

THE LORD OF THE MONOREPOS

AN EFFICIENT SETUP WITH PNPM AND NX

Thomas Huchedé

Front-end consultant at Zenika

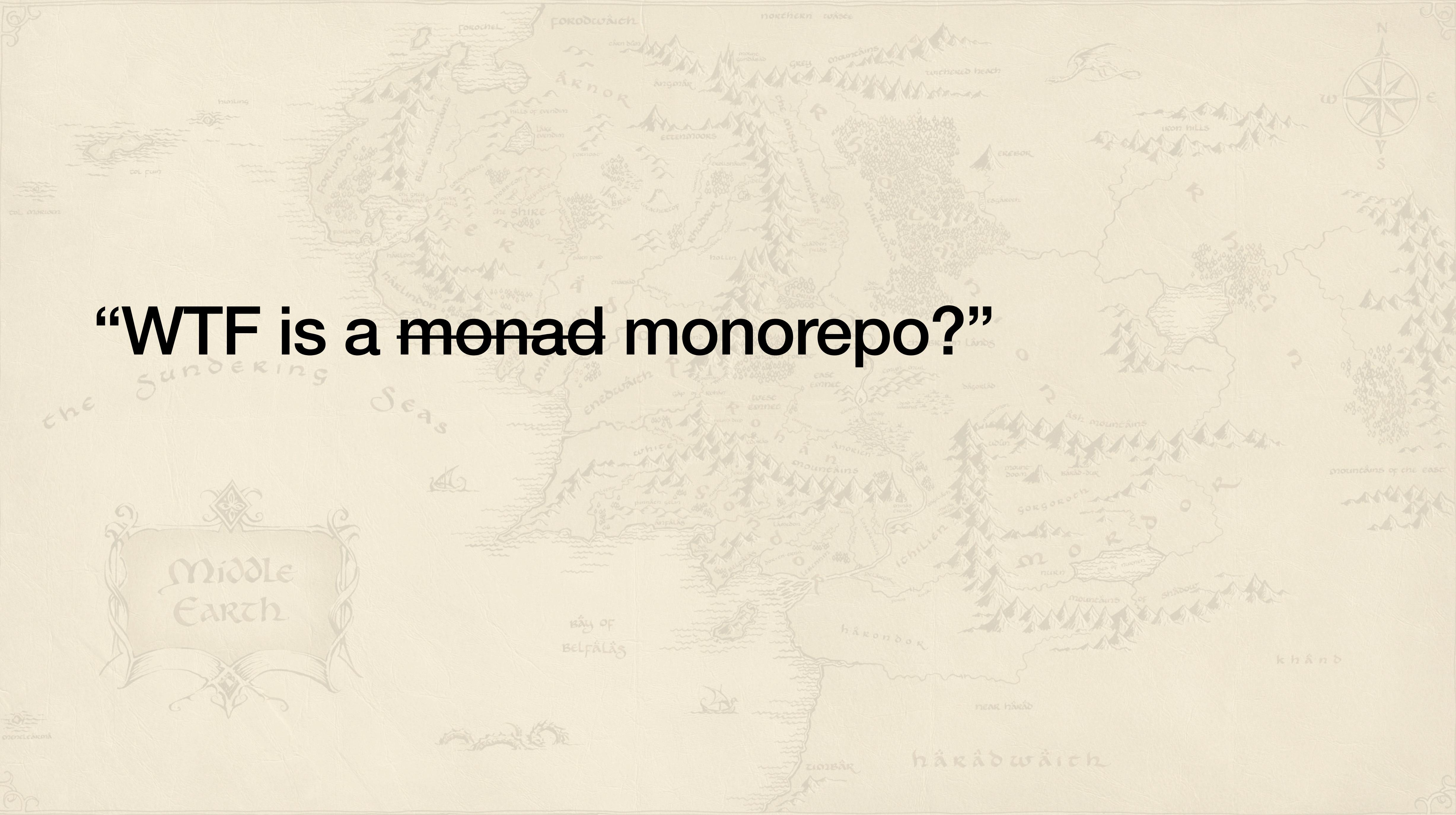
- All things front-end
- Performance and accessibility
- Currently learning Rust

@thuchede



“WTF is a monad monorepo?”

Middle
Earth





All right then. Keep your secrets

MONOLITHIC APP

- One server responsible for everything
 - Serving front-end assets
 - Running business logic
 - Accessing database
 - (Deployment scripts)

The background image is a landscape scene from a video game or movie. It features two large, ancient-looking trees with intricate root systems growing out of rocky outcrops. The sky is filled with orange and yellow hues, suggesting either a sunrise or sunset. In the foreground, there's a dense forest of green trees. On the left side, there's a small town or city with various buildings, some with lit windows. The terrain is rugged with mountains in the distance and some hills in the foreground.

GIT

Bash scripts



“My thought shall weigh as a cloud of
Doom”

Morgoth - The Children of Húrin

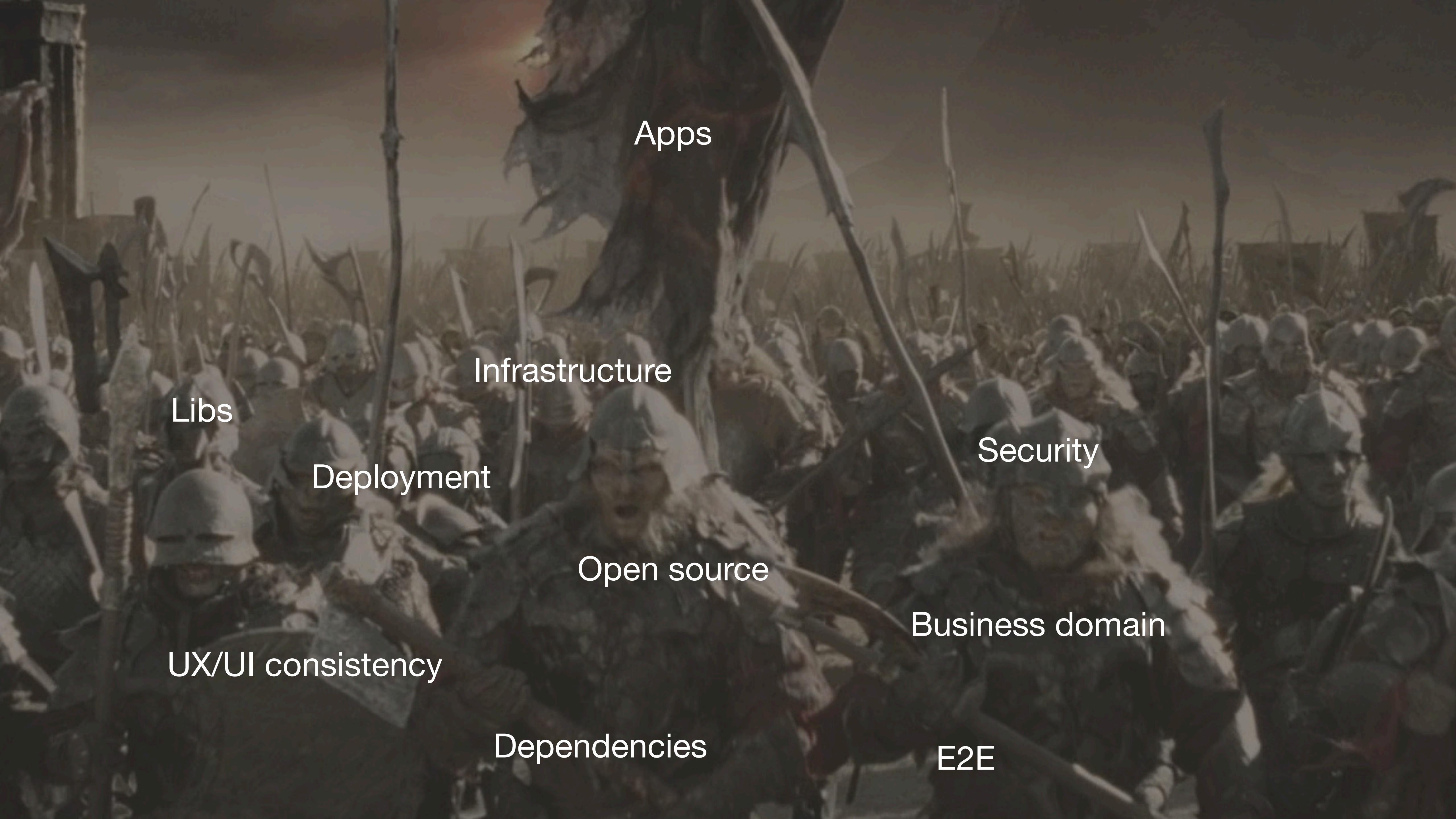
THE LORD OF THE MONOREPOS

AN EFFICIENT SETUP WITH PNPM AND NX

A dark, atmospheric illustration depicting a massive, multi-armed demon standing over a small, lone figure. The demon has long, spiky hair and a beard, and is holding a large, flaming sword. The scene is set against a backdrop of dark, swirling clouds.

Complexity

You



UX/UI consistency

Libs

Deployment

Infrastructure

Dependencies

Apps

Open source

E2E

Security

Business domain

A close-up photograph of a person's hand holding a smartphone. The phone's screen is visible, displaying the word "Versionning". The background is out of focus, showing a warm, bokeh effect with colors like orange, red, and yellow, suggesting a festive or celebratory atmosphere.

Versionning

Building

Deploying



THE HOBBIT

IGNORING THE PROBLEMS

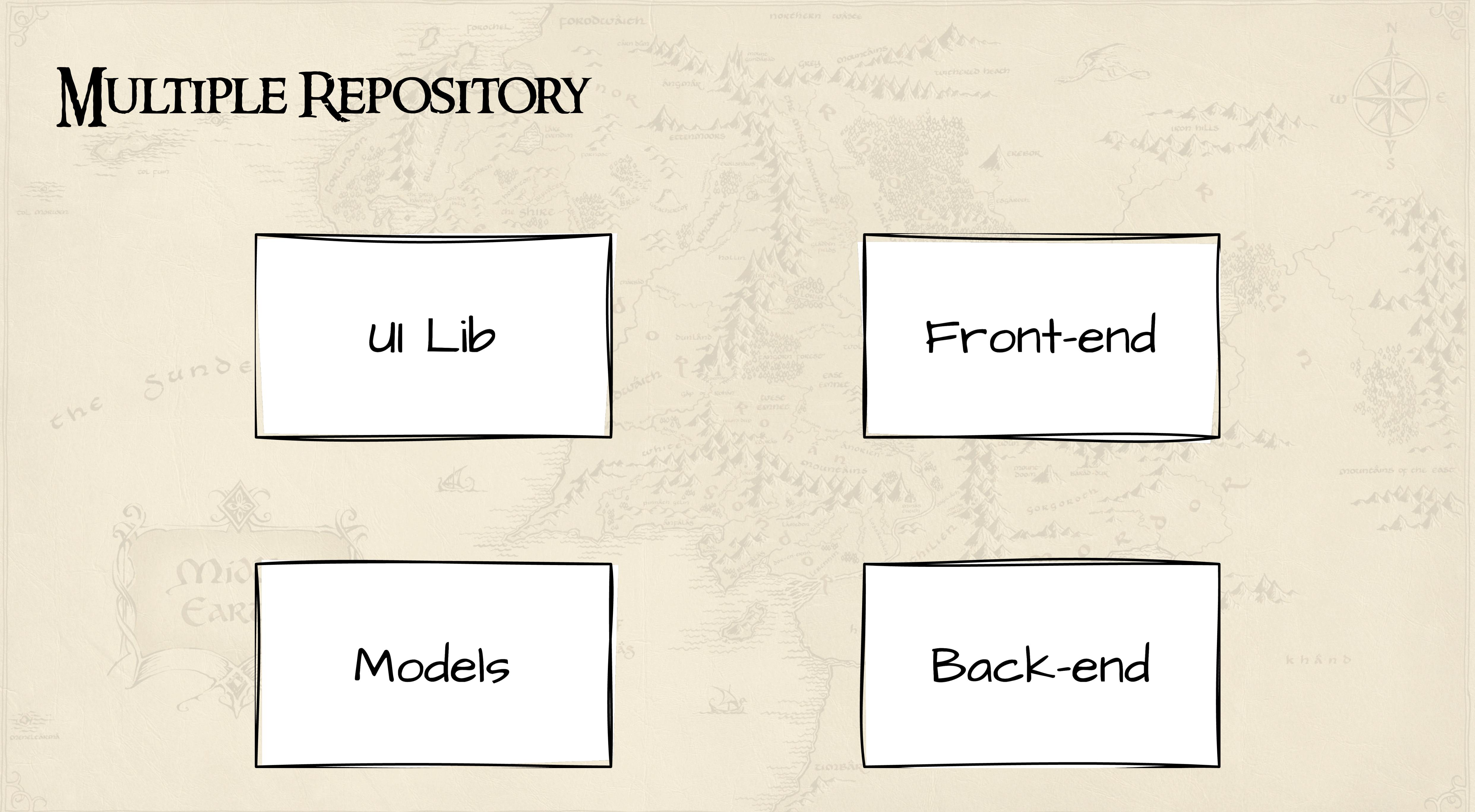
MULTIPLE REPOSITORY

UI Lib

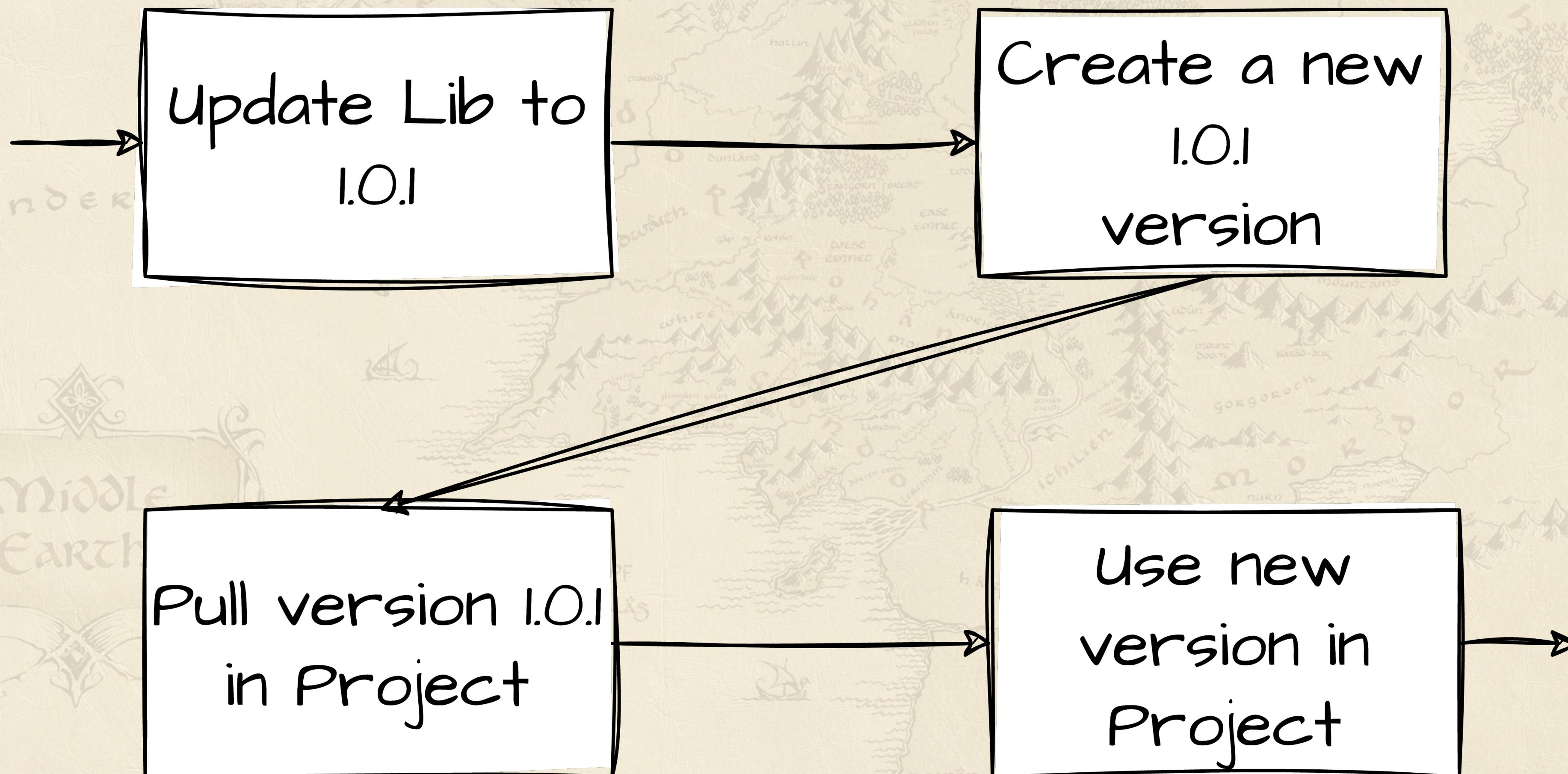
Front-end

Models

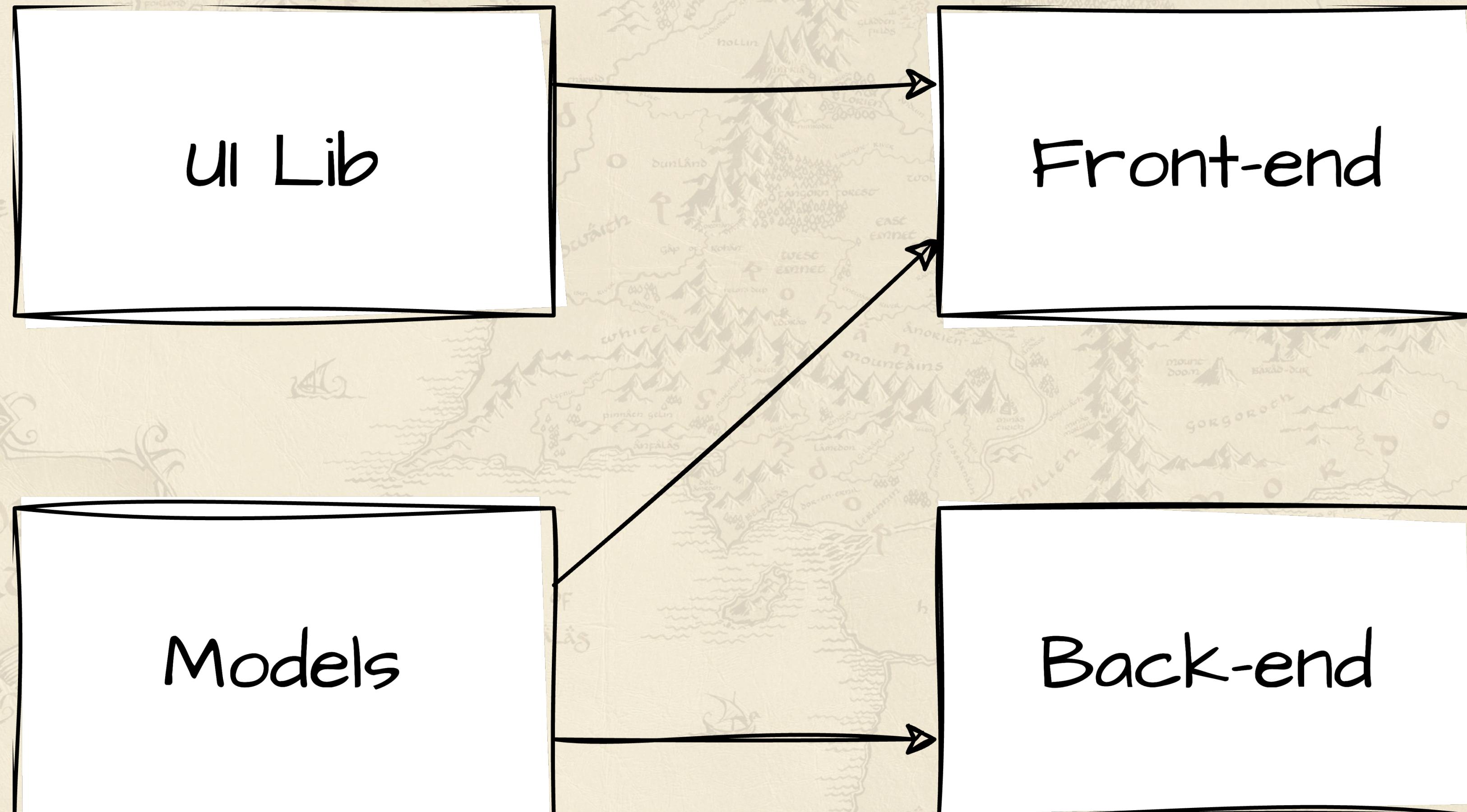
Back-end



“MANUAL” DEPENDENCIES MANAGEMENT



DEPLOYMENT



MONOREPO STRUCTURE

- `your-monorepo/`
 - `packages/`
 - `frontend/`
 - `frontend-e2e/`
 - `backend-1/`
 - `backend-2/`
 - `design-system/`
 - `models/`
 - `i18n/`
 - `utils/`

- `your-monorepo/`
 - `apps/`
 - `frontend/`
 - `frontend-e2e/`
 - `backend-1/`
 - `backend-2/`
 - `libs/`
 - `design-system/`
 - `models/`
 - `i18n/`
 - `utils/`



THE FELLOWSHIP OF THE RINGS

THE TOOLS WE CAN USE

THE FELLOWSHIP OF THE RING

package.json

Git

File system

Cache

Build tool

Package manager

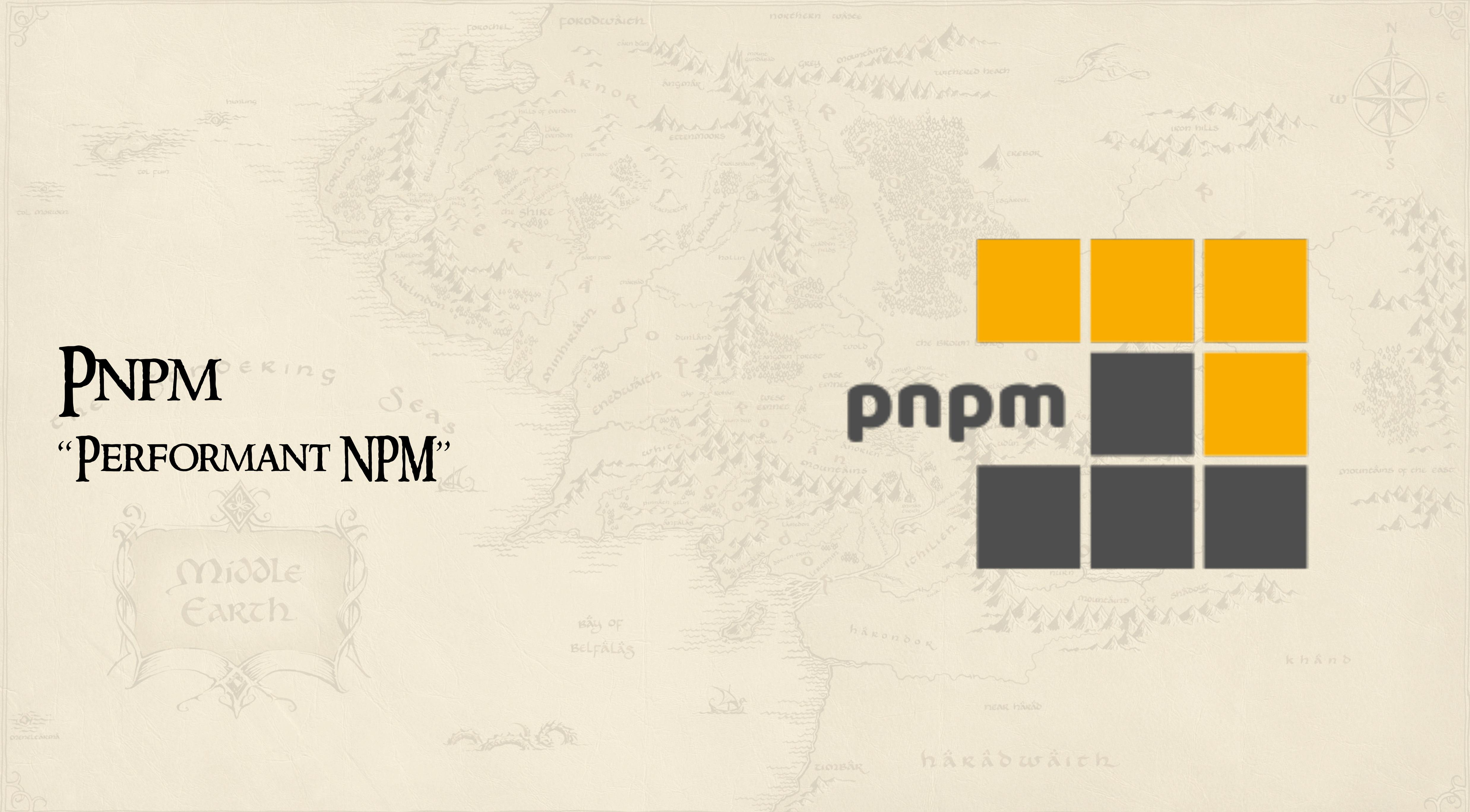


THE TWO TOWERS

PNPM AND Nx

PNPM

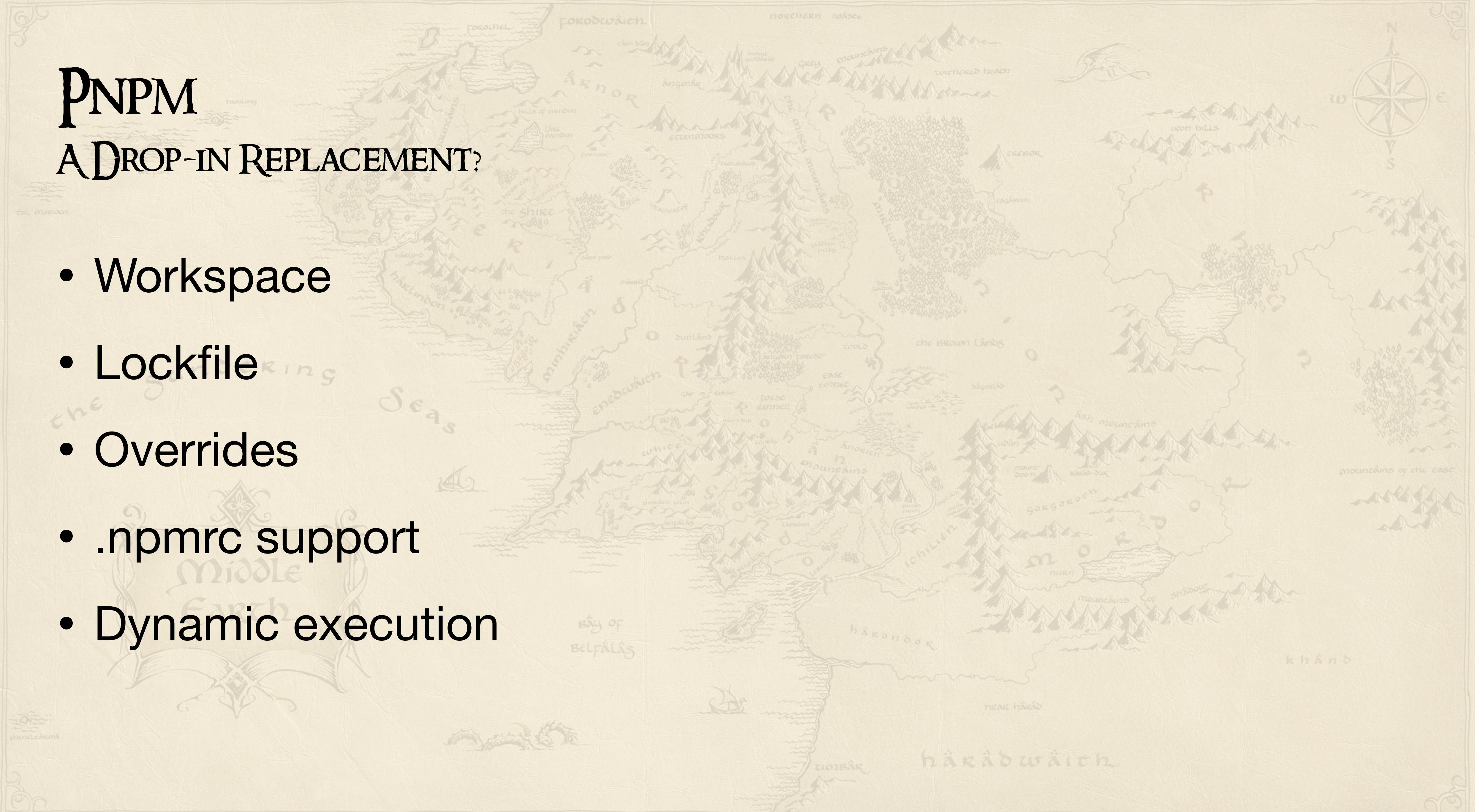
“PERFORMANT NPM”



PNPM

A DROP-IN REPLACEMENT?

- Workspace
- Lockfile
- Overrides
- .npmrc support
- Dynamic execution

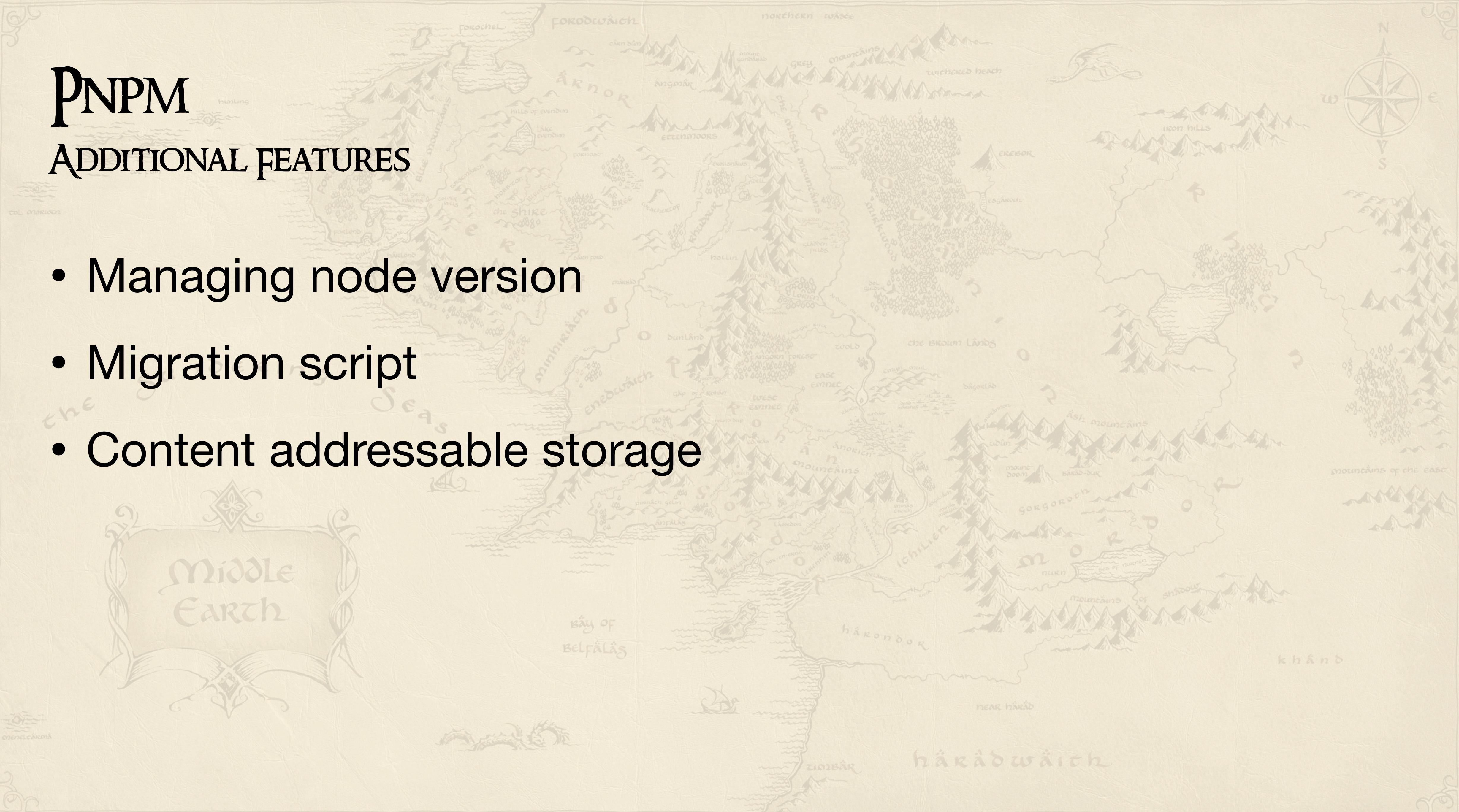


PNPM

ADDITIONAL FEATURES

- Managing node version
- Migration script
- Content addressable storage

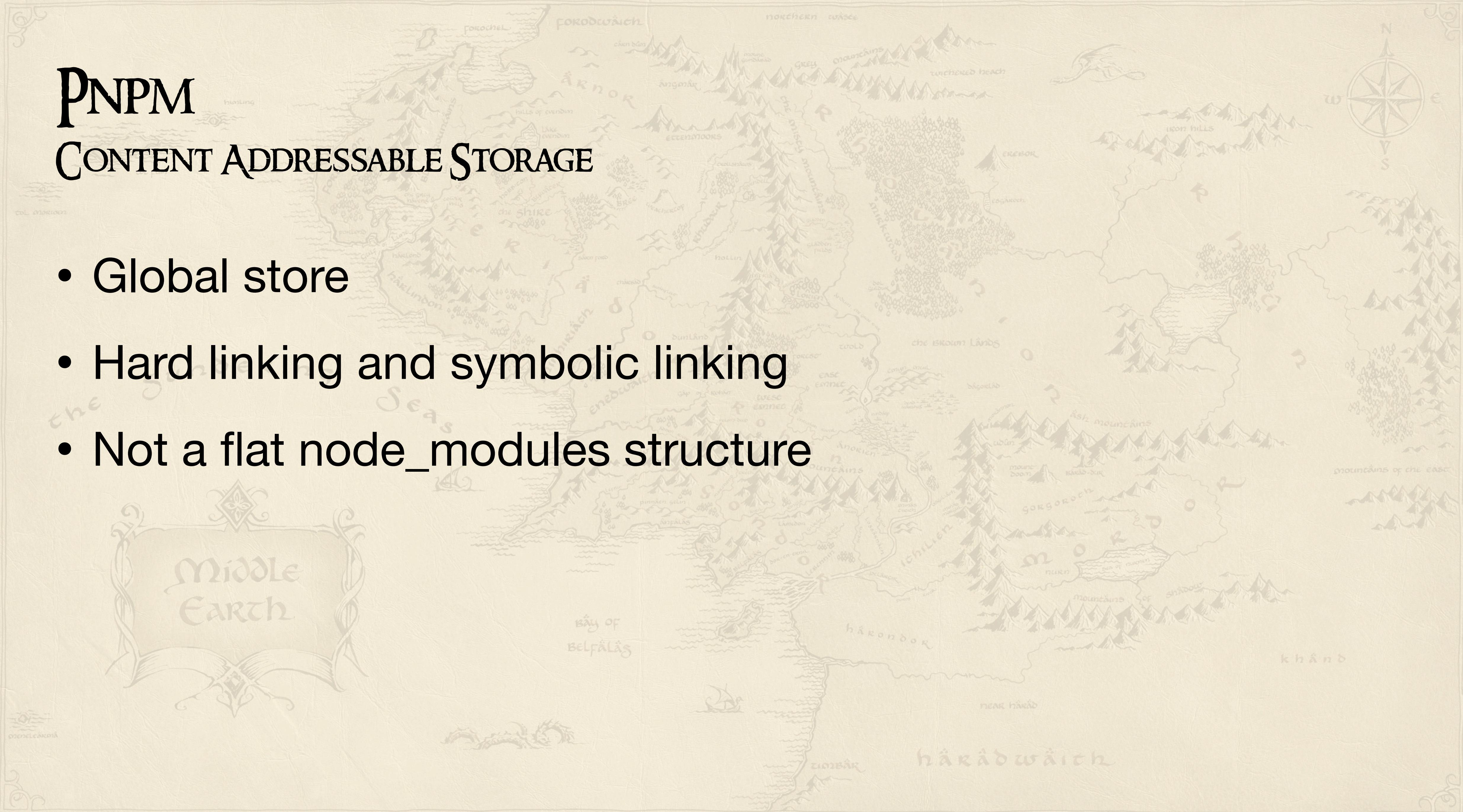
Middle
Earth



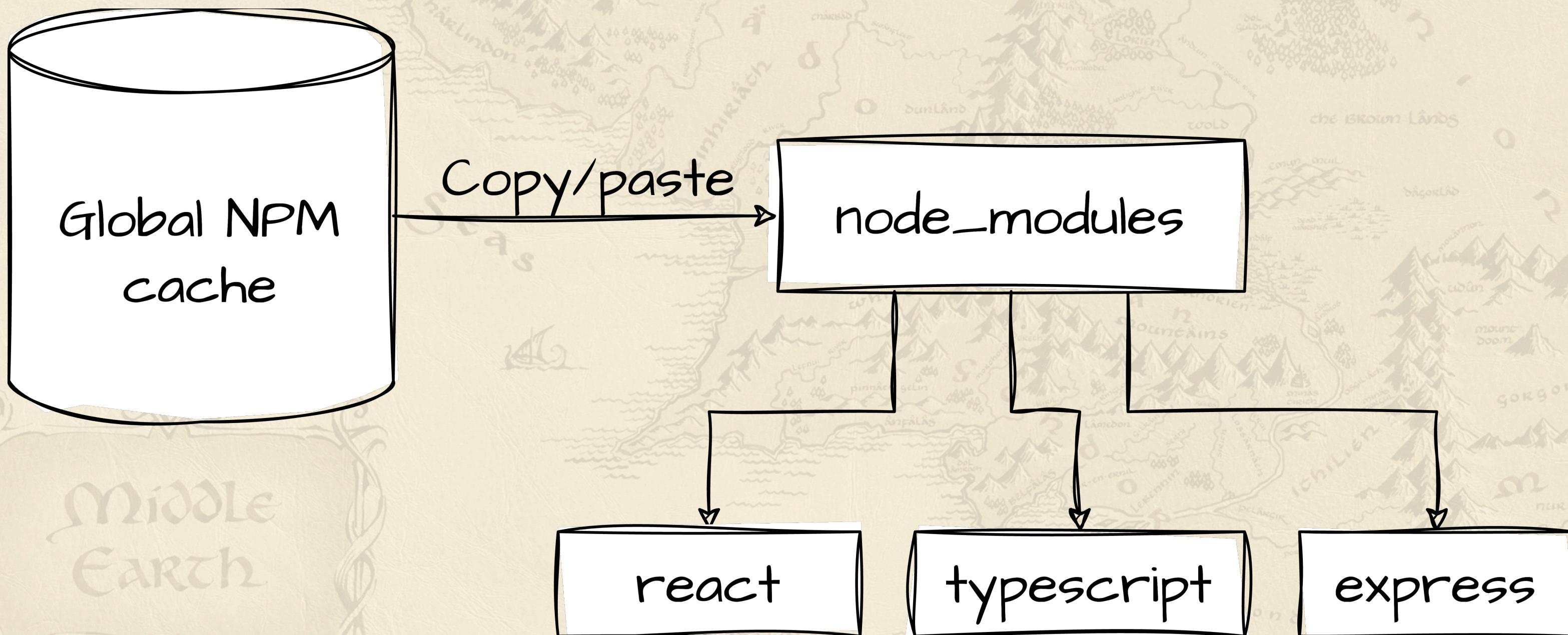
PNPM

CONTENT ADDRESSABLE STORAGE

- Global store
- Hard linking and symbolic linking
- Not a flat node_modules structure



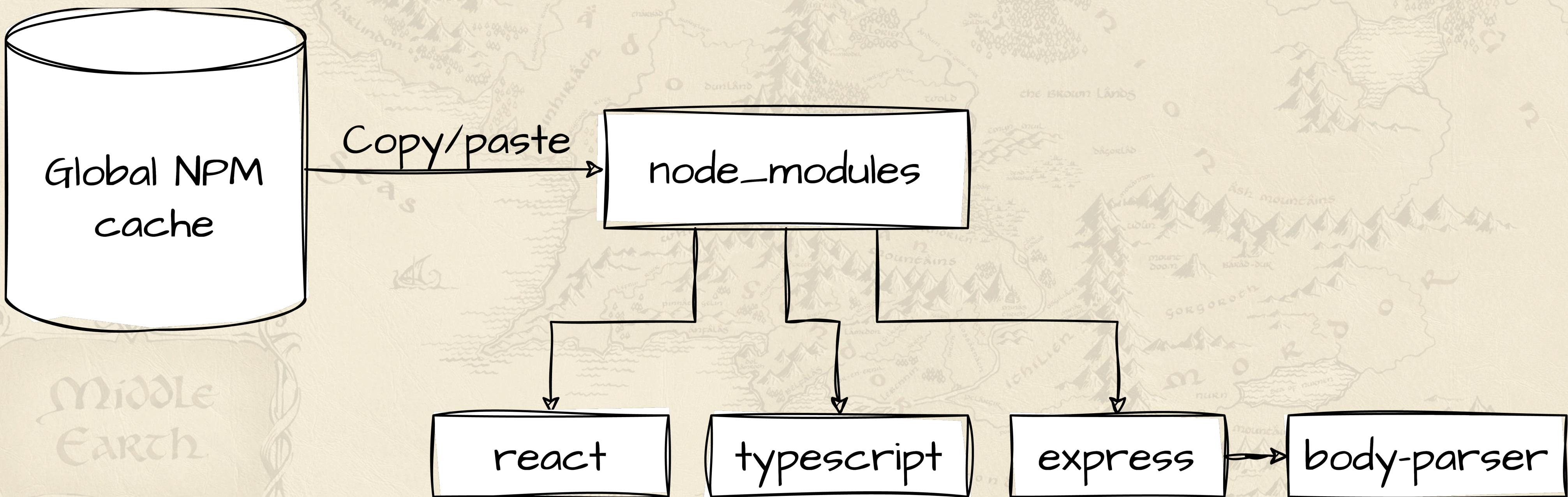
NPM



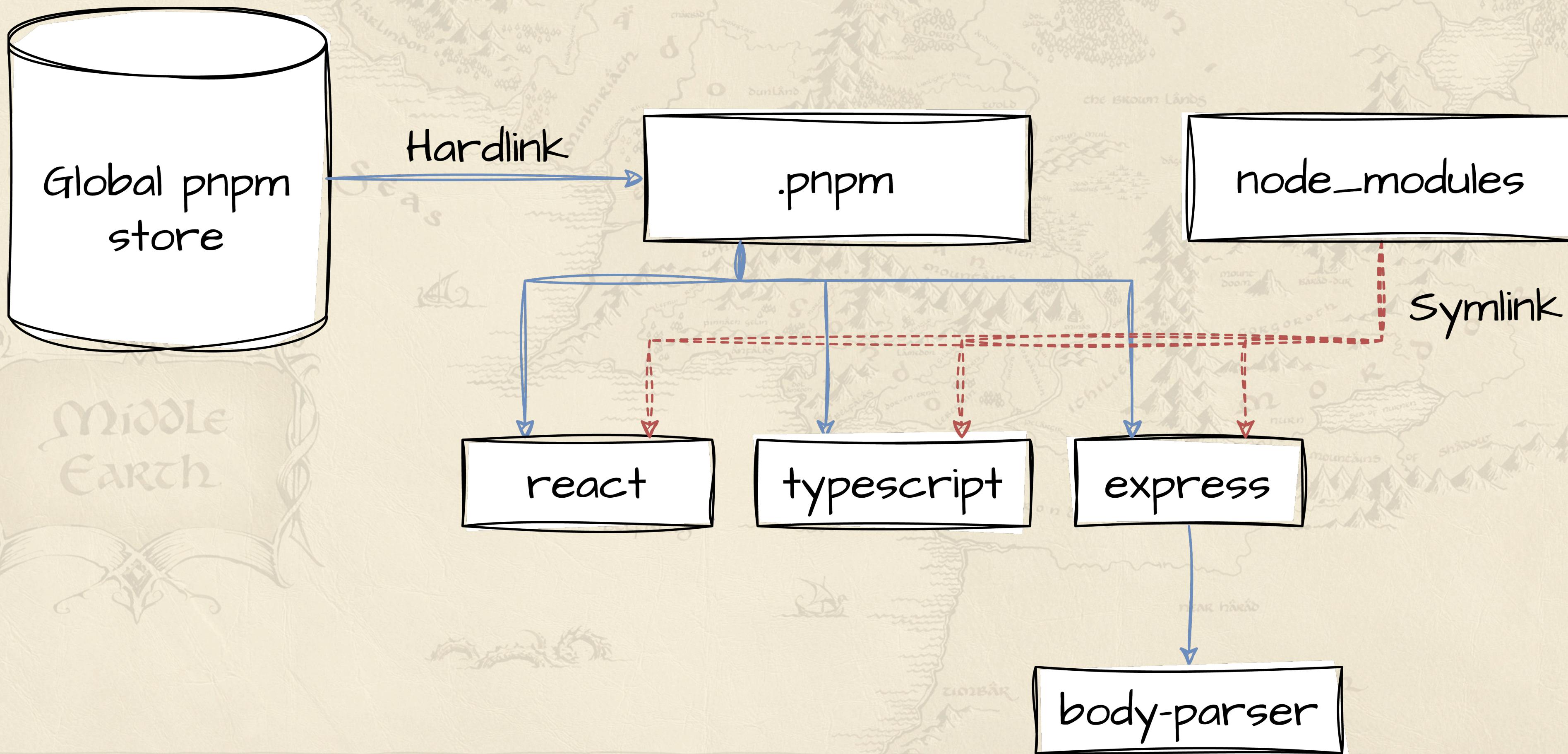


What about second *node_modules*?

NPM



PNPM



NX

“NRWL EXTENSIONS”

Middle
Earth



Nx

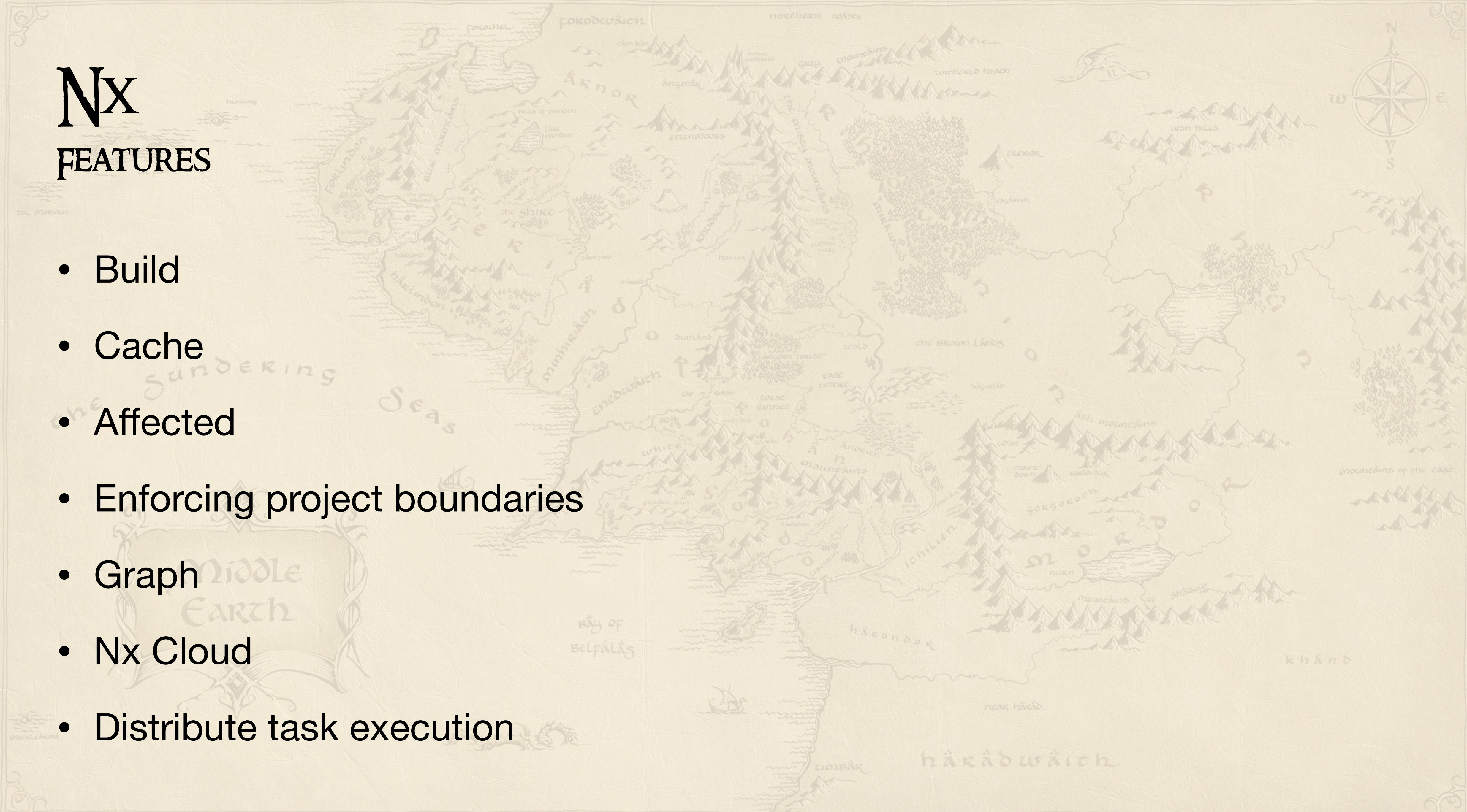
- Inspired by Angular CLI
- Used to have angular.json as config file
- Migration
- Plugins
- Code generators



Nx

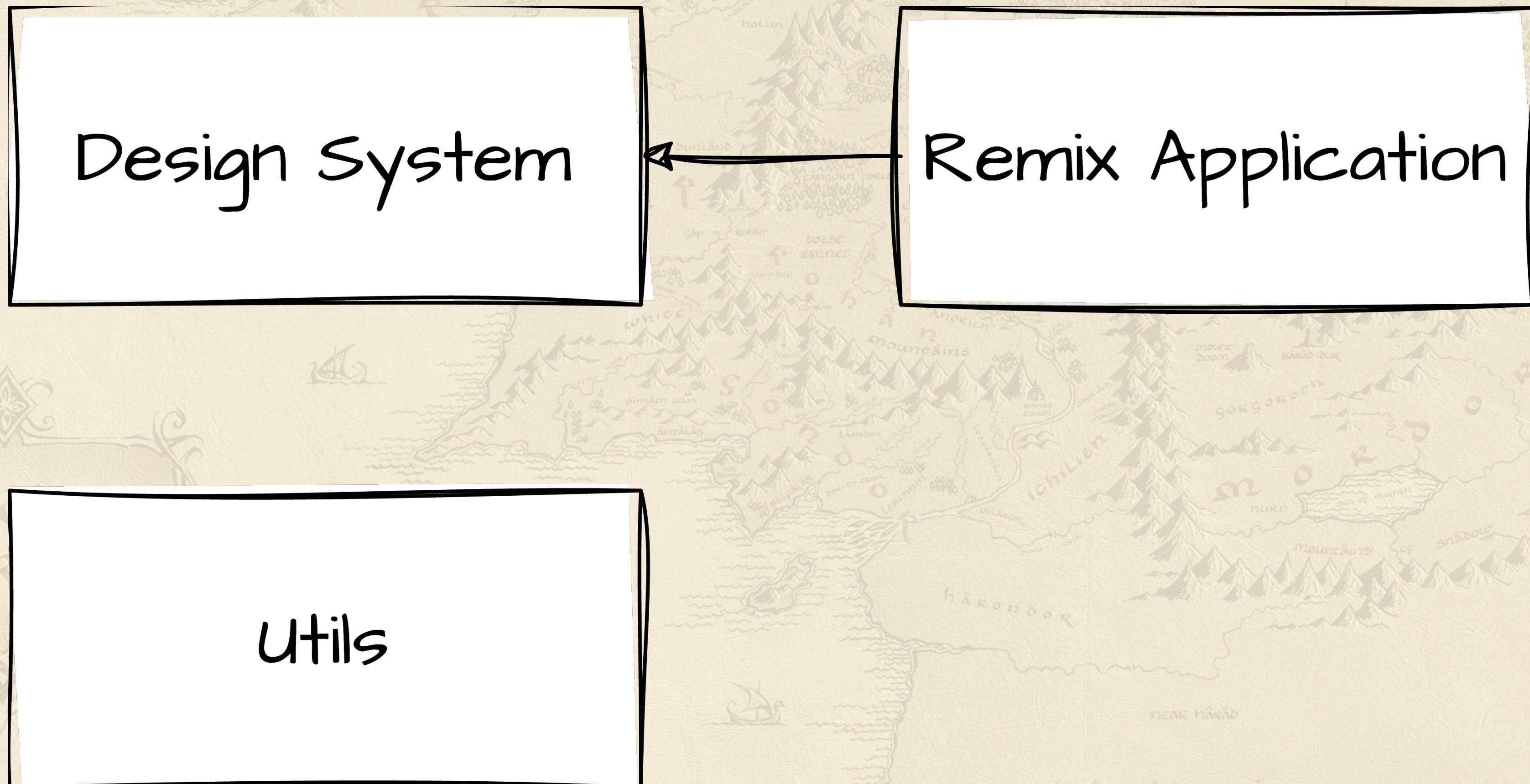
FEATURES

- Build
- Cache
- Affected
- Enforcing project boundaries
- Graph
- Nx Cloud
- Distribute task execution



Nx

OUR FILE STRUCTURE



Nx BUILD

```
> pnpm exec nx build remix
```

```
✓ 1/1 dependent project tasks succeeded [0 read from cache]
```

Hint: you can run the command with `--verbose` to see the full dependent project outputs

```
> nx run remix:build
```

```
> remix@ build /Users/thomas/workspace/perso/react/monorepo-react/packages/remix
```

```
> remix build
```

```
Building Remix app in production mode...
```

```
Built in 228ms
```

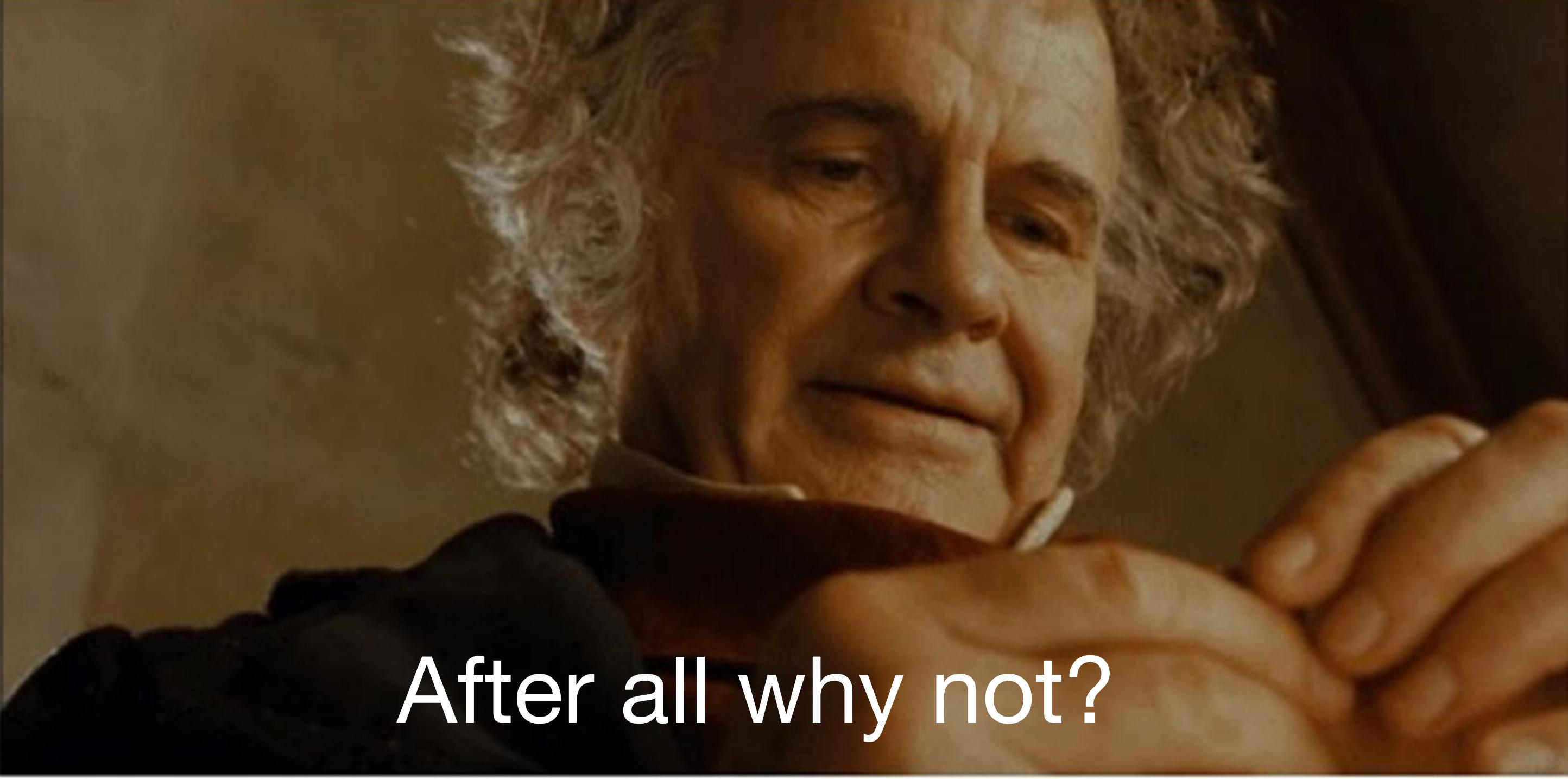
```
> NX  Successfully ran target build for project remix and 1 task(s) it depends on (3s)
```

Nx CACHE

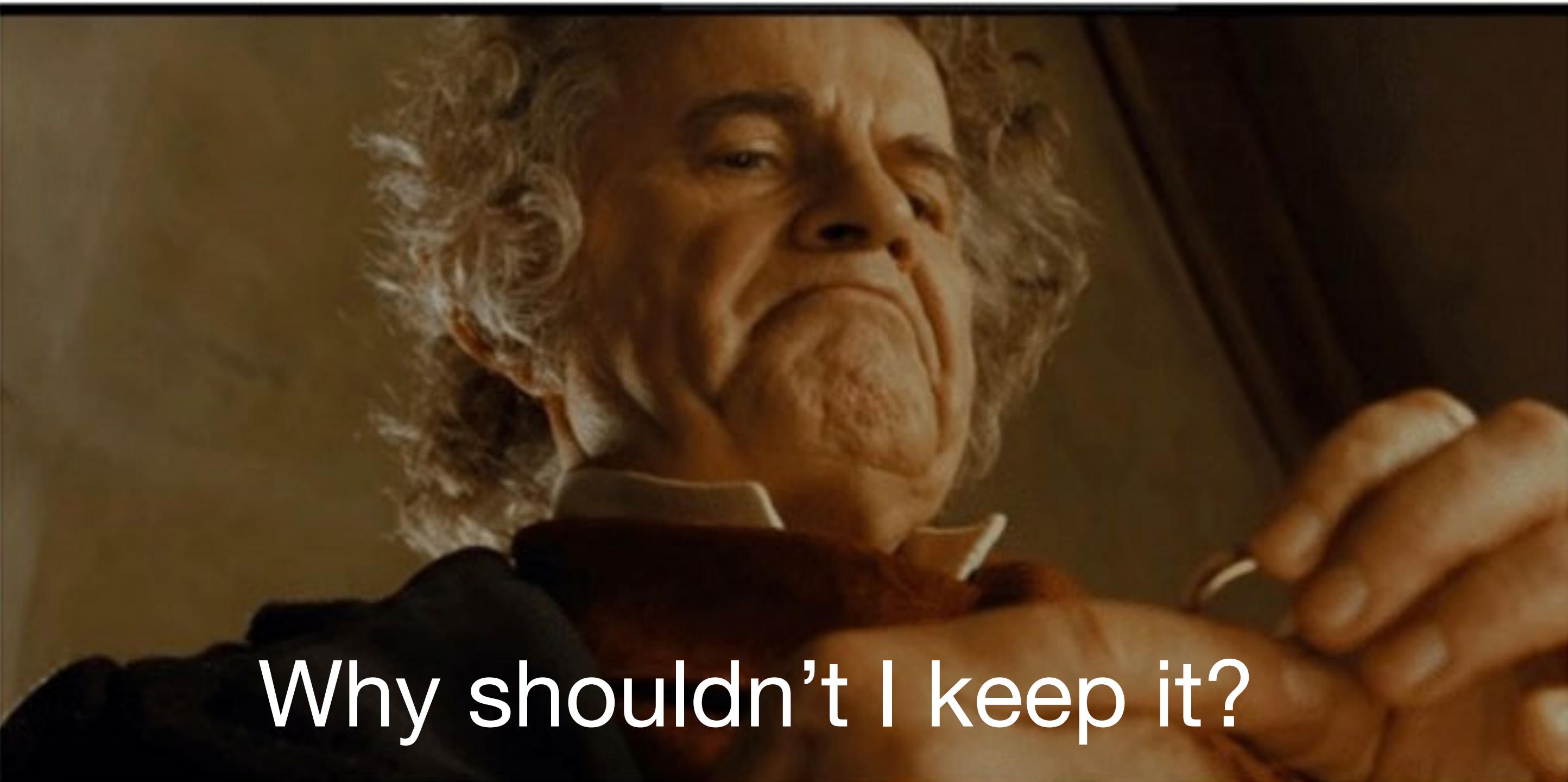
```
> pnpm exec nx run-many --target=build
```

- ✓ nx run design-system:build (2s)
- ✓ nx run utils:build (2s)
- ✓ nx run remix:build (2s)

```
> NX  Successfully ran target build for 3 projects (3s)
```



After all why not?



Why shouldn't I keep it?

Nx CACHE

```
> pnpm exec nx run-many --target=build
```

- ✓ nx run **design-system:build** [existing outputs match the cache, left as is]
- ✓ nx run **utils:build** [existing outputs match the cache, left as is]
- ✓ nx run **remix:build** [existing outputs match the cache, left as is]

```
> NX  Successfully ran target build for 3 projects (38ms)
```

Nx read the output from the cache instead of running the command for 3 out of 3 tasks.

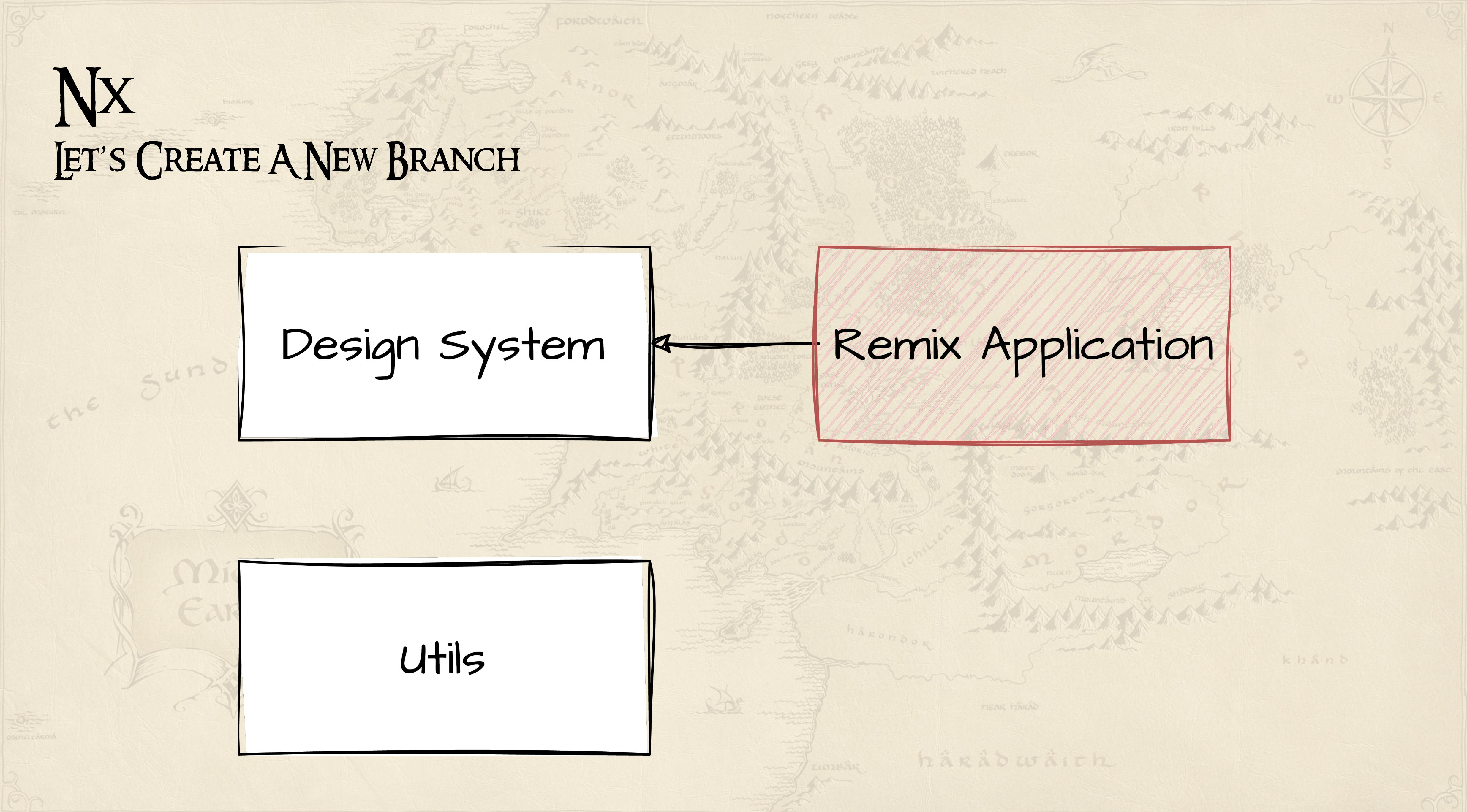
Nx

LET'S CREATE A NEW BRANCH

Design System

Remix Application

utils



Nx AFFECTED

```
nx affected --target=build
```

Nx AFFECTED

```
› nx affected --target=build
```

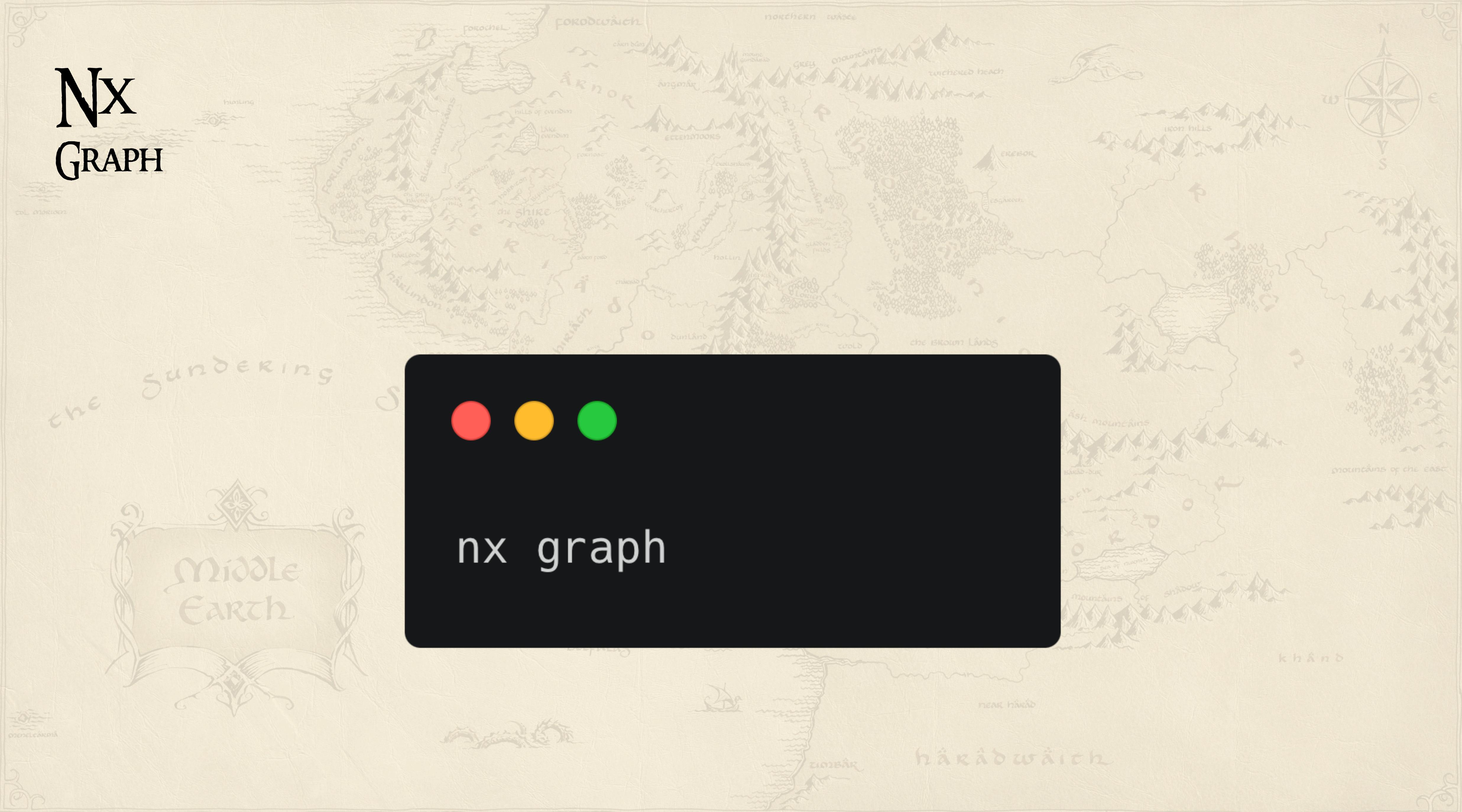
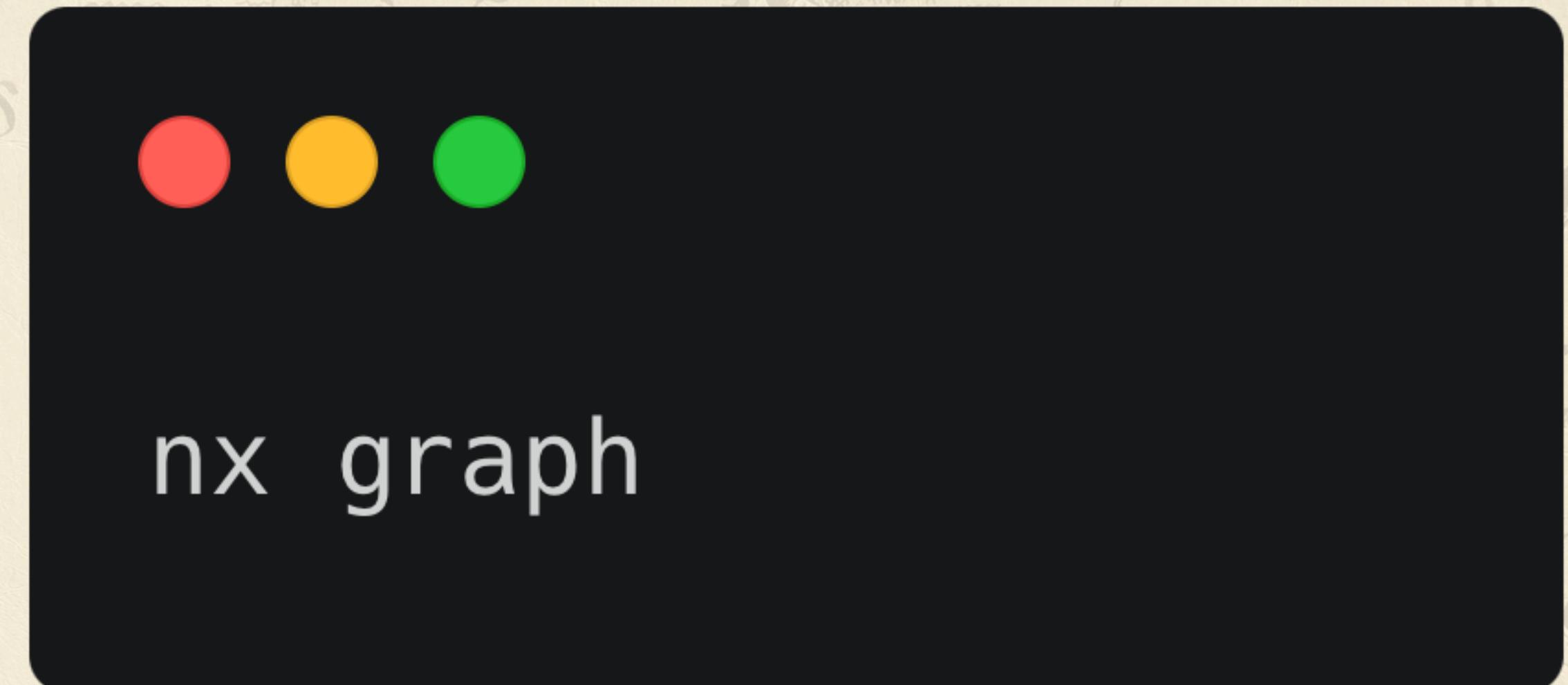
```
> NX  Affected criteria defaulted to --base=main --head=HEAD
```

- ✓ nx run design-system:build [local cache]
- ✓ nx run remix:build (2s)

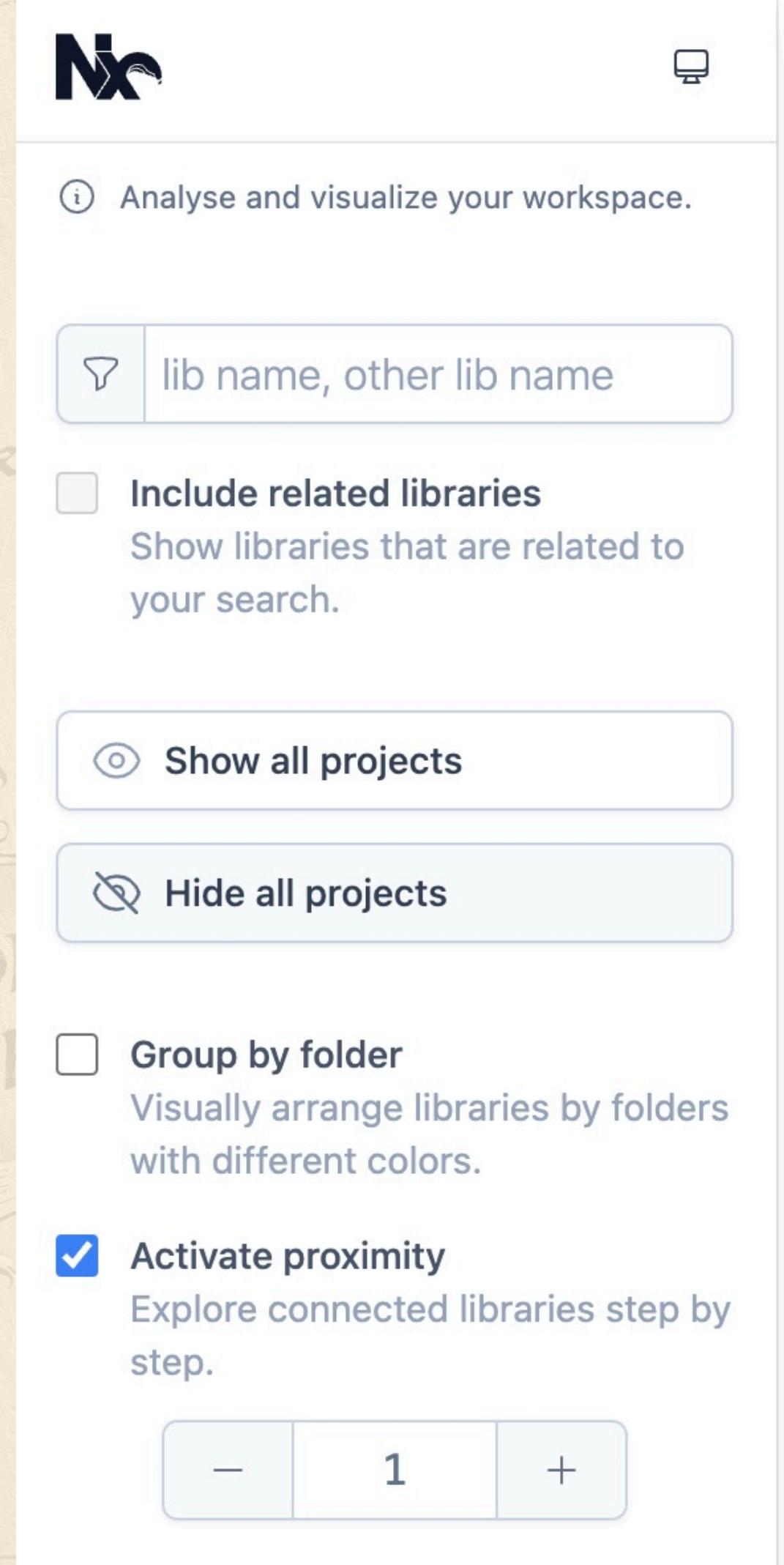
```
> NX  Successfully ran target build for 2 projects (2s)
```

Nx read the output from the cache instead of running the command for 1 out of 2 tasks.

Nx GRAPH



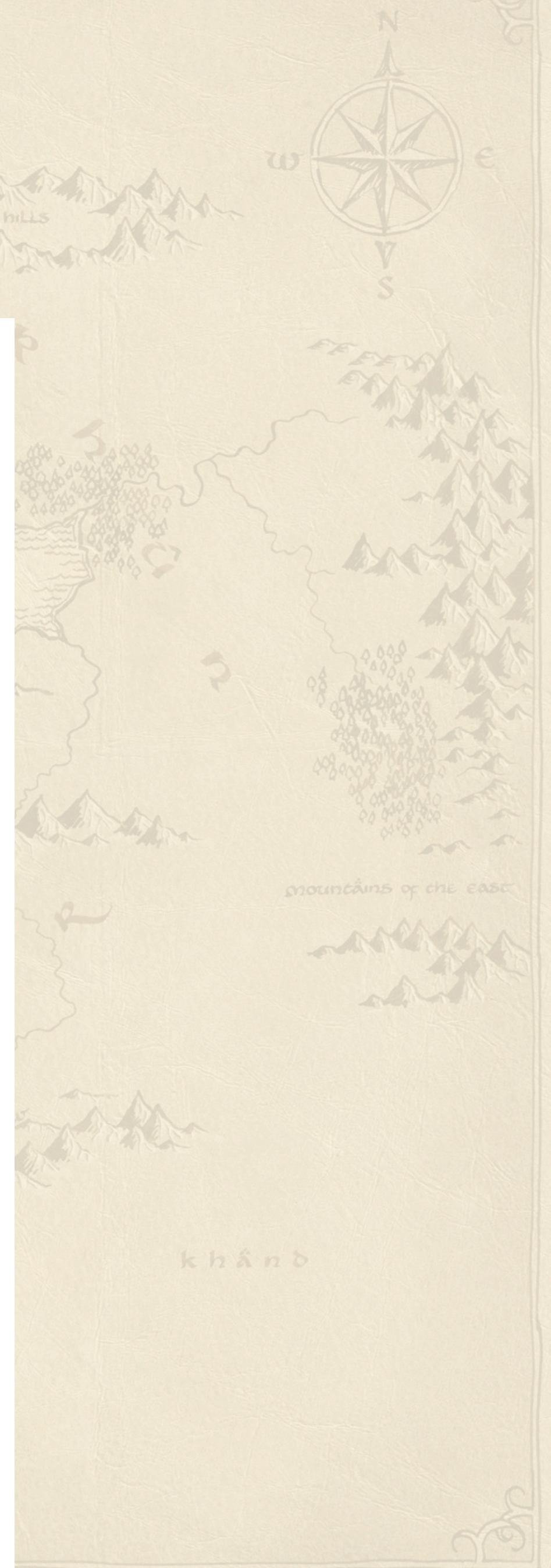
Nx GRAPH



The screenshot shows the Nx Graph interface. At the top left is the Nx logo. To its right are icons for a monitor and a gear. Below these are several buttons and input fields:

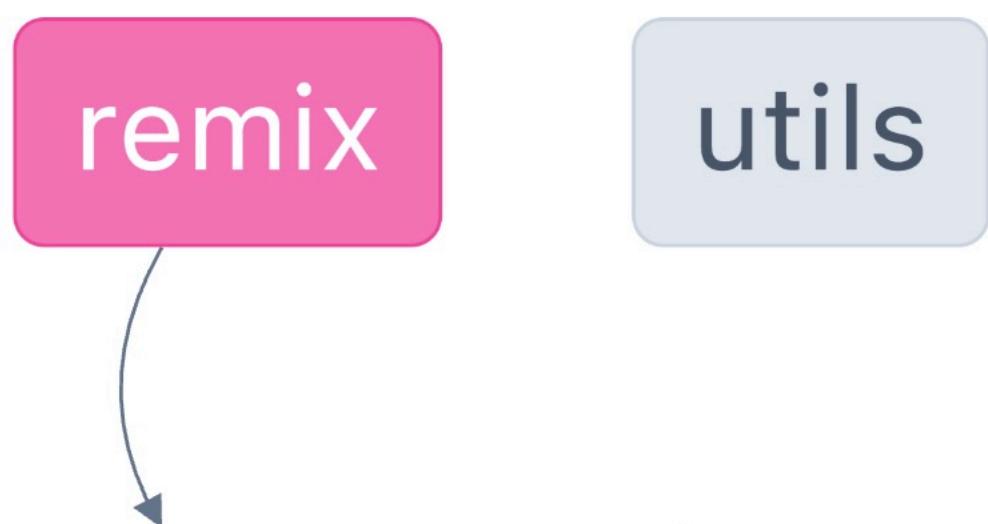
- A button with a magnifying glass icon labeled "lib name, other lib name".
- A checkbox labeled "Include related libraries" with the subtext "Show libraries that are related to your search".
- Two radio buttons: one labeled "Show all projects" and another labeled "Hide all projects".
- A checkbox labeled "Group by folder" with the subtext "Visually arrange libraries by folders with different colors".
- A checked checkbox labeled "Activate proximity" with the subtext "Explore connected libraries step by step".
- Below the checkboxes are three buttons: a minus sign, the number "1", and a plus sign.

On the right side of the interface, there is a large, central button labeled "design-system". Above this button are two smaller buttons: "remix" on the left and "utils" on the right. A curved arrow points from the "remix" button down towards the "design-system" button.

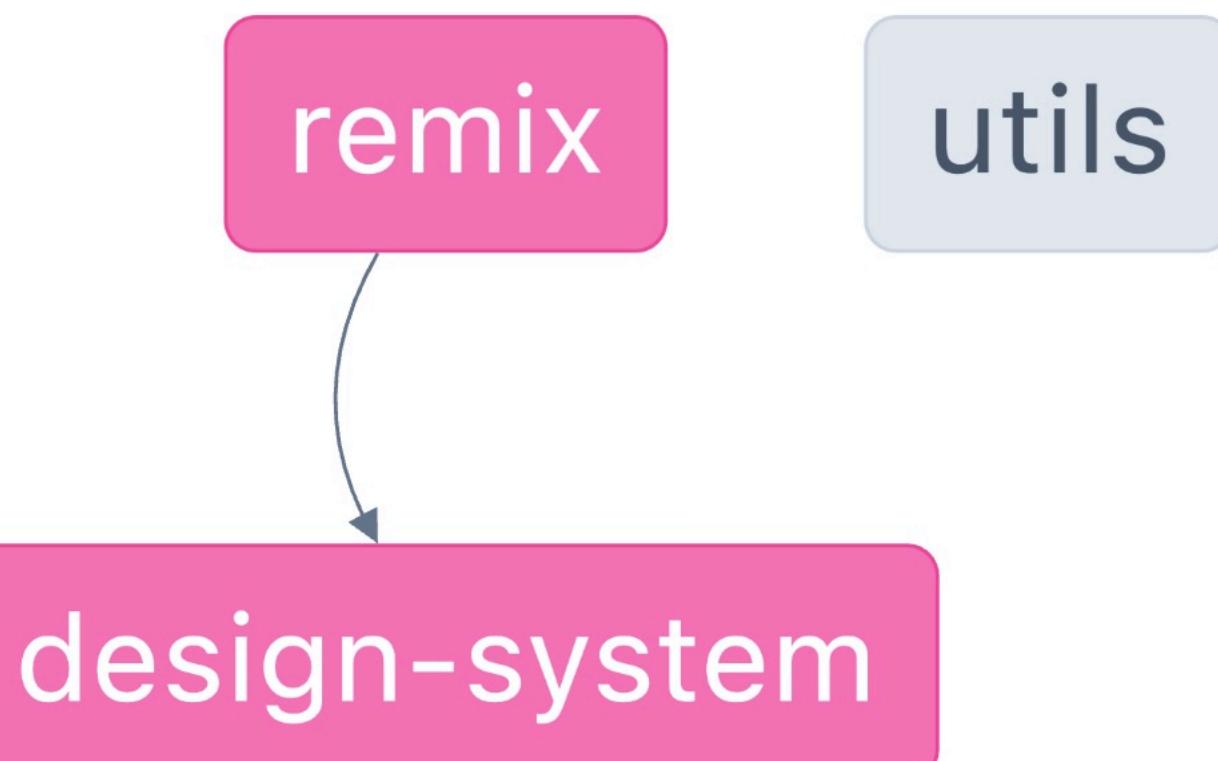


Nx

AFFECTED:GRAPH



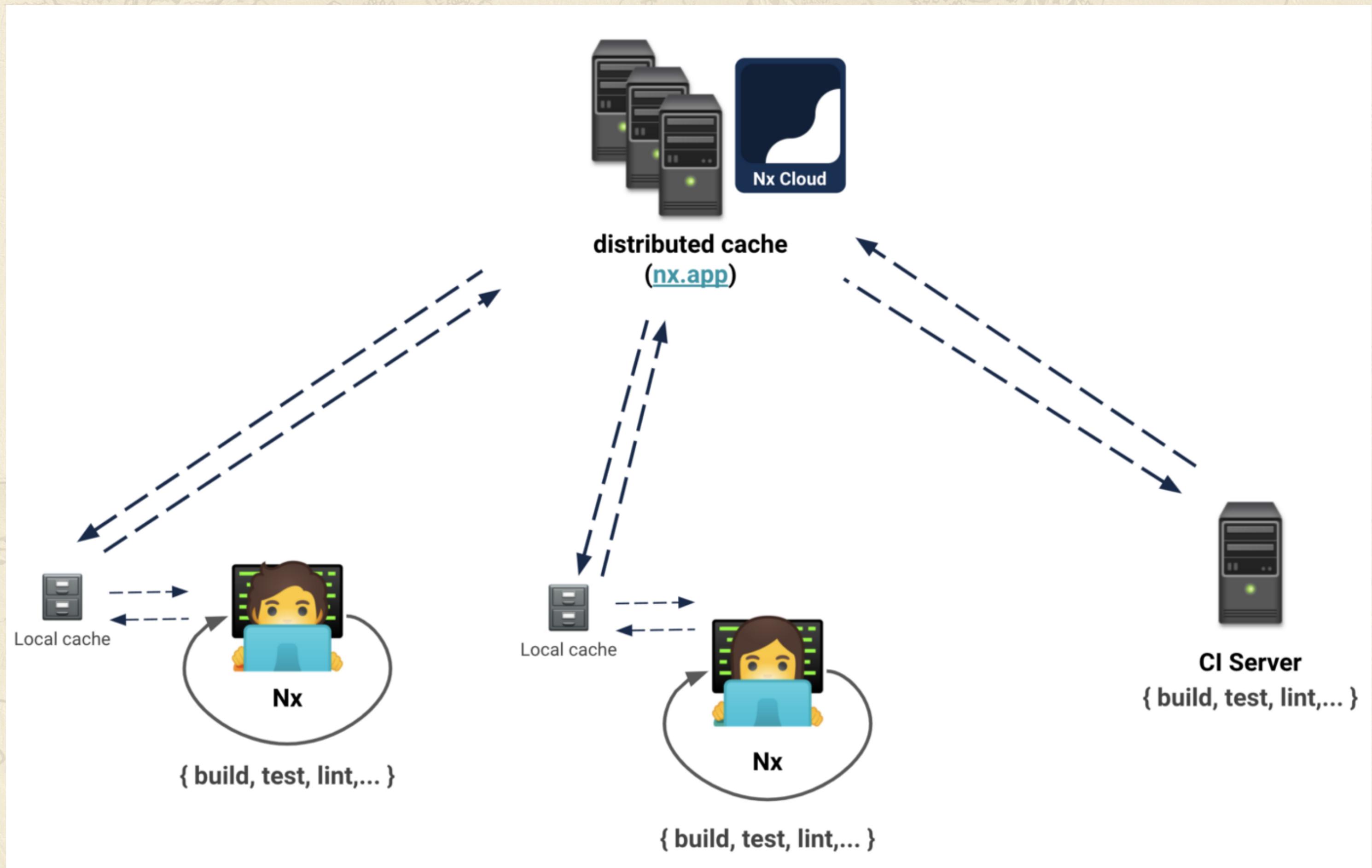
Changes in
remix/



Changes in
design-system/

Nx

Nx CLOUD & DISTRIBUTE TASK EXECUTION



MONOREPO.TOOLS

COMPARING FEATURES

monorepo.tools

Everything you need to know about monorepos,
and the tools to build them.

- Made with love by Nrwl (the company behind Nx)





THE RETURN OF THE KING

THE DEVELOPER EXPERIENCE

REPO STRUCTURE

PNPM



packages:

```
# all packages in direct subdirs of packages/
- 'packages/*'
# all packages in subdirs of components/
- 'components/**'
# exclude packages that are inside test directories
- '!**/test/**'
```

Migration

PNPM



REPO STRUCTURE

Nx

New Package-Based Repo

Create a monorepo with Yarn, NPM or PNPM. Nx makes it fast, but lets you run things your way.

- [Get started with your package-based repo](#)

New Integrated Repo

Get a pre-configured setup. Nx configures your favorite frameworks and lets you focus on shipping features.

- [Get started with your integrated repo](#)

Add to an Existing Monorepo

 Incrementally add Nx to your repo and enjoy faster CI runs and local development. All without modifying your existing setup.

- [Add Nx to an Existing Monorepo](#)

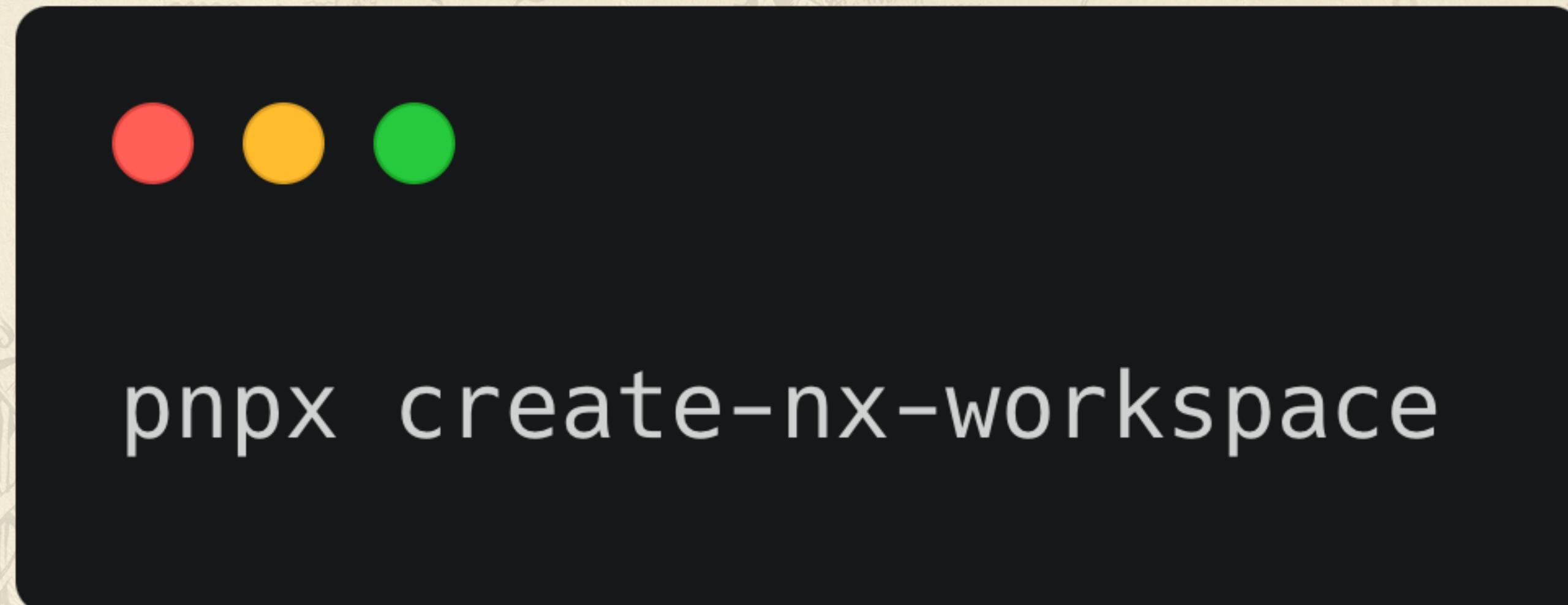
Enter Modern Angular

Enhance your Angular development experience by leveraging its advanced generators and integrations with modern tooling.

- [Switch from the Angular CLI to Nx](#)

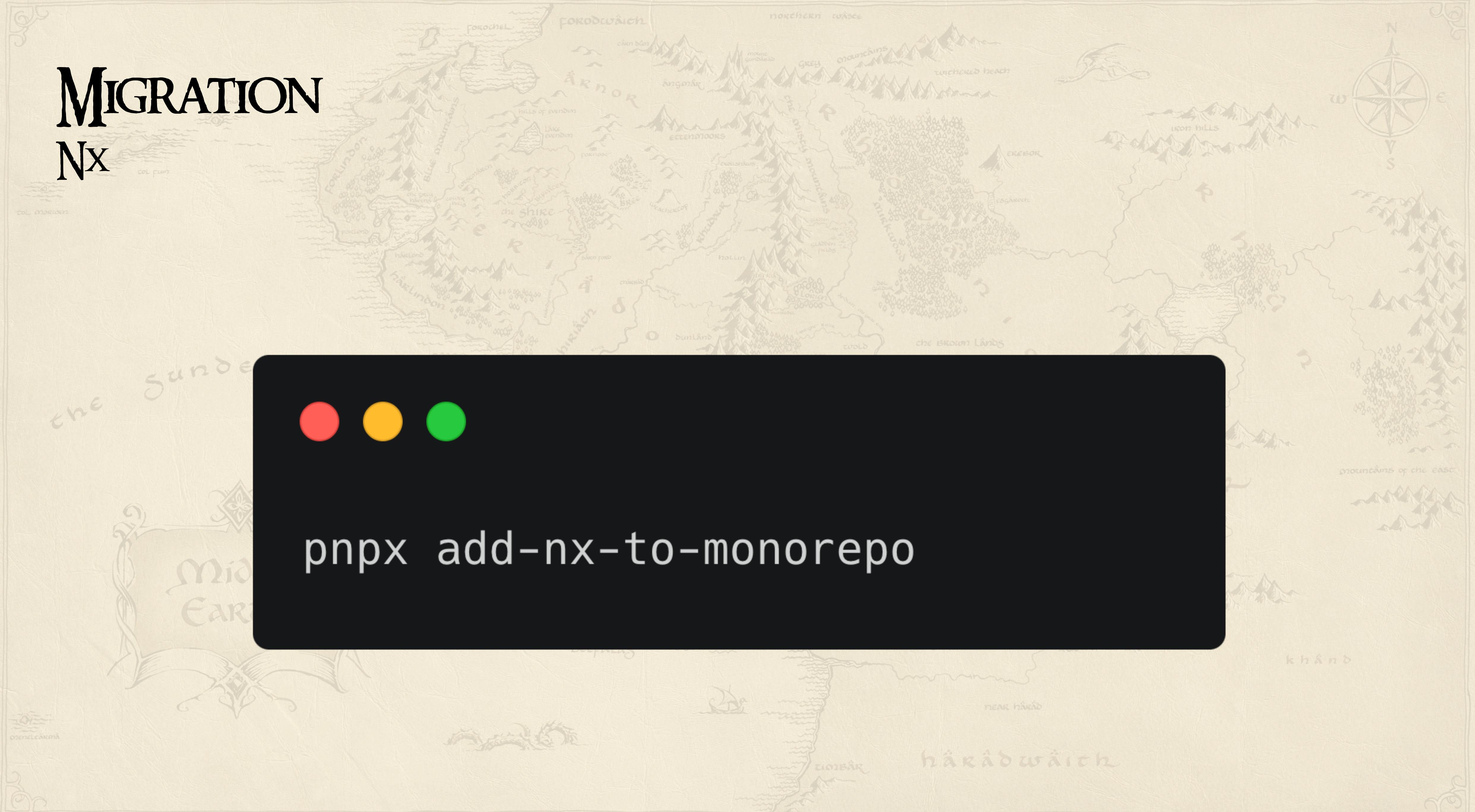
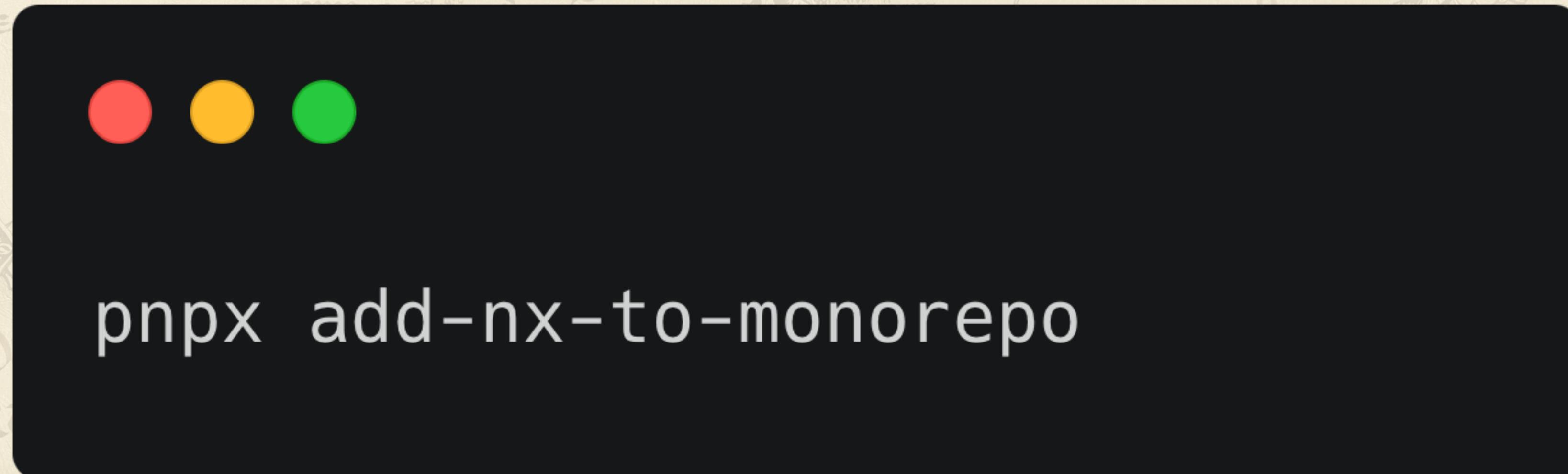
FRESH INSTALL

Nx



Migration

Nx



Nx PLUGINS



@nrwl/angular

The Nx Plugin for Angular contains executors, generators, and utilities for managing Angular applications and libraries within an Nx workspace. It...

Official

@nrwl/cypress

The Nx Plugin for Cypress contains executors and generators allowing your workspace to use the powerful Cypress integration testing capabilities.

Official

@nrwl/detox

The Nx Plugin for Detox contains executors and generators for allowing your workspace to use the powerful Detox integration testing capabilities.

Official

@nrwl/devkit

The Nx Devkit is used to customize Nx for different technologies and use cases. It contains many utility functions for reading and writing files, updating...

Official

@nrwl/esbuild

The Nx Plugin for esbuild contains executors and generators that support building applications using esbuild

Official

@nrwl/eslint-plugin-nx

The ESLint Plugin for Nx contains a collection of recommended ESLint rule configurations wrapped as ESLint plugins and an Nx specific enforce-module...

Official

@nrwl/expo

The Expo Plugin for Nx contains executors and generators for managing and developing an expo application within your workspace. For example, you...

Official

@nrwl/express

The Nx Plugin for Express contains executors and generators for allowing your workspace to create powerful Express Node applications and APIs.

Official

@nrwl/jest

The Nx Plugin for Jest contains executors and generators allowing your workspace to use the powerful Jest testing capabilities.

Official

Nx PLUGINS



@nrwl/linter

The Linter plugin for Nx contains executors, generators and utilities used for linting JavaScript/TypeScript projects within an Nx...

Official

@nrwl/nest

The Nx Plugin for Nest contains executors and generators for allowing your workspace to create powerful Nest best in class APIs.

Official

@nrwl/next

The Next.js plugin for Nx contains executors and generators for managing Next.js applications and libraries within an Nx workspace. It provides: -...

Official

@nrwl/node

The Node Plugin for Nx contains generators and executors to manage Node applications within an Nx workspace.

Official

@nrwl/react

The React plugin for Nx contains executors and generators for managing React applications and libraries within an Nx workspace. It provides: -...

Official

@nrwl/react-native

The Nx Plugin for React Native contains generators for managing React Native applications and libraries within an Nx workspace. It provides: -Integration wit...

Official

@nrwl/rollup

The Nx Plugin for Rollup contains executors and generators that support building applications using Rollup.

Official

@nrwl/storybook

The Nx Plugin for Storybook contains executors and generators for allowing your workspace to use the powerful Storybook integration testing &...

Official

@nrwl/web

The Nx Plugin for Web Components contains generators for managing Web Component applications and libraries within an Nx workspace. It...

Official

PERFORMANCE

yarn 27.40s user 35.62s system 155% cpu 40.578 total

releasable space: 1.79 GB

pnpm install --recursive 10.74s user 71.87s system 274% cpu 30.080 total

releasable space: 0.84 GB

PERFORMANCE

Nx is done in 0.46
seconds

Turbo is done in 4.4
seconds

```
316 B          71.6 kB
warp-drive-manager:build: | o /important-feature-4
315 B          71.6 kB
warp-drive-manager:build: | o /important-feature-5
315 B          71.6 kB
warp-drive-manager:build: | o /important-feature-6
316 B
warp-
315 B
warp-
315 B
warp-
315 B
warp-
71.3
warp-drive-manager:build:   |- chunks/framework-c4190dd27fdc6a34.js
42 kB
warp-drive-manager:build:   |- chunks/main-93fae2578c0fbe55.js
27.8 kB
warp-drive-manager:build:   |- chunks/pages/_app-18b1df30a7622aa9.js
636 B
warp-drive-manager:build:   |- chunks/webpack-514908bffb652963.js
770 B
warp-drive-manager:build:     L css/1cca4d23b6c403c9.css
748 B
warp-drive-manager:build:
warp-drive-manager:build: o (Static) automatically rendered as static HTML (uses no initial props)
warp-drive-manager:build:
warp-drive-manager:build: warn - No ESLint configuration detected. Run next lint to begin setup

Tasks:      5 successful, 5 total
Cached:    5 cached, 5 total
Time:       4.158s >>> FULL TURBO
```

npx turbo run build 5.78s user 6.92s system 290% cpu 4.377 total
vsaykin@mbp large-monorepo %

Docs

The screenshot shows the pnpm documentation page. At the top, there is a navigation bar with links for pnpm, Docs, Blog, FAQ, Community, Benchmarks, version 7.x, language English, sponsorship options, and search functionality. The main title is "pNPM" in large white letters, followed by the subtitle "Fast, disk space efficient package manager". Below this, there is a "Get Started" button and a star icon indicating 20,068 stars. A video thumbnail at the bottom left shows a person switching from Npm/Yarn to pnpm, with a "Regarder sur YouTube" button below it.

pNPM

Fast, disk space efficient package manager

Get Started

Star 20,068

Why I Switched From NPM/Yarn to PNPM And ...
Npm/Yarn → Pnpm Copier le li...

Regarder sur YouTube

Docs



Documentation ▾

Enterprise

Community

Plugins registry

Nx Conf



Smart, Fast Extensible Build System

Next generation build system with first class monorepo support and powerful integrations.

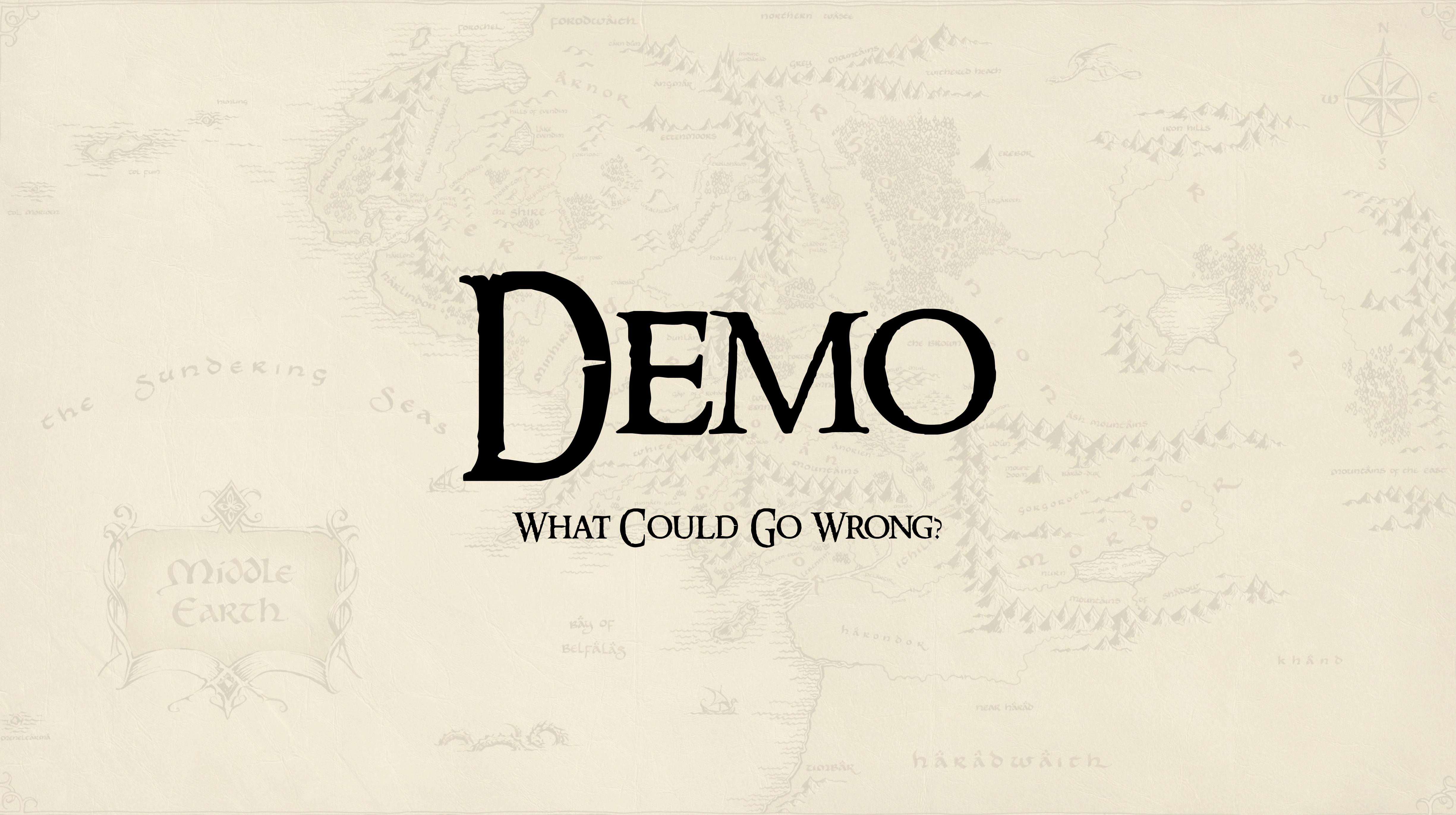
Get started

> `npx create-nx-workspace`

DEMO

WHAT COULD GO WRONG?

Middle
Earth





It's over. It's done.

SOURCES

- <https://pnpm.io>
- <https://nx.dev>
- <https://monorepo.tools>
- <https://blog.nrwl.io/setup-a-monorepo-with-pnpm-workspaces-and-speed-it-up-with-nx-bc5d97258a7e>
- <https://github.com/vsavkin/large-monorepo>
- Lovely arts by Ise Ananphada and Grzegorz Domaradzki

Thank you!

