# Reproducibility of Build Environments through Space and Time

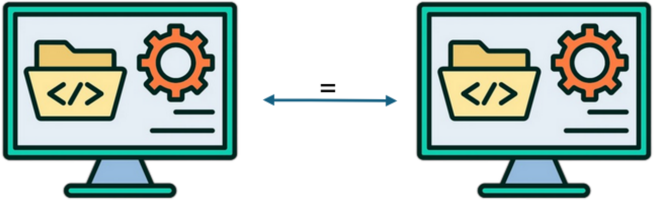
46th International Conference on Software Engineering (ICSE 2024), NIER Track

**Julien Malka (Télécom Paris)**
Stefano Zacchiroli (Télécom Paris)
Théo Zimmermann (Télécom Paris)

19 April 2024

# Reproducibility of build environments

Way say that two build environments are identical if they contain the exact same set of executables, up to their specific versions.



# Reproducibility in Space

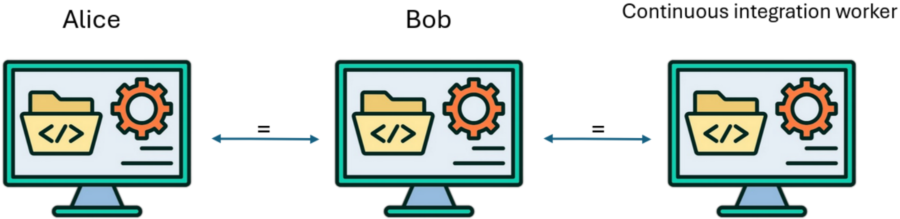
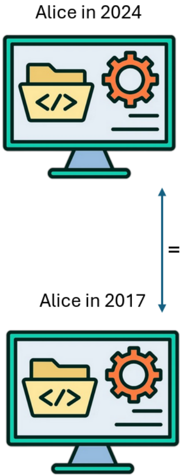


Figure – Reproducibility of build environments in Space

# Reproducibility in Time



Alice in 2024

=

Alice in 2017

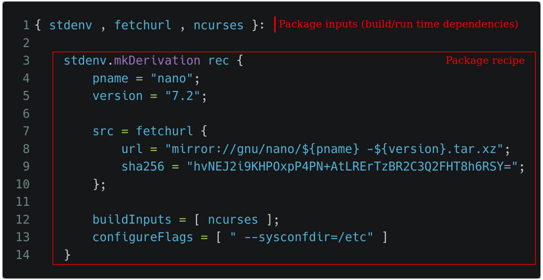
Figure – Reproducibility of build environments in Time

# Our vision

- Reproducibility of build environment is a crucial property for software engineering and research ;
- Functional package managers may be good candidates to provide reproducible build environments.

# Functional package managers

Software deployment/packaging model (from which **Nix**[1] and **Guix**[2] are implementations) based on a functional programming language.



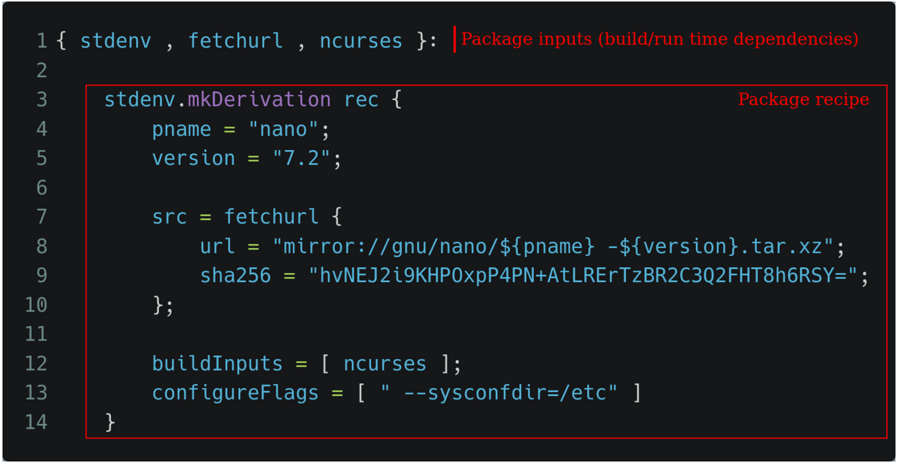
```
{ stdenv , fetchurl , ncurses }:

  stdenv.mkDerivation rec {
    pname = "nano";
    version = "7.2";

    src = fetchurl {
      url = "mirror://gnu/nano/${pname} -${version}.tar.xz";
      sha256 = "hvNEJ2i9KHPOxpP4PN+AtLRErTzBR2C3Q2FHT8h6RSY=";
    };

    buildInputs = [ ncurses ];
    configureFlags = [ " --sysconfdir=/etc" ]
  }
```

Figure – Nix expression for the *nano* package.

[1] https://nixos.org/

[2] https://guix.gnu.org/

# Functional package managers



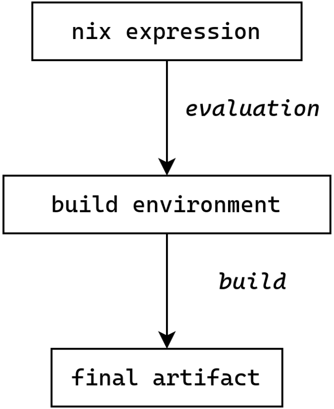
```
{ stdenv , fetchurl , ncurses }:

  stdenv.mkDerivation rec {
      pname = "nano";
      version = "7.2";

      src = fetchurl {
          url = "mirror://gnu/nano/${pname} -${version}.tar.xz";
          sha256 = "hvNEJ2i9KHPOxpP4PN+AtLRErTzBR2C3Q2FHT8h6RSY=";
      };

      buildInputs = [ ncurses ];
      configureFlags = [ " --sysconfdir=/etc" ]
  }
```

Package inputs (build/run time dependencies)

Package recipe

Figure – Nix expression for the *nano* package.

# Nix pipeline


nix expression

*evaluation*

build environment

*build*

final artifact

# Research questions

- **RQ1 :** Is space and time reproducibility of build environments achievable with Nix ?
- **RQ2 :** Does it allow rebuilding of past software versions ?

# Experimental protocol

1. Sample 200 revisions of the Nix software repository, picked from 2017 to 2023 ;
2. For *each sampled revision*, perform the **evaluation** of each package and compare with the historical truth (historical CI results) ;
3. For *the oldest revision* of our samples, perform the **build** of each package and compare with the historical truth.

# Results

***RQ1 :*** *Reproducibility of build environments*

- We were able to reproduce the build environment of 99.99% of the packages we tested ;
- Discrepancies we found were due to the (unfortunate) use of some of Nix's impure builtins.

***RQ2 :*** *Rebuilding past software versions*

- We were able to build successfully 14233 out of the 14242 (99.94%) packages that were built successfully by CI in 2017 ;
- Discrepancies we found were due to leakages of the Nix build sandbox, that we wish to investigate further.

# Future plans

Widen our rebuildability study by compiling more revisions of our sample set, in order to :

1. Study **software preservation** by observing the evolution of the proportion of rebuildable software over time and checking for the presence of the original sources ;
2. Study **bitwise artifact reproducibility** [3] of the Nix package repository for its impact on software supply chain security.

[3] https ://reproducible-builds.org/

# Thank you for your attention !