

HSF and visualization
My two cents



April 2017

HSF and vis



- Is visualization **really** taken into account ?
- It was not considered in first « white papers ».
- Is it going to be « as usual » the third wheel of the story ?

If not in the « grounding equation » we are going to fall on the today situation where « data framework », detector and event models are driven by « blind batch » without the hope to enjoy fully today and future interactive and graphics technologies.

Today

- On experiments, from a vizzier point of view, the overall situation is a raw disaster.
- The disapointment had been high for me around LHC and LHCb.
- Non existing portability and a nightmare of complexity. (People confuse being complete and being complex, or doing simple and being naive). Why not being « smart, simple and complete » ?

No way to read an event file on 99% of devices that people have in hands today. What is the point of a « data framework » that does not permit to open and exploite data in a file on your own device ?



Today

- A foundation may help... for future experiments !
- For today ones it is too late ; their « code entropy » is too high to recover.

How to put HEP software « with vis » back on track ?

Foundation and Empire ?



Avoid building from the unacceptable :

« take it (as it is) or leave it »

« users vote with their feet »

« at the second line of code, the user is an expert »

« the one with a max (?) of users wins »

Is HSF virgin from this kind of spirit ?



NO

- **No**, a user is not an expert on the internal of an IO system or a graphical package after the second line of code. (He is only an expert of some API). Then a « user driven » approach has limits. At some point engineers/computer scientists must enter in the game.
- **No**, I am not in a private software company. My mission at IN2P3 is not to « win to get a max of clients », but to « help a bunch of researchers to build their experiments to do physics ». Some « academic foundation » should enforce that.
- People that confuse « academic/public » with « private » should strongly reconsider their life in public science (and for some join the army).

Something found on a wall at Orsay



La science n'est pas une marchandise.
Une université n'est pas une entreprise.

Science is not a merchandise.
A university is not a private company.

Then : why do we write code ? To run experiments to do physics.

Scientific spirit



- Some foundation makes sense only if it is first of all based on a scientific process and then made by scientists.
- What is a scientist ? A scientist is someone coming with a reasoned argumentation about something but who can change his/her mind if shown to be wrong. If not, he/she is a dictator.
- The scientific process is the debate around various reasoned argumentations up to an agreed conclusion about the « something ».

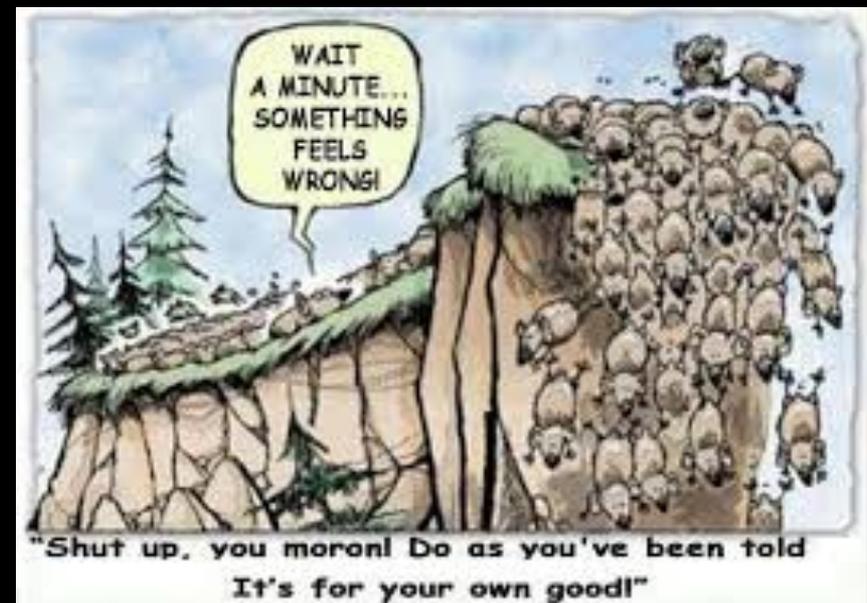
Are there scientists around HEP software ?

HSF and CERN-ROOT

Obviously for me CERN-ROOT can't be a basement if still driven by the motos found in slide #5.

For me HEP & ROOT is still...

(and be sure I would be glad to change my mind) (and somewhere it is not so much funny).



About visualization : does the herd makes a difference now between an interpreter, an IO package and a graphics package ?
Not sure. If not, still no way out.

CERN : E for what ?

This being said...



- I share the idea that « people that write the code decide ». Hell, sure. (For your house, you do not ask an electrician to decide for the plumbing).
- Which is **not** « people that are involved decide », especially if we can't find ten lines of code written by them that had been released one day or another on some experiment.
- But I don't want to be compelled to drink code of others if having no way to question the recipe of their cocktails. Which is why more is needed...

Which software ?

- I learnt to divide « computing » in three parts :
 - Hardware (setup of a data center, perspire to lay cables, etc...)
 - « software for hardware » (OS, IO lib, globus, OpenStack, etc...)
 - « software with physics » : codes in which we can find the words « detector, track, energy, calo cluster, etc... », words that are not found (I hope !) in the upper software category.
- « software with physics » => center of some HSF.
- And mainly because it is in our hands !
- Belong to this category : detector and event models, simulation, reconstruction, « data framework ».

Let us be clear...



- Dividing the « software world » in two is with respect to everybody !
- But, force to state that some knowledge in **physics** (and taste for) is needed for « software with **physics** » *.
- And be sure that to pass days, months, years in some « software with physics » around is definitely not a barrel of laughs or trip to heaven. (For some we may even develop various allergia, be very sick, or finish in some asylum if not having the right holographic brain **).

* It is not needed too to know everything of supersymmetry.

** the code of supersymmetry being not without risk too.

Vis, GUI ?



An IO package, a GUI package, a vis lib, an interpreter are not « software with physics » and are, for me, clearly on the « software for hardware » side :

- IO : related to storage hardware.
- Graphics : related to screens.
- GUI : related to « control things » with a mouse.
- interpreter : related to « control things » with a keyboard.

{Vis, GUI, interpreter} = facilities to handle « interactivity ».

These are **not** central, but must be considered by the « center » to handle various hardware correctly.

A good point for CERN (beside pizzas)

- CERN/EP/SFT to handle « software with physics » being not CERN/IT to handle hardware and « software for hardware ».
- This kind of splitting should be propagated in other labs !
A HEP foundation can/must help to pass the message.
- And this even if you would have only a couple of guys in the
MY_LAB/EP/SFT
It would help to organize (and, again, with respect to everybody).

CERN : E for what ?

EP/SFT versus IT



- This kind of splitting does not mean a divorce.
- It should lead to new ideas for new/better platforms to do physics.
- and then much more interesting activities for everybody...

Bad points for CERN

- Beside **Breaking Bad** with **ROOT** in 1990s, definitely too much « web oriented » *.
- Fine with that as long as it does not stop others to explore other things : **your freedom stops where my freedom starts.**
- Right now I am stucked with mobile devices because of that.
- **web is ok for outreach, but not for intensive visualization.**
- **web = carpet to hide poor « frameworks » in (Linux only) servers ?**
- **Some foundation may help to « raise flags ».**
- **A MyLab/EP/SFT « mobile oriented » plus some foundation may help to balance a WebLab/EP/SFT for the best of the whole HEP.**

* ok, we know, the web had been invented at the JurassicLab, bla, bla, but in... 1989 !



1990s ☹️

Synthetization of **ROOT/meta** and:
gROOT, gFile, gDirectory,
gSystem, gEnv, gProof, etc...



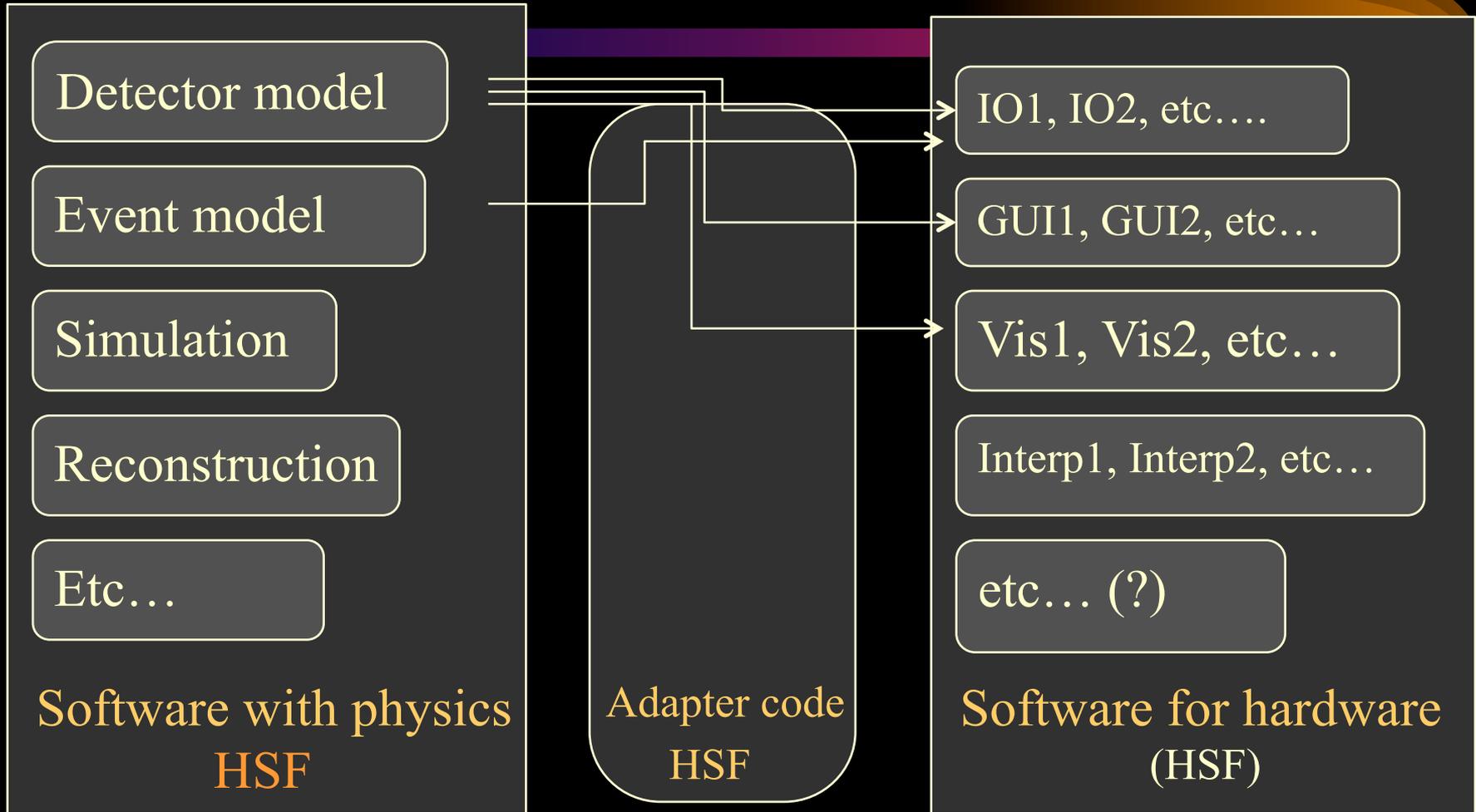
Somewhere around CERN

<labs>/EP/SFT & HSF to help
HEP to recover.



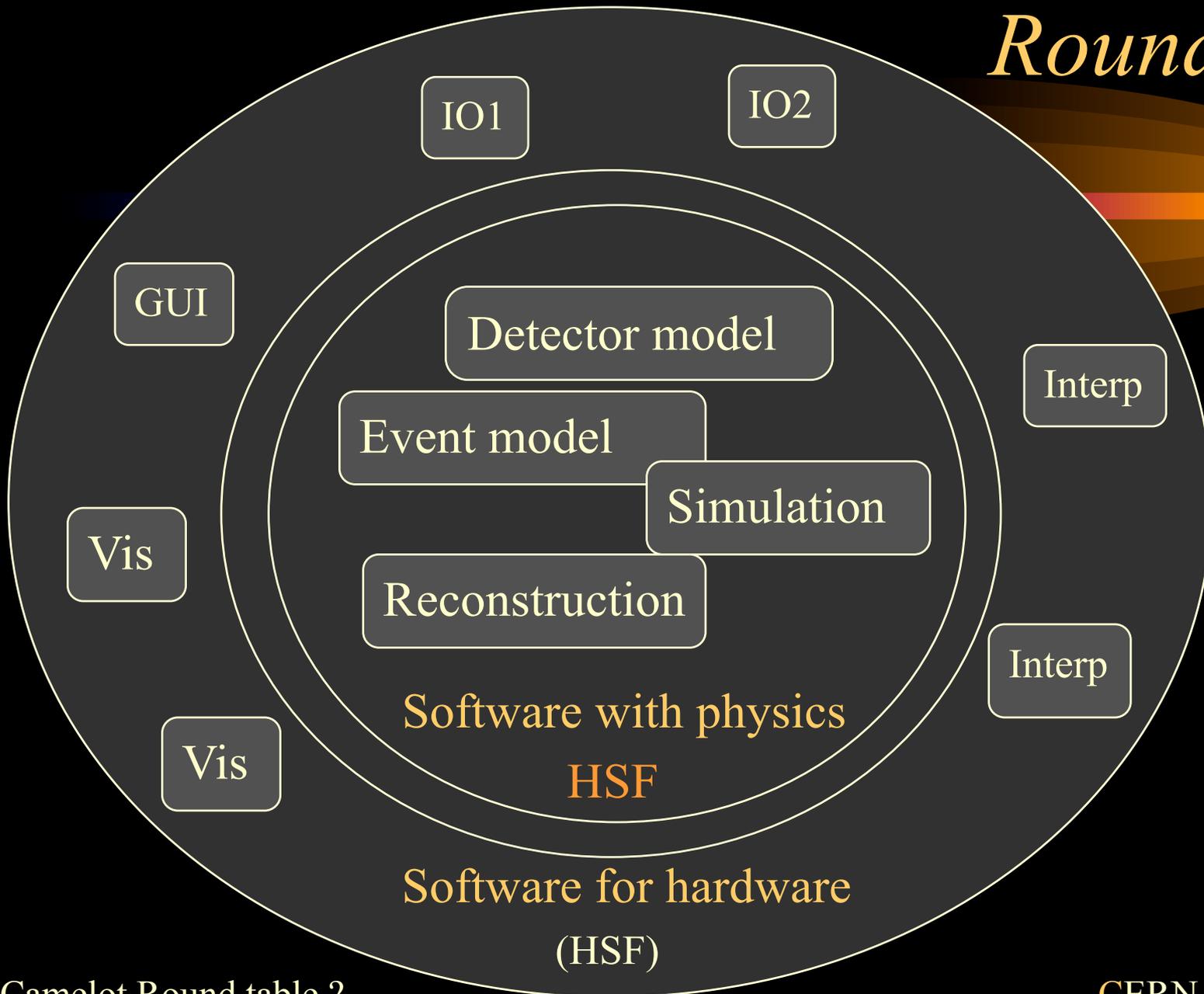
The Code & HSF

A general pattern



Mainly with an « adapter arrow » for « all at left » toward « all at right »

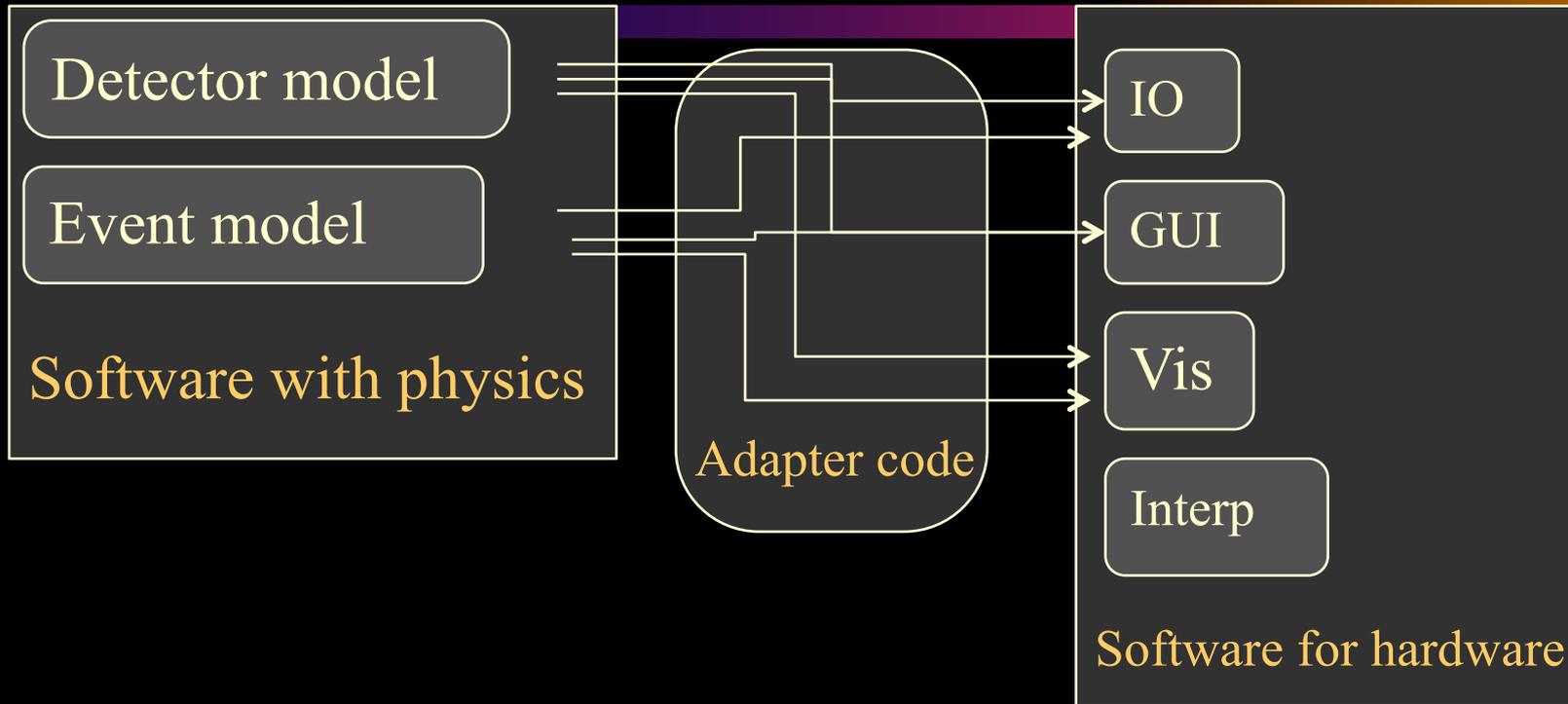
Round view



Camelot Round table ?

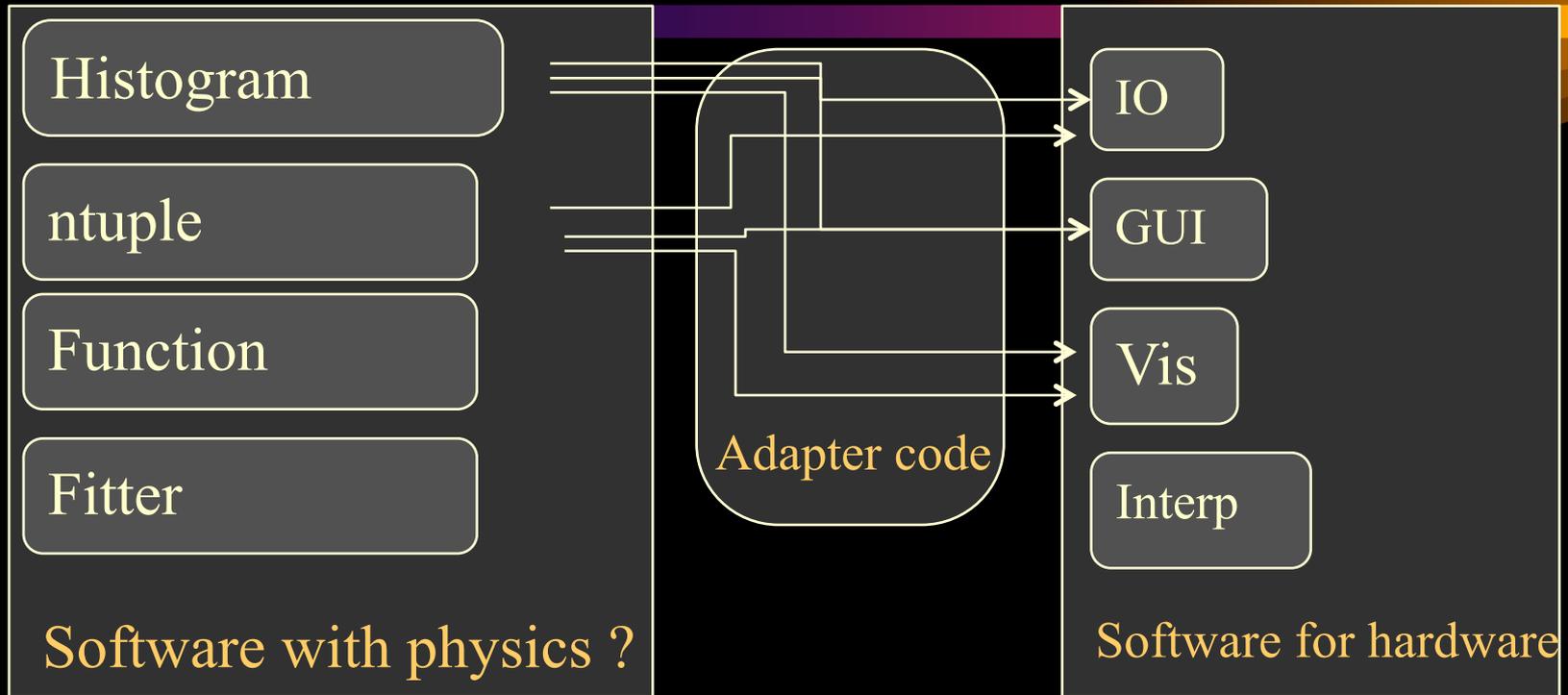
CERN = Camelot ?

Sketch of an event display



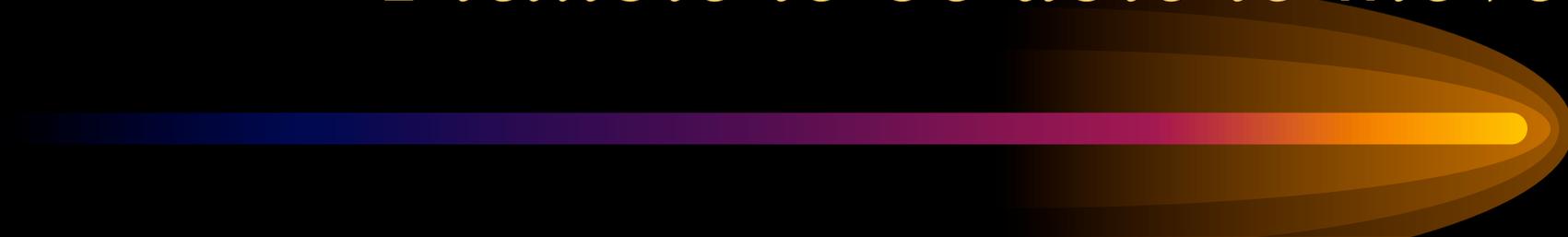
The « vis » adapters are the « data representations ».
Some of the « adapters » can be produced automatically.
(For exemple by swig for the interpreter ones).

Sketch of an analysis tool



- Histo at left ? Hmm ! being a basic « HEP tool » let us say yes...
- The « vis adapter » of an « histo » is a « plot » !
- For me, plotting is part of the HEP visualization equation.

Flexible to be able to move



- The overall code/foundation organization must be highly flexible to be able to adapt for changes « on the right ».
(Which is why « you take (the whole as it is) or leave it » is unacceptable).
- If some « software for hardware » can't be found elsewhere to satisfy some HEP need, it may fall in HEP hands. (Then be in some close perimeter to the center).
- Bright exemple : ROOT file format * twenty years ago. Yippe !
But, alas, tied « to the rest ». ☹ Sigh.

* I had and have nothing against the ROOT file format.
But now do we have something else for the IO ? HDF5 ?

Methodology : decompose problems

- An IO package is not a graphics package, a command line system is not a simulation toolkit. **Obvious no ? except in HEP !**
- Having first a common decomposition of HEP things as described before seems to me mandatory in the methodology.
- In visualization, it would help to share experience and code between **one wanting** a GUI done in Qt, plotting done with ROOT and detector-event graphics done with Coin3D, and **another one wanting** to do graphics with ROOT, plotting with matplotlib, read data from an HDF5 file and agitate things only from a python prompt (and then without having to build and link some cling-clang-clong interpreter or ROOT GUI since not used in his configuration).

Methodology : refactor ever and ever

- Which is not the same than rewriting.
- Refactoring could be continuous or quantised *.
- I am now at my third « refactor quantum » of my things.
- We must be able to deprecate ** a software, and then do another « refactorum », if it is needed to pass to something else. (I have that in my license : « the right to deprecate »).
- Again, our users are not « clients » ; being researchers, they are fully aware that « moving » is part of the game.
- A foundation may help to enforce that.

* quantum software at last ! (What is the wave function of ROOT ?...)

** deprecate = stop developments but still do user support of last release.

Methodology

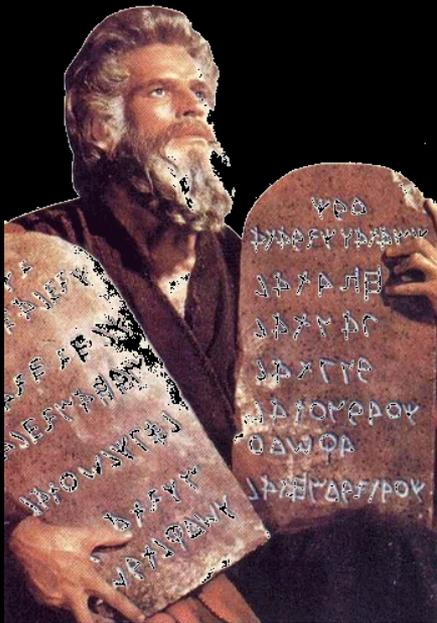


What is astounding, and a little bit exhausting, is to have to repeat for ages these kind of basic engineering principles.

Will HSF help here ?

(... What is sure is that in ROOT we have for long entanglement phenomena).
(In fact, is ROOT dead or alive ?). (Supersymmetry looks dead. Incidentally, I never had the strong feeling to be in a space-time with some extra fermionic dimensions, then it is ok for me).

Then...



HSF and vis ?

My tablet

- Some foundation must have solid and « luminous » sociological and technical grounding enforcing at least :
 - The scientific process. (The basic engine).
 - We are « public academic ».
 - We write code to run experiments to do physics.
 - People that write code decide (according to first item).
 - « software with physics » at the center.
 - « software for hardware » at a close perimeter.
 - Modular code structure to have a maximum freedom to move.
- Specific to visualization : target to run on local devices to offer people « great user experience » (then be able to read and exploit an event file « in the hands of people ») (and then « have fun »).

Without that, bof...



Work plan...

First option : the next millenium

- An HSF tablet that makes sense (around 50 years).
- EP/SFT in labs (around 50 to 100 years).
- Refactor ROOT and Geant4 (around 100 to 500 years).
- Have « core software » people on experiments at work so that we can read and exploit an event file effectively on anything having some silicium inside (500 to 1000 years).

Then HEP software will be **on track** and humankind will have nice in-brain-displays for the **SSHC**.

(Solar System Hyper Collider around the sun planned for 3020)



Wouaou, it is going to be great !

or...

Second option : 2027

- An HSF tablet that makes sense (2 years).
- EP/SFT in labs (2 to 5 years).
- Refactor ROOT and Geant4 (5 to 10 years).
- Have « core software » people on experiments at work so that we can read and exploit an event file effectively on anything having some silicon inside (10 years).

Then HEP software will be **on track** concerning visualization and... just on time for me to say good bye !





Which option do you prefer ?

(or a forever Camelot Empire **ROOT** Network one ?)