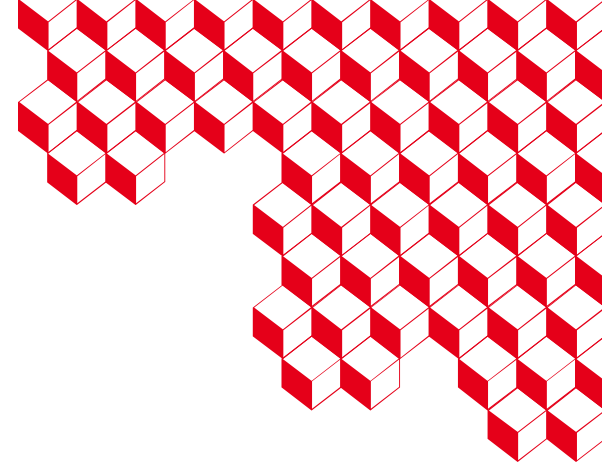




RobinHood v4 Progress Report

LAD'26 – 30th of September, 2026

Gauthier EVRAERD, gauthier.evraerd@cea.fr

Yoann VALERI, yoann.valeri@cea.fr

Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

What is RobinHood?

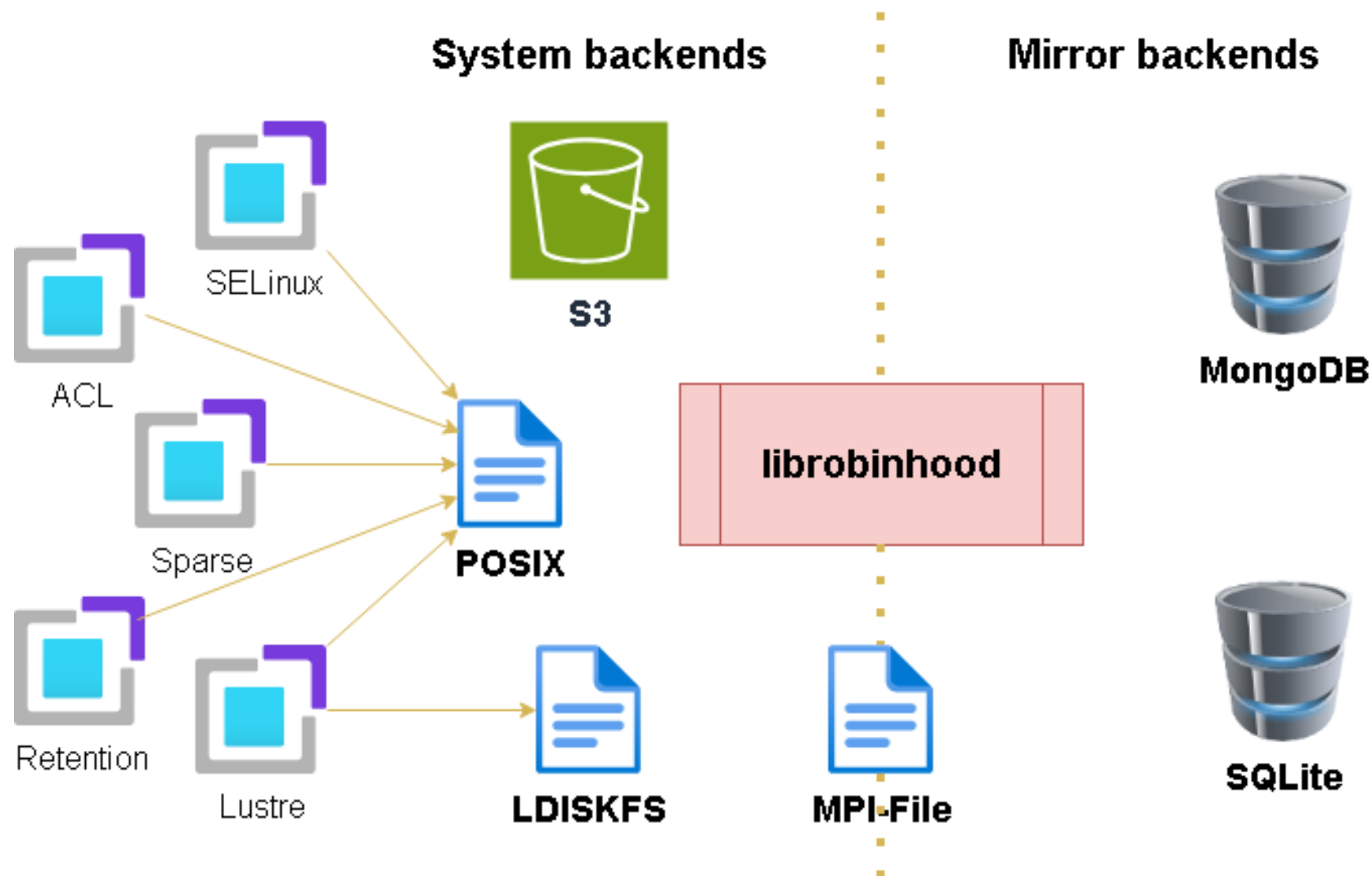


Provide efficient and easy to use tools to replicate and query any storage system's metadata

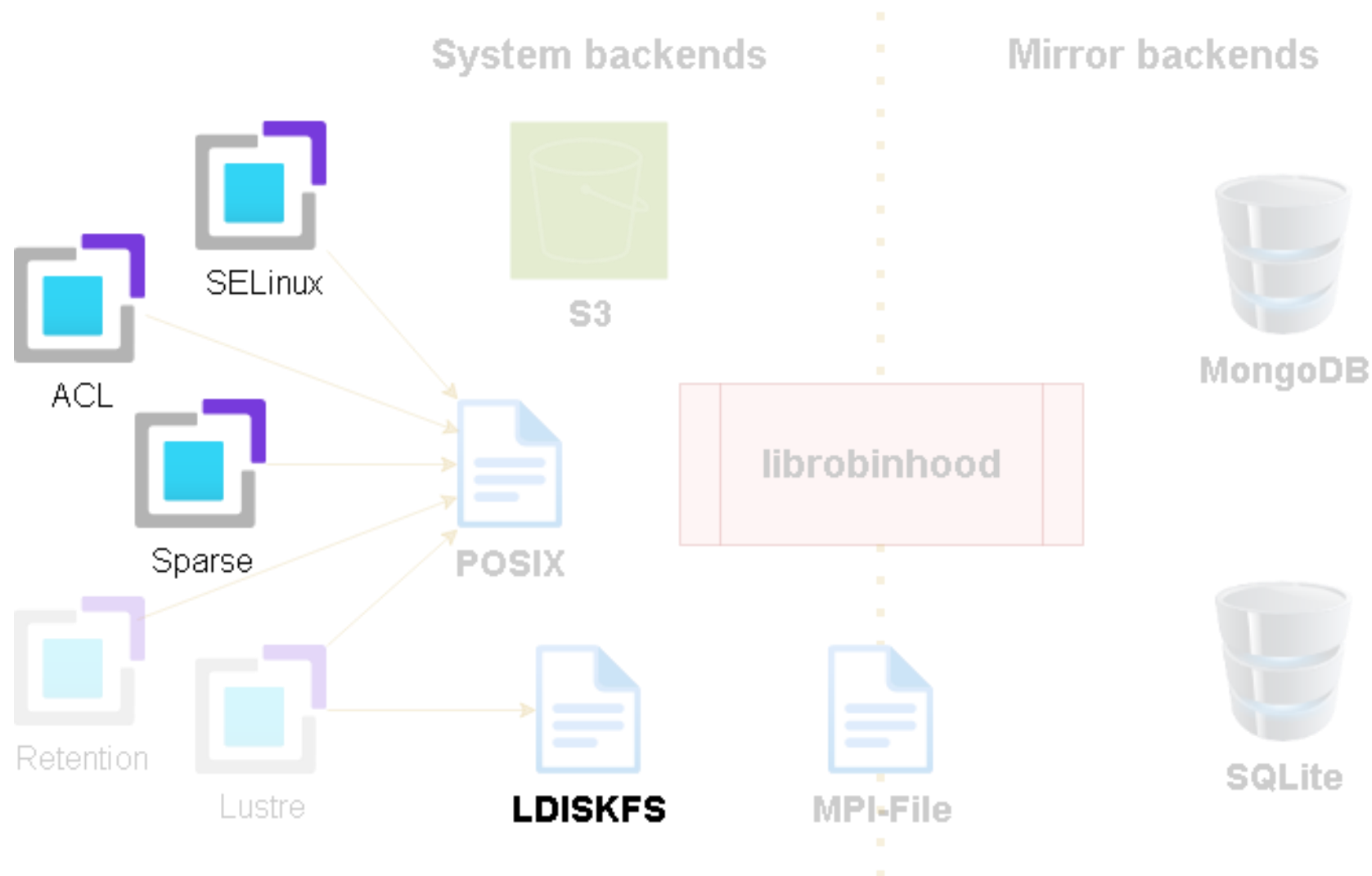


1 ■ Tools and backends

Current RobinHood v4 – Backends



Current RobinHood v4 – Backends



What's new - Backends (1)

■ SELinux and ACL

- Record user, role, type and range + file and directory permissions + default ACLs for directories
- Filter on each of those values + range domination
 - Extended “-readable/-writable/-executable” for ACLs
- Two POSIX extensions

Print of SELinux info

```
rbh-find rbh:mongo:blob -printf "%p: %RZu:%RZr:%RZt:%RZR\n"  
/: user_u:object_r:user_tmp_t:s0  
/file: user_u:object_r:user_tmp_t:s0:c0,c3-s3:c0.c3,c1000
```

Print of ACL info

```
rbh-find rbh:mongo:blob -printf "%p: %RAa::%RAd"  
/: None::None  
/file: user::rwx
```

What's new - Backends (2)

■ LDISKFS

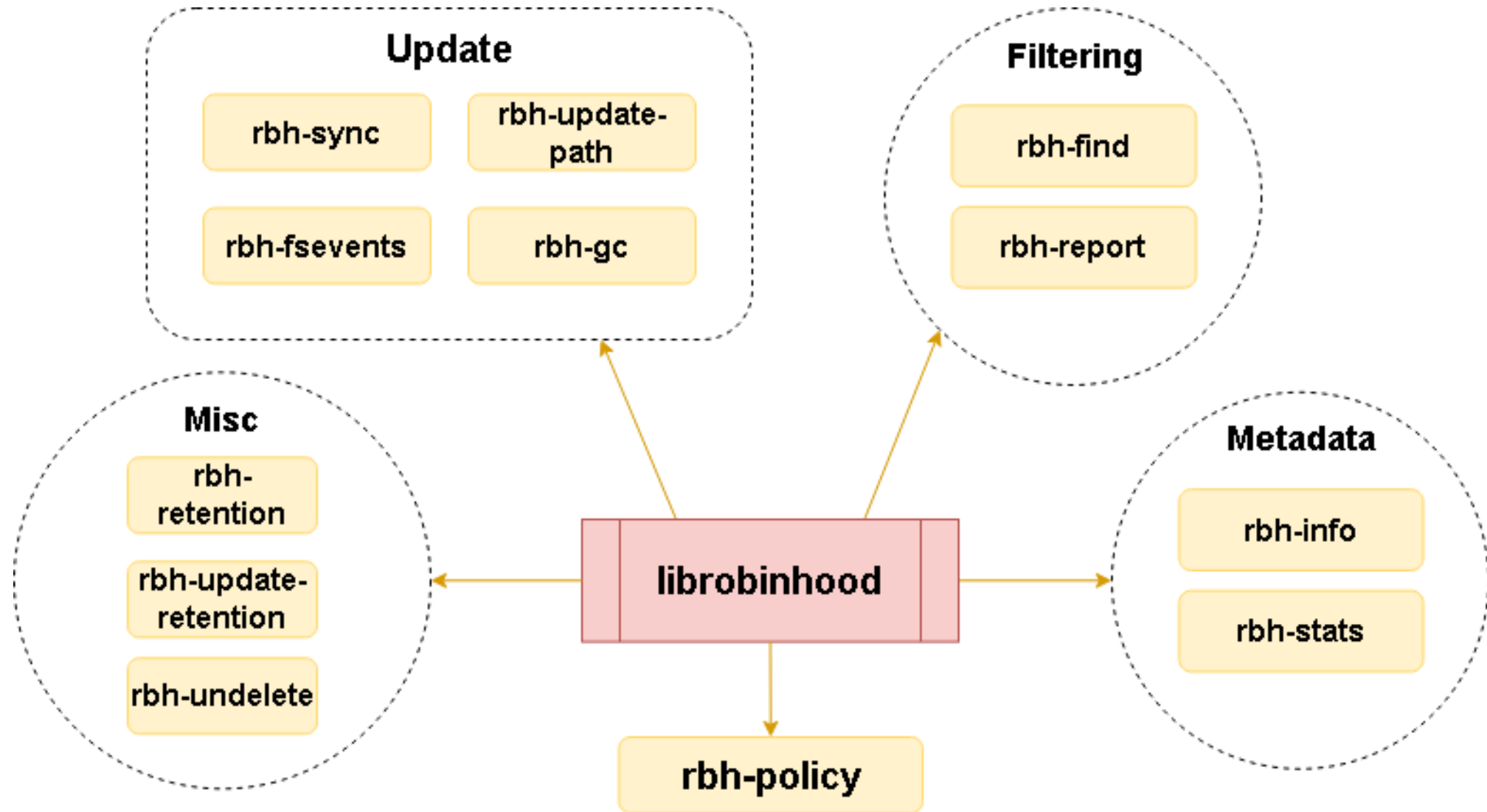
- Scan an MDT or OST directly on disk
- Record every information the Lustre extension can
- Pros
 - Way faster than the Lustre extension
 - Can be used during maintenance of the Lustre FS
- Cons
 - Scaling issues
 - Impact on the MDTs not measured yet
- Still a WIP

```
Time comparison on 1M files

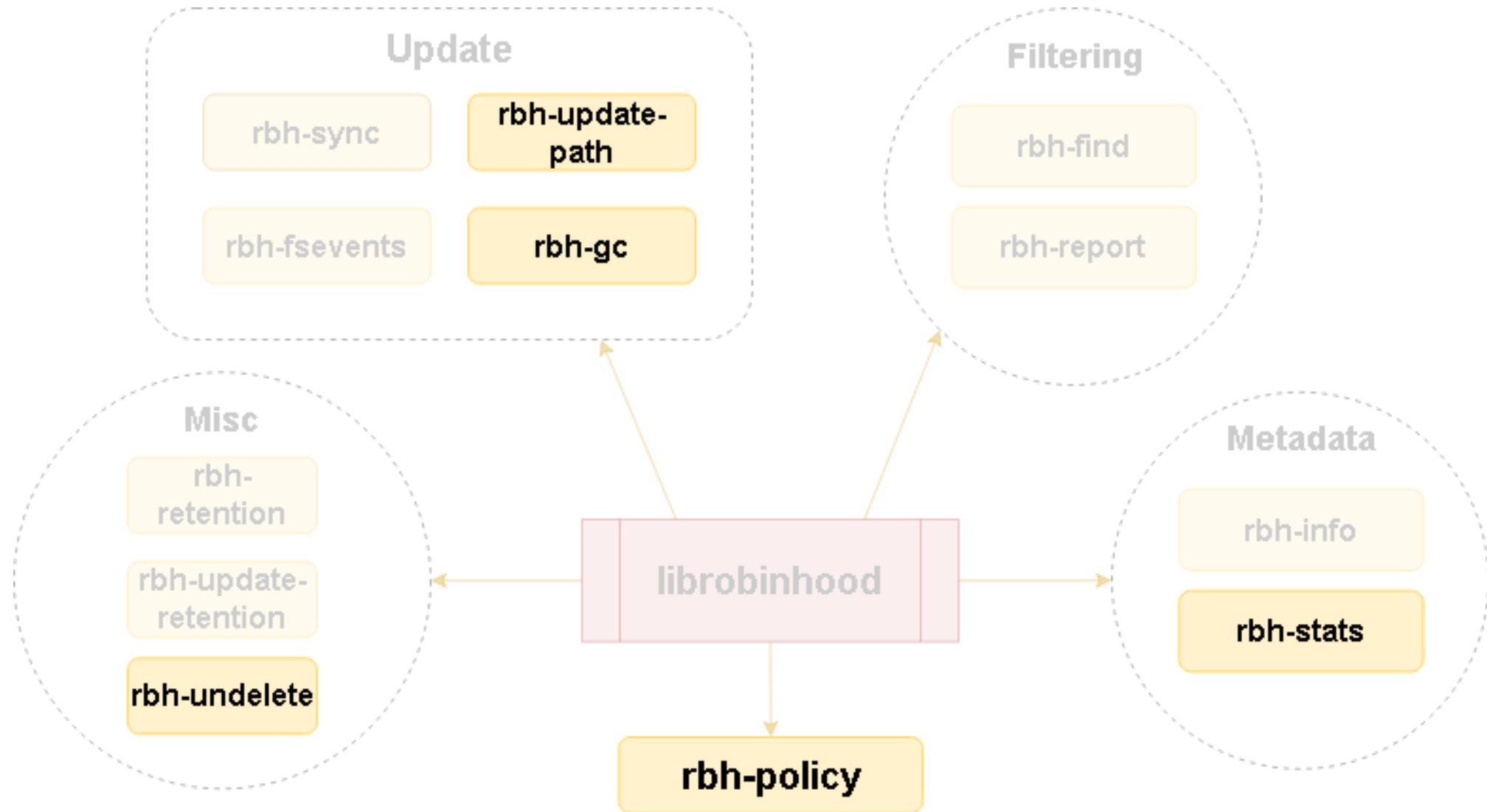
$ time rbh-find
    rbh:ldiskfs:/dev/mapper/mds1_flakey
real 0m6.153s
user 0m4.687s
sys 0m1.464s

$ time rbh-find
    rbh:lustre:/mnt/lustre
real 12m44.986s
user 0m13.279s
sys 6m28.208s
```

Current RobinHood v4 - Tools



Current RobinHood v4 - Tools



What's new - Tools

- **rbh-undelete**, same as RobinHood3
 - Restore deleted entries that have been archived
- **rbh-gc**, same as RobinHood3
 - Delete entries from the mirror system that are no longer in the source system
- **rbh-update-path**
 - Update the path of entries under a directory after its rename
 - Necessary because RobinHood4 stores the full path of entries

Output of rbh-undelete

```
$ rbh-undelete rbh: mongo: blob rbh: lustre: entry --restore
'/mnt/lustre/entry' has been undeleted, new FID is '[0x200000404:0x25:0x0]'
```

Copytool output

```
lhsmtool_posix: 1789728021.190237 lhsmtool_posix[38505]: action=2
src=[0x200000404:0x23: 0x0] dst=[0x200000404:0x25:0x0] mount_point=/mnt/lustre/entry
lhsmtool_posix: 1789728021.190907 lhsmtool_posix[38505]: rebind [0x200000404:0x23:0x0] to
[0x200000404:0x25:0x0]
lhsmtool_posix: 1789728021.191069 lhsmtool_posix[38505]: process finished, errs: 0 major, 0
minor, rc=0 (Success)
```

Output of rbh-gc dry-run

```
$ rbh-gc --dry-run rbh:mongo:blob
'/file' needs to be deleted
1 element total to delete
```

What's new – Metrology

- Two parts: stat printing and a new tool for viewing stats afterwards
- Stat printing
 - During runs of commands (rbh-sync, rbh-fsevents and rbh-gc)
 - Recording of stats at the end of commands (rbh-sync, rbh-fsevents, rbh-find, rbh-report and rbh-gc)
- New tool **rbh-stats**
 - Print stats recorded by the commands
 - Multiple formats available

What's new – Metrology



Output of rbh-sync --stats

```
$ rbh-sync rbh:posix:. rbh:mongo:blob --stats
STATS | ===== Dumping stats at 21/09/2026 11:01:18 =====
STATS | ===== General statistics =====
STATS | Started command: rbh-sync
STATS | Start time: 21/09/2026 11:01:05 (13s ago)
STATS | ===== Backend scan statistics =====
STATS | rbh-sync is running:
STATS |     progress: 100001 entries scanned (0 skipped)
STATS |     current speed: 7692.38 entries/sec
STATS | ===== Plugin 'posix' statistics =====
STATS | POSIX:
STATS |     regular files seen: 99999
STATS |     directories seen: 2
STATS |     symbolic links seen: 0
STATS |     others seen: 0
```

What's new – Project ID

```
Project ID examples

$ touch project_1
$ touch project_2
$ touch vanilla

$ lfs project -p 1 project_1
$ lfs project -p 2 project_2

$ rbh-sync rbh:lustre:. rbh:mongo:blob
$ rbh-find rbh:mongo:blob -project-id 0
/
/vanilla

$ rbh-find rbh:mongo:blob -project-id 1
/project_1

$ rbh-find rbh:mongo:blob -project-id 2
/project_2

$ rbh-find rbh:mongo:blob -project-id 1 -o -project-id 2
/project_1 /project_2

$ rbh-find rbh:mongo:blob -not -project-id 1
/ /project_2 /vanilla

$ rbh-find rbh:mongo:blob -printf '%p: %RLp\n'
/: 0
/project_1: 1
/project_2: 2
/vanilla: 0
```

Project status

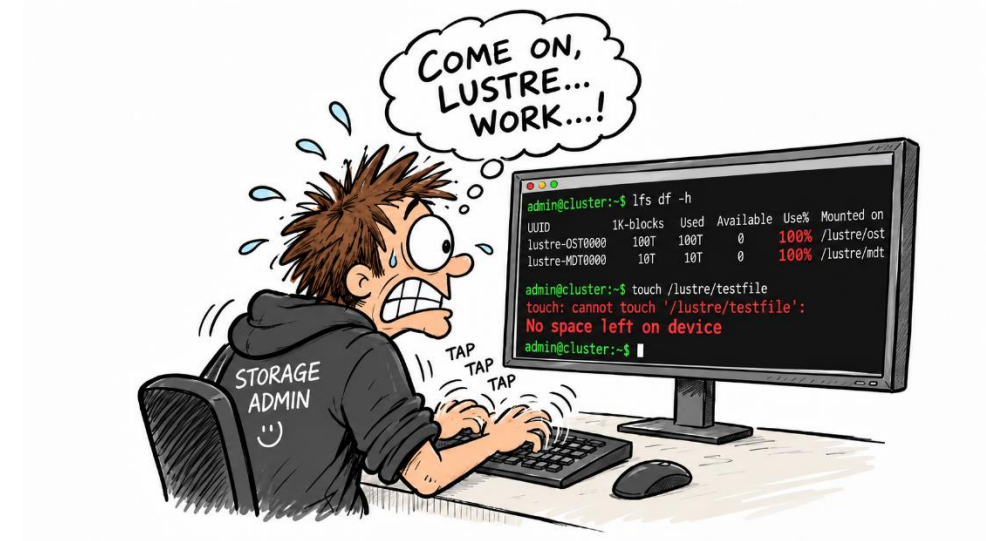
- RobinHood v4 version 1.5.2
- Tested with
 - RHEL 8.8, 8.10, 9.4, 9.8 and 10.2
 - Lustre 2.12 and 2.15, will also test for 2.18/master
 - MongoDB 7 and 8
- Containerfile available for testing
- Deployed on the EXA supercomputer
 - Tailored for the retention feature



2 ■ Focus on rbh-policy

Why rbh-policy ?

- RobinHood 4 was initially designed without a policy engine. However, the lack of a policy engine is a major barrier to RobinHood 3 users migrating to RobinHood 4.
- All of these scripts can now be replaced by policies using *rbh-policy*



Configuration

- Like in RobinHood 3, all the policies are declared in configuration files



- The *rbh-policy* configuration files are written in Python, so they are different from the configuration files used by the other RobinHood 4 tools.
- The Domain-Specific Language (DSL) syntax has changed, but we have tried to keep the same concepts

RBH3 vs RBH4	
RBH3	RBH4

General	<code>Config()</code>
FileClass	<code>declare_fileclass()</code>
<code>*_rules</code>	<code>declare_policy()</code>
<code>*_rule</code>	<code>Rule()</code>
<code>*_trigger</code>	<code>Trigger()</code>

Fileclasses

Fileclasses are sets of rules to define on which targets a policy will apply to

```
FileClasses

declare_fileclass
    name = "big_files",
    target = (Size >= "10GB") & (LastAccess > "60d")
)
```

- Many attributes are available for conditions: path, type, user, size, atime, ...
- Attributes can be combined using logical operators such as and, or, not

Several fileclasses can be combined to create more complex fileclasses

```
Combine fileclasses

purge = big_files & old_files
```

Actions



Types of available actions

- Predefined actions
- External actions (commands, Bash scripts, ...)
- Python functions

Available predefined actions

- **Log** → displays one for each matched entry
 - Printf style formatting is supported
- **Delete** → deletes matched entries

Triggers

Triggers define the conditions under which a policy should be executed

- **Capacity triggers**
 - Triggers based on the capacity of the mount point (no Lustre-specific triggers for now)
- **Mirror triggers**
 - Triggers based on the information from the metadata mirror
- **Temporal triggers**
 - Triggers based on time-related conditions

● ● ● Capacity triggers

GlobalSizePercent	90
GlobalInodePercent	90

● ● ● Mirror triggers

```
MirrorUserInodeCount "alice") > 1000  
MirrorUserDiskUsage("alice") > "5TB"  
MirrorGroupInodeCount "grp") > 5000  
...
```

Triggers can also be combined like fileclasses

Policies

Policies are declared using ***declare_policy()***

- An ordered list of rules which can modify the action applied to a subset of entries
- The processing order of matched entries
- Condition under which the policy should stop

```
Policy

declare_policy
    name,
    target,
    action,
    parameters,
    trigger,
    rules,
    sort,
    stopthreshold
)
```



Example of a purge policy : configuration

Configuration example 1/2

```
rbhpolicy.config.core import *

config(
    filesystem = "rbh:lustre:/mnt/lustre",
    database = "rbh:mongo:mylustre",
    evaluation_interval = "5m"
)

old_files = declare_fileclass(
    name = "old_files",
    target = (Type == "f") & (LastAccess > "60d")
)

logs_files = declare_fileclass(
    name = "logs_files",
    target = (Name == "*.log")
)
```

Configuration example 2/2

```
declare_policy
    name = "cleanup_old_non_logs",
    target = old_files & ~log_files,
    action = action.delete,
    parameters = {"remove_empty_parents": True},
    trigger = Periodic("7d"),
    rules = [
        Rule(
            name = "skip_alice_files",
            condition = (User == "alice"),
            action = lustre.log,
            parameters = {
                "format": "skipped: fid=%Rlf, user=%u"
            }
        )
    ]
)
```

Example of a purge policy : execution



Example

```
$ rbh-policy run /etc/robinhood4.d/pe.py cleanup_old_non_logs

Info | Loaded configuration for '/etc/robinhood4.d/pe.py'
Info | Execution policy 'cleanup_old_non_logs'
DeleteAction | deleted 'purge_all/old_data.bin'
DeleteAction | deleted 'purge_all/old_report.txt'
DeleteAction | removed empty parent directories
LogAction    | skipped: fid=[0x200000404:0x2d:0x0], user=alice
DeleteAction | deleted 'mixed/old_notes.txt'
DeleteAction | deleted empty parent directories
Info | Policy 'cleanup_old_non_logs' completed
```

Incoming features

- Daemonize *rbh-policy*
- Add more actions, such as HSM actions (archive, release, ...)
- Add Lustre-specific triggers, such as pools and OST capacity triggers
- Add execution statistics, such as the number of entries processed by each policy and rule



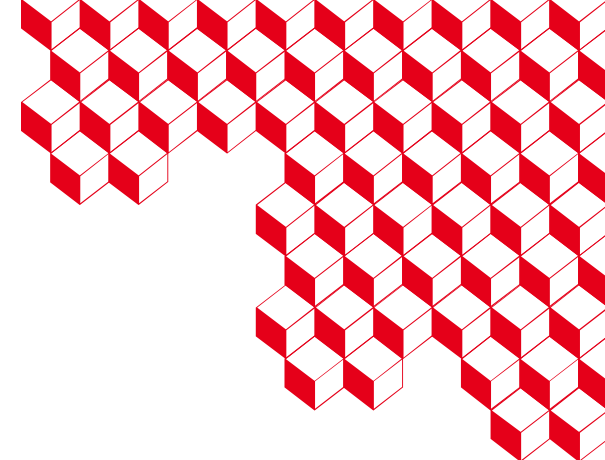
3. Conclusion

What's next?

- Finish **rbh-policy**
- **rbh-fsevents** as a daemon
- Accounting
- Userland usage

Want to help with the development ?

- The tool suite is on Github: <https://github.com/robinhood-suite/robinhood4>
- Reviews done on the Gitlab Forge → Requires access to the Inti supercomputer
- Feel free to install the suite and test it out for yourself, any feedback is appreciated!



Thanks for your attention

RobinHood v4 progress report

Commissariat à l'énergie atomique et aux énergies alternatives – www.cea.fr

Gauthier Evraerd, Yoann Valeri
Software stack developers at CEA
guillaume.courrier@cea.fr
gauthier.evraerd@cea.fr
sebastien.gougeaud@cea.fr
yoann.valeri@cea.fr

Why a version 4 for RobinHood?



	Version 3	Version 4
Scale up	SQL paradigm <i>MariaDB</i>	NoSQL paradigm <i>MongoDB</i>
Code genericity	Software specialised for Lustre filesystems	Generic tools calling specific backends
Inclusion to Linux repositories	Expert system	Library of features and applications
Code refactoring	Heavy code caused by Lustre behaviour evolution	Clean design to better correspond to current filesystems