

THE NIX/HYDRA SYSTEM AND HEP-NIX-OVERLAY

IAN-WOO KIM (PH-TH)
IAN-WOO.KIM@CERN.CH

PH-SFT GROUP MEETING
CERN, 22 SEP 2014

NIX PACKAGE MANAGER



Nix logo (nixos.org)



Eelco Dolstra

- Created by Eelco Dolstra
- Started in 2004

Nix: A Safe and Policy-Free System for Software Deployment

Eelco Dolstra, Merijn de Jonge, and Eelco Visser – Utrecht University

- Ph.D. Thesis, 2006

The Purely Functional Software Deployment Model

NIX PACKAGE MANAGER



Nix logo (nixos.org)



Eelco Dolstra

- Created by Eelco Dolstra
- Started in 2004

Nix: A Safe and Policy-Free System for Software Deployment

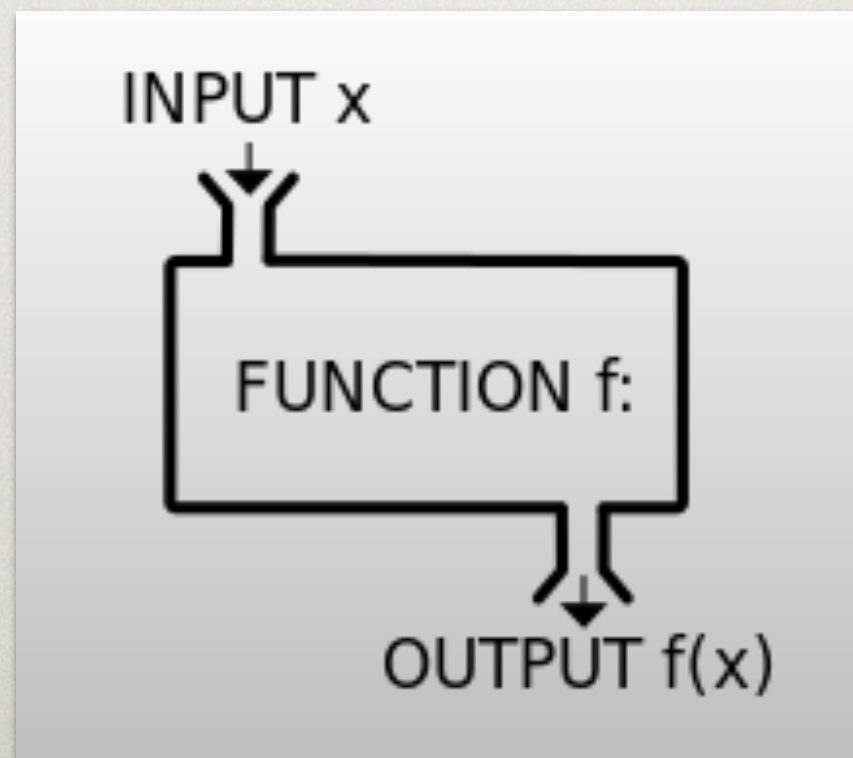
Eelco Dolstra, Merijn de Jonge, and Eelco Visser – Utrecht University

- Ph.D. Thesis, 2006

The Purely Functional Software Deployment Model

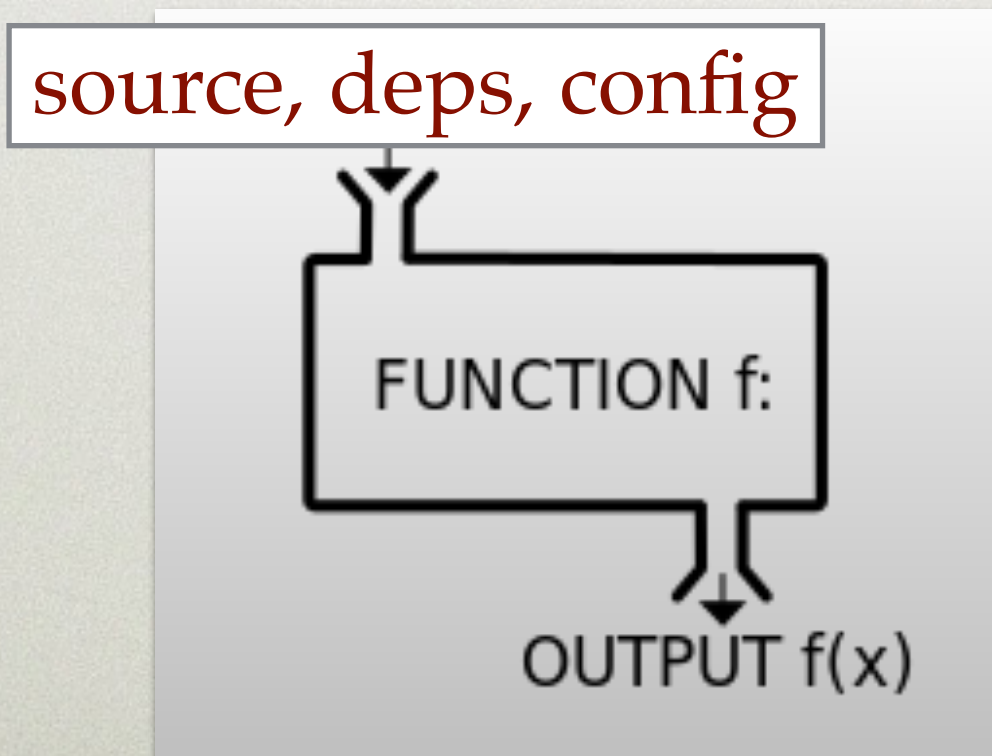
PURELY FUNCTIONAL?

- A pure function = a function in mathematical sense.
 - The same inputs give the same outputs.



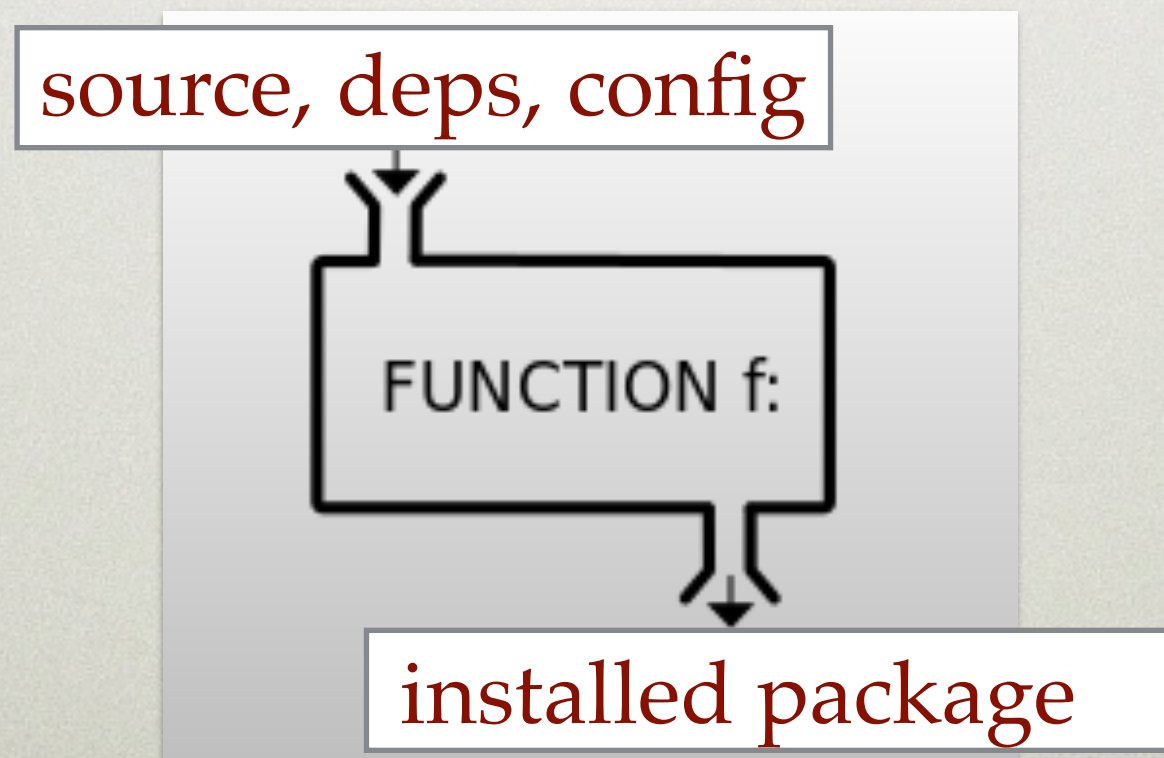
PURELY FUNCTIONAL?

- A pure function = a function in mathematical sense.
 - The same inputs give the same outputs.



PURELY FUNCTIONAL?

- A pure function = a function in mathematical sense.
 - The same inputs give the same outputs.

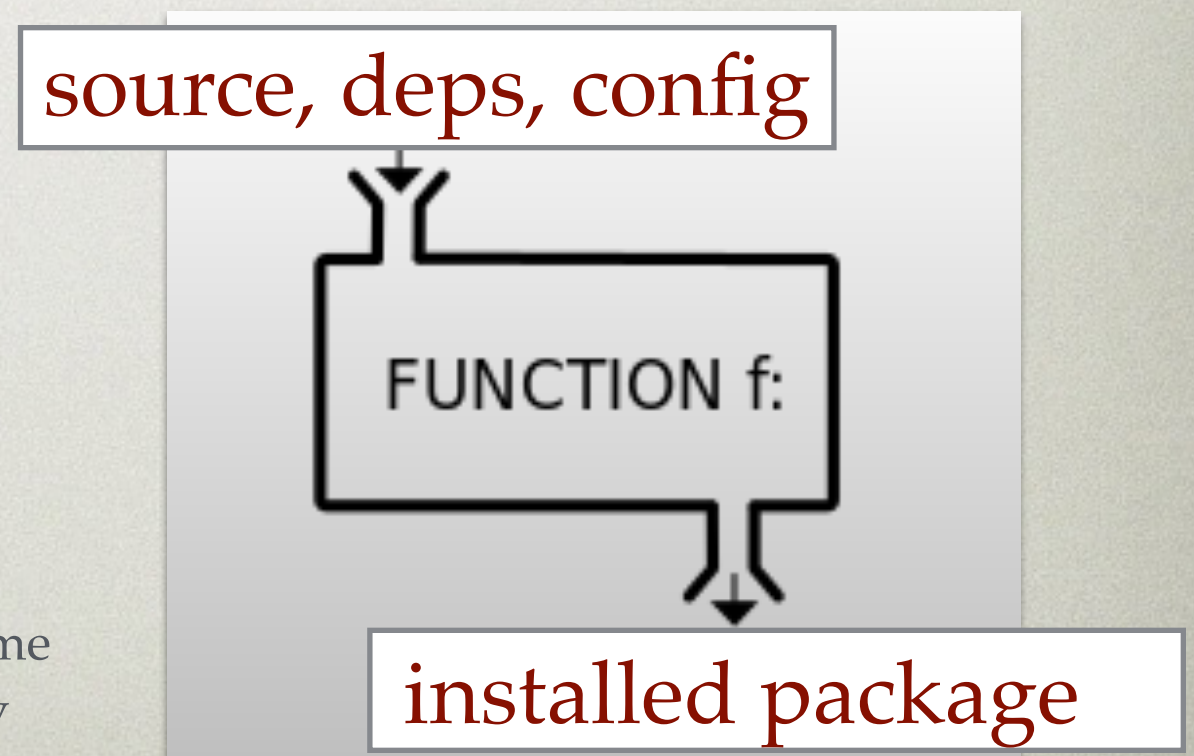


PURELY FUNCTIONAL!

- Dream of build / package managers.
 - Reproducibility guaranteed
 - Can identify build problems
 - Share across systems

Many package managers try to achieve this by fixing *the same build environment*.

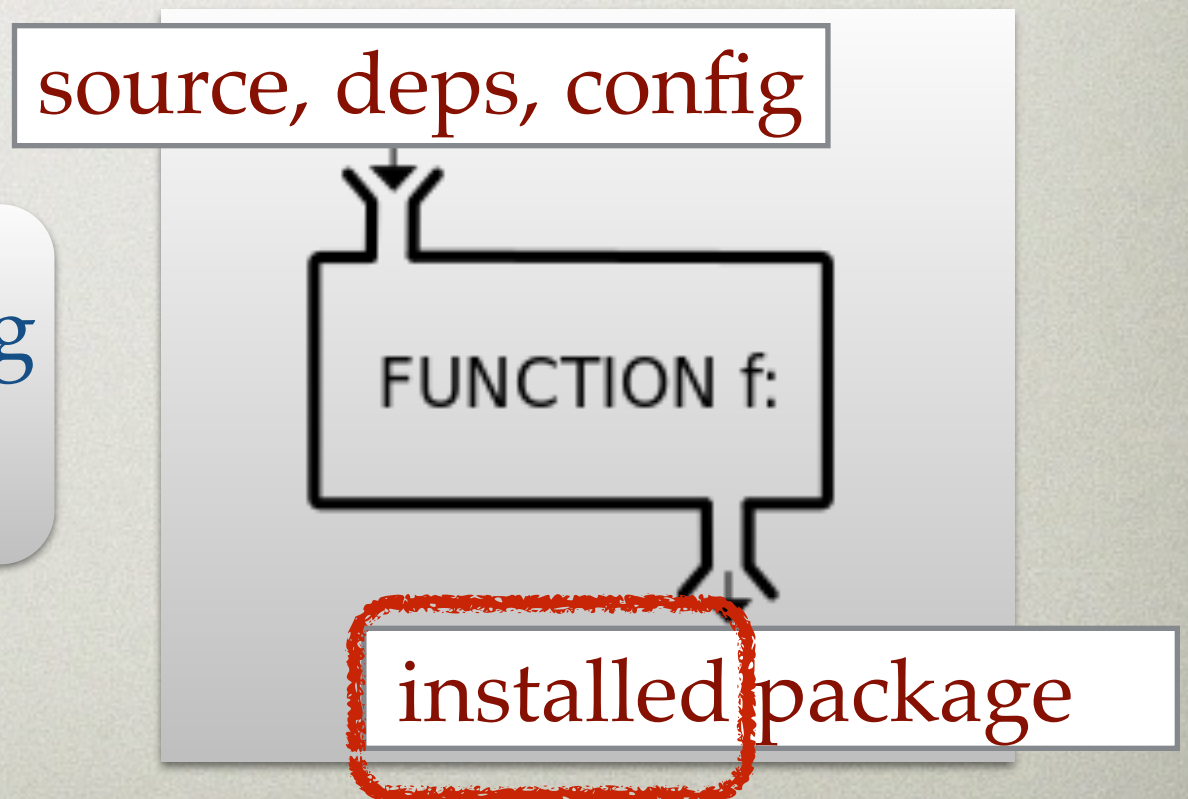
The resultant package file can be the same but the resultant system still differs by package installation history...



PURELY FUNCTIONAL!

- Dream of build / package managers.
 - Reproducibility guaranteed
 - Can identify build problems
 - Share across systems

Nix achieve this by **NOT** allowing
to overwrite previous installation.



EVERYTHING IS HASHED

- Nix packages are installed in nix store with hash. ex) GNU hello

/nix/store/whi5gpcqxx29dg89ffjdi0dg9rb5i0s2-hello-2.9

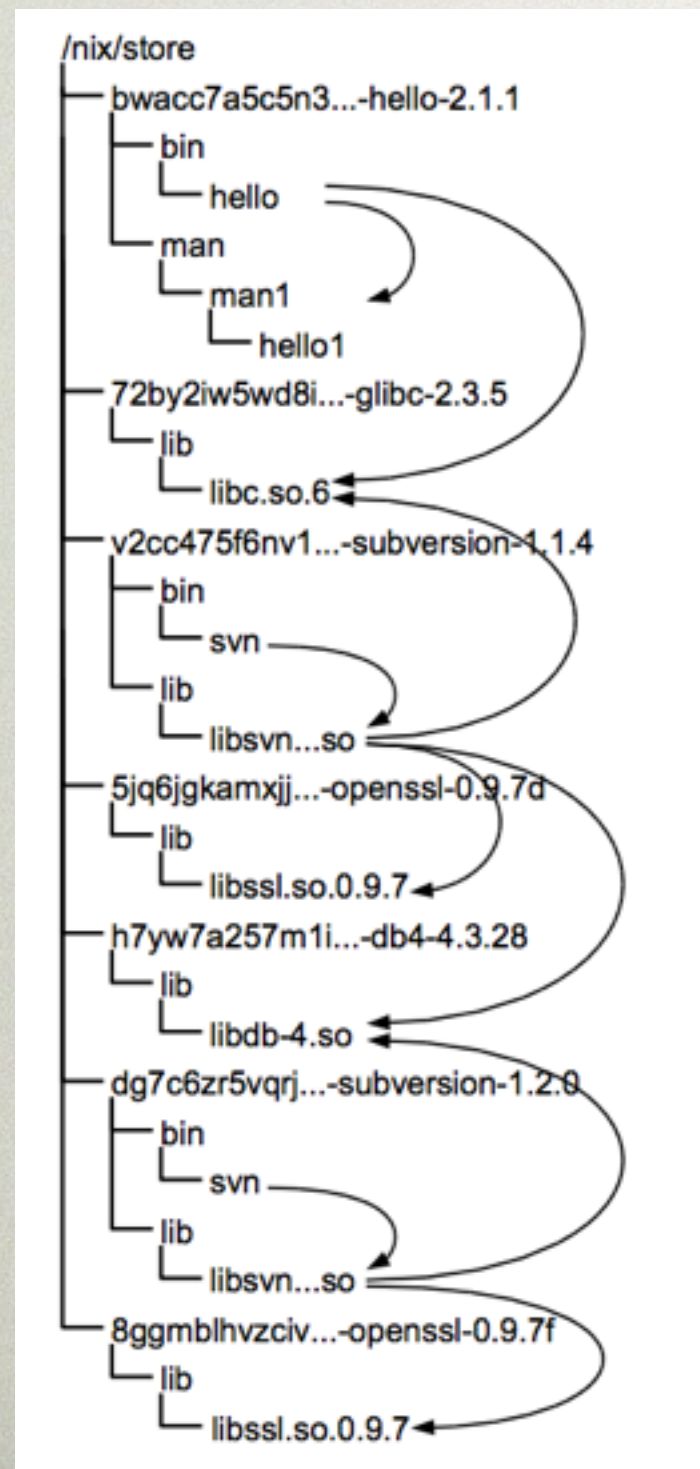
Nix store
location

Hash

Package name,
version, flavor

- The same packages with different version / dependency / configuration can coexist.
- User's profile is a selection of packages aggregated into a common prefix.

EVERYTHING IS HASHED



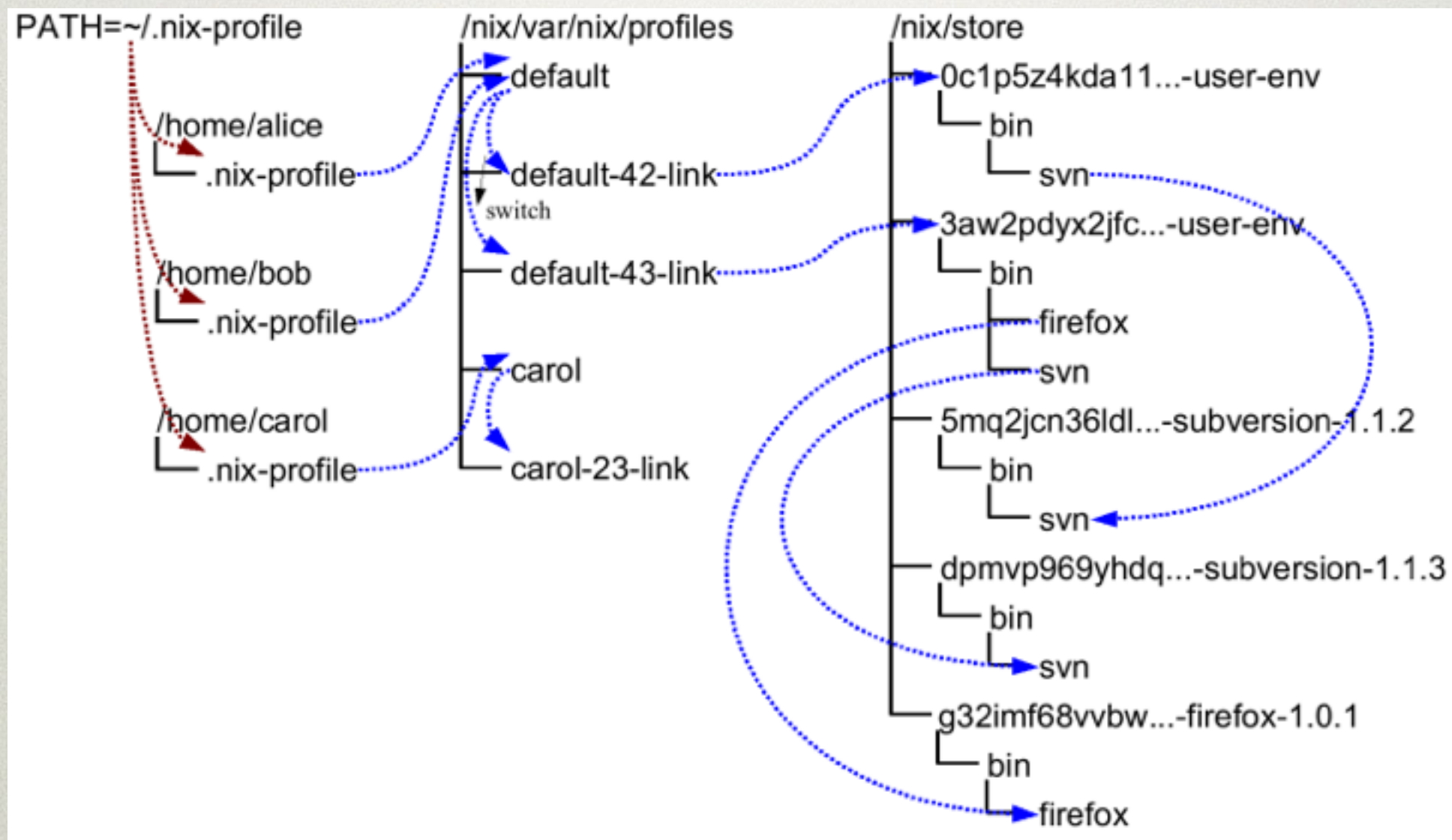
- Dependencies are explicitly specified with hashes.
 - Explicit filepath, use RPATH for DLL
- Share packages with the same hash

WELL-DEFINED CLOSURE



- Subset of nix store that is closed under the dependency graph
 - Everything needed is self-contained.
- This can be even distributed to other machines with Nix.

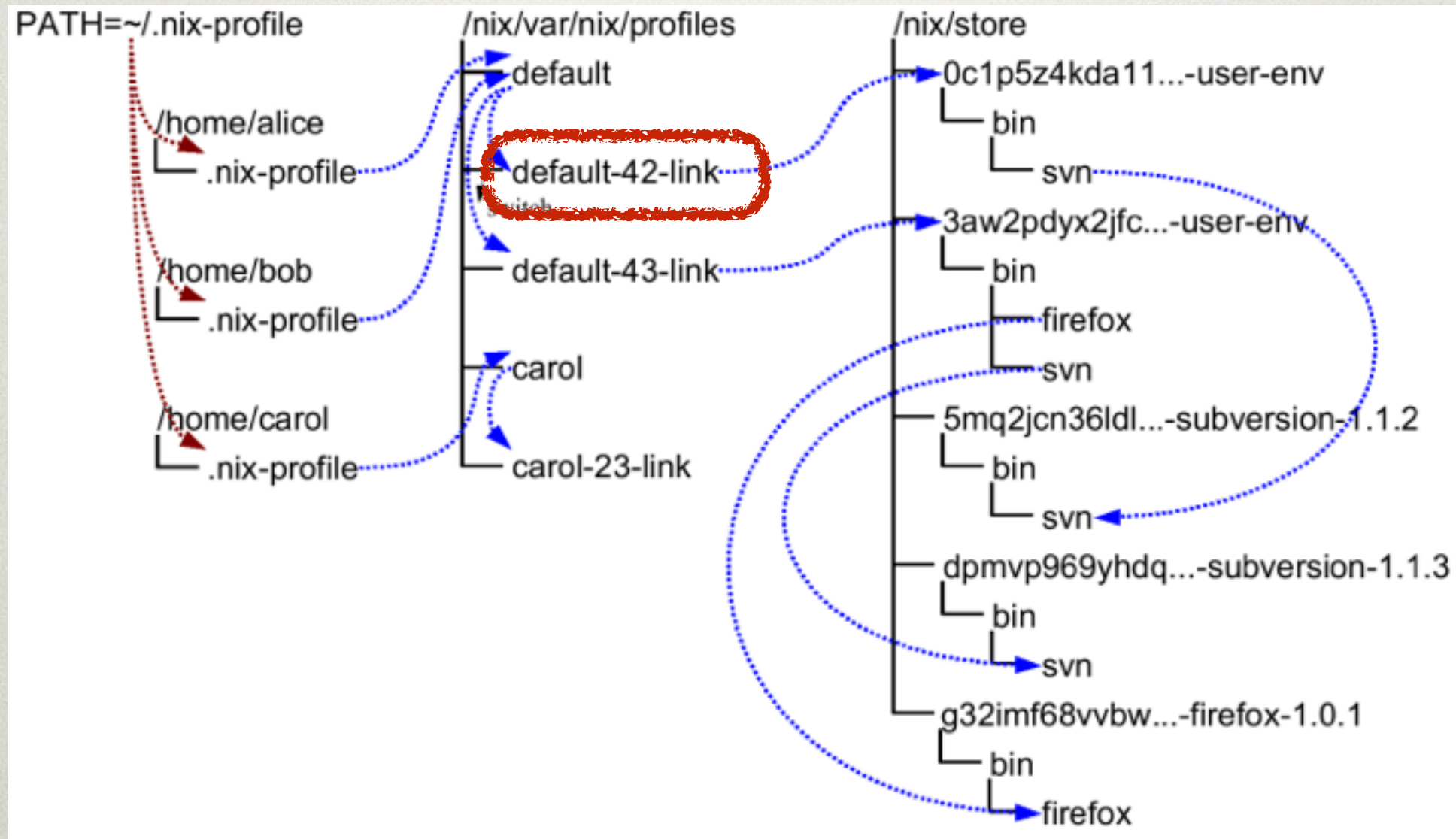
USER PROFILE IS SYMLINK FARM



- Users / Profiles / Generations
- No `/usr`, `/usr/local`, `/opt/local`!

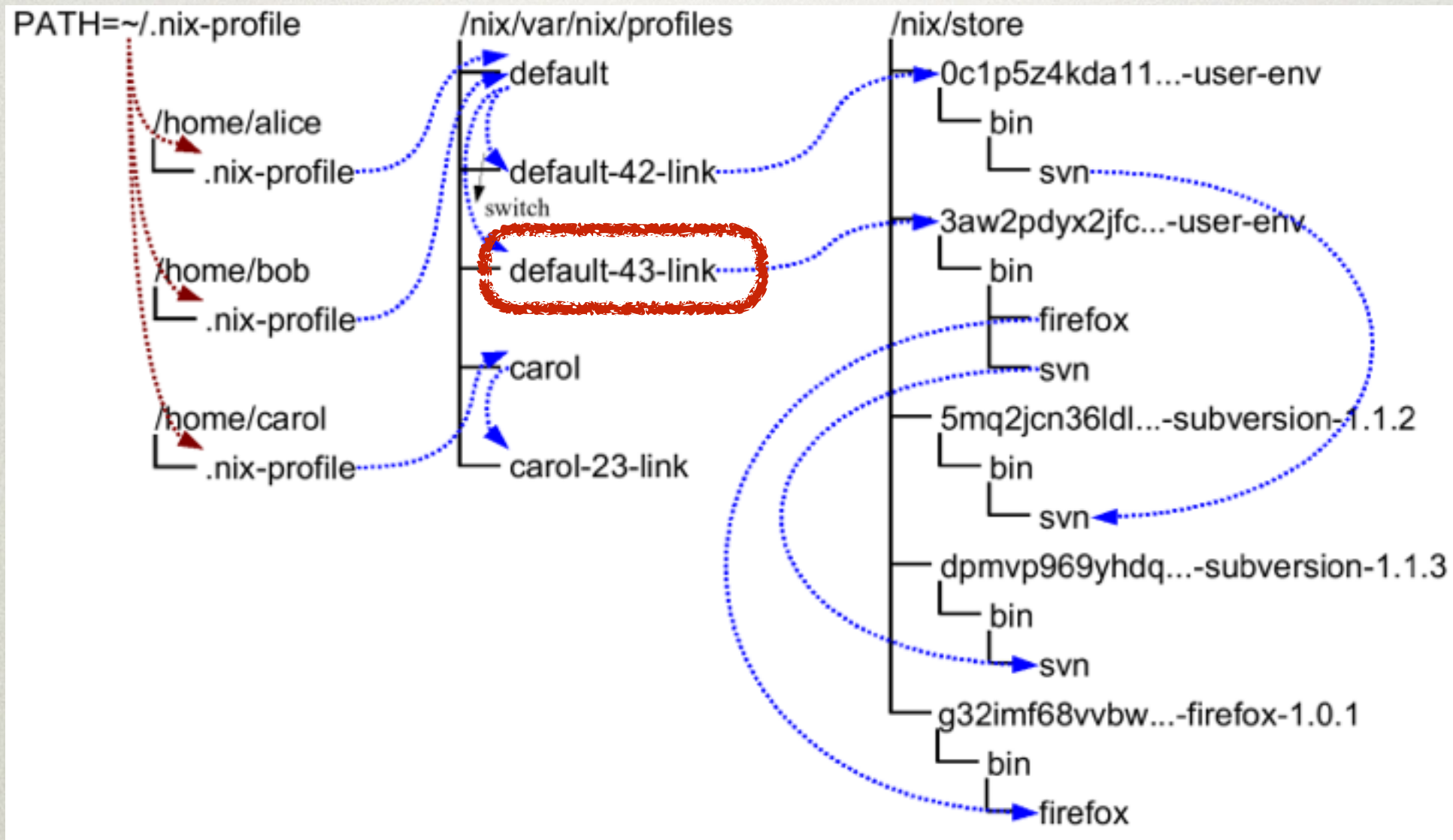
*Escape from FHS
for good!*

USER PROFILE IS SYMLINK FARM



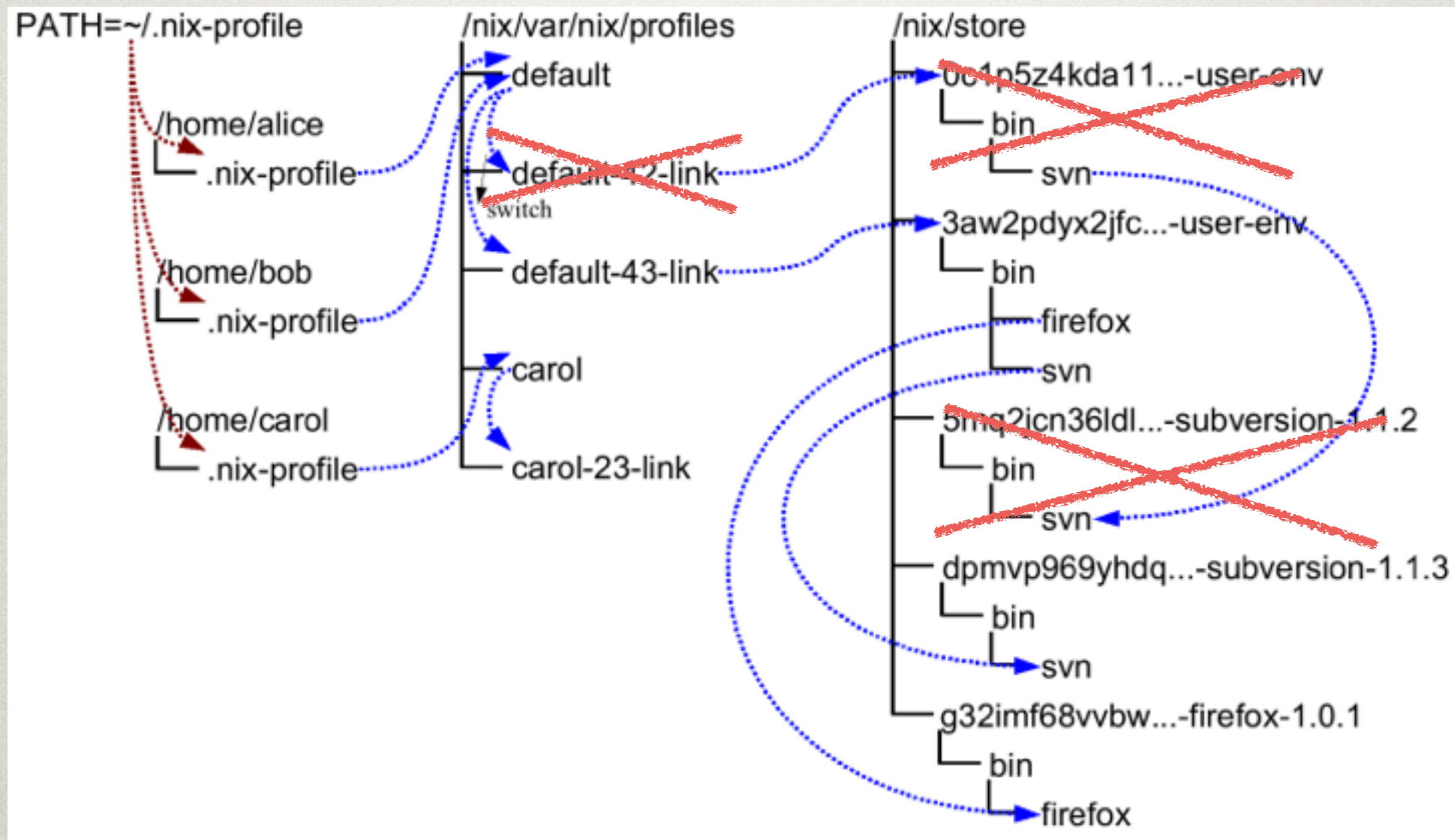
- There are no upgrade. Old versions coexist!
- Switch/rollback generation, profile

USER PROFILE IS SYMLINK FARM



- There are no upgrade. Old versions coexist!
- Switch/rollback generation, profile

USER PROFILE IS SYMLINK FARM



- Garbage collection!

NIX EXPRESSION LANGUAGE

- Language for describing a package in purely functional package manager
- Describes package source, dependencies and build/install steps.

```
{stdenv, fetchurl, perl}: 2

stdenv.mkDerivation { 3
  name = "hello-2.1.1"; 4
  builder = ./builder.sh; 5
  src = fetchurl { 6
    url = http://ftp.gnu.org/pub/gnu/hello/hello-2.1.1.tar.gz;
    md5 = "70c9ccf9fac07f762c24f2df2290784d";
  };
  inherit perl; 7
}
```

Note that this is a function!

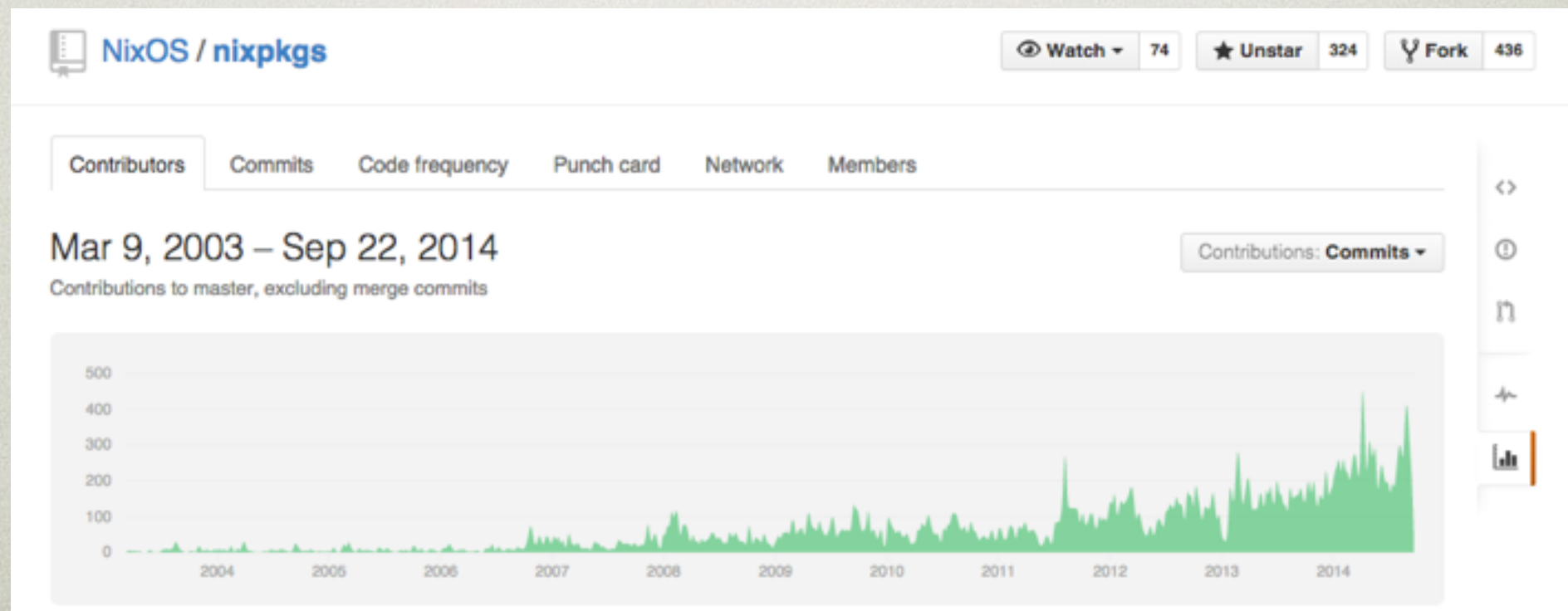
DERIVATION

- Intermediate rep to build a Nix package
 - Fixed schema in terms of Nix expression primitives
 - A Nix expression for a package generates a derivation
 - Every argument is applied and replaced by actual hash.
 - Hash for a package is calculated from dep derivations.
 - Packages in Nix store are finally derived from derivations

```
Derive([("out", "/nix/store/v5af71yn6jrzig3m53r4hqbbpmjqjqqz7-hello-2.9", "", "")]
, [("/nix/store/2k85dpw0b2pmyqbdrdlmb8f510l85b4b-bash-4.2-p47.drv", ["out"])
, ("/nix/store/7h9jks2glccchjwcxv6hdx9lgvyw486-stdenv.drv", ["out"])
, ("/nix/store/daznf9d4hcb4ynz94xfd6r8gkigysrpy-hello-2.9.tar.gz.drv", ["out"])]
, ["/nix/store/9krlzvny65gdc8s7kpb6lkx8cd02c25b-default-builder.sh"]
, "x86_64-linux", "/nix/store/q5wfq0i6w4p6a7155p4hia6p3n8rk7aq-bash-4.2-p47/bin/bash"
, ["-e", "/nix/store/9krlzvny65gdc8s7kpb6lkx8cd02c25b-default-builder.sh"]
, [("buildInputs", "")]
, ("builder", "/nix/store/q5wfq0i6w4p6a7155p4hia6p3n8rk7aq-bash-4.2-p47/bin/bash")
, ("doCheck", "1"), ("name", "hello-2.9"), ("nativeBuildInputs", "")
, ("out", "/nix/store/v5af71yn6jrzig3m53r4hqbbpmjqjqqz7-hello-2.9")
, ("propagatedBuildInputs", ""), ("propagatedNativeBuildInputs", "")
, ("src", "/nix/store/xdilnlzvvsf7r33gs4vy9jq2bmazlc0j-hello-2.9.tar.gz")
, ("stdenv", "/nix/store/v4ii73wpbg78b5p6fn0yq1cn76ya8fp4-stdenv")]
```


NIXPKGS: COLLECTION OF NIX EXPRESSION/PACKAGES

- ~8000 packages
- Wide range of libraries, tools, and applications
- Basis for NixOS : Linux distribution on top of Nix and Nixpkgs



NIX CHANNEL

- Nix allows for simultaneous source / binary deployment.
- Prebuilt nix packages with the same hash can be deployed to any systems with Nix.
- Due to the unique Nix system, completely independent Nix channels can be subscribed by users without clashes.

Very easy and safe to operate Nix channels!

HYDRA: CONTINUOUS BUILD SYSTEM ON NIX

Hydra Dashboard Status Project Jobset Admin Search Preferences Sign out

Jobset hep-nix-overlay:trunk

Actions Evaluations Jobs Configuration Links

Search jobs by name... Show inactive jobs

Job	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
darwin64.Atom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
darwin64.AtomEnv	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
darwin64.AtomEnvMin	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
darwin64.CheckMATE	✓	✓	✓												
darwin64.CheckMATEEnv	✓	✓	✓												
darwin64.Delphes	✓	✓	✓	✓	✓	✓									
darwin64.FastJet	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
darwin64.FastJetEnv	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
darwin64.Fastlim	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
darwin64.FastlimEnv	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
darwin64.HEPUtil	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
darwin64.HERWIGpp	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
darwin64.HERWIGppEnv	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
darwin64.HROOT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
linux64.Atom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
linux64.AtomEnv	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
linux64.AtomEnvMin	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
linux64.CheckMATE	✓	✓	✓												
linux64.CheckMATEEnv	✓	✓	✓												
linux64.Delphes	✓	✓	✓	✓	✓	✓									
linux64.FastJet	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

- Based on Nix
- Can employ multiple build workers with different platforms.
- Still early yet, but already has a nice web interface!

from hep-nix-overlay channel


HYDRA: CONTINUOUS BUILD SYSTEM ON NIX

- Success/Failure Log

Hydra Dashboard Status ▾ Project ▾ Jobset ▾ Job ▾ Admin ▾ Search Preferences Sign out

Build 210 of job hep-nix-overlay:trunk:darwin64.HERWIGpp

Actions ▾ Summary Details Inputs Build steps History chart Build dependencies

 Build ID: 210
Status: Build returned a non-zero exit code
System: x86_64-darwin
Nix name: Herwig++-2.7.0
Part of: evaluation 8
Duration: 25m 7s; finished at 2014-08-25 16:00:18
Logfile: [pretty](#) [raw](#) [tail](#)

Failed build steps

Nr	What	Duration	Machine	Status
1	Build of /nix/store/gw12qgbv8dmq8cy6w4m7xhyxz02b3g9- Herwig++-2.7.0	25m 6s	128.141.253.204	Failed: builder for '/nix/store/2x78yr4bjplm3ki167iyal4cxxx93cyp- Herwig++-2.7.0.drv' failed with exit code 100 (log, raw, tail)

Hydra Dashboard Status ▾ Project ▾ Jobset ▾ Job ▾ Admin ▾ Search Preferences Sign out

Log of build 210 of job hep-nix-overlay:trunk:darwin64.HERWIGpp

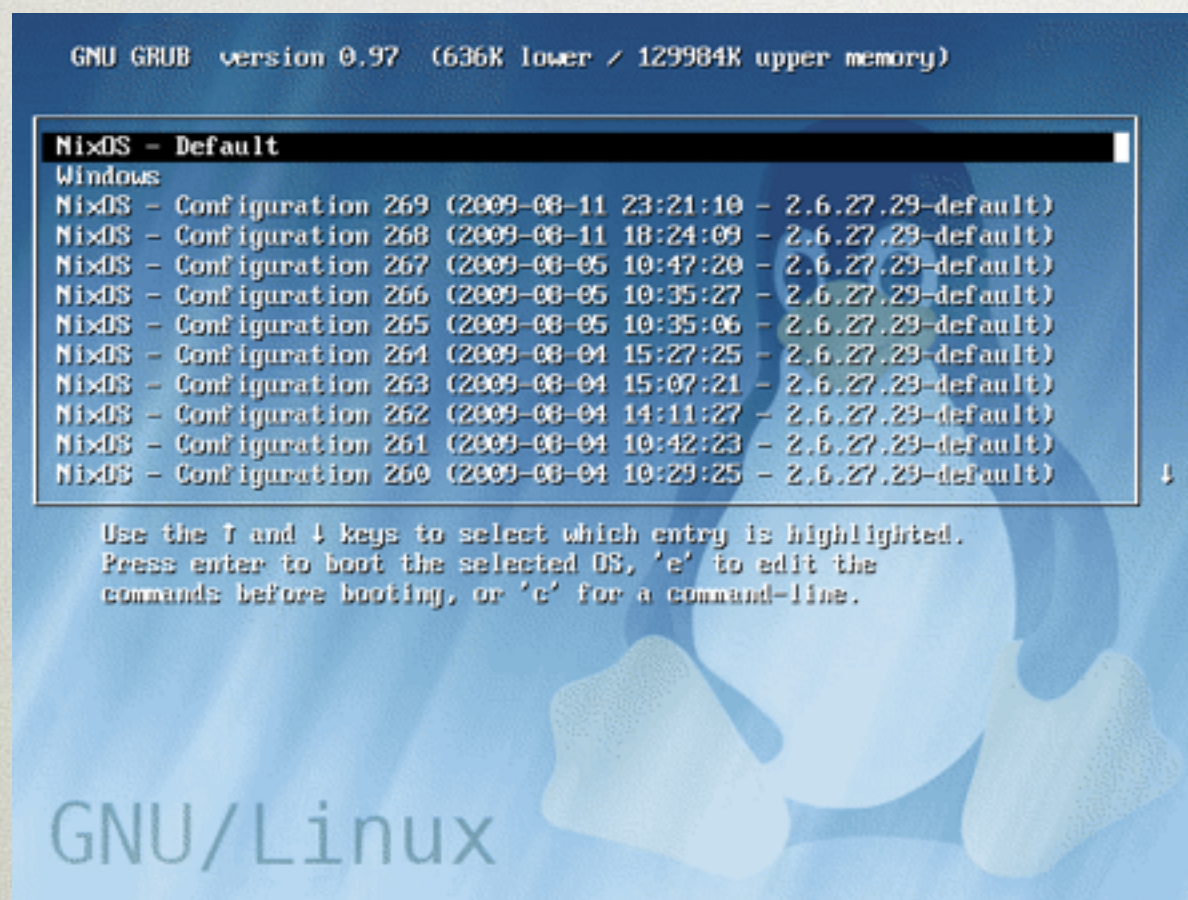
This is the build log of derivation /nix/store/2x78yr4bjplm3ki167iyal4cxxx93cyp-Herwig++-2.7.0.drv.

[+ Expand all](#) [- Collapse all](#)

```
- building /...-Herwig++-2.7.0 1
+ unpacking sources 2
+ patching sources 5
+ configuring 6
+ building 299
- installing 2222
  install flags: install SHELL=/...-bash-4.2-p45/bin/sh 2223
  - building install-recursive 2224
    Making install in include 2225
    + make[1]: Entering directory '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-2226-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/include'
    - make[3]: Entering directory 3053
      '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/src'
      - building install-am 3054
        + make[4]: Entering directory 3055
          '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/src'
          + building install-binPROGRAMS 3056
          + building install-binSCRIPTS 3060
          + building install-dist_pkgdataDATA 3063
          + building install-pkgdataDATA 3067
          - building install-data-am 3070
            make install-data-hook 3071
            - make[5]: Entering directory 3072
              '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/src'
              - building install-data-hook 3073
                Creating repository 3074
                Herwig++(29837,0x7fff7c03e310) malloc: *** error for object 0x106478a60: pointer being freed was not allocated 3075
                *** set a breakpoint in malloc_error_break to debug 3076
                make[5]: *** [install-data-hook] Abort trap: 6 3077
                make[5]: Leaving directory 3078
                '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/src'
                make[4]: *** [install-data-am] Error 2 3079
                make[4]: Leaving directory 3080
                '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/src'
                make[3]: *** [install-am] Error 2 3081
                make[3]: Leaving directory 3082
                '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/src'
                make[2]: *** [install-recursive] Error 1 3083
                make[2]: Leaving directory 3084
                '/private/var/folders/x9/w30hr7wj7yv9l3mt8c63xdr80000gp/T/nix-build-Herwig++-2.7.0.drv-0/Herwig++-2.7.0/src'
```


OTHER INFORMATION AND BEYOND

- Nix / Nixpkgs / Hydra are open source software.
 - Nix: LGPL 2.1, Nixpkgs: BSD3, Hydra: GPL3
- NixOS: Linux distribution



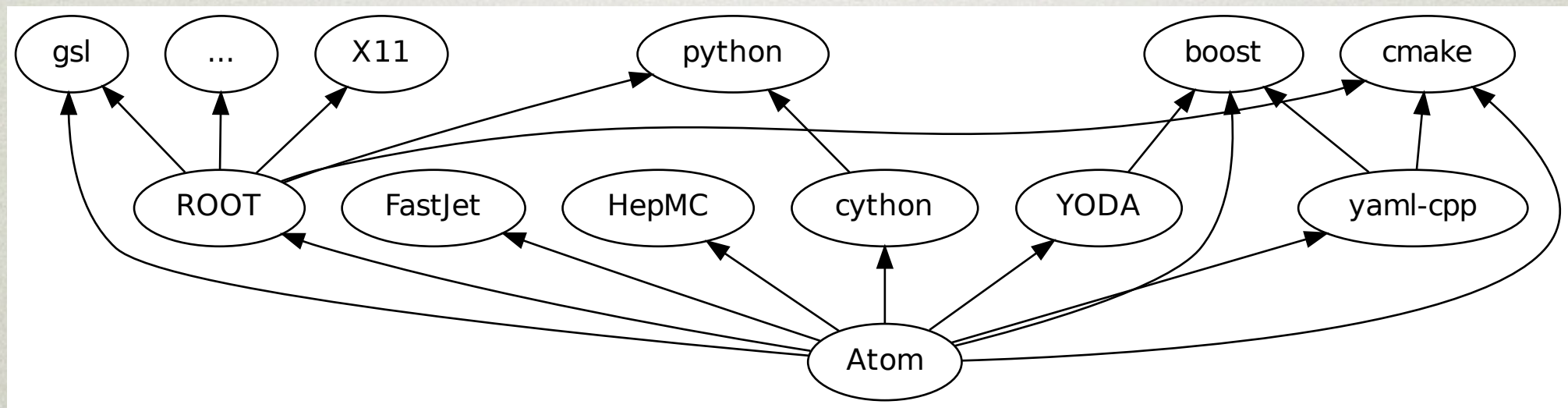
Even one can roll back an OS setup and reboot.

grub screenshot from NixOS

- NixOps: NixOS-based cloud deployment tool

MY PROJECT: HEP-NIX-OVERLAY

- Make a light-weight distribution channel for HEP software in TH community
- I want to deliver my software to newbies!

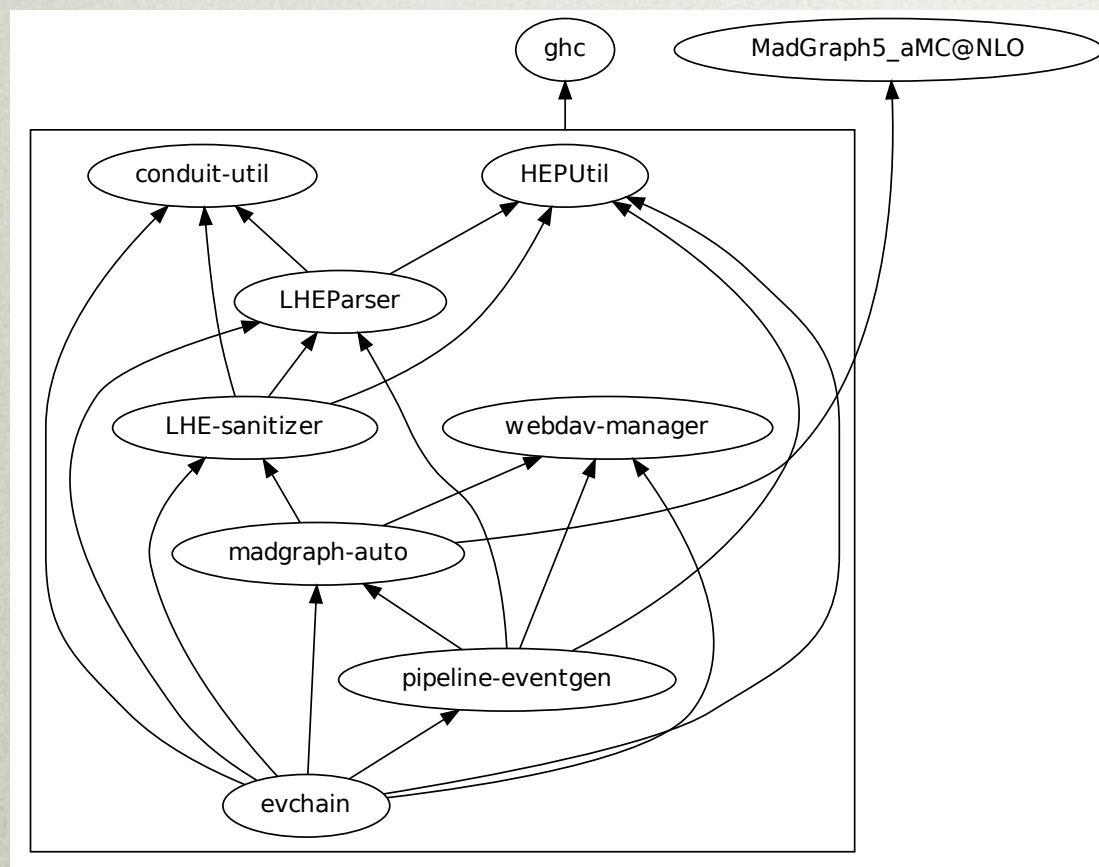


*Dependency graphs of ATOM:
Automatic Tester Of Models*

MY PROJECT:

HEP-NIX-OVERLAY

- Some software is even written in uncommon language, tools and build procedures.



Dependency graphs of evchain

ex) `evchain`: tool for chaining multi-part MC events

- written in Haskell
- Toolchain can be quite unfamiliar to average HEP people.
- Dependency is complicated. Although Haskell build tools (`cabal`) can deal with this well, build procedure may involve install/reinstall dependency packages.

MY PROJECT: HEP-NIX-OVERLAY

- Requirements
 - Flexible and newbie friendly: source / binary installation
 - Based on a variety of development packages
 - User-space installation: considering cluster operation
 - Can allow for massive 3rd party contributions: stable with multiple version of installed packages
 - OS X support

MY PROJECT:

HEP-NIX-OVERLAY

- hep-nix-overlay repo:
<http://github.com/wavewave/hep-nix-overlay>
- hep-nix-overlay hydra status:
<http://hep-nix-overlay.cern.ch>
- hep-nix-overlay channel:
<http://hep-nix-overlay.cern.ch/project/hep-nix-overlay/channel/latest>
- Anyone can contribute to hep-nix-overlay repo and then the build will be tested with Hydra continuously.

HEP-NIX-OVERLAY

EX) HROOT

- HROOT: Haskell binding to ROOT
- Nix installation

```
$ bash <(curl https://nixos.org/nix/install)
```

- Subscribing hep-nix-overlay

```
$ nix-channel --add http://hep-nix-overlay.cern.ch/  
project/hep-nix-overlay/channel/latest hep-nix-overlay
```

```
$ nix-channel -update
```

- Installing HROOT

```
$ nix-env -i haskell-HROOT-ghc7.8.3
```

Will download and install ghc, ROOT, HROOT

CONCLUSION

- Nix is a purely functional package manager.
- Nix manages every package with hash in nix store to achieve purity.
- On Nix systems, multiple versions with different dependencies can coexist with each other.
- Hydra is a continuous build system based on Nix.
- hep-nix-overlay is a light-weight HEP software distribution channel.

REFERENCES

- Nix/NixOS project homepage: <http://nixos.org>
- E. Dolstra's Ph.D. thesis:
"The Purely Functional Software Deployment Model"
<http://nixos.org/~eelco/pubs/phd-thesis.pdf>
- NixOS github organization: <http://github.com/NixOS>
- Nixpkgs monitor: <http://monitor.nixos.org>
- Nixpkgs hydra build system: <http://hydra.nixos.org>
- Blog aggregation: <http://planet.nixos.org>

THANK YOU!