Policies

- Full tape back up at T1 sites? Especially:
- AOD (RECO in ’08?) T1 replication - to keep non custodial data on tape or not?
  - If yes, need a lot of extra tapes and recycling effort when data sets are not in use anymore.
  - If now, may need support for storage space tokens (see next)
  - Leave it sites? Never mind operational side? (data recovery - from another T1 or from tape: what’s easier?)
- Sim data – computing model call for 10% for the disk cache.
  - Too little fraction? Or it will be never accessed by jobs at T1s anyway? MC skimming jobs? Prestaging?
  - MC is not so usefull anyway, will produce much less in ’08?

For megatable discussion

- Input for LCG megatable
  - Provide only total tape and total disk for a site.
  - Not fill up “Storage for T2” parts in T1’s folders
    - Doesn’t make sence!
  - Drop completely breakdown by storage classes
  - Omit buffer space too
    - Leave for site’s discretion
- Should avoid concrete planning going to sites via the LCG channel. This is confusing site admins (“but LCG told us” is very commonly heard from site admins) and creates problems for CMS people. Forth sites talk to local CMS contacts

CMS and storage classes

artem

What is a storage class

- Storage use cases reduced down to three basic ones:
  - Tape1disk0
    - Data on tape, disk cache is system managed
      - Default for tape -based systems, like at most T1s
  - Tape0Disk1
    - Data on disk only, user managed
      - Default for disk-only systems, like at most T2s
  - Tape1Disk1
    - Data on tape and pinned on disk
      - Controversial class!

CMS and storage classes

- IMHO, should stay with proven defaults:
  - Tape1disk0 guaranties that if there is more data than disk, it will be accommodated. If the disk space provide by the site roughly matches to the volume of data (i.e. the disk is what was pledged, the volume is what was anticipated by the C-TDR), there is no problem at all.
  - Too much troubles with disk1tape1, since if there is more data than disk, then requests for new transfers will be aborted.

On top of storage classes - SpaceTokens

- Storage classes are not directly used
- Instead, SpaceTokens are used. They are sort of “symlinks” to classes, but also allow to select a distinct pool in the storage.
  - The only way to select a castor service class via SRM
  - In dCache the same thing could be done by mapping directories to pools. Now dCache could also differentiate pools by SpaceTokens.
- A user passes SpaceToken with transfer requests, i.e. an extra command line option.
- SpaceTokens names are subject to intimate agreement between a site and a CMS representatives to this site. CMS doesn’t need them published in BDII.

Need to use SpaceTokens

- In principle, no need, since there is always a default pool, where files are transferred to when no token is given.
- Use of space tokens is mandatory
  - For Castor, whenever more than one service class is in use
    - Example: one wants to put raw and reco data in different set of pools (and tapes!).
  - For dCache, whenever there are more than 1 set of pools, and mapping directories to them is not sufficient
    - Example: AOD replication – custodial AOD goes to tape-connected pool, while replica AOD from another site goes to disk-only pool.

CMS and space tokens

- CMS should support use of space tokens in Phedex, but be flexible and not require use of them at all T1 sites
  - Phedex architecture allows this
- To simplify our life, CMS should approve file name space with earlier separation by data types, so that dCache sites wont have to mess up with space tokens too much
  - /store/FEVT, /store/AOD …

Space reservation

- Again, getting too complex, save your time and nerves, don’t even talk about it!
- In short, disk space, associated with a space tokens is considered “reserved” by the VO.
- CMS should never fill up a disk, since we are not going to abuse pinning, and not want to use Tape1Disk1 class.