CMS sites: DOME and AAA config.

A. Sartirana

Outline

Couple of (**unrelated) CMS-specific** topics

- **migration to DOME** of a CMS site
  - a couple of **issues** strictly **related to the CMS workflow**
    - **fixed** with **1.10;**
  - **feedback** on the tests/experience at **T2_FR_GRIF_LLR**
    - I ignore if other CMS sites migrated and their feedback;
    - so far we have **only** tested the migration on the **testebed;**
- **AAA configuration** for DPM sites
  - **traditional conf.** turned out being **wrong**
    - **fixed in 2017** with a new standard conf.;
  - I still have a **couple of question** about why the old configuration was wrong
    - introductory to Marian talk.

DOME

DOME Migration turned out to be more tricky than expected

- **T2_FR_GRIF_LLR testbed** is actually **running in dome** mode since ~Q1/2016
  - **SRM-less** CMS workflows have been **tested** (with gftp redirection) and have shown **no problems** (or they were reported and promptly fixed);
  - **but** migrating a **prod endpoint** with all its existing files is a **different matter;**
- on **12/2017** things seemed ready to finally seriously think to **migrate to DOME**
  - but a **realistic test** of migration (populating the testbed as a real endpoint) showed **blocking problem**
    - related to **token-less SRM writings.**

CMS-workflows

A bit in detail…

- all CMS-workflows can be made DOME-compliant (an/or SRM-less)
  - successfully tested **PhEDEx** with **gftp redirection;**
  - **jobs stageout** can be configured to use gfal with **gftp** (checksum pbs with xrootd and http)
    - crab stageout can be configured to use gftp as well;
    - no control on "private" clients but this may not a big issue;
- the real problem is that **CMS does not use ST**
  - so either we go SRM-less or we don't
    - at least up to 1.9;
  - and, most of all, we have to migrate a prod. storage full of **existing ST-less files ...**

With dpm-1.9

Realistic test of migration.

```
POOL p1 DEFSIZE 0 GC_START_THRESH 0 GC_STOP_THRESH ….
CAPACITY 799.73G FREE 296.97G ( 37.1%)
 llrpp02.in2p3.fr /data1 CAPACITY 399.87G FREE 145.82G ( 36.5%)
 llrpp03.in2p3.fr /data1 CAPACITY 399.87G FREE 151.16G ( 37.8%)
```

Filling the pool **with ST-less files.**

now we've to create a QT (e.g. on the root path)

- with a **«big» QT** (e.g. 780GB) **ftp/xrootd/http** are **fine** but **SRM ST-less fail** with SRM_NO_FREE_SPACE
  - because the associated **ST reserve 780GB of the 296GB** that sees free as the existing file are not in the ST;
- with a **«small» QT** (e.g. 200GB) **SRM** is **fine** but **ftp/xrootd/http fail** with «no space left»
  - because the **free QT space is 200GB** while the **counter** of the root dir. says there are already **500GB used.**

We are locked…

With 1.9

A possible ugly way out...

- create a **«big» QT** and **move** all the **existing files** there with a query on the DB...

```
MariaDB [cns_db]> update Cns_file_replica set setname='30927837-de8a-11e7-8db1-001a2b3c4d05' where
poolname='p1';
Query OK, 242 rows affected (0.08 sec)
Rows matched: 242  Changed: 242  Warnings: 0

MariaDB [cns_db]> update dpm_db.dpm_space_reserv set u_space=u_space - (select sum(f.filesize) from
Cns_file_metadata f, Cns_file_replica r where r.fileid=f.fileid and r.poolname='p1') where rowid=1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

- then you still have to **handle ST-less SRM** writings
  - reduce **SRM** usage **at minimum;**
  - **periodically «migrate» new files** into the ST and upgrade counters.

clearly **not** feasible **in production**...

With 1.10

DPM developers **provided the solution in 1.10** (thanks!)

- in 1.10 **ST-less** SRM **writing** are accounted in the **ST of their base path**

```
MariaDB [cns_db]> select setname from Cns_file_replica where sfn like
'%test.nospacetoken.big.201805171241%';
+--------------------------------------+
| setname                              |
+--------------------------------------+
| bc350e05-bce7-41e0-baeb-6dfd0d3196fc |
+--------------------------------------+
1 row in set (0.00 sec)
```

ST of /dpm/in2p3.fr/home/cms path

- we can create a **global QT** with (size of the CMS pool) and leave existing SRM files;
- all **new writings work fine** and are correctly accounted
  - **also SRM ST-less** writing: no need to jump to full SRM-less;
- **counters are wrong** as the files are not in the ST
  - **eventually fixed** as new file replace the old ones;

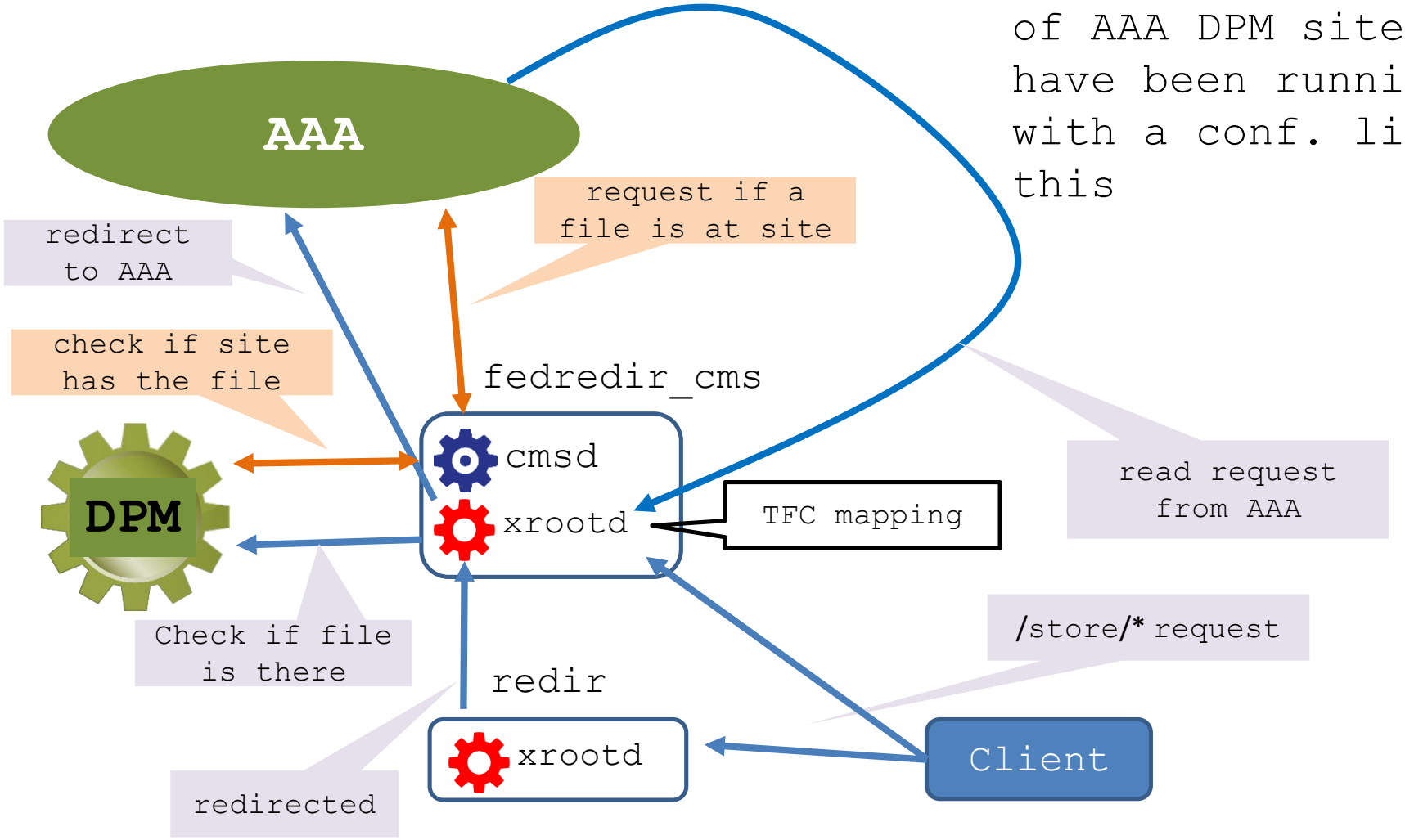
Ready to migrate

Looks like we are **ready to migrate to DOME** in a simple and transparent way which is compliant with CMS workflows

- ... unless I'm (again) missing some detail;
  - maybe we will have to temporary hack a bit the space report;
- **waiting for** the **1.10** to be released in production
  - some bug to fix: xrootd writings not creating the daily directory (but this does not impact CMS), ...
- BTW: there still something that IMHO could be more admin-friendly
  - we have to explicitly list all the groups that have the permission to access a QT. Would it be possible to have catch-all definition (like for the pool) or to use regexp?

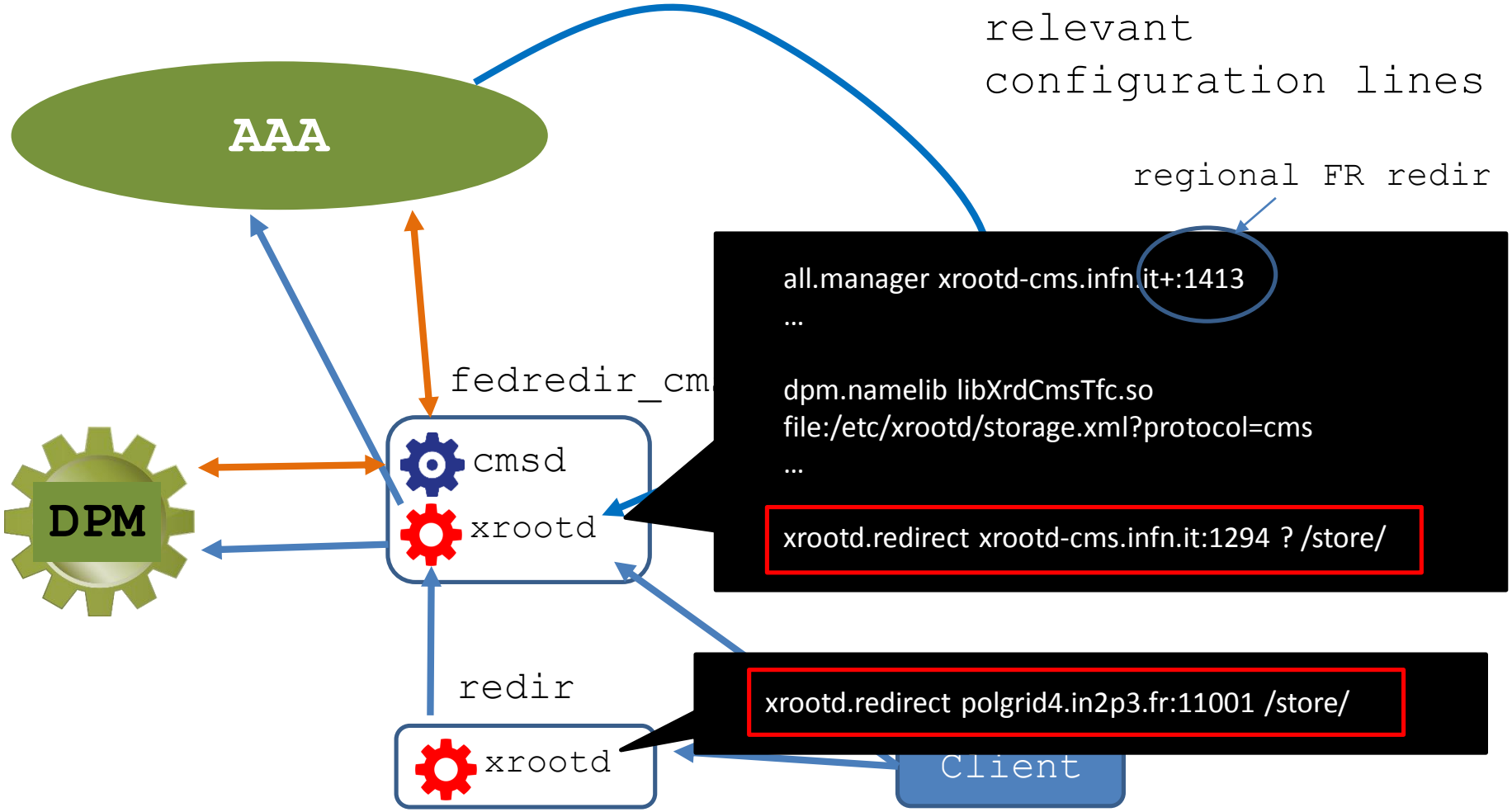
AAA config

Since the beginning of AAA DPM sites have been running with a conf. like this



AAA config

Here are the relevant configuration lines



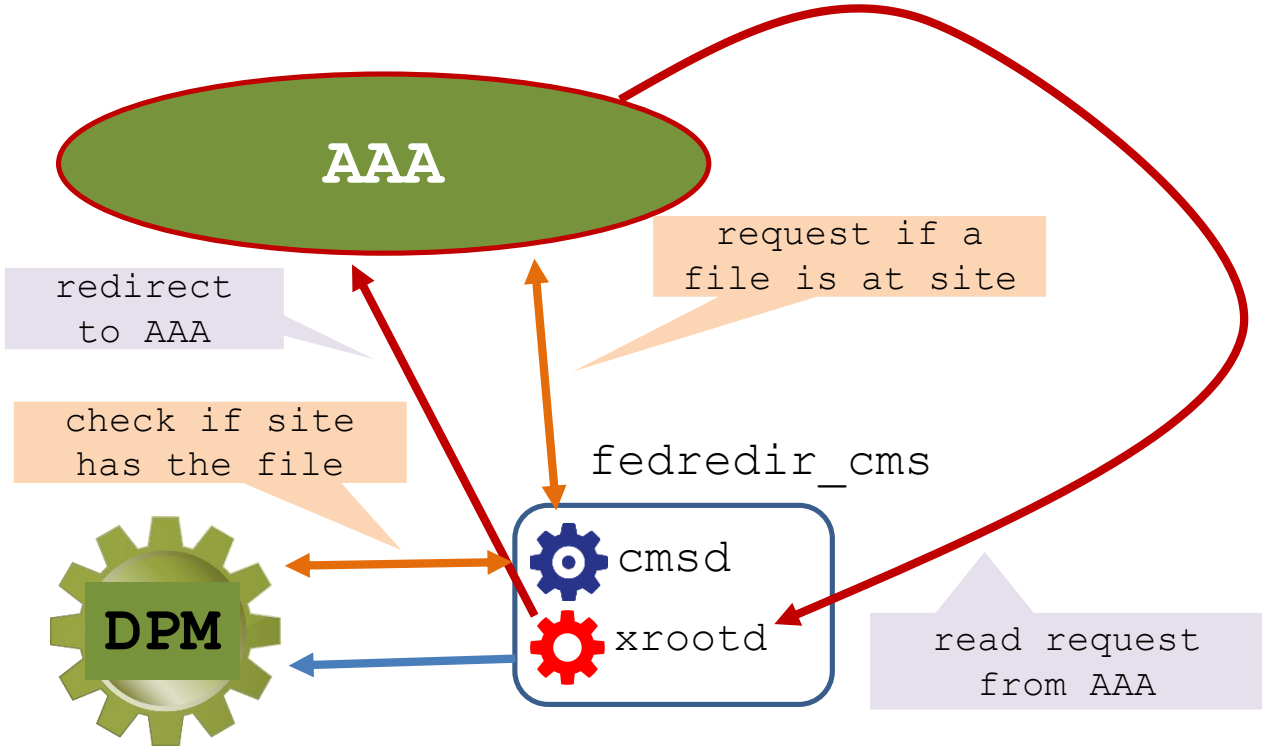
AAA config

- Such setup **seemed** pretty **sensible** and has been running without problems for years
  - modulo some pbs we had when still using xrootd proxy;
  - more or less **C/P from FAX** conf.;
  - pretty much **like** the **regional redirectors** are configured
    - naively: endpoint redirector as a special case of regional redir. with only one endpoint in the region;
- recently (end 2017) we discovered that **this was not what CMS was expecting**
  - **the xrootd AAA service** (p. 11001) **should not redirect**
    - which I thought was the expected behavior…;
  - problem was discovered thanks to new sam tests
    - org.cms.SE-xrootd-contain at [*].

[*]https://etf-cms-prod.cern.ch/etf/check_mk/index.py

AAA config

I think I've understood that the problem is that **this may cause a loop** (but I may be completely wrong...)



If asked by the AAA for a file which is not on storage, the fedredir_cms will loop back to the AAA (which may render the way AAA finds file unpredicable).

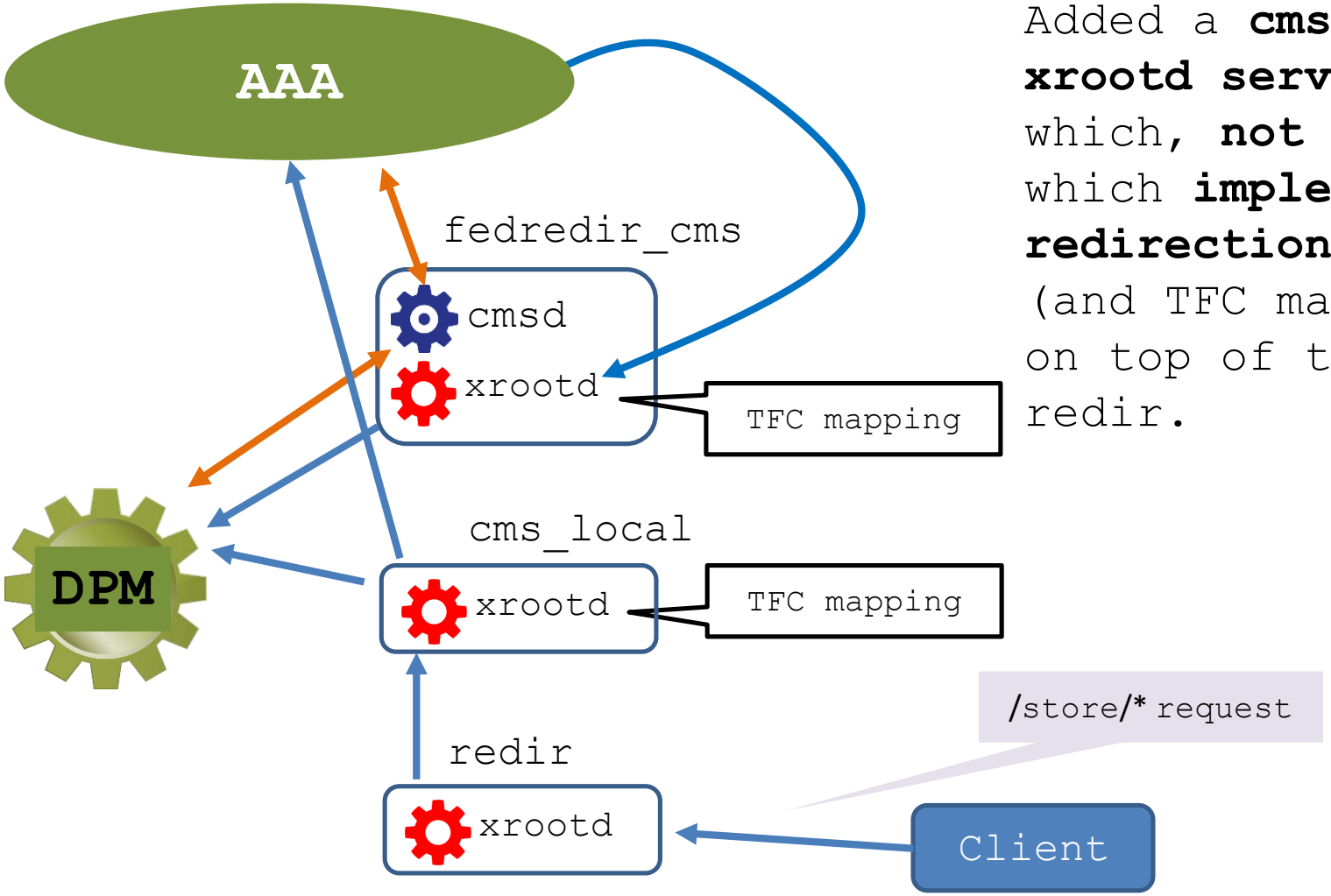
AAA config

- Some of the **symptoms** seen which seem related to this misconfiguration
  - job logs showed that **files only available at** their **site** were **"served" via** an xrootd endpoint of **another site;**
  - during IPv6 comm. sites cannot read file because the **"problematic" site tried to serve even** in case the **tried=problematic** site was **in the URL;**
- one thing I don't understand is that, **if cmsd "publish"** files **correctly** to AAA (and it does), we should **never** enter these **loops/wrong redir.;**
- can this be a problem with **regional redirectors**?
  - (Marian) nature of regional redirectors config. is **completely different** and we haven't noticed issues there.

New config

- We (quickly) had to come up with an **new configuration** corresponding to CMS expectation
  - a.k.a **no redirection** on the fedredir (11001) endpoint;
  - … but this implies no redirection at all
    - unless putting redirection and TFC mapping directly in the dpm-redir which is not ideal as this is a CMS-specific part (in particular not good for multi federation sites);
    - … or put it somewhere else;
- we wanted to **preserve the redirection** (and TFC mapping) for read requests accessing dpm-redir (1095)
  - **user** (at least ours) are by now **used to ask for** '/store/*' **files** to our endpoint (simply putting the LFN in their job config) **and transparently find them** wherever they are in the federation.

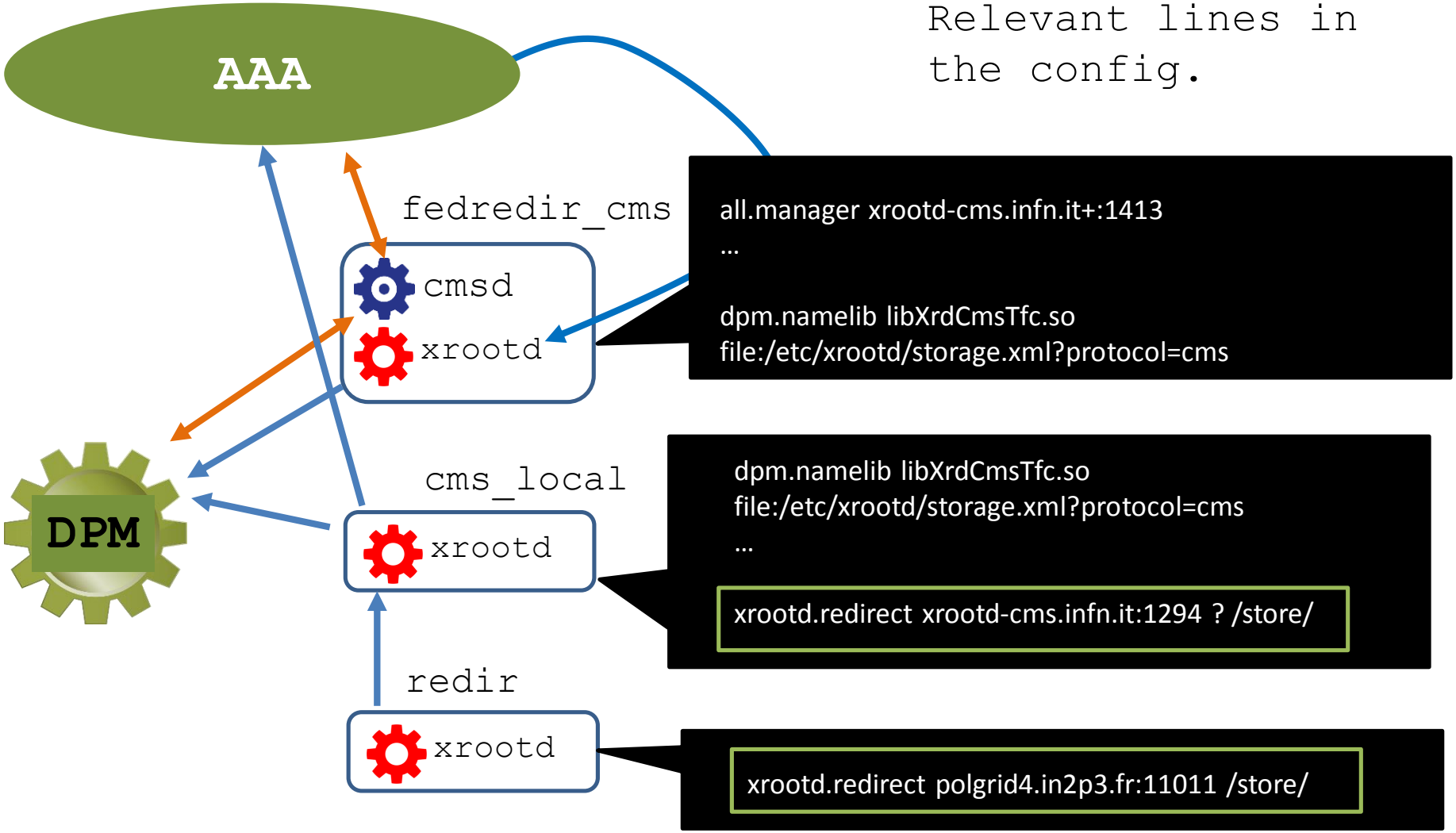
New config



Added a **cms «local» xrootd service** which, **not in AAA**, which **implements redirection** to AAA (and TFC mapping) on top of the dpm redir.

New config

Relevant lines in the config.



New config

- This is running since some time now and **all seems fine**
  - **minor fixes in puppet** modules and a new default puppet conf.;
  - should be deployed everywhere (SAM tests check it);
- personally, there are still **few things I'm puzzled about**
  - which is the exact workflow/sequence of events that could trigger the problem;
  - what about **non-DPM sites in AAA**, were they running the correct conf. since the beginning?
  - what about **Atlas FAX**?