One Config, Any Machine

Jeffery Muter · Ohio LinuxFest 2026

ABOUT ME

Jeffery Muter

Software Developer
at the Wasserstrom Company

Father · Configuring Addict

saorsadev.com

Thank you

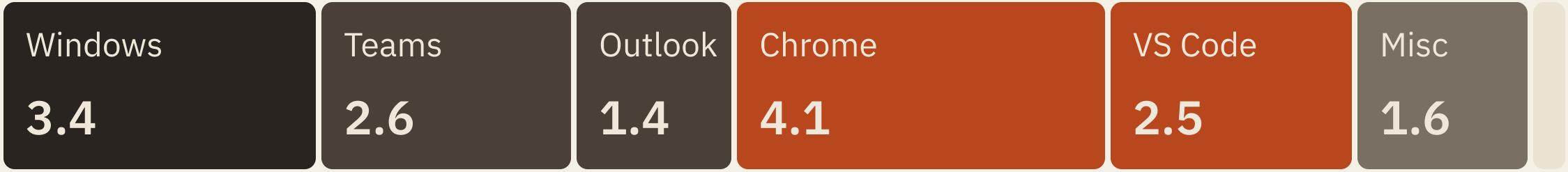
OLF organizers · speakers · volunteers · sponsors



imperative /ɪmˈpɛrətɪv/ · adj.

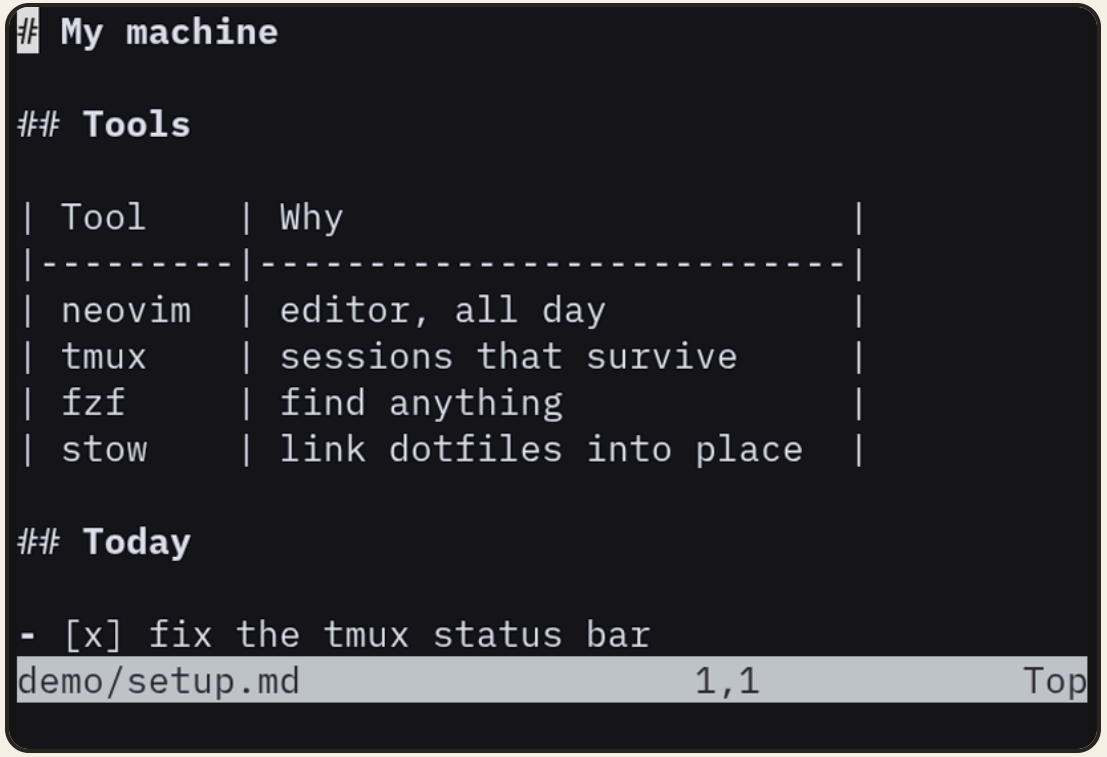
> Giving the steps, one command at a time,
> instead of describing the result you want.

RAM: 15.6 of 16 GB used



~~Chrome~~ → Firefox · barely helped ~~VS Code~~ → JetBrains · worse → **Neovim in WSL**

Neovim: same file, stock vs. my config



```
# My machine

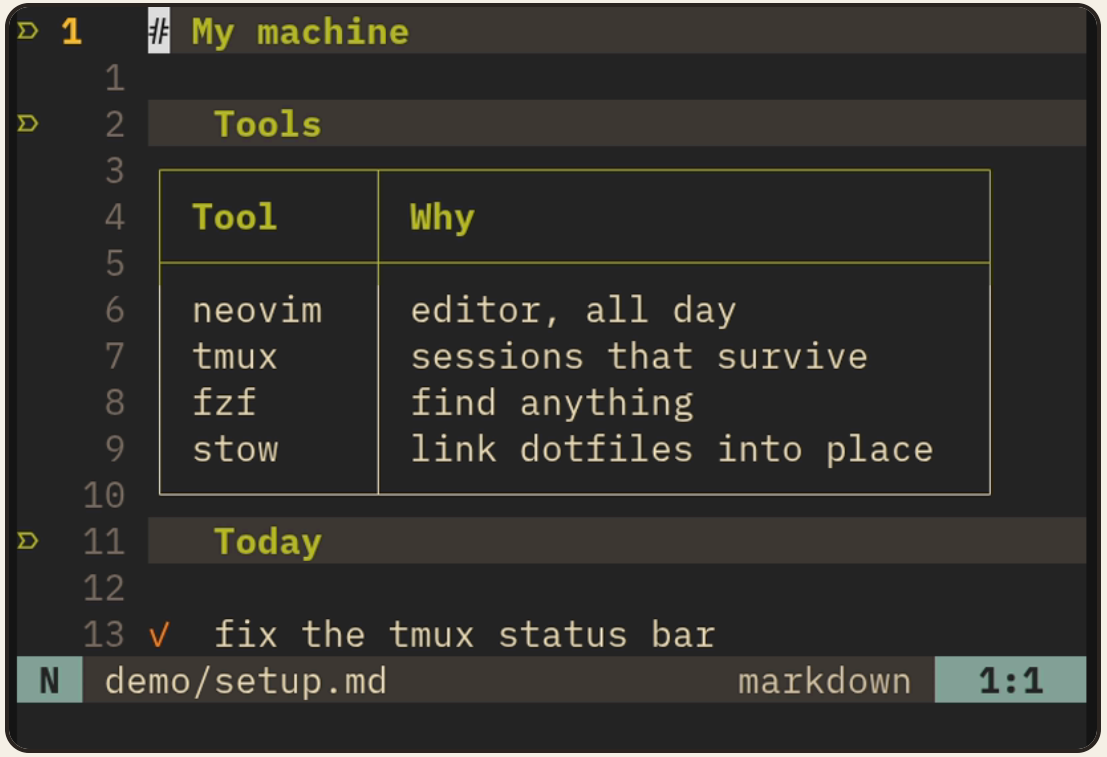
## Tools

| Tool     | Why                          |
|----------|------------------------------|
| neovim   | editor, all day              |
| tmux     | sessions that survive        |
| fzf      | find anything                |
| stow     | link dotfiles into place     |

## Today

- [x] fix the tmux status bar
demo/setup.md                    1,1           Top
```

Out of the box



```
 1  # My machine
 2
 3    Tools
 4
 5  | Tool     | Why                          |
 6  |----------|------------------------------|
 7  | neovim   | editor, all day              |
 8  | tmux     | sessions that survive        |
 9  | fzf      | find anything                |
10  | stow     | link dotfiles into place     |
11
12    Today
13
14  ✓ fix the tmux status bar
N  demo/setup.md                  markdown  1:1
```

After I customized it

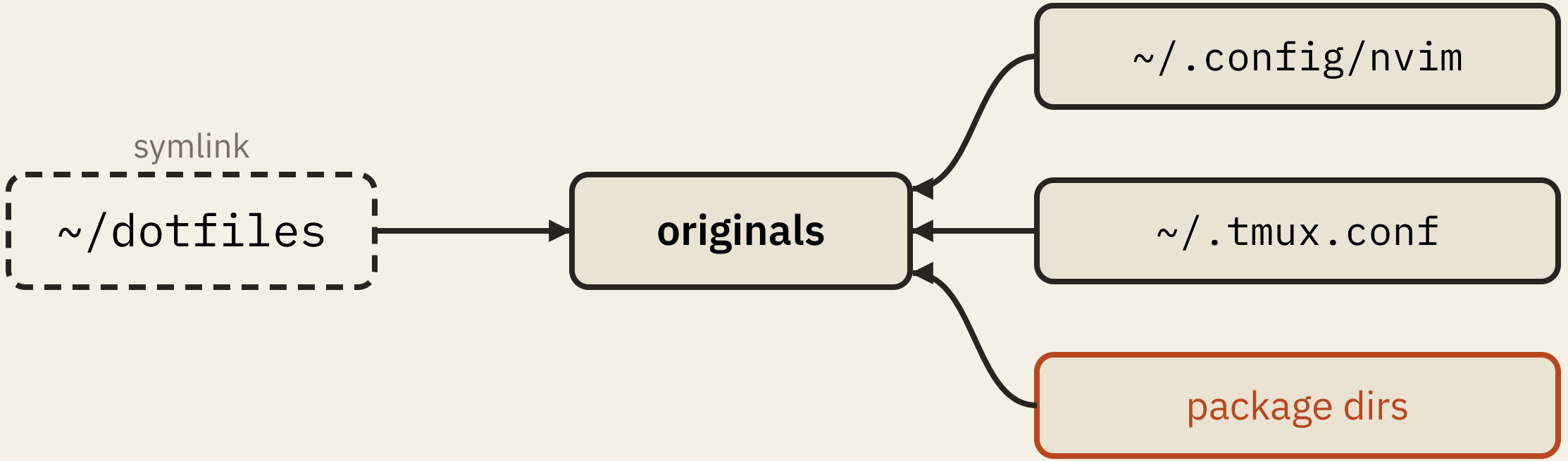
story · why · options · setup

Imperative: no record of what I changed

```
vim.opt.number = true
vim.opt.relativenumber = true
vim.opt.tabstop = 4
vim.opt.scrolloff = 8
vim.opt.undofile = true
```

Which of these did *I* change?

My backup plan



~/dotfiles was a symlink · never in git

story · why · options · setup

Build broke.
I blamed my backup.

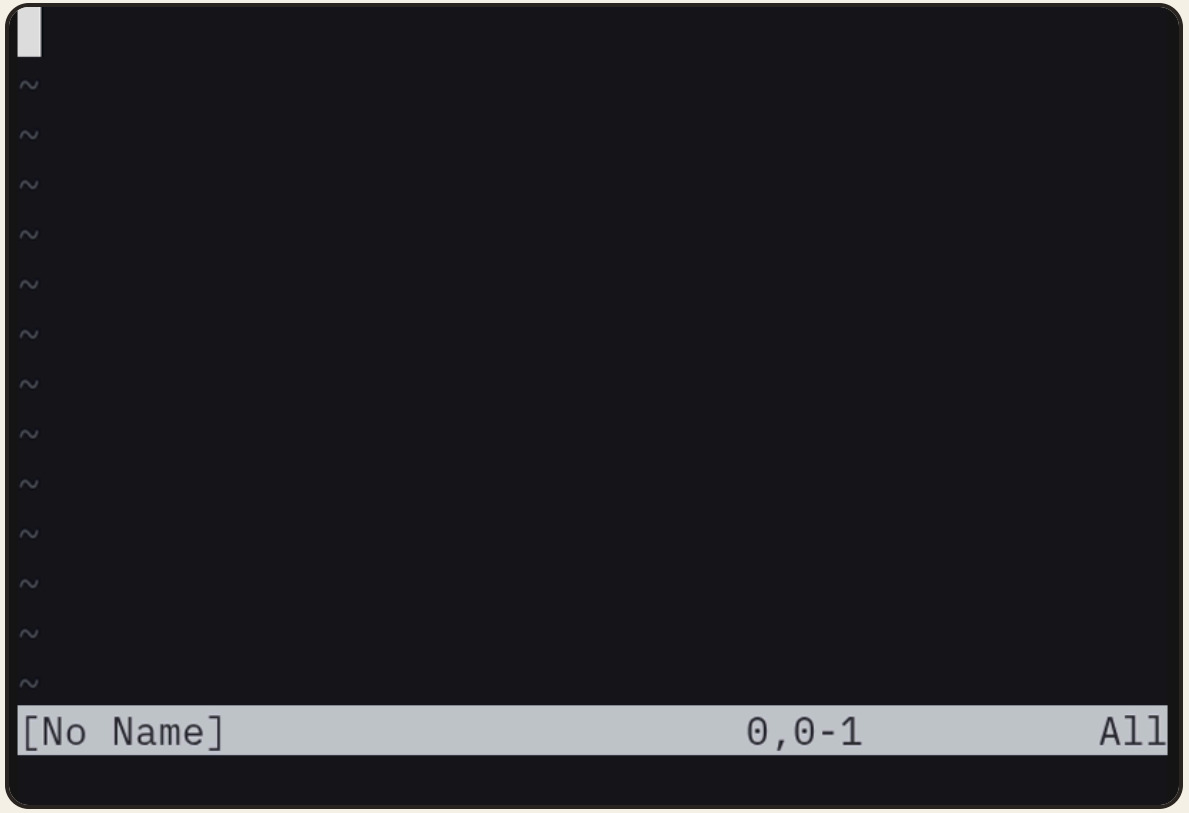
go1.24 go1.25

Two versions of one language, fighting.

```
rm -rf ~/dotfiles/
```

Everything gone

Five seconds later.
Customizations: gone.
New shell: can’t even
open files.

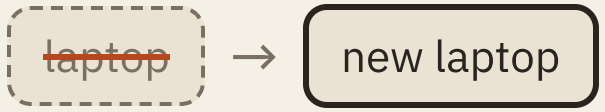


Denial · Anger · Bargaining · Depression

Never again.

“But I wouldn’t do that!”

But do these happen?



Machine dies



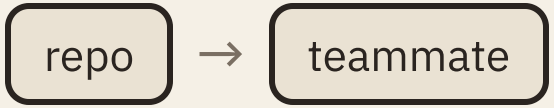
Multiple machines



**One machine,
multiple environments**



Remote in, get your tools



**Reproducible
for teammates**



**Multiple versions of
the same software**

Declarative: end state, not steps

IMPERATIVE

```
$ sudo apt install \
    ripgrep fd neovim
```

DECLARATIVE

```
home.packages = with pkgs; [
  ripgrep fd neovim
];
```

Steps you ran once, vs. a description you can run again on any machine.

Six ways people handle it today

Browser Defaults Remote Scripts App sync AI agent

Live in the browser

\+ nothing to set up · – RAM, offline = dead

Right for you if your work really is all SaaS.

	Survives death	Offline	Keeps custom	No drift	Setup effort
Browser	●	○	○	●	none

Just use defaults

+ zero upkeep, any box · – lose everything that makes you fast

Right for you if you rarely sit at the same machine twice.

	Survives death	Offline	Keeps custom	No drift	Setup effort
Defaults	●	●	○	●	none

Remote into one machine

\+ one machine to maintain · – network both ends, single point of failure

Right for you if you already have solid infra and connectivity.

	Survives death	Offline	Keeps custom	No drift	Setup effort
Remote	◐	○	●	◐	a day

Script the install

\+ familiar, automatable · – drifts, breaks when you need it

Right for you if you run a fleet that rarely changes, and test the script.

	Survives death	Offline	Keeps custom	No drift	Setup effort
Scripts	◐	◐	●	◐	hours → days

Your editor’s built-in sync

VS Code Settings Sync · JetBrains Settings Sync · Firefox Sync

\+ one click · – covers one app, tied to its account

Right for you if your whole world is one editor.

	Survives death	Offline	Keeps custom	No drift	Setup effort
App sync	●	◐	◐	●	1 click

Ask an AI agent

+ fast first draft · – non-deterministic, the prompt becomes the config

Right for you if you commit what it writes.

	Survives death	Offline	Keeps custom	No drift	Setup effort
AI agent	◐	○	◐	○	minutes

git + Stow + NixOS

+ rebuild anything, roll back anything · – learning curve, the Nix language

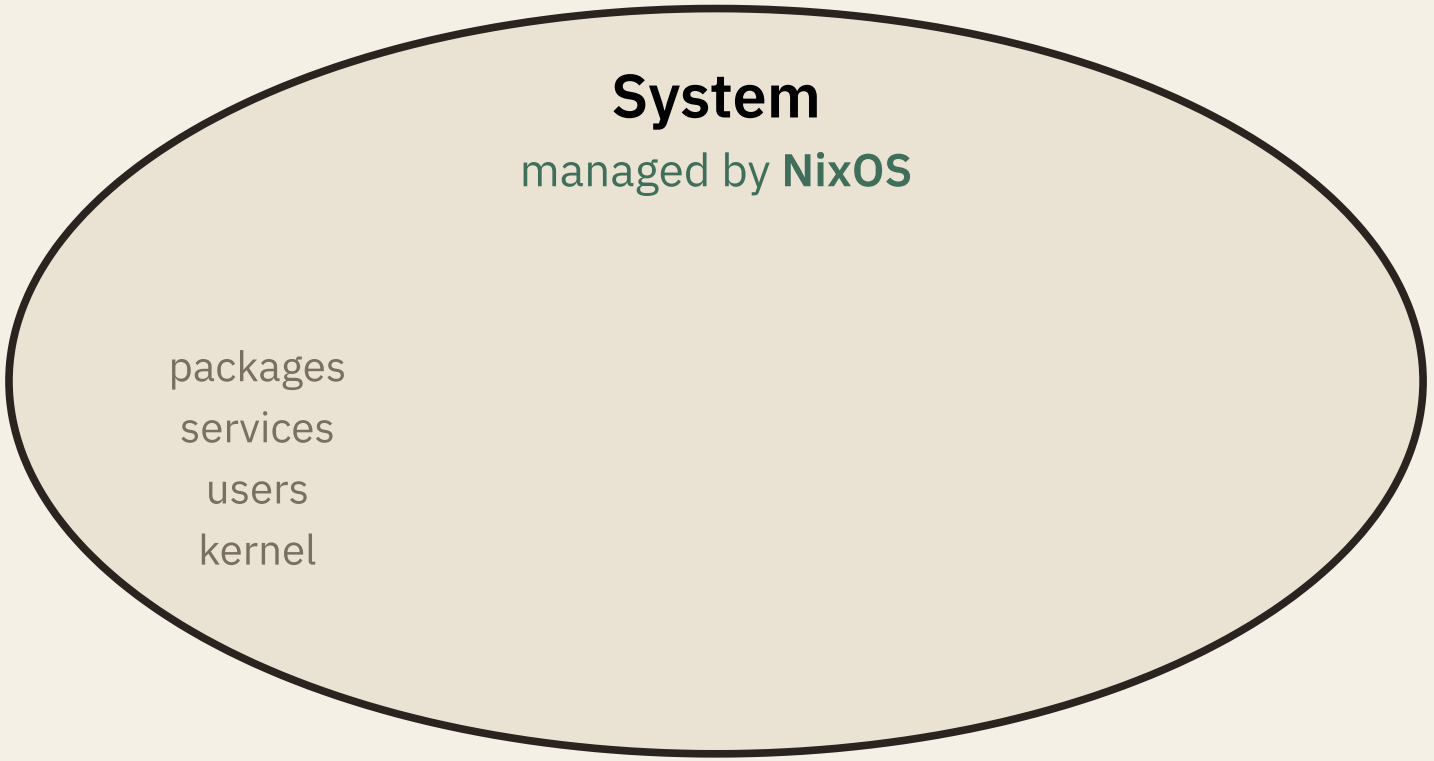
Right for you if you change your setup often and want an undo.

	Survives death	Offline	Keeps custom	No drift	Setup effort
My setup	●	◐	●	◐	**15 min → weekends**

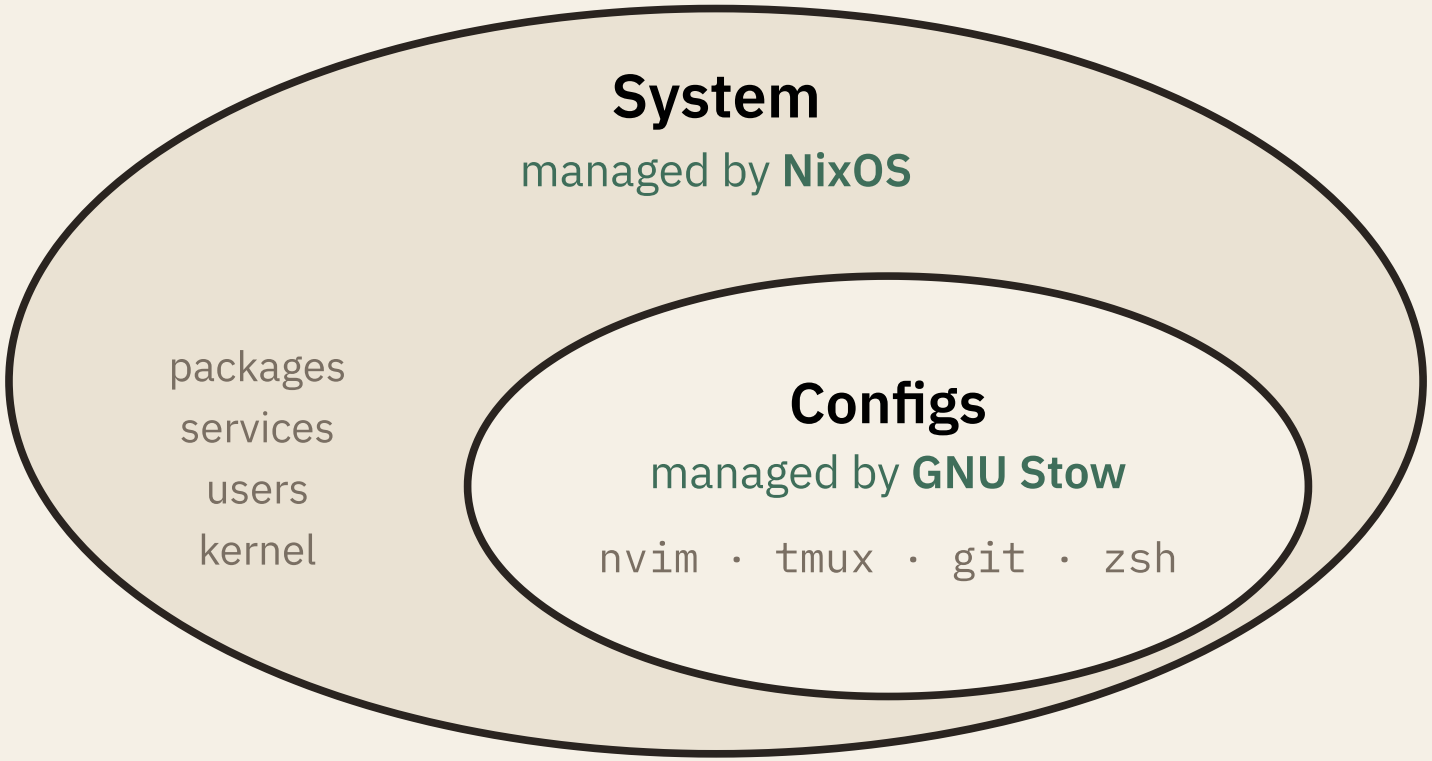
Side by side: pick your trade-off

	Survives death	Offline	Keeps custom	No drift	Setup effort
Browser	●	○	○	●	none
Defaults	●	●	○	●	none
Remote	◐	○	●	◐	a day
Scripts	◐	◐	●	◐	hours → days
App sync	●	◐	◐	●	1 click
AI agent	◐	○	◐	○	minutes
My setup	●	◐	●	◐	**15 min → weekends**

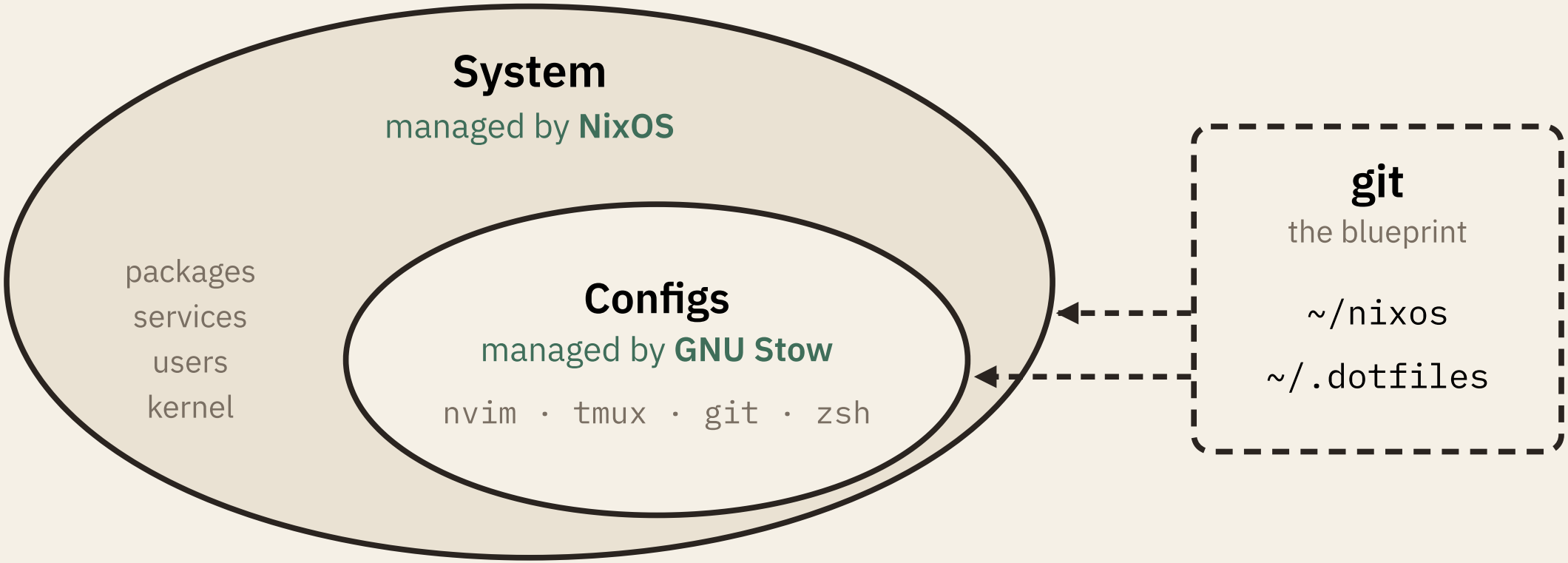
My setup: what manages what



My setup: what manages what



My setup: what manages what



git stores the blueprint · NixOS and Stow build from it

The undo I didn’t have

NixOS · generation 147	today
NixOS · generation 146	yesterday
NixOS · generation 145	last week

nixos-rebuild switch --rollback

git undoes your files · generations undo the system

Worth it for me. Maybe not for you.

306

306 commits · learning curve · Nix language · error messages

AI writes it. You review it.

```
> add ripgrep,
  drop htop
```

```
  fzf
- htop
+ ripgrep
  stow
```

```
$ nixos-rebuild
  switch
activating…
done
```

prompt → diff → deterministic rebuild

Do this tonight

1. `git init` your dotfiles
2. push them somewhere
3. pick a layer

Thank you

One Config, Any Machine

Jeffery Muter · saorsadev.com