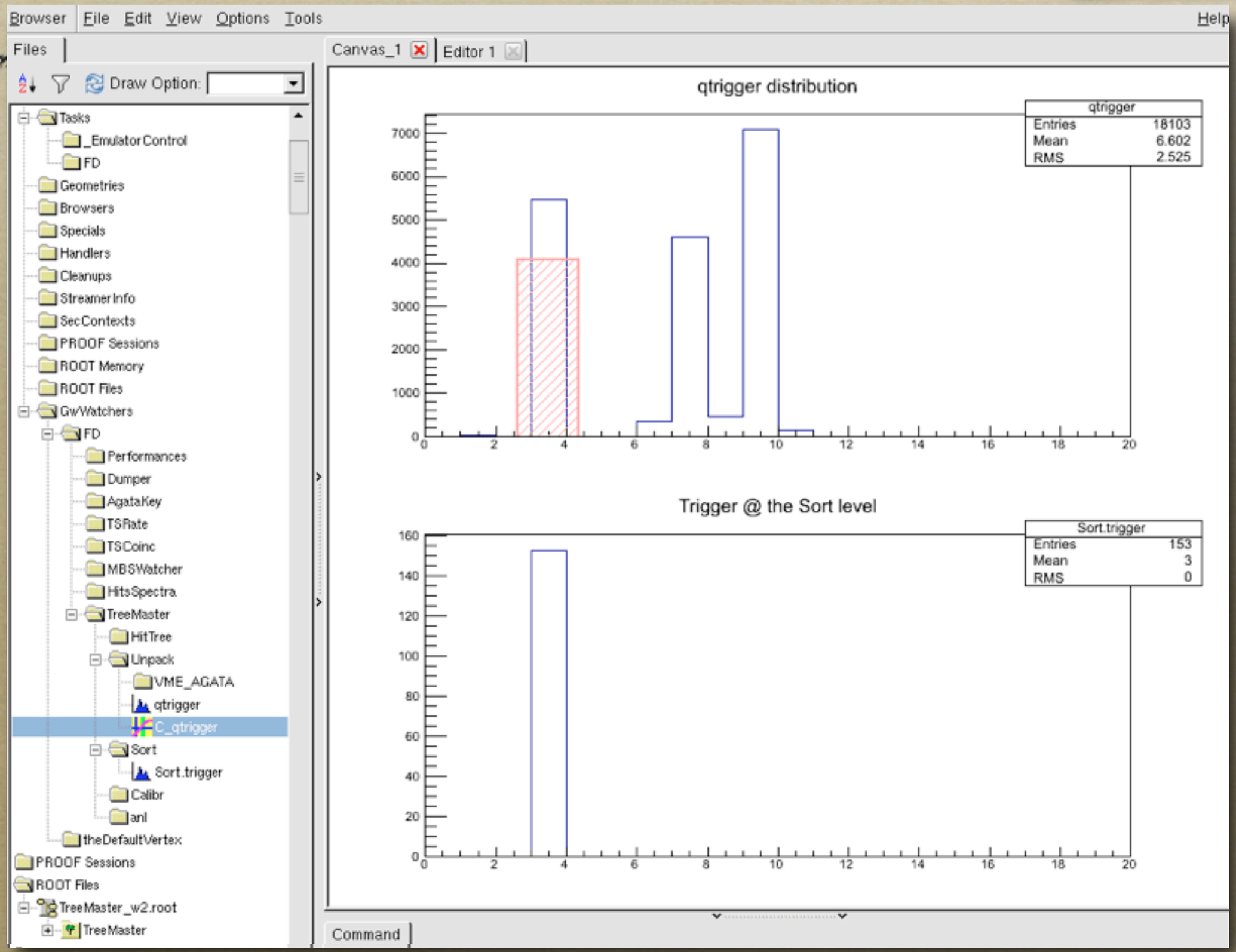


Watchers[Gw] @GSI

Control

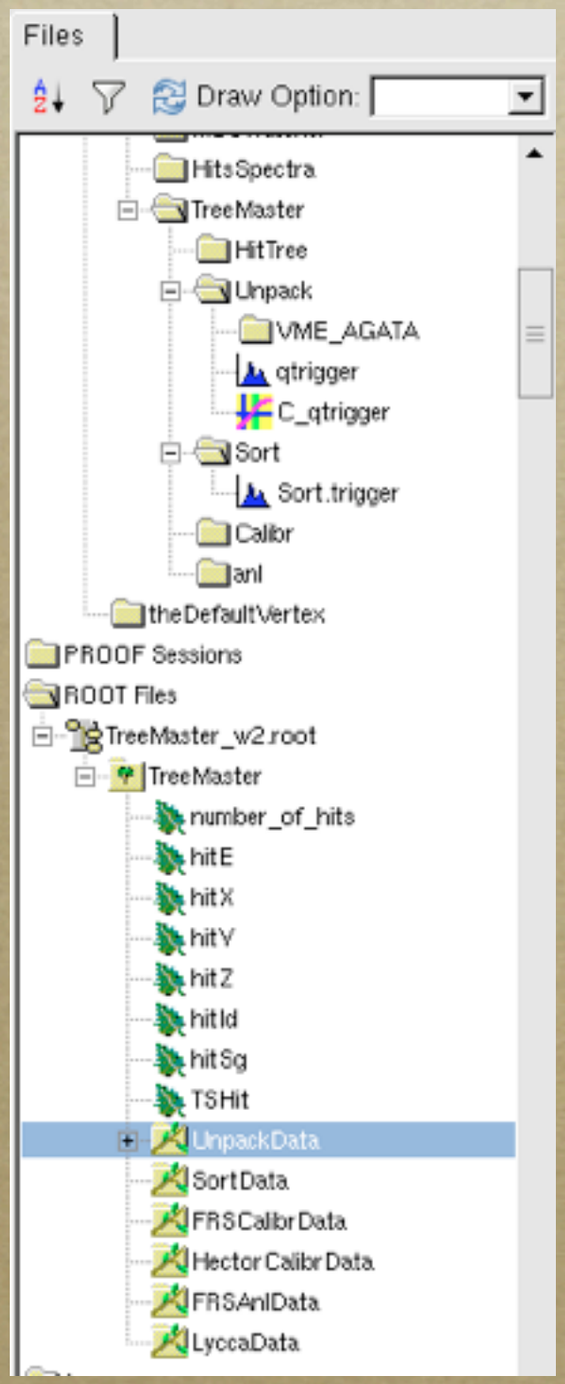
Spectra & conditions

Trees



Watchers[Gw] @GSI

Trees



Watchers[Gw] @GSI

Files

TreeViewer

File Edit Run Options Help

Command | Option | Histogram htemp | Hist | Scan | Rec

Current Folder

- TreeList
- TreeMaster
  - number_of_hits
  - hitE
  - hitX
  - hitV
  - hitZ
  - hitId
  - hitSg
  - TSHit
  - UnpackData
  - SortData
  - FRSCalibrData
  - HectorCalibrData
  - FRSAnlData
  - LyccaData
    - t_DSSD_EcalP[]
    - t_DSSD_EcalN[]
    - t_DSSD_TcalP[]
    - t_DSSD_x
    - t_DSSD_y
    - t_DSSD_CalE_P
    - t_DSSD_CalE_N

SPIDER

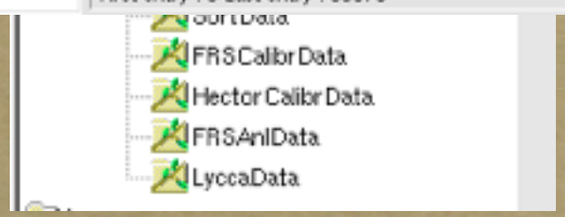
Current Tree : TreeMaster

X : -empty-	hitSg	vme11s[][]	SISMusCtrID	lenTDC86	timestamp
Y : -empty-	TSHit	nhit11[]	vme60_e[][]	lenSIS88	timespill
Z : -empty-	UnpackData	vme12[][]	vme60_t[][]	qlength	pattern
-empty-	detBP[]	vme12ov[]	vme60_errcode[]	qtype	trigger
Scan box	detHP	vme12s[][][]	vme60_npulses[]	qsubtype	sc_long[]
-empty-	orNum[]	nhit2[][]	vme60_mode[]	qdummy	sc_long2[]
-empty-	vme0[][]	vme13[][]	vme60_sis_eventnumber[]	qtrigger	mw_an[]
-empty-	vme0ov[]	vme13ov[]	vme60_tracesize[]	qevent_nr	mw_xl[]
-empty-	vme1[][]	vme13s[][][]	vme32_0[]	agavastatus	mw_xr[]
-empty-	vme1ov[]	nhit13[][]	vme32_1[]	agava	mw_yu[]
-empty-	vme2[][]	vmeAGs[][]	lenFRS	LyccaTCtrID	mw_yd[]
-empty-	vme2s[][]	nhitAG[]	lenTPC	pi_flg	tpc_l[][]
-empty-	nhit5[]	AgataTCtrId	lenUSR	sis[]	tpc_r[][]
-empty-	vme3[][]	Ly0CtrID	lenLy0	tsis[][]	tpc_lt[][]
-empty-	HecCtrID	Ly1CtrID	lenLy1	vme88[]	tpc_rt[][]
number_of_hits	vme10[][]	Ly2CtrID	lenLy2	energyArray[]	mh_tpc_l[][]
hitE	vme10ov[]	vme14[]	lenLyT	lenANTI	mh_tpc_r[][]
hitX	vme10s[][][]	vme14ov	lenTRLO	vme89[]	tpc_grid_signal[]
hitV	nhit10[][]	vme14s[][]	lenFinger	SortData	tpc_dt[][]
hitZ	vme11[][]	nhit4[]	lenHector	ts_id	tpc_a[][]
hitId	vme11ov[]	FingerCtrID	lenScal88	ts_word[]	mh_user_reference

0%

IList | OList | First entry : 0 Last entry : 58575 | RESET

Trees



Watchers[Gw] @GSI

ToDo List :

- *debug ... ?*
- *define some basics cuts / spectra*
- *save / import / export (go4) of cuts*
- *list of events*
- *add the doppler correction*
- ...

Overview

New Watcher

How to extend

About replay

New Actor

Overview

New Watcher

How to extend

About replay

New Actor

About Replay

It depends from what to what !

Tracking from hits (precedent slide) : femul, gw
Pb of Event builder / merger hits ➔ end : femul

From traces
one detector : femul, gw
all of them : Narval or femul (slow ...)

Another ('future') solution : the GRID

tests have been done but not yet fully available for the community

About Replay

It depends from what to what !

Tracking from hits (precedent slide) : femul, gw
Pb of Event builder / merger hits ➙ end : femul

From traces
one detector : femul, gw
all of them : Narval or femul (slow ...)

Another ('future') solution : the GRID

tests have been done but not yet fully available for the community

➥ *The future is there !*