# Nix(OS): An overview

Revolutionizing packaging and configuration management!

NixOS

The Purely Functional Linux Distribution

# About me

- Michael Weiss aka. primeos
- https://www.primeos.dev/
- First Nixpkgs commit: 2016-10-05
- I maintain <68 packages, 2 (5) modules, and 5 VM tests (2023-07-01)
  - Currently I lack time and have to reduce the number of packages that I maintain :(
- Email: nixos@primeos.dev

- Questions + feedback: Please just interrupt me at any time ;)
  - Questions are very welcome (but longer questions at the end please (due to time constraints))

# Motivation: Limitations of classical package managers

- Imperative vs. declarative management/administration
- Upgrades/configuration changes destructively update the system state (overwriting files in sequence -> temporary inconsistency)
- No configuration management
- State/mutable -> undeterministic builds -> not reproducible
- Different versions of a binary: Not possible (exceptions with version suffixes)
- Package conflicts (+ dependency resolution issues)
- No rollbacks
- Difficulty to get involved/started
- Limited “extensibility” (overlays / own package repos / overrides)

# Why declarative, immutable, etc.?

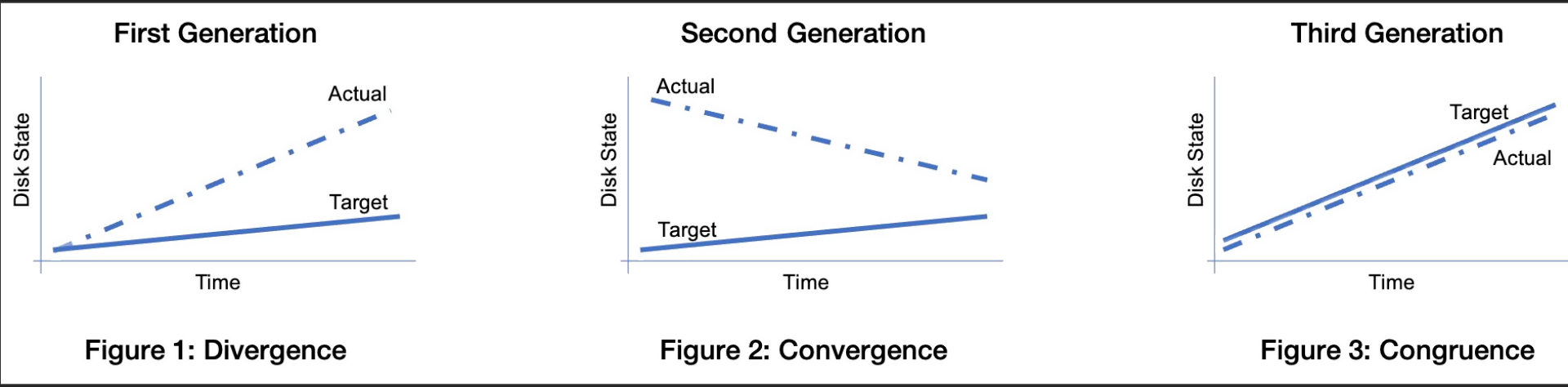



Figure 1: Divergence

Figure 2: Convergence

Figure 3: Congruence

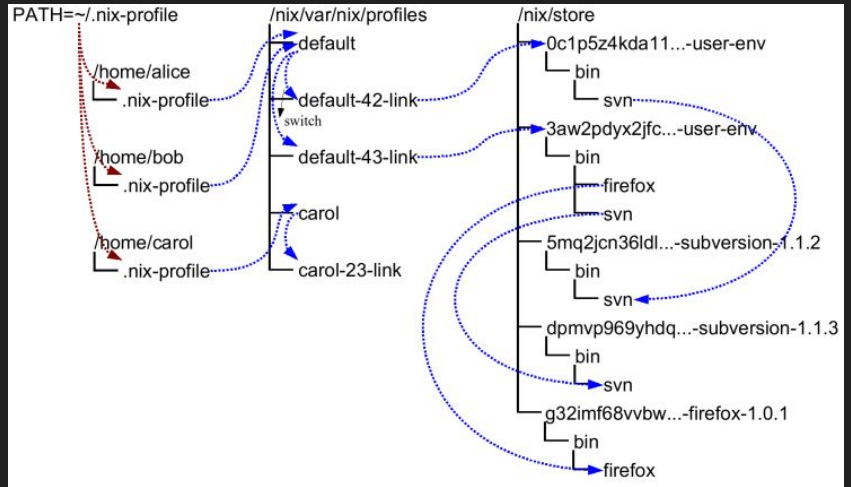
Source: https://tisgoud.nl/2020/02/config-management-camp-2020-day-2/

# Nix(OS): Features

- Declarative system configuration
- Deterministic & Reproducible builds (+ complete rebuilds with caching)
- Immutable package store
- Transparent source/binary deployment
- Atomic upgrades and rollbacks (software & configuration)
  - Also: Automatic service reloads/restarts, etc.
- Unprivileged users can securely install software
- Multiple versions of a package (side-by-side, e.g. testing a new Apache version)

# How?

- Nix expressions / Nix expression language (lazy, functional, dynamically typed)
  - A DSL to describes graphs of build actions ("derivations")
- Nix store (/nix/store/)
  - Packages prefixed by hash, e.g., /nix/store/y72i4llqf5zxvdp6b3j7ysixpdrid7qr-coreutils-9.1/
  - Hash: Captures all derivations (dependencies, configuration, etc.)
    - 160-bit truncations of SHA-256 encoded in base-32 (i.e. 32 characters) -> unique
  - Garbage collection (nix-collect-garbage)
  - New: Content Addressed Store (CAS)
- Symlinks (s. image)
  - For atomic updates and rollbacks

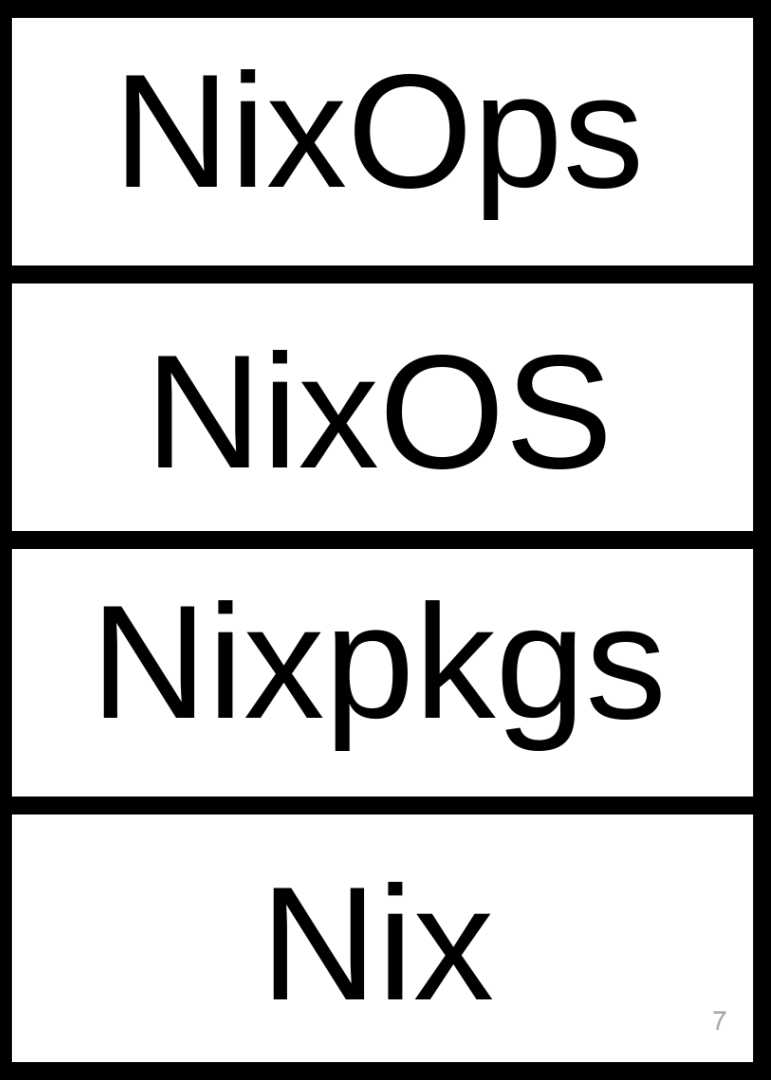


# Main components

- Nix (package manager)
- Nixpkgs (Nix packages collection)
- NixOS (operating system (modules))

- nix-env, nix-shell, nix run, etc.

- NixOps (DevOps / cloud deployment tool)
- Hydra (Nix based continuous build system)
- PatchELF (change dynamic linker and RPATH)
- {cabal,go,node,pip,python,pypi,...}2nix



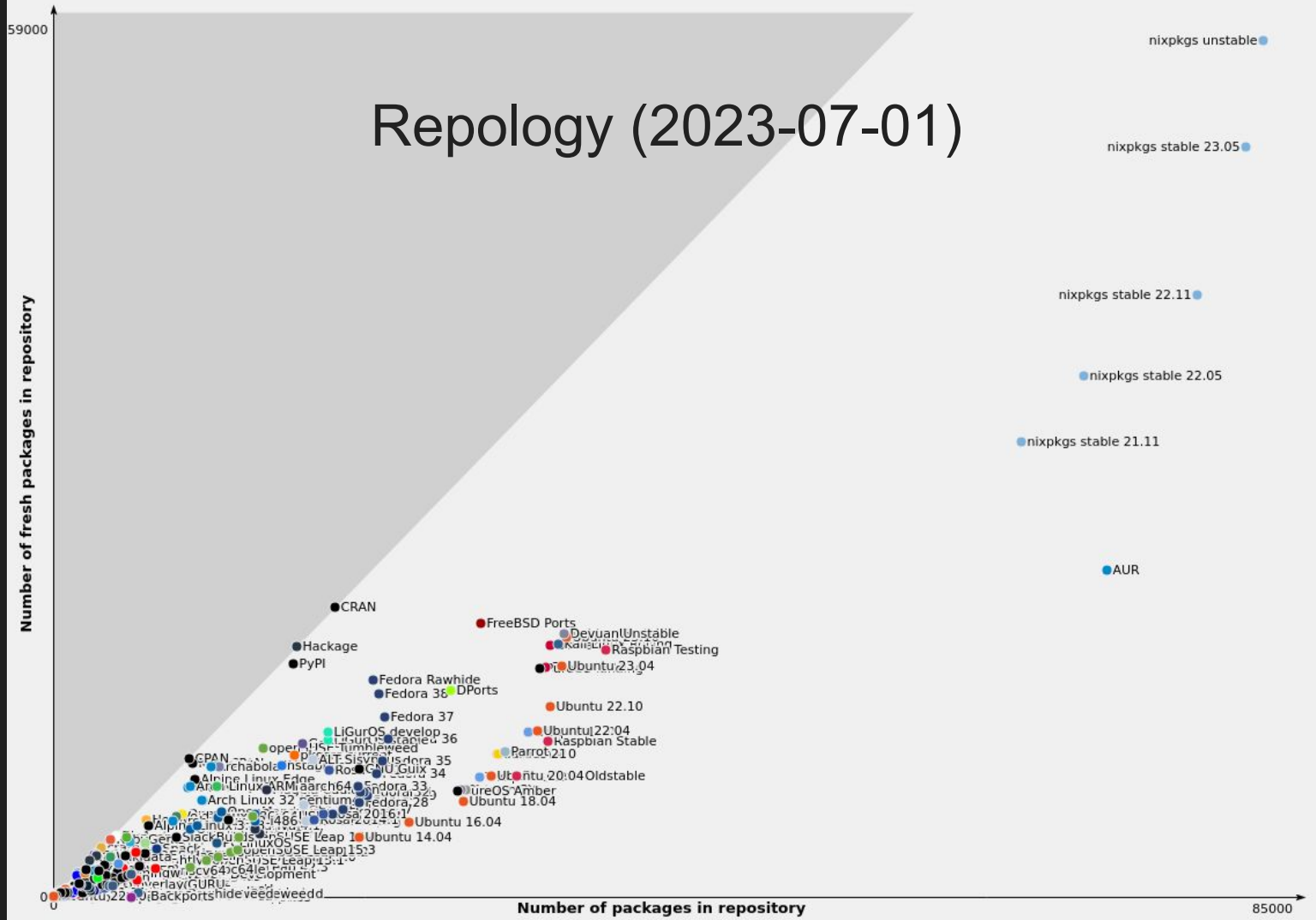
# Optional: History

- ~2003: Started as a research project (with funding)
- 2004: First paper (many will follow)
- Nix package manager developed by Eelco Dolstra as part of his PhD research
- First NixOS prototype developed by Armijn Hemel as his master's thesis project
- Hydra developed as part of the LaQuSo Buildfarm project
- 2007: NixOS becomes usable + x86_64 support
- 2011: Migration from Subversion to Git(Hub)
- 2015: NixOS Foundation + First NixCon (Berlin)

# Problems/drawbacks

- Learning curve (a completely different concept/approach -> new issues)
- Lacking manpower/workforce (e.g. for better testing/security/documentation)
- Running pre-compiled binaries
- Scripts with hard-coded paths don't work (even /bin/bash)
- Hydra / channel delays
- No LTS releases or super stable (i.e. old :P) branches
- Not all use-cases or configuration options supported
  - But: Some tricks available + PRs welcome ;)
  - And it’s always possible to override stuff
- No GUI for package/configuration management?
- Not all packages are reproducible (2016: 12.8% known not to be reproducible)

# Repology (2023-07-01)



# Getting started

- Try out Nix: https://nixos.org/download.html
  - Can be used side-by-side with your regular package manager (on your current Linux distro)
  - `$ sh <(curl -L https://nixos.org/nix/install) --daemon`
  - nix-env, nix-shell/“nix run”, nix repl, etc.
- Try out NixOS (inside a VM or for real :D)
- Learning resources: https://nixos.org/learn.html
- Help/community: https://nixos.org/community/
  - Nix and NixOS are developed and used by a diverse and welcoming community from all around the world.
- Get involved: https://nixos.org/guides/contributing.html
  - Submit changes (Nixpkgs PR): https://github.com/NixOS/nixpkgs/blob/master/CONTRIBUTING.md
    - Hint: Make sure to ping the right people

# Bonus slides

# Nix expressions / Nix expression language

- A DSL (not a GPL!)
  - Describes graphs of build actions ("derivations")
  - Packages, compositions of packages, configurations, ...
- Dynamically typed (~~"Nix won't be complete until it has static typing." @cdolstra~~) - https://typing-nix.regnat.ovh/
- Lazy (a very important feature!)
- Purely functional (no side-effects, immutable store)
- Turing complete (e.g. Dhall is not -> dhall-nix)

# NixOS

- Implements a declarative and purely functional system configuration model
- Based on Nix (package + configuration management)
- NixOS modules (separation of concerns)
    - Form the full "system configuration"

# Code

https://primeos.github.io/nixos-slides/index.html