

# A tier2's perspective

Giacinto Donvito – INFN-Bari



- Federation(s):
  - We indeed have already two different federations
- CMS use case
- How to join the XRootD federation
- How a CMS site could benefit from an XRootD federation
- How a CMS users could benefit from a XRootD federation
- INFN-Bari is also hosting an XRootD redirector
- Feedbacks ...
- ... what we feel is still missing

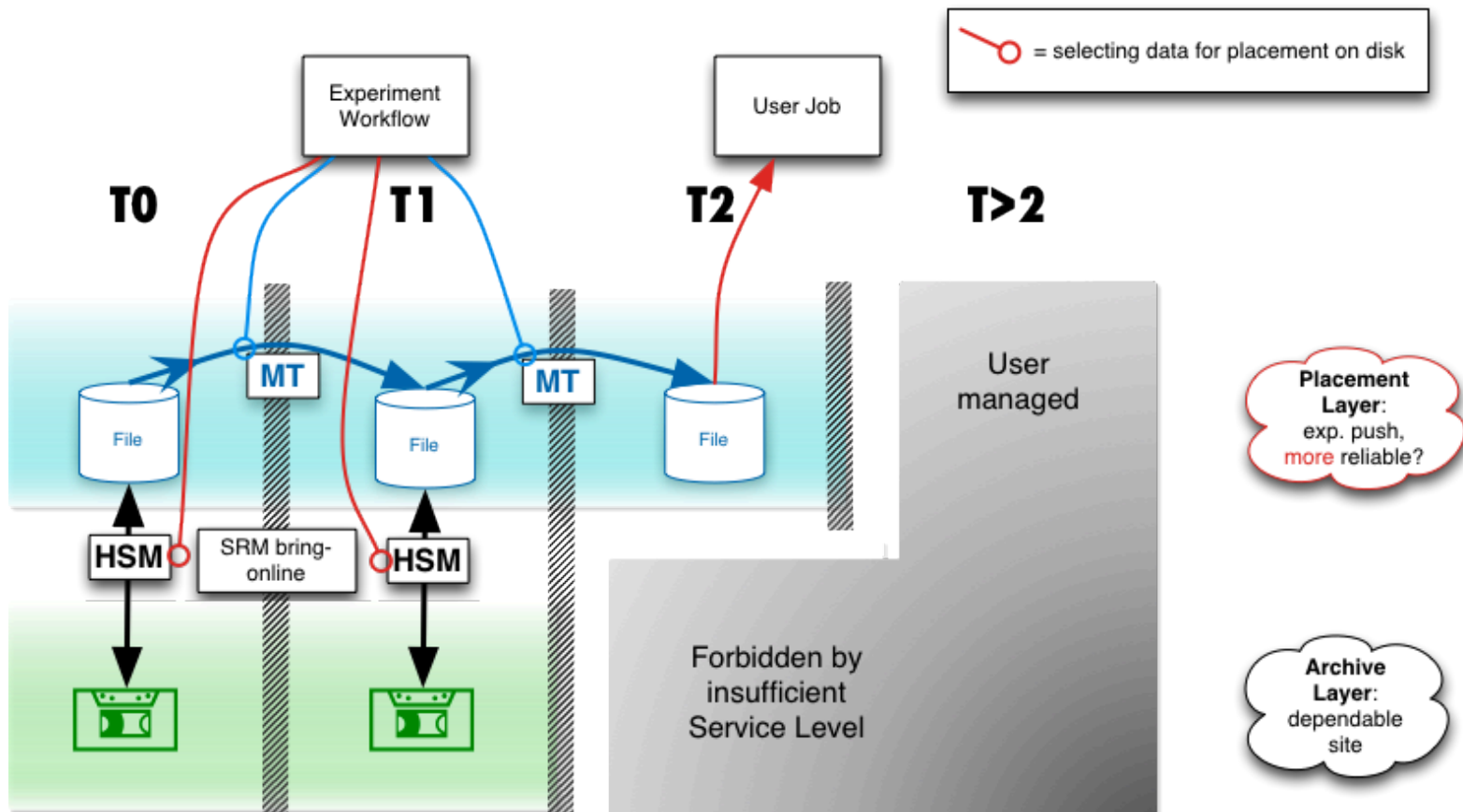
# Federation(s)

- All the CMS sites (Tier0, Tier1, Tier2) are already part of a federation
  - All the CMS files are registered under a common Logical NameSpace.
  - There is an automatic system that is able to deal with data transfers among sites
    - The data transfers need to be “requested” from a “user” and “approved” from an “admin”
  - There is a catalogue that knows where a “block of files” is in order to use it
- I will **NOT** talk about “this federation” today!

# CMS Use Case

- Add a more dynamic and transparent data access layer for “not massive” data reading
  - Writing and debugging code against file not available on your machine/site
  - Automatic fall-back within the CMS framework in case of storage failures
  - Running analysis on small sites with lack of manpower
- AAA has been already described
  - I will focus on Site Admin view ...
  - ... and effort

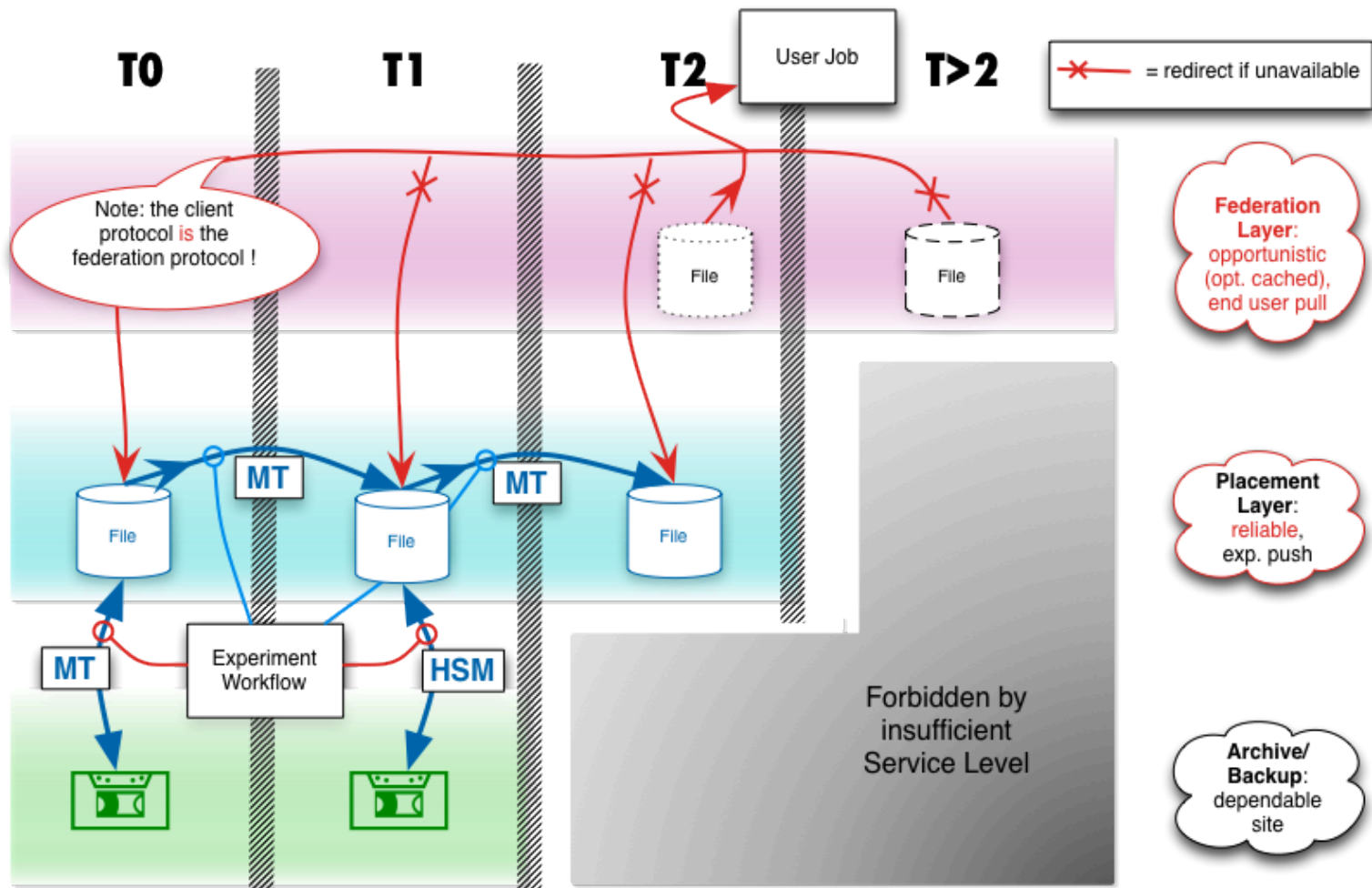
# SE in the Classical Federation



# How a T2 could join the XRootD federation

- The site admin should expose the files using one or more xrootd door
- The practical step needed depends on the storage systems used by the site:
  - Posix native file-system is generally far easier than all the other solutions
  - There are already working solutions for all the storage elements used in the HEP community
  - The capability of xrootd to exploit plugins and libraries is very important here for exploiting new storage systems.

# SE in the Xrootd (AAA) federation



# HW Requirements for a CMS Tier2

- Add at least one more machine:
  - The same hw requirement of a “gridftp door”
  - A good suggestion could be having two-three different server for fail-over and load balancing purpose
- The network connection of the machine could be an easy way of throttling the amount of byte flowing out of your site
  - At INFN\_Bari we have 2 machine 2Gbit/s each
  - This helps us to keep under control the bandwidth utilization
- No need for an HighAvailability solution among xrootd doors:
  - The redirector use only machines that are working and responding



# SW Requirements for a CMS Tier2

- Posix file-system:
  - The xrootd door should mount the file system and install a the native Xrootd+CMSD server
  
- dCache:
  - Xrootd in proxy-mode
  - In the near future a standard CMSD daemon and dCache xrootd door
  
- DPM:
  - New version of DPM
  - DPM Xrootd door + CMSD native daemon
  
- HDFS:
  - Native Xrootd+CMSD servers
  - An ad-hoc plugin for reading data on HDFS

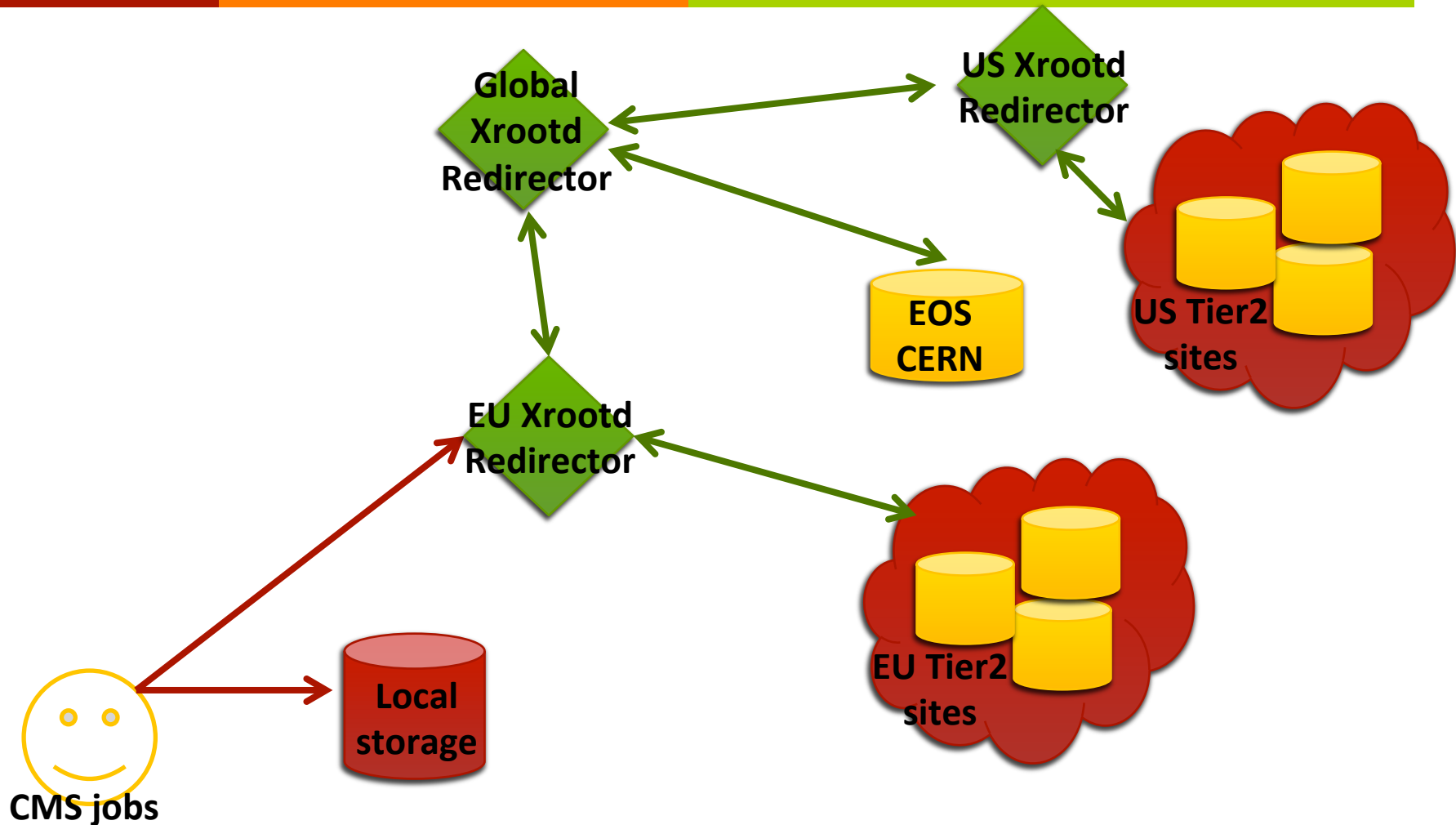
# SW Requirements for a CMS Tier2

- In all the cases the xrootd/cmsd daemons should be configured to join the redirector of the region where the site belongs to
  - At the moment there are only two:
    - US\_Nebraska (US Region)
    - IT\_Bari (EU Region)
  - Plus the global one at CERN
- It is useful also to configure the Xrootd to provide monitoring information to the central monitoring server
- There is a daily report that provide accounting usage record
  - Very useful to know how your site is working and how much data is providing to the federation

# How a CMS site could benefit from an XRootD federation

- CMSSW has the capabilities to fallback in case of failures while reading a file
  - It needs only to edit a configuration file in the CMS software directory
  - It is transparent to the end user and to the jobs
  - It could be very important when you have a disk server temporarily broken
    - Or any other storage problem that could make the jobs fail for input-file-error
- In future it could be possible/desirable to send jobs to a less busy Tier2 when too long queue time is registered to a specific Tier2 hosting very interesting data
  - And read data remotely

# How a CMS site could benefit from an XRootD federation



# How a CMS site could benefit from an XRootD federation

- Supporting local user activity at a TierX site could be quite time consuming for a sys-admin:
  - Users ask:
    - To access one or few files that are not available on the TierX storage for writing or debugging code, before submitting jobs to the grid.
    - To access a whole dataset for only one run
  - This typically could mean transferring few files manually or request to phedex to move all the dataset, and delete it after few days
    - This is time consuming for the site-admin and users
    - Also using phedex could cause waiting for the tails of failing transfers
- “WAN federated access” or “dynamic caching” could be of great help here

# How a CMS users could benefit from a XRootD federation

- The federation infrastructure could be used also with bare ROOT for the last step of the analysis
  - Files stored from grid job to a given Tier2 at the end of the analysis could be read by the users also from his/her laptop/desktop or Tier3
    - At Bari we are supporting a Tier3 exactly for this use case (Perugia)
    - This could be also useful for easy sharing files among different users and groups
- Tier3 could run small set of jobs reading files remotely from a “close” Tier2
  - If the tier3 do not have enough storage or man power to store data reliably
  - At Bari we are already providing such a feature to (Trieste) Tier3

# INFN-Bari is also hosting an XRootD redirector

- Installing a native Xrootd+CMSD server configured as redirector
- Need to configure a fall-back to ask files not in the federation to the Global redirector
- It is stateless:
  - No DB to back-up, only one configuration file.
  - This means that it is also easy to obtain a HighAvailable configuration
- The logs provides good information on what is happening among the client and the data servers

- Documentation could be improved at least for EGI/EU deployment
- The support mailing is quite active and is providing fast answers
- The initial feedback from the users is that it is working quite well and transparently
  - Also if the use base should increase consistently to gain more experience



# ... what we feel is still missing

- Alarms on daemon failures
- Alarms on errors:
  - At a redirector level one could catch errors and try to react on those:
    - Advise the site that is failing
    - Warn the users that is doing something wrong
- More throttling capabilities
  - Per users/sites/etc
  - Number of open streams
- The capabilities to choose data sources that could serve better the client based on the network topology
  - Work in progress

# Links and documentation

- <https://twiki.cern.ch/twiki/bin/view/Main/CmsXrootdArchitecture>
- <https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Xroot/Setup>
- **Monitoring:** <https://twiki.cern.ch/twiki/bin/view/Main/XrootdMonitoring>
- **Throttling:** <https://twiki.cern.ch/twiki/bin/view/Main/XrootdThrottling>
- [hn-cms-wanaccess@cern.ch](mailto:hn-cms-wanaccess@cern.ch)
- US redirector contact: Brian Bockelman, [bbockelm@cse.unl.edu](mailto:bbockelm@cse.unl.edu)
- EU redirector contact: Giacinto Donvito, [giacinto.donvito@ba.infn.it](mailto:giacinto.donvito@ba.infn.it)