An HTTP federation prototype for LHCb

Fabrizio Furano

Introduction

- In September we started setting up an HTTP fed for LHCb
  - Stefan Roiser
  - Fabrizio Furano
- An explorative project
- Very good results in a short time
- We present here the challenges, the results and the status of the prototype

HTTP/WEBDAV federation

- The HTTP/WebDAV LHCb prototype fed for an user appears as just a huge, distributed repository with a friendly feel
  - is accessible from a browser or with a decent HTTP client (curl, wget, davix, …)
  - works quickly and reliably, does not rely on static catalogues
  - takes realtime redirection choices, considering the worldwide status (instead of a static catalogue)
  - never out of sync with the storage elements' content
  - can scale up the size of the repo
  - can scale up the number of clients
- A huge data repository accessible with a browser, fast and always exact
- Exact means "taking into account the status of the endpoints in that moment"
  - It means that the endpoints that are down are not shown

Dynamic Federations

- A project started a few years ago
- Goal: a frontend that presents what a certain number of endpoints would present together
  - Without indexing them beforehand
- These endpoints can be a very broad range of objects that act as data or metadata stores
  - We prefer to use HTTP/WebDAV things, yet that's not a constraint

This is
What we want
to see as users

Sites remain independent and participate to a global view

All the metadata interactions are hidden and done on the fly

NO metadata Persistency needed here, just efficiency and parallelism

Aggregation

```
/dir1
/dir1/file1
/dir1/file2
/dir1/file3
```

With 2 replicas

Europe

Storage/MD endpoint 1

```
.../dir1/file1
.../dir1/file2
```

Storage/MD endpoint 2

```
.../dir1/file2
.../dir1/file3
```

Middle East & Africa

Dynamic Federations

- Opens to a multitude of use cases, by composing a worldwide system from macro building blocks speaking HTTP and/or WebDAV
  - Federate natively all the LHCb storage elements
  - Add third party outsourced HTTP/DAV servers
  - Add the content of fast changing things, like file caches
  - Add native S3 storage backends (a supported dialect)
  - Accommodate whatever metadata sources, bare SEs or catalogues if needed. Even two or more remote catalogues at the same time

- Clients are redirected to the replica closer to them
- Redirect only to working endpoints
- Accommodate whatever other Cloud-like storage endpoint

Why HTTP/DAV?

- It's there, whatever platform we consider
  - A very widely adopted technology

- We (humans) like browsers, they give an experience of simplicity
- Mainstream and sophisticated clients: curl, wget, Davix, …
- ROOT works out of the box with HTTP access (LCG release >= 69)

- Goes towards convergence
  - Users can use their devices to access their data easily, out of the box
  - Web applications development can meet Grid computing
  - Jobs and users just access data directly, in the same way
  - Can more easily be connected to commercial systems and apps

LHCb replica management

- The first action was analyzing the directory trees of a few LHCb SEs
- They look the same everywhere, modulo a string prefix depending on the site
- This is the simplest case that the Dynafeds can handle. My appreciation to whoever made this choice and kept it so clean.
- No catalogues or special things are needed for federating with this kind of schema
- Example:

/lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00030613/0000/00030613_00000134_1.bhadroncompleteevent.dst

remains constant, despite the prefix it may have, like:

https://ccdavlhcb.in2p3.fr:2880/
or
https://fly1.grid.sara.nl:2882/pnfs/grid.sara.nl/data/

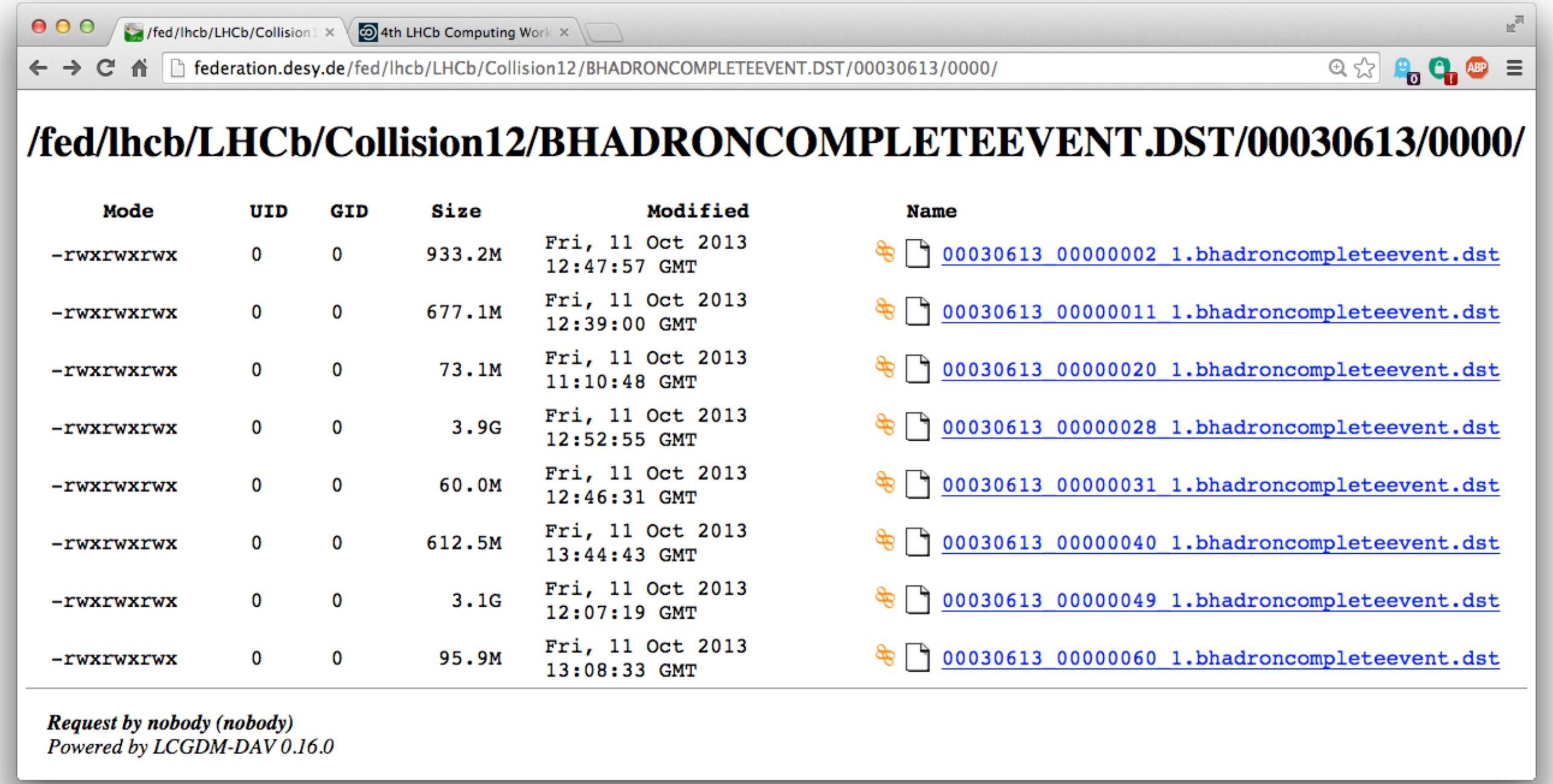
Look and feel

- What we see in the browser is an HTML rendering of a listing. This is the content belonging to LHCb worldwide (modulo some SEs, see later)

- Everything is done on the fly
- Click on a file to download it (if your client is authorized by the endpoint SE through X509)
- Feed the URL of that file to any other client to download it
- Feed the URL of that file to any job

- Click on the strange icon to get a metalink
  - A standard representation of the locations of a file **sorted by increasing distance from the requestor**
    - (Plugin-based, any other metric is possible)
  - It's supported by multi-source download apps

Look and feel, like a normal list



Metalink example

```
<metalink xmlns="http://www.metalinker.org/" xmlns:lcgdm="LCGDM:" version="3.0" gene
<files>
<file name="/lhcb/L">
<size>4189611249</size>
<resources>
<url type="https">
https://ccdavlhcb.in2p3.fr:2880/lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/
    00030613/0000/00030613_00000132_1.bhadroncompleteevent.dst
</url>
<url type="https">
https://fly1.grid.sara.nl:2882/pnfs/grid.sara.nl/data/lhcb/LHCb/
    Collision12/BHADRONCOMPLETEEVENT.DST/00030613/0000/
    00030613_00000132_1.bhadroncompleteevent.dst
</url>
<url type="https">
https://wasp1.grid.sara.nl:2882/pnfs/grid.sara.nl/data/lhcb/LHCb/
    Collision12/BHADRONCOMPLETEEVENT.DST/00030613/0000/
    00030613_00000132_1.bhadroncompleteevent.dst
</url>
</resources>
</file>
</files>
</metalink>
```

LHCb HTTP SE harvesting

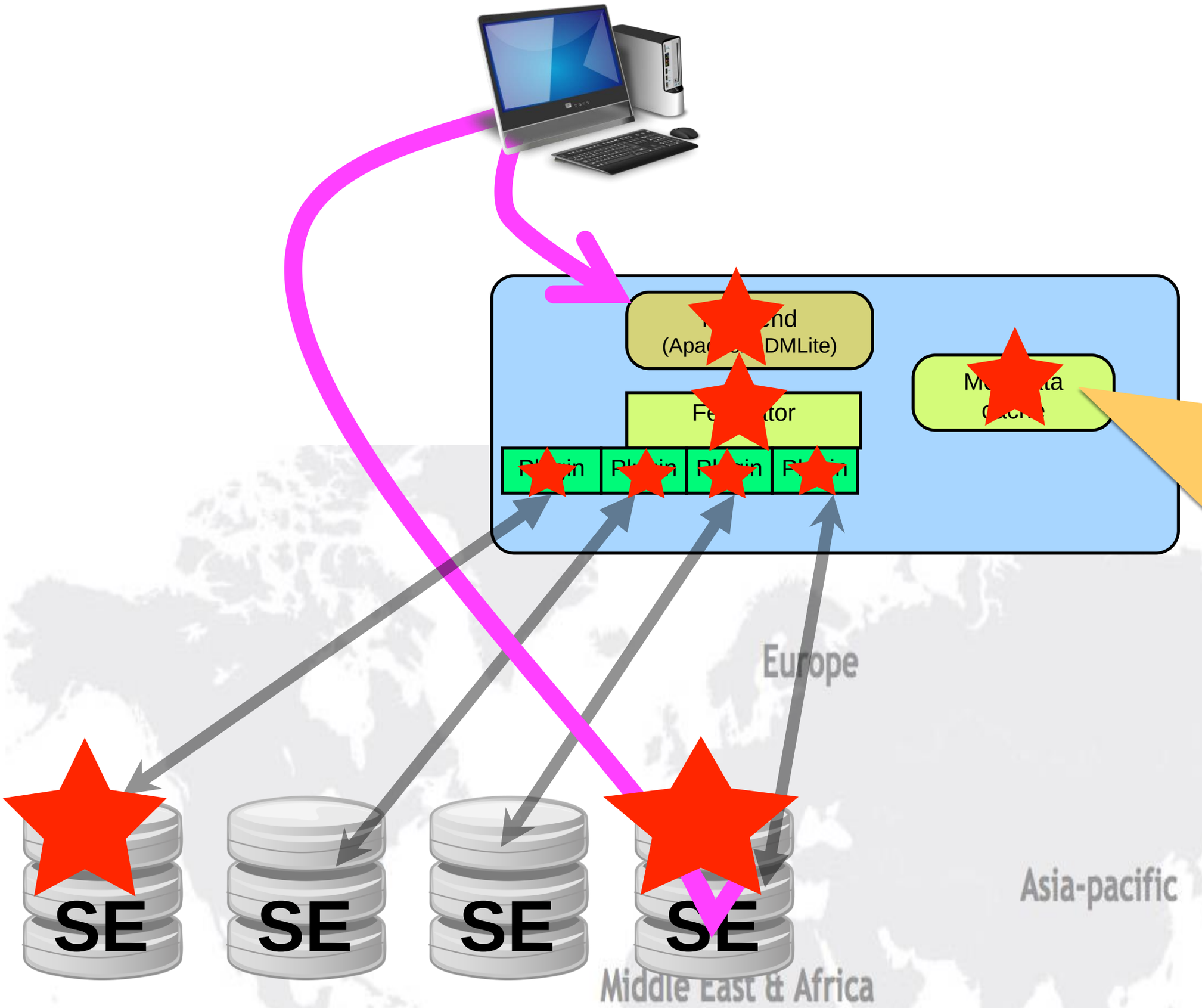
- This step was performed by Stefan Roiser
- Looking at BDII and SRM TURLs to harvest the LHCb SEs that had a working HTTP access
  - Enough for setting up the first little prototype in the machine of our DESY cooperators
  - http://federation.desy.de/fed/lhcb/
- The federator only needs metadata READ access
  - Normally fulfilled, unobtrusive for sites
- Then Stefan wrote to everyone and we started keeping track of them

Status

- 13 sites out of 19 just work
  - All those correctly publishing WebDAV for LHCb with X509

- Missing:
  - EOS@CERN
    - Contacted and exchanged information.
  - CASTOR@CERN
    - Contacted. Will join in Spring ‘15
  - STORM@CNAF
    - CNAF working on a solution for deploying WebDAV for LHCb
  - PIC
  - CASTOR@RAL
    - Some progress, configuration to fix
  - RAL-HEP (dCache)

The Tech corner



The cache remembers what happened

The next **metadata** interactions will very likely be fed by the cache

The 2nd level cache can be shared among federators (memcached)

Dynafeds and metadata catalogues

- A fed and a catalogue fulfil different use cases
- A fed is dynamic: interacts with what's available in that moment
  - Sites up/down, disappeared files, distance of alive sites from the client, …
- A catalogue is static: it tells us what's supposed to be there (data losses… dark data…)

- Static/dynamic examples:
  - checking which site is supposed to have something needs a catalogue
  - selecting datasets for a run needs a (sophisticated) metadata catalogue
  - selecting files for a job will be more resilient with a fed providing fresh metalinks
  - running a job at a site will be more resilient with a fed providing fresh metalinks
  - downloading a file will be more resilient with a fed, and easier to do

Q&A

- Can LFC be put into the fed ?
- Yes, catalogues can be mounted, they would act as:
  - Static listing providers
  - Static providers of replica TURLs for namespaces that are not algorithmic (luckily not the LHCb case)
  - Replicas will still be checked in realtime
  - Dynafeds can translate SRM TURLs into HTTP (sophisticated config)
  - The reliability of the fed will be linked to the reliability of the catalogue

- My opinion…
  - So far, the LHCb federation does not need this, as everything is so clean without it
  - makes sense only if we just want to have an HTTP/DAV frontend to the catalogue itself… or a federation of (partial ?) catalogues

- Can one do a full namespace scan to look for dark data ?
  - Yes. Keep present that all the endpoints will see a namespace scan

What about xrootd ?

- Seems that LHCb is transitioning to using the Xrootd protocol for data access.
- We see all the advantages of the *direct data access* approach supported by HTTP and Xrootd in all the Grid SE techs.

- Many good reasons to grow an HTTP ecosystem that can happily coexist with a preexisting xrootd one
- A door open towards user-friendly, industry standard interfaces
- A decise step towards opportunistic resource exploitation. We could federate an S3 backend today, together with the LHCb data.
  - In fact we already did in the /lhcb parent directory…

- Native Xrootd4 sites can join it too, as Xrootd4 natively supports HTTP/WebDAV (tested with feds too)

Conclusions

- A r/o R&D prototype that exceeded expectations
  - 13 sites out of 19, the others are coming
  - Official site downtimes were always automatically detected so far

- Cleanness of LHCb repos helped
- Please evaluate it and help us improve
  - This is likely to be an actor of a next evolution in *large scale DM,* HEP meeting the Web through proper tools
- New features are coming. Smarter site detection, write support, logging, monitoring, …
  - Next RC of the Dynafeds expected in a couple of weeks
- High flexibility/scalability of the concept, able to deal with a broad range of endpoints
- Can be made to work with WebFTS to find the “right” sources
  - Also endpoint prioritization is pluggable

- Looking at exploiting the potential of mixing S3 storage with other techs
  - We are contributing to a r/w prototype for BOINC (See preGDB by Laurence Field)
- We are cooking an AGIS-based RUCIO-friendly prototype (>40 sites, >200 spacetokens)