AGATA-GRETINA Processing and Neural Networks

O. STEZOWSKI - 4-6 April 2018 - Orsay

Institut de Physique Nucléaire de Lyon

BACK ONE YEAR & A HALF AGO

- Extensive presentation of the data { Processing (NARVAL/DCOD - Online) / Re-Processing (emulators - offline)

- Should be flexible enough to allow processing of GRETINA data (algorithms)
  - ☛ It should be interesting to follow Amel & Torben's work done on Tracking
  - ☛ but the other way around i.e. GRETINA through AGATA

- Solutions to speed up complex / more complex algorithms
  - ☛ hpx* could be a solution

** http://stellar.cct.lsu.edu/projects/hpx/*

BACK ONE YEAR & A HALF AGO

- Extensive presentation of the data { Processing (NARVAL/DCOD - Online) / Re-Processing (emulators - offline)

  ➤ **TODAY : Just a quick reminder**

- Should be flexible enough to allow processing of GRETINA data (algorithms)

  ☛ It should be interesting to follow Amel & Torben's work done on Tracking

  ☛ but the other way around i.e. GRETINA through AGATA

  ➤ **TODAY : what has been done so far, about to be done**

- Solutions to speed up complex / more complex algorithms

  ☛ hpx* could be a solution

  ➤ **TODAY : other solution under investigation since September**

** http://stellar.cct.lsu.edu/projects/hpx/*

BACK ONE YEAR & A HALF AGO

- Extensive presentation of the data { Processing (NARVAL/DCOD - Online) / Re-Processing (emulators - offline)

  ➤ **TODAY : Just a quick reminder**

- Should be flexible enough to allow processing of GRETINA data (algorithms)

  ☛ It should be interesting to follow Amel & Torben's work done on Tracking

  ☛ but the other way around i.e. GRETINA through AGATA

  ➤ **TODAY : what has been done so far, about to be done**

- Solutions to speed up complex / more complex algorithms

  ☛ hpx* could be a solution

  ➤ **TODAY : other solution under investigation since September**

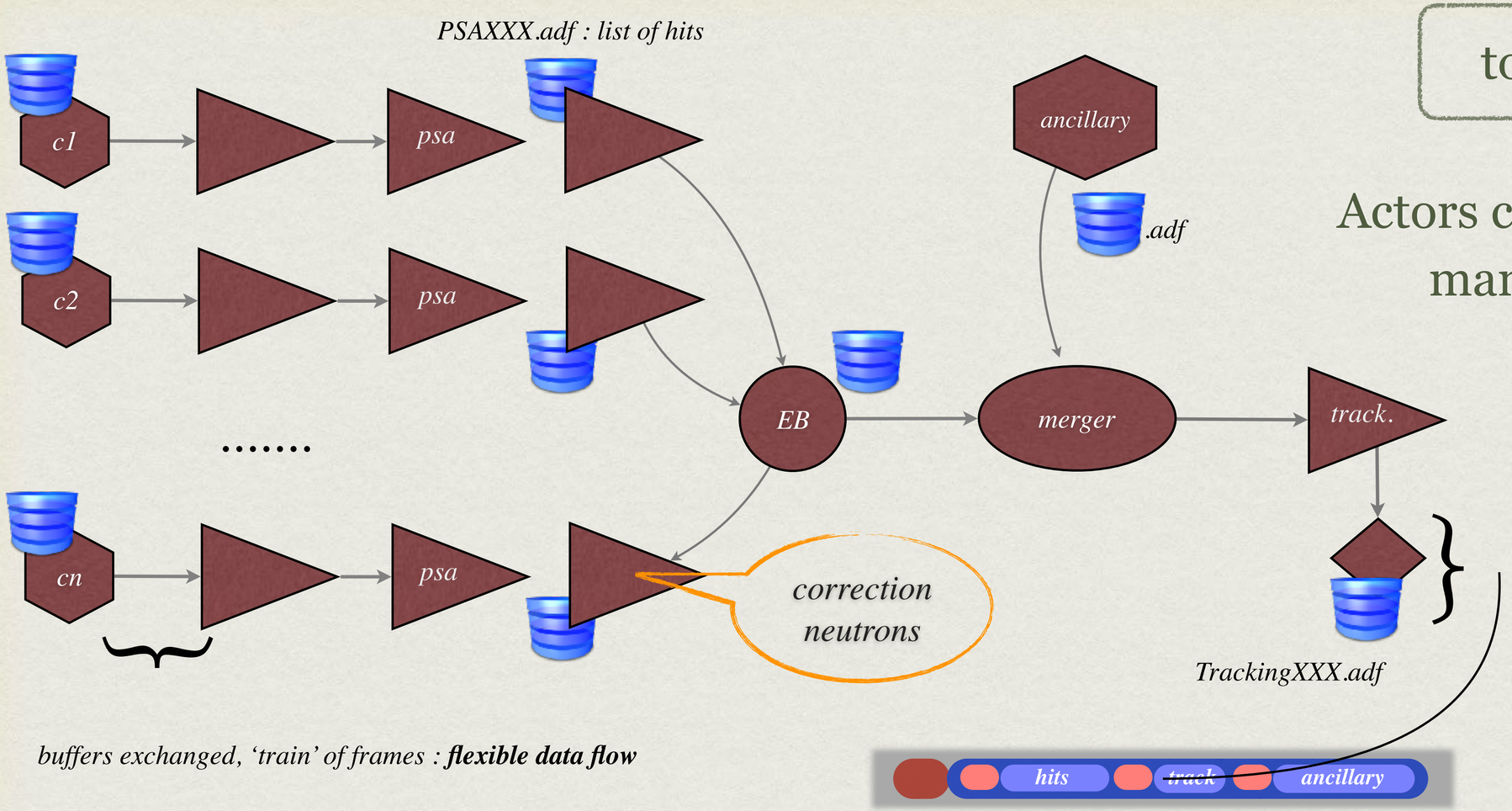
JUST A QUICK REMINDER

The AGATA data flow is structured in terms of { Actors, Exchanges of buffers of frames

buffers exchanged, 'train' of frames : ***flexible data flow***

- Some actors are written in ADA (online mainly)
- Some actors are written in C++ (online/offline)

JUST A QUICK REMINDER



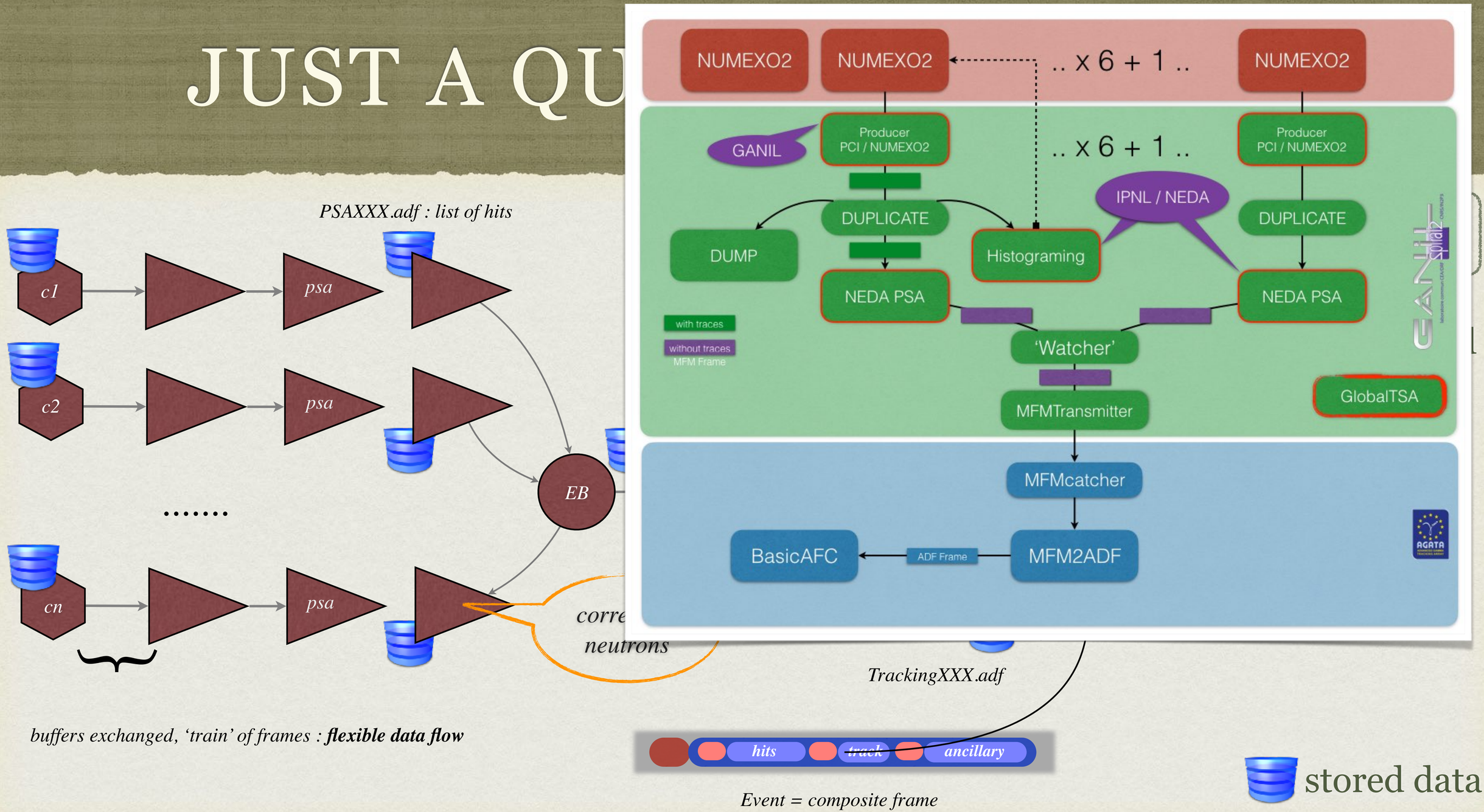
topology with ancillary

Actors could be removed/added
many different ‘replays’

*buffers exchanged, ‘train’ of frames : **flexible data flow***

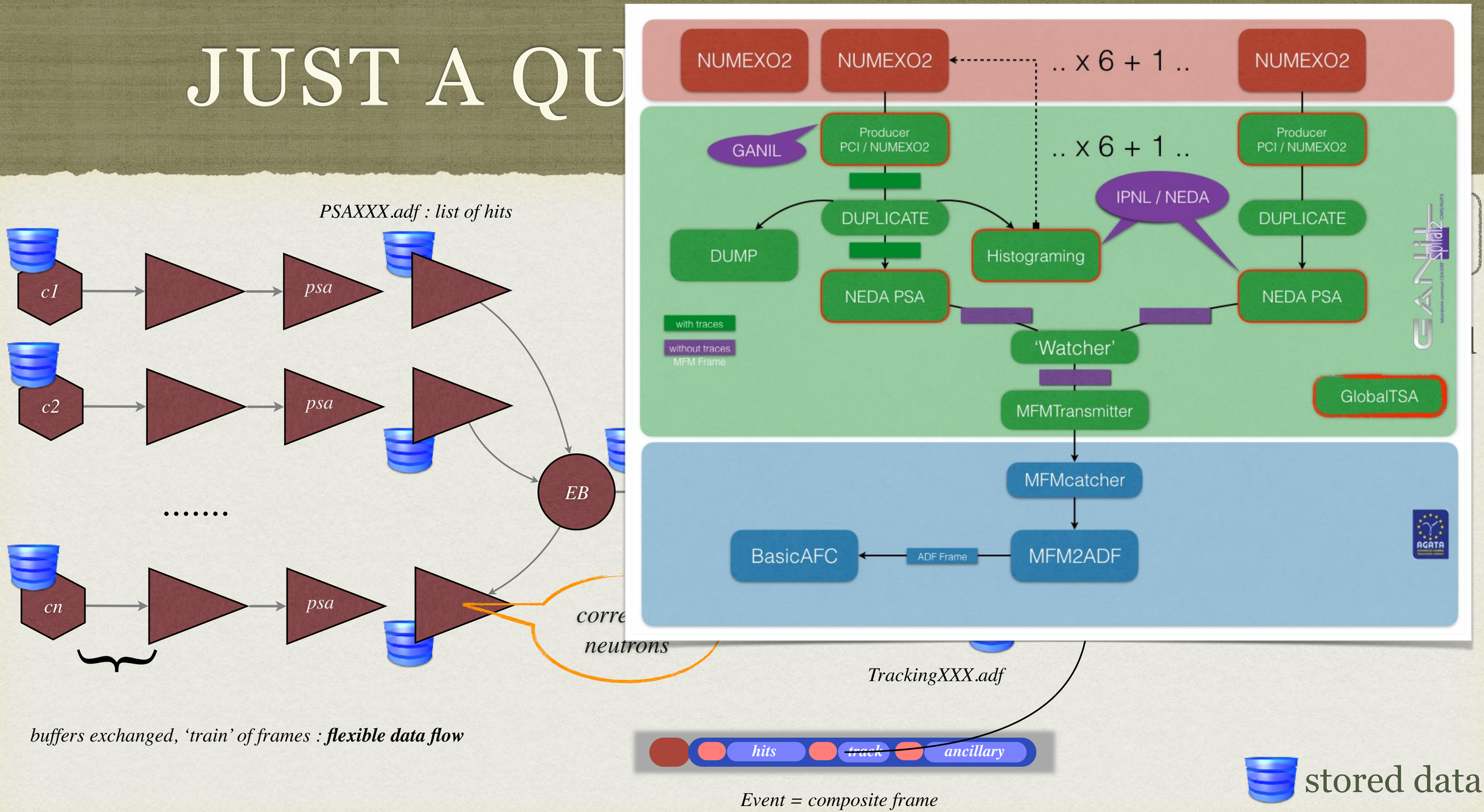
stored data

JUST A QU



buffers exchanged, 'train' of frames : ***flexible data flow***

JUST A QU



buffers exchanged, 'train' of frames : ***flexible data flow***

GRETINA INTO AGATA

- Extensive presentation of the data { Processing (NARVAL/DCOD - Online) / Re-Processing (emulators - offline)

  ➤ **TODAY : Just a quick reminder**

- Should be flexible enough to allow processing of GRETINA data (algorithms)

  ☛ It should be interesting to follow Amel & Torben's work done on Tracking

  ☛ but the other way around i.e. GRETINA through AGATA

  ➤ **TODAY : what has been done so far, about to be done**

- Solutions to speed up complex / more complex algorithms

  ☛ hpx* could be a solution

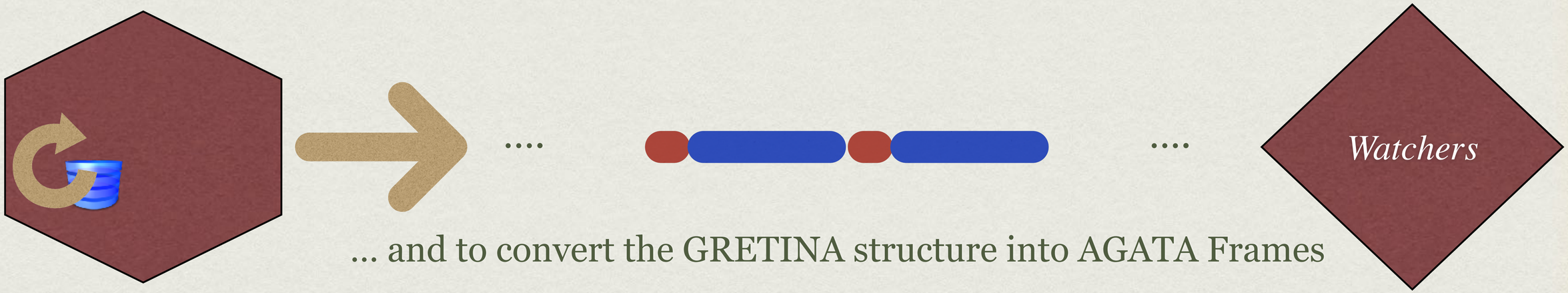
  ➤ **TODAY : other solution under investigation since September**

GRETINA INTO AGATA

The work has started … has been stopped … … … has started again … since last workshop

<u>First step almost completed</u> : a producer able to read GRETINA files …

Asynchronous reading



… and to convert the GRETINA structure into AGATA Frames

GRETINA to ADF Producer

GRETINA INTO AGATA : HITS

- After PSA : this is the easiest case (data are well similars into the two frameworks)
  - ☛ One can see the different labeling between the two arrays
  - ☛ Still some issues (spikes in some spectra)

- Traces: a little bit more difficult (data contents less compatible)
  - ☛ Different labeling
  - ☛ Different organisation: calibration, bases, storage

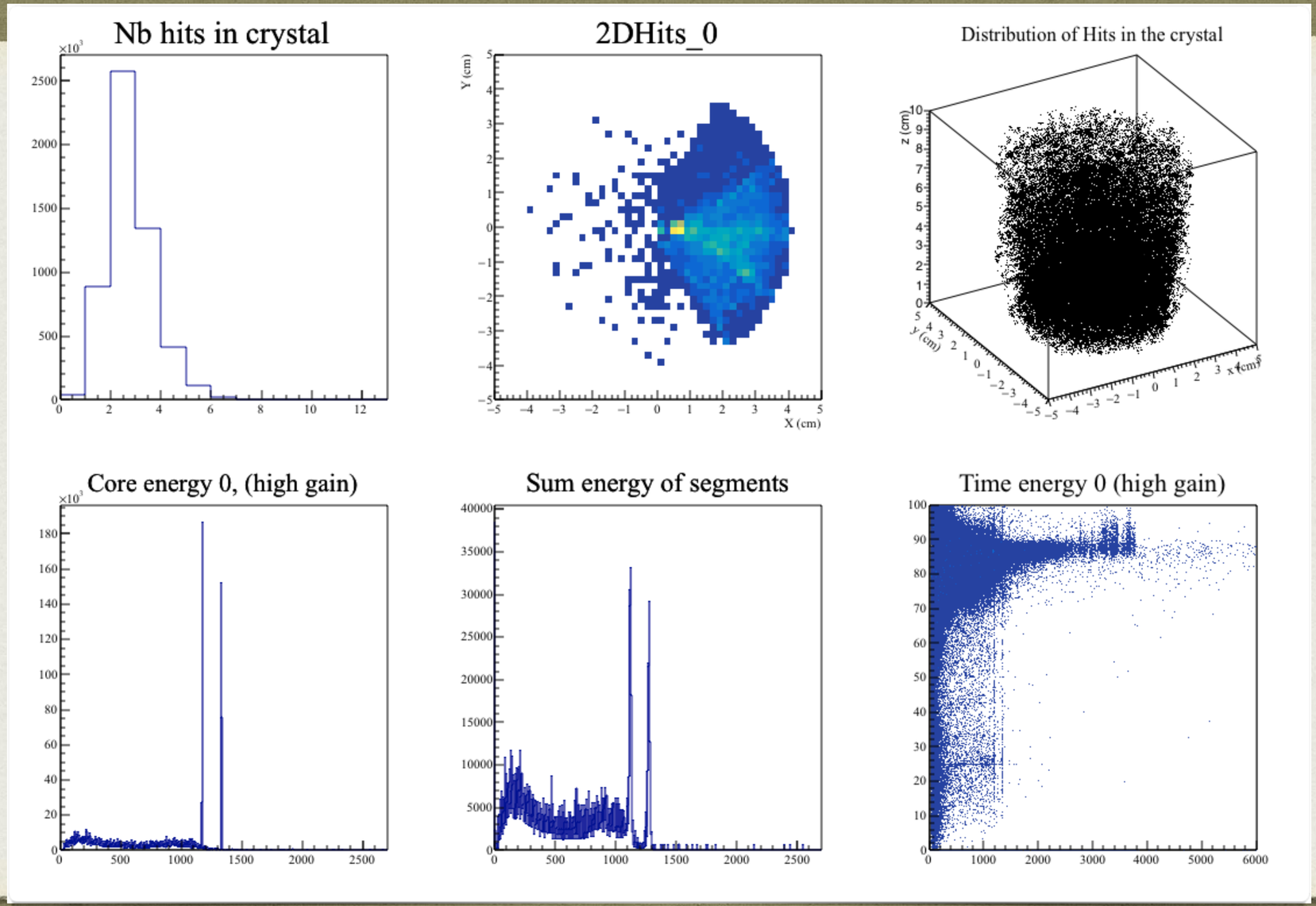
Small data set used so far. Good to start, not enough to fully debug

- ☛ Still some discussions required to define clearly complex topologies

We do need also some reference data set to compare at the end the results

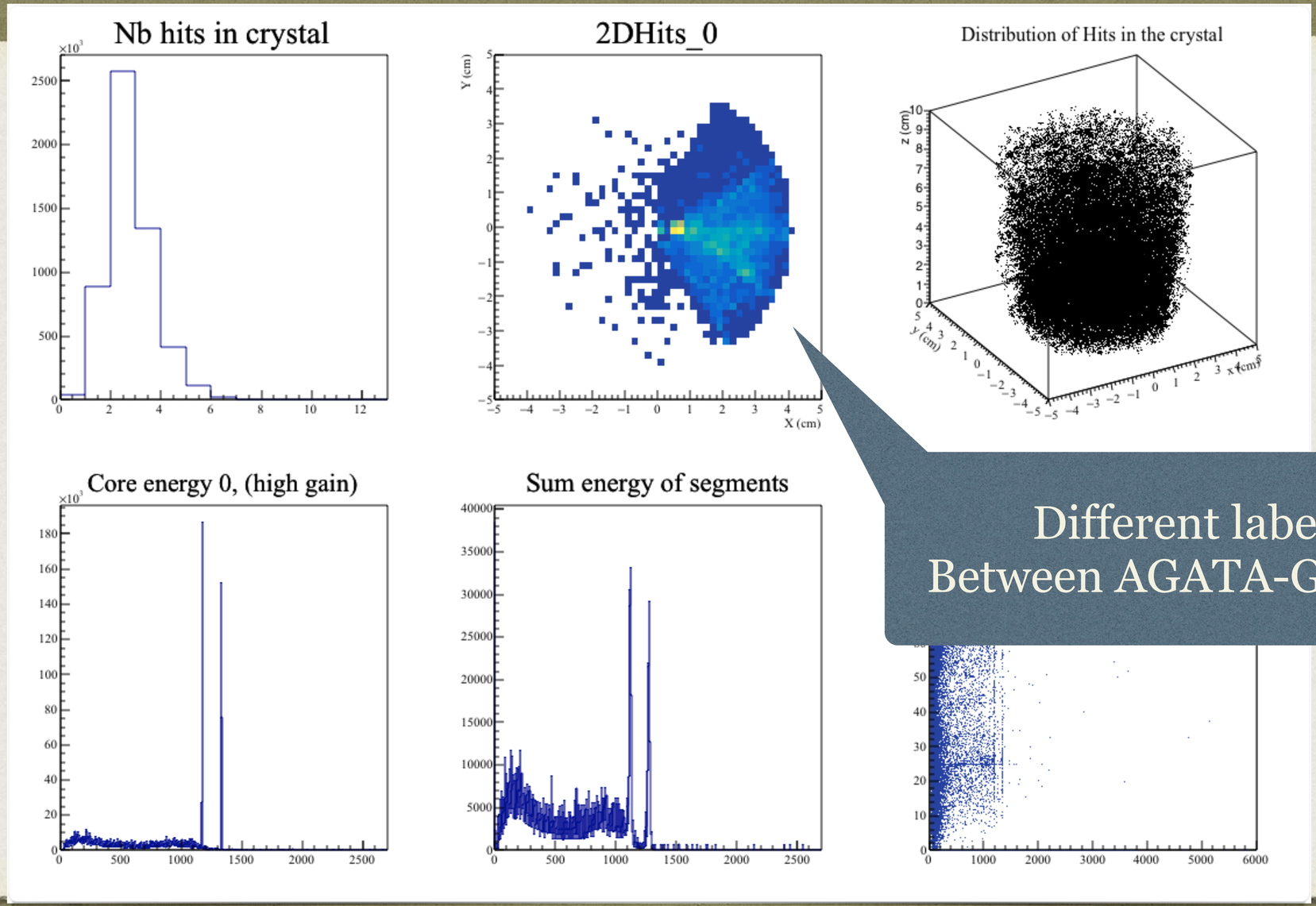
GRETINA INTO AGATA: PSA HITS

PSA Watcher



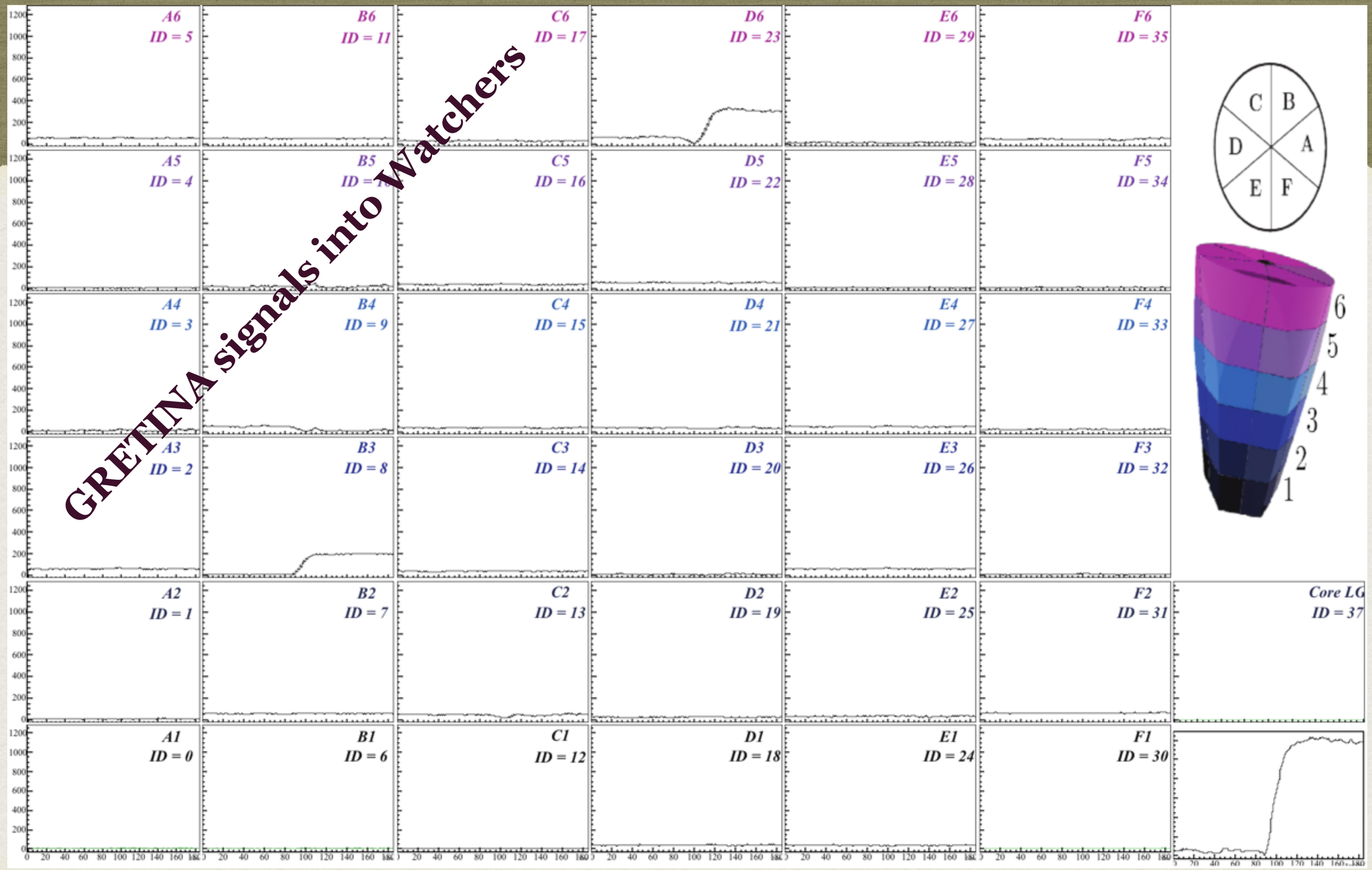
GRETINA INTO AGATA: PSA HITS

PSA Watcher



Different labeling
Between AGATA-GRETINA

Crystal Watcher



GRETINA INTO AGATA : HITS

Second step about to be started: work scheduled before summer

1️⃣ Plugin of GRETINA code into AGATA actors

Tracking ✔
PSA ?

☛ Would allow to process GRETINA through … GRETINA algorithms
But using the AGATA framework ! … a good way to check everything is mastered.

2️⃣ Full processing of GRETINA data through AGATA algorithms

3️⃣ It might be an opportunity for us to test new solutions …

NEDA: NEURAL NETWORK

- Extensive presentation of the data { Processing (NARVAL/DCOD - Online) / Re-Processing (emulators - offline)

  ➤ **TODAY : Just a quick reminder**

- Should be flexible enough to allow processing of GRETINA data (algorithms)

  ☛ It should be interesting to follow Amel & Torben's work done on Tracking

  ☛ but the other way around i.e. GRETINA through AGATA

  ➤ **TODAY : what has been done so far, about to be done**

- Solutions to speed up complex / more complex algorithms

  ☛ hpx* could be a solution

  ➤ **TODAY : other solution under investigation since September**

NEDA: NEURAL NETWORK

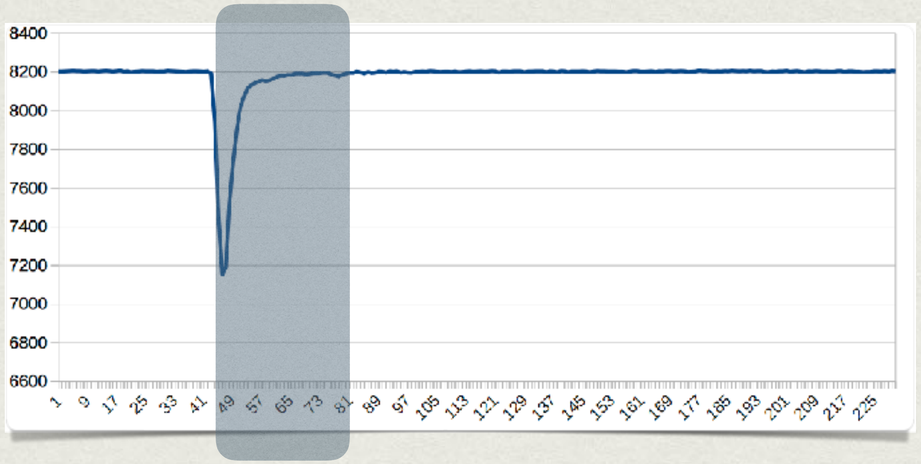
The new AGATA campaign coupled AGATA with

Commissioning this week !

} NEDA 96 neutron detectors
DIAMANT charged particle detectors

Both with digital electronics

NEDA requires also Pulse Shape Analysis !

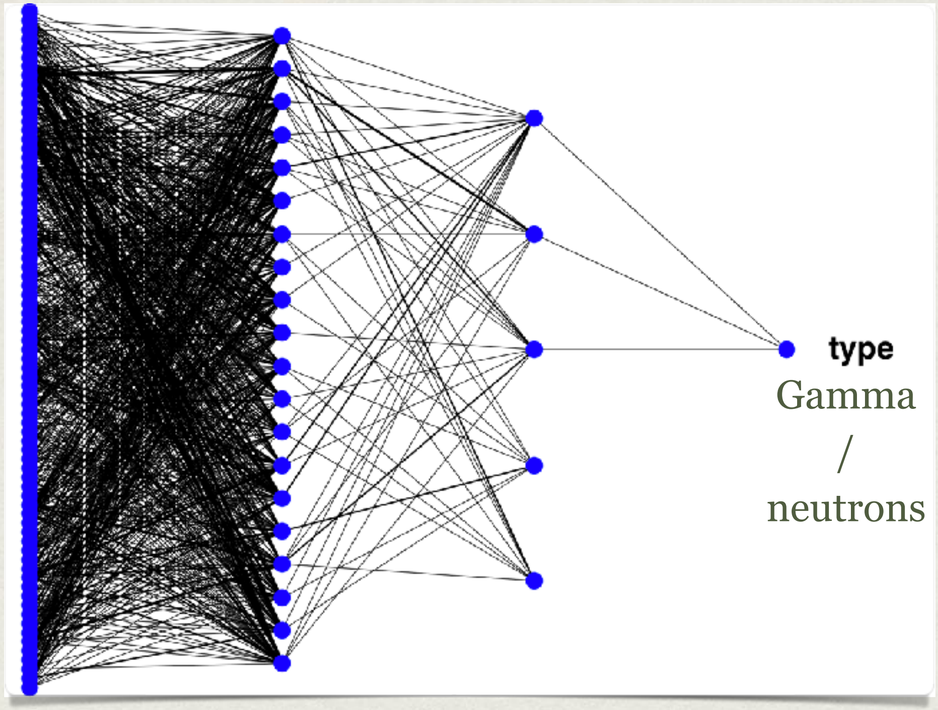


75 samples

Two versions
Charge comparison - Neural Network

It requires computing powers !
A ‘simple’ case to learn before using this technic for AGATA

NEDA: NEURAL NETWORK



First implementation of the NN using ROOT API
Problem of performances / kind of network limited

First benchmarks we done :
ROOT Neural Network ~ 50 slower that Charge Comparison
(on single core)

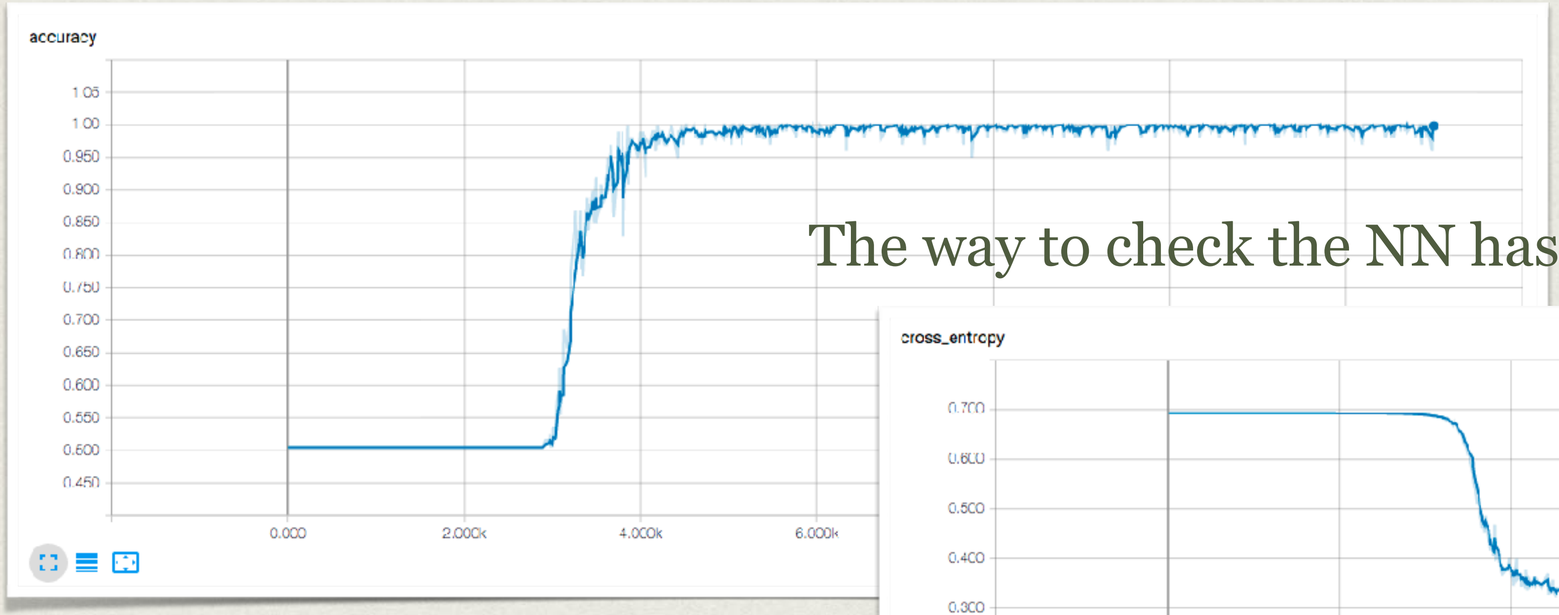
We decided to move to Tensor Flow (google)
☛ It allows running (offline) on GPU !

NNPSA

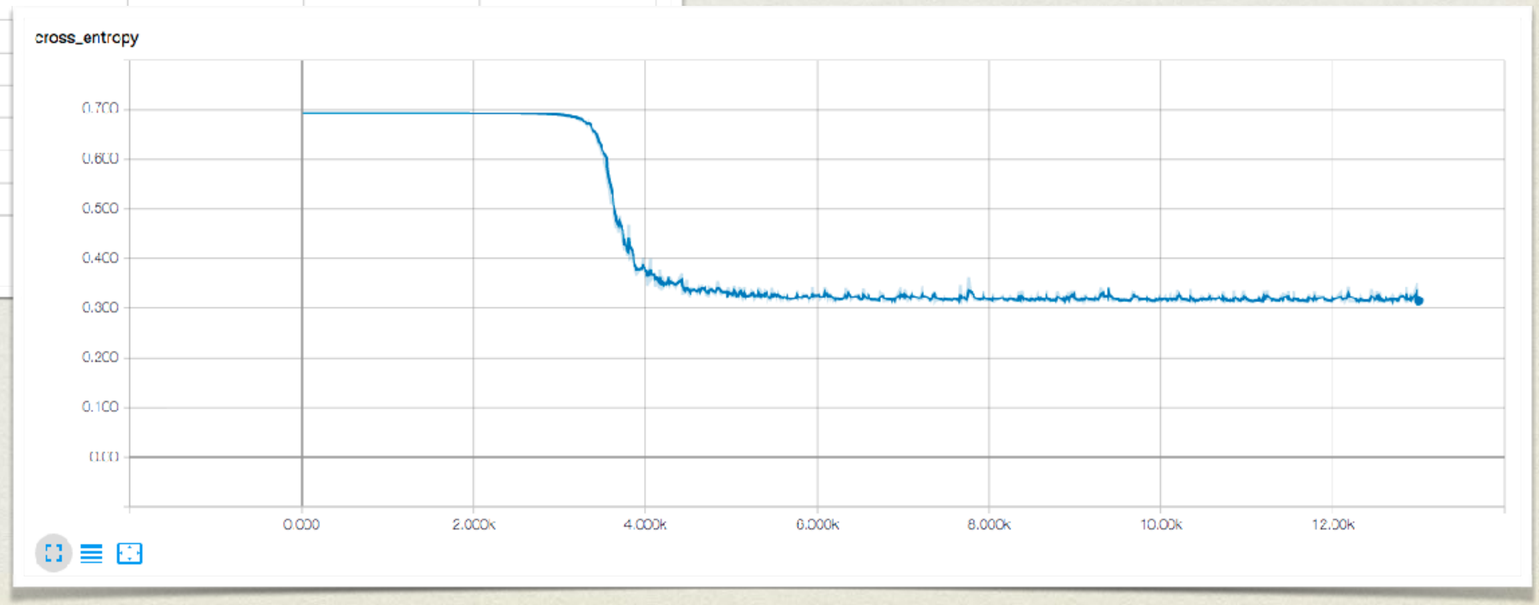
A Filter, embedded Tensor Flow, has been written :
Used to train the NN
Used to perform PSA

NEDA: NEURAL NETWORK

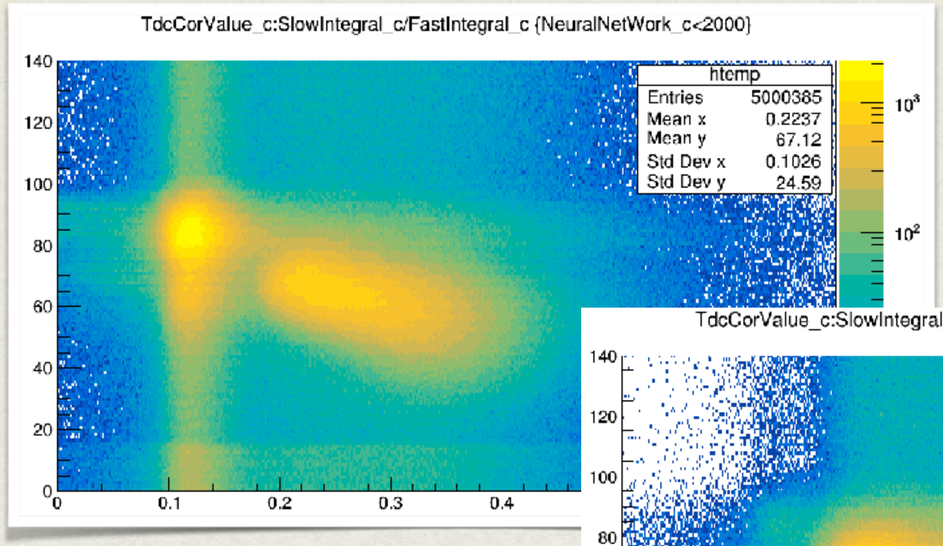
Training part, huge gain running it on GPU farm @ CCIN2P3



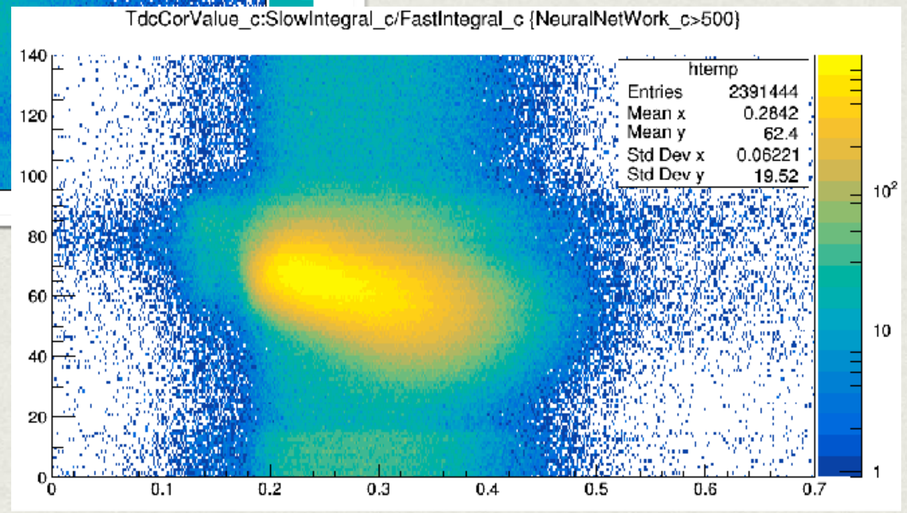
The way to check the NN has learned correctly



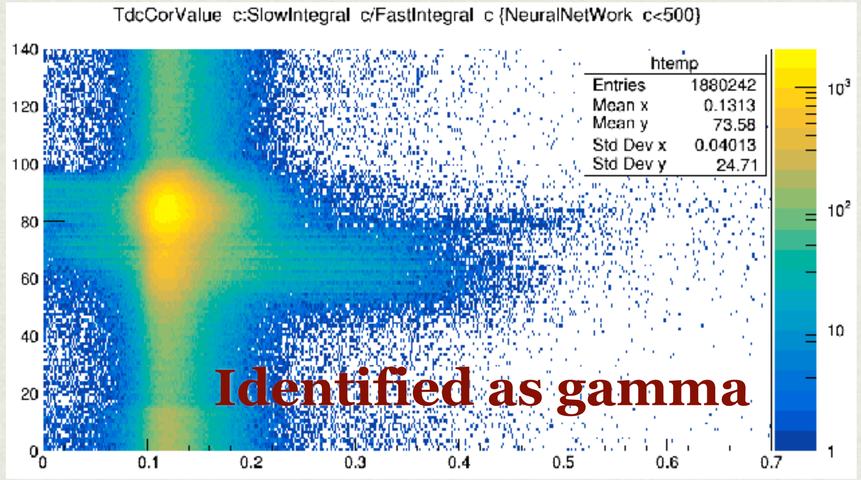
NEDA: NEURAL NETWORK

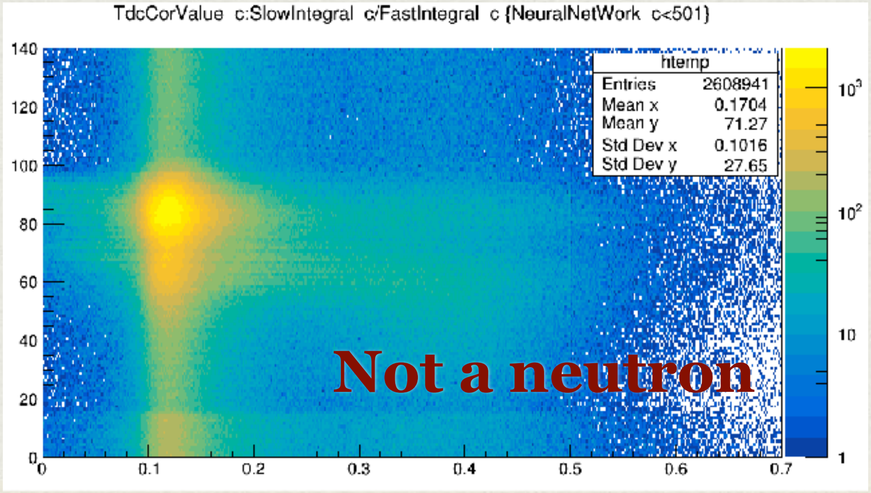


All signals

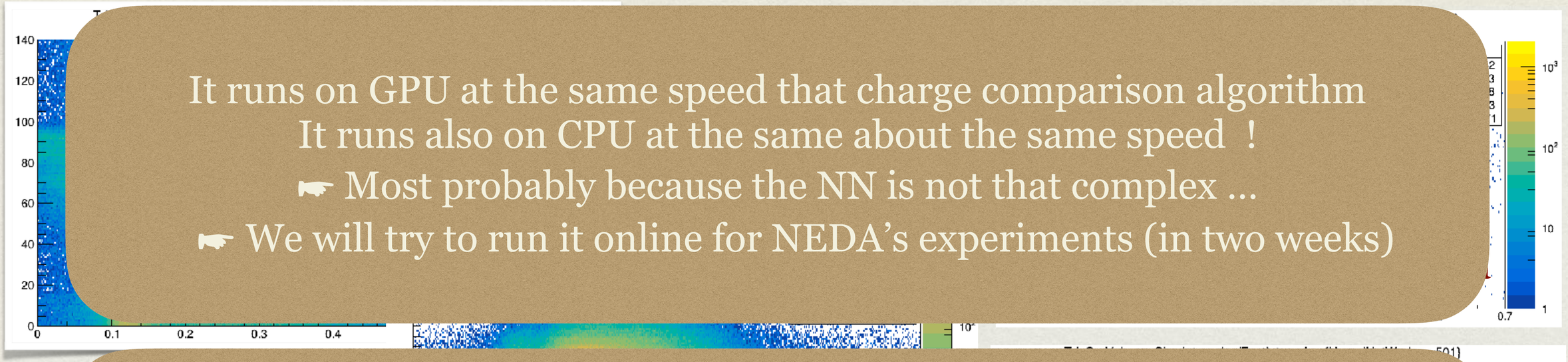


signals identified as neutrons





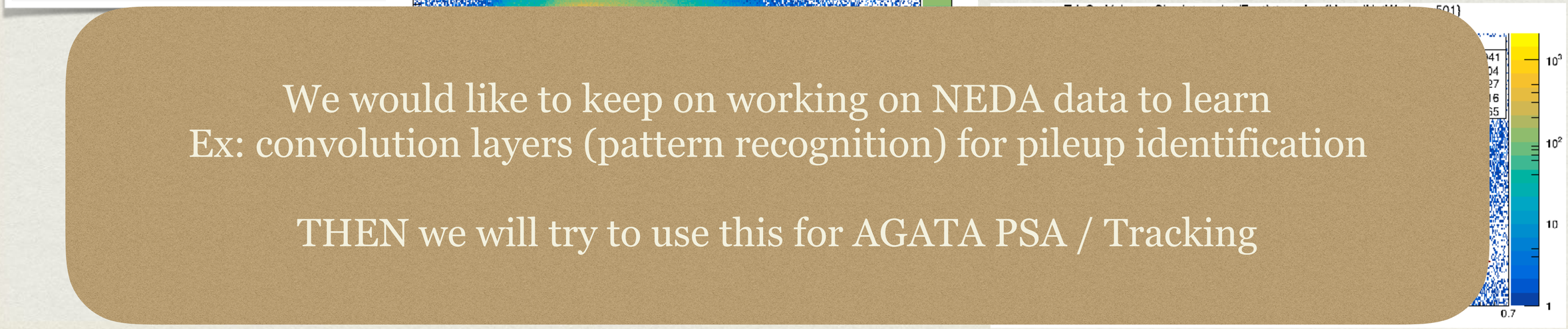
NEDA: NEURAL NETWORK



It runs on GPU at the same speed that charge comparison algorithm
It runs also on CPU at the same about the same speed !

☛ Most probably because the NN is not that complex …

☛ We will try to run it online for NEDA's experiments (in two weeks)



We would like to keep on working on NEDA data to learn
Ex: convolution layers (pattern recognition) for pileup identification

THEN we will try to use this for AGATA PSA / Tracking

CONCLUSIONS

The work to play with GRETINA data in the AGATA Data Processing chain has started

☛ The main brick is there, still some debug / adjustments

☛ The real work to compare the performances of GRETINA / AGATA algorithms to come

We have started to investigate Neural Network solutions using Tensor Flow

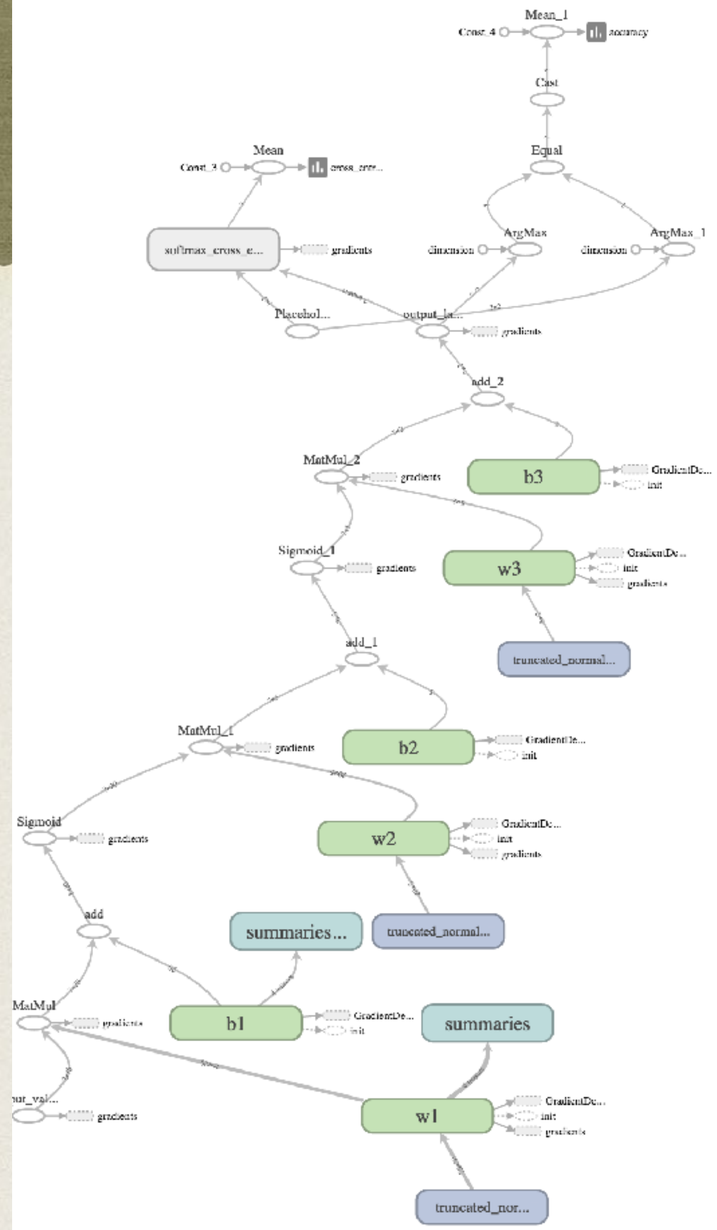
☛ NEDA data to learn

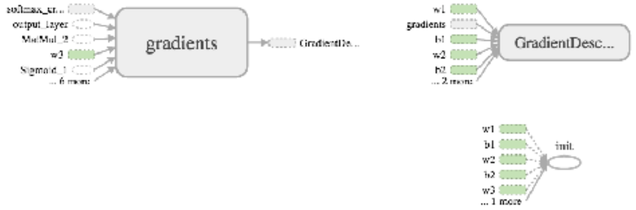
☛ then try on AGATA/GRETINA data

Software distribution : we have started working with docker

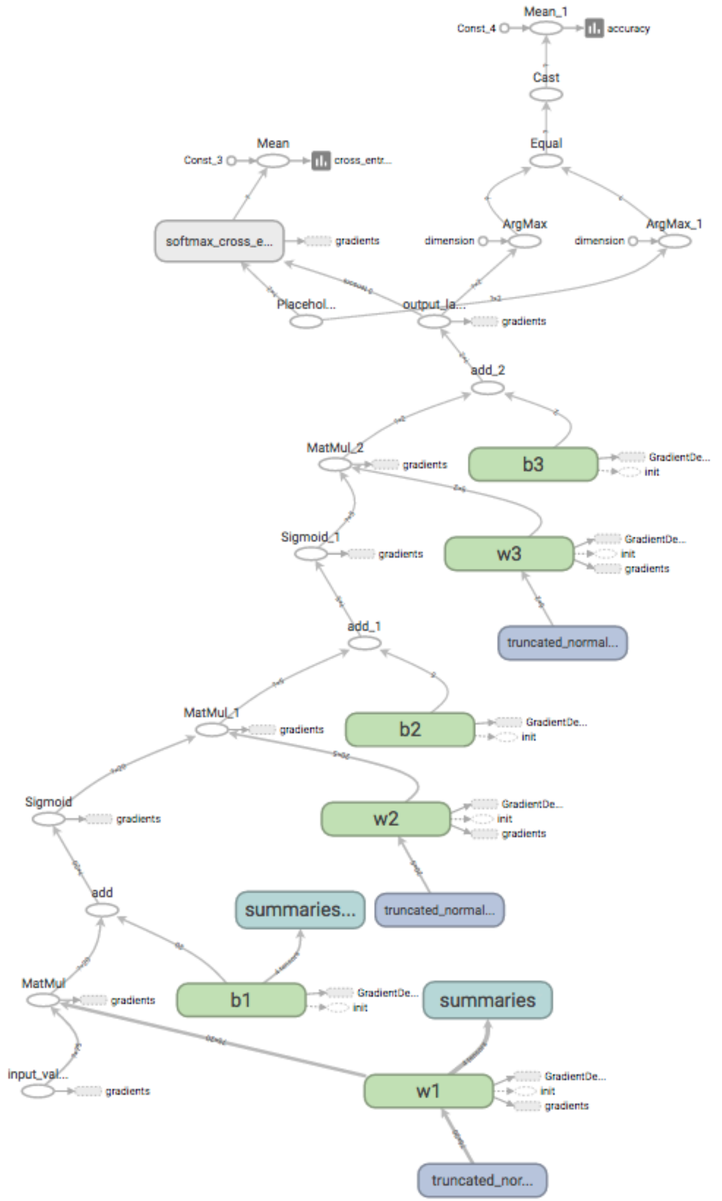
IPNL team : G. Baulieu, X. Fabian, L. Ducroux

Many thanks to Amel, Heather and Torben for their help


Mean_1
Const_4
accuracy
Cast
Equal
Mean
Const_3
softmax_cross_e...
gradients
ArgMax
ArgMax_1
dimension
Placehol...
output_la...
add_2
MatMul_2
b3
GradientDe...
init
Sigmoid_1
w3
truncated_normal...
add_1
MatMul_1
b2
Sigmoid
w2
add
summaries...
MatMul
b1
summaries
w1
truncated_nor...


gradients
GradientDesc...
init

Main Graph


Mean_1
Const_4
accuracy
Cast
Equal
Mean
Const_3
softmax_cross_e...
gradients
ArgMax
ArgMax_1
dimension
Placehol...
output_la...
add_2
MatMul_2
b3
GradientDe...
init
Sigmoid_1
w3
truncated_normal...
add_1
MatMul_1
b2
Sigmoid
w2
add
summaries...
MatMul
b1
summaries
input_val...
w1
truncated_nor...