Reproducible and Customizable Deployments with GNU Guix

Ludovic Courtès

FOSDEM 2016

The difficulty of keeping software environments under control.

#1. Upgrades are hard.

Distribution Upgrade of all the files:

WARNING

Following the upgrade instructions found in the release notes is the best way to ensure that your system upgrades from one major Debian release to another (e.g. from lenny to squeeze) without breakage!

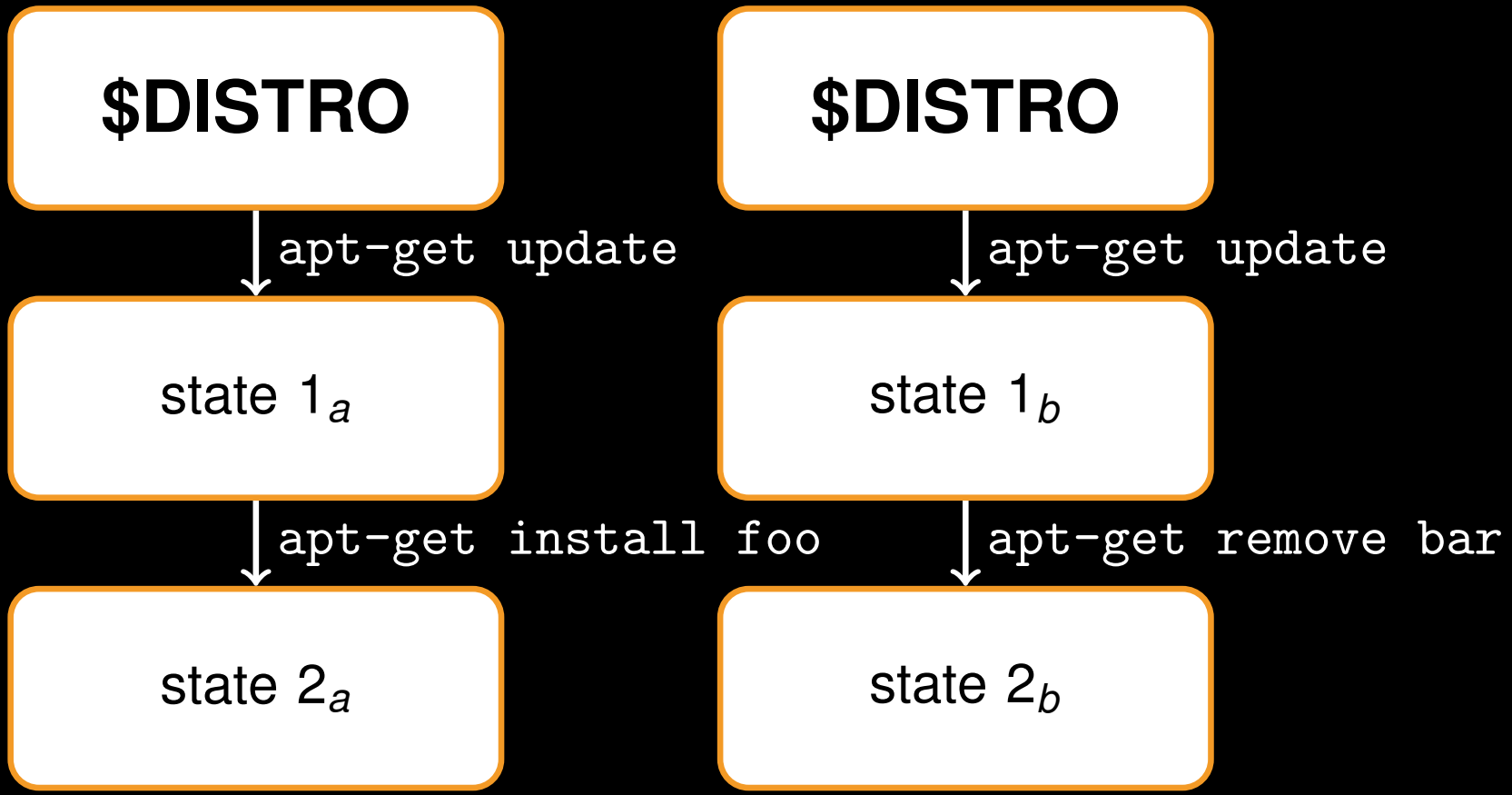
These instructions will tell you to do a `dist-upgrade` (instead of `upgrade`) in the case of apt-get or `full-upgrade` (instead of `safe-upgrade` in the case of aptitude) at least once. So you would have to type something like

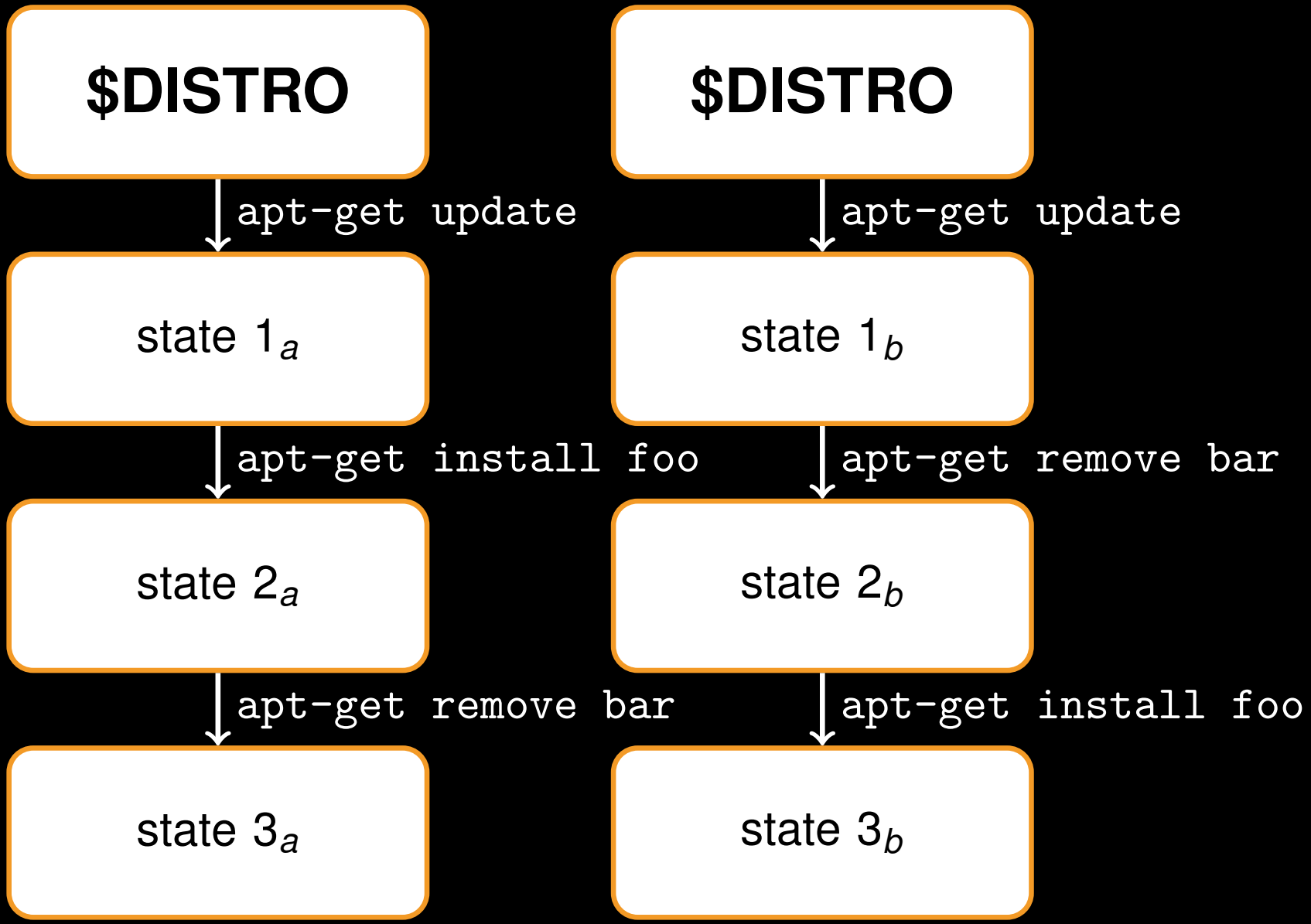
```
# aptitude full-upgrade
```

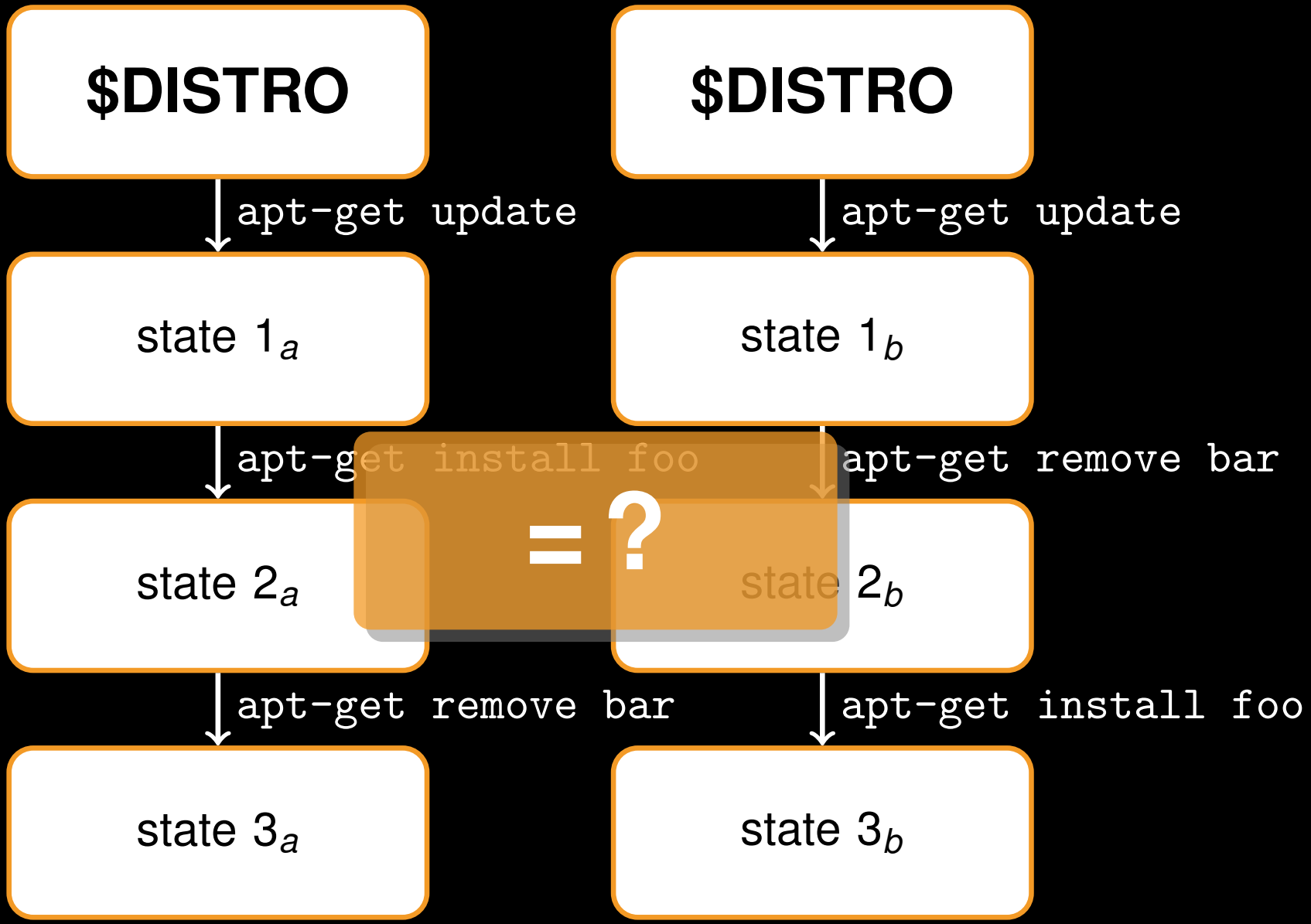
#2. Stateful system management is intractable.


$DISTRO
$DISTRO


$DISTRO
apt-get update
state 1_a
$DISTRO
apt-get update
state 1_b


$DISTRO
$DISTRO
apt-get update
apt-get update
state 1a
state 1b
apt-get install foo
apt-get remove bar
state 2a
state 2b


$DISTRO
$DISTRO
apt-get update
apt-get update
state 1a
state 1b
apt-get install foo
apt-get remove bar
state 2a
state 2b
apt-get remove bar
apt-get install foo
state 3a
state 3b


$DISTRO
$DISTRO
apt-get update
apt-get update
state 1a
state 1b
apt-get install foo
apt-get remove bar
= ?
state 2a
state 2b
apt-get remove bar
apt-get install foo
state 3a
state 3b

#3. It’s worse than this.

Application-level package managers [edit]

- Anaconda - a package manager for Python
- Assembly - a partially compiled code library for use in Common Language Infrastructure (CLI) deployment, versioning and security.
- Biicode - a file-focused dependency manager for C/C++ languages and platforms (PC, Raspberry Pi, Arduino).
- Bower - a package manager for the web.
- UPT - a fork of Bower that aims to be a universal package manager, for multiple evironments and unlimited kind of package
- Cabal - a programming library and package manager for Haskell
- Cargo - a package manager for Rust (programming language)
- CocoaPods - Dependency Manager for Objective-C and RubyMotion projects
- Composer - Dependency Manager for PHP
- CPAN - a programming library and package manager for Perl
- CRAN - a programming library and package manager for R
- CTAN - a package manager for TeX
- DUB - a package manager for D

As of **`npm@2.6.1`**, the **`npm update`** will only inspect top-level packages. Prior versions of **`npm`** would also recursively inspect all dependencies. To get the old behavior, use **`npm --depth 9999 update`**, but be warned that simultaneous asynchronous update of all packages, including **`npm`** itself and packages that **`npm`** depends on, often causes problems up to and including the uninstallation of **`npm`** itself.

To restore a missing **`npm`**, use the command:

```
curl -L https://npmjs.com/install.sh | sh
```

Giving up?

Giving up?

$\rightarrow$ “app bundles” (Docker images)

*“Debian and other distributions are going to be **that thing you run docker on**, little more.”*

— Jos Poortvliet, ownCloud developer

http://lwn.net/Articles/670566/

Tree: **bd378...** ▾ **owncloud** / 7.0 / apache / **Dockerfile**

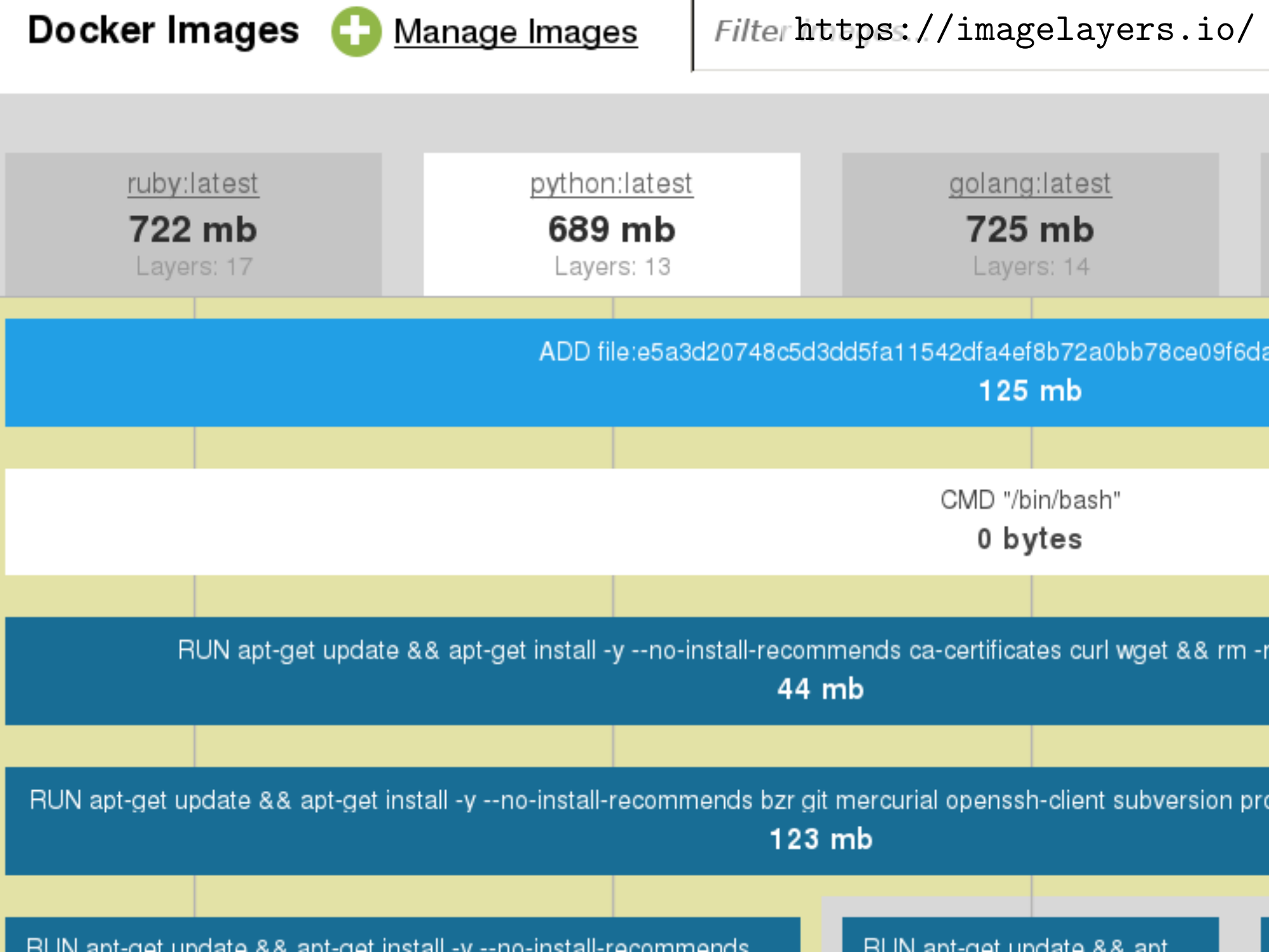
tianon Update to 7.0.12, 8.1.5, and 8.2.2

2 contributors 

50 lines (40 sloc) | 1.58 KB

It's also that thing you run *inside* Docker!

```
1  FROM php:5.6-apache
2
3  RUN apt-get update && apt-get install -y \
4          bzip2 \
5          libcurl4-openssl-dev \
6          libfreetype6-dev \
7          libicu-dev \
8          libjpeg-dev \
9          libmcrypt-dev \
10         libpng12-dev \
11         libpq-dev \
12         libxml2-dev \
13         && rm -rf /var/lib/apt/lists/*
14
```


Docker Images
Manage Images
Filter
https://imagelayers.io/
ruby:latest
722 mb
Layers: 17
python:latest
689 mb
Layers: 13
golang:latest
725 mb
Layers: 14
ADD file:e5a3d20748c5d3dd5fa11542dfa4ef8b72a0bb78ce09f6da
125 mb
CMD "/bin/bash"
0 bytes
RUN apt-get update && apt-get install -y --no-install-recommends ca-certificates curl wget && rm -r
44 mb
RUN apt-get update && apt-get install -y --no-install-recommends bzr git mercurial openssh-client subversion pro
123 mb
RUN apt-get update && apt-get install -y --no-install-recommends
RUN apt-get update && apt


FEN
PIZZA STEINOFEN
Abtropfgewicht € 700 g

Over 30% of Official Images in Docker Hub Contain High Priority Security Vulnerabilities

Docker Hub is a central repository for Docker developers to pull and push container images. We performed a detailed study on Docker Hub images to understand how vulnerable they are to security threats. Surprisingly, we found that more than 30% of images in official repositories are highly susceptible to a variety of security attacks (e.g., Shellshock, Heartbleed, Poodle, etc.). For general images – images pushed by docker users, but not explicitly verified by any authority – this number jumps up to ~40% with a sampling error bound of 3%.


HOPE

Functional package management.

$$\texttt{gimp} = f(\texttt{gtk+}, \texttt{gcc}, \texttt{make}, \texttt{coreutils})$$

where $f =$ `./configure && make && make install`

$$\texttt{gimp} = f(\texttt{gtk+}, \texttt{gcc}, \texttt{make}, \texttt{coreutils})$$
$$\texttt{gtk+} = g(\texttt{glib}, \texttt{gcc}, \texttt{make}, \texttt{coreutils})$$

$$\texttt{gimp} = f(\texttt{gtk+}, \texttt{gcc}, \texttt{make}, \texttt{coreutils})$$
$$\texttt{gtk+} = g(\texttt{glib}, \texttt{gcc}, \texttt{make}, \texttt{coreutils})$$
$$\texttt{gcc} = h(\texttt{make}, \texttt{coreutils}, \texttt{gcc}_0)$$
$$\ldots$$

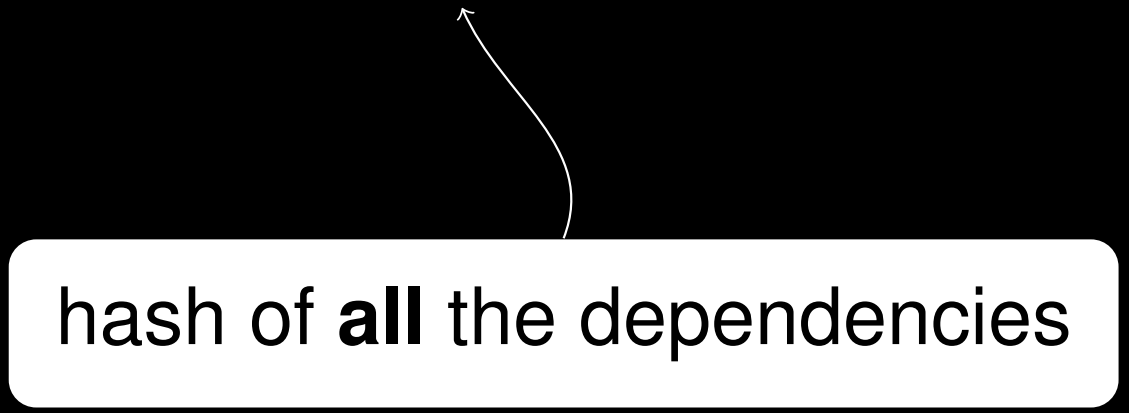
$$\texttt{gimp} = f(\texttt{gtk+}, \texttt{gcc}, \texttt{make}, \texttt{coreutils})$$
$$\texttt{gtk+} = g(\texttt{glib}, \texttt{gcc}, \texttt{make}, \texttt{coreutils})$$
$$\texttt{gcc} = h(\texttt{make}, \texttt{coreutils}, \texttt{gcc}_0)$$
$$\ldots$$

the complete DAG is captured

```
$ guix build hello
```

isolated build: chroot, separate name spaces, etc.

```
$ guix build hello
/gnu/store/ h2g4sf72... -hello-2.10
```



```
$ guix build hello
/gnu/store/ h2g4sf72... -hello-2.10

$ guix gc --references /gnu/store/...-hello-2.10
/gnu/store/...-glibc-2.22
/gnu/store/...-gcc-4.9.3-lib
/gnu/store/...-hello-2.10
```

```
$ guix build hello
/gnu/store/ h2g4sf72... -hello-2.10

$ guix gc --references /gnu/store/...-hello-2.10
/gnu/store/...-glibc-2.22
/gnu/store/...-gcc-4.9.3-lib
/gnu/store/...-hello-2.10
```

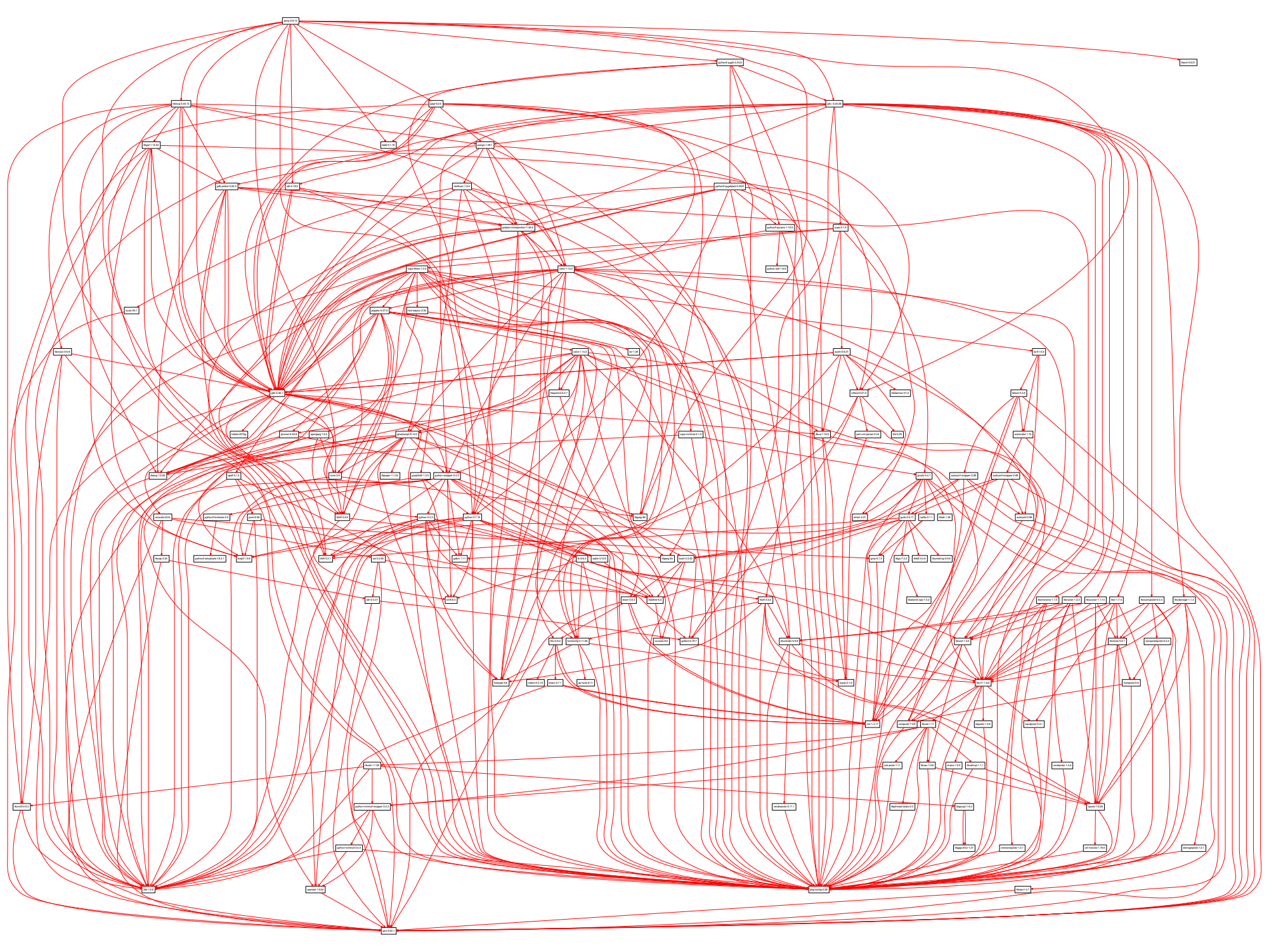
(nearly) bit-identical for everyone

```
$ guix package -i gcc-toolchain coreutils sed grep
…

$ eval `guix package --search-paths`
…

$ guix package --manifest=my-software.scm
…
```

Want to get started hacking on GIMP?

Want to get started hacking on GIMP?

A simple matter of installing the deps, right?

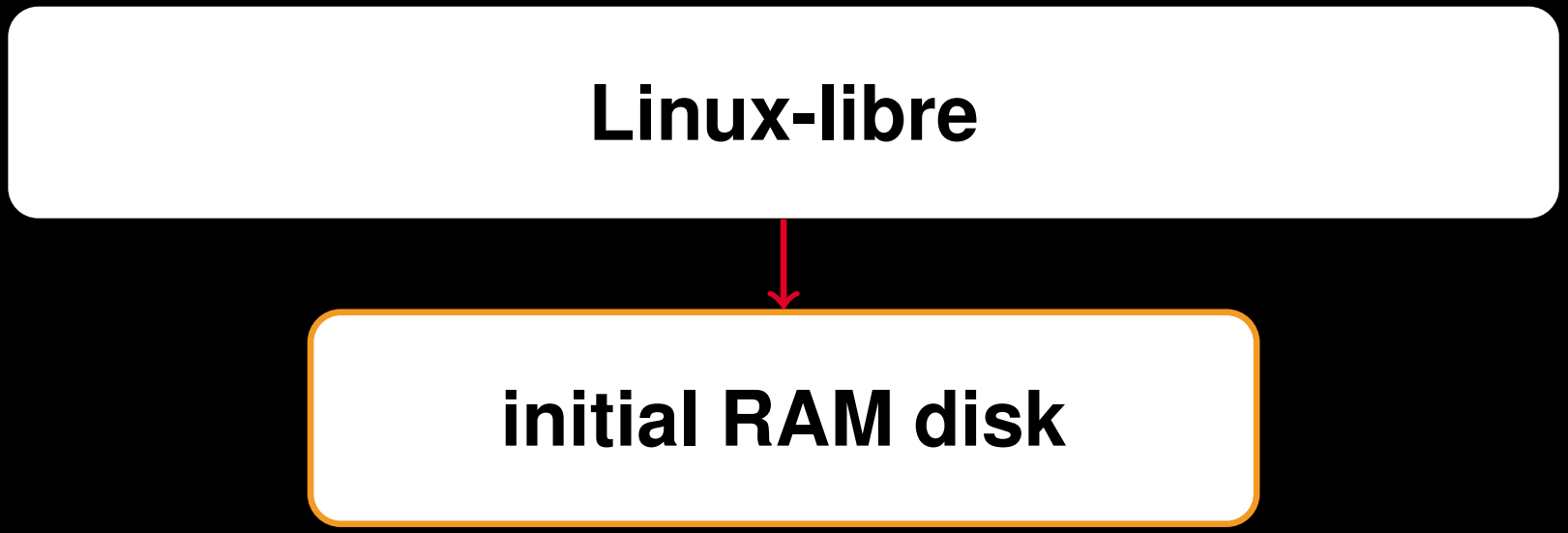

gmp-6.3.0
libxcrypt-4.4.36
gdbm-1.23
libffi-3.4.4
bzip2-1.0.8
ncurses-6.4
expat-2.6.2
libuuid-2.38.1
readline-8.2
sqlite-3.45.3
tk-8.6.13
xz-5.4.6
zlib-1.2.13
libtirpc-1.3.4
libnsl-2.0.1
openssl-3.0.13
ca-certificates-2024.2.2
python-3.12.3
setuptools-68.2.2
pip-23.3.1
wheel-0.41.2
tzdata-2024a
python-build-1.2.1
python-installer-0.7.0
python-wheel-0.43.0
python-flit-core-3.9.0
python-packaging-24.0
python-gpep517-16.0.0
python-setuptools-scm-8.0.4
python-typing_extensions-4.11.0
python-wcwidth-0.2.13
python-prompt_toolkit-3.0.43
python-pygments-2.17.2
python-mypy_extensions-1.0.0
python-pathspec-0.12.1
python-tomli-2.0.1
python-trove-classifiers-2024.3.25
python-hatchling-1.24.2
python-editables-0.5
python-pluggy-1.4.0
python-hatch-vcs-0.4.0
python-hatch-fancy-pypi-readme-24.1.0
python-installer-0.7.0
python-build-1.2.1
python-flit-3.9.0
python-poetry-core-1.9.0
python-poetry-dynamic-versioning-1.3.0
python-tomlkit-0.12.4
python-cleo-2.1.0
python-crashtest-0.4.1
python-dulwich-0.21.7
python-rapidfuzz-3.6.1
python-pkginfo-1.10.0
python-requests-toolbelt-1.0.0
python-requests-2.31.0
python-urllib3-2.0.7
python-certifi-2023.11.17
python-charset-normalizer-3.3.2
python-idna-3.6
python-shellingham-1.5.4
python-tomli_w-1.0.0
python-trove-classifiers-2024.3.25
python-virtualenv-20.25.1
python-distlib-0.3.8
python-filelock-3.13.1
python-platformdirs-4.2.0
python-six-1.16.0
python-webencodings-0.5.1
python-zipp-3.17.0
python-importlib_metadata-7.0.1
python-importlib_resources-6.1.1
python-backports.tarfile-1.0.0
python-jaraco.context-4.3.0
python-jaraco.functools-4.0.0
python-more-itertools-10.2.0
python-keyring-24.3.1
python-secretstorage-3.3.3
python-jeepney-0.8.0
python-cryptography-42.0.5
python-cffi-1.16.0
python-pycparser-2.21
python-build-1.2.1
python-installer-0.7.0
python-flit_core-3.9.0
python-poetry-core-1.9.0
python-pyproject_hooks-1.0.0
python-tomli-2.0.1

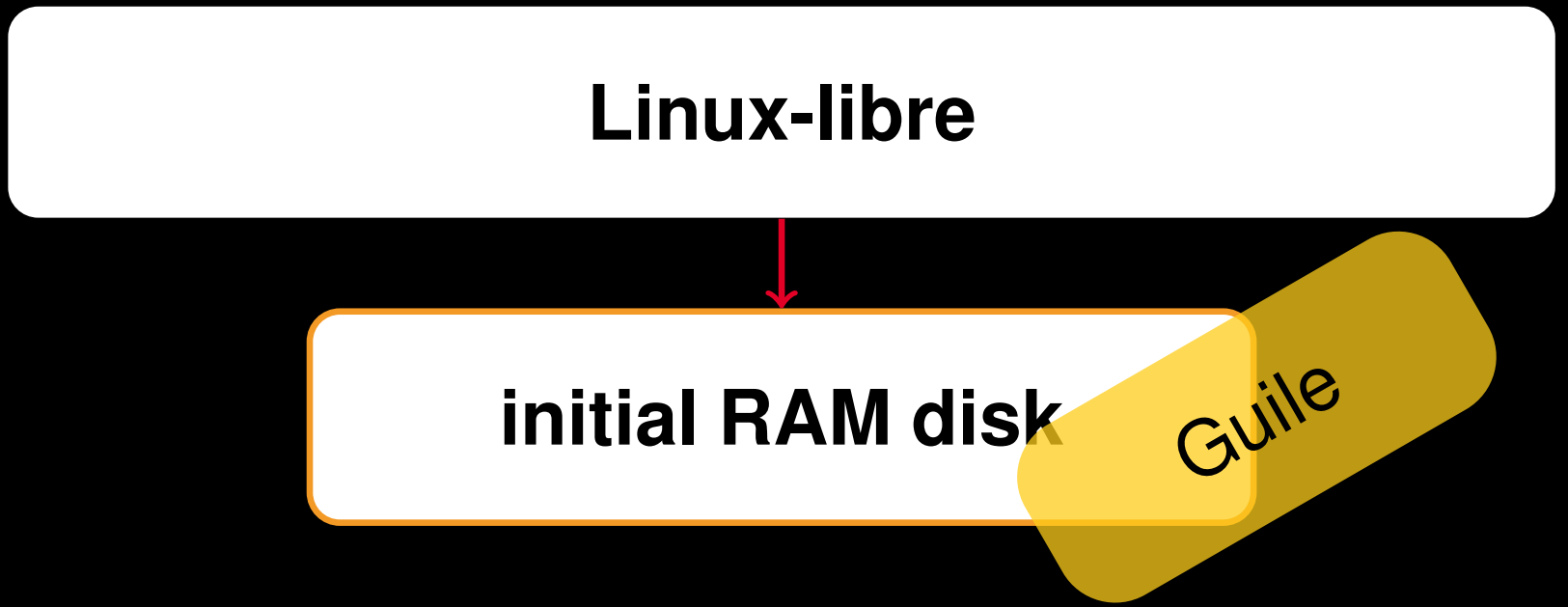
```
$ guix environment --container gimp
…

$ guix environment --container gimp \
     --ad-hoc git autoconf automake gdb
…
```

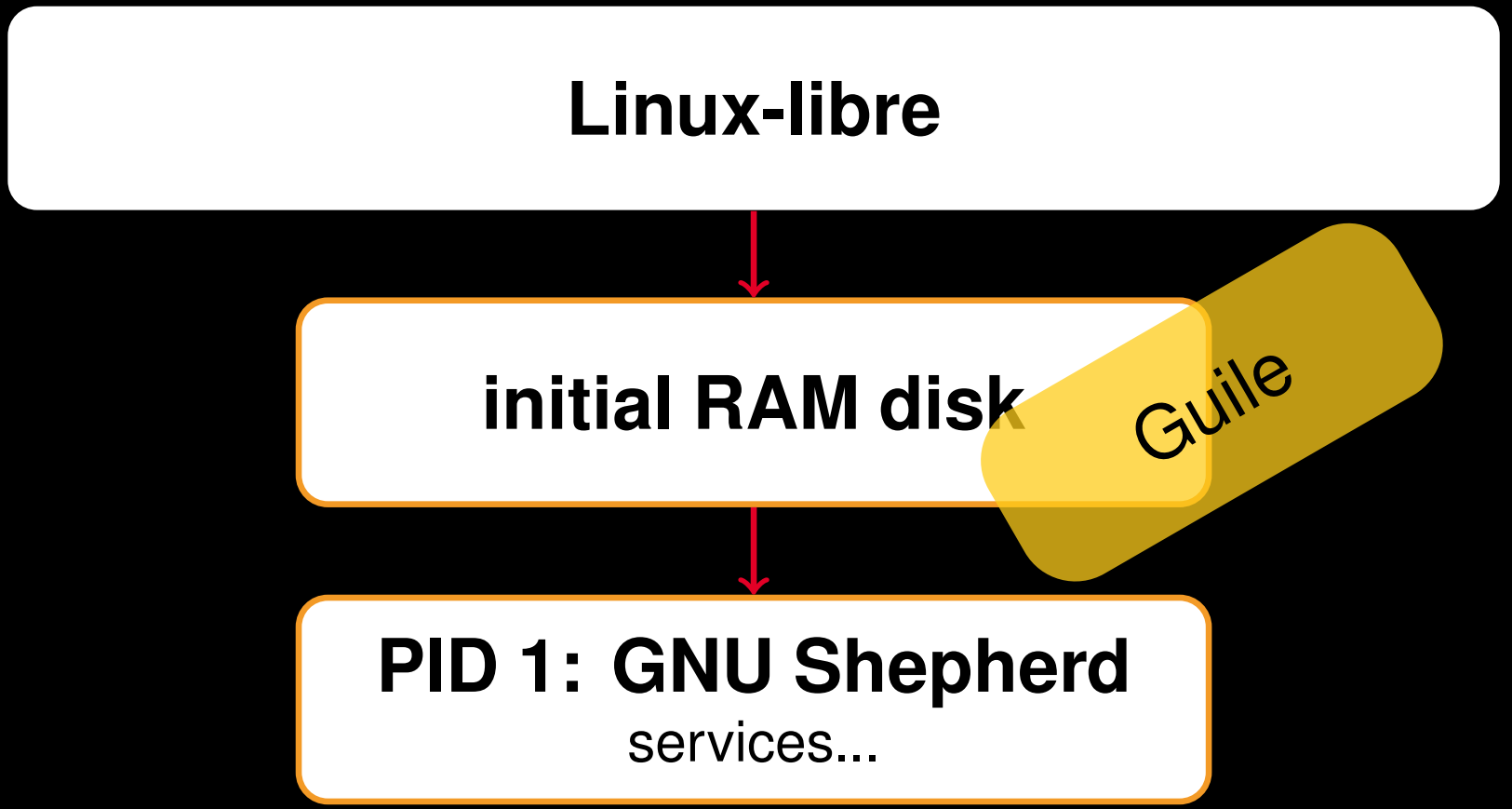
Whole-system deployment.

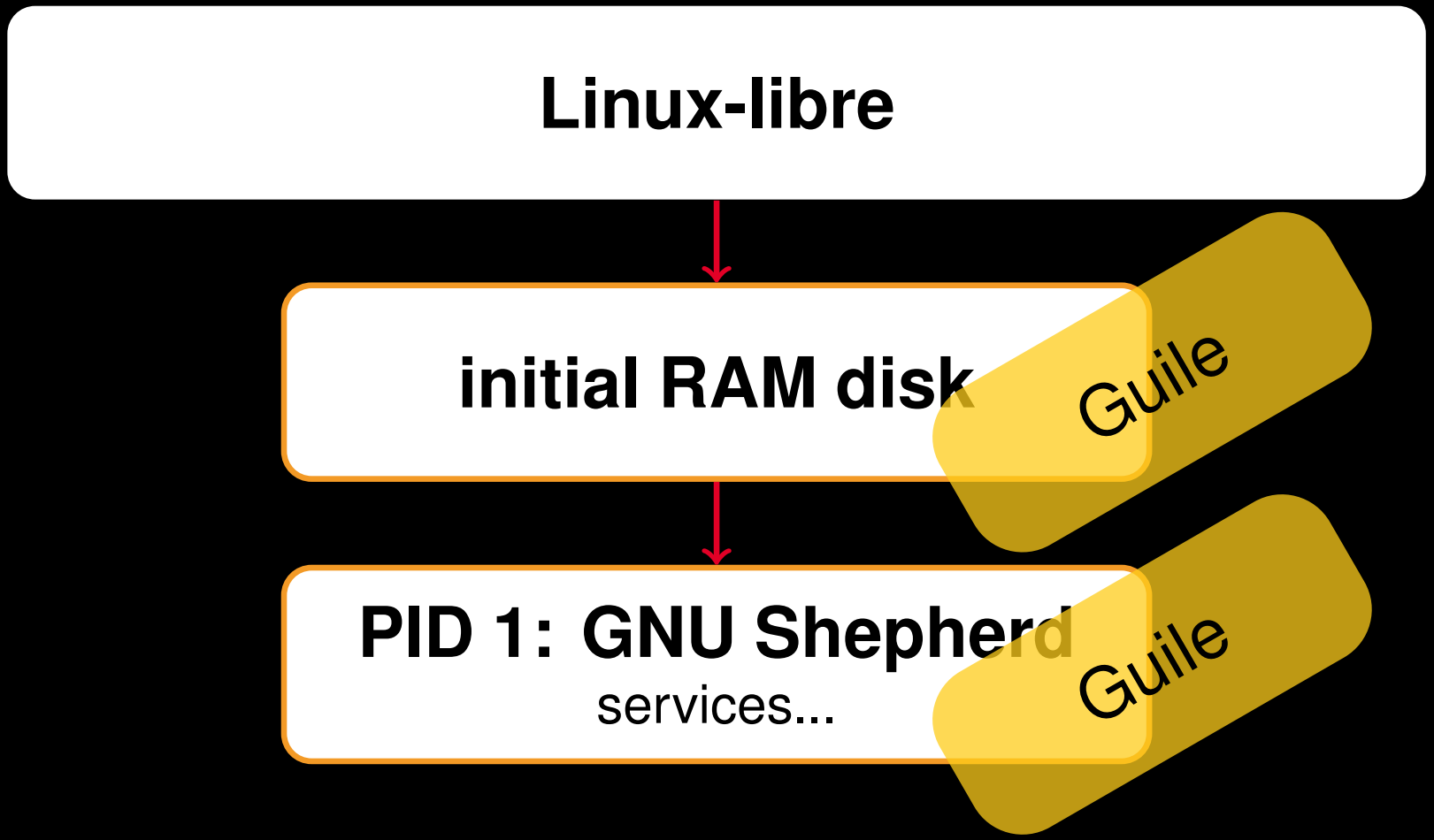
GuixSD: declarative OS config

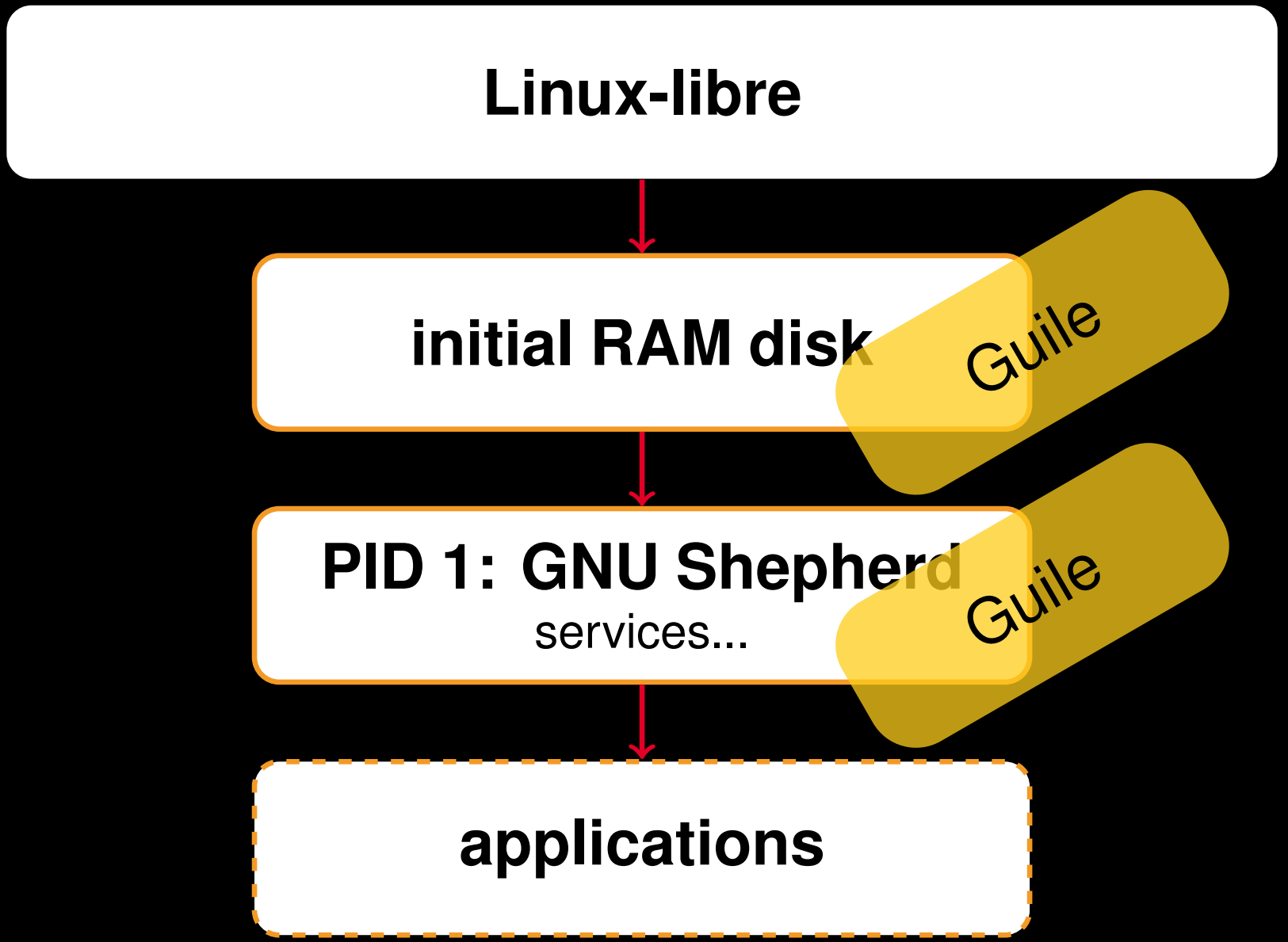
Linux-libre


Linux-libre
initial RAM disk


Linux-libre
initial RAM disk
Guile


Linux-libre
initial RAM disk
Guile
PID 1: GNU Shepherd
services...


Linux-libre
initial RAM disk
Guile
PID 1: GNU Shepherd
services...
Guile


Linux-libre
initial RAM disk
Guile
PID 1: GNU Shepherd
services...
Guile
applications

Trustworthiness.

Debian's dirtiest secret:
Binary packages built by developers are used in the archive

— Lucas Nussbaum, FOSDEM 2015

binary/source deployment

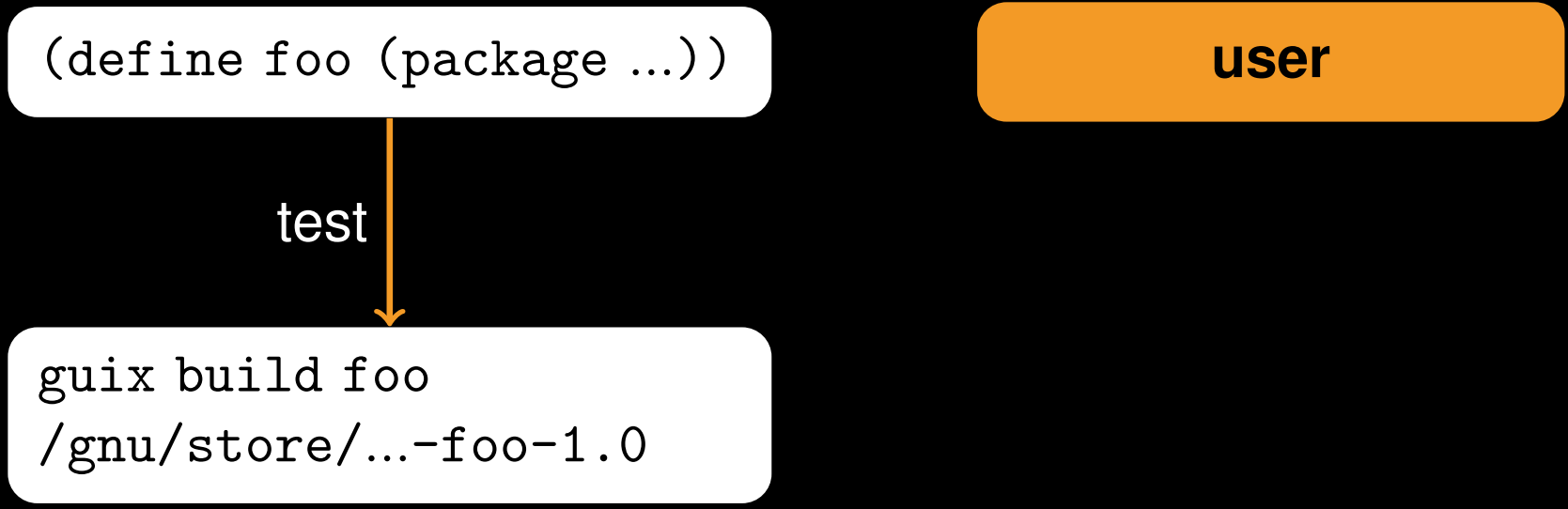
```
alice@foo$ guix package --install=emacs
The following package will be installed:
   emacs-24.5 /gnu/store/...-emacs-24.5

The following files will be downloaded:
   /gnu/store/...-emacs-24.5
   /gnu/store/...-libxpm-3.5.10
   /gnu/store/...-libxext-1.3.1
   /gnu/store/...-libxaw-1.0.11
```

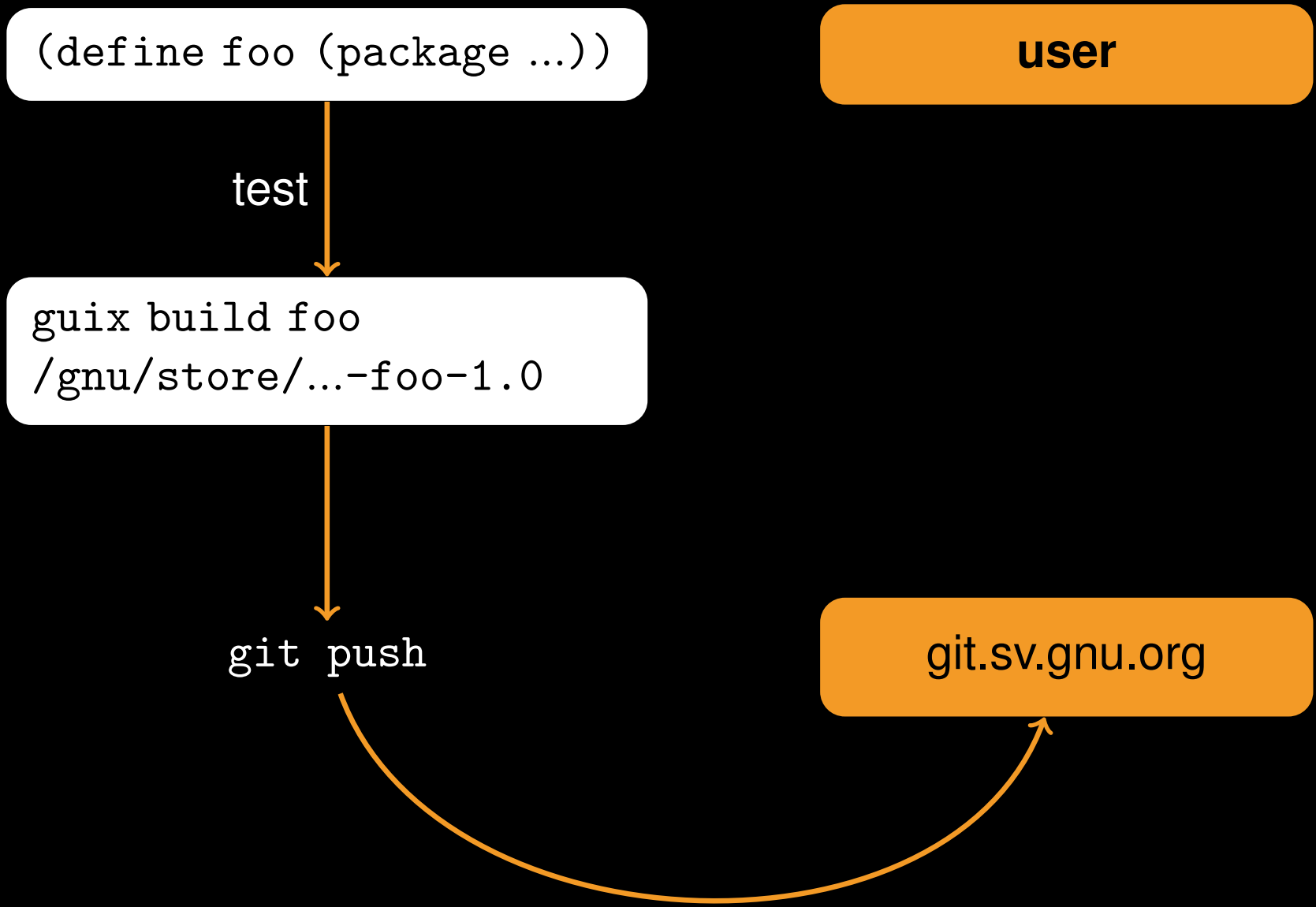
binary/source deployment

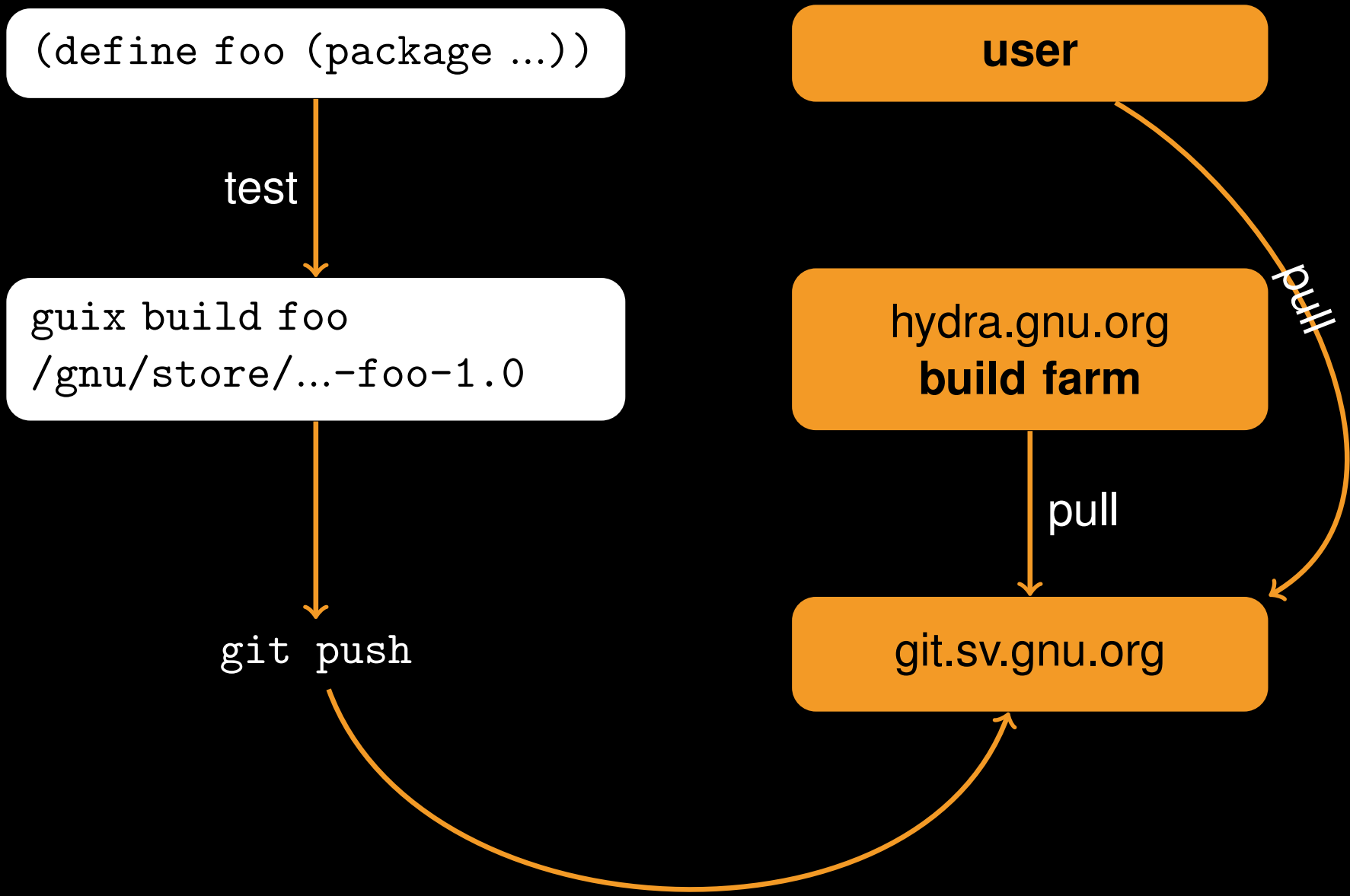
```
alice@foo$ guix package --install=emacs
The following package will be installed:
   emacs-24.5 /gnu/store/...-emacs-24.5

The following files will be downloaded:
   /gnu/store/...-libxext-1.3.1
   /gnu/store/...-libxaw-1.0.11
The following derivations will be built:
   /gnu/store/...-emacs-24.5.drv
   /gnu/store/...-libxpm-3.5.10.drv
```

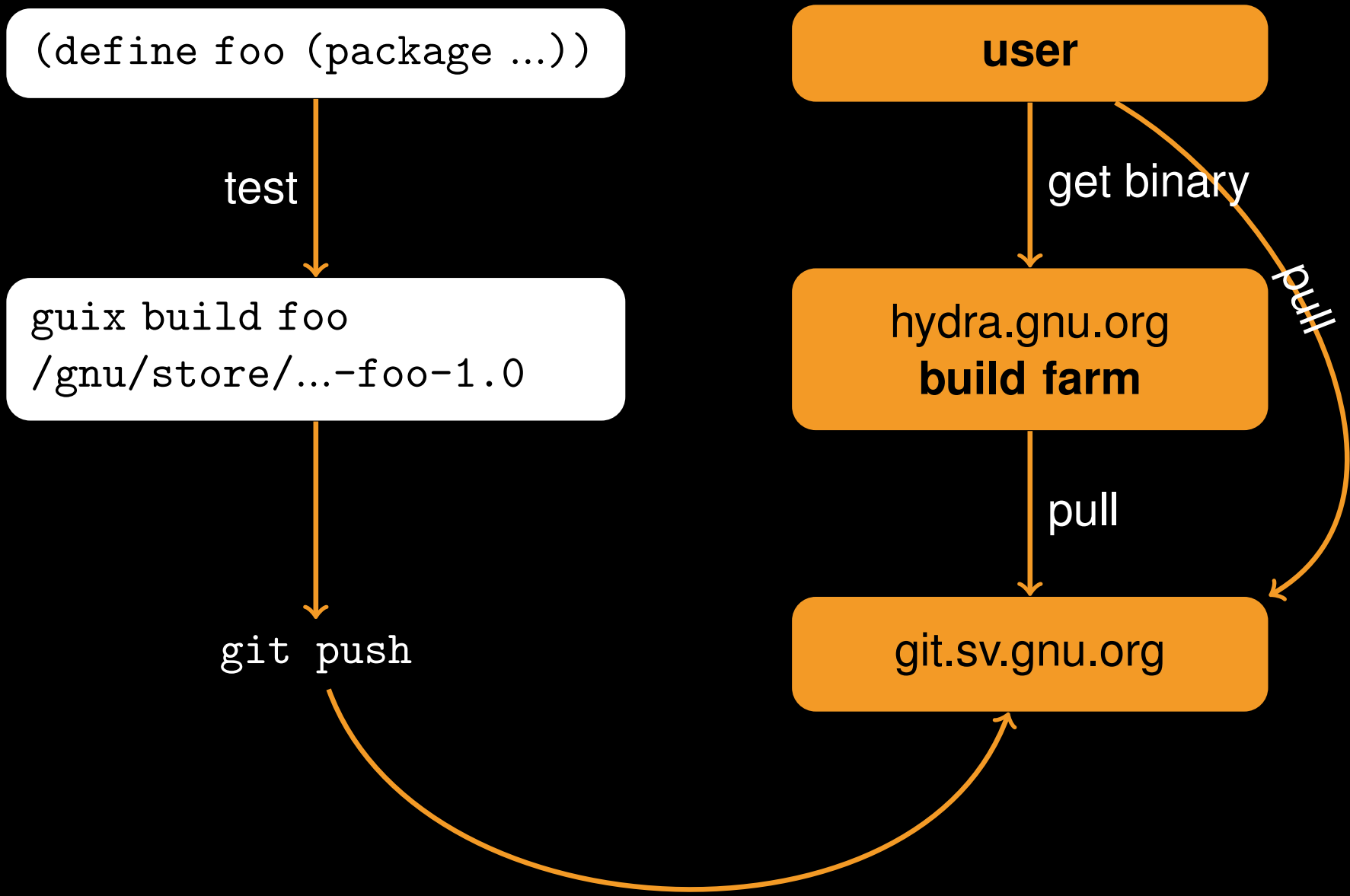
```
(define foo (package ...))
```

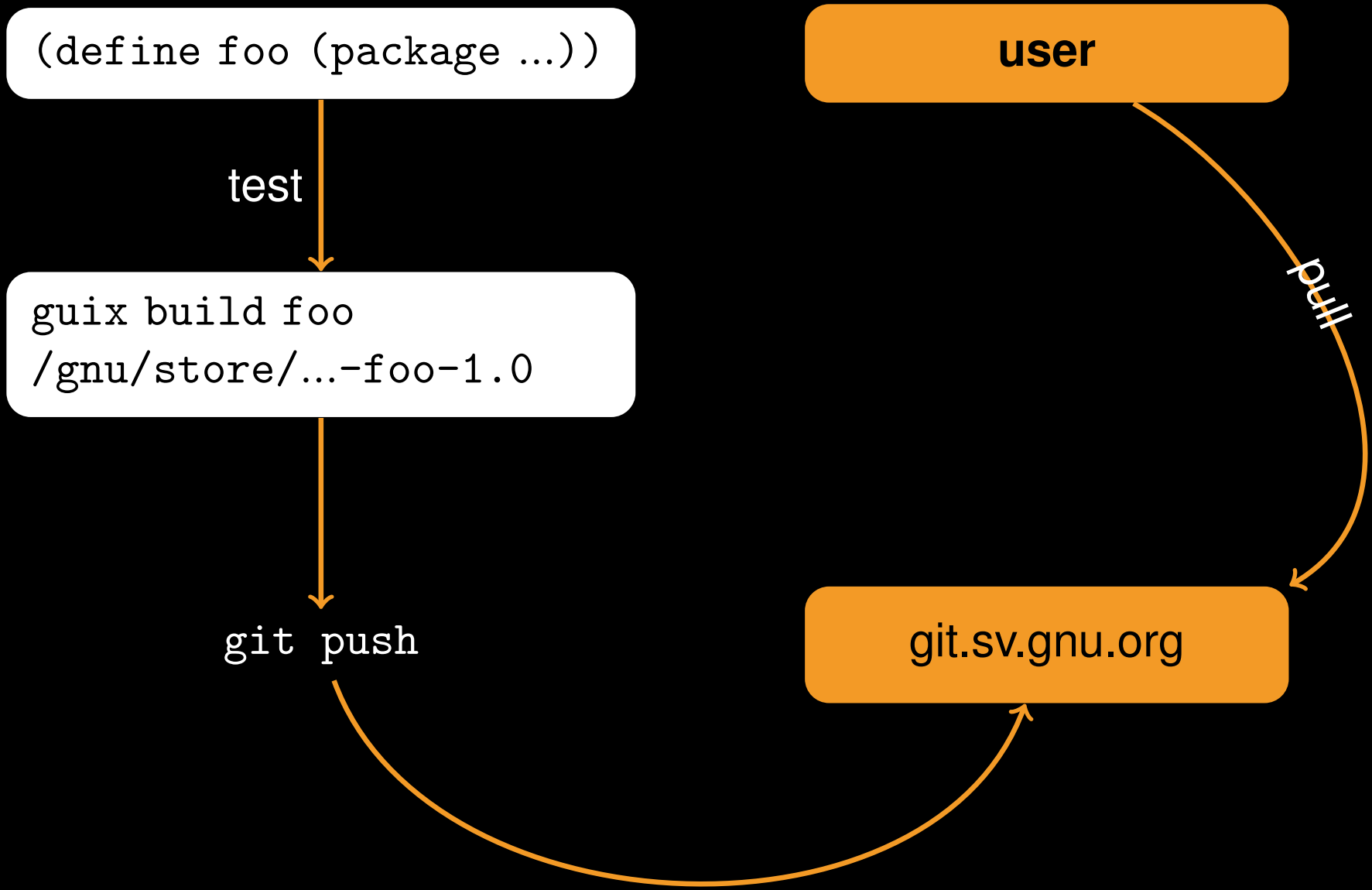
user

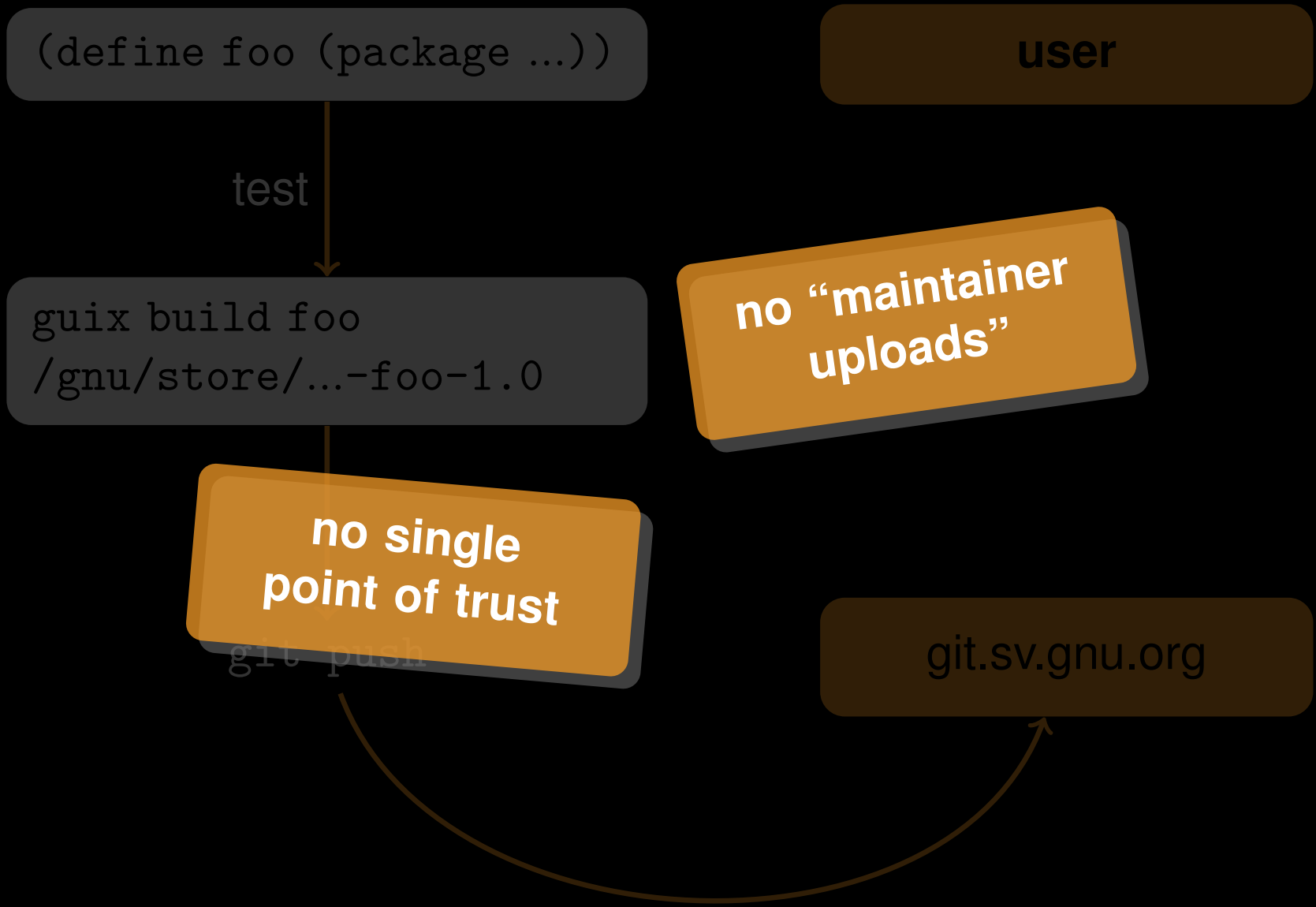

(define foo (package ...))
user
test
guix build foo
/gnu/store/...-foo-1.0

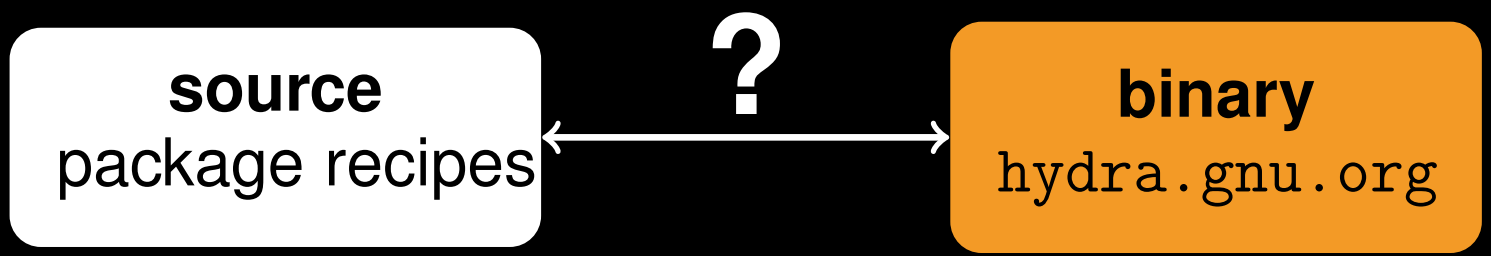

(define foo (package ...))
user
test
guix build foo
/gnu/store/...-foo-1.0
git push
git.sv.gnu.org


(define foo (package ...))
test
guix build foo
/gnu/store/...-foo-1.0
git push
user
pull
hydra.gnu.org
build farm
pull
git.sv.gnu.org


(define foo (package ...))
test
guix build foo
/gnu/store/...-foo-1.0
git push
user
get binary
pull
hydra.gnu.org
build farm
pull
git.sv.gnu.org


(define foo (package ...))
test
guix build foo
/gnu/store/...-foo-1.0
git push
user
pull
git.sv.gnu.org


(define foo (package ...))
user
test
guix build foo
/gnu/store/...-foo-1.0
no “maintainer uploads”
no single point of trust
git push
git.sv.gnu.org




```
(define emacs (package ...))    /gnu/store/...-emacs-24.5
```

towards greater user control

1. Bit-reproducible builds
2. No single binary provider
3. Tools for users to challenge binaries

towards greater user control

1. **Bit-reproducible builds**
   - we have **isolated build environments**!
   - ... but we need builds to be **deterministic**
   - `http://reproducible-builds.org`
2. **No single binary provider**

3. **Tools for users to challenge binaries**

towards greater user control

1. **Bit-reproducible builds**
   - we have **isolated build environments**!
   - ... but we need builds to be **deterministic**
   - `http://reproducible-builds.org`
2. **No single binary provider**
   - `guix publish`
   - P2P publishing over GNUnet? (GSoC 2015)
3. **Tools for users to challenge binaries**

towards greater user control

1. **Bit-reproducible builds**
   - we have **isolated build environments**!
   - ... but we need builds to be **deterministic**
   - `http://reproducible-builds.org`
2. **No single binary provider**
   - `guix publish`
   - P2P publishing over GNUnet? (GSoC 2015)
3. **Tools for users to challenge binaries**

```
$ guix challenge --substitute-urls="http://hydra.gnu.org ht
/gnu/store/...-openssl-1.0.2d contents differ:
  local hash: 0725l22...
  http://hydra.gnu.org/...-openssl-1.0.2d: 0725l22...
  http://guix.example.org/...-openssl-1.0.2d: 1zy4fma...
/gnu/store/...-git-2.5.0 contents differ:
  local hash: 00p3bmr...
  http://hydra.gnu.org/...-git-2.5.0: 069nb85...
  http://guix.example.org/...-git-2.5.0: 0mdqa9w...
/gnu/store/...-pius-2.1.1 contents differ:
  local hash: 0k4v3m9...
  http://hydra.gnu.org/...-pius-2.1.1: 0k4v3m9...
  http://guix.example.org/...-pius-2.1.1: 1cy25x1...
```

Status.

timeline

- Nov. 2012 — dubbed GNU
- Jan. 2013 — **0.1**
- ...
- Apr. 2014 — **0.6**, signed binaries, `guix system`
- July 2014 — **0.7**, **installable operating system**
- ...
- 29 Jan. 2015 — **0.8.1**, **ARMv7 port**
- ...
- 5 Nov. 2015 — **0.9.0**, new service framework, etc.
- Jan. 2016 — successful **fundraiser** for new **build farm**


better
now

status

- full-featured package manager
- 3,000+ packages, 4 platforms
- **Guix System Distribution**$^{\beta}$
- binaries at `http://hydra.gnu.org`
- tooling: auto-update, “linting”, etc.

- ≈25 contributors per month
- ... and lots of friendly people!
- ≈400 commits per month
- 200–500 new packages per release

your help needed!

- **install the distribution**
- **use it**, report bugs, add packages
- help with the **infrastructure** + admin
- **donate** hardware/money
- share your **ideas**!

GuixSD

ludo@gnu.org

http://gnu.org/software/guix/

Copyright © 2010, 2012–2016 Ludovic Courtès ludo@gnu.org.

GNU GuixSD logo, CC-BY-SA 4.0, http://gnu.org/s/guix/graphics
Copyright of other images included in this document is held by their respective owners.

This work is licensed under the **Creative Commons Attribution-Share Alike 3.0** License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

At your option, you may instead copy, distribute and/or modify this document under the terms of the **GNU Free Documentation License, Version 1.3 or any later version** published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is available at http://www.gnu.org/licenses/gfdl.html.

The source of this document is available from http://git.sv.gnu.org/cgit/guix/maintenance.git.