

Retours d'expériences iOS/Android en HEP et Astro



JoSy / 4 Octobre 2016

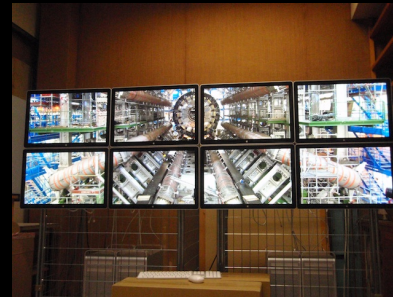
2010

- iPad :



(RIP +5 !)

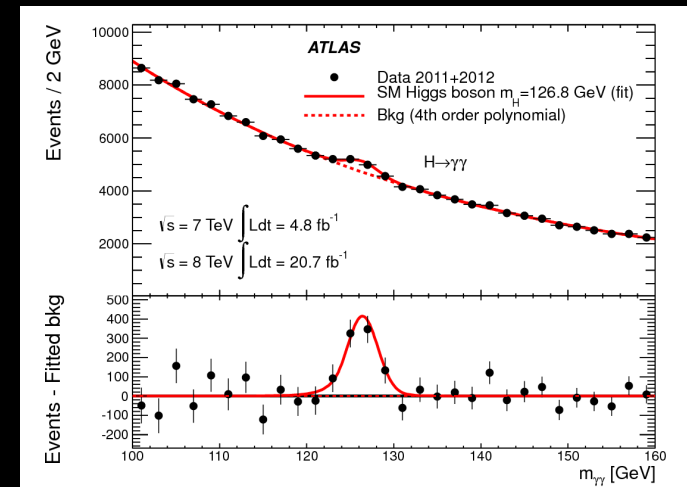
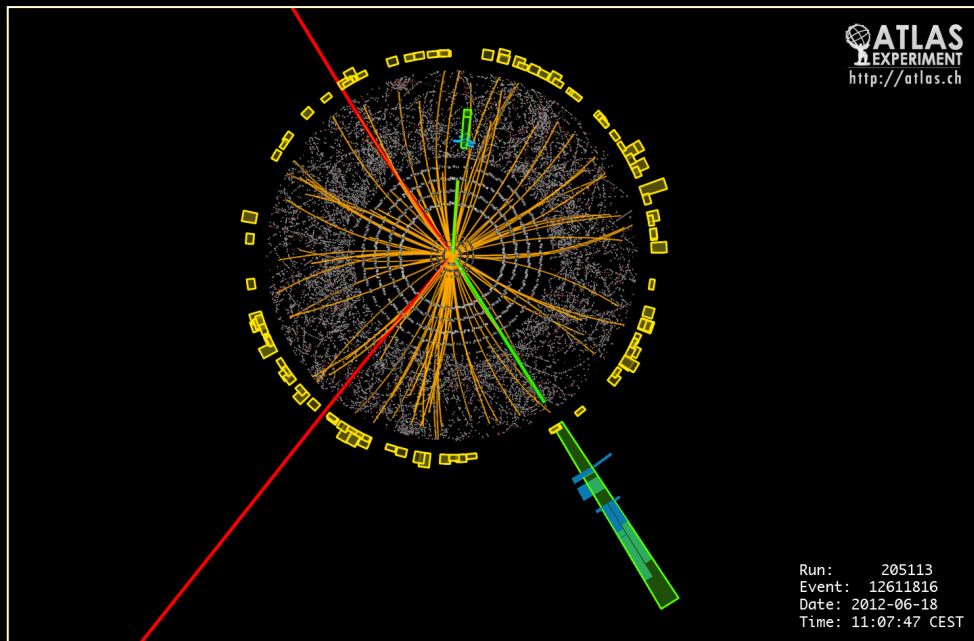
- Tablettes Android peu de temps après.
- Les « app store » : Apple AppStore, AndroidMarket.
- Petit mur d'écrans au LAL :



Je remets à plat ma manière d'aborder le graphique et l'interactivité en général autour de mon activité HEP (High Energy Physics).

Refaire le graphique ?

- En fait le genre de graphique que l'on fait sur nos expériences :



- Donc des carrés, des traits et des points avec, soyons fou, du texte !

mi-2010 / début 2011 : au boulot !

- Avant : OpenGL, Coin3D, Motif, gtk, Qt, scripting (tcl, python) : rien de tout cela n'était disponible sur iOS & Android.



iOS : Objective-C, UIKit, GL-ES.

Android : java, View, GL-ES.

- Mais C/C++ et GL-ES sont communs: on construit sur cela.
Aussi disponible sur Linux, macOS, Windows !
- Grosse expérience d'OpenInventor (Coin3D)
- On refait un « gestionnaire de graphes de scène ».
- On fait aussi le GUI avec cela ! (donc sur GL-ES).
- 99% du code est commun à toutes les plateformes.



Donc...

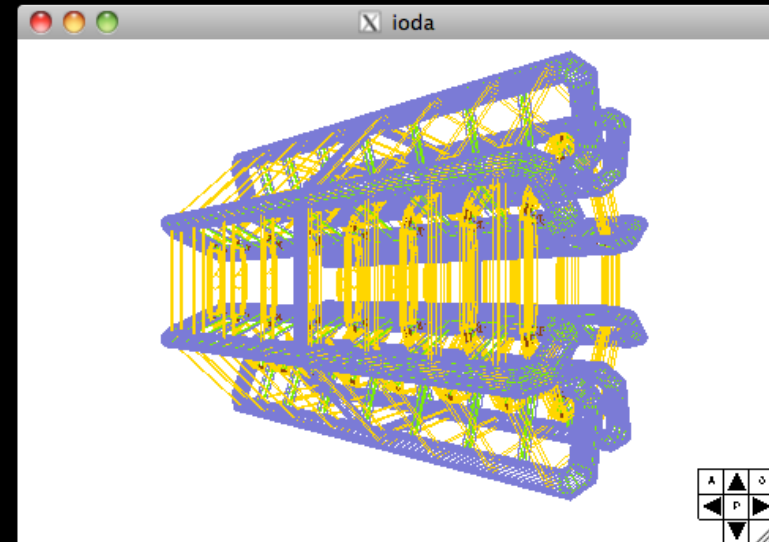
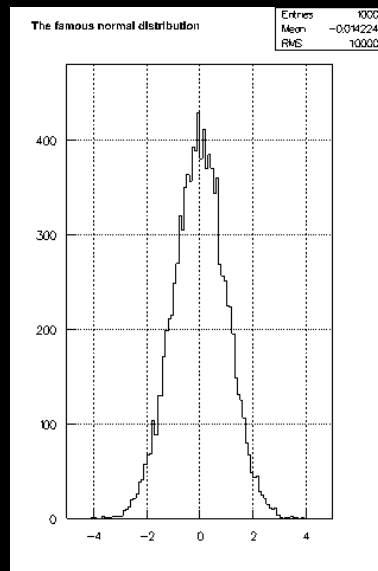
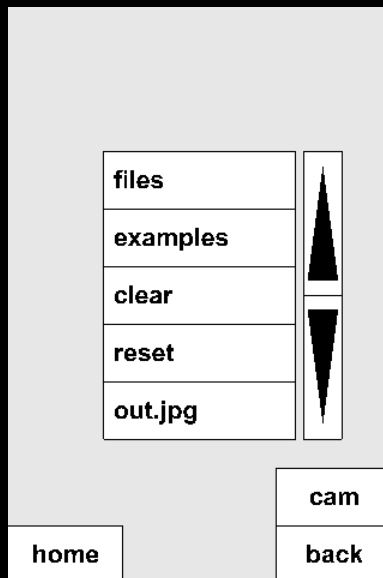
- iOS : on fait un minimum d'Objective-C.
- Android : on fait un minimum de java (max de NDK).
- À Tim, Sundar, Satya, Linus je ne demande : qu'un compilo C++, une surface pour faire du GL-ES et un moyen de récupérer les événements "touch" (ou "pick") et c'est tout !

- Et avec cela (et quatre devices) je refais **mon** monde.



début 2011 / ioda

- **IODA** : première « app » sur les stores orientée « analyse » : visualisation d'histogrammes d'un fichier local au format CERN-ROOT et d'un peu de détecteur (format Saclay/fog) :



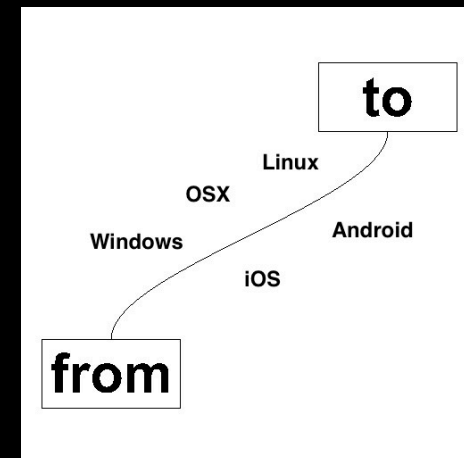
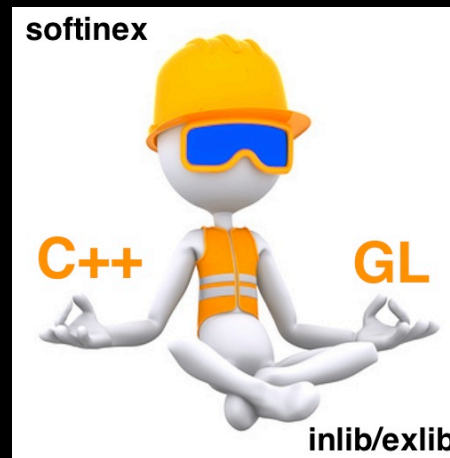
- **Surtout déploiement facile** quasi mondial grâce à l'Apple Store et Android Market (maintenant Google Play) : **énorme progrès !**
- 99% du code commun à toutes les plateformes.

ioda / softinex

- IODA évolue gentiment vers un visualisateur de divers formats de fichiers :
 - HEP : **root** (histogrammes, ntuples, géométries), **gdml** (géométries Geant4), **heprep** (géométries).
 - Astro : **fits** (images astro, tables), **hdf5** (histogrammes, ntuples).
 - Aussi : **VRML** (avec Coin3D finalement porté sur iOS et Android), **png**, **jpeg**.
- On a fait le portage : **cfitsio**, **hdf5**, **Coin3D**, **png**, **jpeg**, etc...
- On **thésaurise** tout cela dans les bibliothèques **softinex** (**inlib**, **exlib**, **ourex**). <http://softinex.lal.in2p3.fr>

softinex

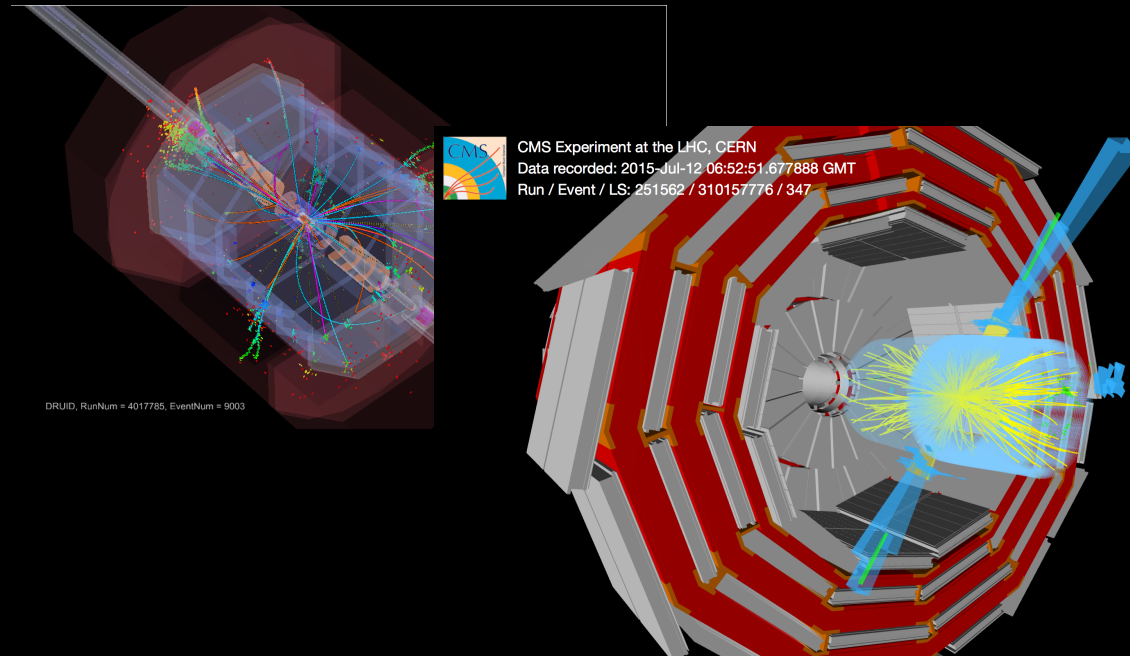
- **inlib** : code C/C++ « pure header » **ultra portable**.
- **exlib** : code C/C++ faisant le lien avec les « packages externes ».
- **ourex** : versions de packages externes « critiques » comme freetype, expat, png, jpeg, zlib, zip dont on maîtrise, embarque les sources. Mais aussi cfitsio, hdf5, Coin3D, lua, Python (2.7) et... Geant4.
- Dans softinex, il y a des choix de développements forts... (quasi philosophiques ☺)



Lien avec les expériences

ioda, g4view apps génériques, mais on aimerait faire des apps « dédiées » pour les expériences, en particulier des « event display » exploitant les capacités locales (graphiques) des devices.

- pmx : LHCb.
- agora : ATLAS.
- nemoview : NEMO3.
- TouchSky : LSST.



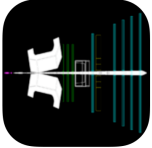
Là, ça coince....

app démonstrateur qui peut montrer le détecteur LHCb (lu depuis un fichier root) et montrer quelques traces.

pmx Plus par cet éditeur

par **Guy Barrand**

Ouvrez iTunes pour acheter et télécharger des apps.



Description

pmx is an event display for the LHC/LHCb experiment. With it you can view the detector, the magnetic field and load events. It can give to someone novice in high energy physics (HEP) a glance at what a HEP detector looks like.

Assistance : [pmx](#)

Nouveautés de la version 1.2

New sub detectors. Visualize the magnetic field. You can have a "manip" on the mag field cut plane and move the cut plane in the field. New GUI look and feel (round corners buttons).

[Afficher sur iTunes](#)

⊕ Cette app a été conçue pour iPhone et iPad.

Gratuit

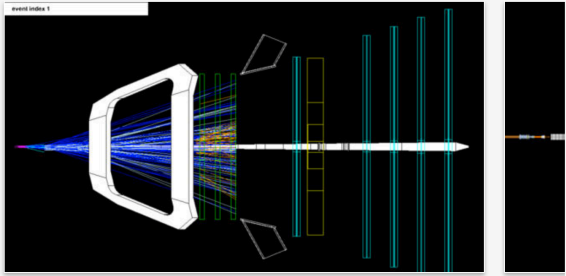
Catégorie: [Utilitaires](#)
Mise à jour : 31 mars 2014
Version : 1.2
Taille : 35.1 Mo
Langue : Anglais
Éditeur : Guy Barrand
© 2012 Guy Barrand
[Classé 4+](#)

Compatibilité : Nécessite iOS 5.1.1 ou une version ultérieure. Compatible avec l'iPhone, l'iPad et l'iPod touch.

Note

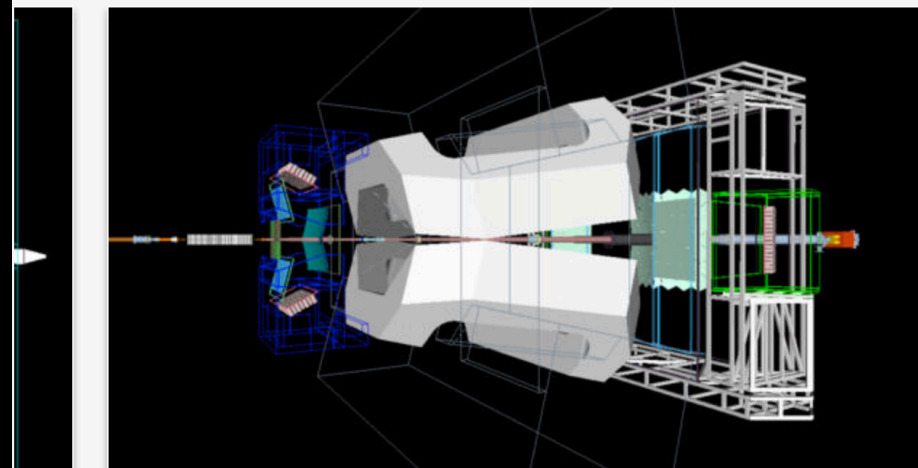
Nous n'avons pas reçu suffisamment de notes pour évaluer la moyenne de la version actuelle de cet article.

Captures d'écran iPhone | iPad

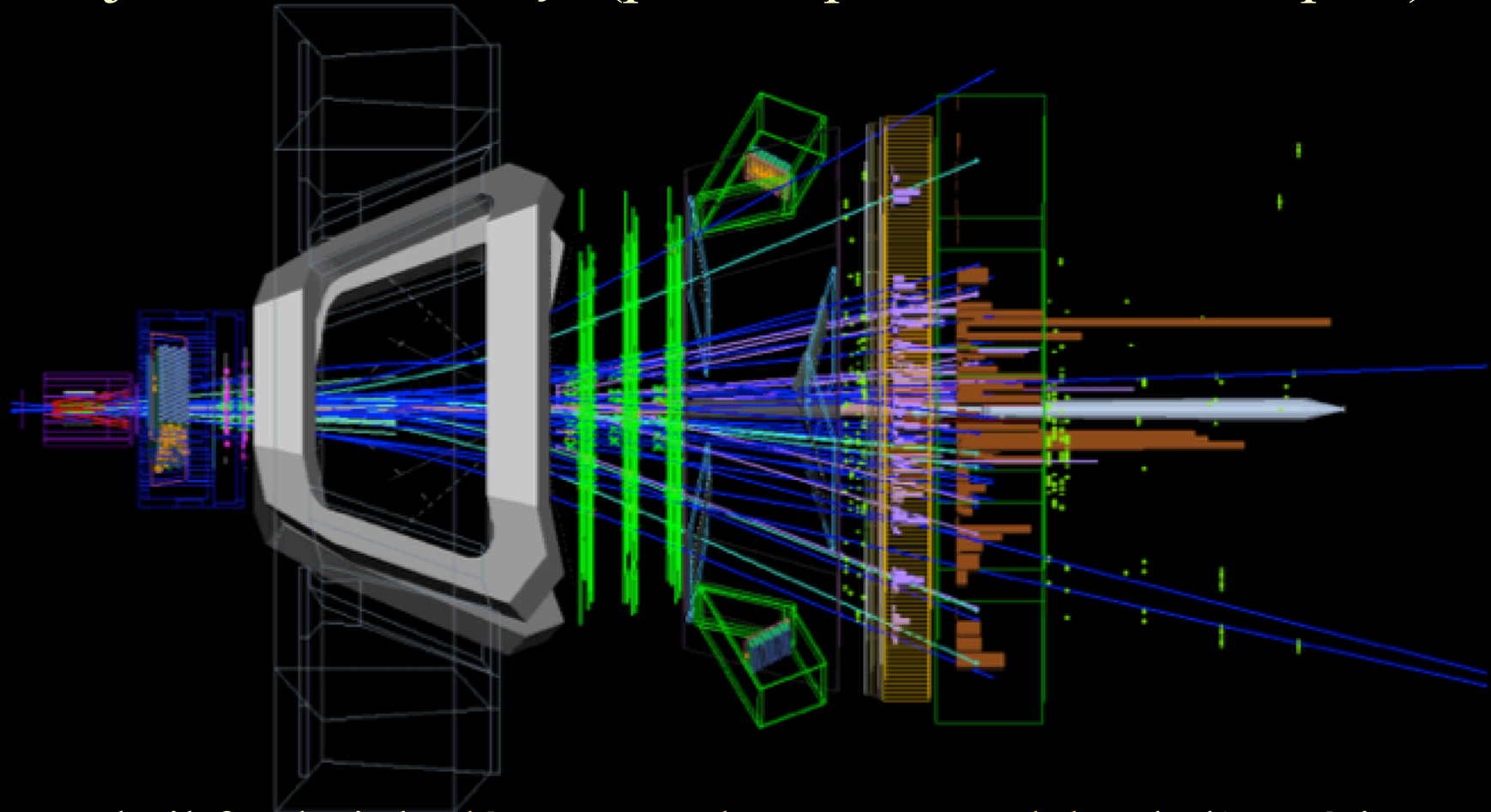


Captures d'écran

iPhone | iPad



mais j'aimerais avoir ça (produit péniblement sur lxplus) :



Pour cela il faudrait le détecteur et le « event model » de l'expérience
et JE NE LES AURAI PAS ! ☹

Pourquoi ?

Software philosophies ...

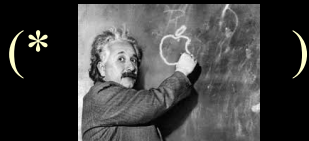
- Approche Applelienne du net : on donne aux gens des devices survitaminés : l'intelligence et la puissance de visu et interactive sont dans les mains des gens, seules certaines données sont dans les serveurs. On tient beaucoup à la « great user experience » avec les devices.
- Approche Googlelienne du net : tout est dans les serveurs (chez Google !) (et donc à la NSA ☺ ☹), les données mais aussi l'intelligence. Pour la visu et l'interactif on ne donne aux gens que des devices « montrant une image finale » (genre Kindle).

Je suis un Apple Guy ! ☺

Hélas HEP est Googlelien ! ☹

Le « **batch aveugle** » est prépondérant : ceci vient du fait de la nature de la physique qu'on y fait qui est fondamentalement statistique (Heisenberg ! *). Il faut « mouliner » beaucoup d'événements et ceci se fait « côté serveurs » (sur les grilles par exemple).

Gros impact sur les softwares qui sont des sortes de gros « **patchwork** » hétéroclites ultra non portables ! (On ne peut même pas dire qu'ils sont portables Linux puisque fortement dépendants d'une distribution, voire d'une version d'une distribution !).



Hélas HEP est Googlelien ! ☹

Or un HEP event display a besoin du détecteur et de l'event modèle de l'expérience qui sont, hélas, « immergés » dans ces softwares « batchiens » .

Il faudrait faire l'effort de « dégager », « extraire » ces parties pour pouvoir les embarquer sur les devices et faire du display en local. (Seul les « fichiers d'événements » seraient côté serveur).

Ça bloque sociologiquement

- Techniquement isoler le détecteur & event modèle est possible ; ce n'est, après tout, que du C++, dont on a le code source, à réorganiser.
- Mais il y a clairement des blocages sociologiques, en particulier dans des labos à l'esprit foncièrement Googlelien (CERN).
- HEP se dirige vers « du tout web », y compris pour l'interactivité et les displays avec, hélas, des applis (web) foncièrement peu réactives « au final » pour l'utilisateur (forever poor user experience !).

Domage, car déjà en 2012 on aurait pu mettre le Higgs dans la poche d'une partie de l'humanité : gros raté de com pour le CERN & Co !

Interactivité par le web ?

- Qu'on se débrouille comme on veut, une approche web accumule les couches matérielles et logicielles entre les données et l'utilisateur !
Et chaque couche introduit une perte de **réactivité** de l'interactivité !

Ça marche mais ça rame ! ☹️



- Et la dernière tendance « cloud » à passer, coté serveur, par des usines à gaz à machines virtuelles, en rajoute une grosse couche !
- Pas mal de gens ont compris cela et ont maintenant des **apps dédiées** : e-commerçants, banques, snCF, transilien, vélib, la centrale, programme télé, etc...

Interactivité par le web ?



HEP ... soupirs, gros soupirs



Workshop Gaudi 21/Sep/2016

Simplifying development and compilation

Developers would like to be able to compile Gaudi on their own machine

Action item 8:

Port Gaudi to Mac

Action item 9:

Port Gaudi to Ubuntu

Action item 10:

Separate optional subsystems and introduce configuration switches



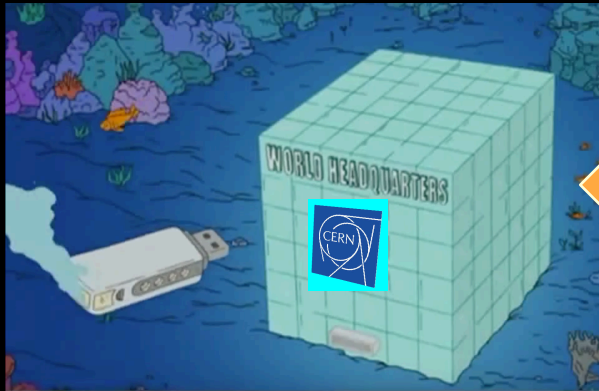
Enfin ! Et iOS, Android donc pour... 2050 !
Mais le Mac est plus scandaleux...

Rodrigue, as-tu des coeurs ?

Multicore, GPU, HSF : les buzzwords du moment ! Et hop on rajoute une grosse couche de complication ! Pour gagner quoi ? Un facteur 2 ? (L'essentiel de là où on perd du temps en HEP n'est pas parallélisable ni vectorisable ! Le Cray du CERN n'a pas révolutionné grand chose)

Et le code ne tournera plus que sur un seul type de processeur et donc une seule plateforme : le CERN grillaplus.

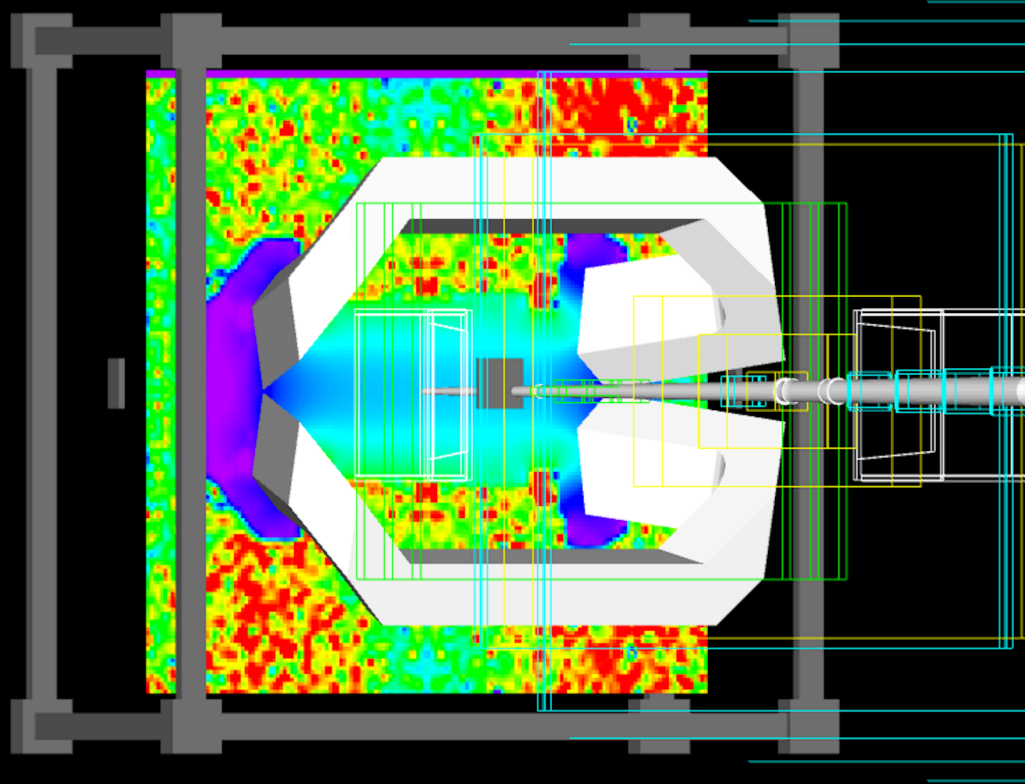
Et HEP aura parachevé sa Googlelisation



Data center sous le lac Léman

(Youpi, l'iPhone7 est water proof !)

Un histoire... magnétique



pmx : lecture de la « carte du champ » : 30 secs => 5 secs ! Facteur 6.
Au lieu de supercomputers, il faudrait équiper les développeurs d'
« HEP framework » d'iPads !

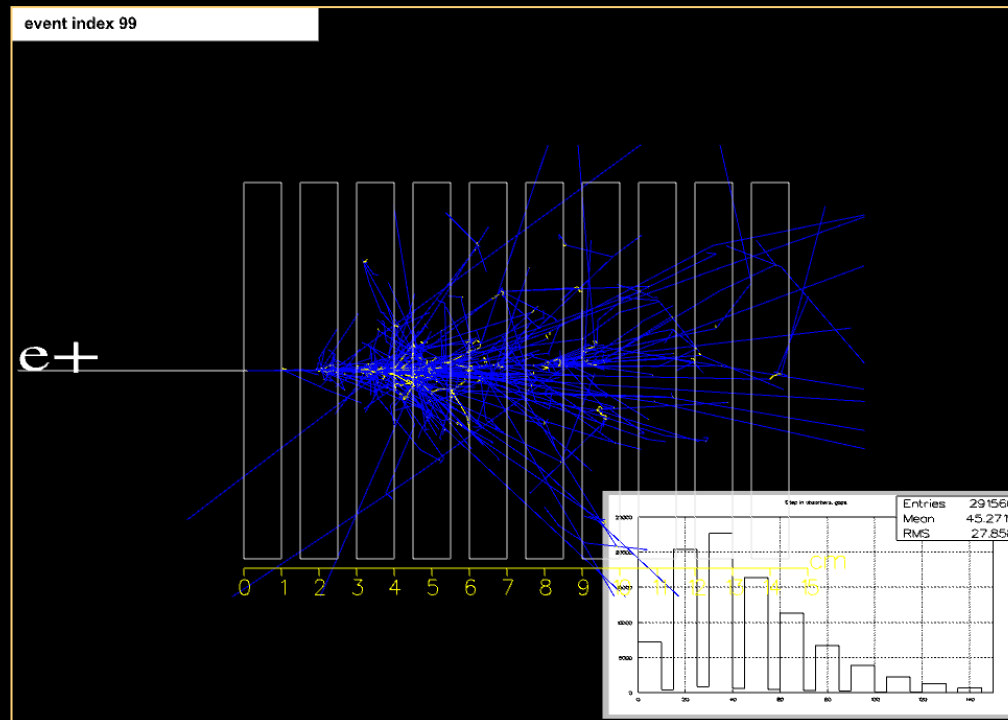
Petites comparaisons...

- Le .apk de pmx pour Android est de 25 Megas, temps de démarrage rapide (< 3secs), code ultra portable, 3D fluide (exploitant le GPU local).
- Ceci est à comparer à un display fait sur un patchwork batch où la distrib fait... 30 Gigas ! (3x la taille d'un OS comme macOS !), temps pour avoir le GUI de plus d'une minute ! (si,si), et intrinsèquement non-portable (sur Mac, il faut passer par une machine-à-gaz virtuelle). ☹

(La règle des “3 secs” : si après un click/touch l'utilisateur a un retour après + de 3 secondes, il devient nerveux...)

Le Grand Geant(4)

- Geant4 : flagship de la simulation en HEP
- g4view sur les stores en 2012. Quasiment pas de résistance au niveau du code de Geant4 lui même ! Mais iOS & Android toujours ignorés par la « collaboration » et le groupe « visu » en particulier ! ☹



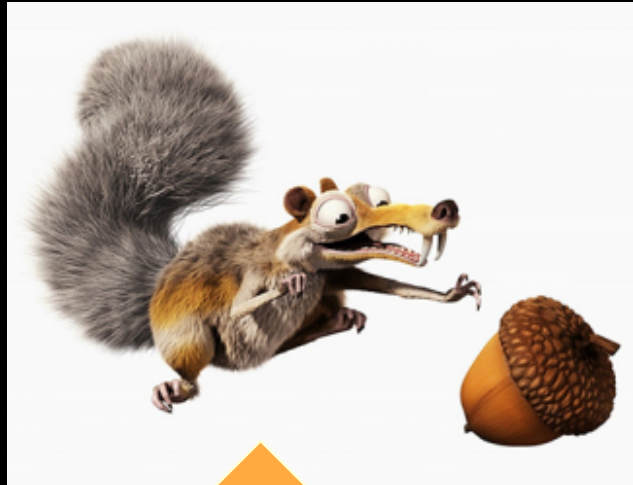
Les « collaborations » ?

Grosse collaboration  Grande qualité du code

- Collaboration permet d'être complet (exa : description du détecteur, processus de physique d'un simulateur), mais conduit à un assemblage « patchworkien » hétéroclite ingérable sur le long terme.
- Les outils (genre « intégration continue » *) ne suffiront pas à élever le niveau car ils ne règlent pas les problèmes de sociologie (exa : personnes productives n'ayant pas les mêmes « visions » des choses).

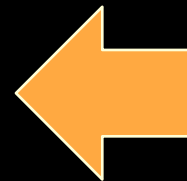
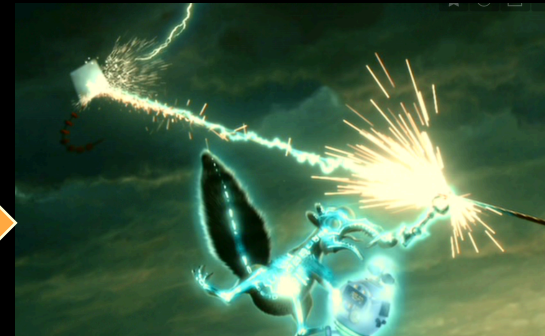
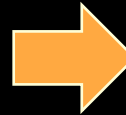
* moi j'utilise des outils de « désintégration quantique » ☺

HEP in a nut... shell !



Euh... moi !

qui se prend la foudre...



« événement » pris dans une coquille logicielle dure et épaisse (ROOT+Gaudi+...).



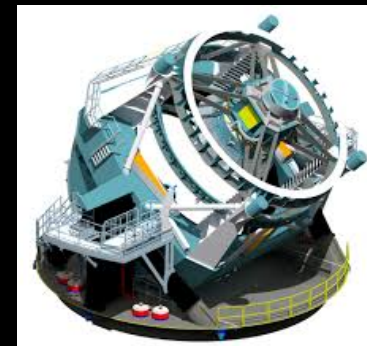
Vers l'infini...ment grand



On regarde vers le ciel !

Je change d'échelle : $10^{-15} \Rightarrow 10^{+15}$!!! *Astro/LSST.*

Côté astro, déjà, la visualisation est d'entrée de jeu dans « l'équation software » de départ. (La visu est dans la cahier des charges de départ de LSST par exemple).



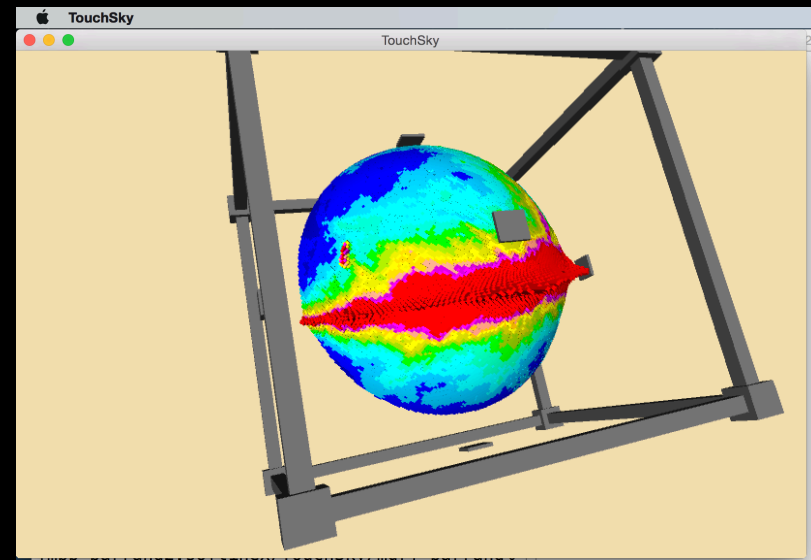
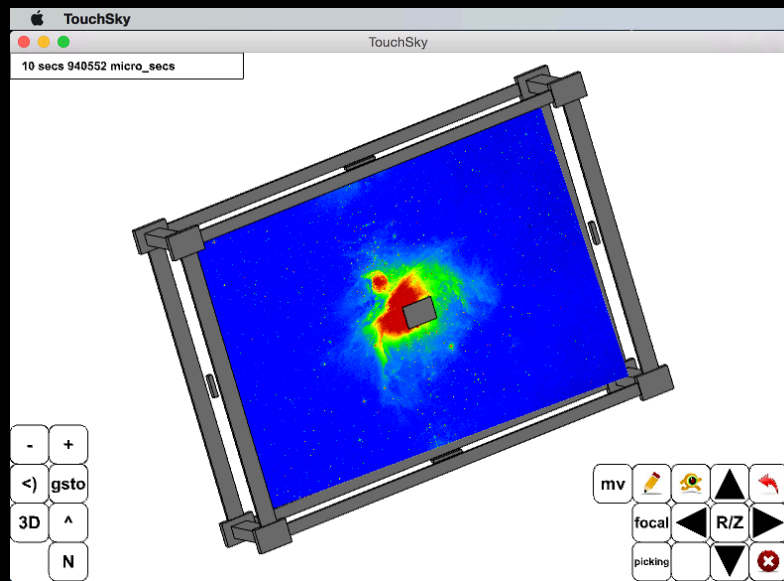
Astro et interactivité

- Communauté (amateurs en particulier) **Applelienne** très active sur iOS et Android ! Avec des apps exploitant par exemple la localisation et l'orientation de l'appareil (j'adore mon « SkyMap »).
- **MAIS l'esprit Google est très présent sur les grands instruments et autour des « surveys ».** Il y a clairement une approche « tout web » pour la visu sur LSST. (Leur « stack » est clairement « batchien »).
- **Mariage/jonction possible ?** Si oui, cette communauté à les cartes en main pour faire des choses très « sexy ».

Une grosse différence avec HEP, est de ne pas être coincé par un event et détecteur modèle « pris » dans la glace ! Et que le **format de fichier des données (du fits) est lisible partout avec du soft léger dédié à l'IO.** (Mais attention aux « catalogues »...)

TouchSky

- TouchSky : app « modeste » pour se faire la main sur ce genre de données : lecture de fits, visualisation de HEALPIX, projection « calotte sphérique dynamique », etc...
- Visualisation d'une « grande image » (40x40k) par navigation dans une pyramide multi-résolution tuilée (si, si ça existe)
- Mais connexion avec les surveys pas très claire pour l'instant...



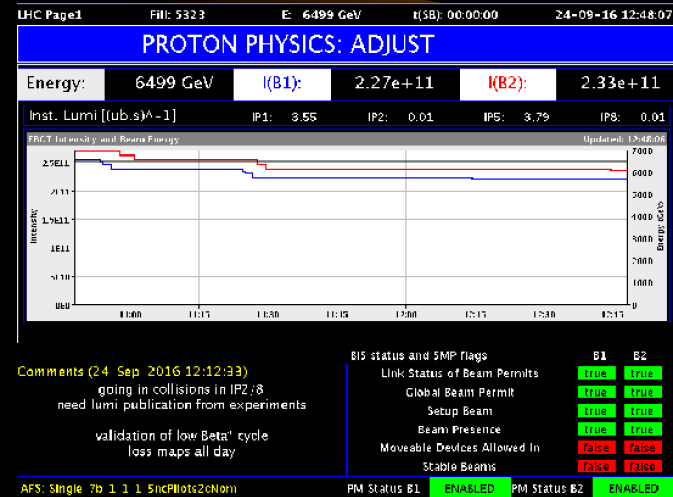


Mes apps

- ioda : visualisateur “analyse”.
- pmx : démonstrateur (hélas) d’event display pour LHCb.
- agora : démonstrateur (hélas aussi) d’event display pour ATLAS.
- g4exa : un exemple Geant4 “cablé”.
- g4view : un viewer Geant4.
- **LHCStatus** : montre le status du LHC : appli la + simple et la + installée ! Mais rejetée par Apple !
- TouchSky : astro.
- ArcheryTune : app “hors boulot” faite avec JE.Campagne pour le tir à l’arc.
- Difficile d’avoir des stats d’**utilisation**, on va dire entre 500 et 1000 **installations** (des personnes qui ont gardé une appli).

LHCStatus, un mot...

LHCStatus : montre le status du LHC



appli la + simple et la + installée sous GooglePlay!

Mais rejetée par Apple parce que faisable facilement en HTML : de quoi je me mêle !? Je ne veux pas passer par un web browser !

Mais dispo sous agora et ioda/exemples (non mais).

<http://softinex.lal.in2p3.fr>



- Voir portail softinex.
- Voir démos d'utilisation sur YouTube (chercher « Guy Barrand »)
- (mais attention, j'ai un homonyme Anglais qui fait des exposés de finance !) (mais moi, je pourrais vous expliquer comment en être de votre poche pour faire de la science 😊).

Futur ?



- On continue l'effort...
- Peut être un rayon de soleil en HEP autour d'ATLAS & masterclass... (grâce à un format de fichier XML simple).

Réalité Virtuelle ?

- La **VR en carton** : bien adaptée au budget « visualisation/interactivité » de mon labo Googolo-HEPien ☺ . J'ai commencé à écrire du code...



- Tel, Tablette comme base d'extensions immersives ?



On peut rêver ...

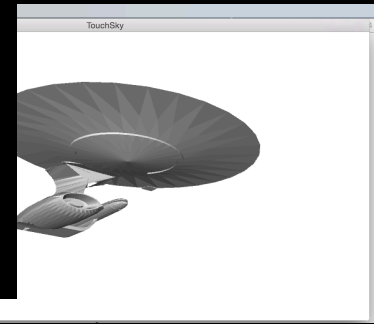
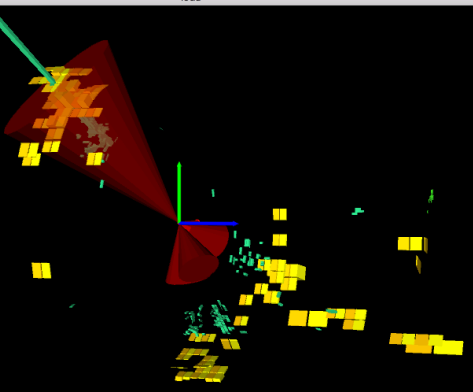
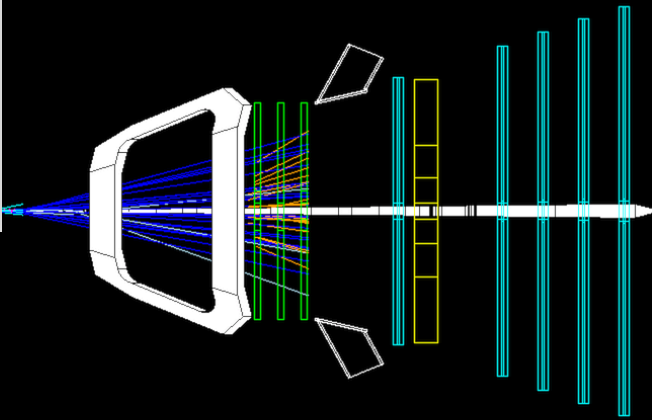
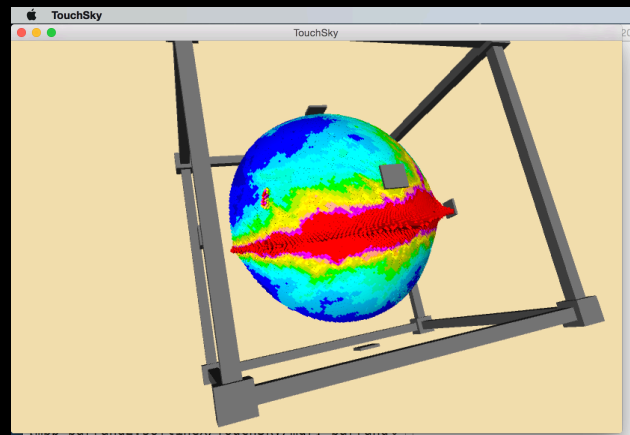
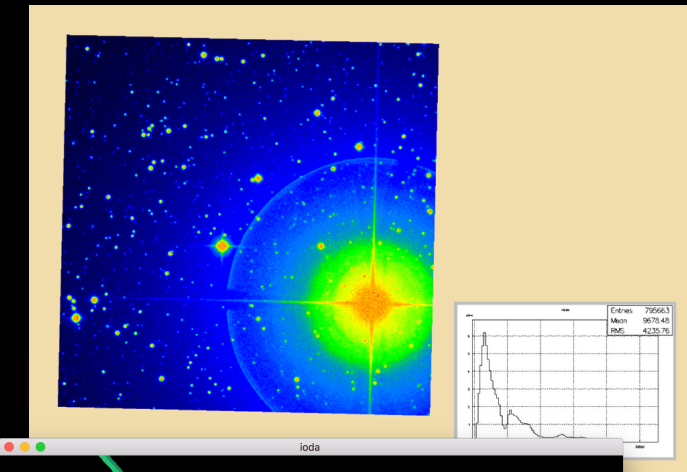
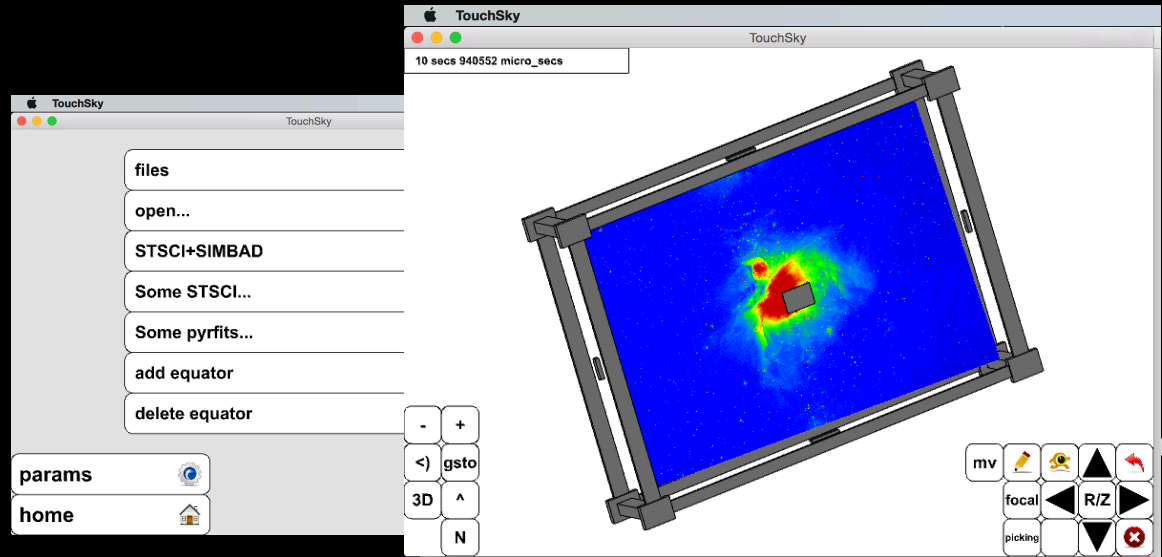


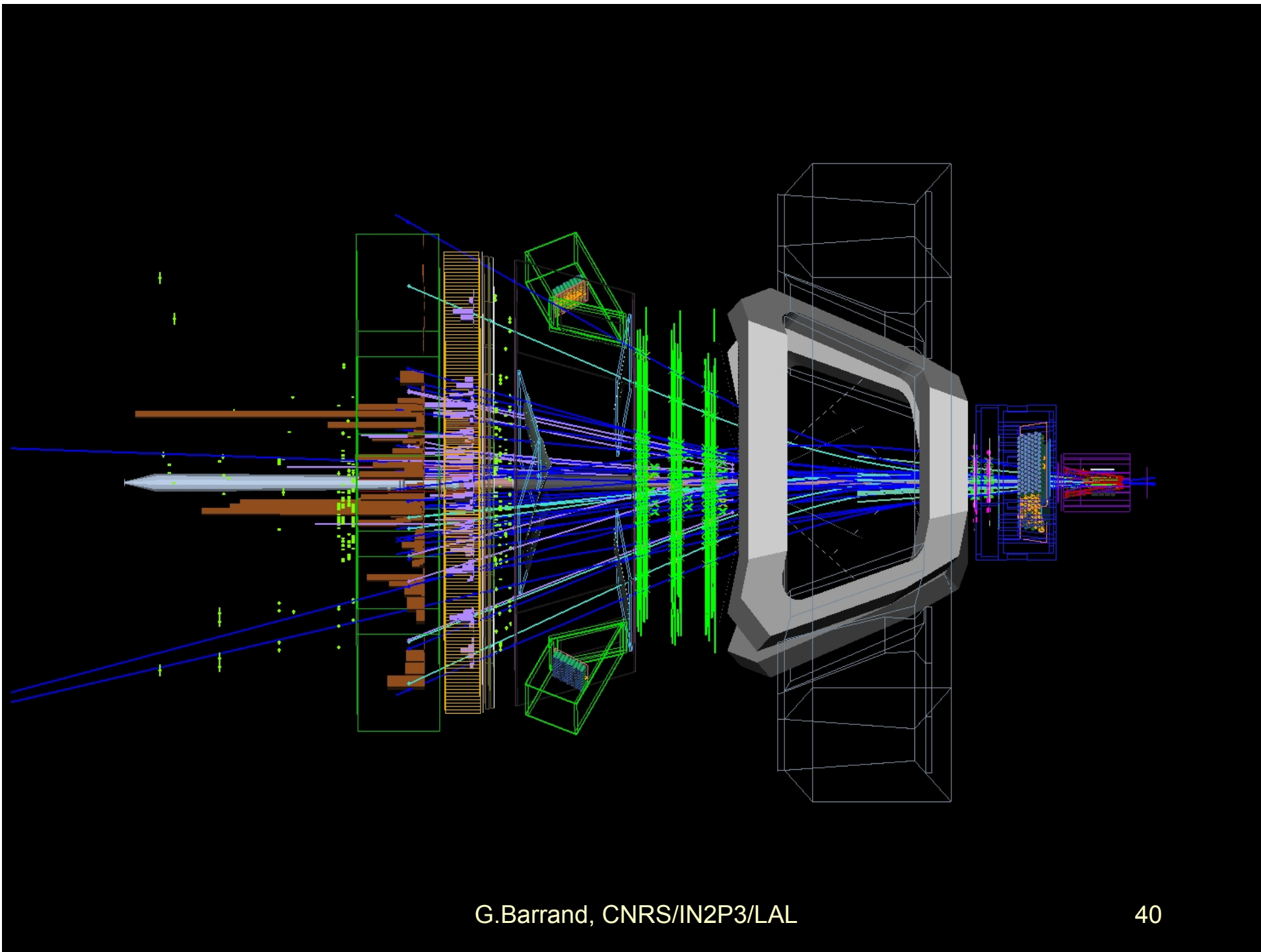
Conclusions

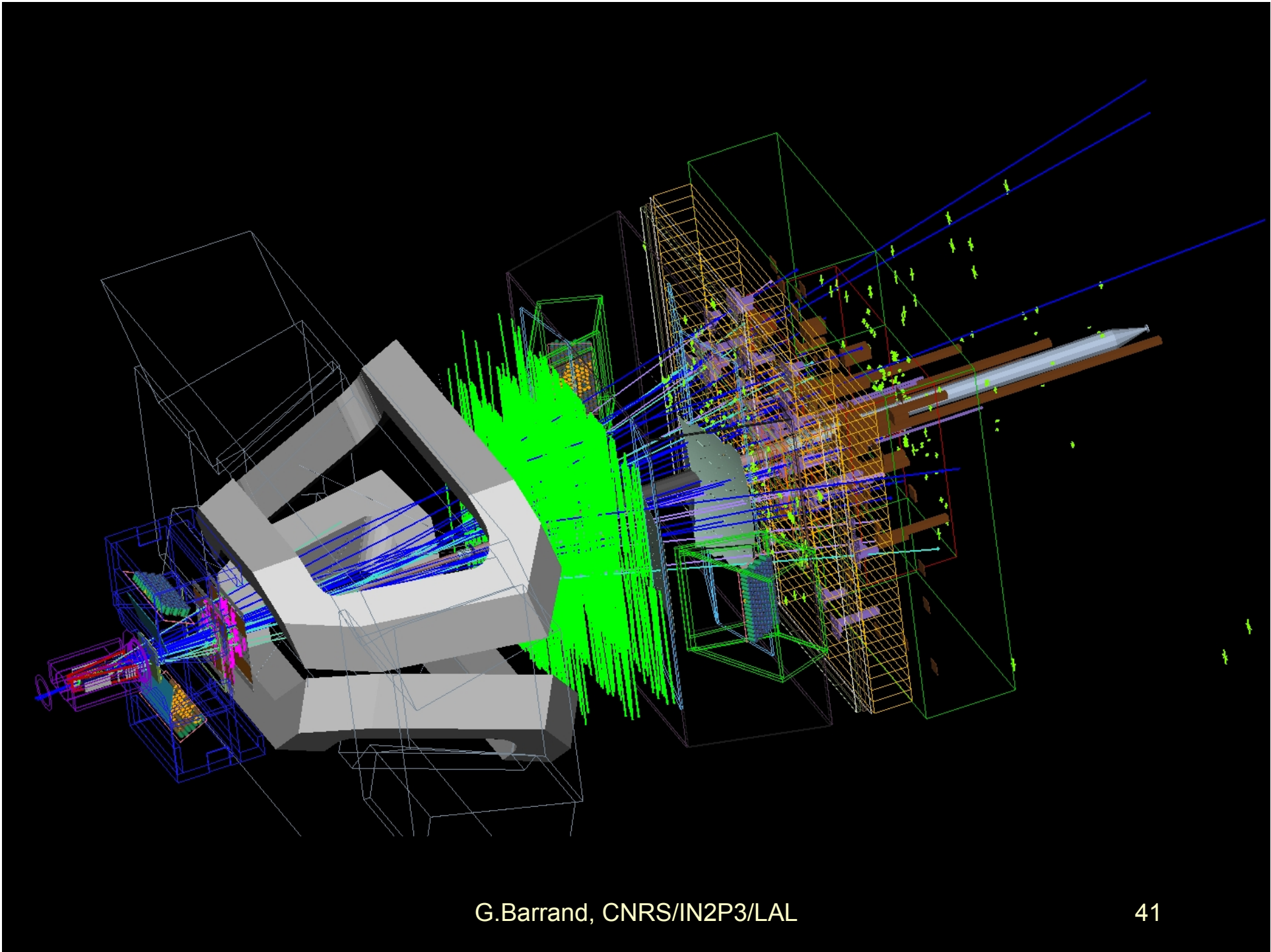
- Interactivité : **réactivité** + ergonomie + portabilité.
- Il y aura de plus en plus de puissance (et de stockage) dans la poche des gens : **il serait profondément stupide d'ignorer cela**, au moins pour des aspects de com !
- HEP... ? **Soupir**. Au moins un Guy essaie...
- Astro, autres domaines de science : probablement là que des choses très sympas se font et vont se faire !
- La visu doit être dans l'équation de départ...
- Les choses bougent (encore). La VR ? La réalité augmentée ?

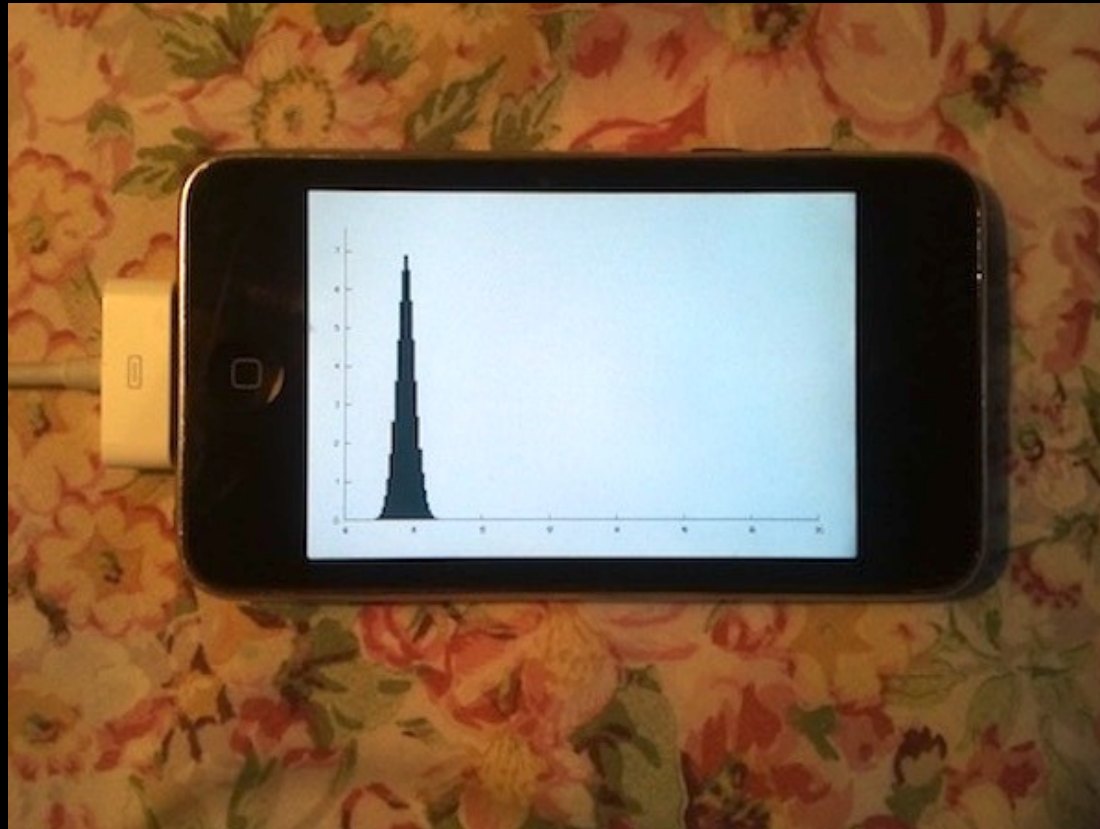


Un florilège de desserts









Premier plot sur un iPod touch, 12 Novembre 2010.