Adding a caching volatile pool to a DPM - case study

Davide Michelino
INFN - Napoli
DPM Work Shop - Bern 13-14 June 2019

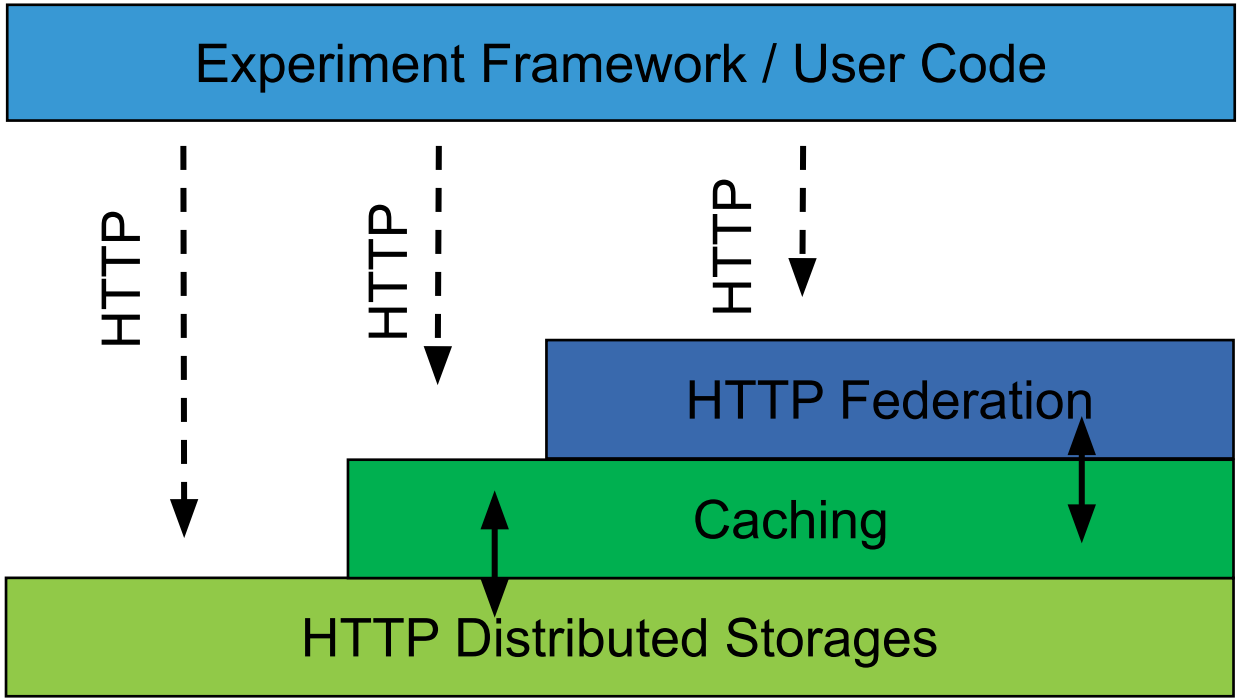
Outline

- ➔ DPM Testbed setup in Napoli
- ➔ Cache Setup: Volatile Pool + Dynafed
- ➔ Script details
- ➔ Cloud-Hammer test
- ➔ Plans

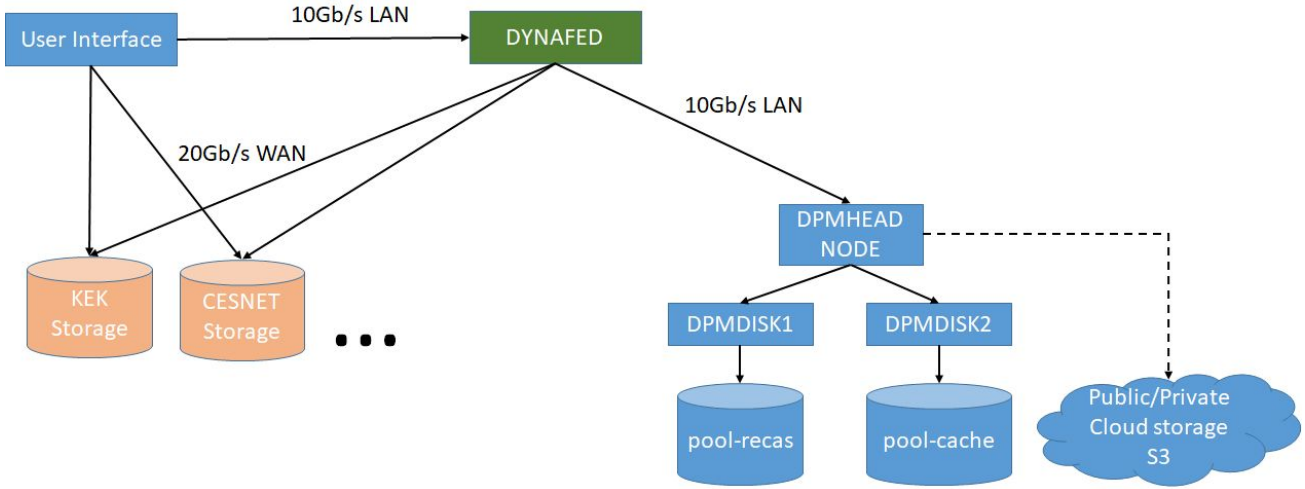
DPM Cache Setup

- 1 Physical Head Node
- 1 Physical disk node
  - 4x 19TB filesystems
- 1x 19TB “traditional” pool
- 1x 19TB “volatile” pool
- CentOS7 & Dmlite version 1.12 with dome fully enabled
- Gridftp redirection
- Custom scripts for stat/pull operations

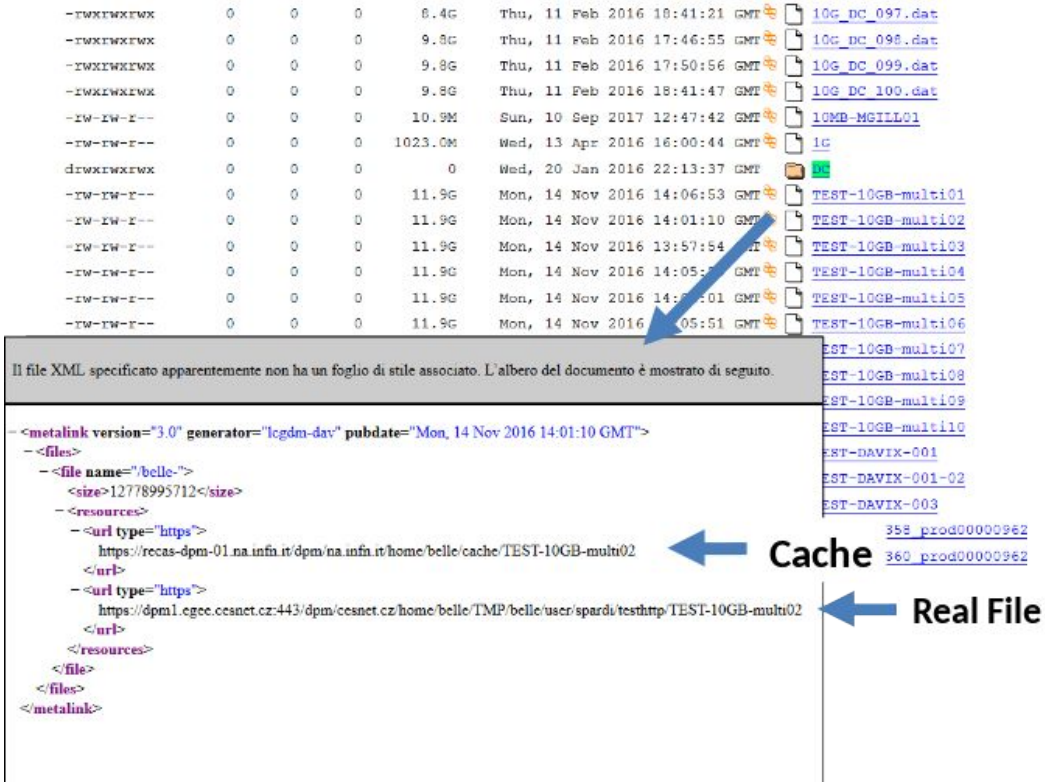
Cache Architecture



Cache Setup: Volatile Pool + Dynafed

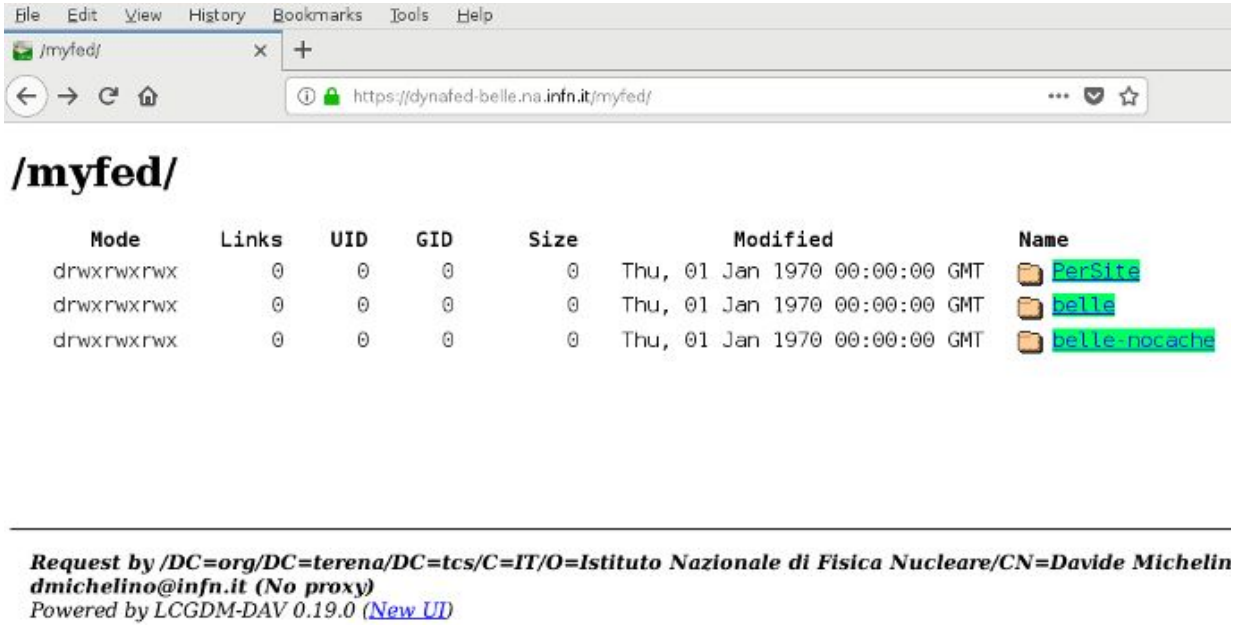


Dynafed configuration



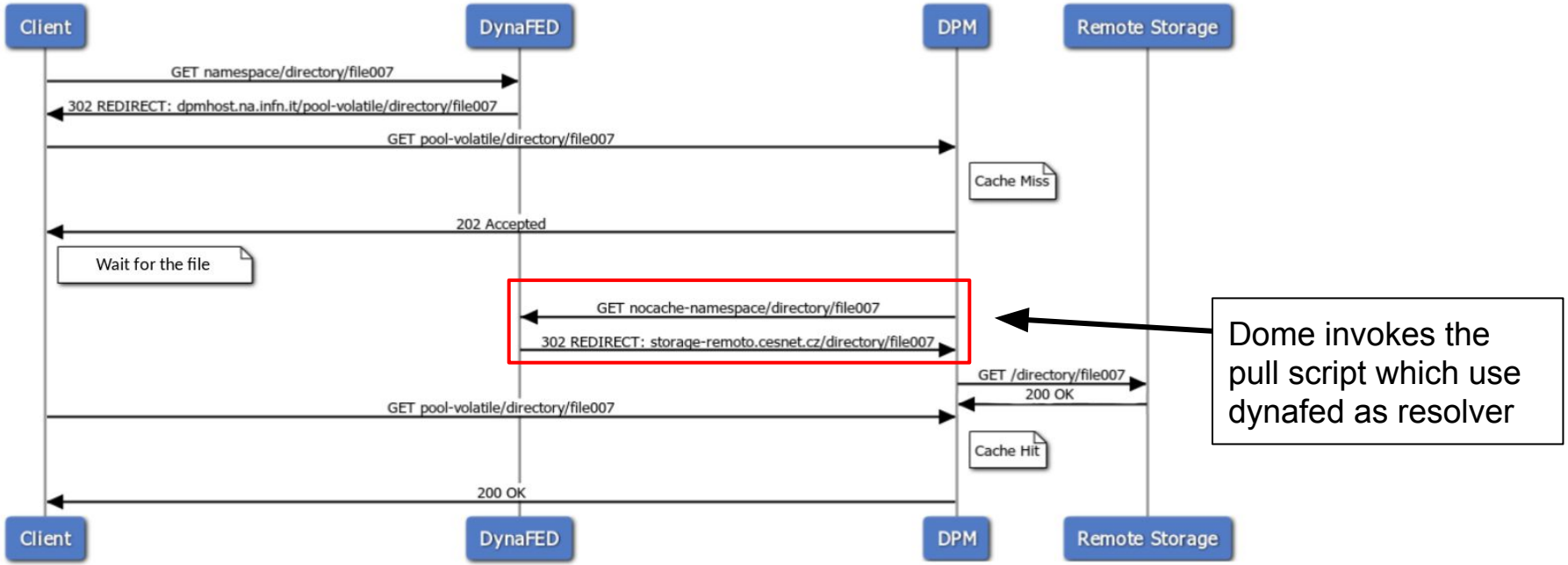
Dynafed is configured to aggregate the volatile pool with all the experiment endpoints, so every file has at least two replicas: the one in the actual storage and the one provided by the volatile pool (even if the latter does not exist yet).

Dynafed as resolver



Dynafed provides an additional aggregation with all the experiment storage but the volatile pool. This aggregation is used by dome's scripts to resolve the location where fetch the requested data.

Dynafed as resolver



Dome scripts details - File Stat

```
PATHL=`echo $1 |cut -d/ -f 7-`
DYNAPATH="davs://dynafed-belle.na.infn.it/myfed/nocache/$PATHL"
STAT=$(gfal-stat $DYNAPATH; exit $?)
STATout=$?
stat_size=`echo $STAT|cut -d' ' -f 4`
stat_type=`echo  $STAT|cut -d' ' -f 5`
if [ "$stat_type" = "regular" ]; then
   echo ">>>>> STAT $stat_size"
elif [ "$stat_type" = "directory" ]; then
   echo ">>>>> STAT 0 16384"
else
   exit 1
fi
exit 0
```

Resolver url

Just a stat!

Check if it was a file or a directory and tell it dome

Dome scripts details - File Pull

```
PATHL=`echo $1 |cut -d/ -f 7-`
DYNAPATH="davs://dynafed-belle.na.infn.it/myfed/nocache/$PATHL"
echo `date` pull: download of $DYNAPATH in $2 >> /tmp/pull.log
davix-get $DYNAPATH $2 >> /dev/null 2>&1
PULLstatus=$?
if [ $PULLstatus -eq "0" ];then
   echo "`date`: OK ($PULLstatus)" >> /tmp/pull.log
else
   echo "`date`: KO ($PULLstatus)" >> /tmp/pull.log
   exit 1
fi
echo `date` pull: Done >> /tmp/pull.log
exit 0
```

Resolver url

Just a get!

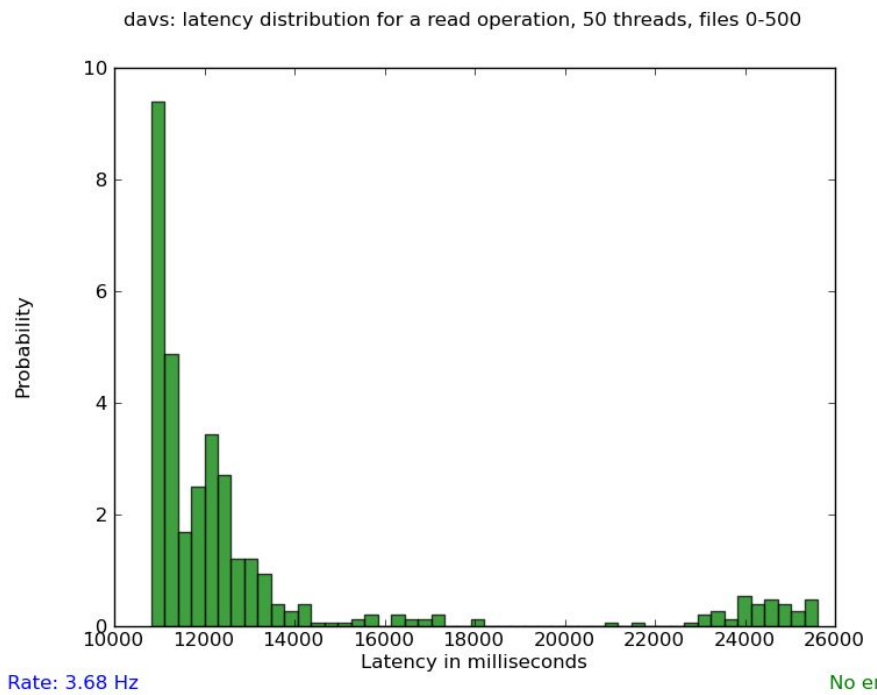
Check if the get was successful and tell it dome.

Grid-Hammer test

- Some tests have been performed with the “grid-hammer” framework in order to simulate concurrent access and test the system stability and its performance.
  - 1 grid-hammer instance with 50 parallel threads to test 500 operations
  - Dpm with 1 disknode configured with a limit of >50 parallel pulls per node
  - Tested read operations both with the cold and warm cache
  - The same tests have been performed regarding stat operations
  - Something to remember:
    - Hammer-cloud performs read tests with very small files (few bytes)
    - Davix by default waits 10 seconds on 202
    - All the requests target dynafed

Grid-Hammer - Read operations

Cold cache

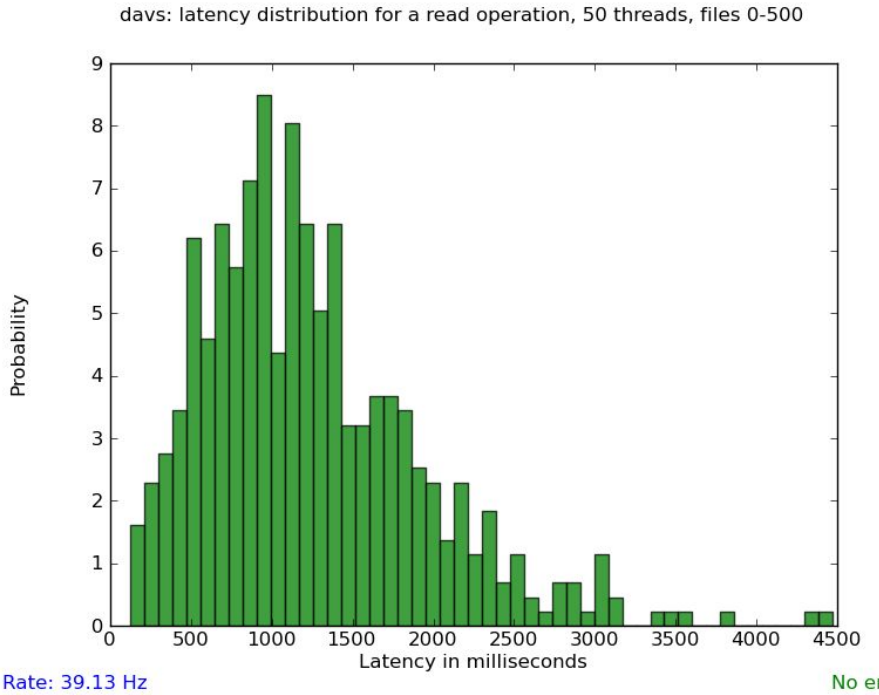


Rate: 3.68 Hz

No errors

Min 10.82 Max 25.62

Warm cache



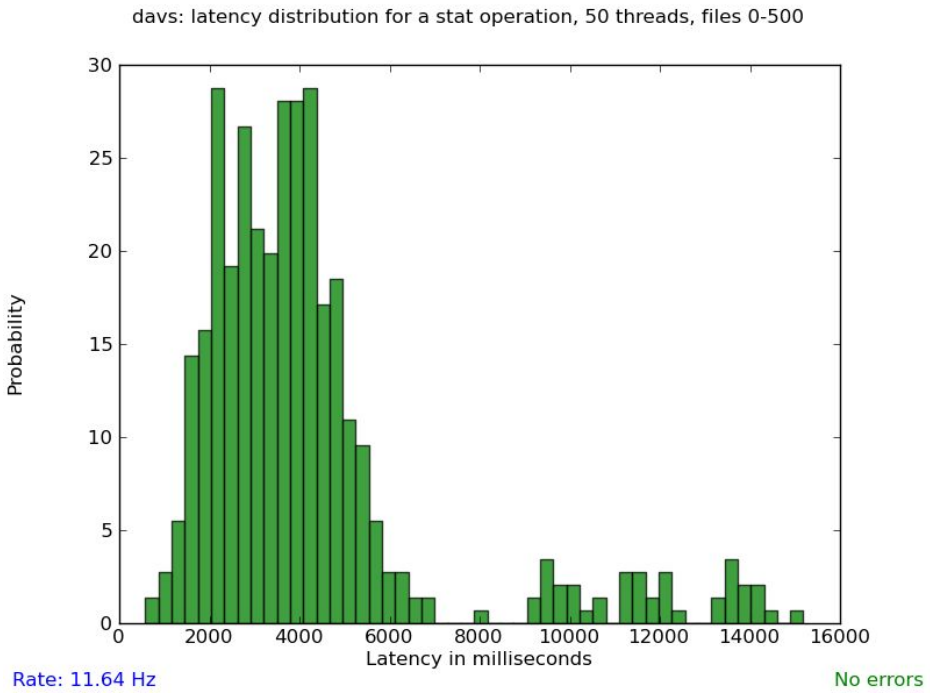
Rate: 39.13 Hz

No errors

Min 0.12 Max 4.47

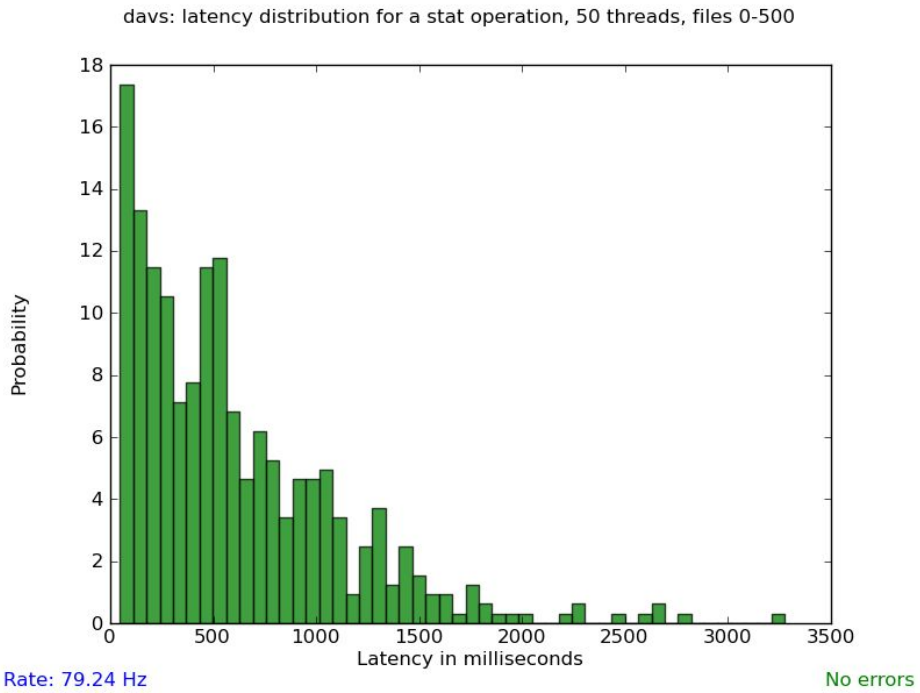
Grid-Hammer - Stat operations

Cold cache



Min 0.58 Max 15.19

Warm cache



Min 0.05 Max 3.27

Grid-Hammer - Outcome

- Performed tests prove the overall system reliability
- More performance oriented test can be done with **bigger files** or with **shorter waiting time**
- Something can be tuned or investigate more
  - In case of transfer failure investigate the use of metalink

Polling issues with Dynafed

Sometime dynafed fail to poll some of storages (the volatile pool for example), after this event the storage is removed from the list in all metalinks for a while. The procedure seems to be affected by an high rate of “false positive”

Suggestion:

Allow admins to tune the polling parameters in the configuration file i.e.:

- Number of polling failures before remove a storage temporarily
- # of positive test required before put the storage in the list again
- Related timeouts (if any)

Projects and collaborations

- IDDLS (Italian distributed datalake for science, INFN internal project)
- XDC project
- DOMA Access

Plans

In Belle II we are starting to discuss how to use/integrate new storage service in the computing model.

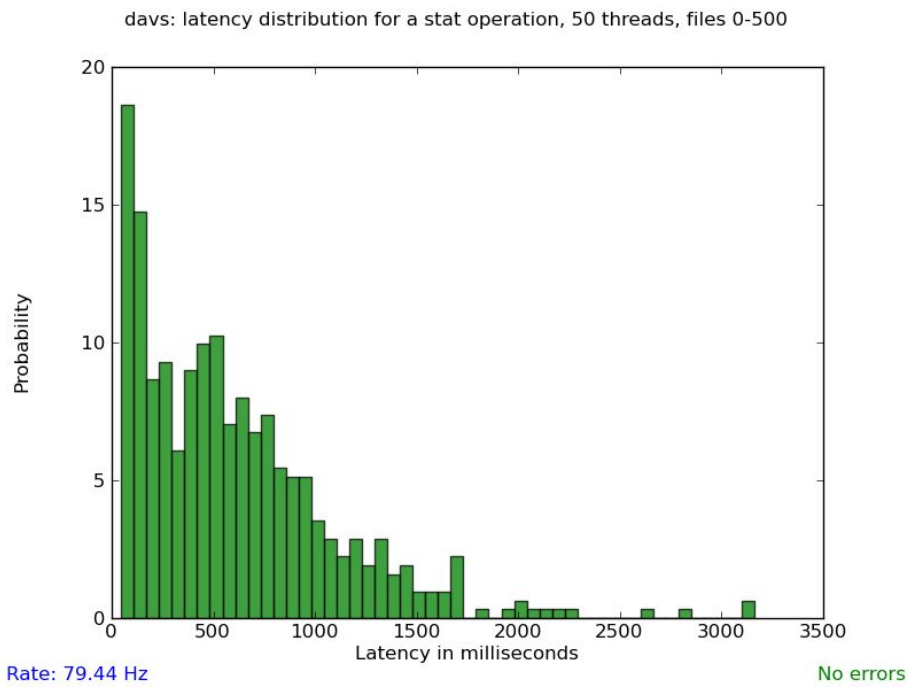
Among them Dynafed and Volatile pools will be presented as possible tools for the community.

- In the current computing model jobs go where data is. Can resource usage be improved running analysis jobs in a site hosting a Cache? (merging and skimming jobs).

BACKUP

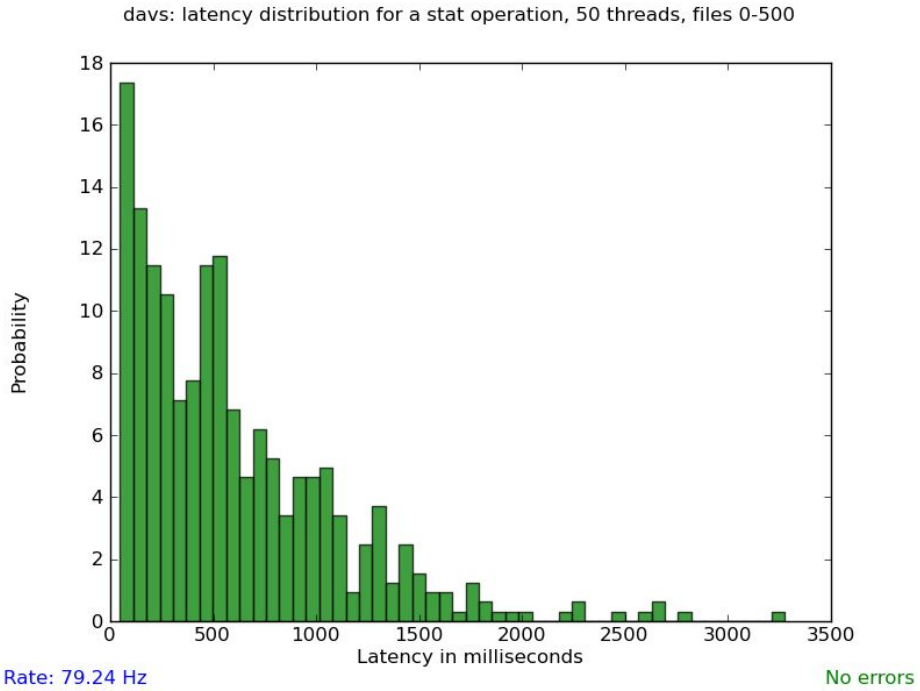
Backup 1: Grid-Hammer - Stat operations cold cache w/ metadata cached by dynafed

Cold cache

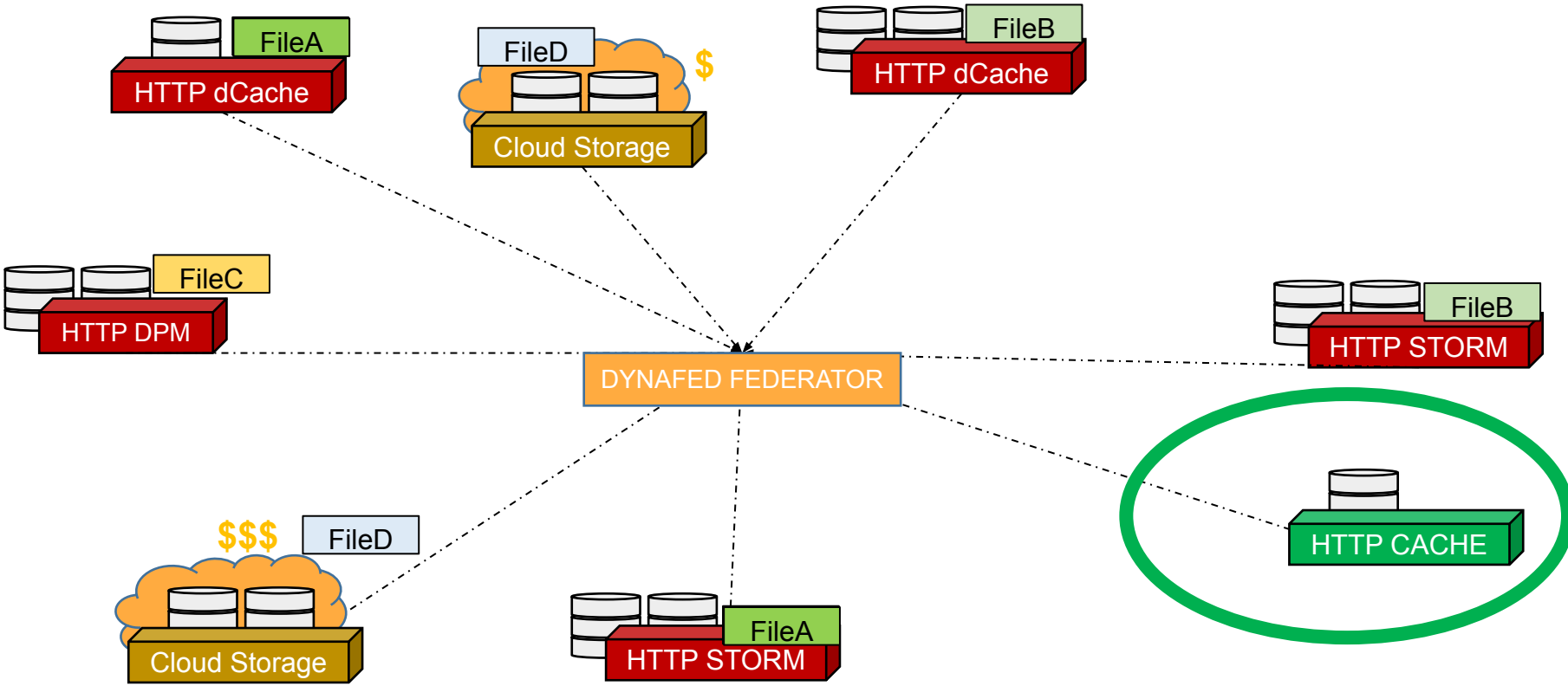


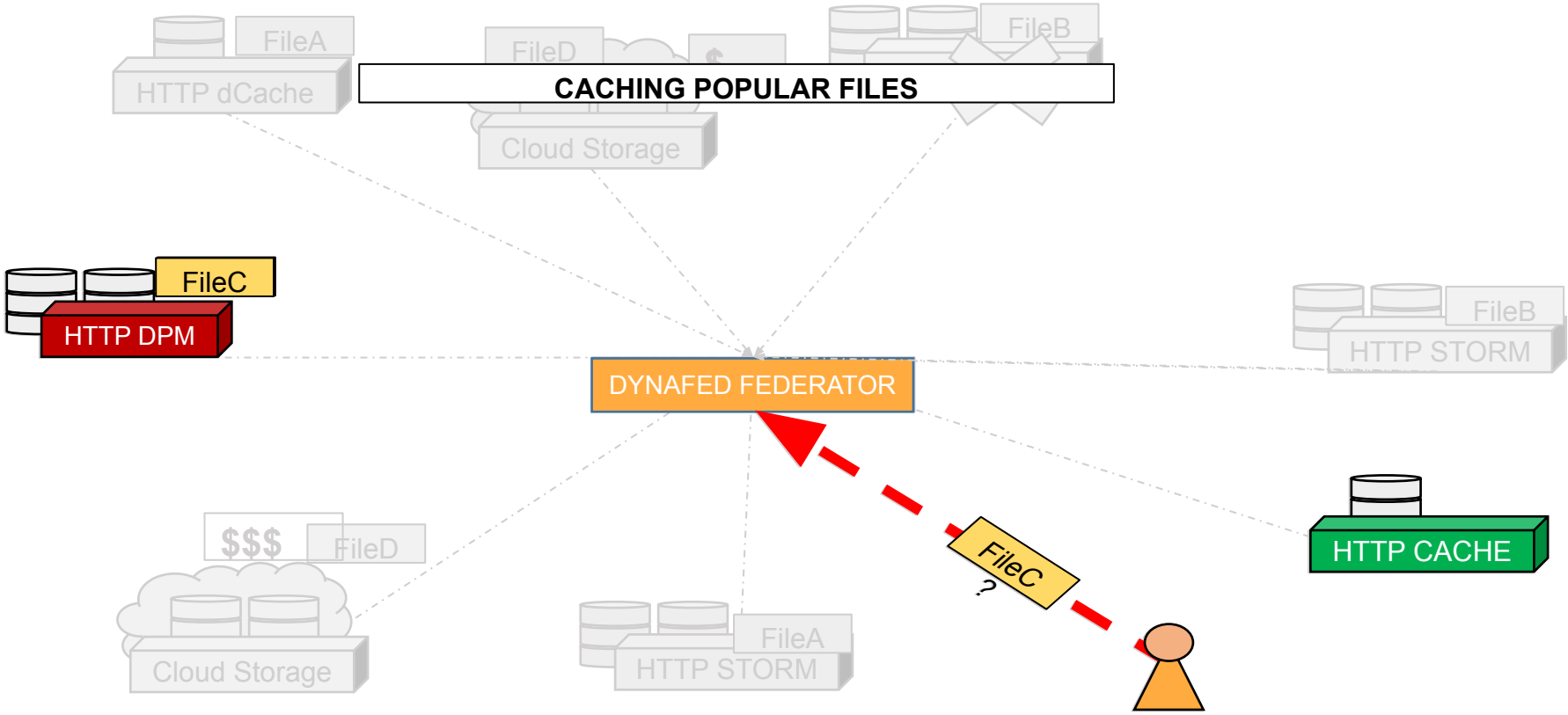
Min 0.05 Max 3.16

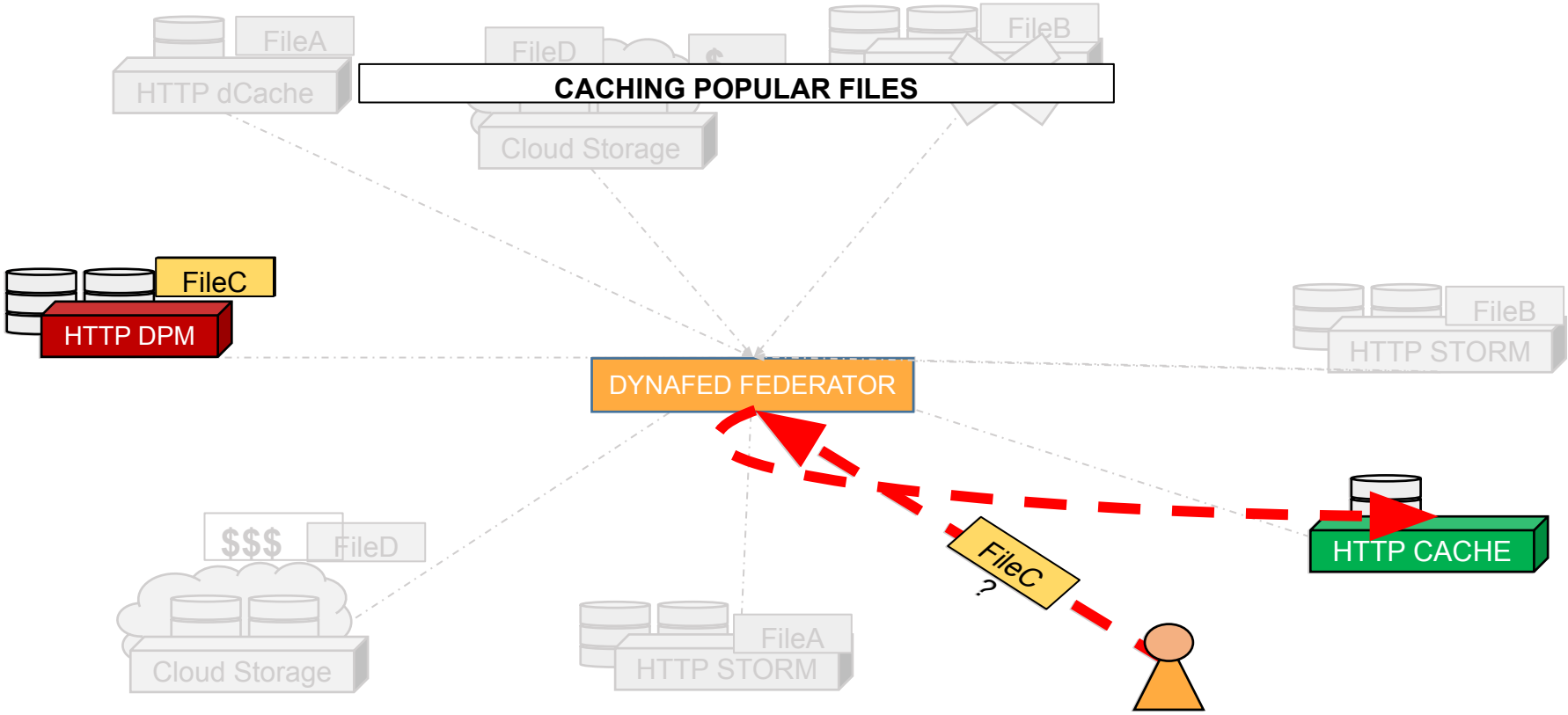
Warm cache

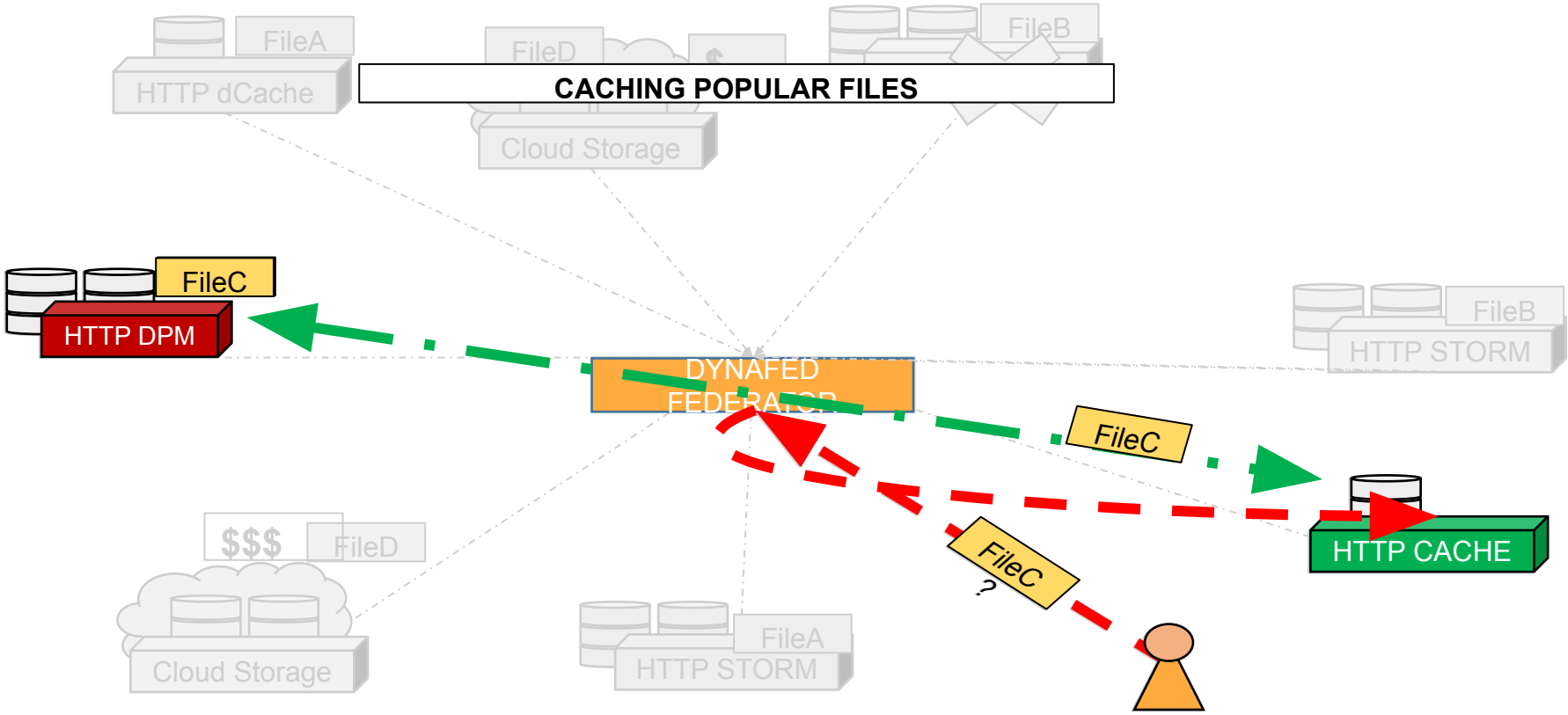


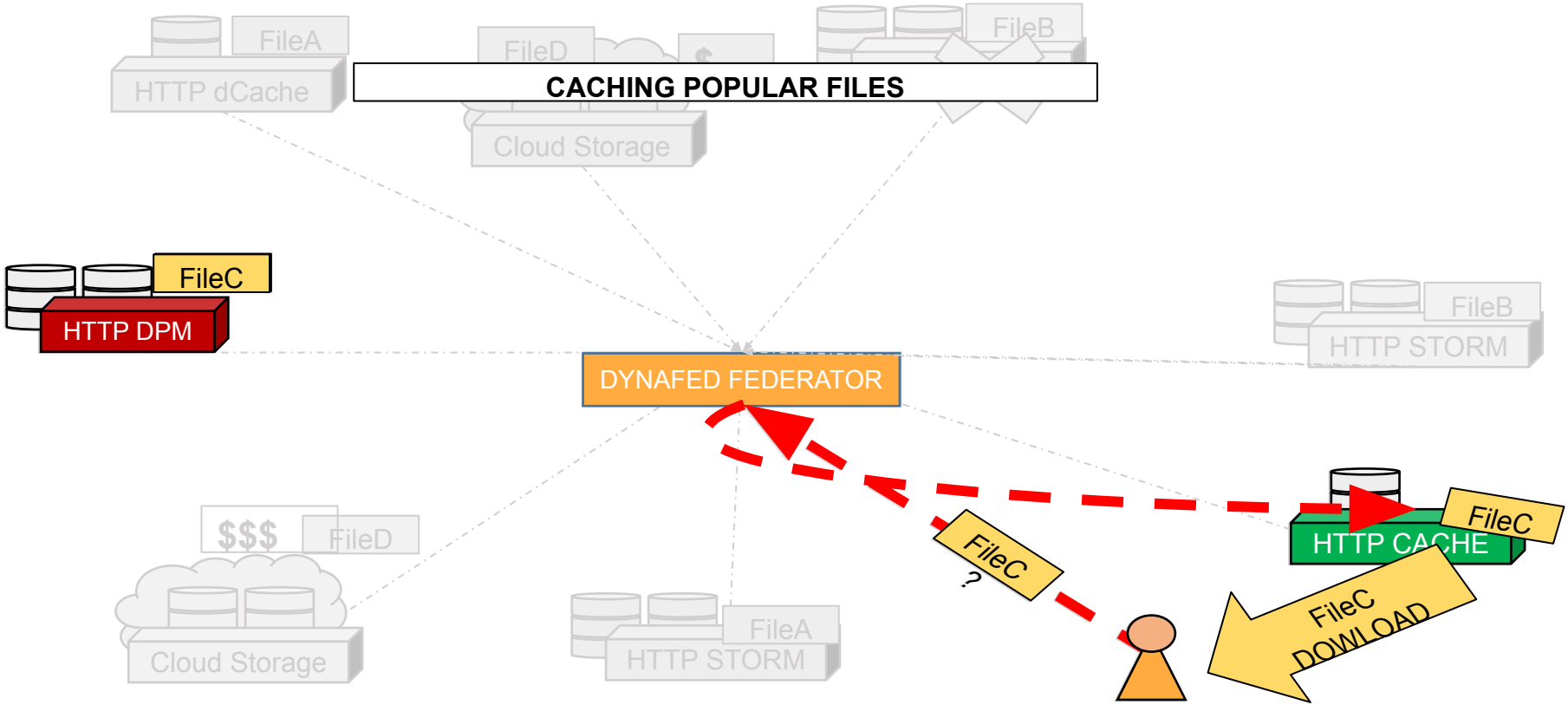
Min 0.05 Max 3.27

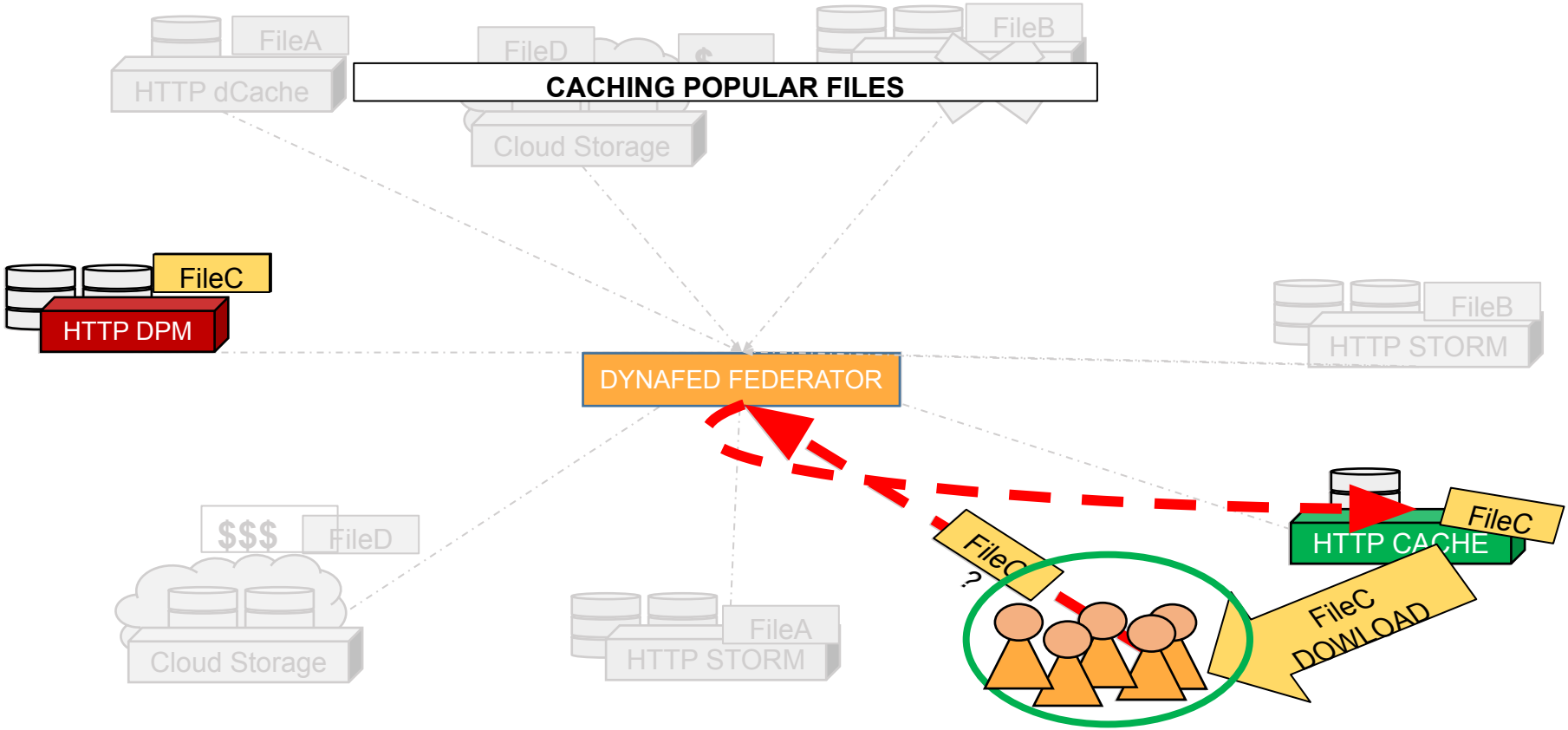

FileA
HTTP dCache
FileD
$
Cloud Storage
FileB
HTTP dCache
FileC
HTTP DPM
DYNAFED FEDERATOR
FileB
HTTP STORM
HTTP CACHE
$$$
FileD
Cloud Storage
FileA
HTTP STORM

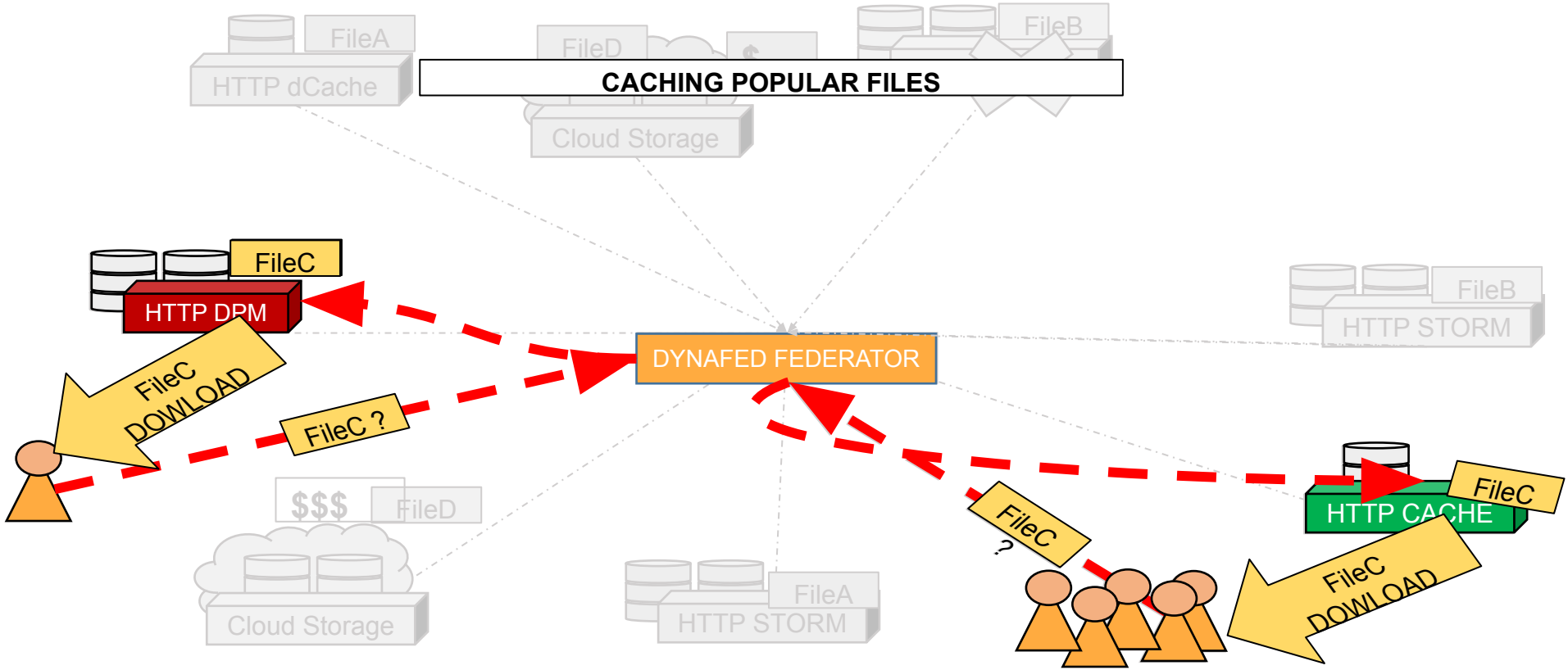

FileA
HTTP dCache
FileD
Cloud Storage
FileB
CACHING POPULAR FILES
FileC
HTTP DPM
FileB
HTTP STORM
DYNAFED FEDERATOR
HTTP CACHE
FileC
?
$$$
FileD
Cloud Storage
FileA
HTTP STORM

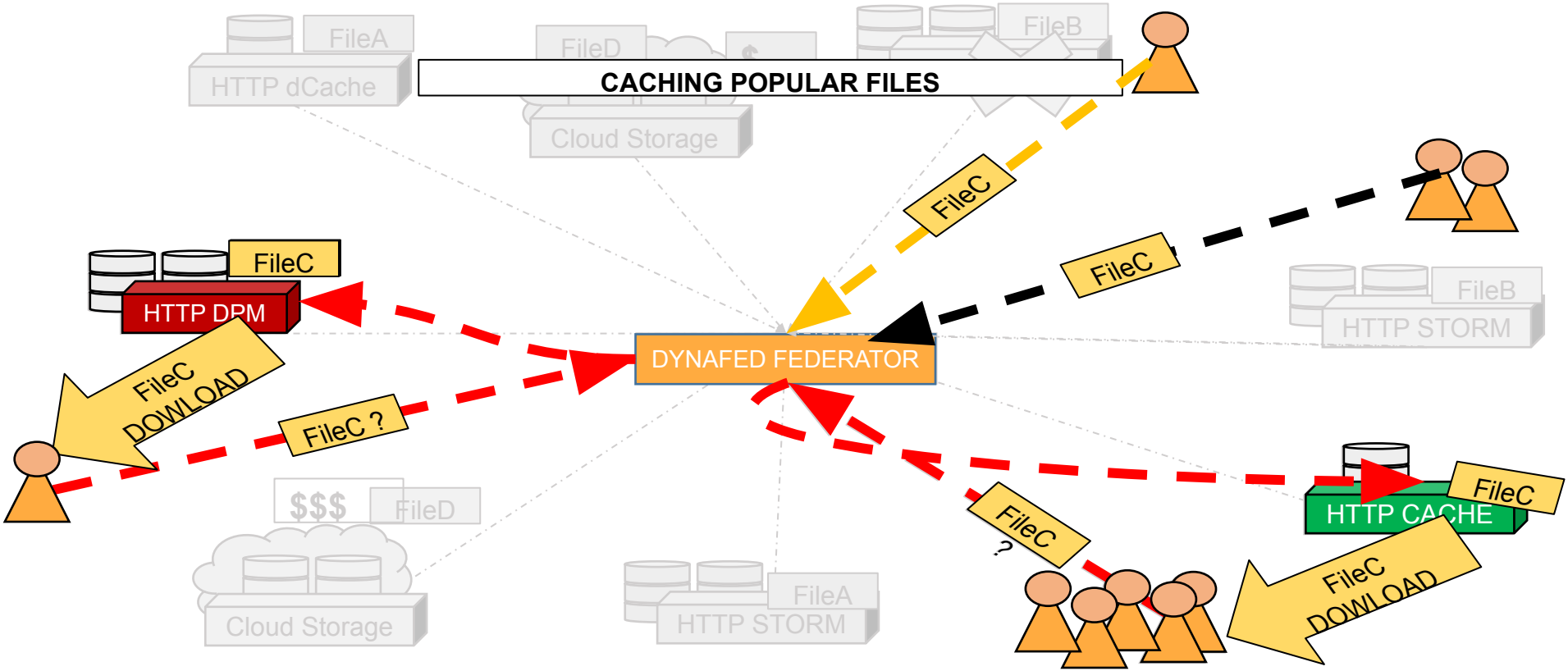

FileA
HTTP dCache
FileD
Cloud Storage
FileB
CACHING POPULAR FILES
FileC
HTTP DPM
FileB
HTTP STORM
DYNAFED FEDERATOR
HTTP CACHE
FileC
?
$$$
FileD
Cloud Storage
FileA
HTTP STORM

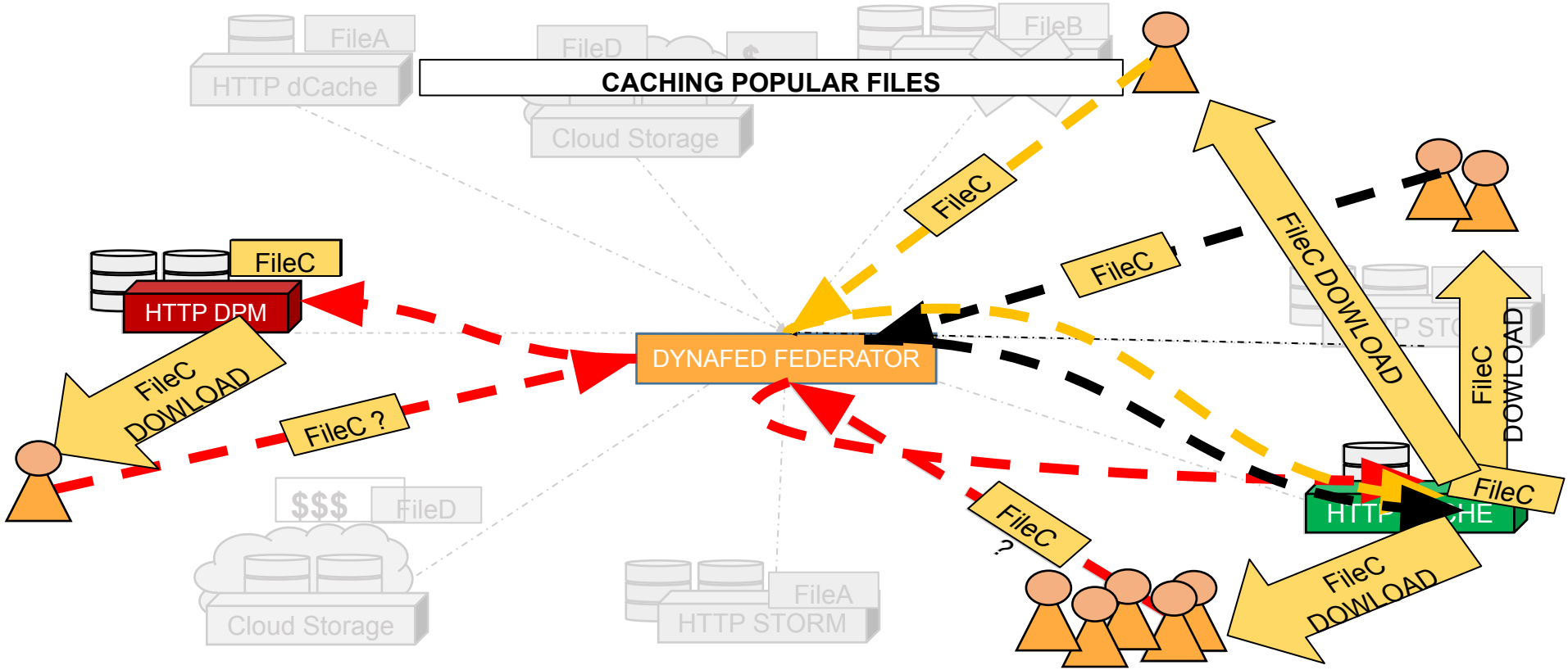

FileA
HTTP dCache
FileD
Cloud Storage
FileB
CACHING POPULAR FILES
FileC
HTTP DPM
DYNAFED
FEDERATOR
FileB
HTTP STORM
FileC
HTTP CACHE
FileC
?
$$$
FileD
Cloud Storage
FileA
HTTP STORM

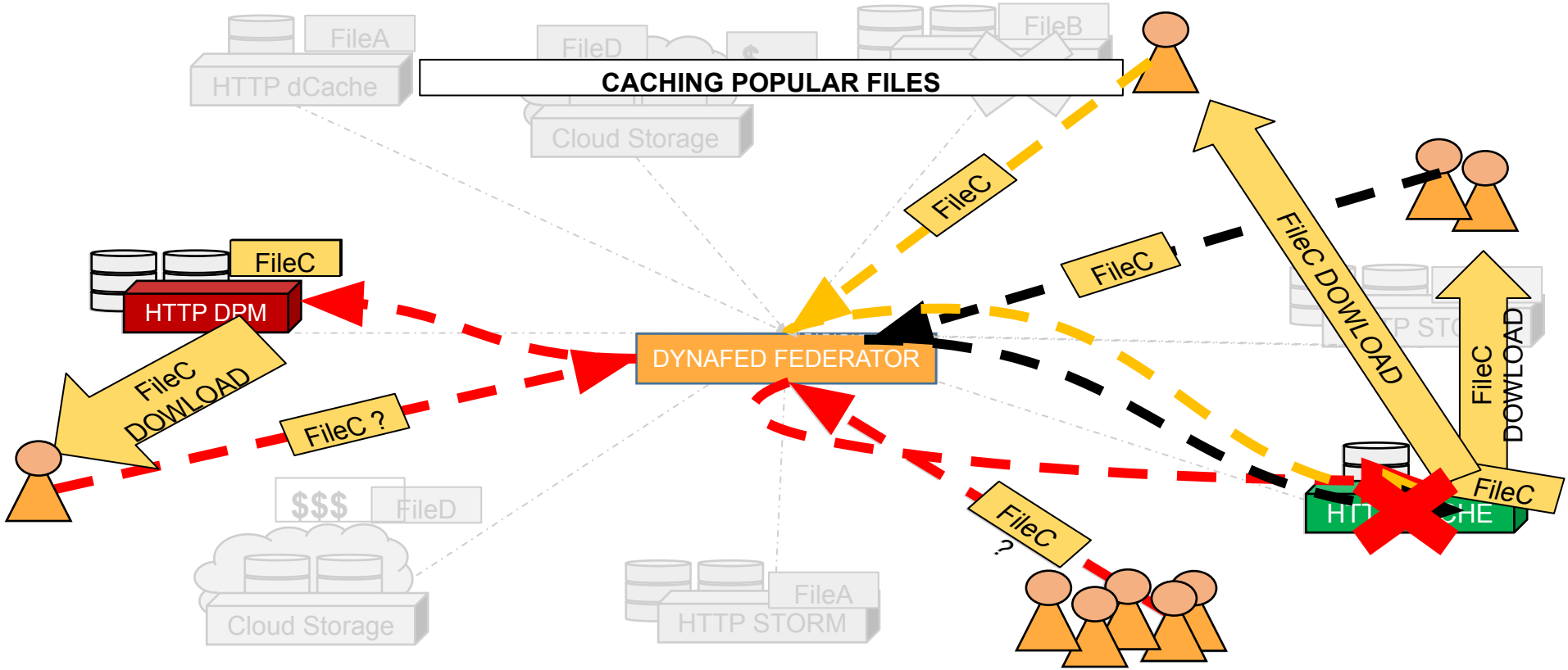

CACHING POPULAR FILES
FileA
HTTP dCache
FileD
Cloud Storage
FileB
FileC
HTTP DPM
DYNAFED FEDERATOR
FileB
HTTP STORM
$$$
FileD
Cloud Storage
FileA
HTTP STORM
FileC
?
HTTP CACHE
FileC
FileC
DOWLOAD

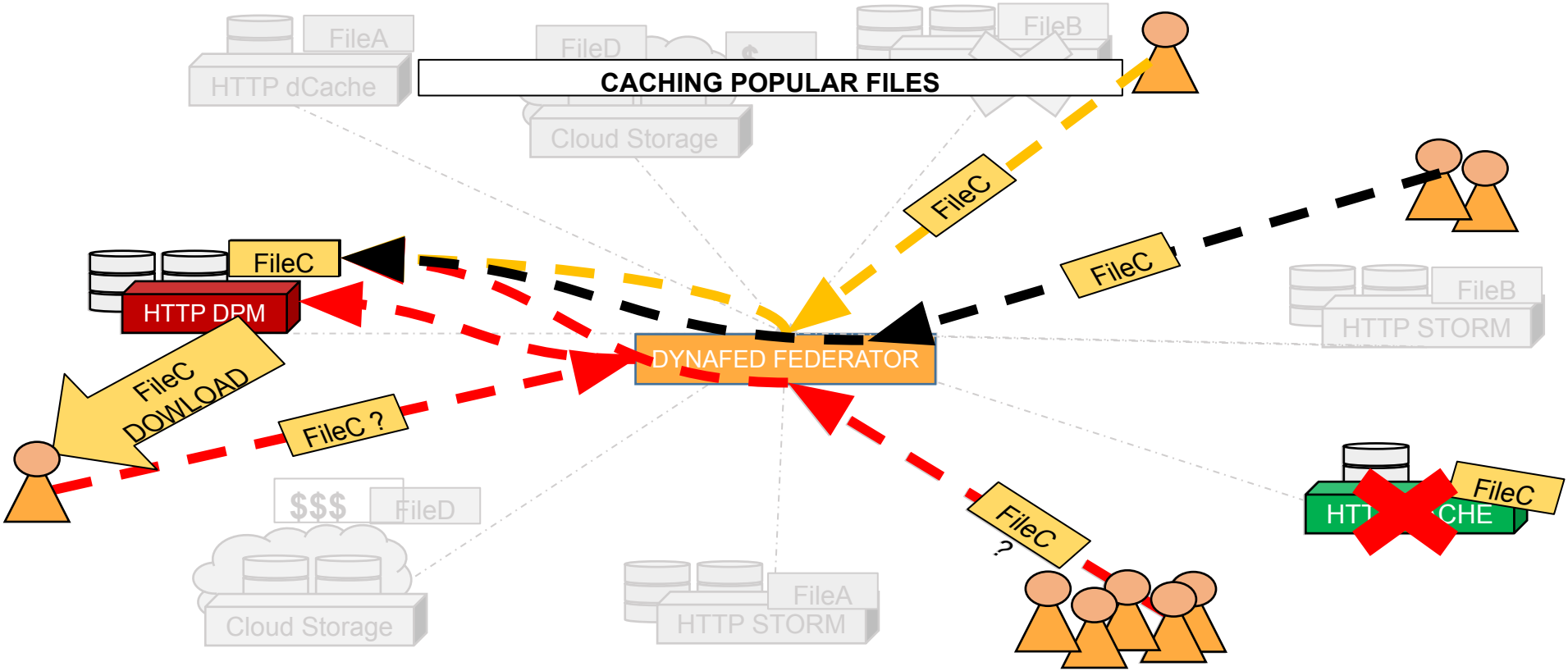

CACHING POPULAR FILES
FileA
HTTP dCache
FileD
Cloud Storage
FileB
FileC
HTTP DPM
DYNAFED FEDERATOR
FileB
HTTP STORM
$$$
FileD
Cloud Storage
FileA
HTTP STORM
FileC
HTTP CACHE
FileC
?
FileC
DOWNLOAD

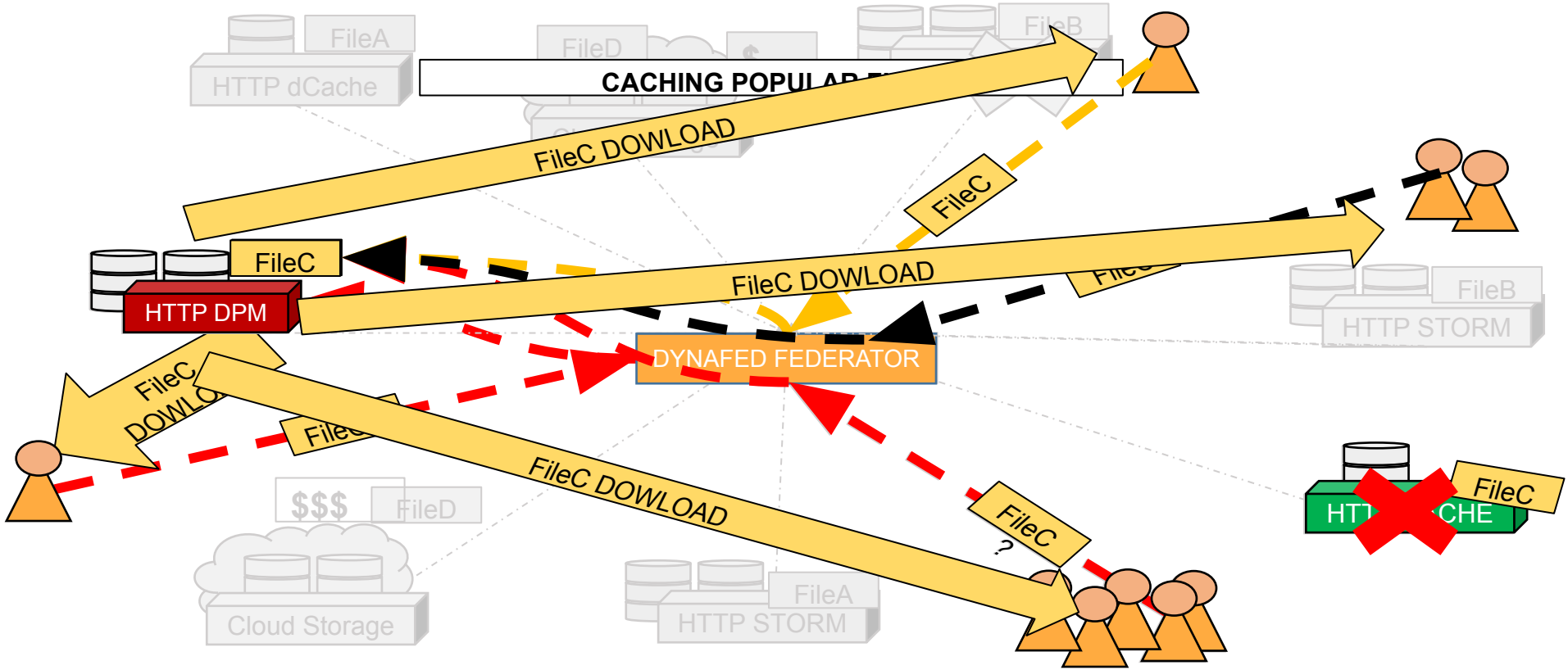

CACHING POPULAR FILES
FileA
HTTP dCache
FileD
Cloud Storage
FileB
FileC
HTTP DPM
FileC DOWLOAD
FileC ?
DYNAFED FEDERATOR
FileB
HTTP STORM
$$$
FileD
Cloud Storage
FileA
HTTP STORM
FileC ?
HTTP CACHE
FileC
FileC DOWLOAD


CACHING POPULAR FILES
FileA
HTTP dCache
FileD
Cloud Storage
FileB
FileC
FileC
FileC
HTTP DPM
FileB
HTTP STORM
DYNAFED FEDERATOR
FileC DOWLOAD
FileC ?
$$$
FileD
Cloud Storage
FileA
HTTP STORM
FileC
?
HTTP CACHE
FileC
FileC DOWLOAD


CACHING POPULAR FILES
FileA
HTTP dCache
FileD
Cloud Storage
FileB
FileC
FileC
FileC DOWLOAD
FileC DOWLOAD
FileC
HTTP DPM
DYNAFED FEDERATOR
FileC DOWLOAD
FileC ?
FileC
FileC
?
FileC DOWLOAD
$$$
FileD
Cloud Storage
FileA
HTTP STORM


CACHING POPULAR FILES
FileA
HTTP dCache
FileD
Cloud Storage
FileB
FileC
HTTP DPM
FileC
DOWLOAD
FileC ?
DYNAFED FEDERATOR
FileC
FileC
FileC DOWLOAD
FileC
DOWLOAD
FileC
?
$$$
FileD
Cloud Storage
FileA
HTTP STORM


CACHING POPULAR FILES
FileA
HTTP dCache
FileD
Cloud Storage
FileB
FileC
FileC
FileB
HTTP STORM
FileC
HTTP DPM
DYNAFED FEDERATOR
FileC DOWLOAD
FileC ?
$$$
FileD
Cloud Storage
FileA
HTTP STORM
FileC ?
FileC
CHE

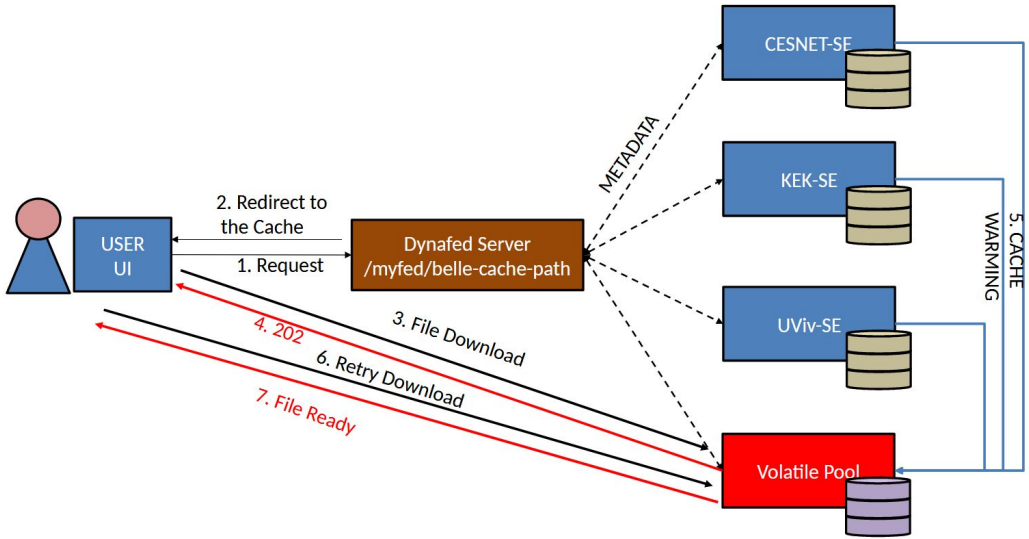

FileA
HTTP dCache
FileD
FileB
CACHING POPULAR
FileC DOWLOAD
FileC
FileC
HTTP DPM
FileC DOWLOAD
FileB
HTTP STORM
DYNAFED FEDERATOR
FileC
DOWLO
FileC DOWLOAD
FileC
?
$$$
FileD
Cloud Storage
FileA
HTTP STORM
FileC
CHE

USE CASE Belle II: testbed

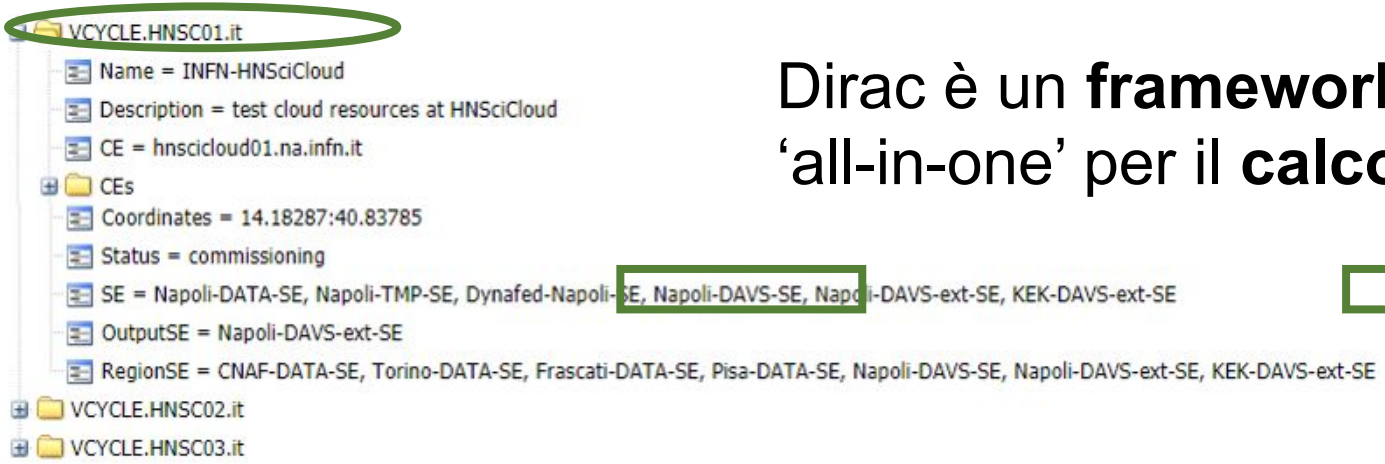
#	STORGE NAME	HOSTNAME	TYPE
1	DESY-DE	dcache-belle-webdav.desy.de	DCACHE
2	GRIDKA-SE	f01-075-140-e.gridka.de	DCACHE
3	NTU-SE	bgrid3.phys.ntu.edu.tw	DCACHE
4	SIGNET-SE	dcache.ijs.si	DCACHE
5	UVic-SE	charon01.westgrid.ca	DCACHE
6	BNL-SE	dcbldoor01.sdcc.bnl.gov	DCACHE
7	Adelaide-SE	coepp-dpm-01.ersa.edu.au	DPM
8	CESNET-SE	dpm1.egee.cesnet.cz	DPM
9	CYFRONNET-SE	dpm.cyf-kr.edu.pl	DPM
10	Frascati-SE	atlasse.lnf.infn.it	DPM
11	HEPHY-SE	hephyse.oeaw.ac.at	DPM
12	Melbourne-SE	b2se.mel.coepp.org.au	DPM
13	Napoli-SE	belle-dpm-01.na.infn.it	DPM
14	ULAKBIM-SE	torik1.ulakbim.gov.tr	DPM
15	IPHC-SE	sbgse1.in2p3.fr	DPM
16	CNAF-SE	ds-202-11-01.cr.cnaf.infn.it	STORM
17	ROMA3-SE	storm-01.roma3.infn.it	STORM
18	KEK-SE	Kek-se03.cc.kek.jp	STORM
19	McGill-SE	gridftp02.clumeq.mcgill.ca	STORM

- DynaFED aggregates almost all **production storage**
- Heterogeneous tecnologies
- Storages **distributed** all over the world

USE CASE Belle II

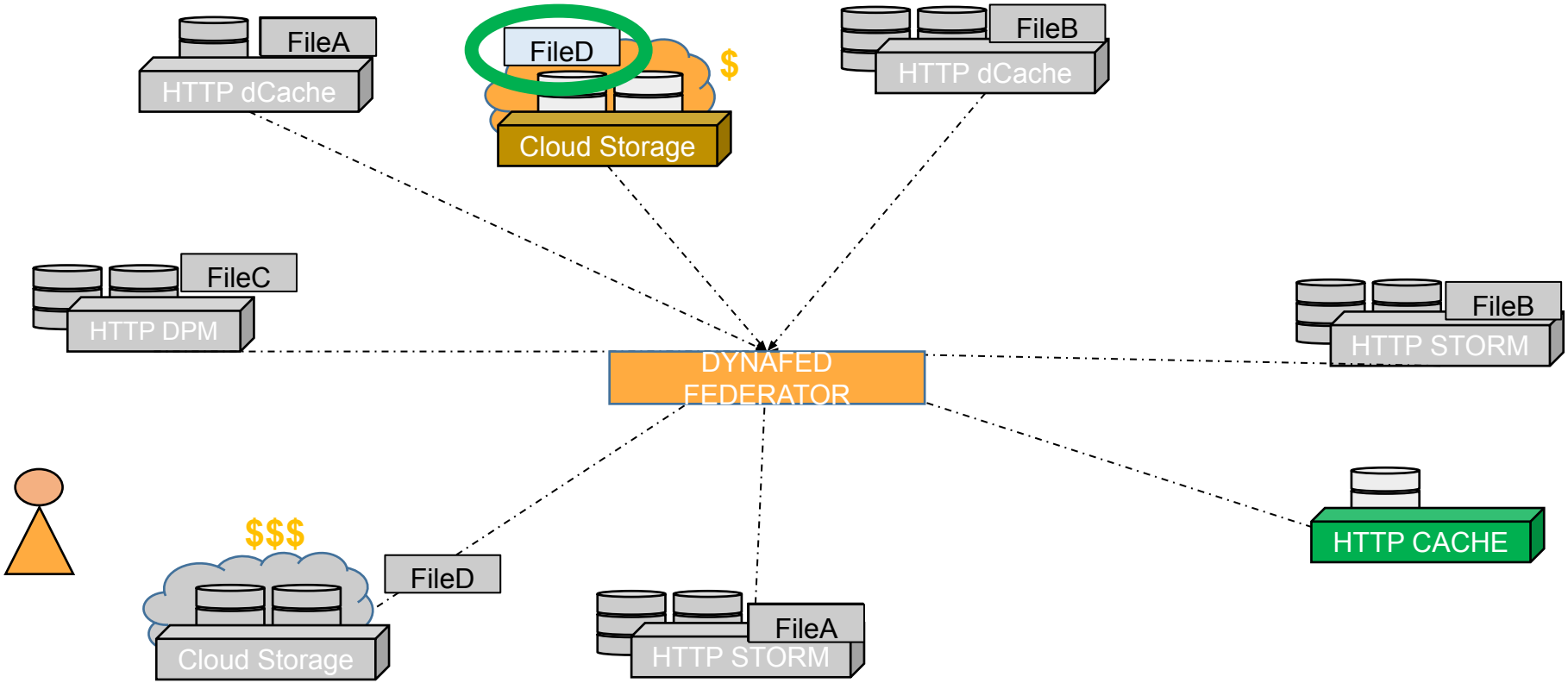


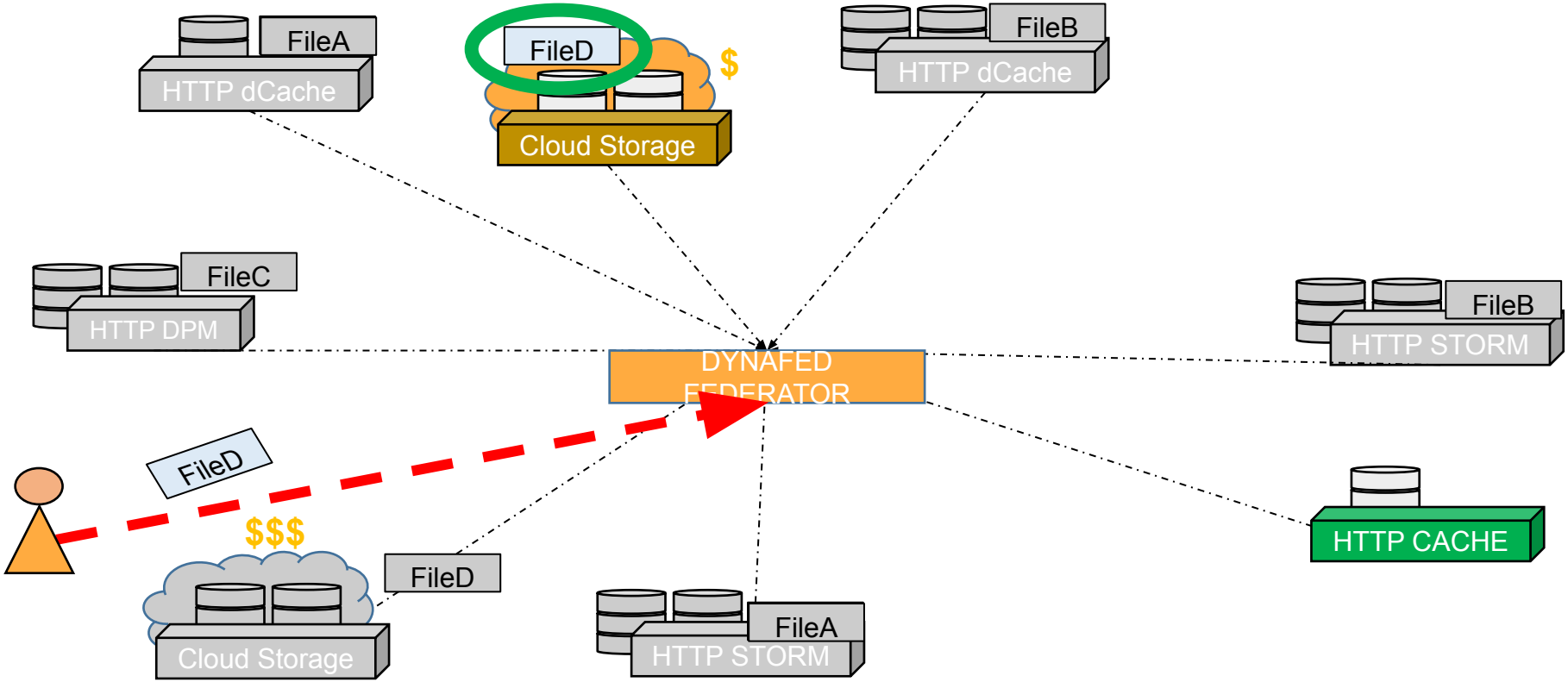
USE CASE Belle II: DIRAC

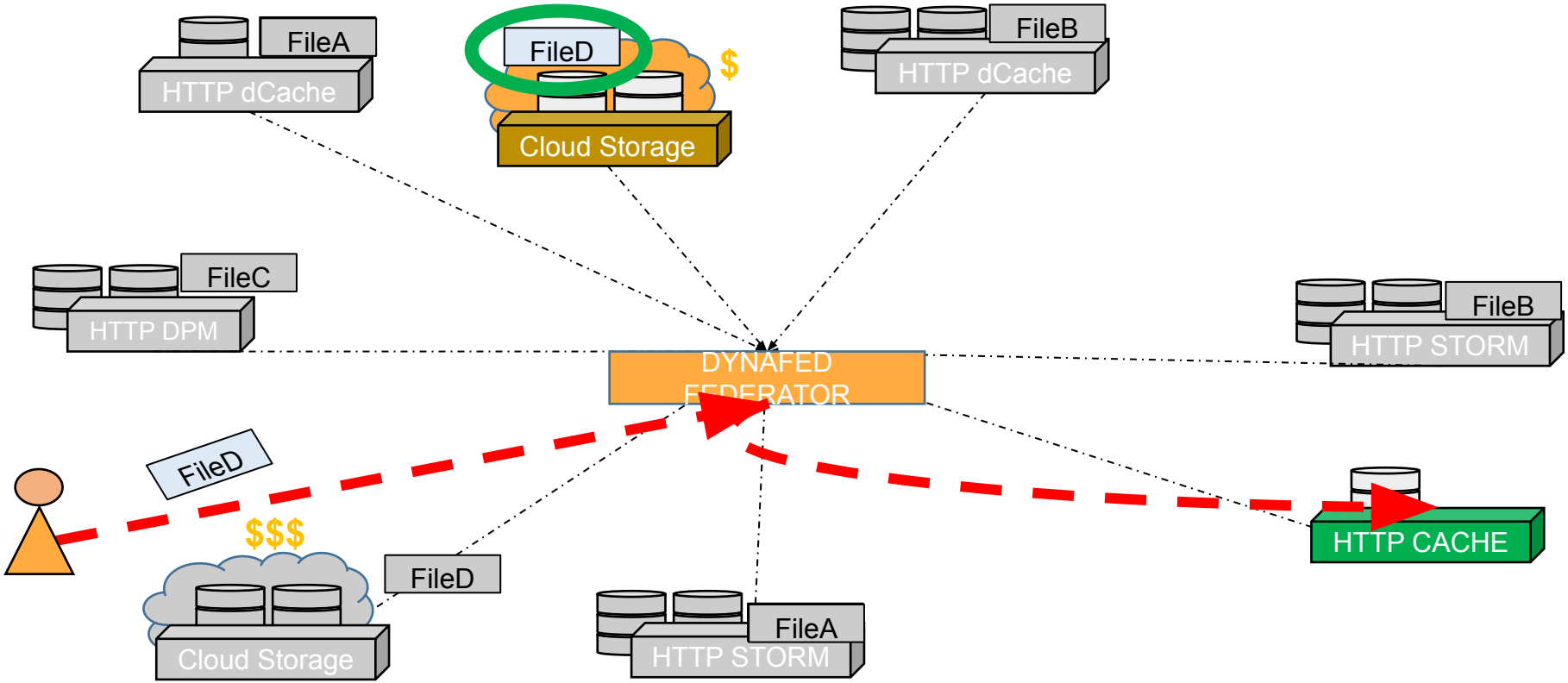


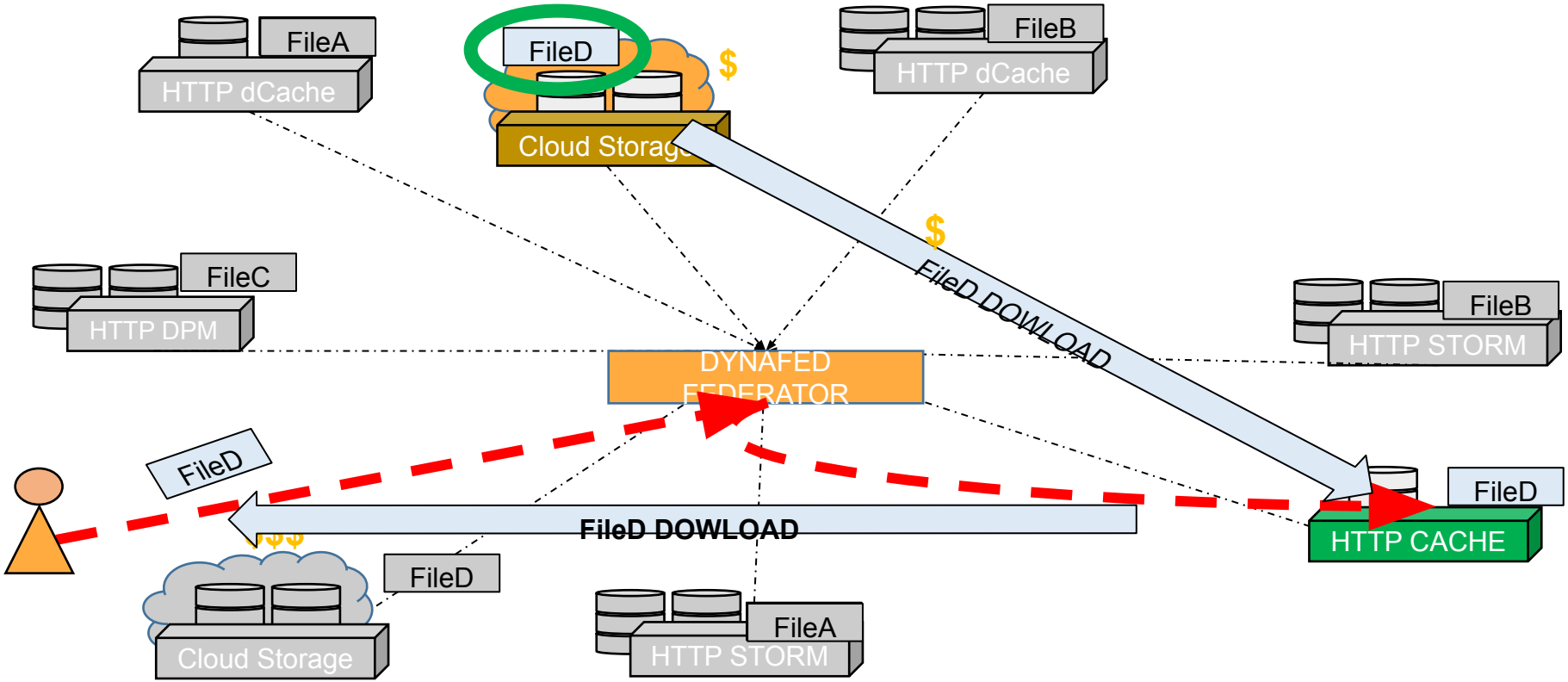
Dirac è un **framework** per la gestione ‘all-in-one’ per il **calcolo distributo**.

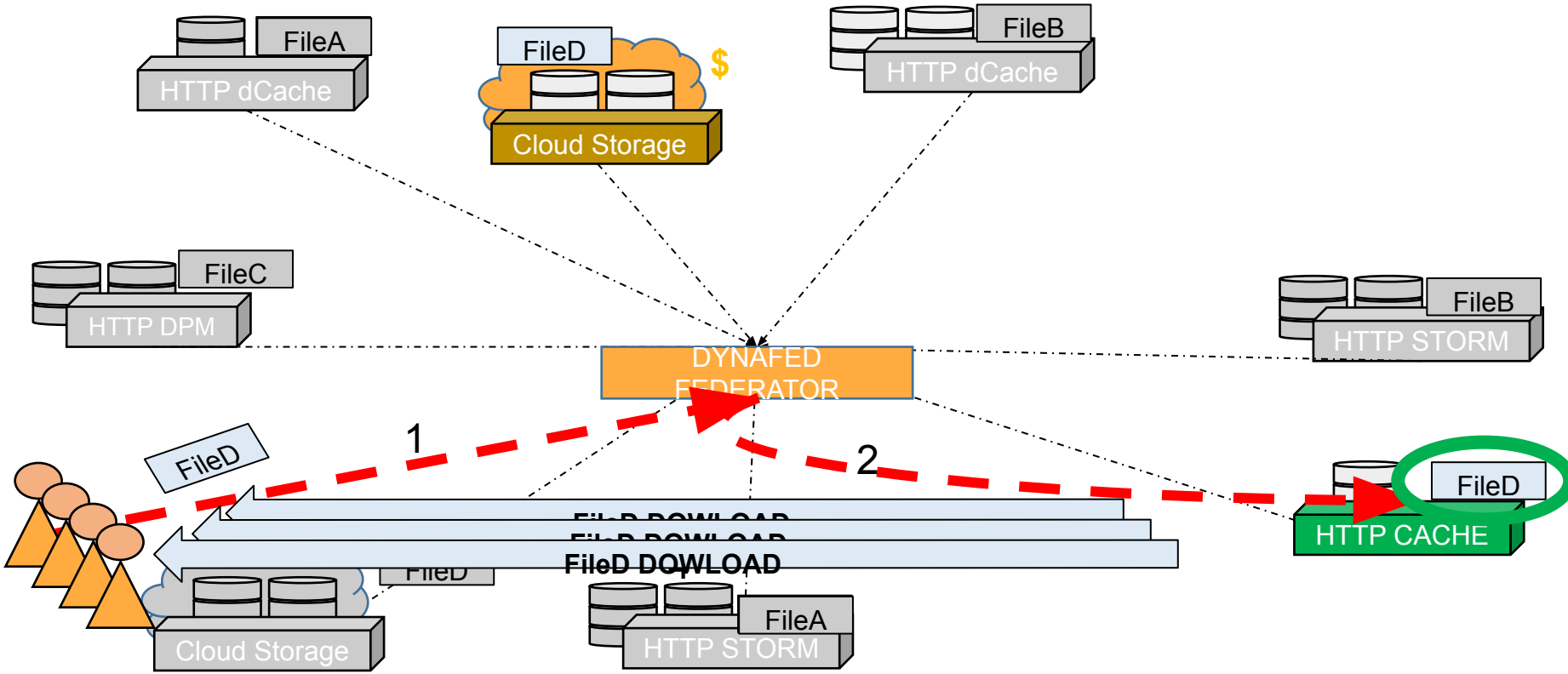
- VCYCLE.HNSC01.it site in DIRAC is configured to use **HTTP endpoints**, including **DynaFed**.


FileA
HTTP dCache
FileD
$
Cloud Storage
FileB
HTTP dCache
FileC
HTTP DPM
DYNAFED
FEDERATOR
FileB
HTTP STORM
HTTP CACHE
$$$
FileD
Cloud Storage
FileA
HTTP STORM


FileA
HTTP dCache
FileD
$
Cloud Storage
FileB
HTTP dCache
FileC
HTTP DPM
DYNAFED
FEDERATOR
FileB
HTTP STORM
FileD
$$$
FileD
Cloud Storage
FileA
HTTP STORM
HTTP CACHE


FileA
HTTP dCache
FileD
$
Cloud Storage
FileB
HTTP dCache
FileC
HTTP DPM
DYNAFED
FEDERATOR
FileB
HTTP STORM
FileD
$$$
FileD
Cloud Storage
FileA
HTTP STORM
HTTP CACHE


FileA
HTTP dCache
FileD
Cloud Storage
$
FileB
HTTP dCache
FileC
HTTP DPM
$
FileD DOWLOAD
FileB
HTTP STORM
DYNAFED
FEDERATOR
FileD
FileD DOWLOAD
FileD
HTTP CACHE
FileD
Cloud Storage
FileA
HTTP STORM


FileA
HTTP dCache
FileD
Cloud Storage
$
FileB
HTTP dCache
FileC
HTTP DPM
FileB
HTTP STORM
DYNAFED
FEDERATOR
1
2
FileD
FileD DOWLOAD
FileD
HTTP CACHE
Cloud Storage
FileA
HTTP STORM