LHCb and the protocol zoo

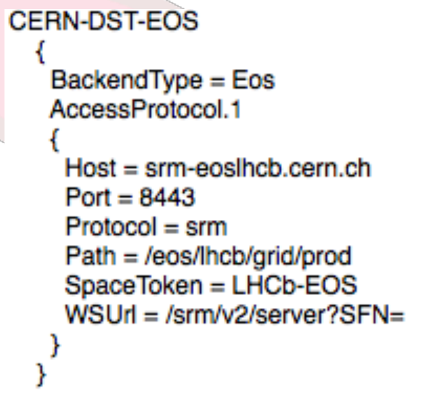
Ph. Charpentier / CERN
For the LHCb Data Management

Reminder: use cases

- File transfer
  - Dataset replications, centrally managed (from SE1 to SE2)
  - Single file transfer: e.g. file upload/download from/to job or desktop
  - (Pre-)Staging: ransfer from tape to disk (production only)
    - Requires pinning
- File access
  - Use case: open a file using Gaudi, i.e. ROOT
  - Production jobs: 95% of workflows download data to local disk
  - User jobs (Grid or interactive): in principle any access protocol can be used (currently file: and root: are preferred)
- URLs vs LFNs
  - In LHCb, all URLs can be constructed from the LFN
  - From 2006 (in)famous Mumbay WLCG agreement: Dirac was based on SRM
    - All other URLs (tURLs) returned by SRM
    - Service class independent on namespace

Dirac and LHCb status

- URL in FC
  - Irrelevant… Must be unique for the LFC as used for removing replicas
  - Currently: SURL at creation time
- SURL from FC
  - Constructed by DMS from LFN + SE description
  - `srm:<endpoint>[:<port>][<WSUrl>]<SAPath><LFN>`



```
CERN-DST-EOS
  {
    BackendType = Eos
    AccessProtocol.1
    {
      Host = srm-eoslhcb.cern.ch
      Port = 8443
      Protocol = srm
      Path = /eos/lhcb/grid/prod
      SpaceToken = LHCb-EOS
      WSUrl = /srm/v2/server?SFN=
    }
  }
```

  - **SURL**
    - **With BDII**
      - `srm://srm-eoslhcb.cern.ch/eos/lhcb/grid/prod<LFN>`
    - **No BDII**
      - `srm://srm-eoslhcb.cern.ch:8443/srm/v2/server?SFN=/eos/lhcb/grid/prod<LFN>`

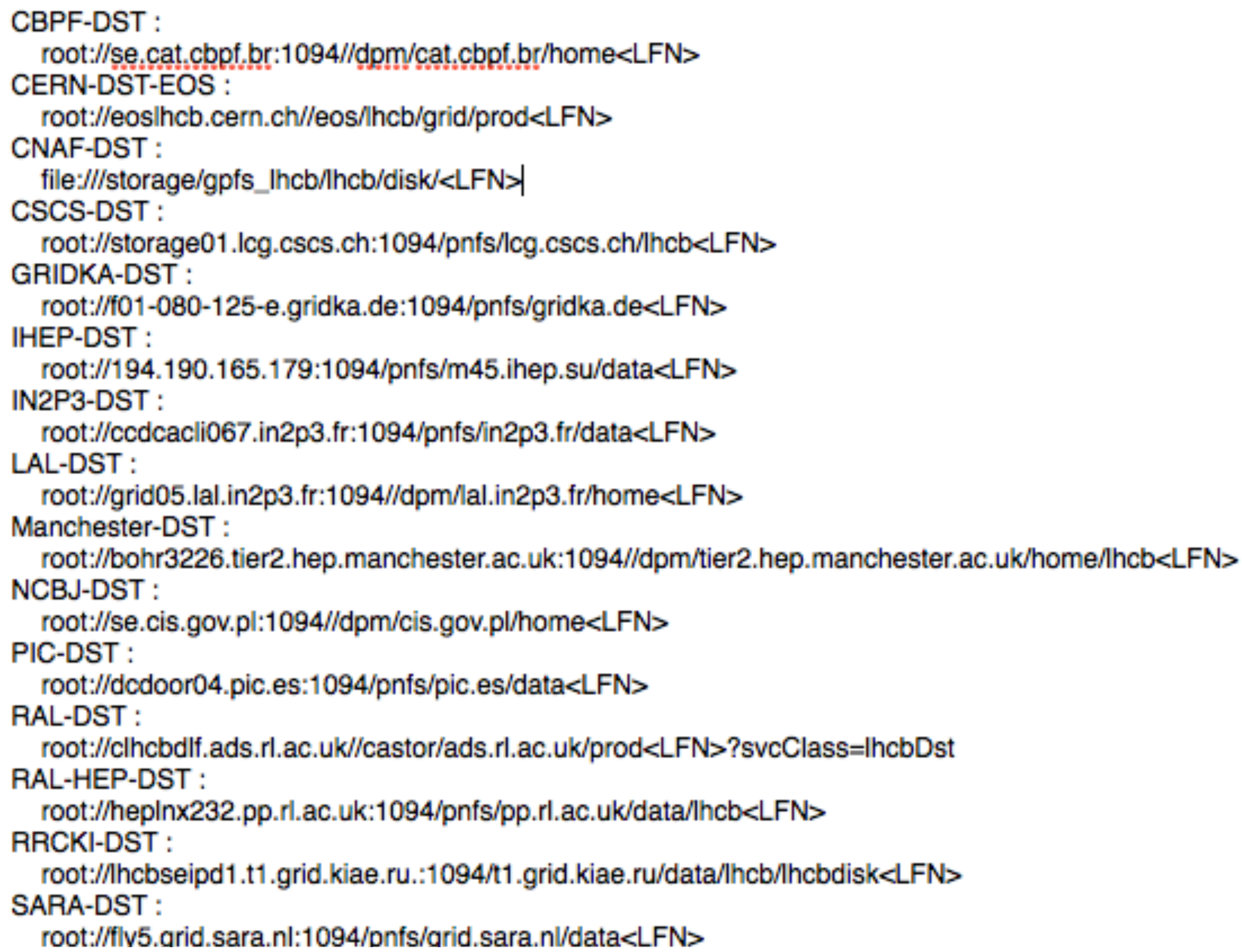
Dirac and LHCb Status (cont'd)

- "No BDII" SURLs always used: BDII never used
- Transfers:
  - In all cases source and destination SRM space tokens are used
    - Avoid disk2disk copy of source
    - Put file in correct service class
  - Replication: SURL passed to FTS3
  - Local transfer: SURL used by lcg-cp (python binding)
- File protocol access
  - tURL requested to SRM
    - Ordered list of protocols (for gfalGetTurl)
      - `file,xroot,root,dcap,gsidcap,rfio`
    - Supported protocols
      - CNAF: file
      - All other sites (Tier0, 1, 2): xrootd
        - xroot for Castor (!) as root is for the "castord" protocol
        - root for all others

Federating storage elements

- Download: from any disk replica (local first)
- For protocol file access (user jobs only)
  - Gaudi/FC federation
- Based on FC and Gaudi
  - Assumption: the FC is almost correct
  - Anyway used for brokering jobs
  - Aim: recover cases when the replica is absent or temporarily unavailable
- Implementation
  - Gaudi uses a local XML catalog with all replicas
  - Replicas are ordered
    - First replica is local
    - Other replicas for the time being random
  - If Gaudi fails to open a replica, it moves to the next in the list
    - Requires xroot to be WAN enables
    - Currently OK at all sites for LHCb

tURLs at LHCb sites



```
CBPF-DST :
   root://se.cat.cbpf.br:1094//dpm/cat.cbpf.br/home<LFN>
CERN-DST-EOS :
   root://eoslhcb.cern.ch//eos/lhcb/grid/prod<LFN>
CNAF-DST :
   file:///storage/gpfs_lhcb/lhcb/disk/<LFN>
CSCS-DST :
   root://storage01.lcg.cscs.ch:1094/pnfs/lcg.cscs.ch/lhcb<LFN>
GRIDKA-DST :
   root://f01-080-125-e.gridka.de:1094/pnfs/gridka.de<LFN>
IHEP-DST :
   root://194.190.165.179:1094/pnfs/m45.ihep.su/data<LFN>
IN2P3-DST :
   root://ccdcacli067.in2p3.fr:1094/pnfs/in2p3.fr/data<LFN>
LAL-DST :
   root://grid05.lal.in2p3.fr:1094//dpm/lal.in2p3.fr/home<LFN>
Manchester-DST :
   root://bohr3226.tier2.hep.manchester.ac.uk:1094//dpm/tier2.hep.manchester.ac.uk/home/lhcb<LFN>
NCBJ-DST :
   root://se.cis.gov.pl:1094//dpm/cis.gov.pl/home<LFN>
PIC-DST :
   root://dcdoor04.pic.es:1094/pnfs/pic.es/data<LFN>
RAL-DST :
   root://clhcbdlf.ads.rl.ac.uk//castor/ads.rl.ac.uk/prod<LFN>?svcClass=lhcbDst
RAL-HEP-DST :
   root://heplnx232.pp.rl.ac.uk:1094/pnfs/pp.rl.ac.uk/data/lhcb<LFN>
RRCKI-DST :
   root://lhcbseipd1.t1.grid.kiae.ru.:1094/t1.grid.kiae.ru/data/lhcb/lhcbdisk<LFN>
SARA-DST :
   root://fly5.grid.sara.nl:1094/pnfs/grid.sara.nl/data<LFN>
```

What can we do?

- URL creation
  - Dirac allows without problem to build URLs for any protocol
  - Matter of writing a plugin and defining in CS
  - XROOT plugin exists, http can easily be made
    - Was waiting for the official release of the python binding, OK
    - Being commissioned this week
  - Generic gfal2 plugin is also under development
- Download to WNs or desktops, laptops, local clusters
  - Fairly easy, we can use xrdcp (Dirac XROOT plugin)
- Protocol access
  - Easy, create the tURL
- Staging from tape
  - We use FTS3, replicating to a disk SE
  - Allows to considerably decrease the disk cache
  - SRM must be used
- Replication
  - See next slide for usage of gsiftp or xroot

File upload and FTS replication

- Major obstacles for getting rid of SRM
  - i.e. create xroot tURL and use xrdcp (upload), or gsiftp tURL
- Increasing order of importance
  - We need a single xrootd (local redirector) endpoint at all sites
  - See slide 6: it is not a zoo but a jungle
  - Only few sites have a single stable alias
    - Several sites gave us an alias, which is not used by SRM to return the tURL, but can be used from the WAN
  - Efficiency (for gridftp):
    - Some SEs provide servers directly on disk server
    - Can gridftp redirectors be deployed?
    - Alternative: use dedicated gridftp set of machines
  - Destination service class
    - Currently we use SRM spaces
    - Impossible with gridftp or xrdcp
    - See next slide

Selecting destination service class

- Basically: tape or disk backend
  - Only at Tier1s (no problem at Tier2s for DPM or dCache)
  - Castor
    - CERN: OK as only tape (EOS used for disk)
    - RAL: two Castor instances? (until RAL moves away from Castor for disk)
  - StoRM
    - Uses namespace: OK with gridftp and xrdcp
  - dCache
    - AFAWK there is no way
    - Currently no namespace difference between tape and disk
    - Gridftp and xroot tURLs are undistinguishable
      - Service class is selected by srmPrepareToPut
    - Solutions?
      - Change namespace, but requires changing all existing files ☹
      - Two gridftp and xrootd instances? Is this enough? Do we need two dCache instances?
    - Additional problem
      - Currently pools are virtual on a large number of disk servers
      - A dedicated set of servers for tape may jeopardize scalability

Conclusion

- Commissioning Dirac for using xrootd and tURL creation for some use cases
  - Protocol access
  - File download
  - Proviso all sites publish WAN accessible xroot local redirectors (stable name)
- SRM still needed for tape handling (bringOnline, pin)
- Infrastructure problems
  - Scalability for Castor
    - Or keep SRM for RAL until replaced for disk
  - Destination service class selection for dCache
    - Split tape and disk also at site level
- Longer term
  - http/webdav dynamic federation being set up (thanks Fabrizio and Stefan)
  - Could be used in longer term as transfer and access protocol
    - Easier as unique URL space, still the service class is an issue