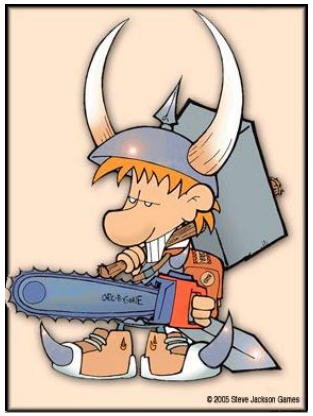
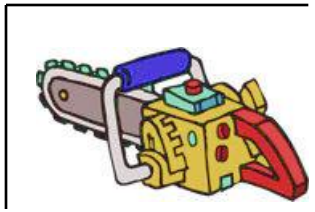
TSkim

Un outil pour découper les arbres ROOT



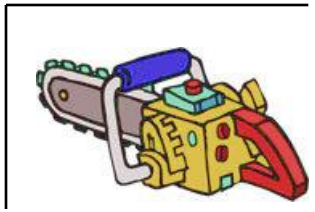
Introduction

- De nombreuses expériences de physique stockent leurs données sous forme d’arbres ROOT.
- Une activité récurrente des chercheurs consiste à établir des critères pour définir quels événements physiques sont intéressants, et à extraire des fichiers ROOT les données attachées à ces événements particuliers .
- TSkim a été développé pour les y aider. Initialement un ensemble de scripts PERL et ROOT pour l’expérience FERMI, l'outil est à présent devenu suffisamment général et élaboré pour s’ouvrir à d'autres expériences, notamment CMS.



Plan

- Motivation
- Utilisation
- TSkim pour CMS
- Epilogue

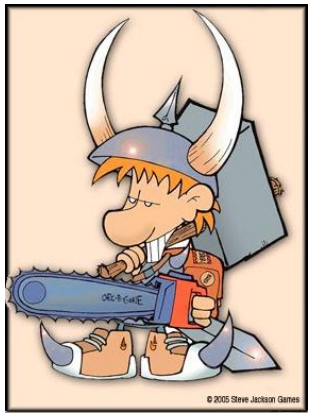


Motivation

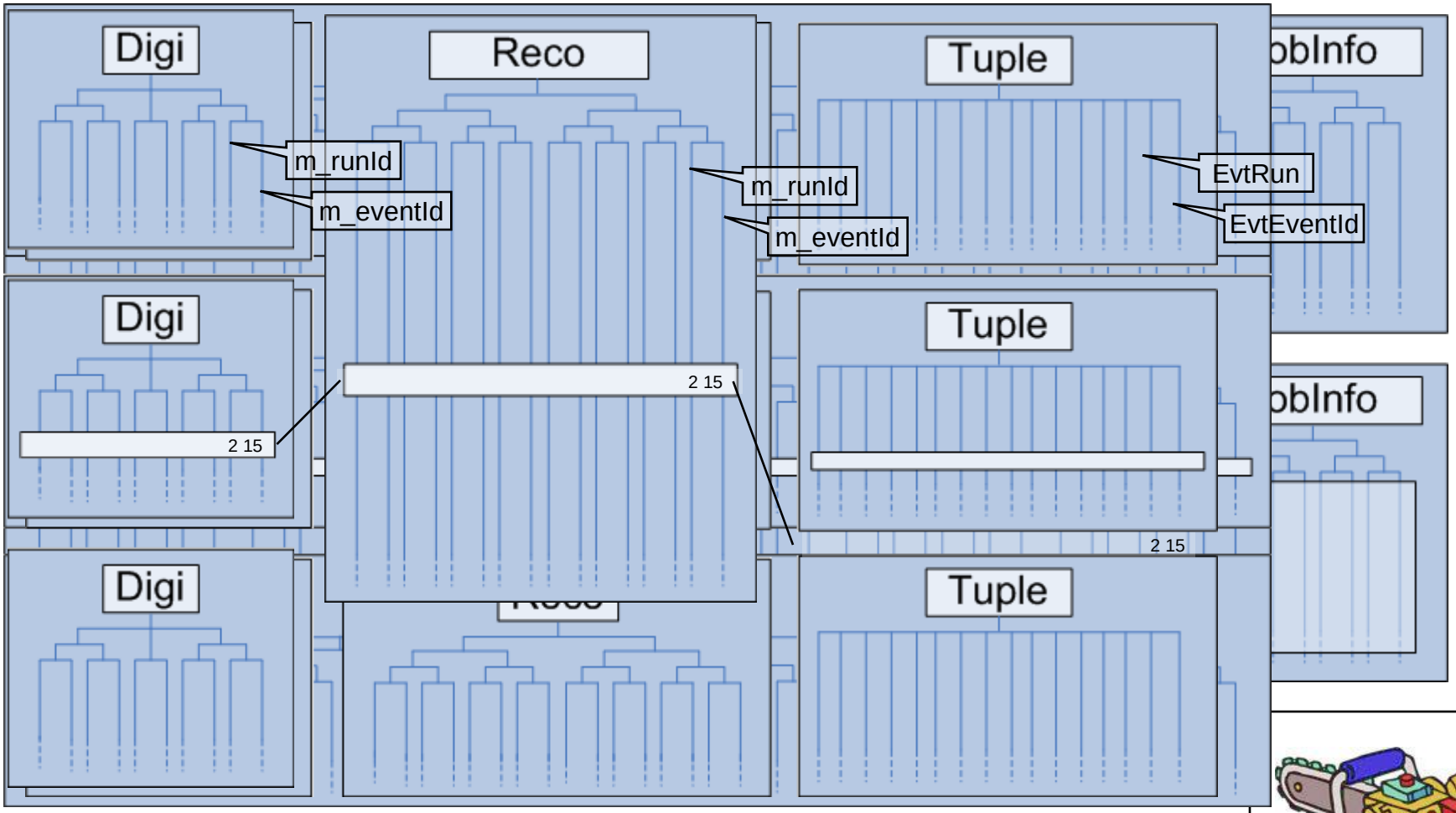
Faciliter la manipulation de données complexes

Factoriser les bonnes pratiques ROOT

Réduire la duplication répétée des données à l'aide de listes

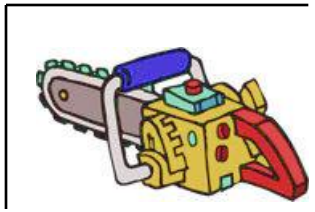


Organisation des données



Bonnes pratiques ROOT

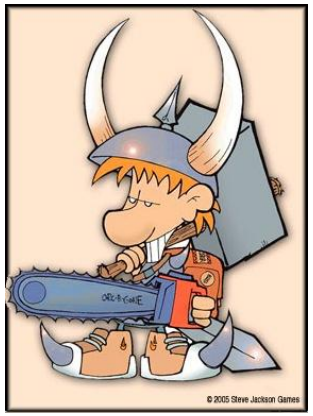
- Optimiser les performance :
  - s'efforcer de ne lire que les branches utiles,
  - s'efforcer de ne lire chaque fichier qu'une seule fois,
  - sélectionner l'option « Fast » quand c'est approprié.
- Détecter certains cas pathologiques :
  - des changements inattendus dans la structure des branches,
  - des Run/Event IDs dans le désordre,
  - des Run/Event IDs trop grands.



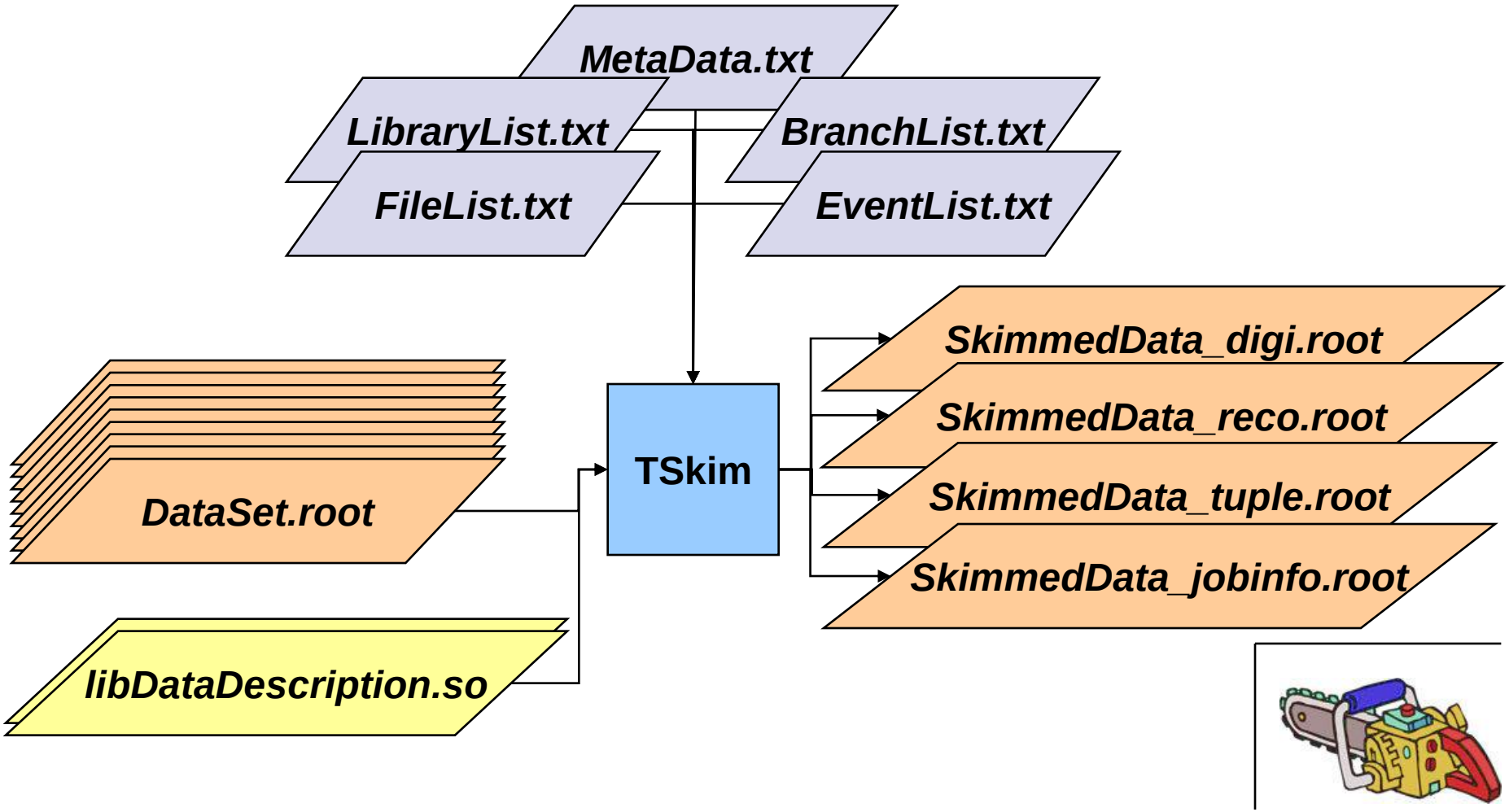
Utilisation

Configuration et entrées/sorties.

Détails sur les fichiers de configuration, les variables d'environnement, les executables externes.



Config/Entrées/Sorties de TSkim



Exemple de job

csh> setenv **ROOTSYS** /…/v5.18.00c/root

csh> setenv **TS_META_DATA** .../MetaData.txt

csh> setenv **TS_INPUT_LIBRARY_LIST** .../LibraryList.txt

csh> setenv **TS_INPUT_FILE_LIST** .../FileList.txt

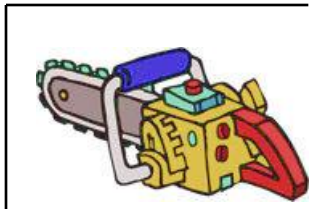
csh> setenv **TS_INPUT_BRANCH_LIST** .../BranchList.txt

csh> setenv **TS_INPUT_EVENT_LIST** .../EventList.txt

csh> setenv **TS_DATA_TYPES** "digi:reco:tuple:jobinfo"

csh> setenv **TS_OUT_DIR** $PWD

csh> **tskim**



Types de données

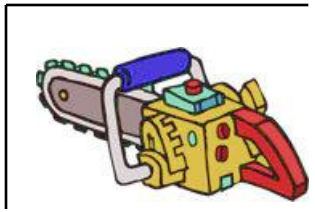
MetaData.txt

#! SECTION MetaData

(**tuple.treeName**) Tuple
(**tuple.runIdBranchName**) EvtRun
(**tuple.eventIdBranchName**) EvtEventId

(**jobinfo.treeName**) jobinfo

(**digi.treeName**) Digi
(**digi.runIdBranchName**) m_runId
(**digi.eventIdBranchName**) m_eventId
(**digi.topBranchName**) DigiEvent
(**digi.topBranchType**) DigiEvent
(**digi.libName**) libdigi.so



Fichiers ROOT d'entrée

FileList.txt

#! SECTION Files

(**tuple:jo**…_tuple.root
(**tuple:jo**…_tuple.root
(**tuple:jobinfo**)…
(**tuple:jo**…8/Zjets-10_0008_tuple.root
(**tuple:jo**…8/Zjets-10_0009_tuple.root

(**digi**)/nf…digi.root
(**digi**)/nf…digi.root
(**digi**)…
(**digi**)/nf…digi.root
(**digi**)/nfs/u35/dc08/Zjets-10_0009_digi.root

TS_DATA_DIRS = /nfs/u35 : … : …

root://… rfio://…

FERMI_TASK = dc08/Zjets
FERMI_RUN_MIN = 0
FERMI_RUN_MAX = 9

TS_DATA_COMMAND ?

Bibliothèques C++ précom

LibraryList.txt

#! SECTION Libraries

/nfs/u09/builds/rh9_gcc32/DataFormats/310pre7/libcommon.so
(**digi**)/n digi.so
(**recon**)/nfs/u09/builds/rh9_gcc32/DataFormats/310pre7/libreco.so

TS_LIB_DIRS = /nfs/u09/builds/rh9_gcc32/ **: … : …**

TS_LIB_DIR = DataFormats/310pre7

Pour Fermi, mécanisme de détection automatique de la version à partir d'un objet « entête » stocké dans tous les fichiers.

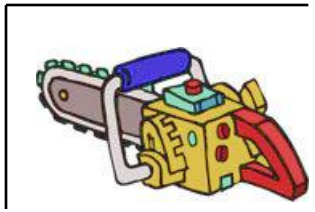
TS_LIB_COMMAND ?

Sélection des branches

BranchList.txt

#! SECTION Branches

(**tuple**) **-***
(**tuple**) +PtT*
(**tuple**) +Cal*
(**digi**) +m_eventId
(**digi**) +m_runId
(**digi**) +m_acd
(**digi**) **-**m_cal
(**digi**) …



Sélection des évènements

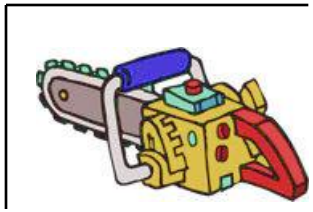
EventList.txt

#! SECTION Events
#! 2000 entries in original dataset.
#! 7 events after cut:

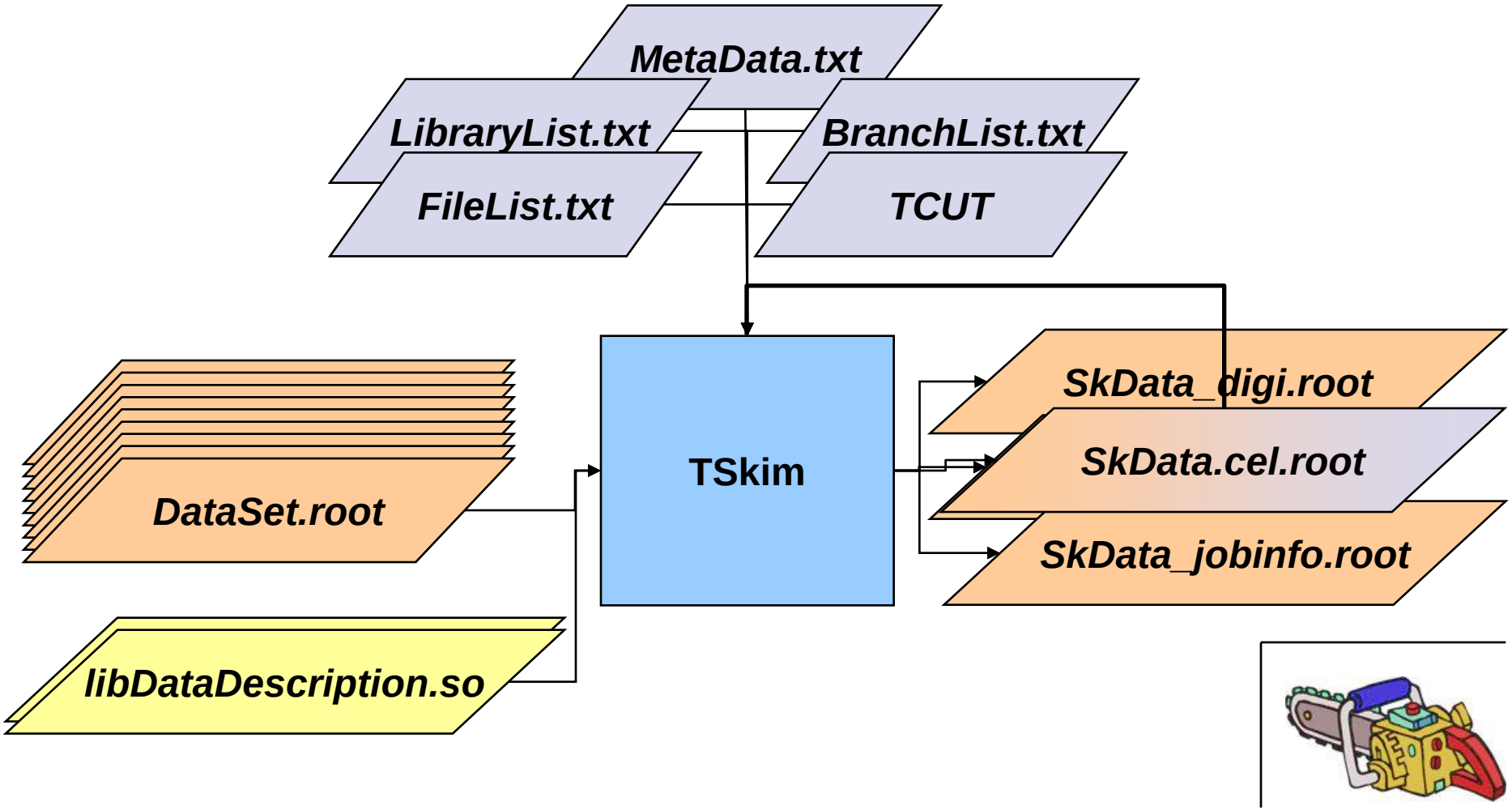
1 8
1 183
1 344
1 553
2 117
2 517
2 980

TS_TCUT = "TkMomentum>200"
TS_TCUT_DATA_TYPE = tuple

En l'absence de TCUT et de liste d'évènements, TSkim fusionne les données.



Utilisation d’un fichier CEL

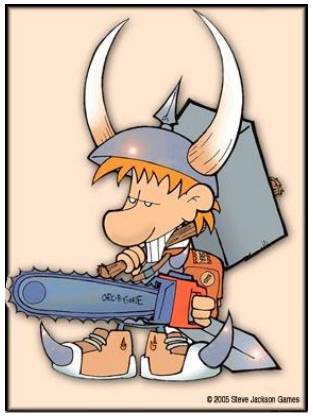


TSkim pour CMS

Pour quoi faire ?

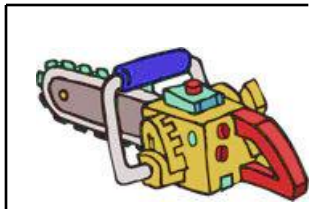
Qu’est-ce qui manque ?

Prototype en préparation



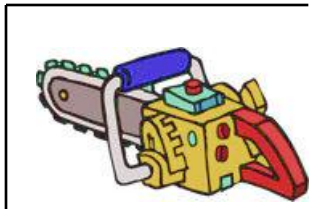
Pour quoi faire ?

- Les productions centralisées de “skims” officiels ?
  - CMS a déjà une infrastructure.
- Les filtrages de données des utilisateurs finaux, à un stade où ils ne sont plus dépendants de CMSSW, et n’utilisent plus que ROOT
  - Vrai besoin pour compenser l’absence de CMSSW.



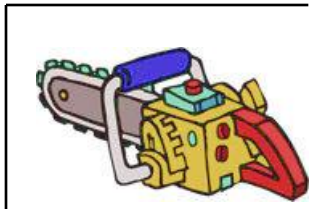
Qu’est-ce qui manque ?

- Une interface utilisateur simplifiée, ne passant plus par des variables d’environnement, mais par des options de ligne de commande ?
- Une connexion avec DBS.
- Une bibliothèque C++ pour manipuler les fichiers CEL.



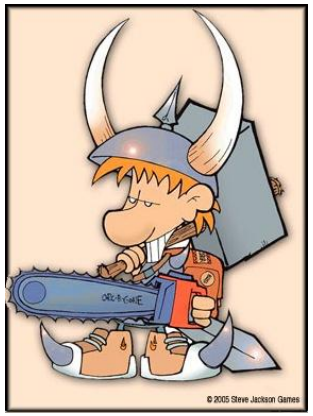
Prototype

```
tskim.py \
-rel 3_1_0_pre7 \
-set RelValQCD_Pt_80_120 -tiers GEN-SIM-RECO \
-cut ‘recoGsfElectrons_gsfElectrons__electrons.@obj.size()>0’
```



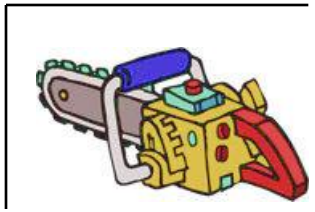
Epilogue

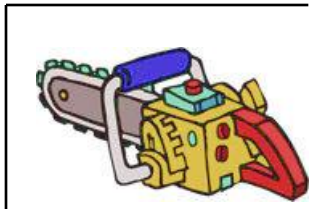
Conclusions
Références



Conclusions

- En production pour FERMI.
- Prototype en préparation pour CMS.
- On recherche des utilisateurs pilotes.



Références

- http://llr.in2p3.fr/trac/tskim
- /afs/cern.ch/user/c/chamont/…
  …/w1/TSkim/bin/tskim.py