DPM status and directions 2018

A tech outlook

Introduction

- Another quite dense year and a half has passed
- Two releases, 1.9.x (maintenance) and 1.10.2 (feature)

- 1.10.2 is the latest stop in DPM's architectural evolution
  - DOME got mature and totally independent
  - The DMLite config is at the maximum level of simplicity possible
  - No architectural changes foreseen
  - Enhancements (e.g. better shell commands) become easier

DPM status

- Infosys numbers
- ~90PBs in total, provided to Grid computing. ~130 instances.
- DPM lost a tail of tiny sites, while the overall storage capacity continued to grow (was ~70PB at the last appointment)

- Several sites larger than 2 PB, 20 sites larger than 1PB. The largest so far is 6.5PB

- Our focus continues to be on
  - Consolidation, keeping sysadmin cost at the lowest
  - Performance, scalability
  - High quality HTTP, WebDAV, Xrootd, GridFTP support
  - Support. In touch with sysadmins as much as we can
  - Thank you very much for helping on dpm-users-forum !

Themes

- SRM activity… will it decrease?
- Space reporting on directories and tokens
- DPM overall manages more storage
- Less and less tiny TB-size sites in the infosys

- Our focus on:
  - support, debug, problem solving
  - supporting the sites that upgrade
  - user-friendliness of the admin tools
  - latest trends (Macaroons, SciTokens, 3rd party copy enhancements)

- We discuss the current DPM status
- We present the roadmap for the transitions ahead

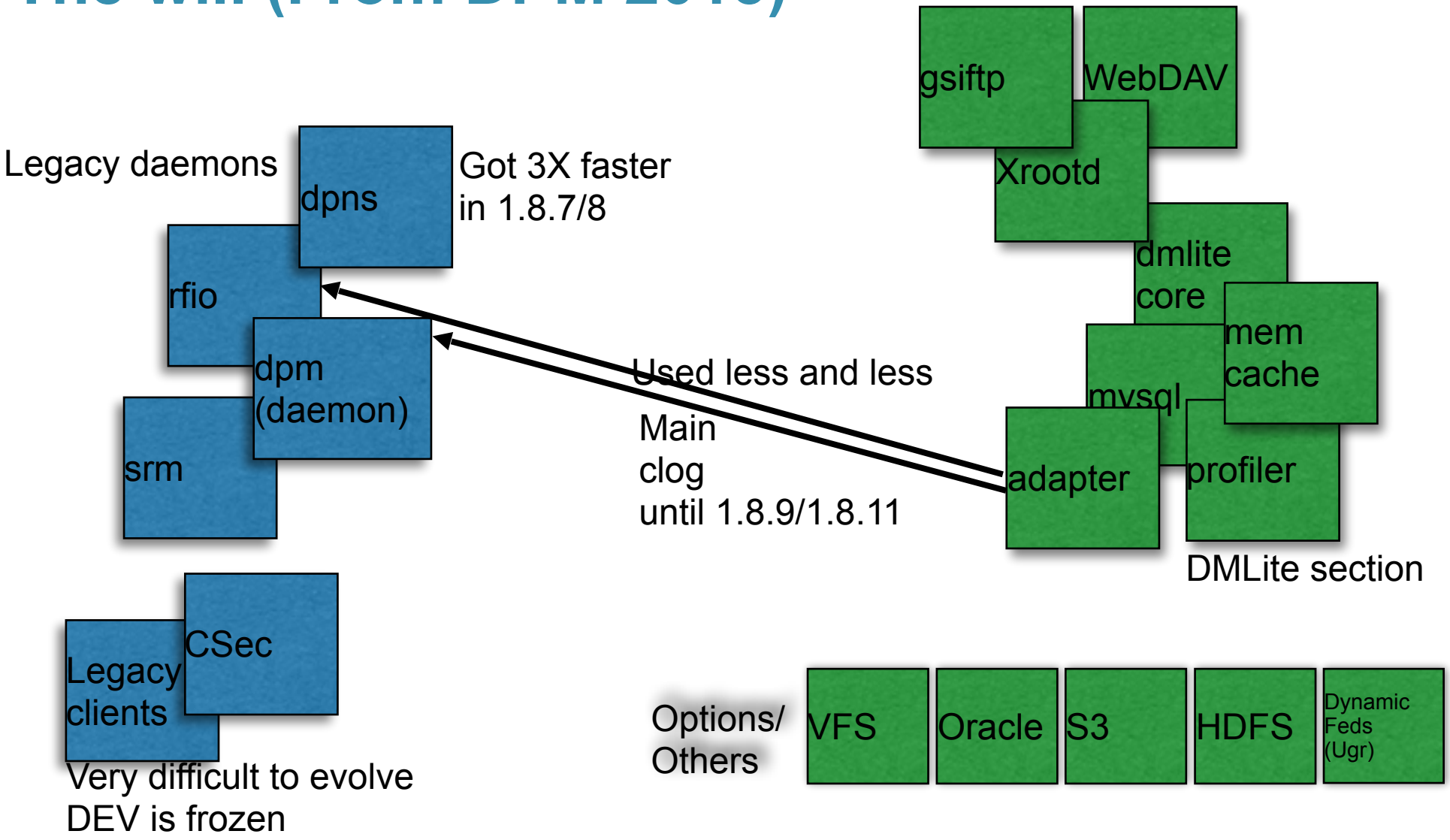
Main direction (1/2)

- Manageability and long term support of the DPM system
- Can only be achieved with a clean, straightforward system based on open, contemporary technologies
- The DPM core (DOME: Disk Operations Management Engine) and its companion components have been built for that
- The last release was under constant, massive test since April 2017

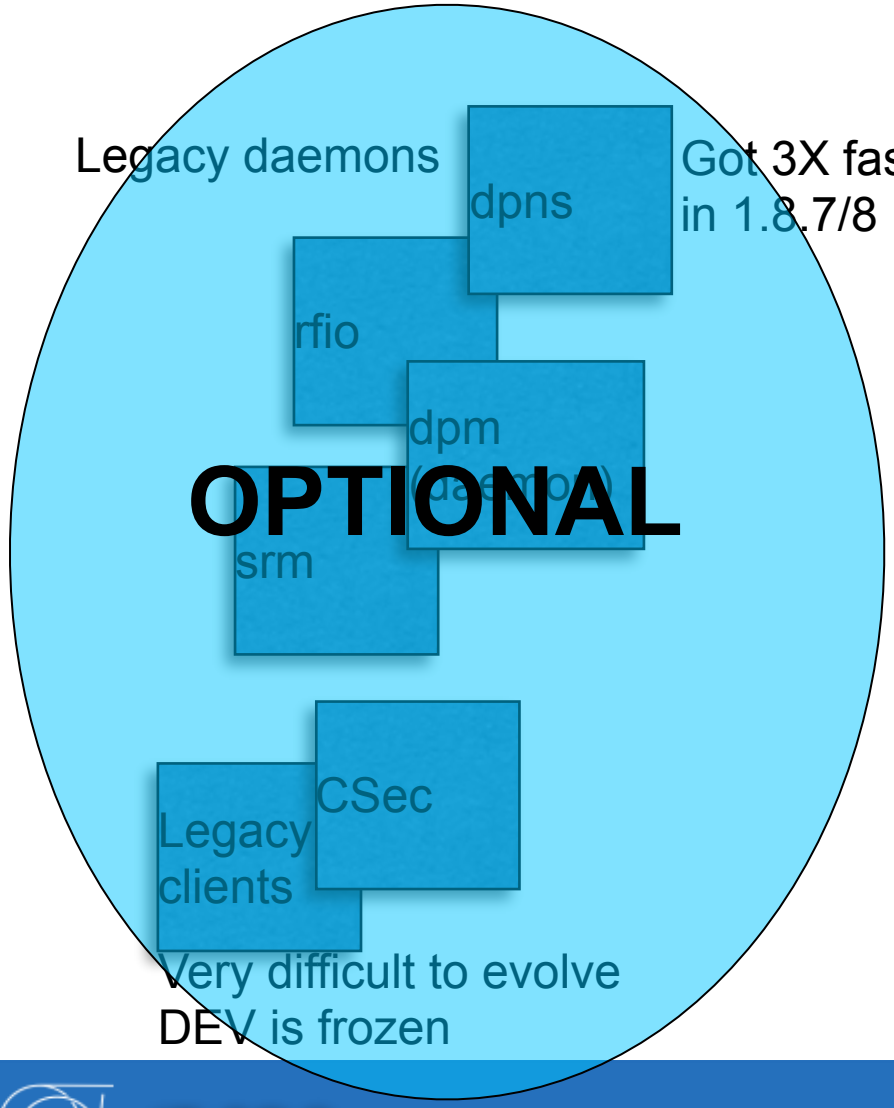
Main direction (2/2)

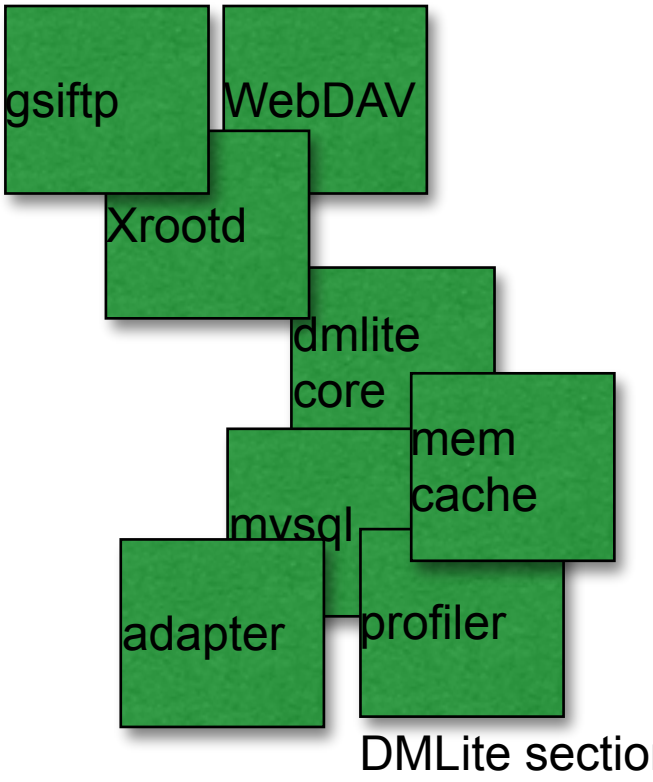
- The best way to reduce the effort on DPM is making it work well
- Find ways to simplify the system keeping the functionalities
- Find ways to simplify the configuration
- Example: The resource consumption to respond to an FTS campaign has been reduced by more than 10 times in the last release in non-legacy mode
  - The related failure modes (failure rate) too, as a consequence

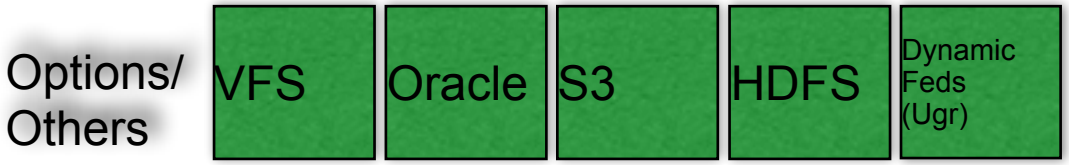
The will (From DPM 2015)



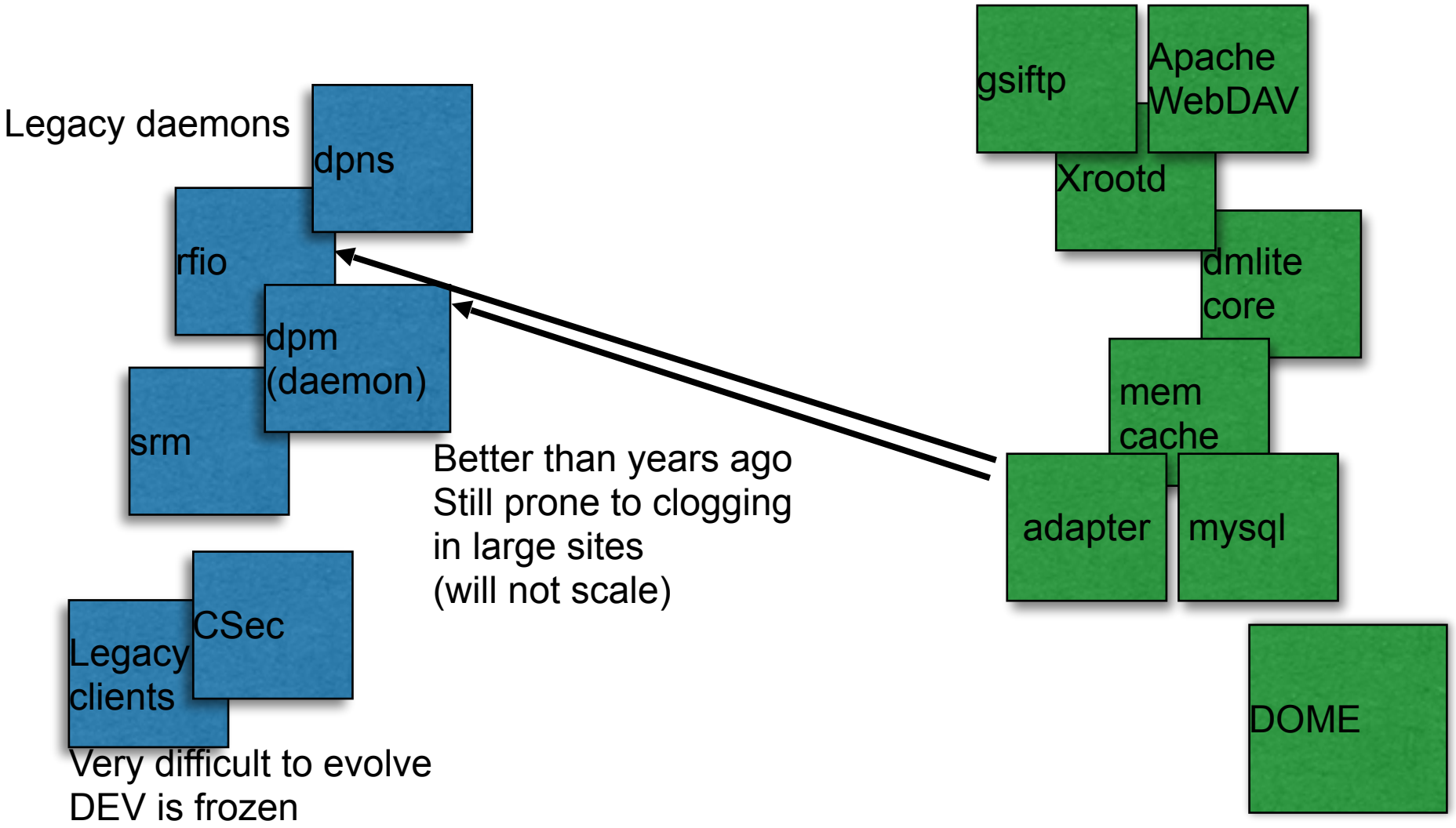
The will (From DPM 2015)



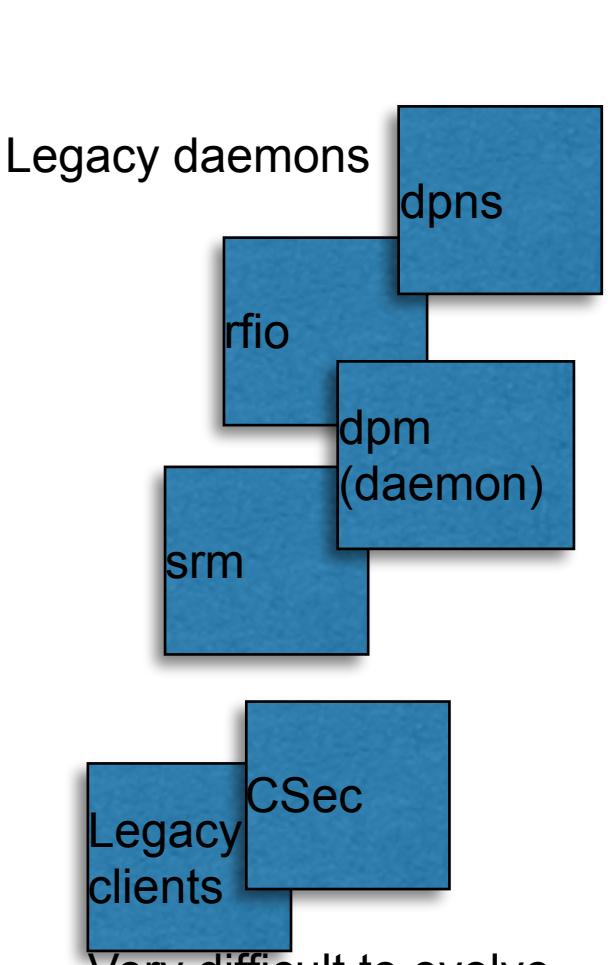


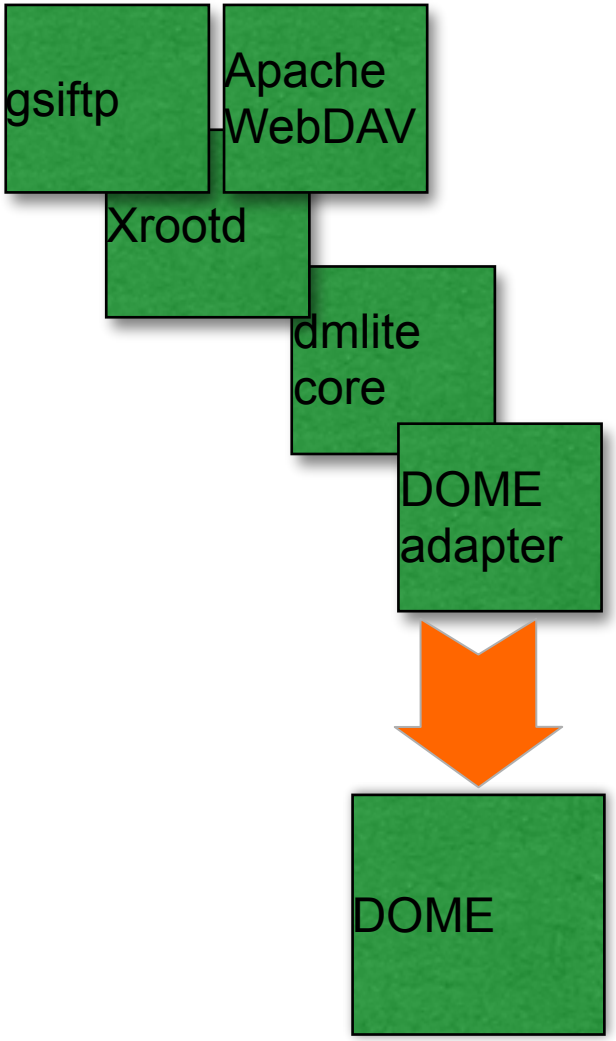


DPM components and plugins (2017/2018)

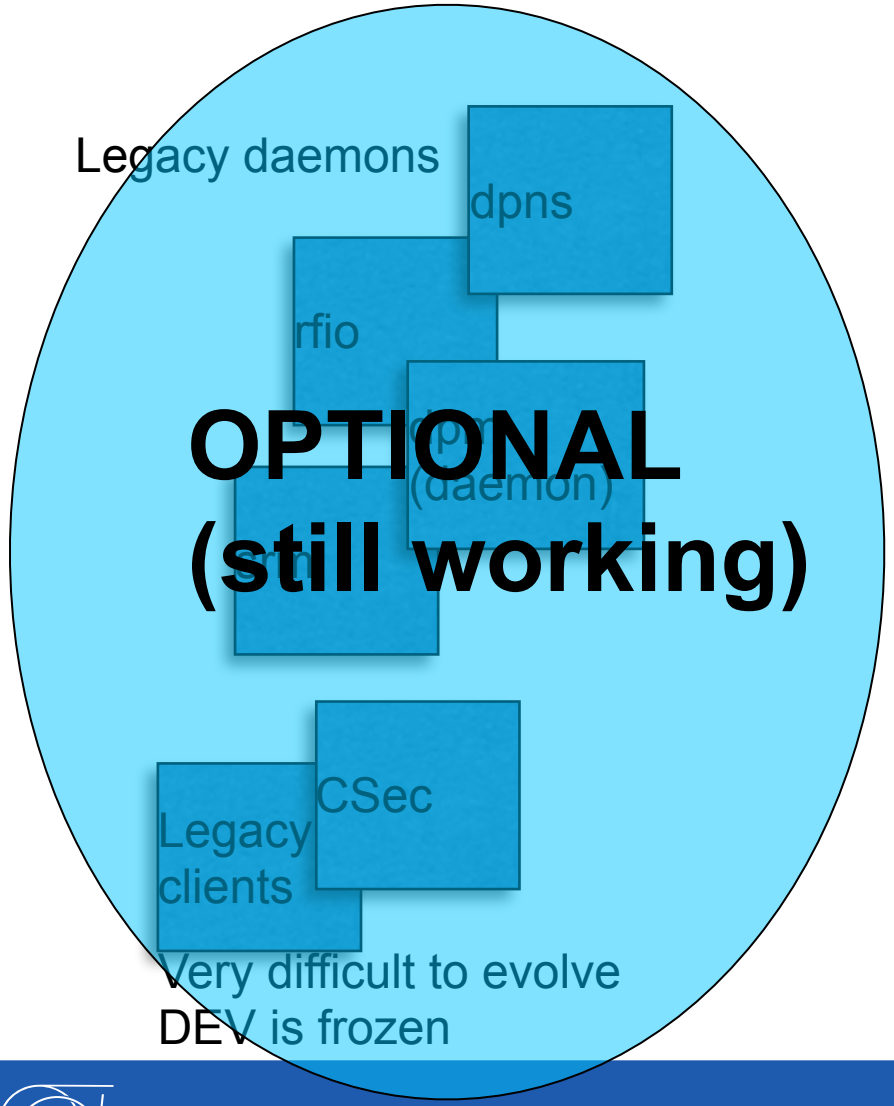


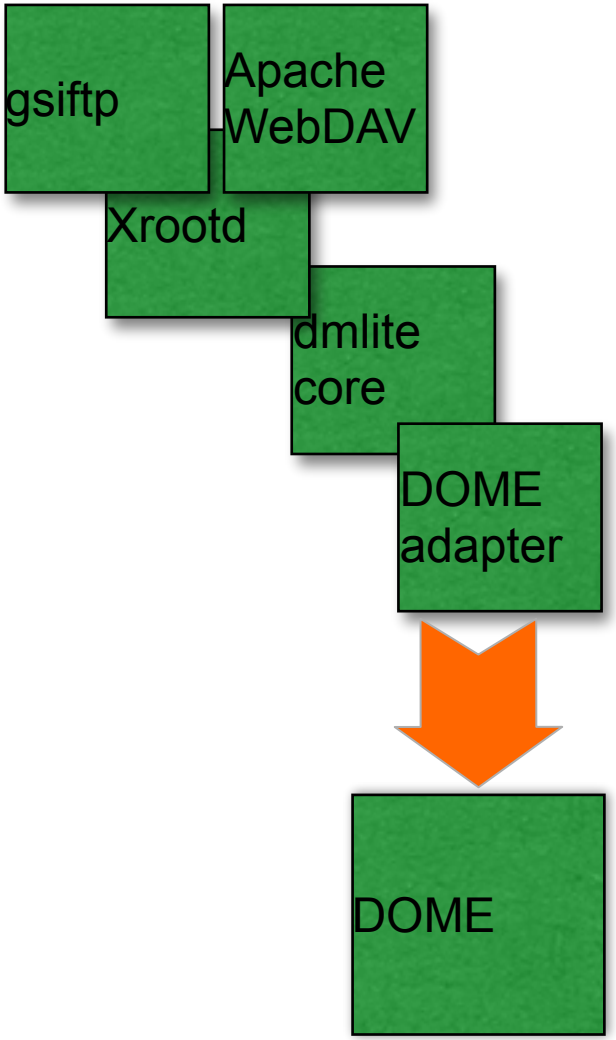
DPM components and plugins (2017/2018)





DPM components and plugins (2017/2018)





Legacy and non-legacy (DOME) mode

- Legacy mode is when DMLite loads the Adapter+Memcache+MySQL plugins
  - The good old DPM daemon does the coordination work
  - Every process loading dmlite needs a new pool of MySQL connections

- Non-legacy mode is when DMLite loads DOMEAdapter
  - DOME does the coordination work and talks to mysqld
  - The DPM daemon coordinates itself and SRM
  - Only one internal MySQL pool is used, ever

One plugin to rule them all

- **In non-legacy mode** DMLite now loads only DOMEAdapter
- dmlite-memcache, dmlite-mysql are no longer necessary
- Resource consumption (FDs, mysql, etc.) is reduced by an order of magnitude, and so complexity and cost for us all

Be prepared

- Our wish is to deprecate the legacy mode
- Our goal is to reduce support for components that can't cope with the increasing requirements of WLCG computing
- LCGDM received good enhancements 2-3 years ago, and it can survive some time
  - it won't scale in larger sites
  - Its performance may not scale
    - (= more client failures)

DOME speaks REST and JSON

- DOME and its companion DOMEAdapter give the functionalities of dpm+dpns+rfio

```
GET /domehead/command/dome_getstatinfo
    HTTP/1.1
User-Agent: libdavix/0.6.8 neon/0.0.29
Keep-Alive:
Connection: Keep-Alive
TE: trailers
Host: dpmhead-trunk.cern.ch:1094
Content-Length: 17

> Body block (17 bytes):
{ "lfn": "/dpm" }
```

```
HTTP/1.1 200 OK
Content-Length: 250
{ "fileid": “3",
  "parentfileid": “2",
  "size": “265623786530",
  "mode": “16877",
  "atime": “1523455123",
  "mtime": “1522229608",
  “ctime": “1522229608",
  "uid": “0",
  "gid": “0",
  "nlink": “2",
  "acl": “A70,C50,F50,a70,c70,f50",
  "name": “dpm",
  "xattrs": "{\"type\": 0}”
}
```

DOME in 2018

- DOME and DOMEAdapter now have all the primitives
  - They support all the historical functionalities of dpm+dpns+rfio
  - Many tiny limitations have been removed, the behaviour is very linear and has much less code

- DOME got an internal write-through metadata cache (taken from Dynafed)
  - We will not mention it very much… its config is minimalistic
  - It's a write-through cache. Operations do not invalidate its content. Friendly with threads, efficient and simple also in complex cases.
  - Eliminates the strange race conditions that made the memcache plugin become complex and less effective (basically discarding its content way too often)

DOMEAdapter

- DOMEAdapter is the DMLite plugin that improves on Adapter
- Talks REST/JSON to DOME (head or disk) for the metadata and control functions

- Talks HTTP to Apache for the rare cases that need data tunnelling RFIO-style
  - E.g. a gridftp client contacting the wrong disk server, or a gridftp client that does not support redirection

The fastCGI surprise

- DOME speaks REST and JSON, and the choice in the previous version (1.9) was to access it through Apache and fastCGI

- We had the surprise of seeing our favourite fastCGI modules:
  - disappearing: no support for mod_fcgid anymore
  - regressing performance badly, increasing the resource usage by 100 times (for some time we used mod_proxy_fcgi)

XrdHTTP

- XrdHTTP is the HTTP/WebDAV protocol implementation of the Xrootd framework.
- Allows extensions (e.g. new HTTP verbs) through a very simple C++ plugin interface

- Additional bonus: Brian Bockelman selected XrdHTTP for implementing SciTokens and HTTP third party copy plugins
  - The CGI interface received several small enhancements, and it's ready for the next generation Grid storage authorization schemes, following modern standards

XrdHTTP

- Porting DOME to XrdHTTP took 2 days, ~50 lines of code and I never looked back. This was April 2017, the test systems are under massive test since then

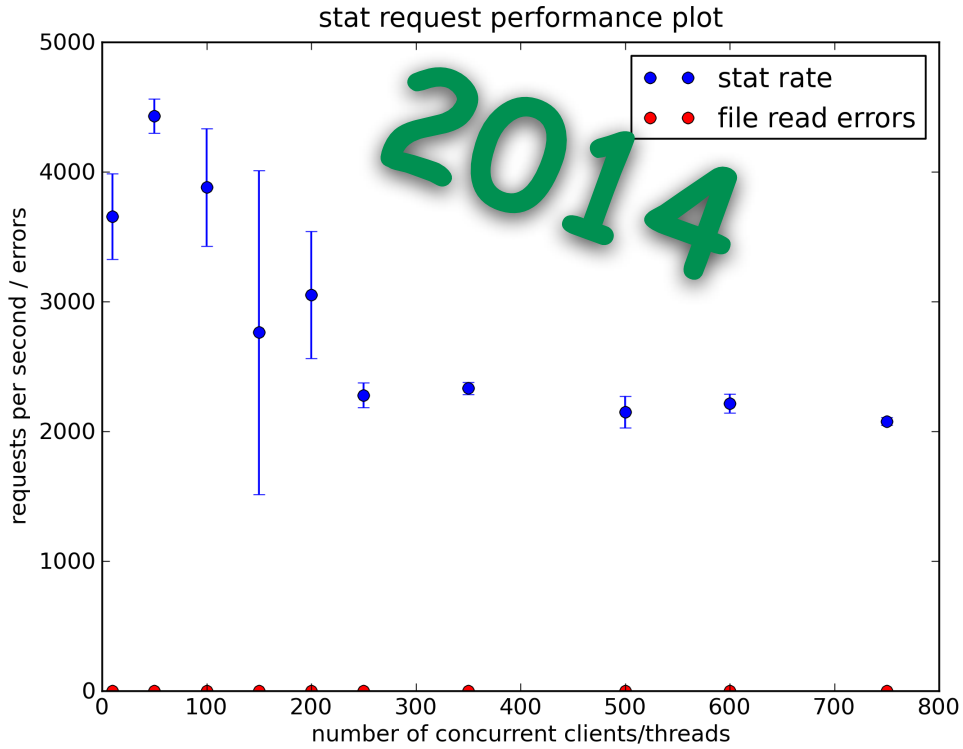
- Apache stays for the data access, like before
- DOME does internal metadata and coordination and resides in the Xrootd process

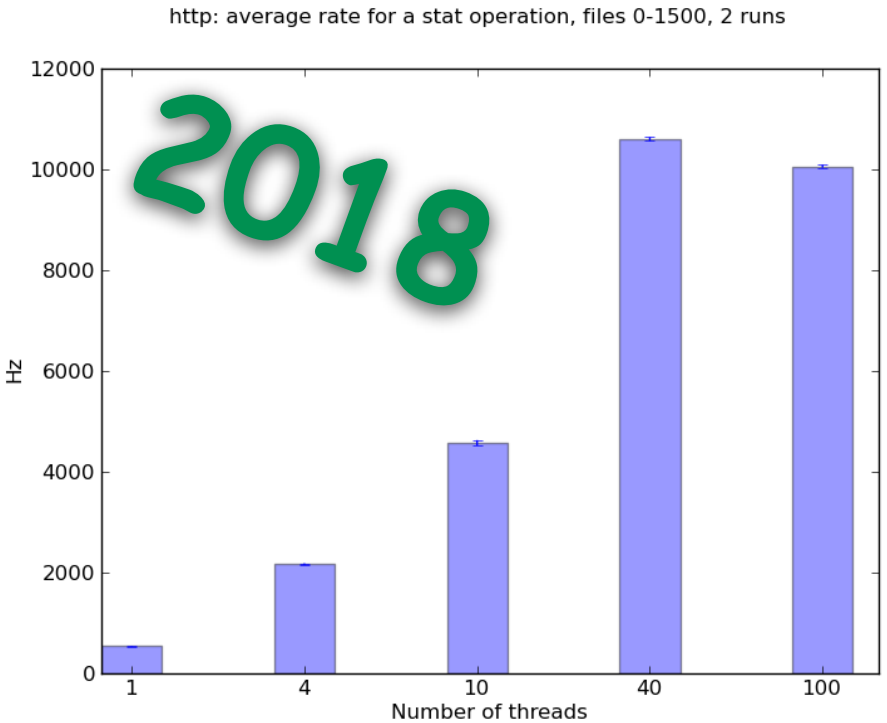
- Result: the metadata transaction rate is now more than 10KHz in our test machine
- A massive stat() test on DPM/DOME now rates ~9-10KHz, stable (more on performance later)
- Incredibly, there's still space for improvement, and I don't think we need it now

- (The author of lcgdm-dav proposed to start using XrdHTTP also for the data, instead of keeping Apache and lcgdm-dav. Surely it will be cheaper, but I am not yet convinced)

Stable performance

- We have a little slot to dig into the perf tests
- https://indico.cern.ch/event/324705/
- Good sign: the absence of fluctuations





No errors

DOME metadata cache

- We like to consider it an internal thing
- Unclear if it will need to touch/tune its parameters
  - Size and TTLs

- Write-through: warm entries are never purged when modified
- Only one cache, with proper locking/synchronization that benefit multithreaded access
- No DMLite API race conditions anymore
- Less items to configure in DMLite

Space accounting

- Only small fixes with respect to 1.9. Based on feedbacks from Oliver, Edith, AndreaM, AndreaS
- Small fix to the dpm daemon to harmonize its behaviour with QTs
- Refinements in DOME and dpmd to support tokenless SRM writes (e.g. for CMS)
- We have a full talk on this

New info provider

- dpm-listspaces belongs to and needs the LCGDM stack to work
- Oliver contributed a new, simpler script that uses DOME
- More in Oliver's talk

Volatile pools and caches

Volatile pools and caches

- In 2016 they were announced as a wish, they are there
- Marking a pool as “Volatile” triggers the cache-like behaviour for that pool. Other pools stay like before
- **A full-file site data cache that works seamlessly and interchangeably with all the data protocols: HTTP, Xrootd, GridFTP**
  - SRM can’t
- More details in Oliver’s talk, nontrivial examples in the two talks by Alessandra Doria, Silvio Pardi and Davide Michelino

Checksums

- Checksums are normally fetched from the DB or calculated/verified

- The same queueing logic as the volatile pools applies to calculating checksums
  - No more than N retrievals per server
  - No more than M retrievals overall
  - Clients peacefully wait their turn, it's transparent
  - Supports a number of checksum types, makes checksum storms safer

Checksums and protocols

- Checksum queueing is is a core feature of DOME
- It's transparent, dmlite and clients only see a checksum request
- Status:
  - HTTP/WebDAV works fine
  - xrootd will, at the next dpm-xrootd minor version
  - Gridftp can't use DOME for checksums, as Globus misses checksum callbacks in the DSI plugins. Gridftp remains vulnerable to checksum storms

Puppet setup

- Working fine, Andrea will give more details

- We published the “blessed” templates directly into the metapackages
  - Makes clear what we wrote and are responsible for
  - Less space for errors or for using older ones on a new system
  - Shields a little bit from puppetforge, which IMO is like a sort of “trunk” repository
    - Puppet templates are code. Where is the quality control there ? Tests, release candidates, etc?
  - Of course we encourage modifications and sharing. Puppetforge may be useful for that

DPM multi-site

- A distributed DPM setup has always been possible
- The older components (libshift, rfio) can pose challenges, solvable on the firewalls

- Alessandra Doria (INFN-NA) kindly accepted to contribute a talk about this. Other people who did it have been Francesco Sciacca(UNIBE) and Alessandro di Salvo (INFN-Rome)

- In DOME mode the setup becomes simpler, as there's no libshift and rfio anymore

1.10.2 is there

- DPM 1.10.2 has finished the quarantine in EPEL-testing since quite some time
- It has been under test in our testbed for many months, with asymptotical fixes, there's nothing since quite some time
- Andrea Sartirana and Alessandra Doria will tell us some impressions tomorrow

- Oliver will push it to EPEL stable on Monday

Some directions

Almost a roadmap

Main technical direction

- Ensure stability for the next years through quality
- Lower maintenance/support cost by promoting simplicity
- Some smaller additions and refinements
- Support is always one of our primary tasks

What we see

- No core changes at the horizon
- A few peripheral additions, maybe some package/build tree refactoring to further reduce the maintenance cost
- The “site consolidation” (less tiny sites, big ones become bigger) will likely continue
- When the SRM load reaches a certain level the clogging problems may force sites to drop it
  - Our effort has been towards making this possible
  - The sysadmin now has the choice, big improvement

xrootd checksums

- Now there is a proper way to request checksum calculations from xrootd to DPM
  - Relatively recent addition to the xrootd API

- The next release of dpm-xrootd will support checksums
  - Full-featured through DOME
  - Usual lcg-dm/dmlite features in legacy mode

DPM-xrootd and DMLite version

- The next release will be harmonized with the version of DMLite
- This cuts the cost of managing the Fedora/EPEL releases
- Consequence: the next dpm-xrootd version will be 1.11.x (epoch 2) for all the xrootd plugins that we provide
  - Now it's 3.6.x

Tuning hints

- Applying the tuning hints can make a difference of even 10-20 times. Evident with just an ‘ls’ on a medium DPM dir
- It’s my standard support suggestion

- Maybe a tool to verify their presence could be an interesting little addition

- The next minor of DOME will check (and clearly log) about file descriptors
- Maybe something can also be done in the startup scripts

3rd party copy

- Would be interesting to queue COPY requests and schedule them like the file pulls or the checksums
- Would be interesting to use DPM as a protocol translator
  - E.g. talking HTTP, request to push a file to an external gridftp erndpoint

- For HTTP (lcgdm-dav) it's doable, and we will explore these possibilities for Dynafed (HTTP/Cloud storage federations) [XDC project]

- DPM uses the same Apache/lcgdm-dav frontend as Dynafed. We may want to explore the possibility of acquiring this feature in DPM

DMLite C++ republish

- At the end, LCGDM is not costing much to us, and its retirement is being quite natural, it's basically there untouched
- A big source of cost has instead been the **DMLite C++ interface**
  - Sets many constraints that apply only to its authors :-(
  - Difficult to evolve and simplify, due to ABI things… academical because the user is again us…
- Please note the C++… the C interface is just fine instead

- I don't think that it's used by other external packages or components
- Solution: republish it it as private headers, to cut the unnecessary cost of a public C++ interface not designed to be public

A new logo/TWiki for DPM

- Breaking news, DPM will get a professional logo

- We will also migrate from TRAC to TWiki
- Likely also from SVN to Gitlab

- Right now the new TWiki is almost complete (except for the logo…)
  - https://twiki.cern.ch/twiki/bin/view/DPM/WebHome

Good old LCGDM

- The good old LCGDM has given unprecedented service to the community
- It contains components (e.g. libshift) that are more than 30 years old
- These components have played a big role of the history of CERN data management, including the various CASTOR generations
- The DPM SRM daemons are there, and have pioneered the Grid
- A few particularly unhappy choices (e.g. imake, or SEDding the code while compiling it, or an outdated approach to TCP/ threads) made life difficult
- Lots of glory, and very problematic to maintain nowadays

LCGDM support from 01/Jun/2019

- From **1st of June, 2019** our standard LCGDM support answer will be "**there is an alternative: upgrade to DOME flavour, please**"
  - That affects: dpns, dpmdaemon, rfio, CSec, dmlite::Adapter, SRM

- LCGDM will stay in EPEL as long as it compiles untouched in Rawhide (EPEL rules will remove it the day it breaks)

- It's pure C, hence that can be even years, we don't give limits

LCGDM support from 01/Jun/2019

- From Mid-2019 we will not fix it if it breaks
- From Mid-2019 the standard support answer will be “upgrade, please”, and… what about SRM ?

- This is why we are interested in the talk of Wei Yang later today

- Wei will summarise steps to allow an ATLAS tier-2 to work without SRM

- Of course, if DPM (newer DMLite codebase) misses something we will work on it, e.g. the xrootd checksums

Spare slides

The fastCGI saga

- The best hint I had from the forums for the bad regression is that PHP programmers prefer monothreaded, synchronous components
- Hence the “natural” solution for them was to disable the fastCGI connection reuse and the overlapping requests directly in the code of the Apache module
  - Yes, connection reuse with mod_proxy_fcgi is broken and will very likely stay broken

- Result: performance lower than 100 transactions per second, with very high resource consumption (hence high instability). Almost worse than SRM.
- I (FF) have wasted one month full time on this, around Feb/March 2017

fastCGI… other options ?

- Nginx surely fits the use case
  - Its community seems certainly more performance-aware than Apache's
  - Who wants one more daemon technology in the head node?
  - A new framework to learn and write low-ish level software for
  - More setup hassle for sysadmins and puppet gurus

- The Xrootd framework has an HTTP interface: XrdHTTP
  - Well known by our community
  - Designed to be lightweight
  - Provides a pragmatic sort-of-CGI interface
  - Every WLCG site already has Xrootd

XrdHTTP

- XrdHTTP is the HTTP/WebDAV protocol implementation of the Xrootd framework.
- Allows extensions (e.g. new HTTP verbs) through a very simple C++ plugin interface

- Additional bonus: Brian Bockelman selected XrdHTTP for implementing SciTokens and HTTP third party copy plugins
  - The CGI interface received several small enhancements, and it's ready for the next generation Grid storage authorization schemes, following modern standards