functional package management with GNU Guix for developers & power users
Package managers are really useful.
Package managers are really useful. But they can be so frustrating!
Old software
Version conflicts
Suspenseful upgrades
The only way is forward
Changes affect all users
because package managers are imperative & stateful
big binary bundles?
big binary bundles are indistinguishable from giving up
Functional packaging

headers
sources
build tools
libraries

\[ f(x) \]

cabba9e-\texttt{emacs}-24.5/
  \mid \texttt{bin}
  \mid \texttt{lib}
  \mid ...
Functional packaging

Same inputs?  
Same output!

Different inputs?  
Different outputs.

cabba9e-emacs-24.5/
  bin
  Emacs
  lib
  ...

dedbeef-emacs-24.5/
  bin
  Emacs
  lib
  ...

Different inputs?  
Different outputs.
GNU Guix

Guix client
(guix packages)
(guix store)

RPCs

Build daemon

Isolated build processes
chroot with declared inputs

Store
(define glpk
  (package
   (name "glpk")
   (version "4.57")
   (source
    (origin
     (method url-fetch)
     (uri (string-append "mirror://gnu/glpk/glpk-
                      version ".tar.gz"))
     (sha256 (base32 "0p17jj1lxxd2...")
     )
    (build-system gnu-build-system)
    (inputs
     `(("gmp" ,gmp)))
    (arguments
     '#:configure-flags '("--with-gmp")
    (home-page "http://www.gnu.org/software/glpk/")
    (synopsis "GNU Linear Programming Kit")
    (description
     "GLPK is a C library for solving large-scale linear
      programming (LP), mixed integer programming (MIP), and
      other related problems ...")
    (license gpl3+)))
  )
Learn more!

#guix on irc.freenode.net
http://gnu.org/s/guix

rekado@elephly.net