



---

# Environnement logiciel *LHCb*

A. Tsaregorodtsev  
*CPPM, Marseille*

Journées informatiques de l'IN2P3 et du DAPNIA  
*Cargèse, 26 Juillet 2001*

# Plan

---



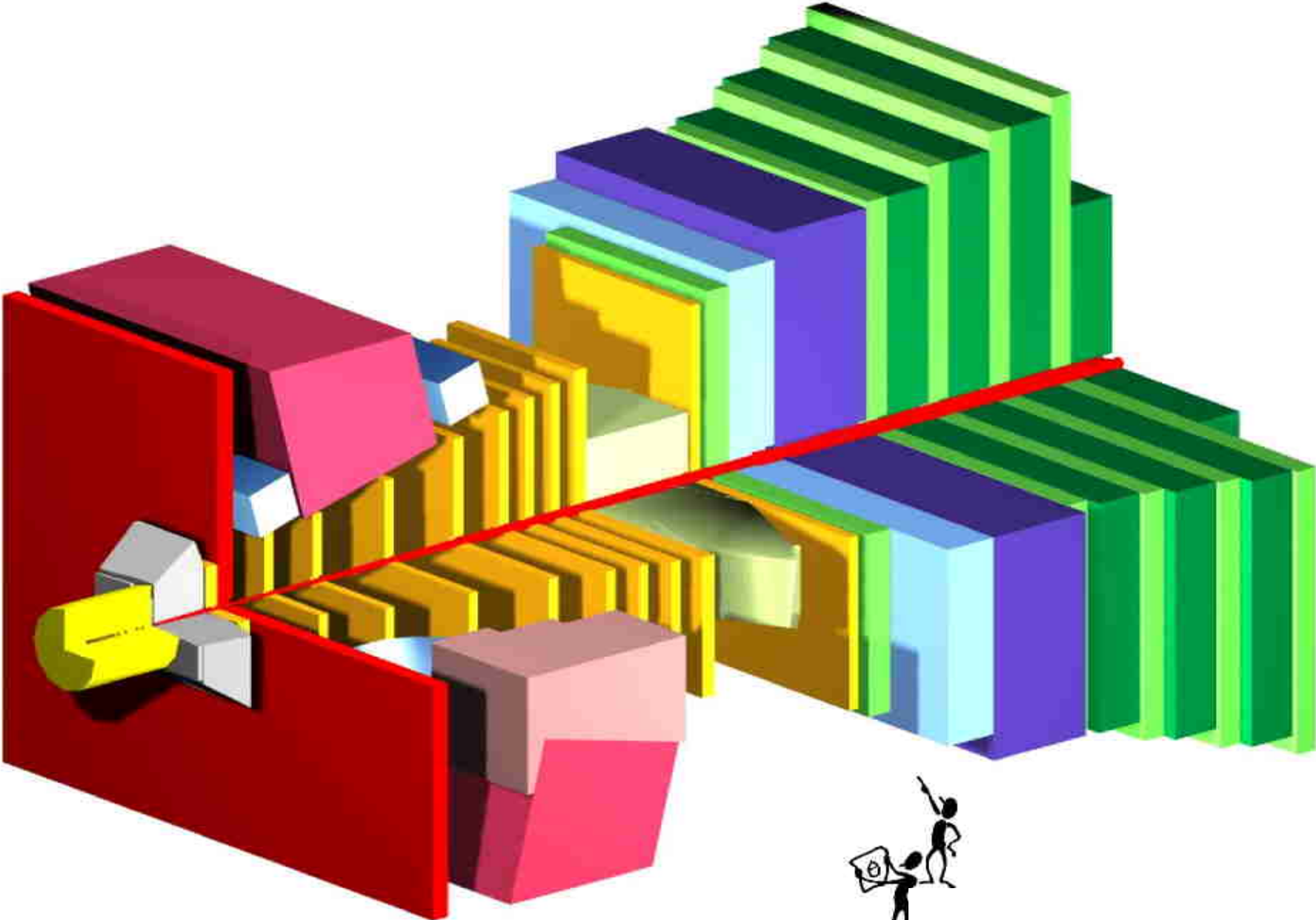
- Introduction de l'expérience **LHCb**
- Défis logiciels
- Solution – **GAUDI**
- Architecture de base
- Composants sélectionnés
- Conclusions

# Expérience LHCb



- Expérience dédiée à l'étude de la violation CP et des désintégrations rares dans le système du méson  $B$
- **LHCb** utilisera **LHC** du **CERN** – la source la plus abondante des mésons  $B_u$ ,  $B_d$ ,  $B_s$  et  $B_c$



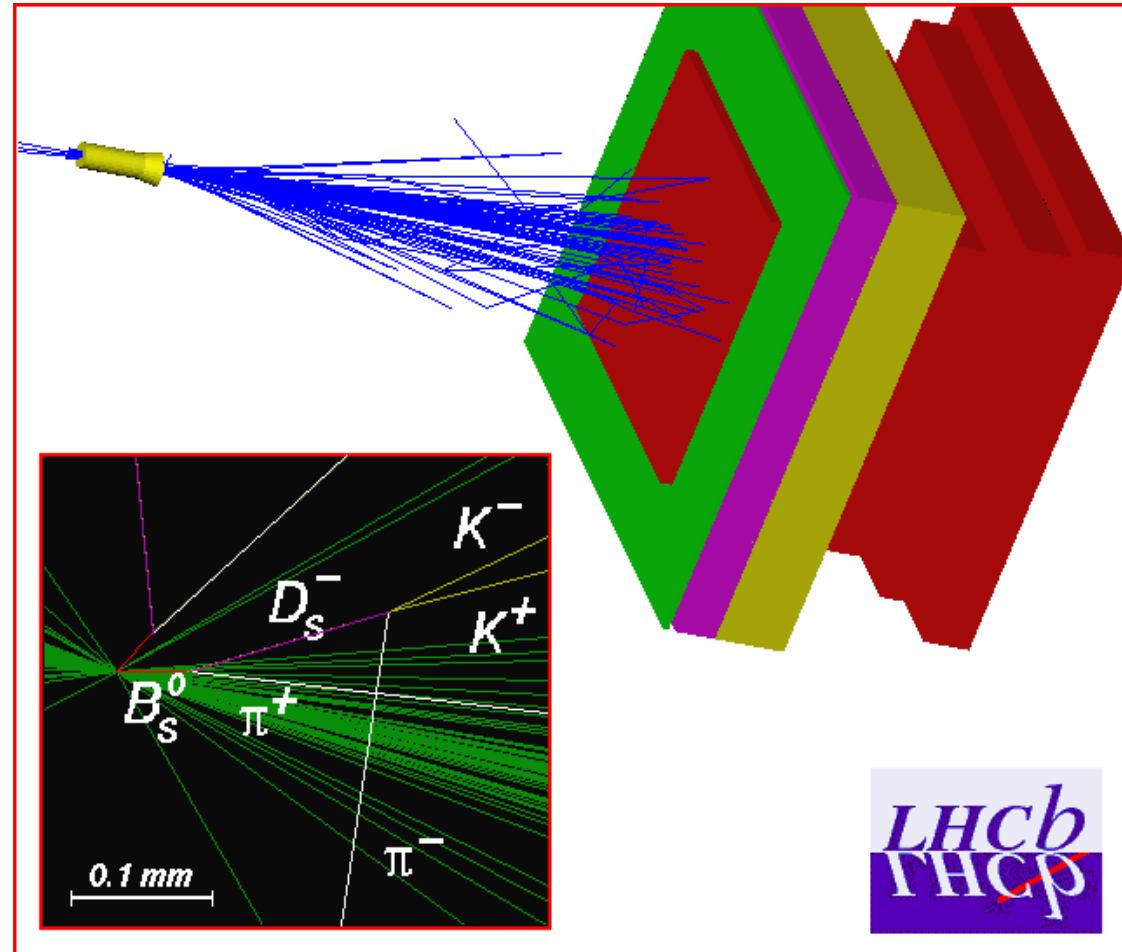


# Detecteur LHCb



## Atouts de l'expérience:

- Le système de déclenchement de haute performance
  - leptons et hadrons de grand  $P_T$
  - vertex secondaire détaché
- Excellente identification des particules chargées
  - séparation  $K/\pi$ :  
 $\sim 1\text{GeV}/c < p < 100\text{GeV}/c$
- Excellente résolution en temps propre du B
  - mixing et violation CP dans le système du  $B_s$



# Detecteur LHCb



- Luminosité basse  $2 \times 10^{32} \text{cm}^{-2} \text{s}^{-1}$ 
  - physique dès le debut du LHC
  - événements propres 0.3/BC
- Le taux de collisions  $pp$  inélastiques –  $\sim 15$  MHz
- Le taux de la production des hadrons  $B$  –  $\sim 75$  kHz
- Le rapport de branchement des canaux intéressants est dans le range  $10^{-5}$ – $10^{-4}$
- Le taux de la physique intéressante  $\sim 5$  Hz

## *Rates*

Bunch crossing rate	40 MHz
Level 0 accept rate	1 MHz
Level 1 accept rate	40 KHz
Level 2 accept rate	5 KHz
Level 3 accept rate	200 Hz

## *Sizes*

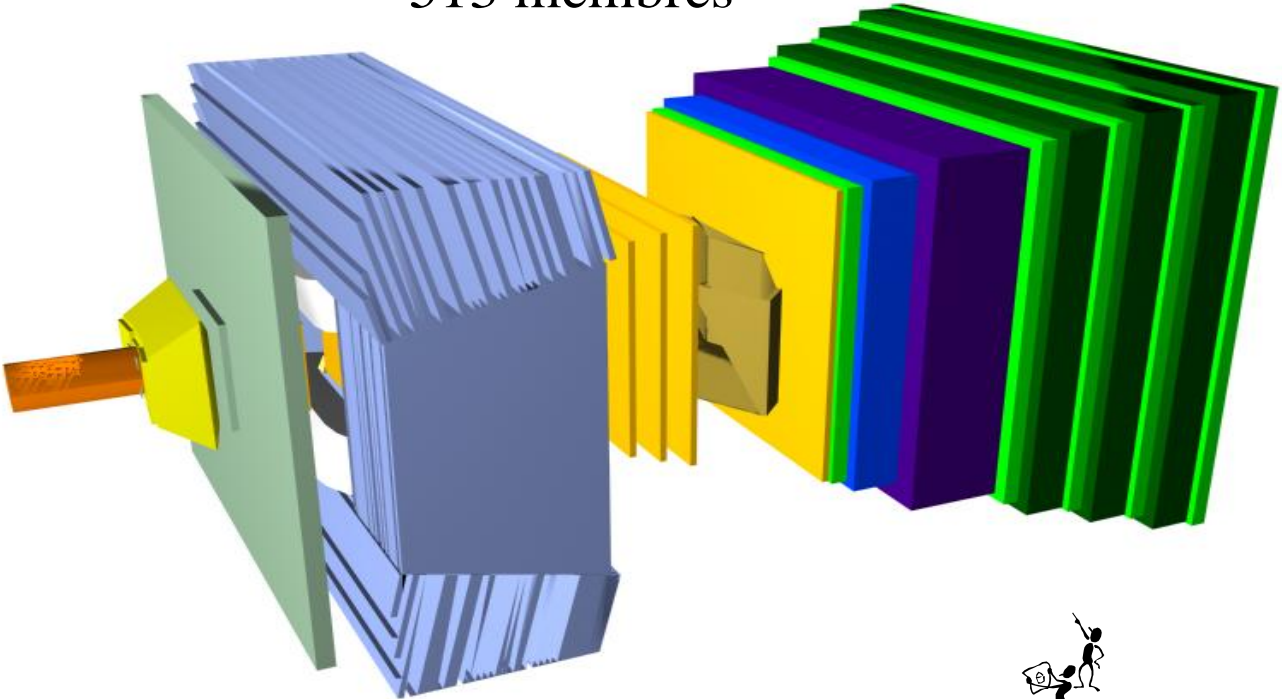
Number of channels	1.1 M
Event size	125 KB
Total RAW data per year	250 TB
Total ESD data per year	400 TB
Simulation data per year	360 TB

## *CPU*

Level2/3 CPU	35 KSI95
Reconstruction CPU	100 KSI95
Analysis CPU	20 KSI95
Simulation CPU	750 KSI95

# Collaboration *LHCb*

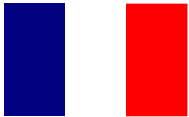
49 instituts  
513 membres



Brazil



Finland



France



Germany



Italy



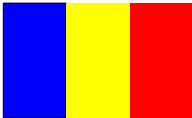
Netherlands



PRC



Poland



Romania



Russia



Spain



Ukraine



UK



Switzerland

# Défi

---



- Cahier de charge est plus exigeant:
  - ➔ Grand nombre d'éléments à contrôler – capacité d'extension
  - ➔ Utilisation du logiciel dans le trigger – qualité
  - ➔ Beaucoup plus de données et de CPU – performance
- Manque de main d'oeuvre expérimentée:
  - ➔ la technologie évolue très vite (hardware et software)

## Processus unifié:

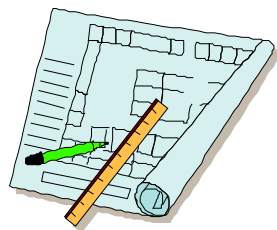
- Investir plus d'efforts dans la création des composants de haute qualité
- Réutiliser ces composants dans des contextes différents
- Pour bien choisir les composants, il faut avoir une vision globale du projet:
  - ➔ il y a une tendance de se séparer en domaines indépendants
  - ➔ il faut surmonter des préjugés personnels



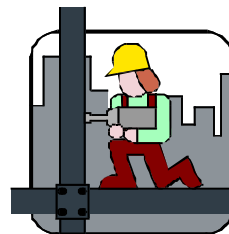
# Cathédral et Bazar



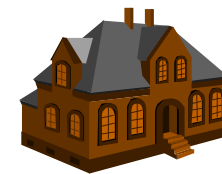
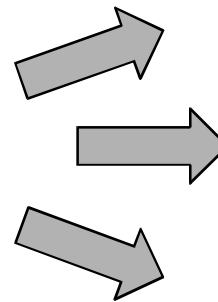
- Poussé par les besoins des utilisateurs ("use cases")
- Basé sur l'architecture:
  - ➔ Langage commun, meilleure compréhension du système à construire;
  - ➔ Liason faible entre les développements parallèles, intégration plus facile;
  - ➔ Plus de tolérance par rapport aux changements de spécifications;
  - ➔ Impose la réutilisation du code.



architecture



framework



applications

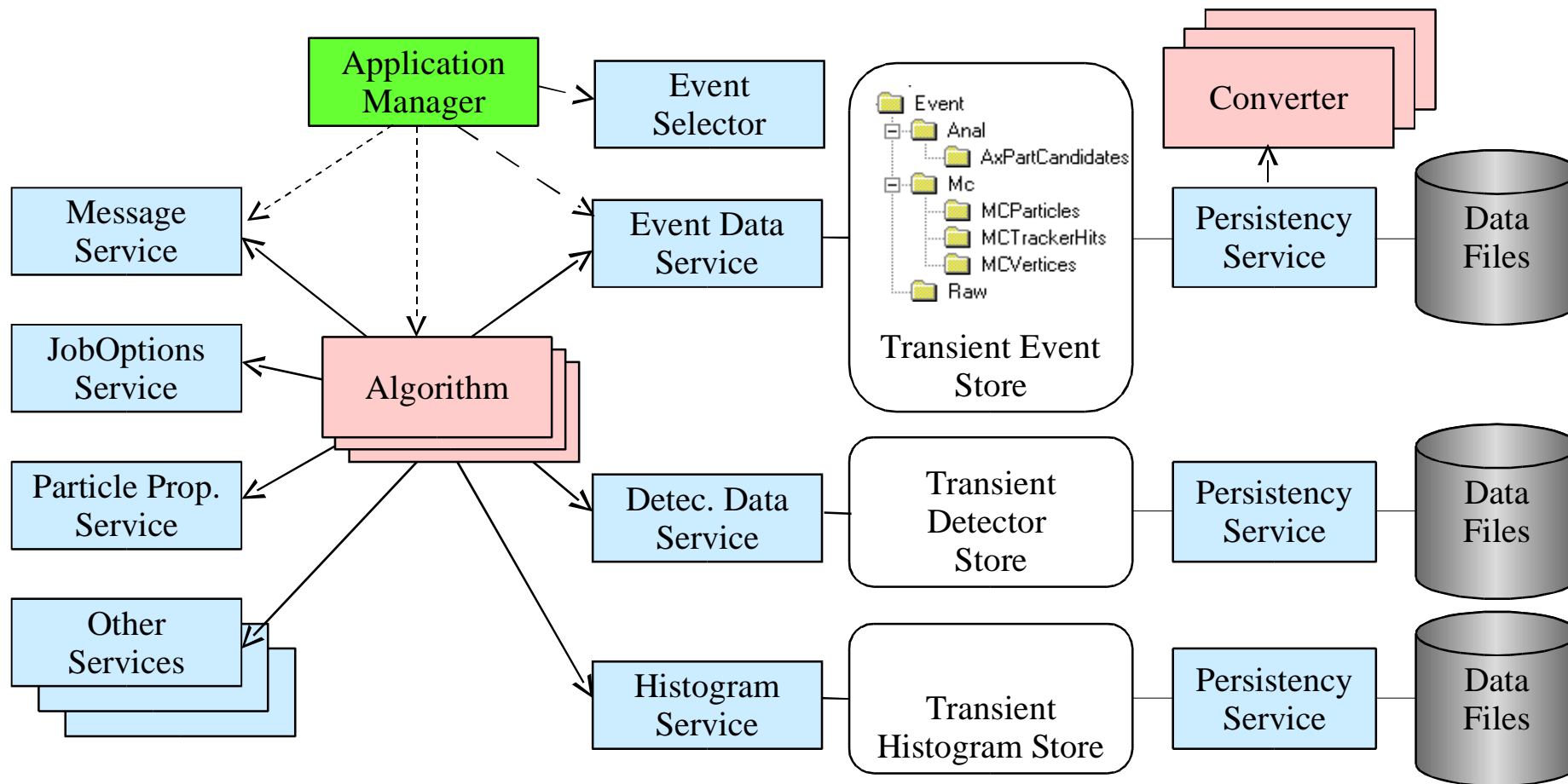
# Le but du projet GAUDI

---



- Développer un Framework pour utiliser à tous les étapes du traitement de données:
  - trigger Level2 et Level3, simulation, reconstruction, analyse, etc
- Eviter la duplication d'efforts:
  - la seule équipe des développeurs dans les domaine "on-line" et "off-line"
  - le seul framework pour les utilisateurs (physicien) pour développement des application
- Utilisation transparente des composants logiciels de tiers où c'est possible:
  - GUI, persistance, simulation

# GAUDI framework object diagram



# Choix conceptuels

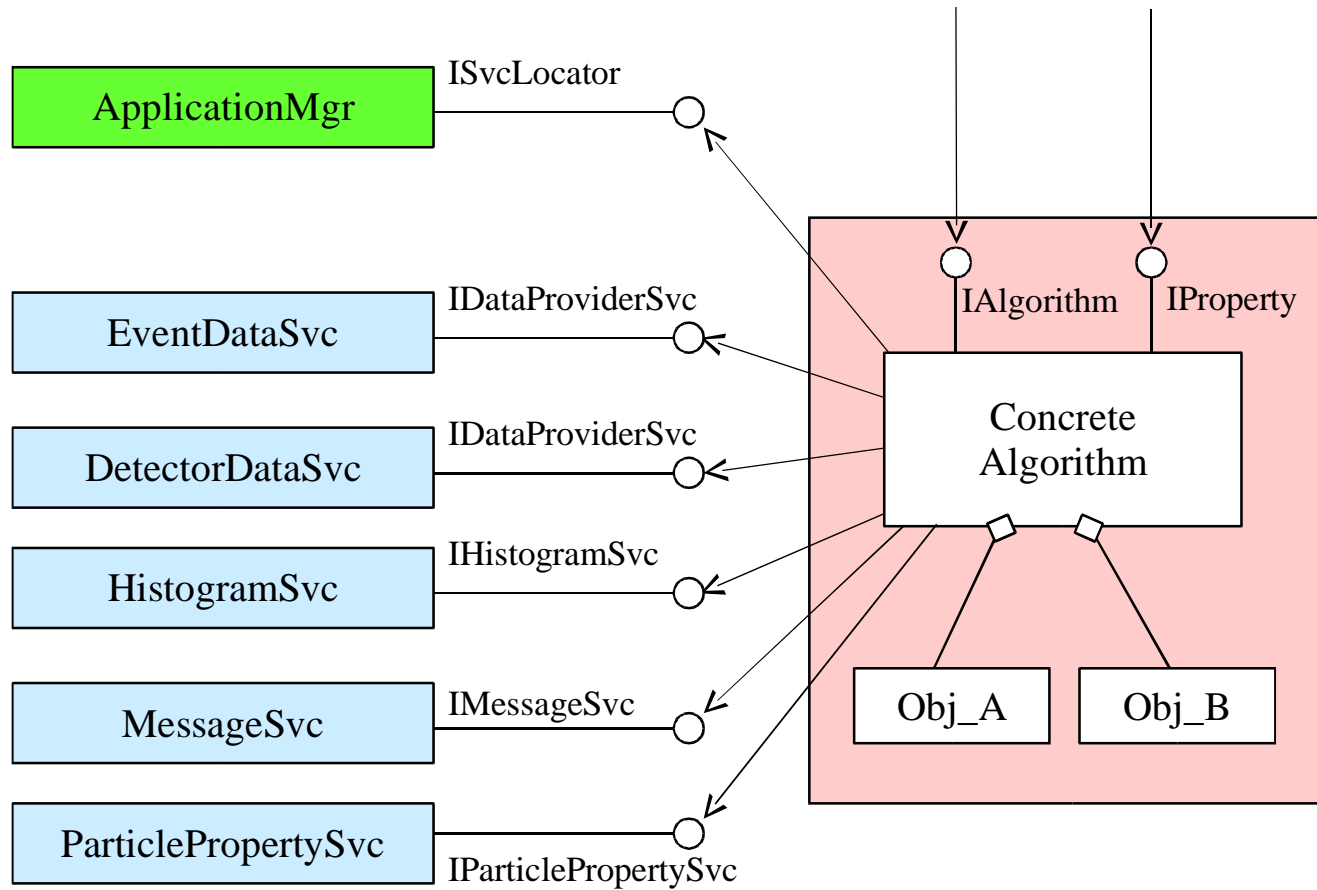
---



- Séparation claire entre les **données** et les **algorithmes**
- Trois types de données de base: **event, detector, statistics**
- Séparation claire entre les données **persistantes** et données "**transientes**"
- Le code d'utilisateur est localisé dans quelques endroits spécifiques: **algorithmes** et **convertisseurs**
- Tout les composants possèdent des **interfaces** bien définis et le plus **générique** possible



# Interfaces



# Interfaces en pratique...

---



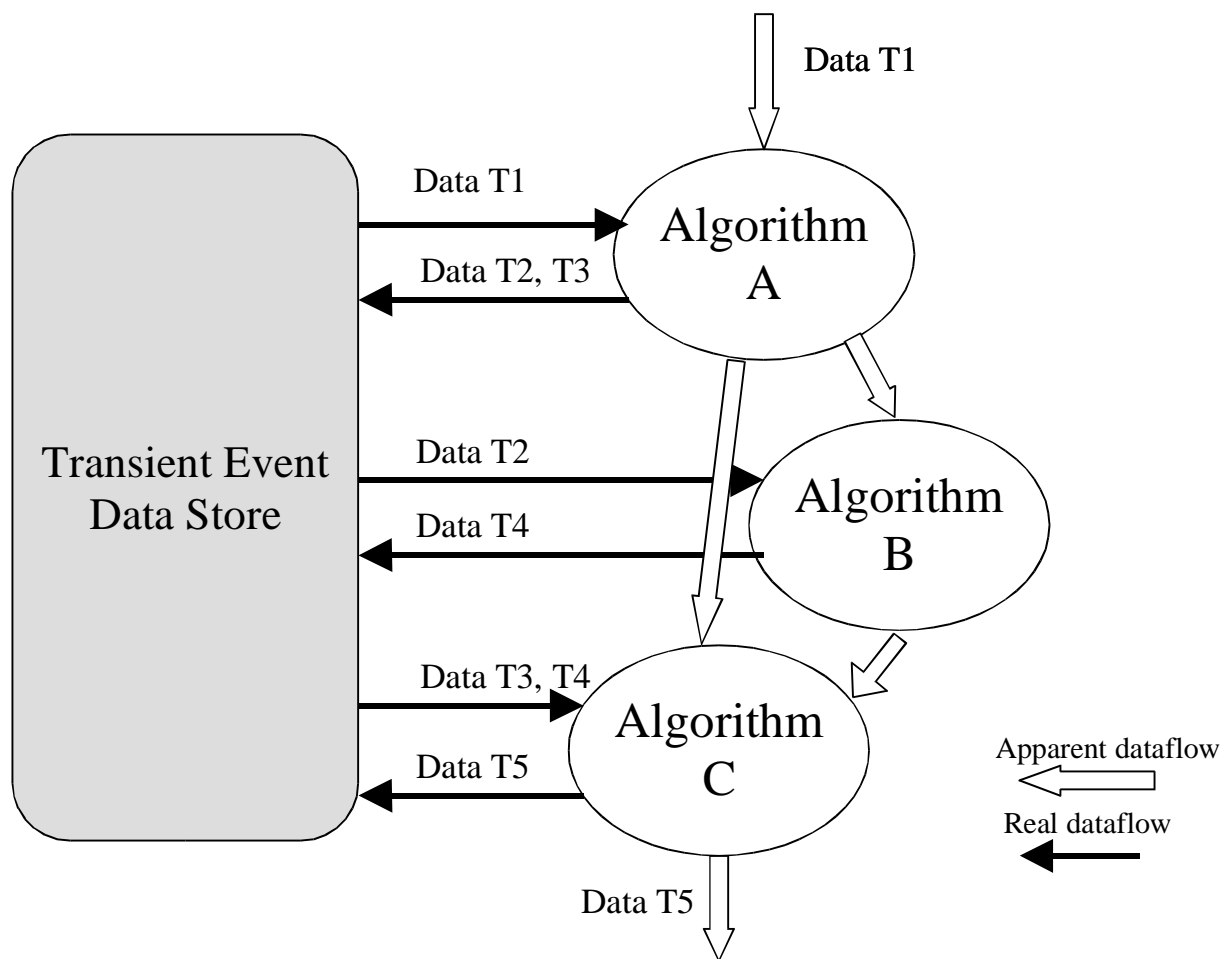
## IMyInterface.h

```
class IMyInterface {  
    void doSomething( int a, double b ) = 0;  
}
```

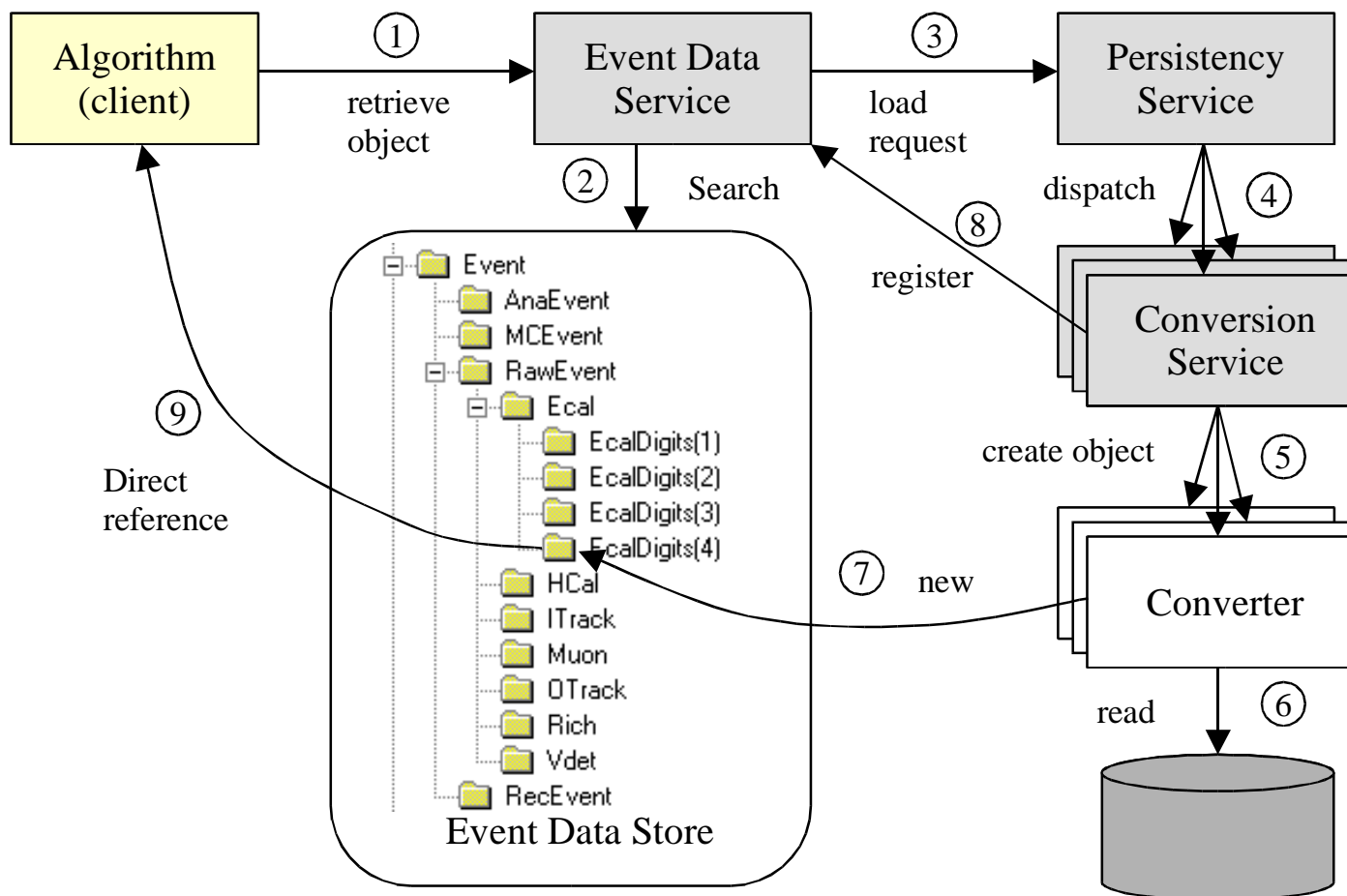
## ClientAlgorihtm.cpp

```
#include "IMyInterface.h"  
  
ClientAlgotihm::myMethod() {  
    // Declare the interface  
    IMyInterface* myinterface;  
    // Get the interface from somewhere  
    service("MyServiceProvider", myinterface );  
    // Use the interface  
    myinterface->doSomething( 10, 100.5 );  
}
```

# Déroulement des algorithmes

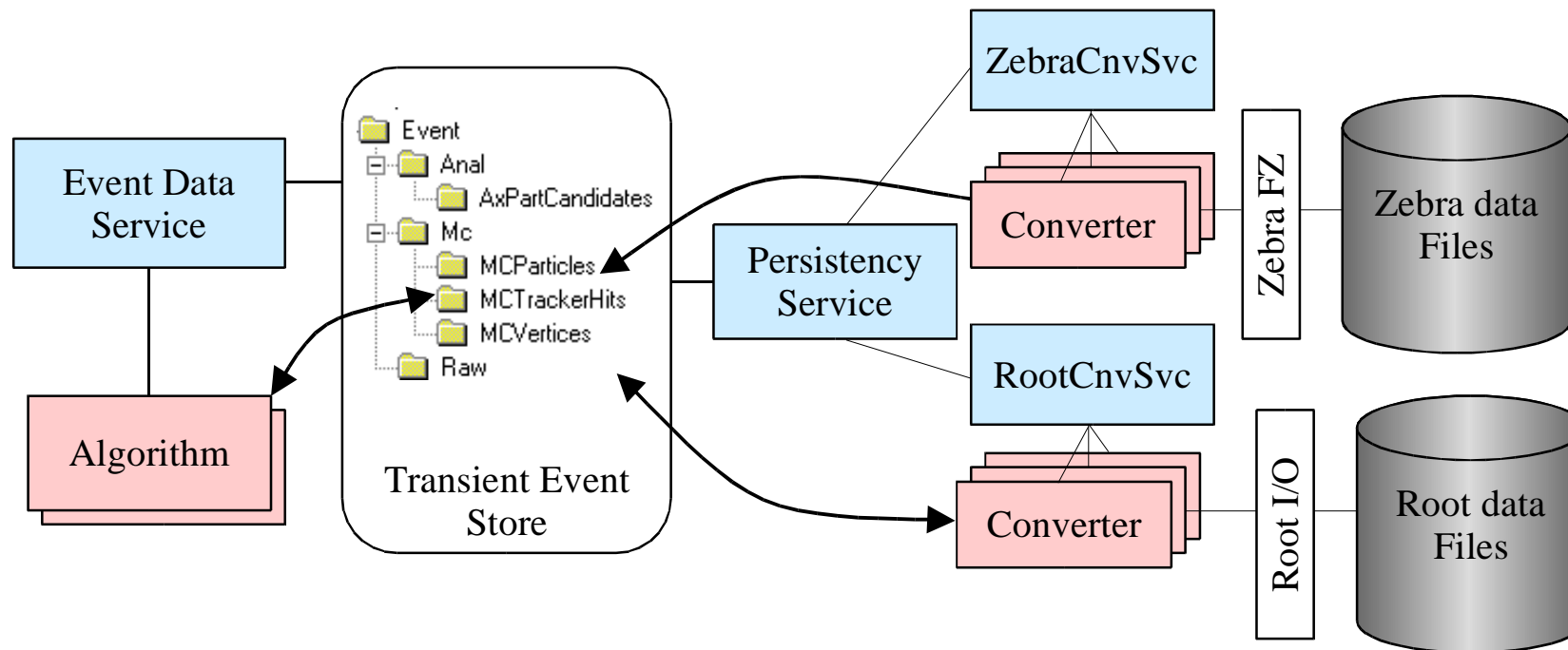


# Chargement de données sur demande



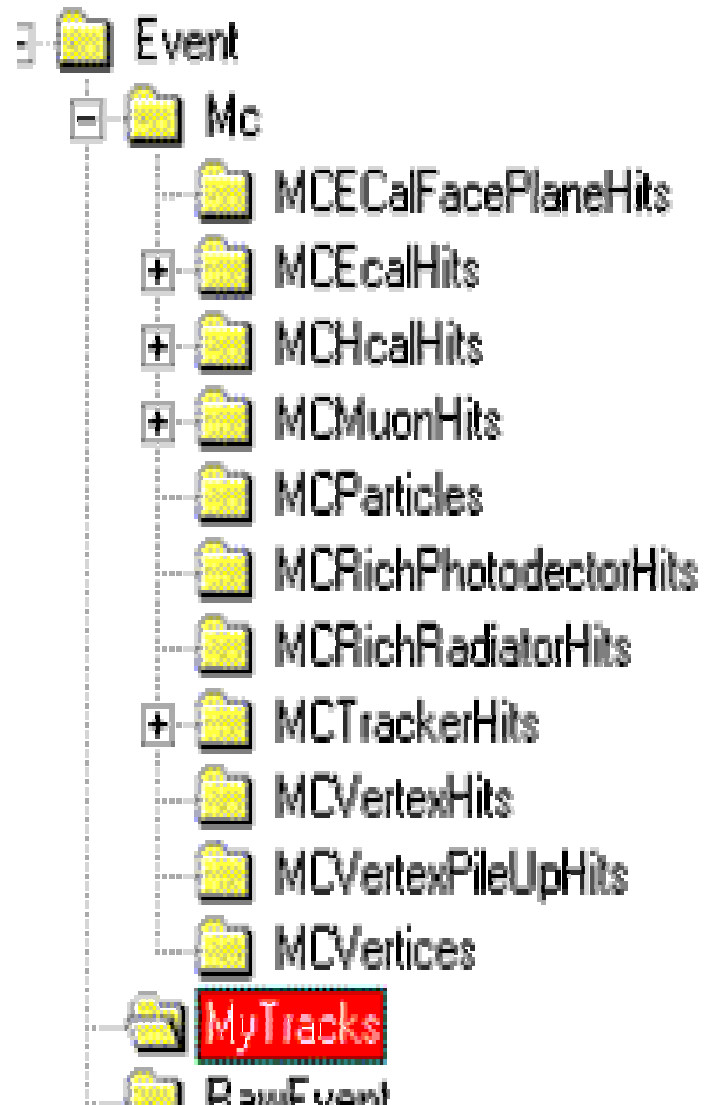


# Persistency



- Plusieurs technologies sont disponibles: Objectivity, Root, Zebra, ...
- Convertisseurs transforment des objets d'une représentation à une autre

# Persistency



## Map of data store items to ROOT tree names

#Event

#Event#MC

#Event#MC#MCECalFacePlaneHits

#Event#MC#MCECalHits

.....

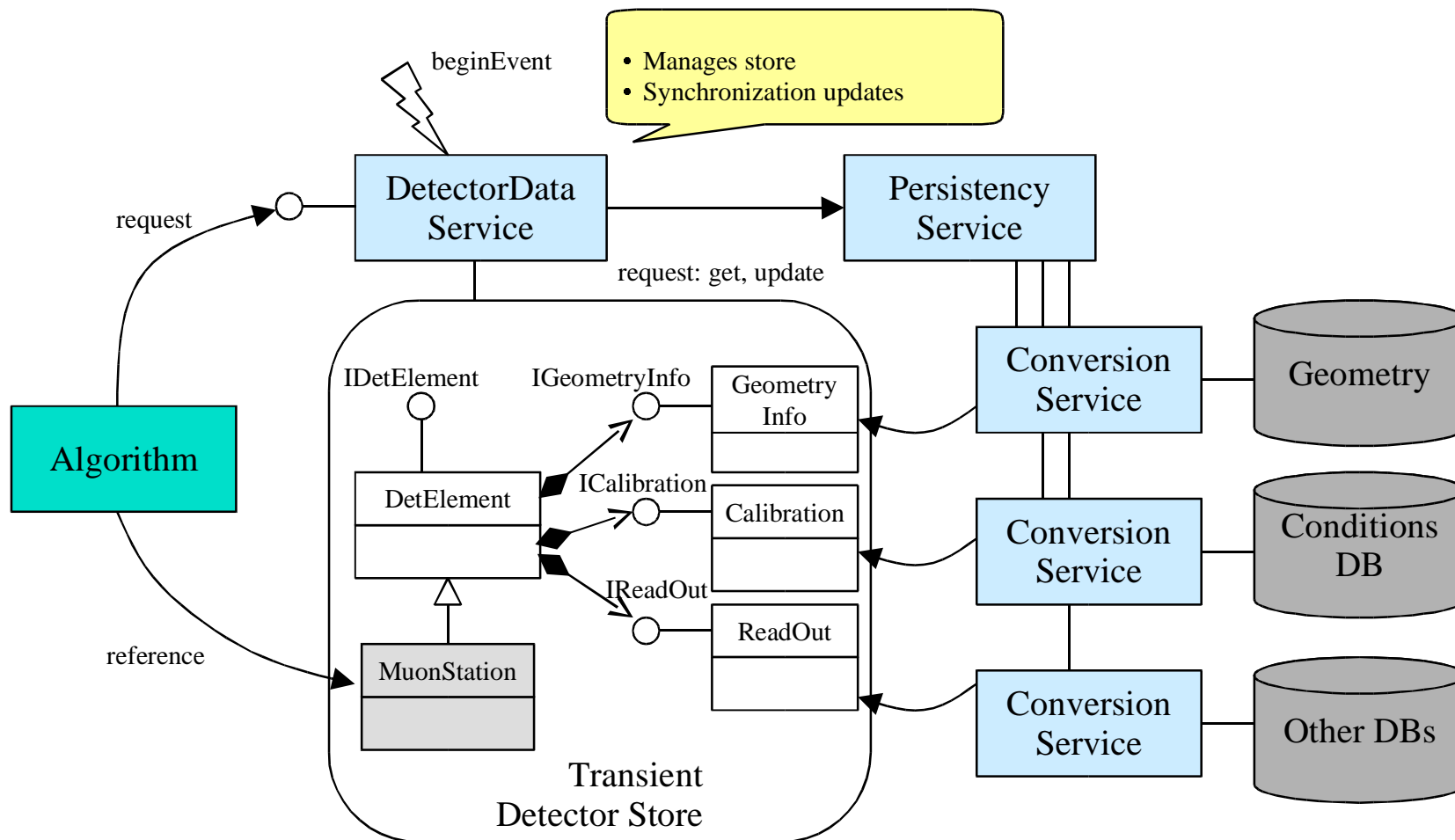
# Autres frameworks

---



- Detector Description
- Detector Conditions
- Statistics (Histogram and Ntuple) service
- Geant4 Services (GiGa)
- GUI, Scripting service
- Visualization service
- Data Dictionaries Services
- ...

# DetElement – accès aux données



# Interface avec GEANT4

---



- Intégration de GAUDI avec GEANT4 est assurée par des "Services GAUDI" dédiés (package GiGa)
- **GiGaGeomCnvSvc** fait la conversion des objets dans le *Transient Store* en objets géométriques de GEANT4:
  - n'utilise pas de code "utilisateur";
  - permet la flexibilité de la correspondance du modèle GAUDI au modèle du GEANT4
- Une source unique des paramètres géométriques
  - hiérarchie des fichiers XML

# Environnement de développement

---



- Développement sur Linux et NT
- Utilisation des bibliothèques logicielles:
  - STL, CLHEP, AIDA, NAG\_C, XML, ...
- Gestion de configuration logicielle (packetaage physique)
  - minimisation des dépendances entre les packages
  - organisation des librairies dynamiques
  - distribution et installation du logiciel
  - utilise CMT avec beaucoup de satisfaction
- Documentation
  - DOXYGEN pour la documentation du code
  - Manuel d'utilisateur détaillé
  - Beaucoup d'échanges sur la liste de discussion

# Collaboration

---



- GAUDI devient un projet de type Open–Source:
  - utilise l'espace Web et l'espace de release independant
- ATLAS maintenant fait des contributions importantes:
  - e.g. interface scripting
- Autres expérience utilise GAUDI:
  - HARP, GLAST, OPERA
- Nous comptons sur les contributions d'autres expériences, projets:
  - e.g. scripting, dictionnaires de données, analyse interactive, ...
- Meilleur vérification des composant dans les environnements differents
- La maintenance de long terme partagée

# Status

---



- Septembre 1998 – démarrage du projet;
- Novembre 1998 – évaluation de l'architecture par les experts HEP;
- February 1999 – première release de GAUDI;
- Releases incrémentales ( ~3/ans);
  - ➔ v8 maintenant
- Les applicatifs de LHCb sont en cours de migration dans le framework de GAUDI:
  - ➔ reconstruction (BRUNEL) est en production depuis Juin 2001;
    - ◆ grande partie – "wrapped FORTRAN";
    - ◆ parties FORTRAN sont progressivement remplacées par C++;
  - ➔ autres applicatifs suivront (simulation GAUSS, analyse DaVinci).



# LHCb au CC/IN2P3

---



## Maintenant:

- CC héberge l'analyse pour les groupes français
  - ➔ Tout l'environnement logiciel LHCb est maintenu;
- Production MC très importante:
  - ➔ 500 Kh consommées en 2001;
  - ➔ 4TO de données dans HPSS, dont ~2TO exportées au CERN
- Tests dans le cadre du projet DataGRID sont sur le point de démarrer

## "LHC":

- centre "Tier1" pour LHCb:
  - ➔ demande équivalente aux autres expériences LHC

# Conclusions

---



- LHCb a pris l'approche unifiée pour le développement de toutes les applications de traitement de données
- Importance d'une architecture unique bien définie avec des composants faiblement couplés
- Utilisation de composants logiciels de tiers est facilitée
- Collaboration active avec d'autres expérience a démarrée