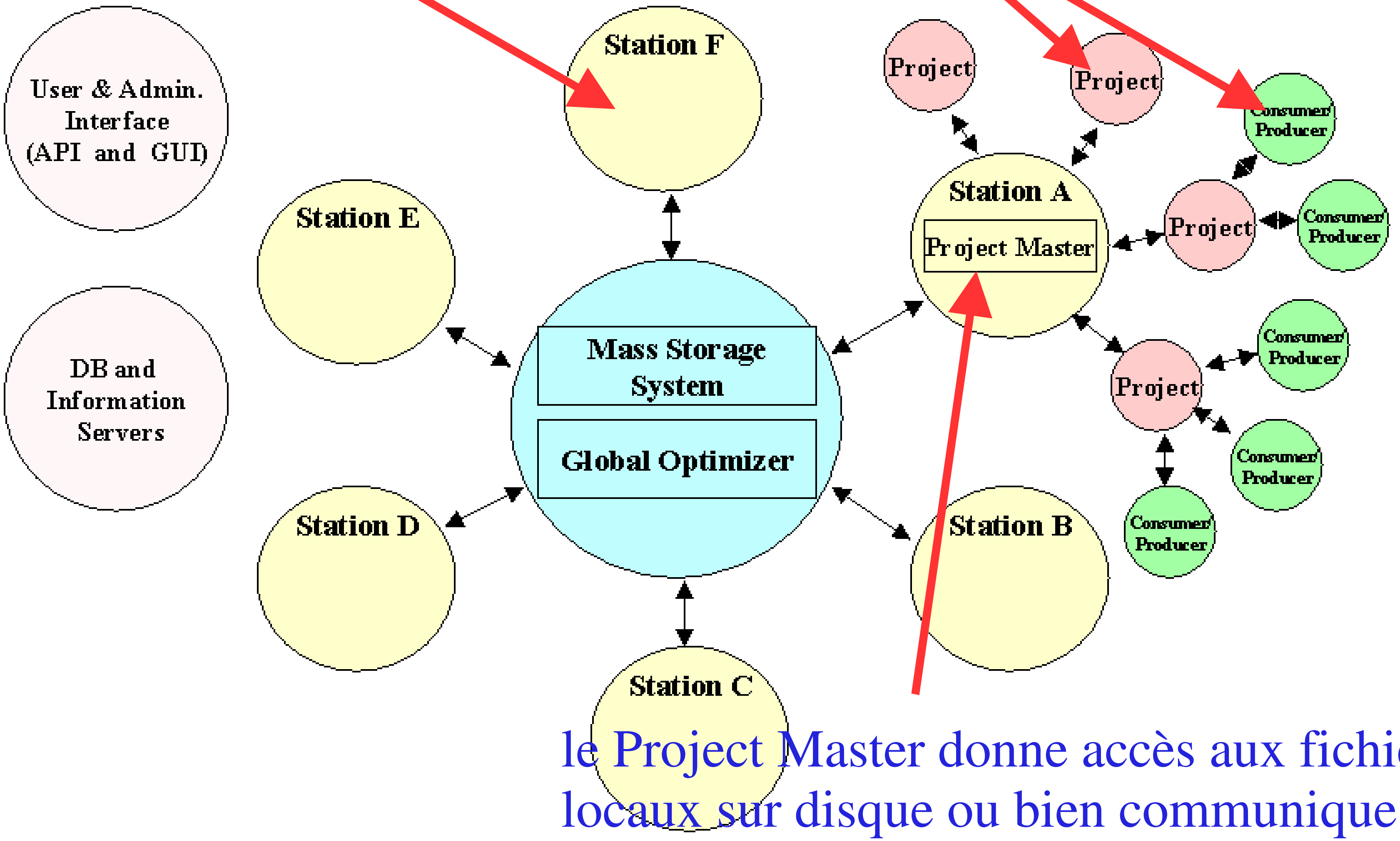
SAM au CCIN2P3

Laurent Duflot

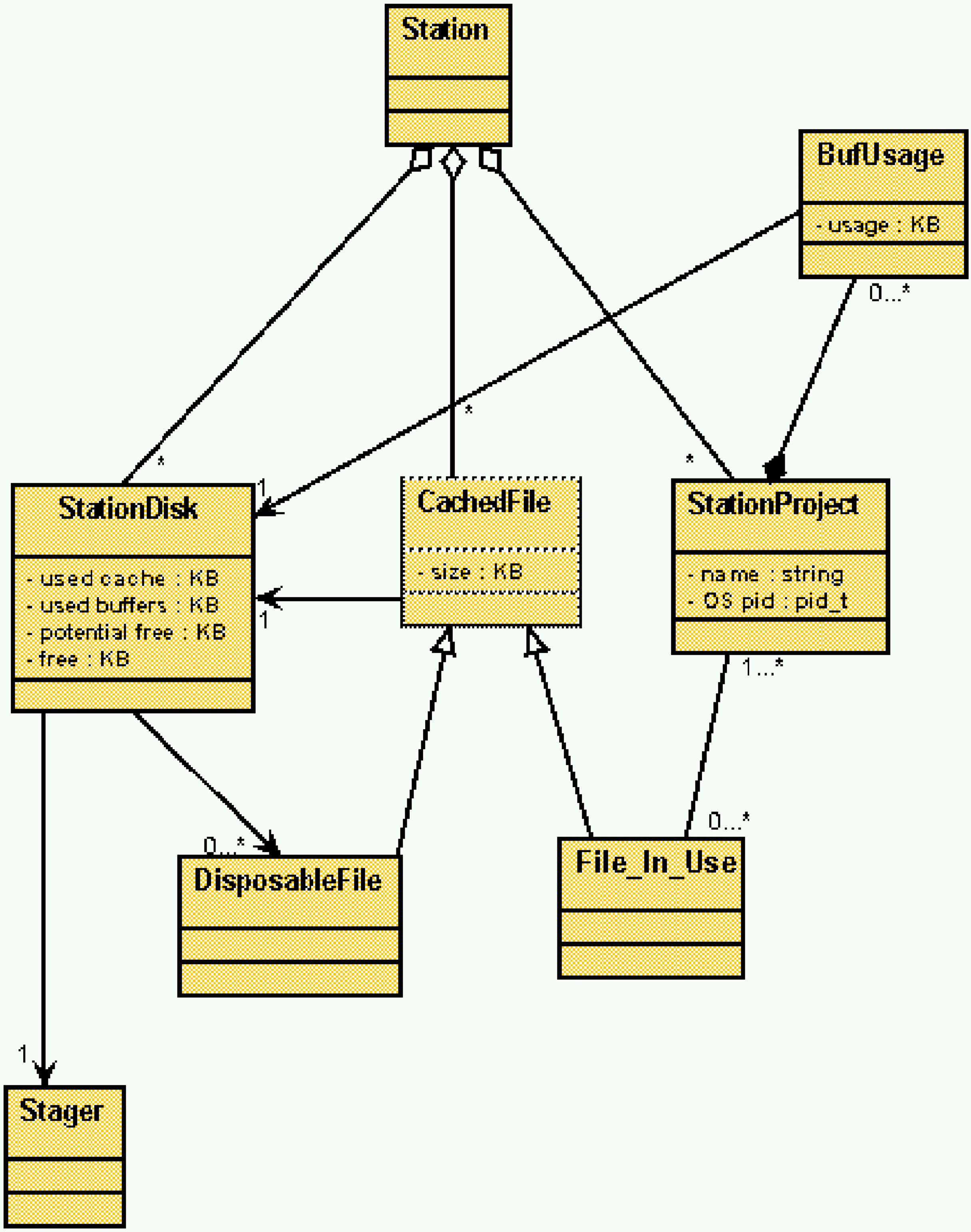
- Description
- Les problèmes rencontrés
- État des lieux
- Modifications nécessaires pour le CC

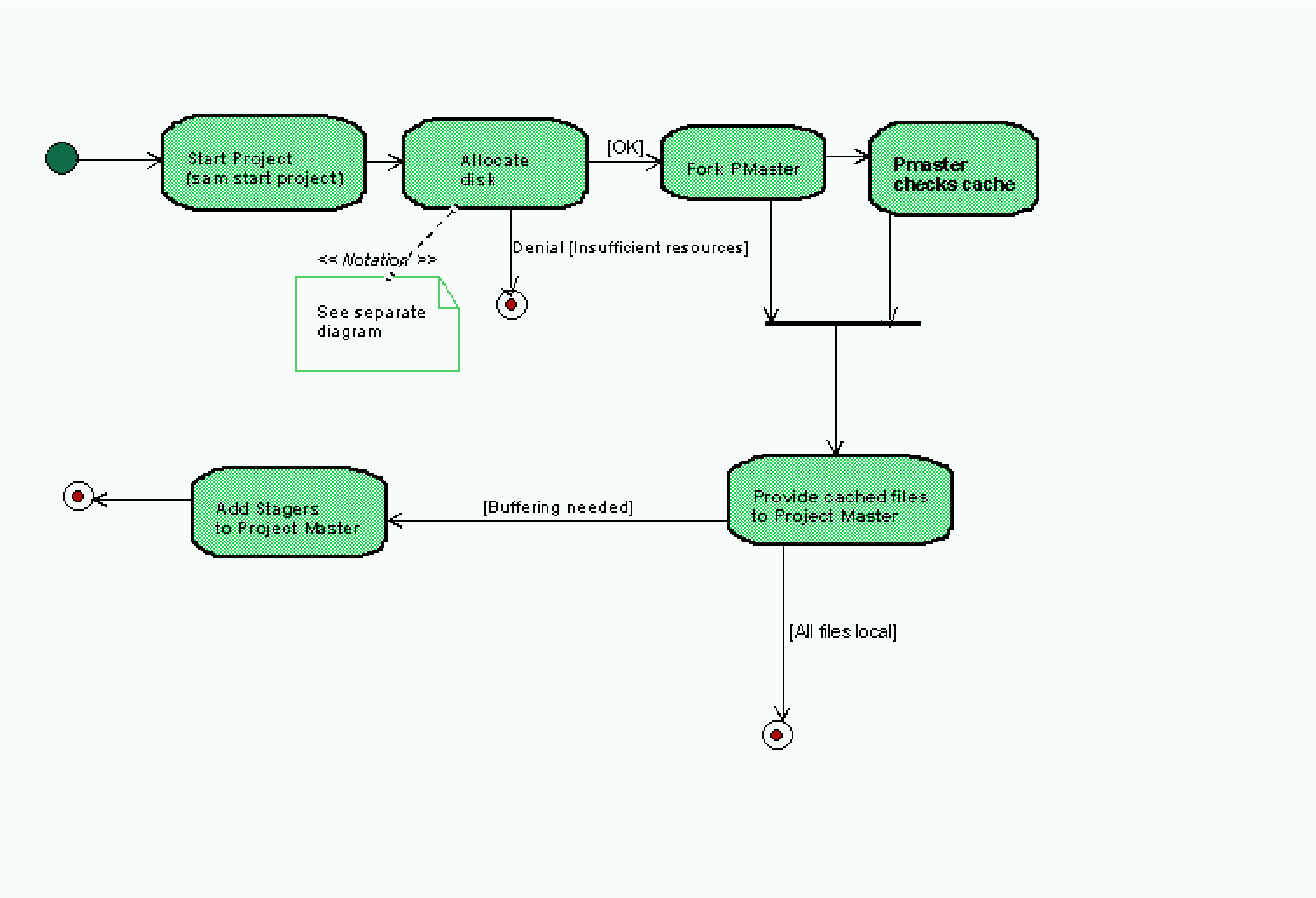
une ou plusieures stations
par machine

le job communique à la station
à travers un projet



le Project Master donne accès aux fichiers
locaux sur disque ou bien communique
avec le robot à travers un eworker.


Station
BufUsage
- usage : KB
0...*
*
*
*
1
StationDisk
- used cache : KB
- used buffers : KB
- potential free : KB
- free : KB
1
CachedFile
- size : KB
StationProject
- name : string
- OS pid : pid_t
1...*
0...*
0...*
DisposableFile
File_In_Use
1
Stager


Start Project
(sam start project)
Allocate
disk
[OK]
Fork PMaster
Pmaster
checks cache
Denial [Insufficient resources]
<< Notation >>
See separate
diagram
Provide cached files
to Project Master
[Buffering needed]
Add Stagers
to Project Master
[All files local]

SAM au CC: problèmes rencontrés

- .fnal.gov manquant dans la database...
- Bugs dans Fnorb (interface Python – CORBA) pour les communications inter–domaines, i.e. in2p3.fr $\leftrightarrow$ fnal.gov
- Le transfert de fichier inter–machine utilise rcp: on a créé un script rcp faisant appel à scp et engendré des clefs d'autorisation: ca marche, mais on encrypte les données, pas le plus efficace pour un transfert...
- Passage à sam v2.0 (sur machines RH61) pénible mais sam v2.1 ok.

État des lieux

- Station ccin2p3–analysis définie dans la DB de développement.
- Récupération de fichier depuis enstore (FNAL) pas encore en service (à mettre en oeuvre rapidement), station démarrée avec les disques de d0mino en "preferred locations"
- Lecture d'un fichier à FNAL avec mc_exam tournant au CC OK !
- Lecture d'un fichier dans HPSS (HPSS –> cache par rfcp)
- Procédure d'export CC –> FNAL avec sam store XXX testée en août OK ! Passe par un disque buffer sur d0mino. Devrait être bon dans la version courante, à re–tester avant la prod pmc03

Quel SAM au CCIN2P3 ?

- Option 1: utilisation intégrale de HPSS
  - pas de cache, pas de station
  - le Project Master retourne directement le nom de fichier dans l'espace HPSS et le code D0 y accède avec rfio
  - on perd toutes les optimisations de SAM qui n'a aucun moyen de savoir
    - quels fichiers sont réellement sur disque
    - sur quelles cassettes sont les fichiers: pas de 'stage ahead' possible
  - Les outils de tranfert doivent avoir accès à l'espace HPSS (scp, bbftp,...) ou bien il faut mettre en place des procédures compliquées de disques buffer

- Option 2: accès directement aux bandes
  - besoin d'un cache disque important pour avoir de bonnes performances
  - besoin d'une station
  - on doit modifier eworker pour l'accès au robot du CC
    - nouveau design pour l'eworker, codage va démarrer (bbftp, xtage?)
  - utilisation des optimisations de SAM (qui connaît toutes les demandes de tous les jobs D0)
  - besoin d'un outil
    - nom de fichier <–> VID fseq size
    - gestion automatique d'un pool de bandes pour l'écriture
- Option 3: idem mais HPSS comme stager
  - pas besoin d'outil nouveau mais pas d'optimisation SAM

DSPACK / evpack et RFIO

- RFIO redéfinit des fonctions I/O du C (fortran)
- DSPACK (Evelyne)
  - écrit en C
  - inclusion du header shift.h aux endroits stratégiques
  - quelques petites modifs (plugins)
  - accès zone xtaging et HPSS marche
- evpack
  - écrit en C++ I/O par les iostream
  - pas de solutions pour RFIO actuellement
  - déjà tenté par EROS pour g++ sans réel succès (pas de seek/tell dans RFIO ?). A revoir, mais demandera pas mal de temps pour développer une solution.

eworker

- Une nouvelle classe abstraite FileMover
  - décide si peut s'occuper du transfert selon la source et la destination (disque, enstore, robot, HPSS, ...)
  - construit les commandes nécessaires, qui seront exécutées dans une autre classe
  - vérifie si le tranfert s'est bien passé et retourne le VID de la bande en sortie
- Des classes concrètes FileMoverRcp FileMoverBbftp FileMoverEncp FileMoverXtage ...
- Francois Touze et Guy le Meur

Proposition

- A court terme (–> fin 2000), comme les accès RFIO ne marchent pas, choix de l'option 3
  - fichiers dans HPSS
  - cache SAM accessible par NFS (AFS ???)
  - machine de service Linux pour SAM
  - mettre les données de pmc02 dans HPSS (espace disponible?), tests de stress de HPSS pour valider le choix (pertes des optimisations de SAM)
  - scp/rfio, bbftp/rfio
- Développer eworker
- Investiguer iostream_rfio
- Si HPSS pas satisfaisant, développer les outils xtage