# Haiku's package management

## Design and issues with ported software

François Revol
revol@free.fr

FOSDEM'16

# Haiku

- Free Software rewrite of BeOS
- An Operating System for the desktop
- A lot of POSIX
  - But we don't claim to be Unix®
- Some more funny things
  - Typed & indexable xattrs

# Before

- Plain ZIP files
  - Extract anywhere, “Drag foo here” symlinks…
- BeOS PKG files
  - self-contained click-through + scripts
- `installoptionalpackage` command
  - Still zip files
- Not maintainable 😱
- Not scalable 😫

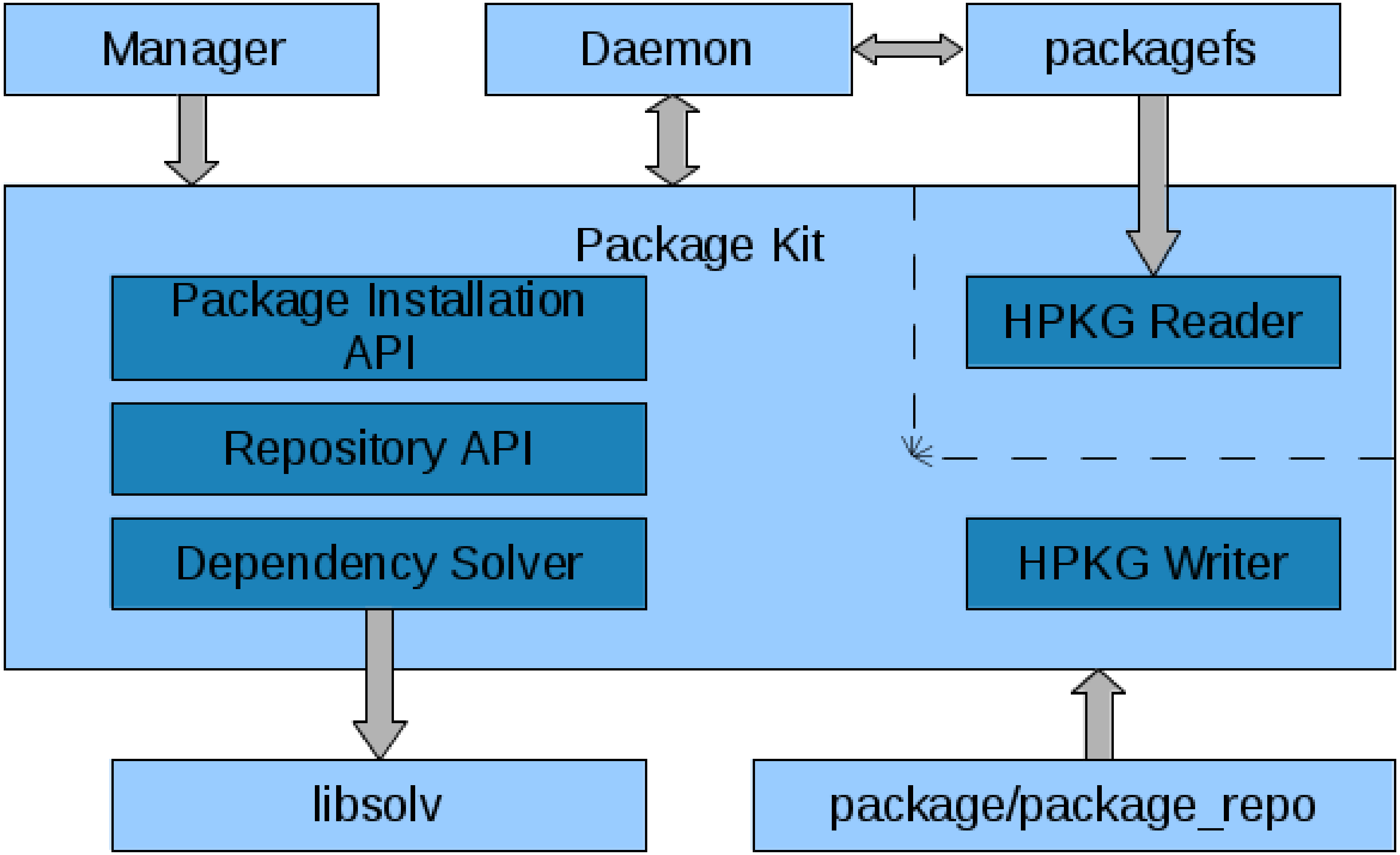
# Needs

- Simplicity for the user
- Dependency management
- Automated building
- Secondary architecture support
  - gcc4 vs gcc2 (binary compat)
  - x86_64

# The vision

- First draft, jan. 2011
- HaikuPorts tree similar to portage or BSD ports
- `haikuporter` builds packages
- `package` & `package_repo` tools
- `packagefs` (readonly) virtual file-system
- `package_daemon` handling transactions
- `pkgman` & HaikuDepot (GUI) handle updates

# The vision


Manager
Daemon
packagefs
Package Kit
Package Installation API
HPKG Reader
Repository API
Dependency Solver
HPKG Writer
libsolv
package/package_repo

# Consequences

- Some old ugly tricks don't work anymore
  - MIME sniffing system applications to populate xattrs
    - Packages contain them already now
- Some changes in the FS hierarchy
  - `packagefs` has shine-through writable directories
  - `non-packaged/` counterparts in `/system` and `~/config/` to allow manual install
  - `find_directory()` helps, and `find_path()`

# HPKG file format

- Similar to XAR
  - But binary TOC
  - Attribute tree
    - Contains the manifest
- Manifest

```
name           mypackage
version        0.7.2-1
architecture   x86
summary        "is a very nice package"
description    "has lots of cool features\nand is written in MyC++"
vendor         "Me, Myself & I, Inc."
packager       "me@test.com"
copyrights     { "(C) 2009-2011, Me, Myself & I, Inc." }
licenses       { "Me, Myself & I Commercial License"; "MIT" }
provides {
       cmd:me
       lib:libmyself = 0.7 }
requires {
       haiku >= r1
       wget }
```

- Bootloader package not compressed

# Package Kit

- API to
  - Read & Write packages
  - Install
  - Browse repositories
- libsolv used for dependency constraints

# package tool

- → demo

# packagefs

- Virtual filesystem, read-only
- Publishes union of active packages contents
- Mounted on
  - /boot/system
  - ~/config
  - In chroots for haikuporter
- package-links/
  - For each package, symlinks to dependencies

# `package_daemon`

- Handles [de]activating packages
- `/system/packages/administrative/`
  - Transactions
  - Previous states
    - Bootloader can select any
  - Writable files
    - System settings, package gives merging rules

# HaikuDepot

- → demo

# pkgman

- → demo

# Haikuports

- Gentoo-like recipes
- cf. sample

# haikuporter

- (python)
- Chroots, activates BUILD_*REQUIRES'd packages
- Runs the BUILD() and INSTALL() phases
  - `runConfigure ./confiture && make …`
- Creates source package by default
- Still depends on system-installed packaged
  - Not yet bit-reproducible
  - Being fixed RSN

# Repositories

- Web server
- HPKGR binary index files

# Shortcomings

- Most of the filesystem is readonly
  - Applications menu can't be customized
    - There's a workaround
  - Some software don't like it
    - Python/perl/whatever “list of installed stuff”
- Hardcodes dependencies versions in package
  - Things like (libtool archives) `.la` files dislike…
    - Just `rm` them.
  - Ditto with `foo-config` & `pkg-config` files.

# Haiku / PM Tips

- Please, please, don't assume /usr/whatever
  - Always honor all --foo-dir configure args
- Not everywhere is writable even as root
- Sometimes root is called “user” 😛
- Substitute properly in foo-config & .pc files
- Don't require default settings file

# Compared to GUIX?

- Separate tools for building & installing
- Package content is readonly
  - Settings files not handled completely
- Package names for humans (no base32)
- Manual rollback
- No package GC (yet?)
- Not *functional*, but ~~func~~working anyway 😜

# Thanks

- Questions?