Pacman can have the cookies and eat the ghosts too

HIP2022: Binary package repository management tooling for pacman based distributions

David Runge

2022-12-27

Contents

Outline

Packages

Package repositories

Dbscripts

Repod

Try out

Contact

Who?

- Arch Linux Package Maintainer (2017)/ Developer (2019)
- Pro-audio, Python, Rust, installation process, packaging, infrastructure
- Free and open source software development

What?

- libalpm[1] based packages
- Metadata validation
- Package repositories for pacman[2] based distributions
- Privilege separation when manipulating and hosting package repositories
- Hoster, user and end-user use-cases

[1] https://man.archlinux.org/man/libalpm.3

[2] https://man.archlinux.org/man/pacman.8

What not?

- Building packages
- Package sources (just a little...)

Packages

What is a package anyway?

- A (compressed) tar[3] file
- Files to be installed on a target system
- Files describing metadata
- Scripts running actions

[3] https://man.archlinux.org/man/tar.1

Package metadata

- .BUILDINFO[4] - build environment information
- .MTREE[5] - package contents
- .PKGINFO[6] - package metadata

[4] https://man.archlinux.org/man/core/pacman/BUILDINFO.5.en
[5] https://man.archlinux.org/man/mtree.5
[6] https://gitlab.archlinux.org/pacman/pacman/-/merge_requests/27

Package repositories

Package repositories

- Architecture-specific directory structure
- Packages, signatures and repository sync databases
- Exposed to user systems by a web server
- Pacman downloads files and synchronizes against local state

Repository sync databases

- Live alongside packages in a repository
- Contain information about packages in a repository
- Two per repository
- Detached signatures may be provided
- Used by pacman to install and update packages from a repository

```
.
|-- /srv/ftp/repo
|   `-- x86_64
|       |-- package-1.0.0-1-x86_64.pkg.tar.zst -> ../../pool/package-1.0.0-1-x86_64.pkg.tar.zst
|       |-- package-1.0.0-1-x86_64.pkg.tar.zst.sig -> ../../pool/package-1.0.0-1-x86_64.pkg.tar.zst.sig
|       |-- repo.db
|       `-- repo.files
`-- /srv/ftp/pool
    |-- package-1.0.0-1-x86_64.pkg.tar.zst
    `-- package-1.0.0-1-x86_64.pkg.tar.zst.sig
```

Wait, there are two types?

- Default database (*.db* suffix) used for syncing current state of repository and calculating diff for installation and update

```
.
|-- package-1.0.0-1
|   `-- desc
[..]
```

- Files database (*.files* suffix) used for operations on filenames of files contained in packages of a repository

```
.
|-- package-1.0.0-1
|   |-- desc
|   `-- files
[..]
```

dbscripts[7] - The good, the bad and the ugly

[7] https://gitlab.archlinux.org/archlinux/dbscripts

History

- Started in 2002
- Developed and used by Arch Linux
- Used to maintain core, extra, community and multilib

The good

- Served and worked (mostly) well for what it does
- Extensive (integration) test suite
- Design pattern (e.g. symlinks for files in repo)
- Somewhat integrated with devtools[8]
- Soon able to deal with git

[8] https://gitlab.archlinux.org/archlinux/devtools

The bad

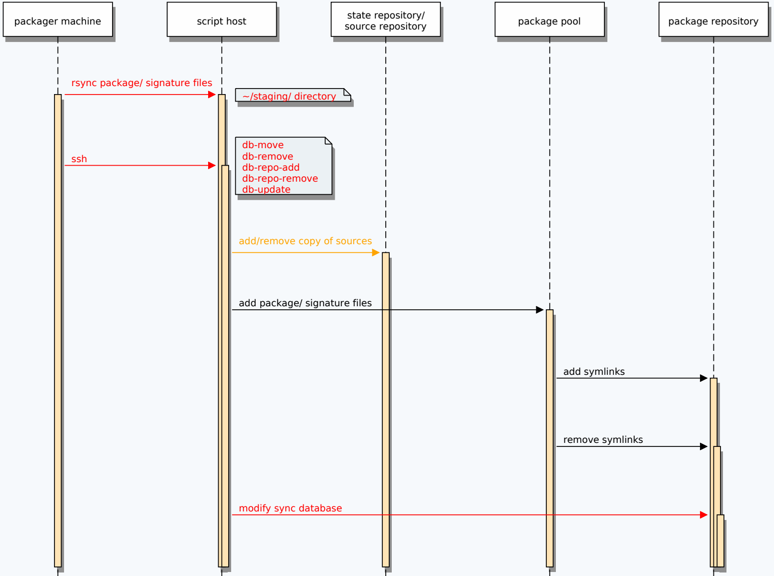
- Bash based scripts
- Called by packagers on central system providing binary package repositories
- Synchronous / blocking
- Hard-wired to SVN based package sources repository (for ”state”)
- Limited input (package metadata) validation

The ugly

- Centered around repo-add[9]
- Some transactions are not safe (e.g. move between (multiple) repositories/ stability layers)
- User management using Unix groups
- Sync database creation is additive
- Close to no documentation

[9] https://man.archlinux.org/man/repo-add.8

Workflow overview



repod[10] - A new beginning

[10] https://gitlab.archlinux.org/archlinux/repod

Background

- Workgroup created Python based PoC during/after Arch Conf 2019
- Implementation from 2021 onwards (mostly internals)
- Documentation[11] (also covering libalpm/pacman internals)
- Attempts at getting funding: Prototype Fund, NLnet, Valve

[11] https://repod.archlinux.page/

Core concepts

- Validation
- (Real) Client-server model
- (Cross-repository) Atomic actions
- Robustness
- Documentation

Management repository[13]

- Contains machine readable files (i.e. JSON)
- Can be verified (using JSON schema)
- Maintains repository state
- Repository sync databases are created *in full* and *reproducibly* from it
- Tooling (*repod-file*[12]) for importing and exporting of sync databases
- Can be backed by a version control system to track changes

```
.
`-- x86_64
    |-- repo
    |   |-- pkgnames
    |   |   `-- pkgname.json -> ../pkgbase.json
    |   `-- pkgbase.json
[..]
```

[12] https://repod.archlinux.page/repod/man/repod_file.html

[13] https://repod.archlinux.page/repositories/management_repository.html

Metadata and file validation

- Extensive validation of package metadata
- Various parsers (*.BUILDINFO*, *.MTREE*, *.PKGINFO*)
- Creation of versioned metadata objects to cover past and future upstream changes (pacman mostly does not track changes to metadata or sync database files)
- Tooling (*repod-file*) for inspecting package metadata

Configuration[17]

- Sane and flexible configuration file format (TOML[14])
- Allow per repository overrides and drop-ins
- Per user vs. system-wide (XDG base dir[15] and FHS[16])
- Distribution vs. user needs
- Implied stability layers

[14] https://toml.io/en/

[15] https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html

[16] https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

[17] https://repod.archlinux.page/repod/man/repod_conf.html

As a service

- Repod as system service (separate user), exposing an API
- Integration with identity and access management services
- Packagers interact using authenticated client-side-tooling
- Service interacts with signing enclave (for sync databases)

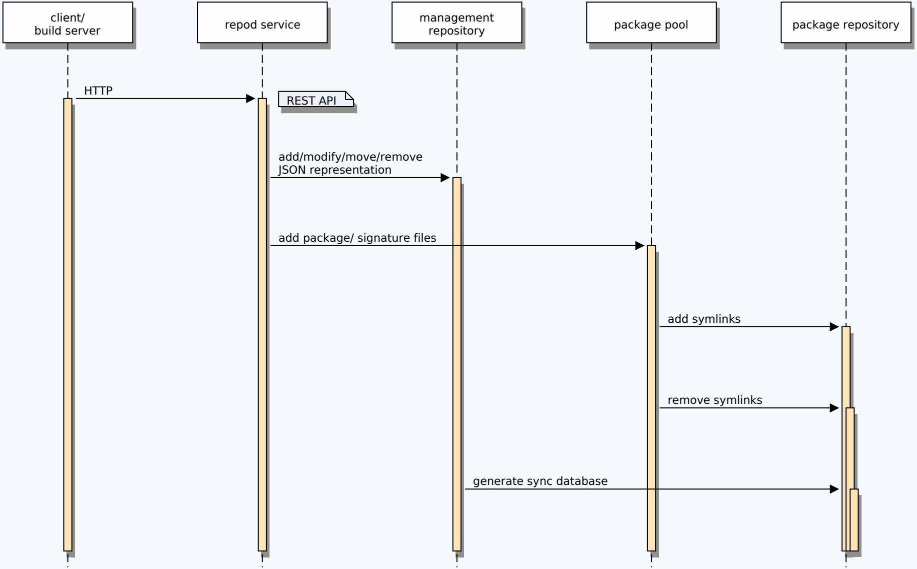
Code and documentation

- Typed Python $>=$ 3.10
- 100% test coverage
- *All* code is documented
- Extensive user-facing documentation (also on concepts)
- Man pages

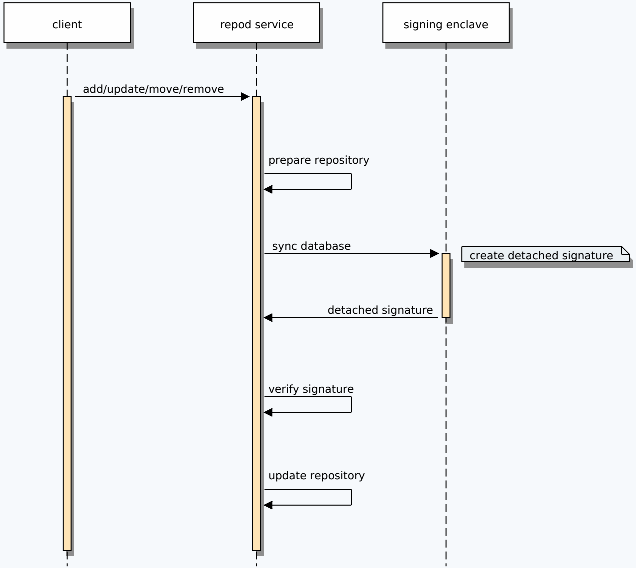
Workflows

- Atomic actions, relying on revertible transactions
- Reusable building blocks
- Grouped multi-repository use-cases

Overview: Workflow



Overview: Signing enclave



Still a lot to do...

Upcoming work (midterm)

- Further workflows (move, remove, cross-repository add/move/remove)
- Integration of git tooling (for source repository[18] integration and management repository backend)
- Validation of package source repositories (using .SRCINFO[19] files)
- Integration of PGP tooling for validation and signing of repository sync databases
- Improve logging concept
- More integration tests
- Find more contributors
- Find funding

[18] https://repod.archlinux.page/repositories/source_repository.html

[19] https://repod.archlinux.page/repositories/source_repository.html#srcinfo

Upcoming work (longterm)

- Git backend for management repository
- Cache for management repository
- Repository snapshots
- Repository ACLs
- User management (via configuration and identity management service)
- API and client

Contribute

- Bi-weekly meetings (announced on arch-projects mailing list[20])
- IRC: #archlinux-projects on libera.chat

[20] https://lists.archlinux.org/archives/list/arch-projects@lists.archlinux.org/

Let’s try it!

Contact

David Runge

Mail: dave@sleepmap.de

Matrix: @dvzrv:matrix.org

IRC: dvzrv@{hackint,libera,oftc}