



SPEAKER'S
CORNER

JUNE, 20

17:30 (CET)

ZOOM

PRACTICAL MONOREPO WITH NX



TOMÁŠ HORÁČEK
PRINCIPAL TECHNOLOGY LEAD
FOR WEB & MOBILE AT CIKLUM

Meet Ciklum



We empower companies to meet their digital initiatives by providing end-to-end software, integration and innovation services

Our services:

Digital
Commerce

Intelligent
Automation

ITO & Managed
Service

Data &
Analytics

Cloud

Engineering
Services

2002
founded

4000+
professionals

20+
offices

300+
clients

Leading companies choose us:



Greensill

KANTAR RETAIL



Speaker:

Tomáš Horáček

Principal Technology Lead for Web & Mobile

-
- Works with JavaScript since 2005 (IE6 + jQuery ages 😅)
 - Experienced with many web/mobile-related languages since then: Python, Ruby, Objective-C, Swift, Flow, Elm, and TypeScript
 - Keeps his primary focus on React, React Native, TypeScript and functional programming since 2015
 - Migrated 3 large projects to Nx monorepo in the last 2 years



Table of Contents

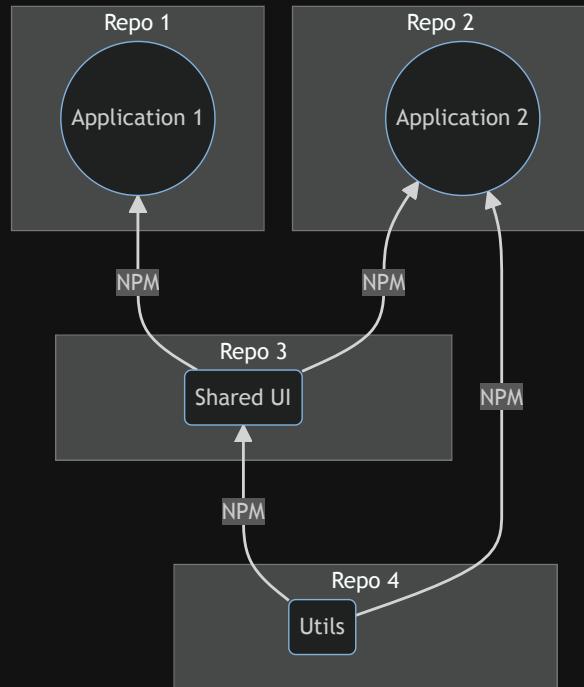
1. Monorepo 
2. Nx 
3. Demo 1  - Migrate App to Nx
4. Demo 2  - Break a Monolith to Smaller Parts
5. CI/CD Setup 

1) Monorepo

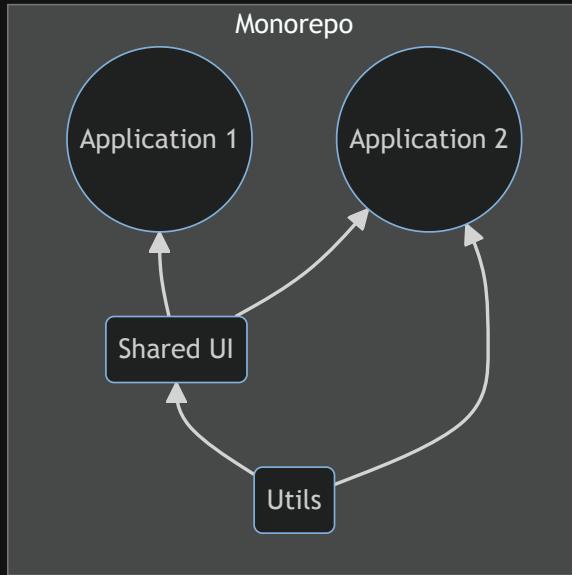


What is Monorepo? 🍱

Multi-Repo 🍔, 🍜, 🍲



Monorepo



Monorepo 🌿 != Monolith 🛡



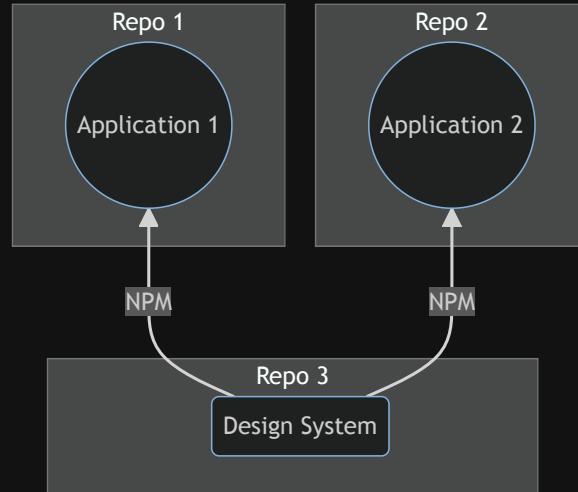
Monolith?

-  property of **code**
-  not property of **version control**
-  **monorepo tooling** will work with this

Why Monorepo? 🍱 🤔

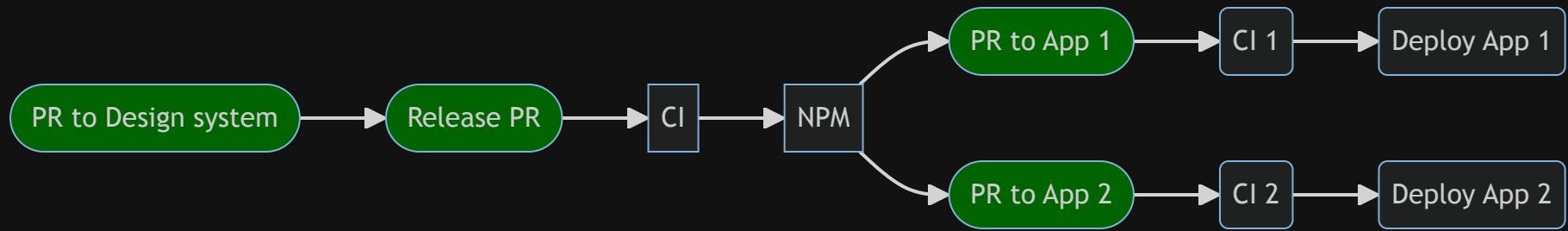
- #1: 🎁 Shared Code for Multiple Projects
- #2: ✂️ Split Monolith to Smaller Parts

#1 🎁 Shared Code



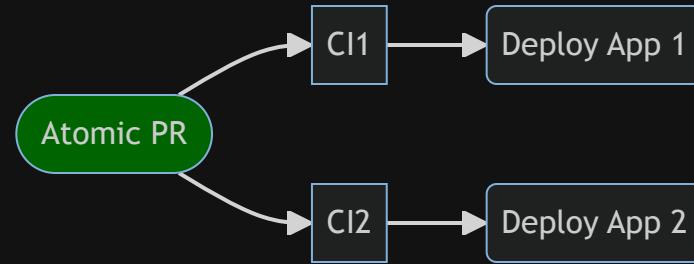
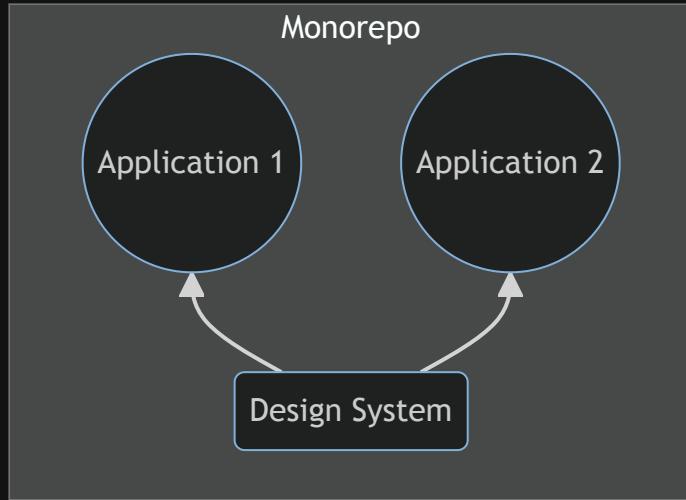


Problem?



+1 Day of Bureaucracy

Monorepo



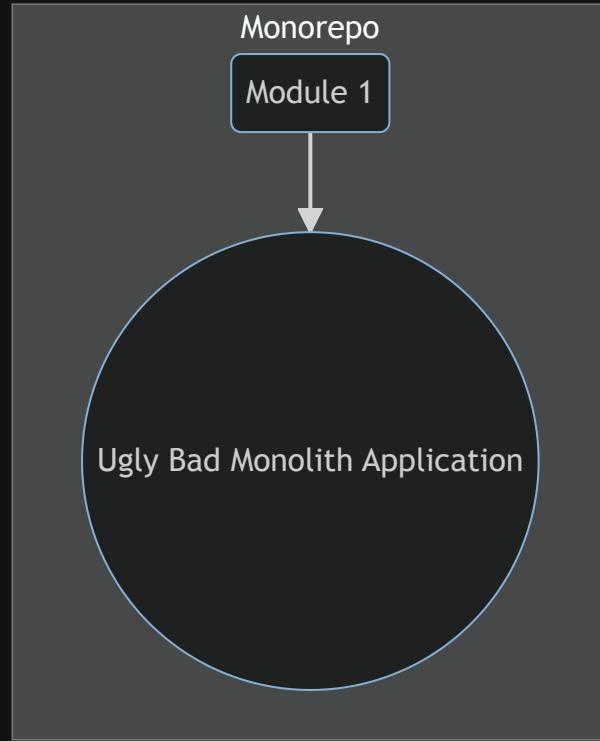
Less Bureaucracy

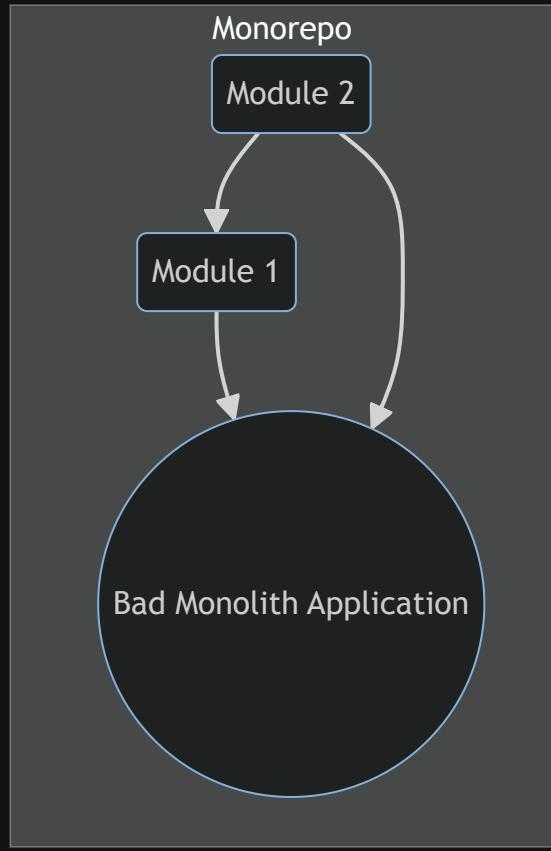
#2: 💥 Split monolith to smaller parts

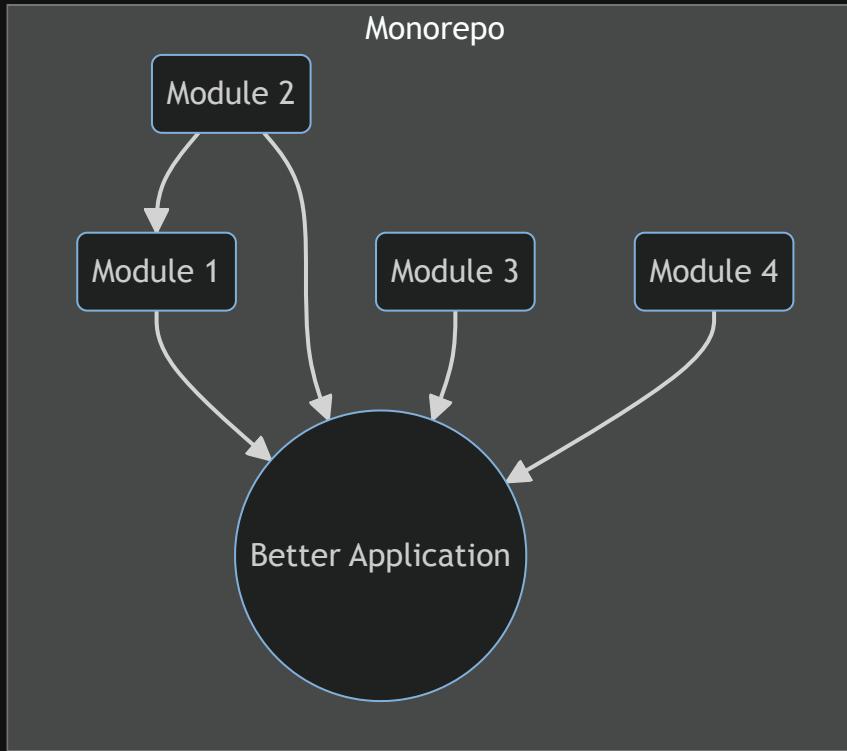


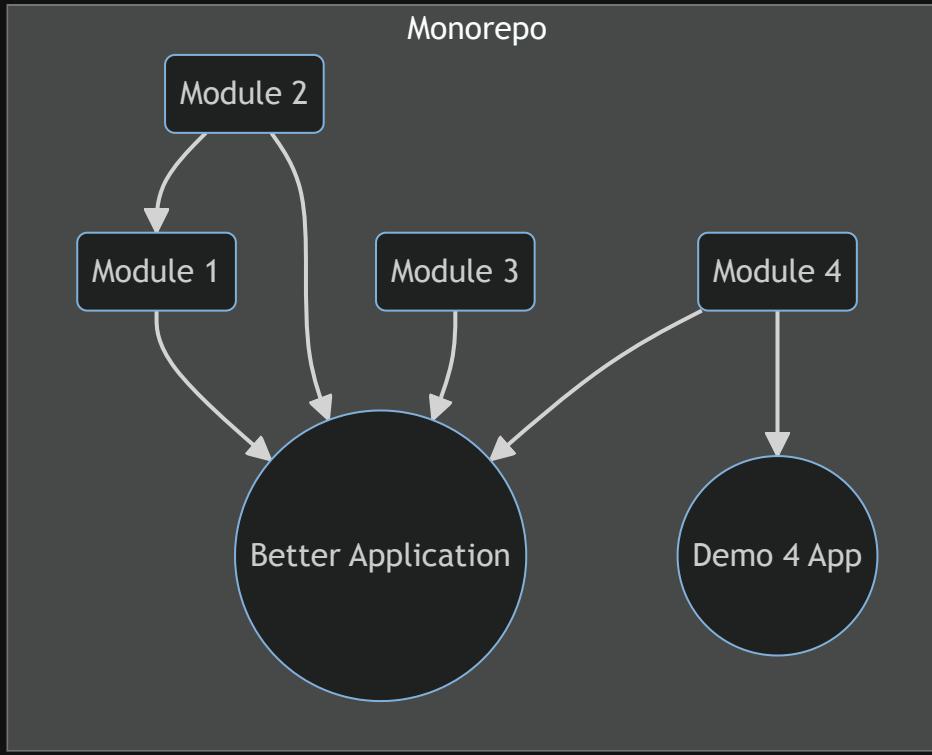
- 🌎 No clear boundaries
- 🐌 Compilation is slow













More About Monorepo



<https://monorepo.tools/>

2) Nx 🐋





- monorepo tooling: <https://nx.dev/>
- plugins
 - JS: React, Vue, Angular, Svelte, Express, Vite, Rollup, Jest, Cypress, ...
 - Go, .Net, Docker, ...
 - community plugins: Rust, Java, ...
- std commands for: **build, test, lint, format**
- generators
- dependency graph
- distributed cache
- IDE plugins (VS Code, WebStorm)

3) Demo 1



Migrate App to Nx

<https://github.com/heracek/practical-monorepo-nx>



Quacker App

<https://github.com/heracek/practical-monorepo-nx>

- we will work on `main`
- branch `initial-state` - "Demo 1" starting point

- ```
cd backend
yarn
yarn dev
```

- ```
cd frontend
yarn
yarn start
```

- branch `final` - "Demo 2" final state

```
yarn
yarn nx graph --open=false --port=8888
```

Steps - Migrate App to Nx

1. generate empty Nx workspace
2. add Nx plugins
3. generate empty Nx apps (backend + fronted)
4. move backend to Nx  BACK
5. move frontend to Nx 

⚠ Step 1. - generate empty Nx workspace

```
npx create-nx-workspace@16.3.2 --packageManager=yarn
```

Or:

```
npx create-nx-workspace@latest
```

Prompts:

```
? Where would you like to create your workspace? > quacker
? Which stack do you want to use? > None
? Package-based or integrated? > Integrated
? Enable distributed caching to make your CI faster > No
```

Step 2. add Nx plugins

```
yarn add --dev @nx/react@16.3.2 @nx/express@16.3.2 @nx/node@16.3.2
```

Or:

```
yarn add --dev @nx/react @nx/express @nx/node
```

Init plugins:

```
yarn nx g @nx/express:init  
yarn nx g @nx/react:init
```

```
yarn nx list
```

⚠ Step 3. - Generate Express.js App ⏪

```
yarn nx g @nx/express:app --help
```

generate Express.js app:

```
yarn nx g @nx/express:app quacker-backend --dryRun
```

```
yarn nx g @nx/express:app quacker-backend
```

⚠ Step 3. - Generate Frontend App 🎨

```
yarn nx g @nx/react:app --help
```

generate React app:

```
yarn nx g @nx/react:app quacker-frontend \
--bundler=webpack --style=css --routing=false --pascalCaseFiles --dryRun
```

```
yarn nx g @nx/react:app quacker-frontend \
--bundler=webpack --style=css --routing=false --pascalCaseFiles
```

⚠ Step 3.

```
yarn nx serve quacker-backend
```

```
yarn nx serve quacker-frontend
```

⚠ Step 4. move backend to Nx ← BACK - Packages

<https://stedolan.github.io/jq/>

```
jq -r '.dependencies | [ keys[] as $k | "\($k)\(.[$k])" ] | join(" ")' backend/package.json
```

```
yarn add apollo-server-core@3.12.0 apollo-server-express@3.12.0 argon2@0.30.3 body-parser@1.20.2 cors@2.8.5 \
dotenv-flow@3.2.0 express@4.18.2 graphql@^15.8.0 jsonwebtoken@9.0.0 mariadb@2.5.6
```

⚠ Step 4. move backend to Nx - Code

```
cd backend/src

git mv * .. / .. /apps/quacker-backend/src/
cd ..
git mv db .gitignore README.md .. /apps/quacker-backend

cd .. /apps/quacker-backend/src

git rm main.ts
git mv index.ts main.ts

cd .. / .. / ..
rm -rf ./backend
```

```
# apps/quacker-backend/.env
PORT=4000
JWT_SECRET=somesuperlongsecret
MOCKS=true
DB_HOST=localhost
DB_NAME=
DB_USER=
DB_PASSWORD=
DB_PORT=3306
```

⚠ Step 5. move frontend to Nx 🌈 - Packages

<https://stedolan.github.io/jq/>

```
jq -r '.dependencies | [ keys[] as $k | "\($k)\(.[$k])" ] | join(" ")' frontend/package.json
```

```
yarn add @apollo/client@^3.5.10 @fortawesome/fontawesome-svg-core@6.4.0 @fortawesome/free-regular-svg-icons@6.4.0 \
@fortawesome/free-solid-svg-icons@6.4.0 @fortawesome/react-fontawesome@0.2.0 @testing-library/user-event@^13.5.0 \
@types/react-router-dom@^5.3.3 classnames@2.3.2 date-fns@2.30.0 formik@2.4.2 graphql@^15.8.0 lodash@4.17.21 \
react@18.2.0 react-dom@18.2.0 react-router-dom@6.13.0 tachyons@4.12.0 yup@0.32.11
```

⚠ Step 5. move frontend to Nx 🎨 - Code

```
git rm -rf apps/quacker-frontend/src/app  
  
cd frontend  
  
git mv -f ./src/index.tsx .. /apps/quacker-frontend/src/main.tsx  
git mv -f ./src/index.css .. /apps/quacker-frontend/src/styles.css  
git mv -f ./public/favicon.ico .. /apps/quacker-frontend/src/favicon.ico  
git mv ./src .. /apps/quacker-frontend/src/app  
git mv -f ./env ./README.md .. /apps/quacker-frontend  
  
cd ..  
  
rm -rf ./frontend
```

⚠ Step 5. - Fixes in code #1

Search & Replace: `REACT_APP_` ➔ `NX_`

```
diff --git a/tsconfig.base.json b/tsconfig.base.json
--- a/tsconfig.base.json
+++ b/tsconfig.base.json
@@ -14,7 +14,9 @@
      "skipLibCheck": true,
      "skipDefaultLibCheck": true,
      "baseUrl": ".",
-    "paths": {}
+    "paths": [
+      "src/*": ["apps/quacker-frontend/src/app/*"]
+    ]
  },
  "exclude": ["node_modules", "tmp"]
}
```

A Step 5. - Fixes in code #2

```
diff --git a/apps/quacker-frontend/src/index.html b/apps/quacker-frontend/src/index.html
--- a/apps/quacker-frontend/src/index.html
+++ b/apps/quacker-frontend/src/index.html
@@ -4,3 +4,3 @@
     <meta charset="utf-8" />
-    <title>QuackerFrontend</title>
+    <title>Quacker</title>
     <base href="/" />
@@ -10,3 +10,3 @@
     </head>
-
+    <body class="w-100 sans-serif black-90 bg-white">
        <div id="root"></div>
```

```
diff --git a/.eslintrc.json b/.eslintrc.json
--- a/.eslintrc.json
+++ b/.eslintrc.json
@@ -10,7 +10,7 @@
     "error",
     {
         "enforceBuildableLibDependency": true,
-
+        "allow": ["src"],
         "depConstraints": [
             {
```

⚠ Step 5. - Fixes in code #3

```
diff --git a/apps/quacker-frontend/src/main.tsx b/apps/quacker-frontend/src/main.tsx
--- a/apps/quacker-frontend/src/main.tsx
+++ b/apps/quacker-frontend/src/main.tsx
@@ -1,5 +1,3 @@
 import ReactDOM from 'react-dom';
-import 'tachyons';

-import './index.css';
 import { App } from 'src/App';
diff --git a/apps/quacker-frontend/src/styles.css b/apps/quacker-frontend/src/styles.css
--- a/apps/quacker-frontend/src/styles.css
+++ b/apps/quacker-frontend/src/styles.css
@@ -1 +1,3 @@
+@import 'tachyons/css/tachyons.css';
+
body {
```

4) Demo 2

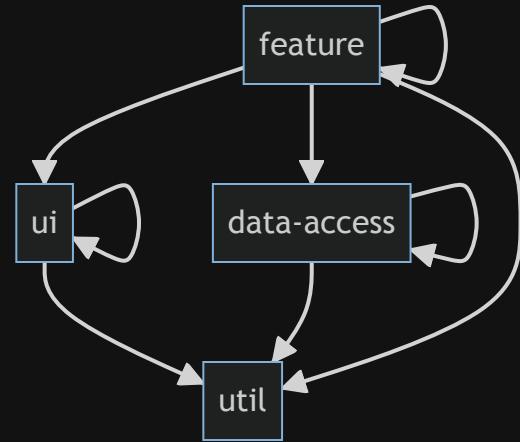


Break a Monolith to Smaller Parts

<https://github.com/heracek/practical-monorepo-nx>

Library Types

- **feature**: UI logic + data-access
- **ui**: pure UI
- **data-access**: API calls, state management
- **util**: low level code (date, formatting, types, ...)



Steps - Break app to smaller parts

1. add `libs/navigation/utils`
2. add `libs/shared/design-system/ui`
3. final demo

⚠ Step 1. - add `libs/navigation/utils`

```
yarn nx g @nrwl/react:library \
--directory=libs/navigation \
--name=util \
--pascalCaseFiles \
--style=none \
--unitTestRunner=jest \
--bundler=none \
--dryRun
```

`route` object:

- `apps/quacker-frontend/src/app/Routes.tsx` ➔ `libs/navigation/util/src/lib/route.ts`

```
-import { route } from 'src/Routes';
+import { route } from '@quacker/navigation/util';
```

Step 2. - add `libs/shared/design-system/ui`

```
yarn nx g @nrwl/react:library \
--directory=shared/design-system \
--name=ui \
--importPath=@quacker/ui \
--pascalCaseFiles \
--style=None \
--unitTestRunner=jest \
--bundler=None

rm -rf libs/shared/design-system/ui/src/lib

git mv apps/quacker-frontend/src/app/ui libs/shared/design-system/ui/src/lib

echo "export * from './lib';" > libs/shared/design-system/ui/src/index.ts
```

Search & Replace: `src/ui` ➔ `@quacker/ui`

⚠ Step 3. - final demo

```
git checkout final
yarn install

yarn nx graph --open=false --port=8888

yarn nx run-many --all --targets=test,lint,build --fix
yarn nx format --all

NX_BASE=final yarn nx affected --targets=test,lint,build --fix
NX_BASE=final yarn nx format

NX_BASE=final yarn --silent nx print-affected --type=app --select=projects
```

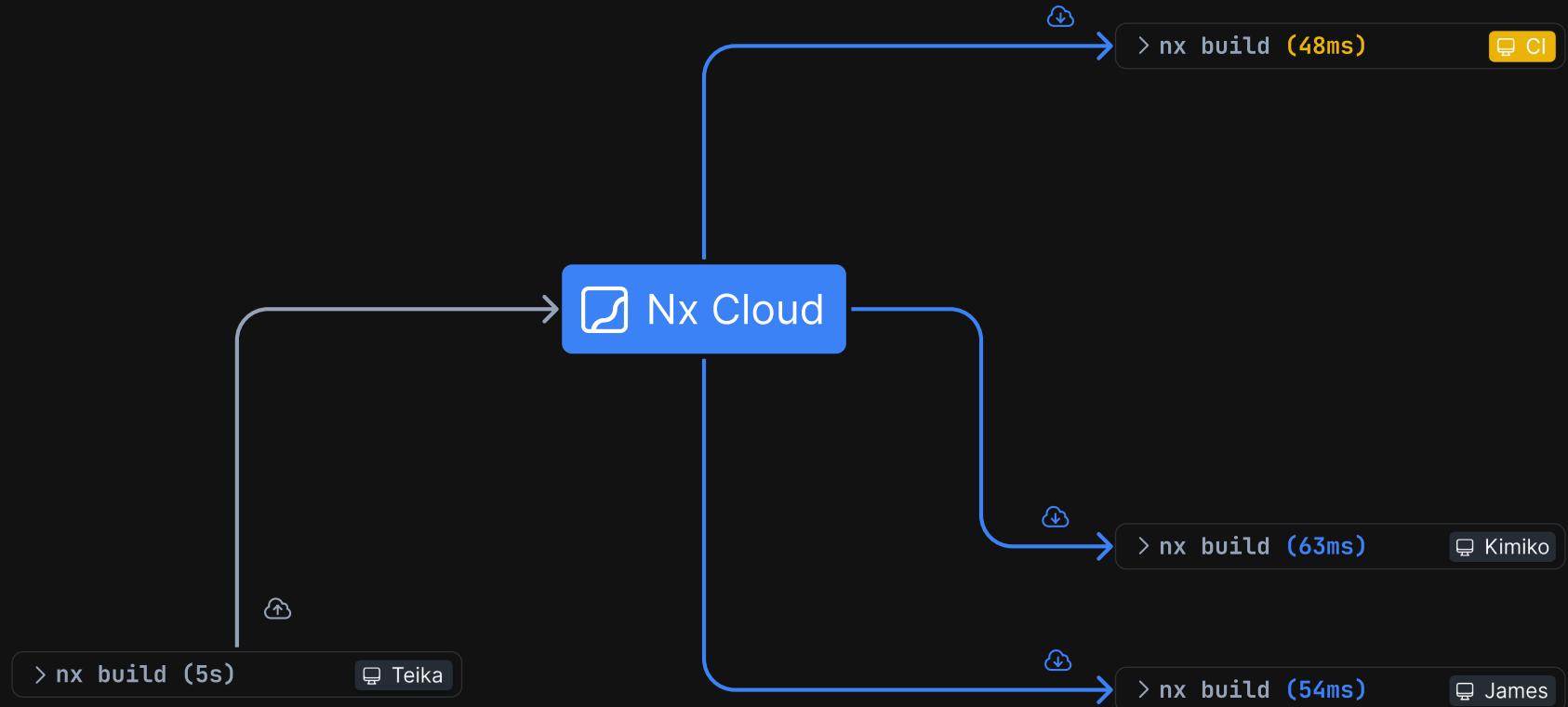
5) CI/CD Setup



Nx Cloud 

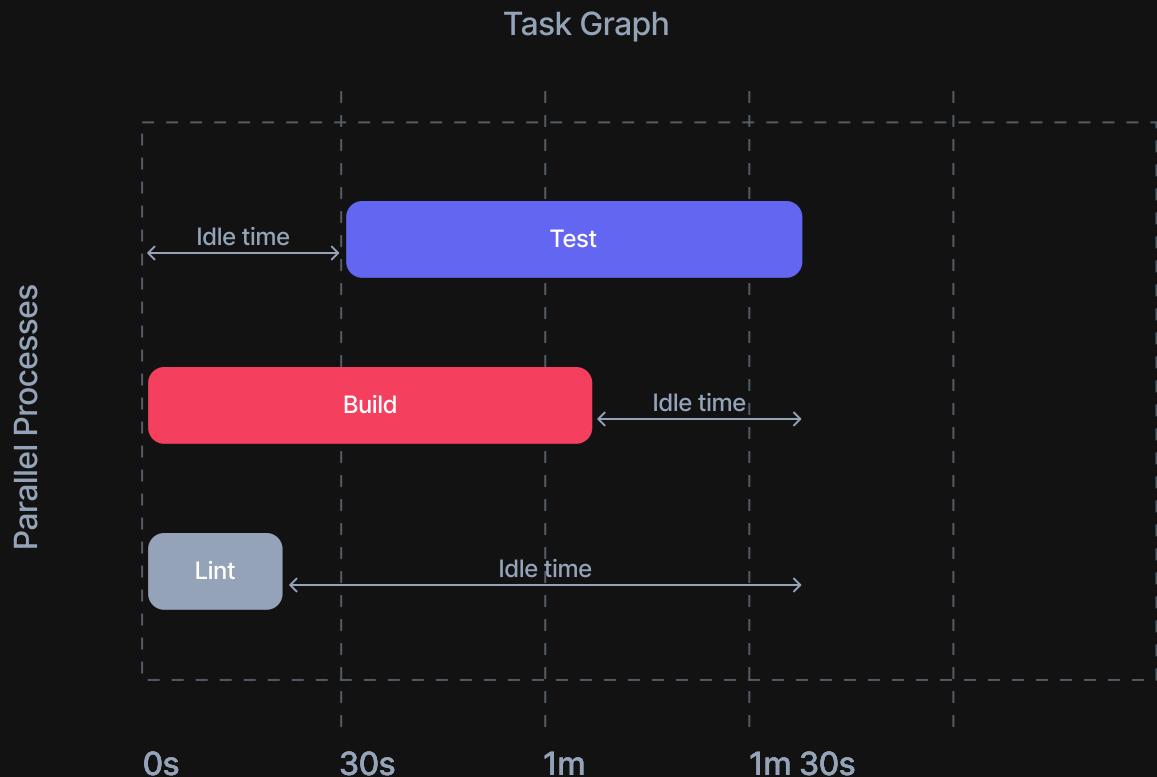
<https://nx.app/>

Nx Cloud ☁ - Distributed 📦 Cache

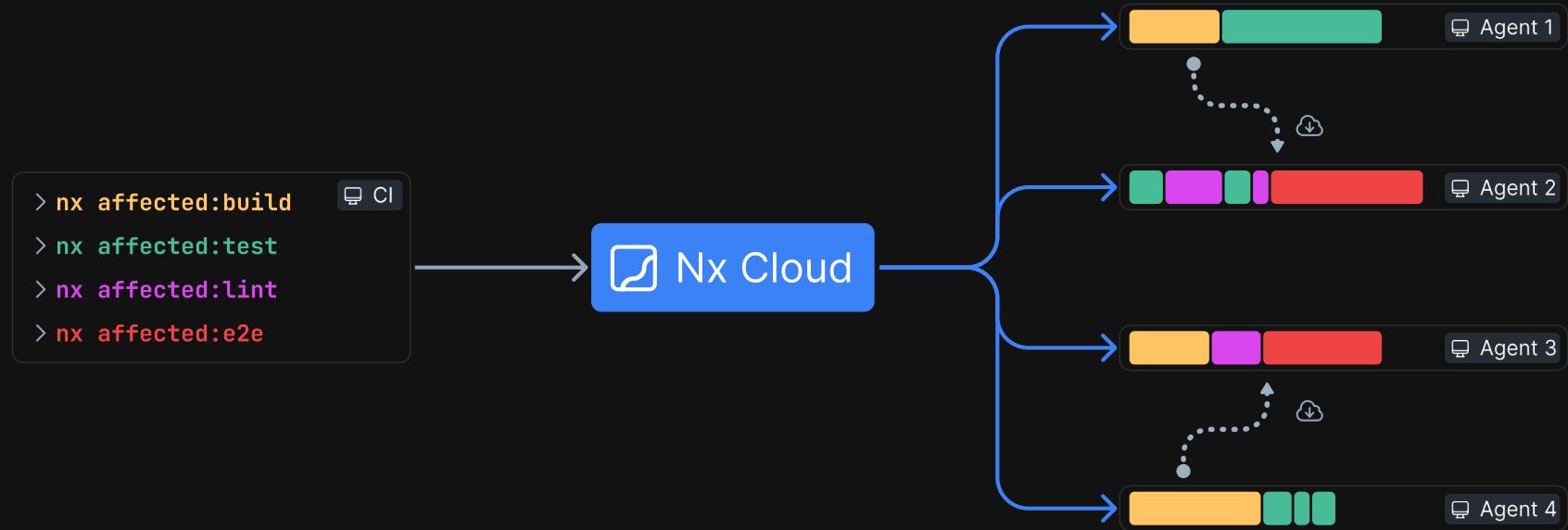


Open Source solution: <https://github.com/NiklasPor/nx-remotecache-custom>

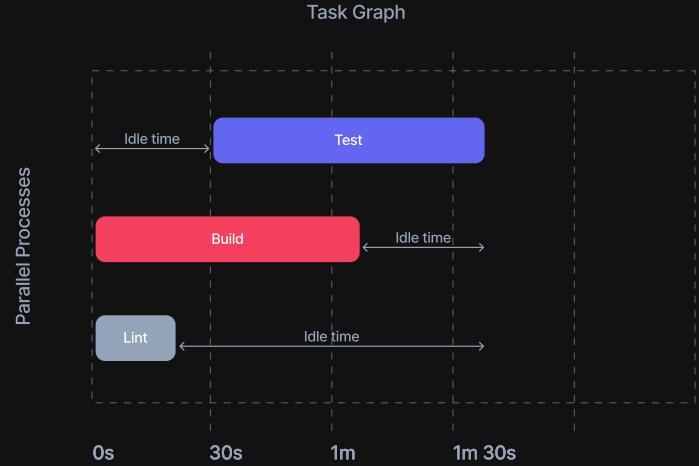
Nx Cloud ☁ - Distributed 🏃 Task Execution



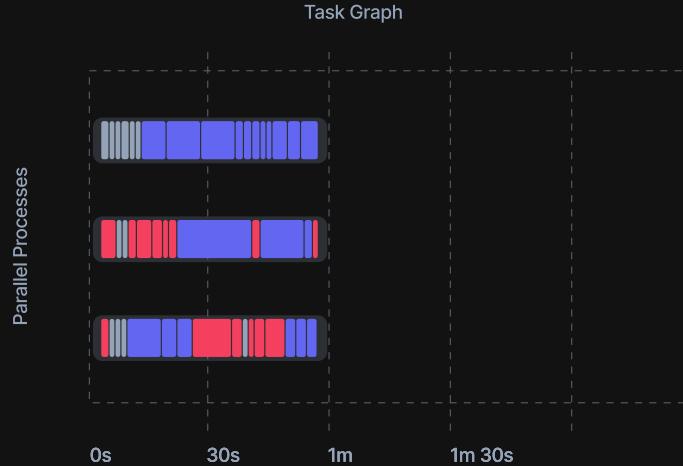
Nx Cloud ☁ - Distributed 🏃 Task Execution



Parallel



Distributed

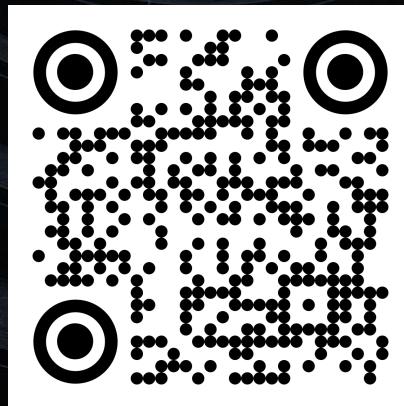




CI/CD Setup

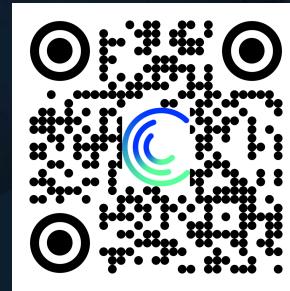
<https://nx.dev/recipes/ci>

Any questions?



Share your feedback!

Thank you!



Tomáš Horáček

toh@ciklum.com | ciklum.com