React at Scale with Nx Monorepos

Why Monorepos?

01

Atomic Changes

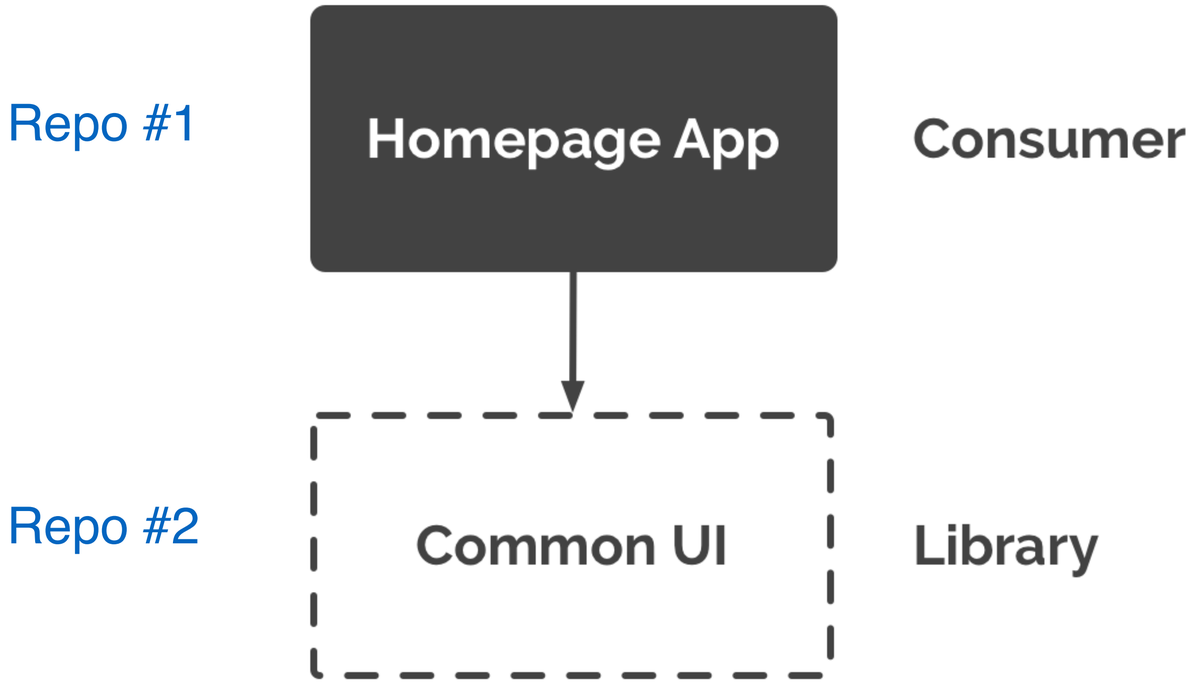
02

Shared Code

03

Single Set of Dependencies

01 Atomic Changes



Why Monorepos?

01

Atomic Changes

02

Shared Code

03

Single Set of Dependencies

02 Shared Code

```
function usernameIsValid(username: string): boolean {
  return username.length > 4 && username !== 'jeffbcross';
}
```

Why Monorepos?

01

Atomic Changes

02

Shared Code

03

Single Set of Dependencies

03 Single Set of Dependencies

Why Not Code Collocation?

01

Running Unnecessary Tests

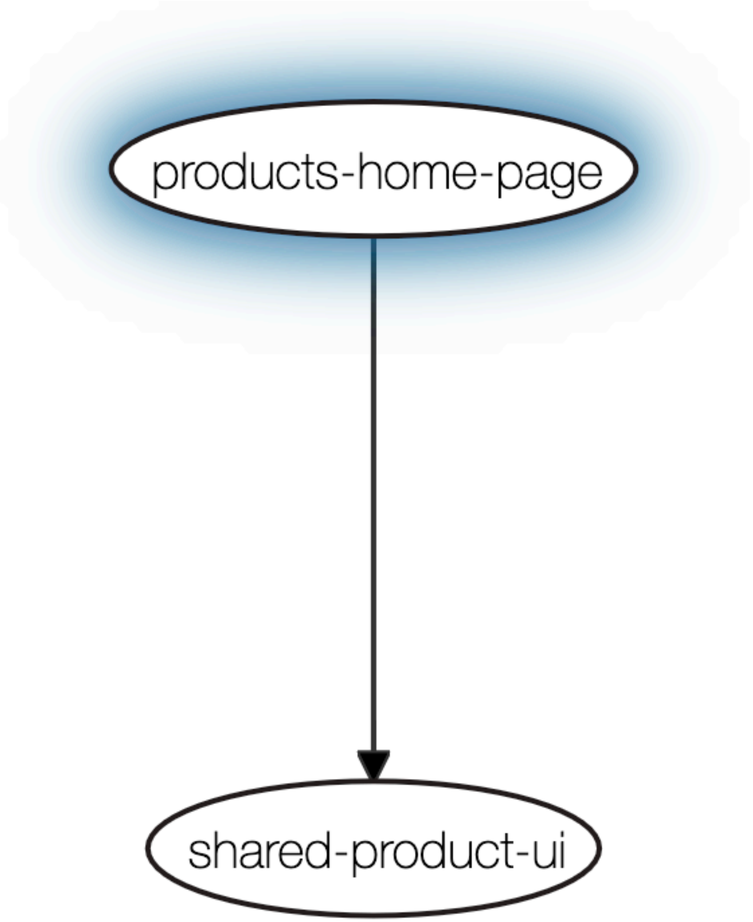
02

No Code Boundaries

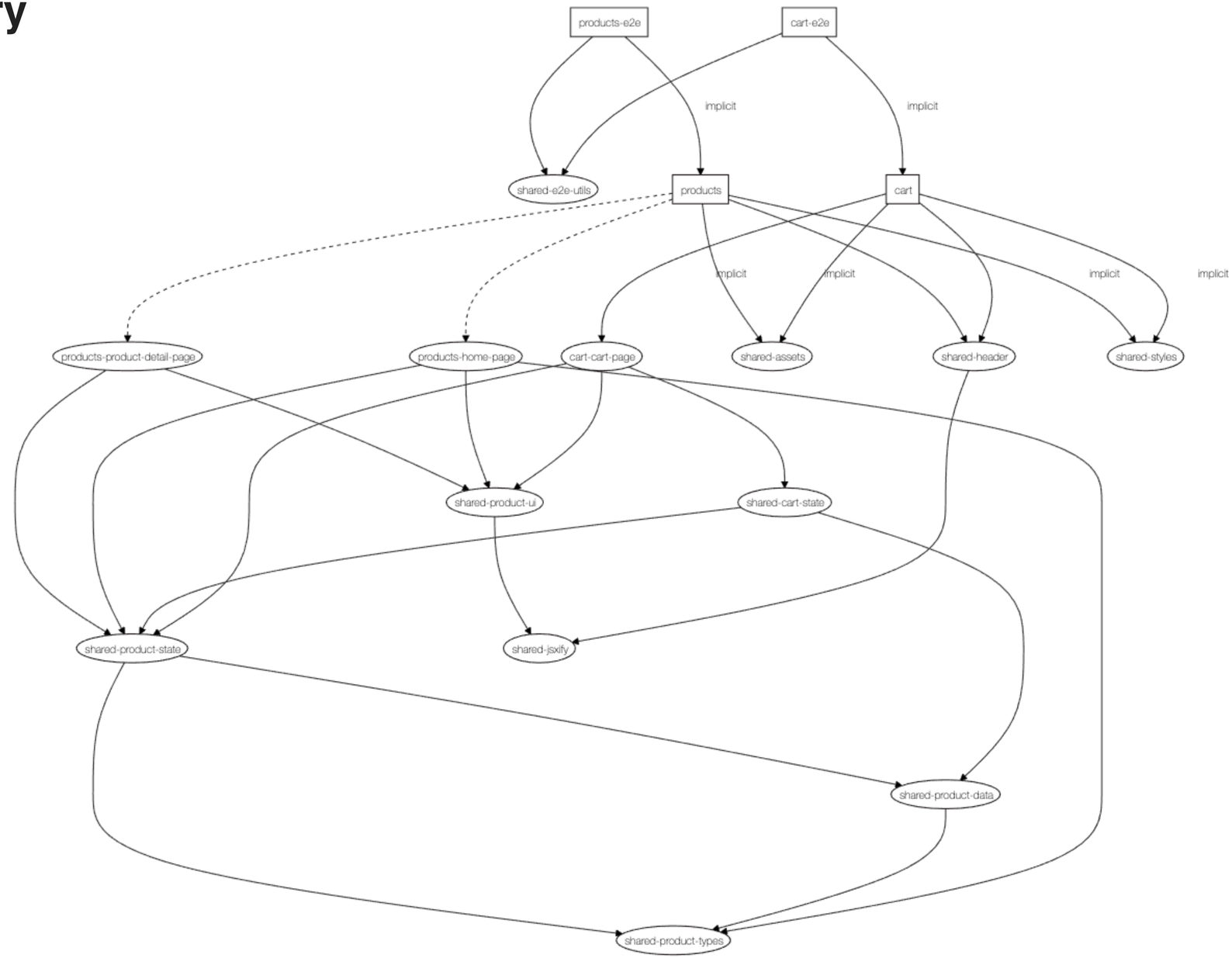
03

Inconsistent Tooling

01 Running Unnecessary Tests



01 Running Unnecessary Tests



Why Not Code Collocation?

01

Running Unnecessary Tests

02

No Code Boundaries

03

Inconsistent Tooling

02 No Code Boundaries

```
import { unstableDontUse } from '@myorg/shared-ui/i/hope/no/one/finds/this';
```

Why Not Code Collocation?

01

Running Unnecessary Tests

02

No Code Boundaries

03

Inconsistent Tooling

03 Inconsistent Tooling

Nx Can Help

01 Faster Command Execution

02 Controlled Code Sharing

03 Consistent Coding Practices

04 Accurate Architecture Diagram

01 Faster Command Execution

- Executors
- nx affected
- Local and distributed caching

Nx Can Help

01 Faster Command Execution

02 Controlled Code Sharing

03 Consistent Coding Practices

04 Accurate Architecture Diagram

02 Controlled Code Sharing

- Library API
- Tags
- Publishable Libraries
- CODEOWNERS

Nx Can Help

01 Faster Command Execution

02 Controlled Code Sharing

03 Consistent Coding Practices

04 Accurate Architecture Diagram

03 Consistent Coding Practices

- Linting
- Generators
- Nrwl Plugins
- Community Plugins

Nx Can Help

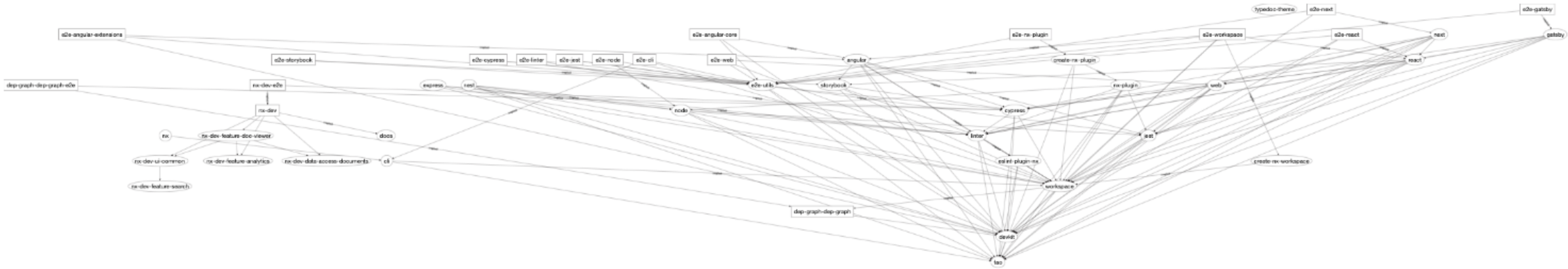
01 Faster Command Execution

02 Controlled Code Sharing

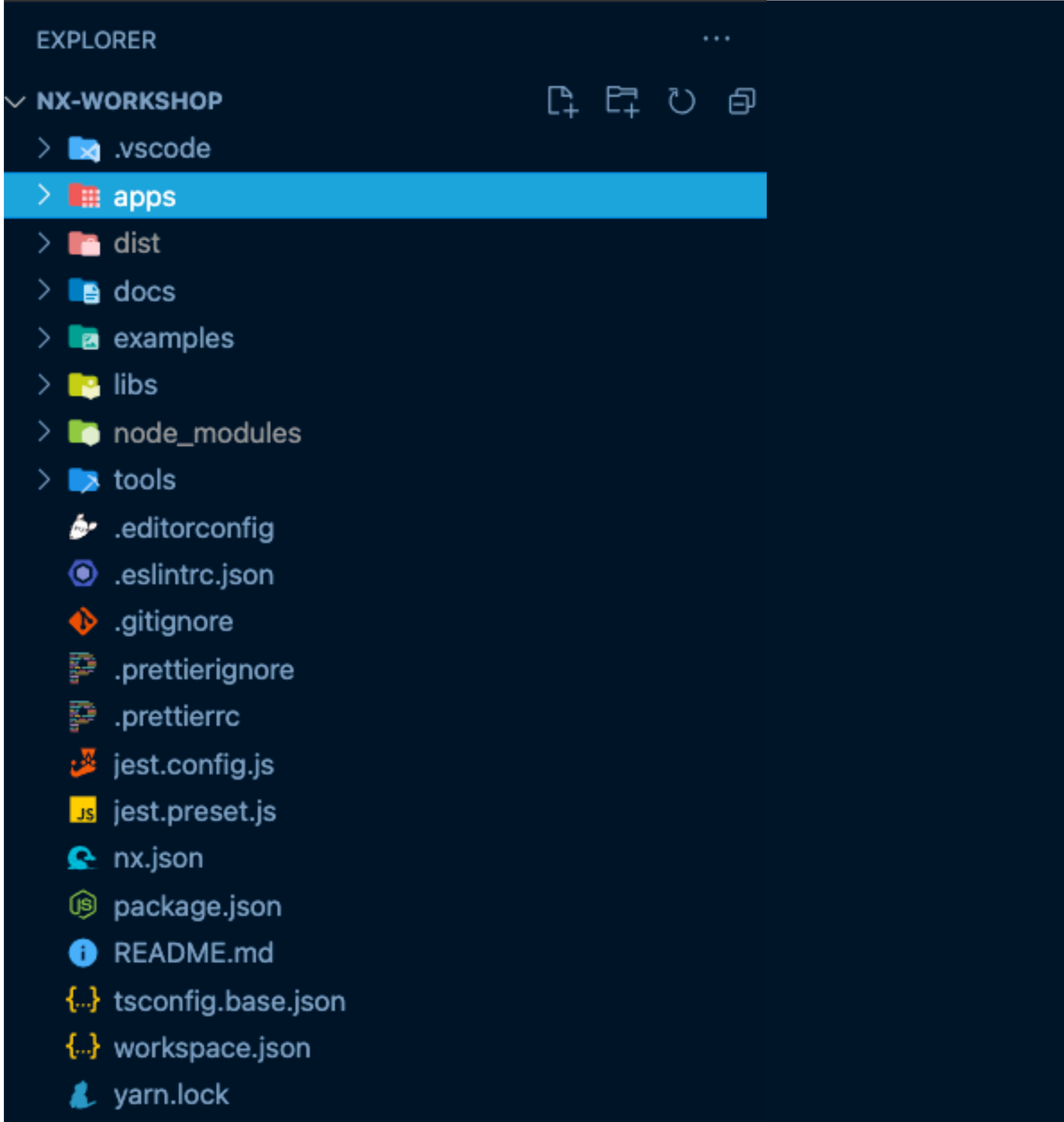
03 Consistent Coding Practices

04 Accurate Architecture Diagram

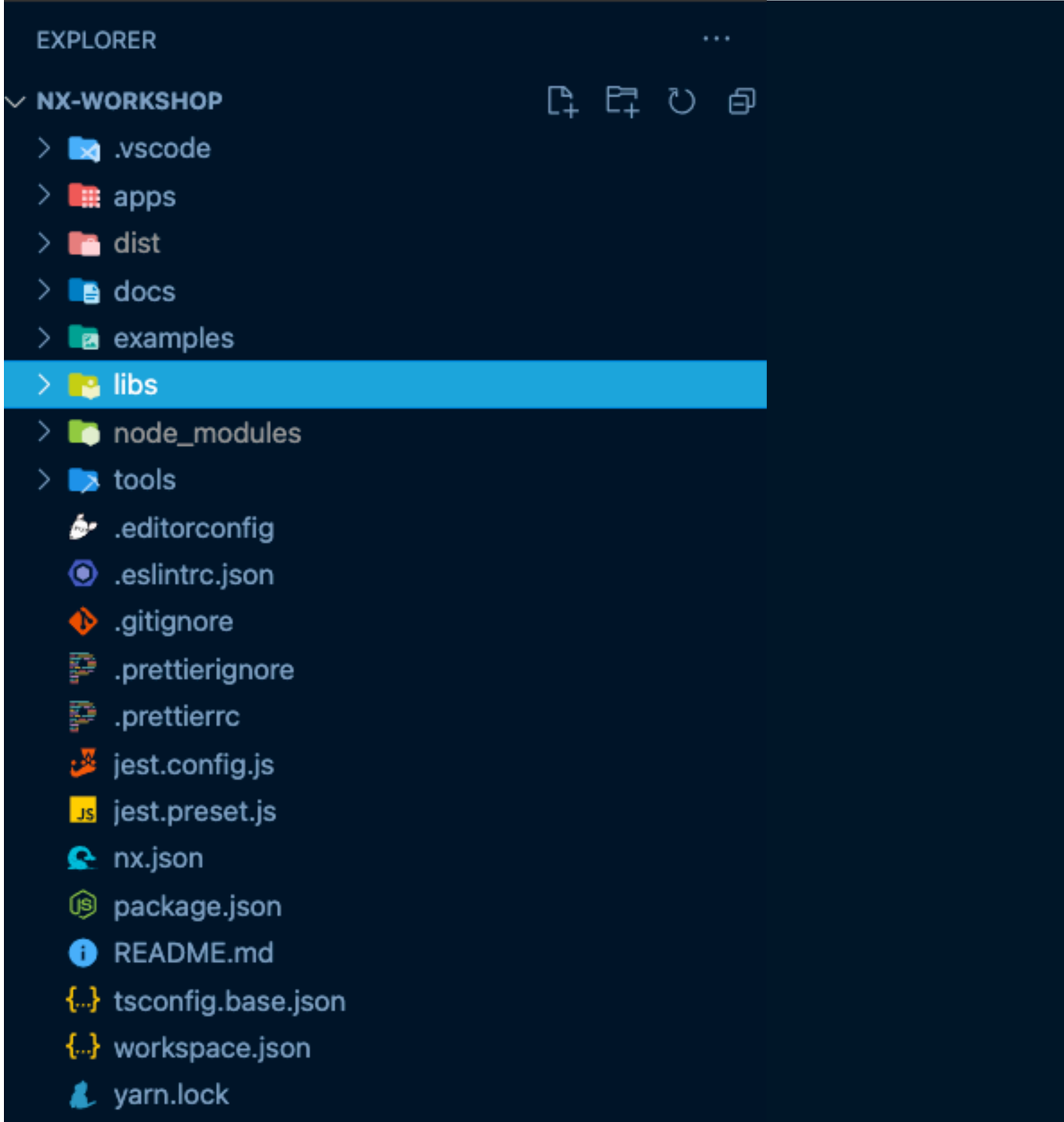
04 Accurate Architecture Diagram



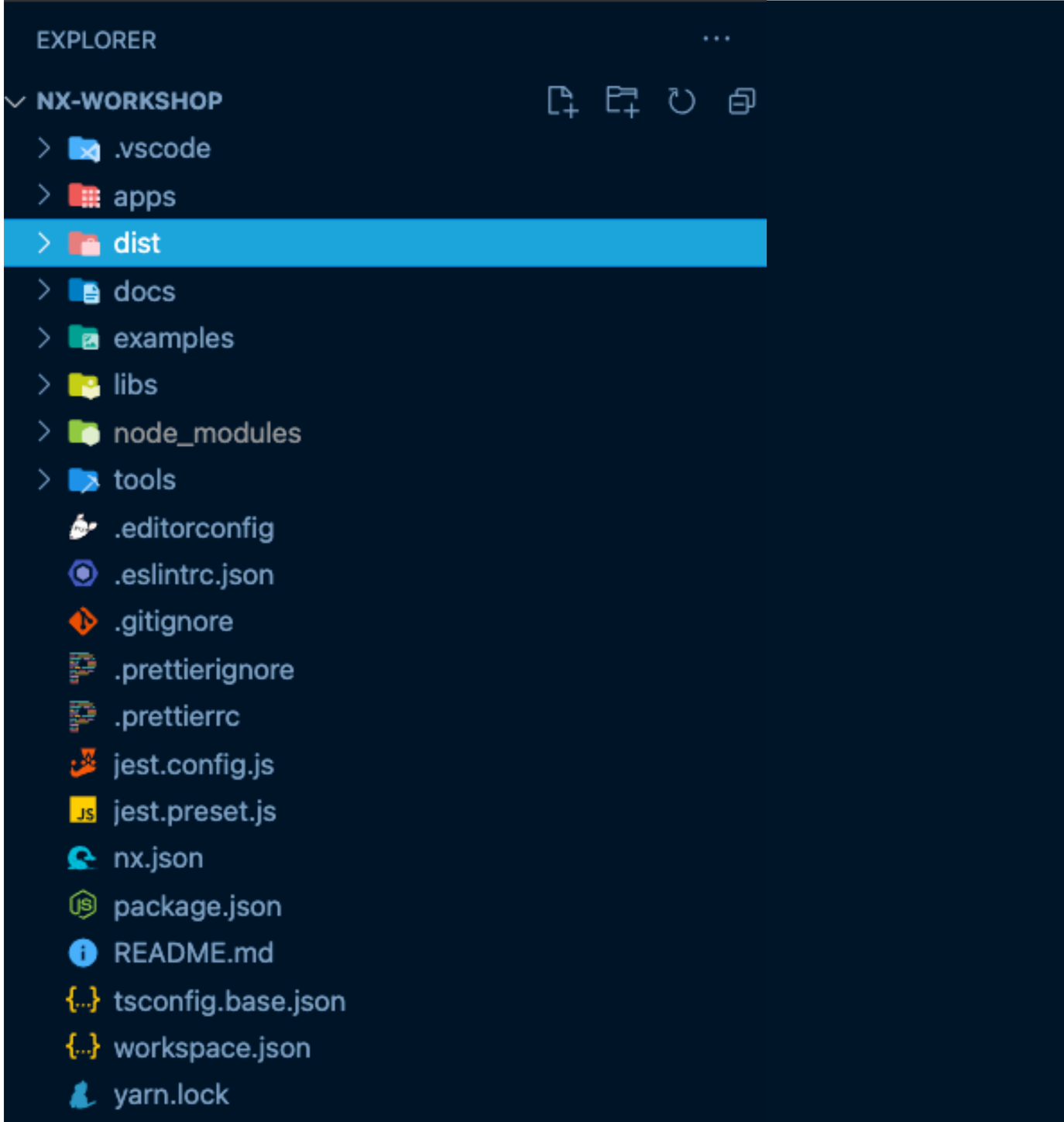
Workspace File Structure



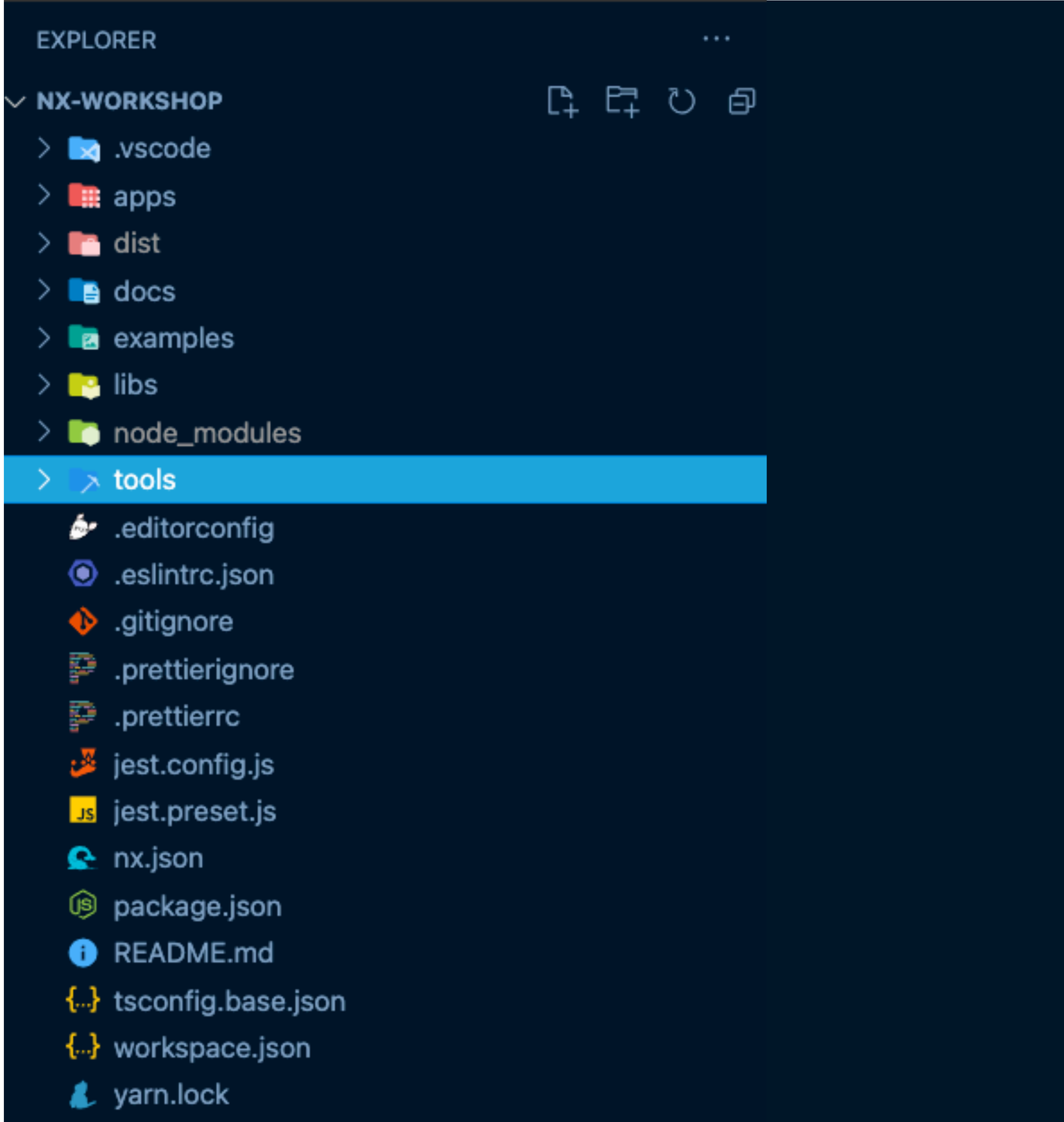
Workspace File Structure



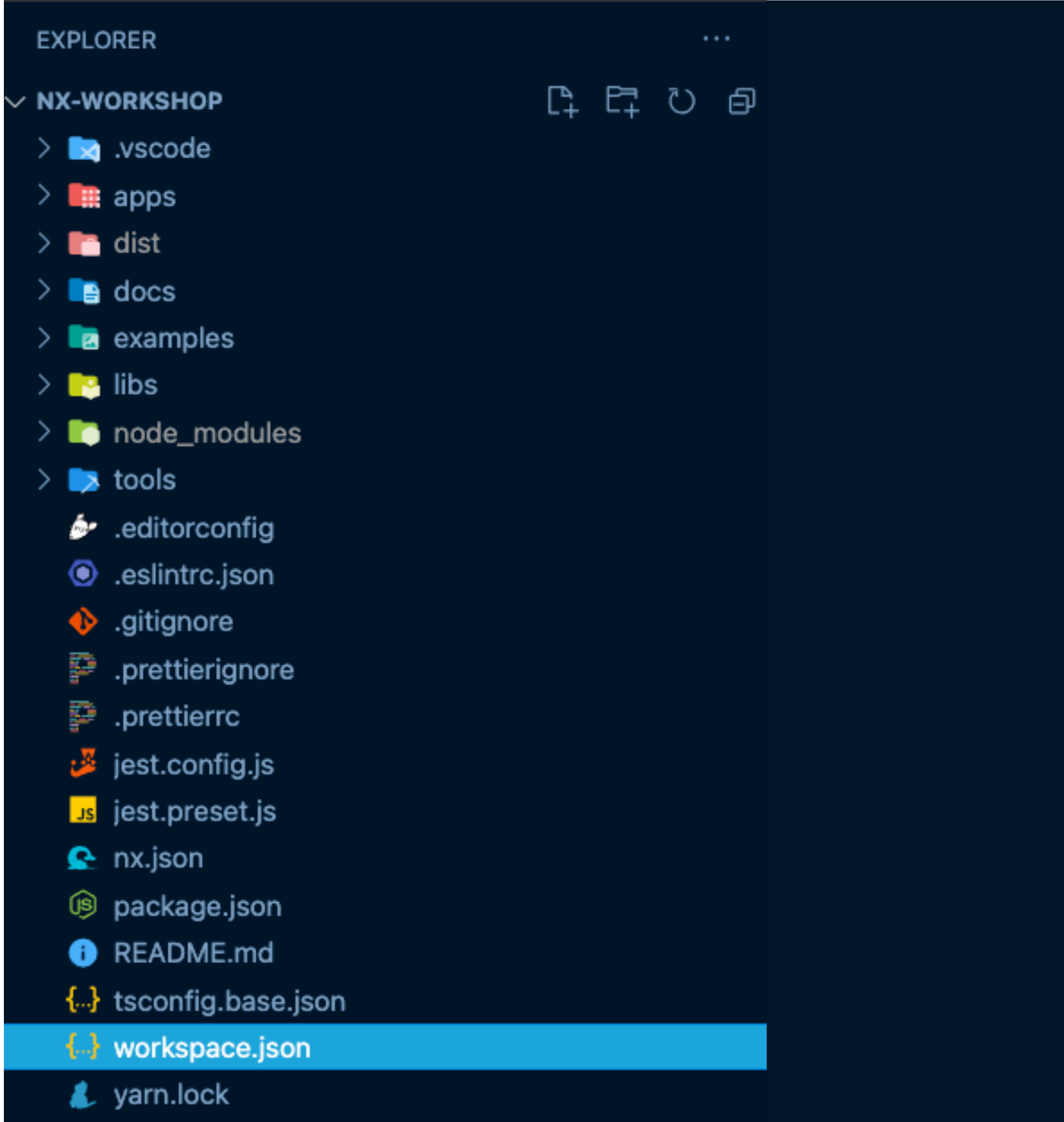
Workspace File Structure



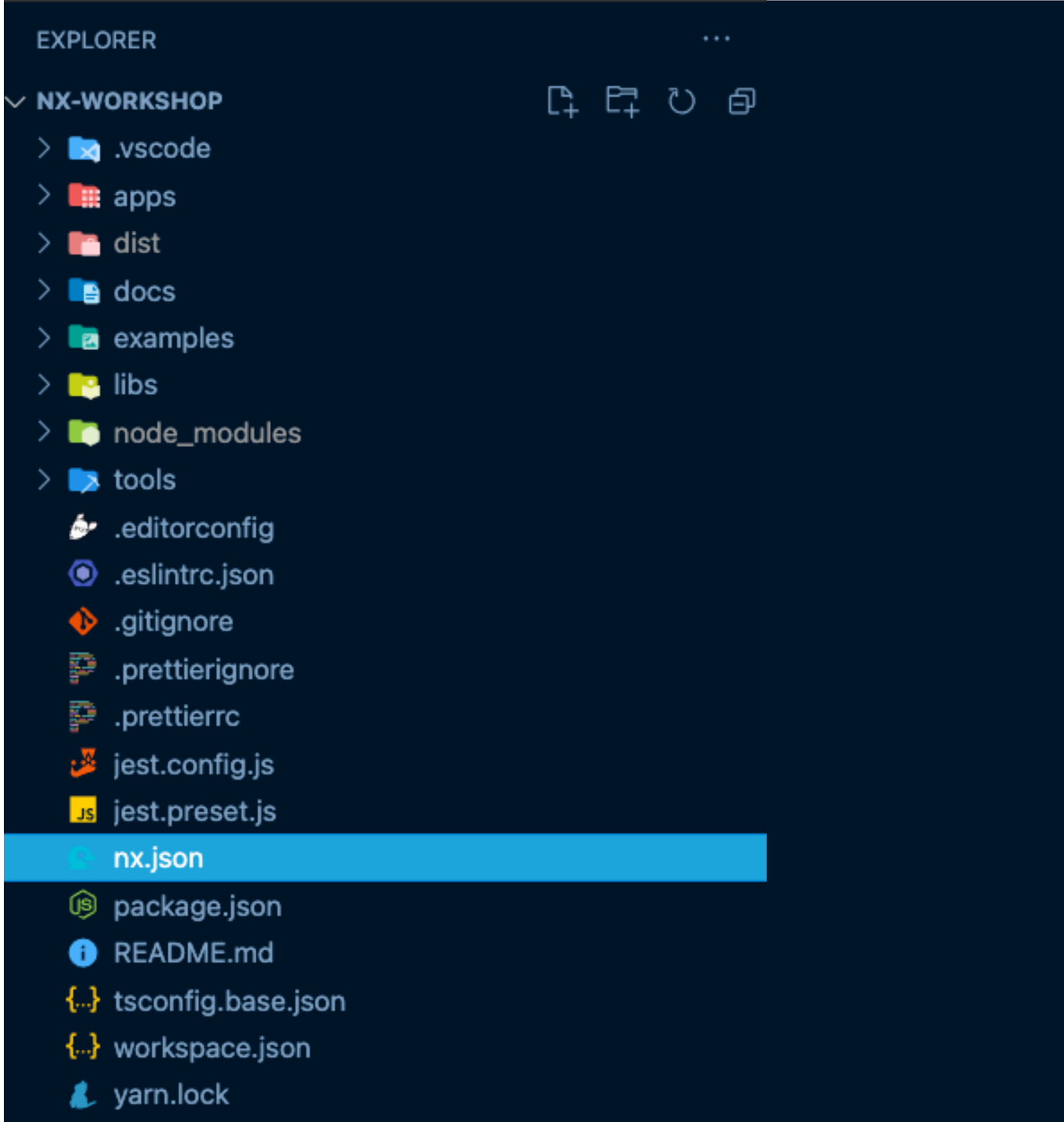
Workspace File Structure



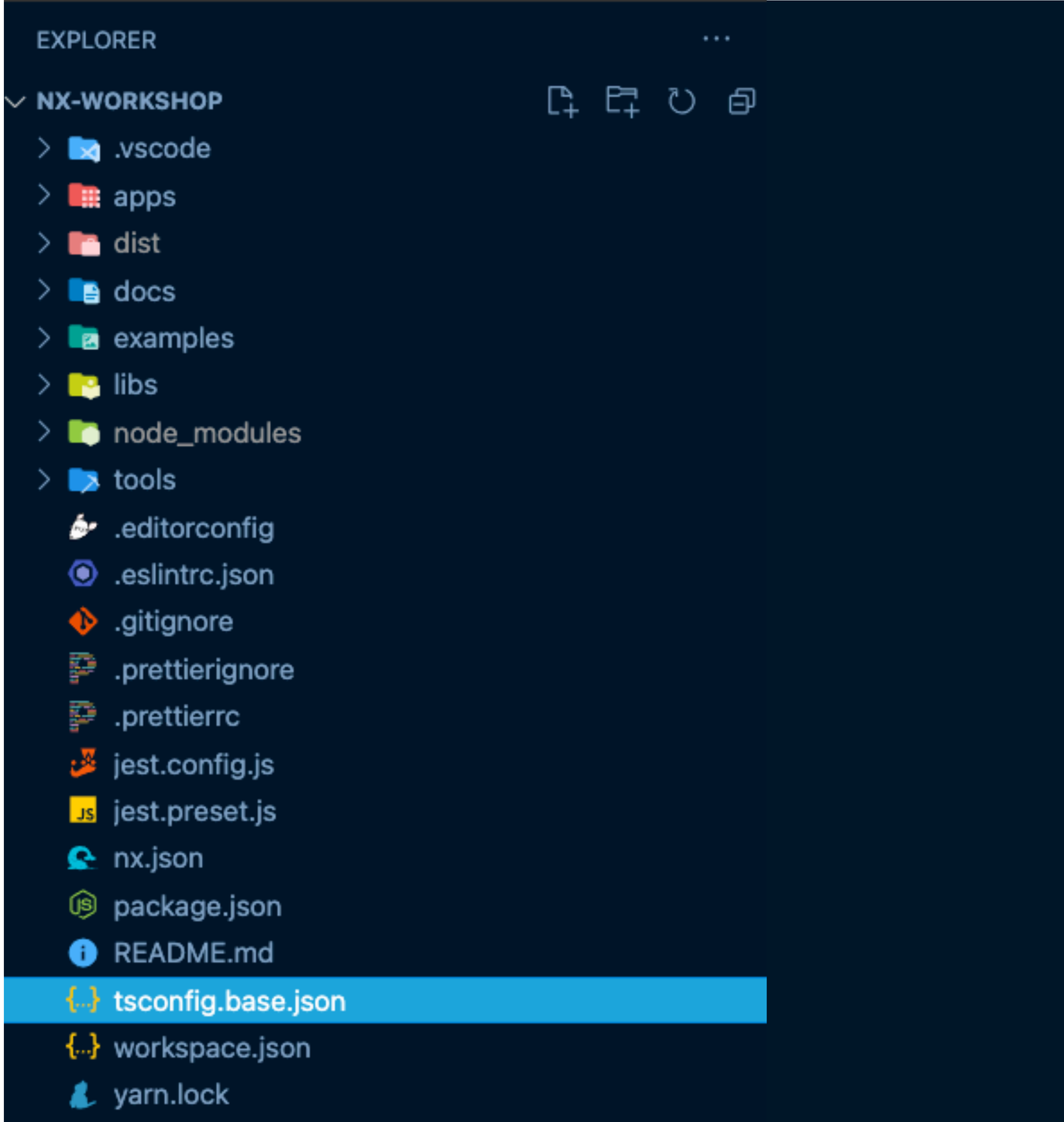
Workspace File Structure



Workspace File Structure

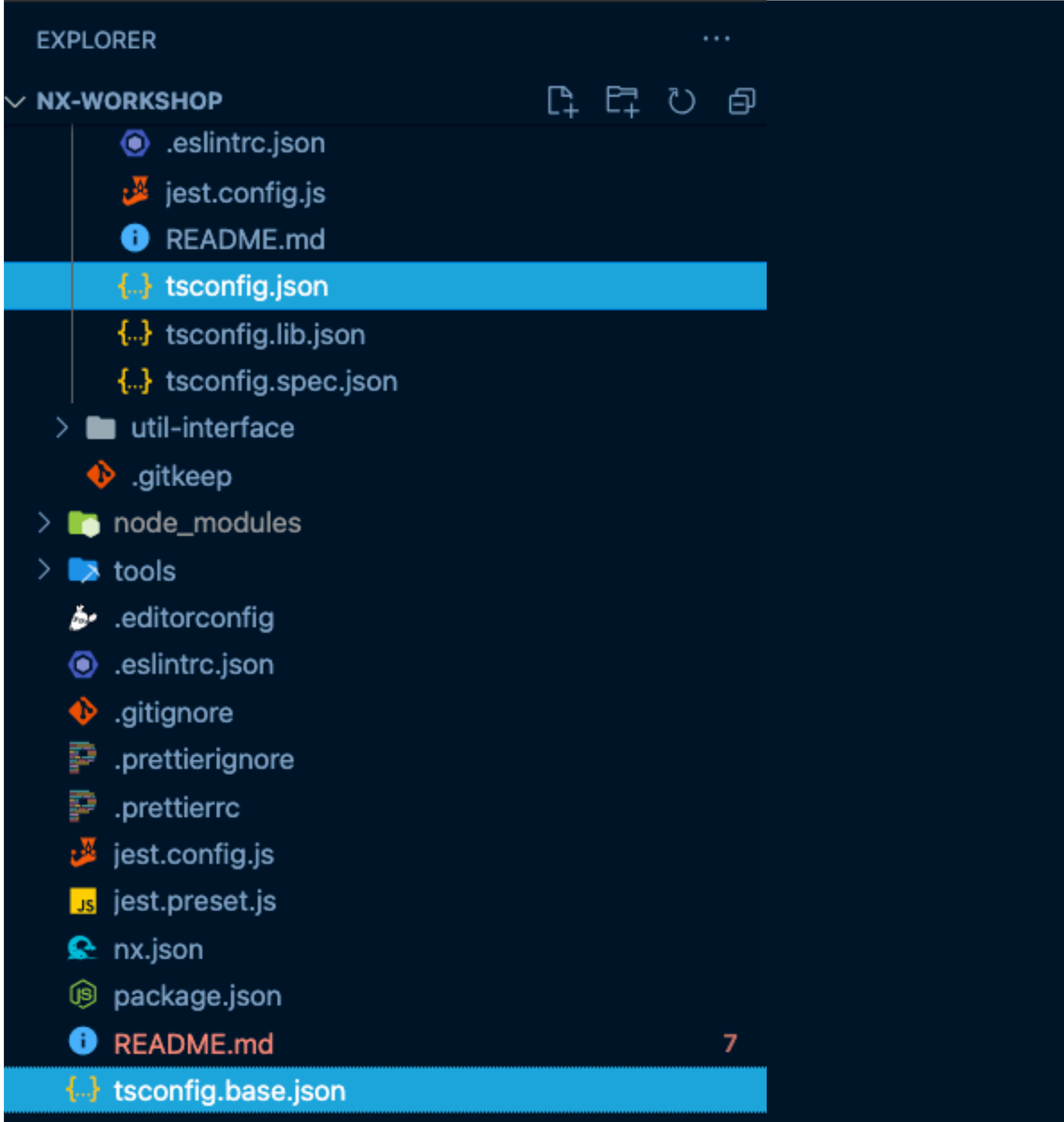


Workspace File Structure



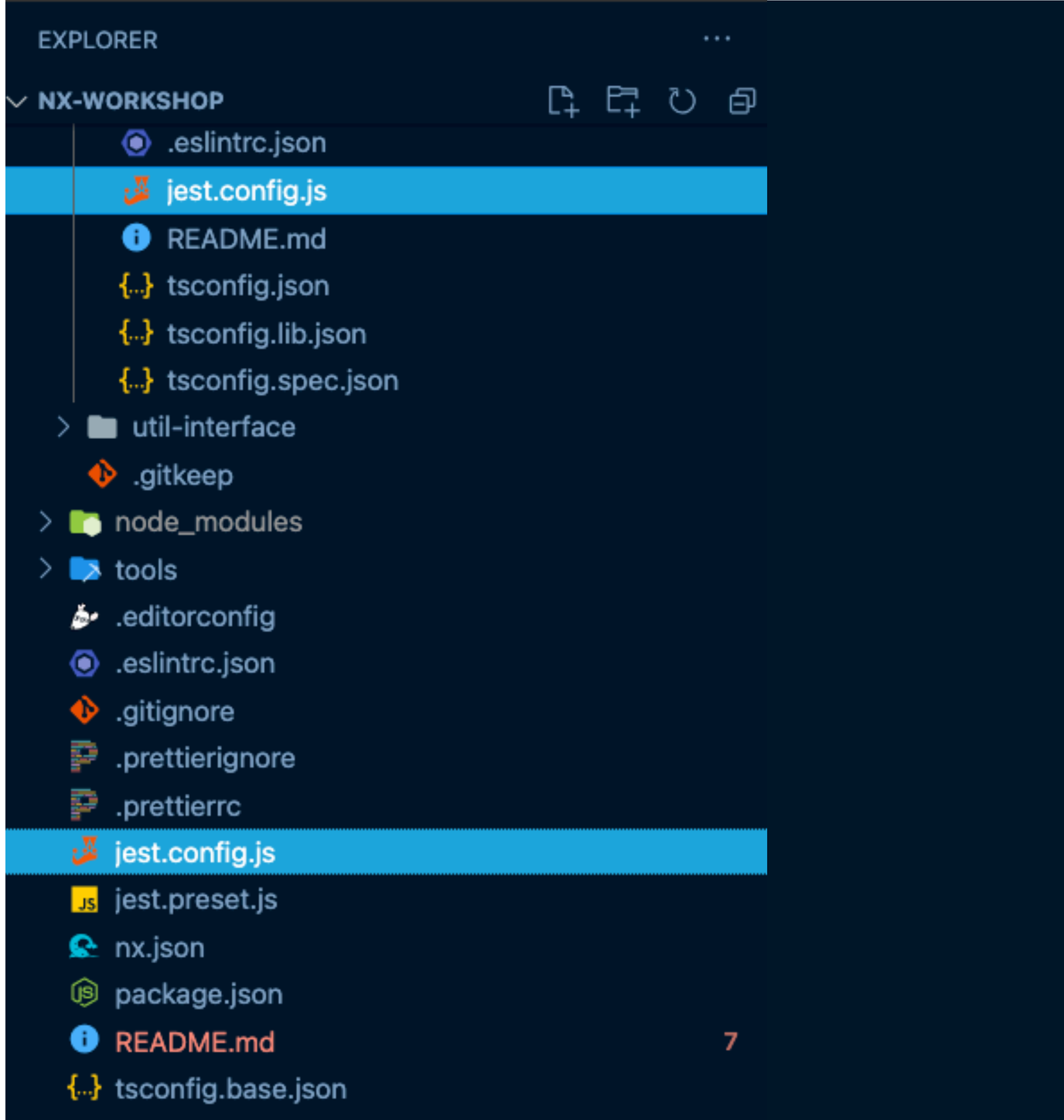
Workspace File Structure

Workspace Config
and
Project Config



Workspace File Structure

Workspace Config
and
Project Config



- 🔬 Lab 1 - Generate an empty workspace
- 🤿 Lab 2 - Generate an React app
- 🧪 Lab 3 - Executors
- 🔭 Lab 4 - Generate a component lib
- 🧬 Lab 5 - Generate a utility lib
- 🧱 Lab 6 - Generate a route lib
- 🤖 Lab 7 - Add an Express API
- 📐 Lab 8 - Displaying a full game in the routed game-detail component
- 💻 Lab 9 - Generate a type lib that the API and frontend can share
- 🤷‍♀️ Lab 10 - Generate Storybook stories for the shared ui component
- ⌨️ Lab 11 - E2E test the shared component

Create a New Workspace

- ```
  npx create-nx-workspace [workspace name]
  ```

- Workspace name sets three things
  - Directory (/Users/bob/Documents/my-org)
  - Path alias (import {} from ‘@my-org/some-projects’;)
  - npm scope (npm install @my-org/published-library)
  ```

Lab 1

Generate an empty workspace

Plugins

```
nx list
```

```
yarn add [plugin]
```

```
Ex: yarn add @nrwl/nest
```

Schematics

- CLI
  - Syntax:

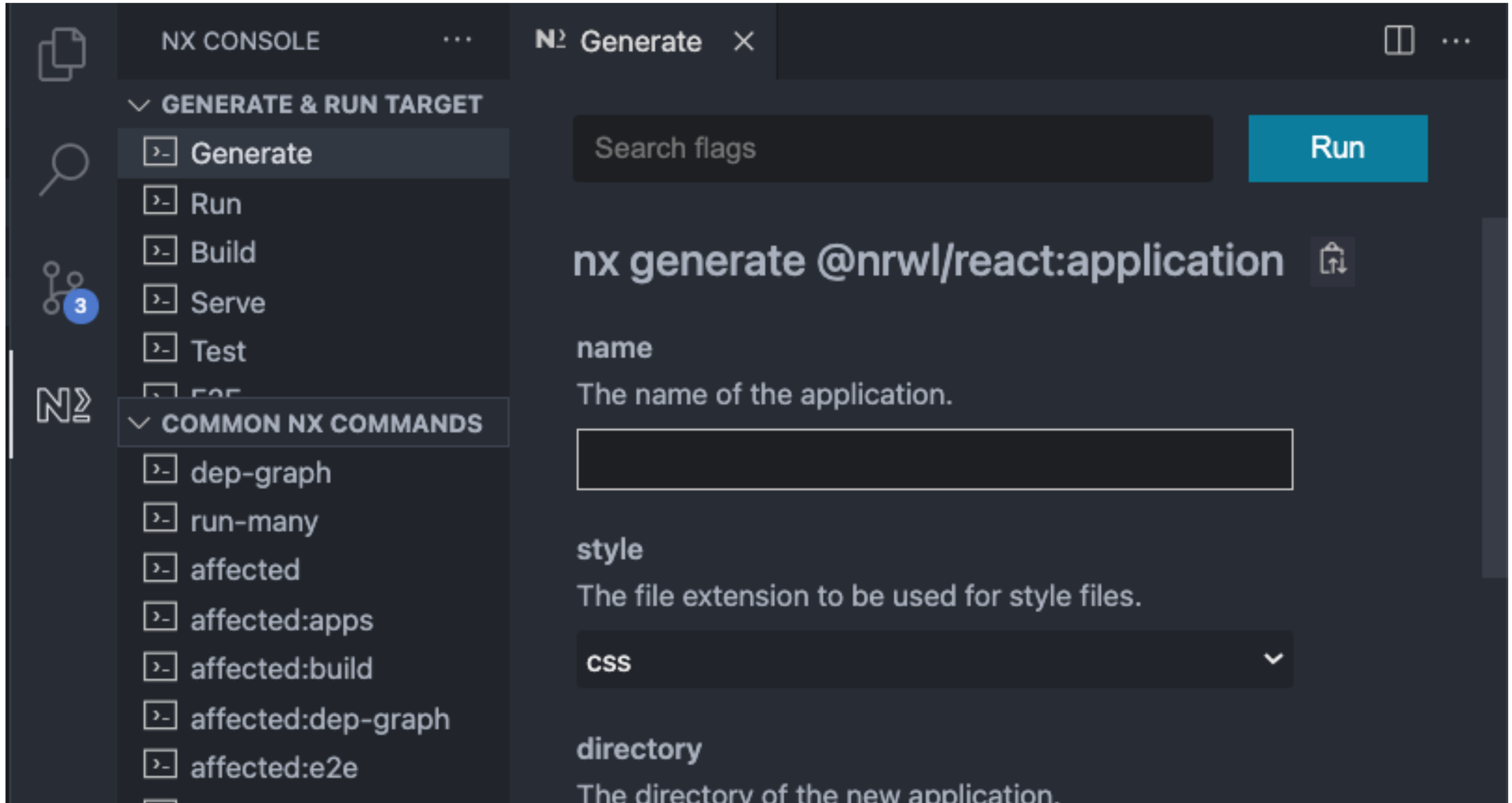
```
nx generate [plugin]:[generator] [options]
```

  - Example:

```
nx generate @nrwl/react:app my-app
```

Schematics

- Nx Console



Code Generation for...

Angular, React, Ngrx, Storybook,...

NestJS, NextJS, Node

TSLint, ESLint, Prettier

Protractor, Jest, Cypress

```
nx generate @nrwl/workspace:workspace-generator my-generator
```

Nx Workspace Generators

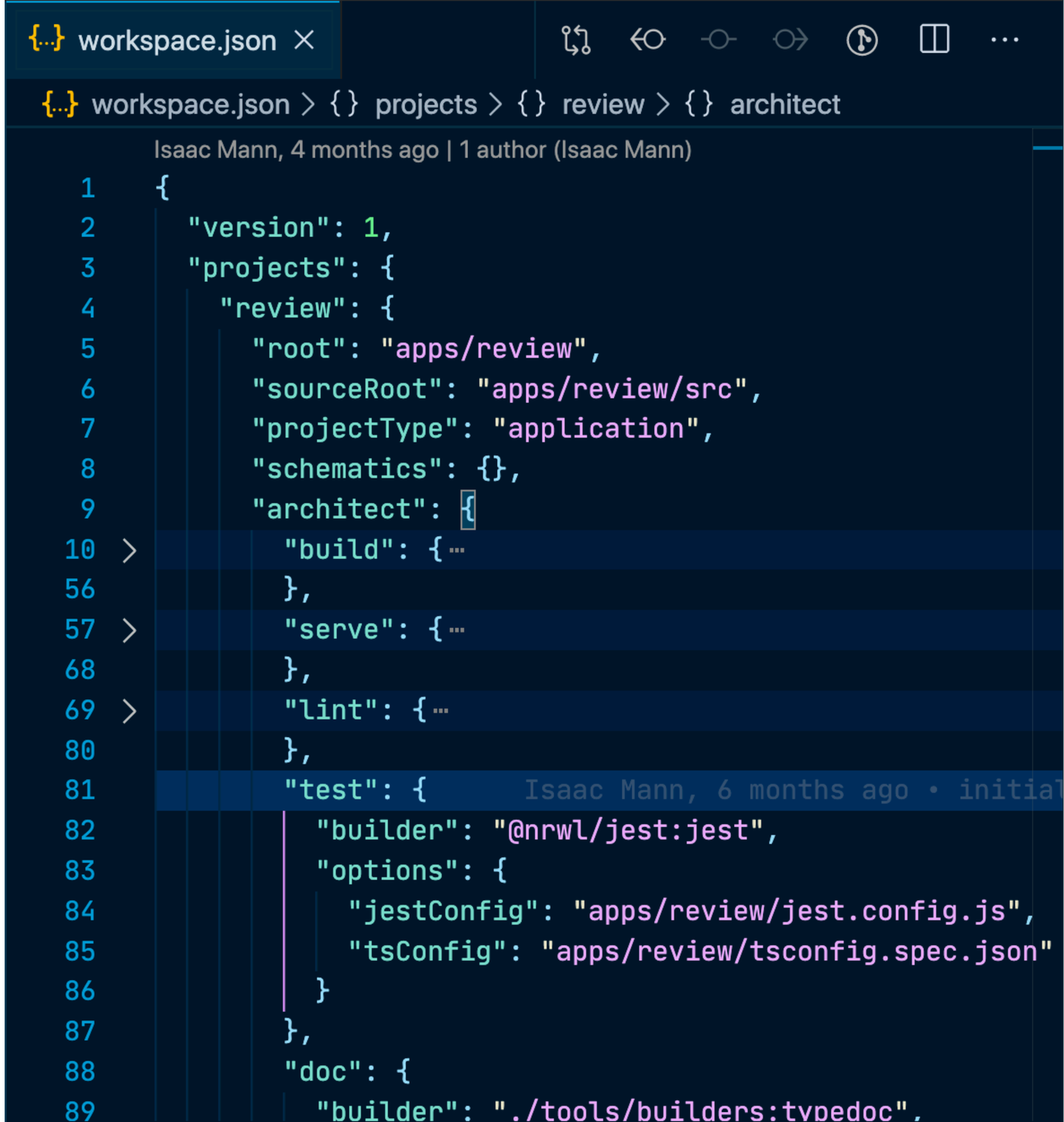
- Create custom generators
- Optimized for your business
- Increase Dev Velocity
- Guarantee compliance

Lab 2

Generate a React app

Executors

- Defined in workspace.json



Executors

- CLI

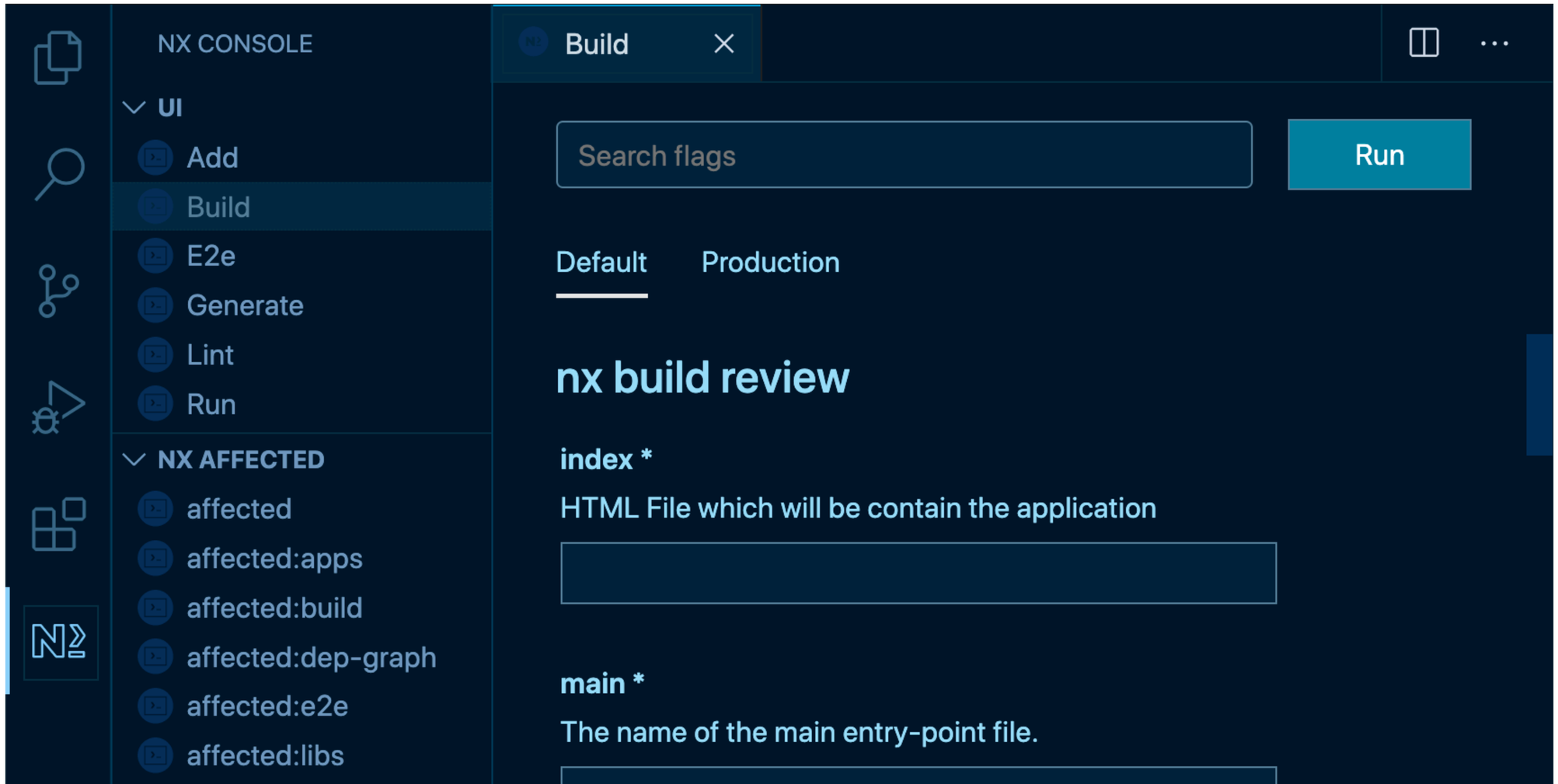
```
nx run [project]:[target] [options]
Ex: nx run my-app:serve
```

- CLI Shorthand

```
nx [target] [project] [options]
Ex: nx serve my-app
```

Executors

- Nx Console



Lab 3

Executors

Libraries

- Feature
  Ex: feat-home

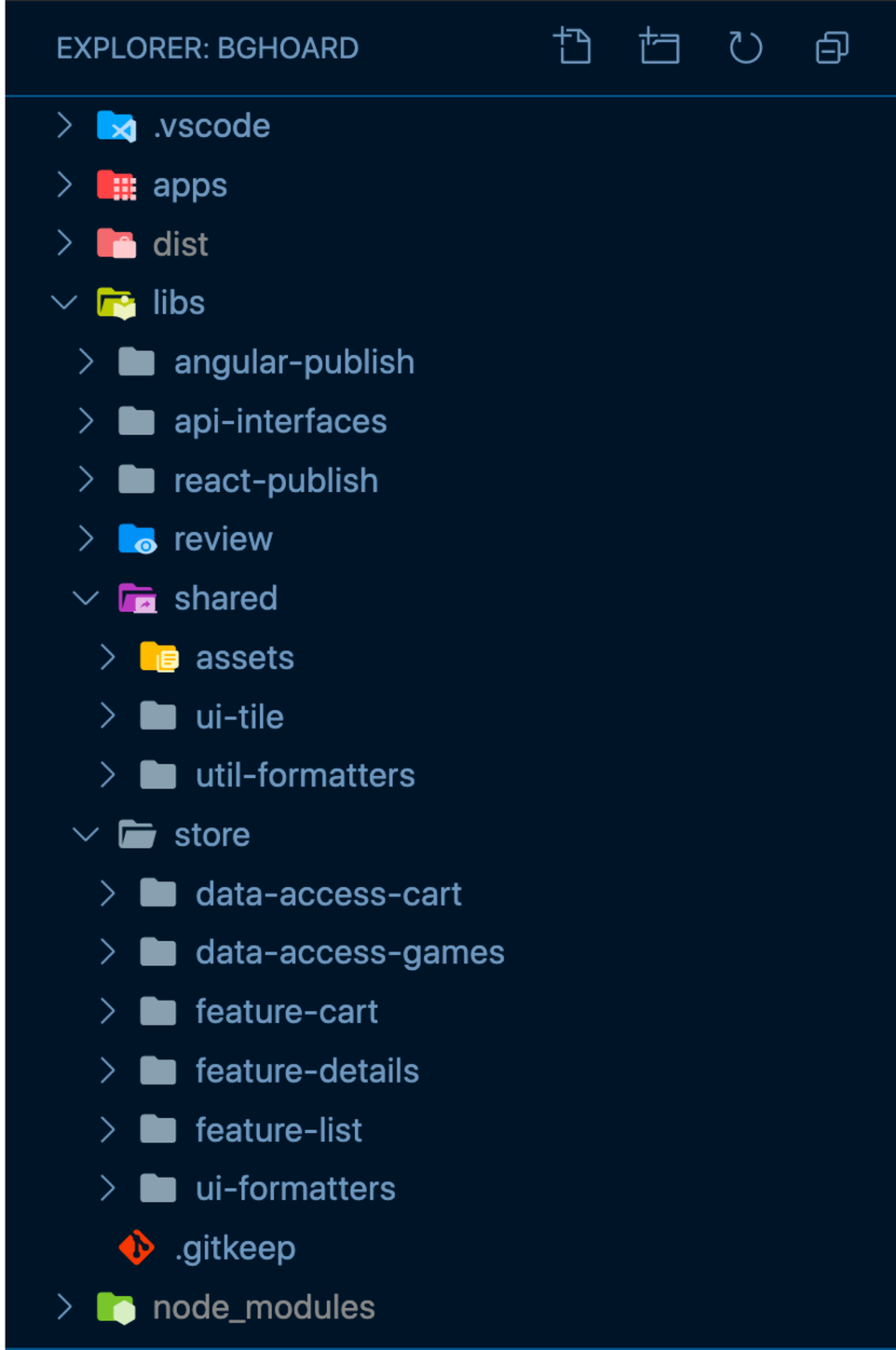
- UI
  Ex: ui-input-forms

- Data
  Ex: data-access-authentication

- Util
  Ex: util-validation-fns

Libraries

- Directory structure
- Note: Moving libraries is easy (see Lab 9)



Libraries

- When should I split code into a new library?
- When do I have too many libraries?

Libraries

- Dependency Graph

Focused on cart

Unfocus

Display Options

Select Affected | Select All

group by folder

app projects

apps

products

cart

e2e projects

e2e

products-e2e

cart-e2e

lib projects

cart

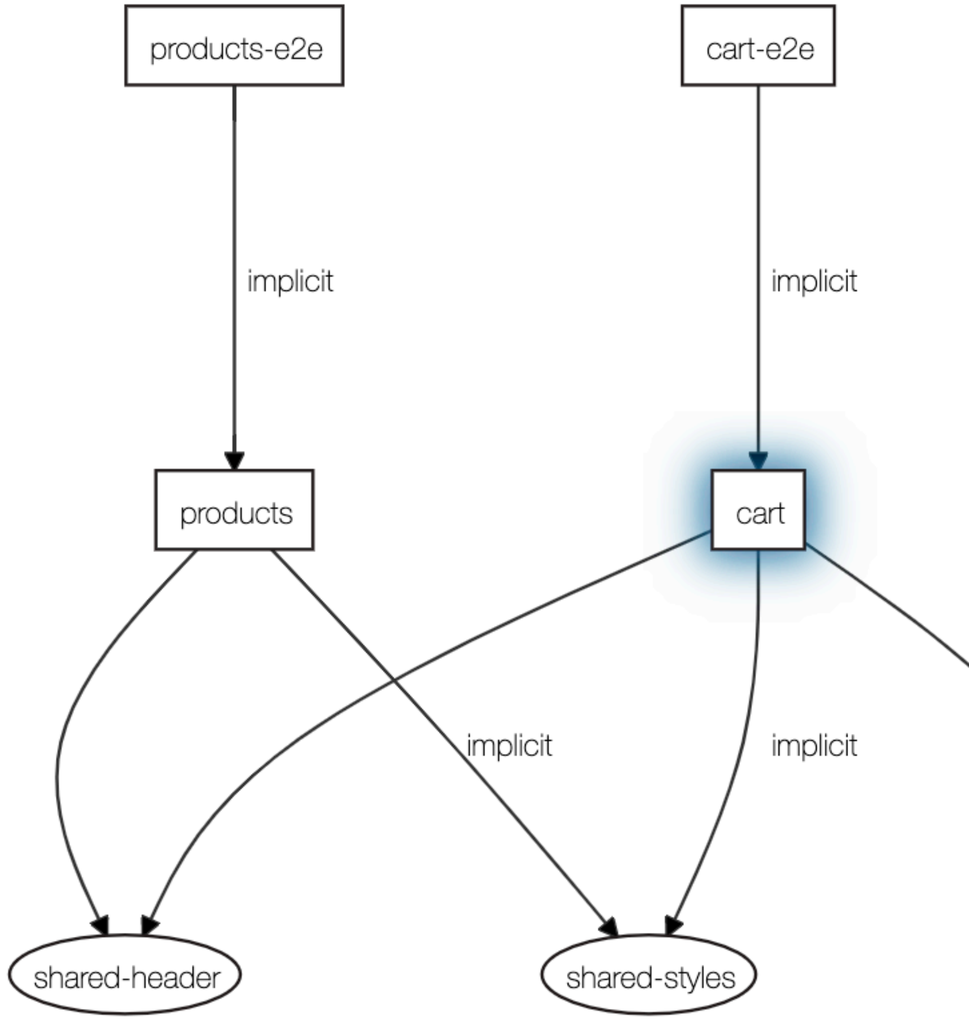
cart-cart-page

products

products-product-detail-page

products-home-page

shared



Lab 4

Generate a Component Library

Lab 5

Generate a Utility Library

Lab 6

Generate a Route Library

Lab 7

Add an Express API

Lab 8

Displaying a Full Game in the Routed game-detail Component

Lab 9

Generate a Type Library that the API and Frontend Can Share

Poll / Q&A

React at Scale with Nx Monorepos

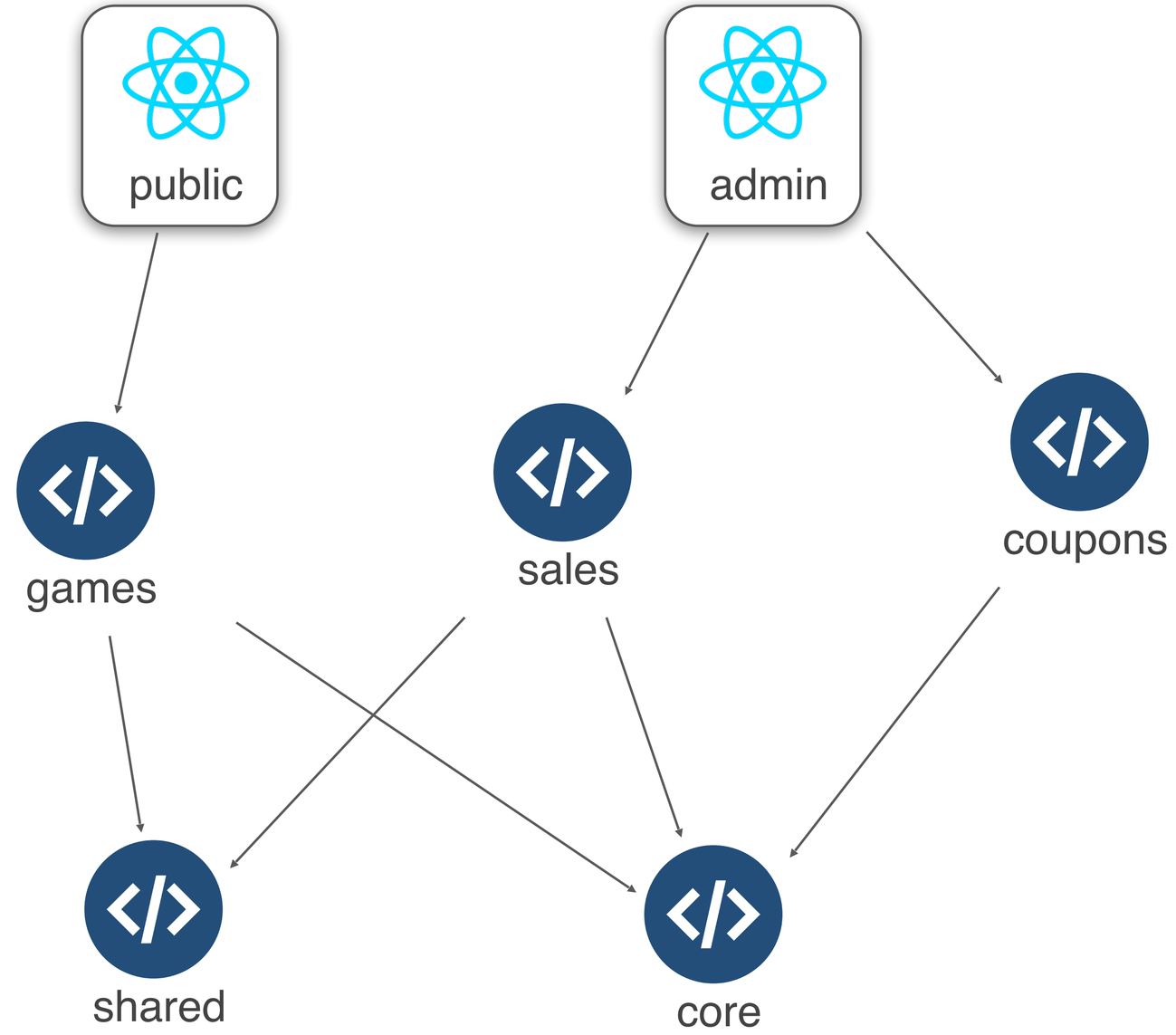
Day 2

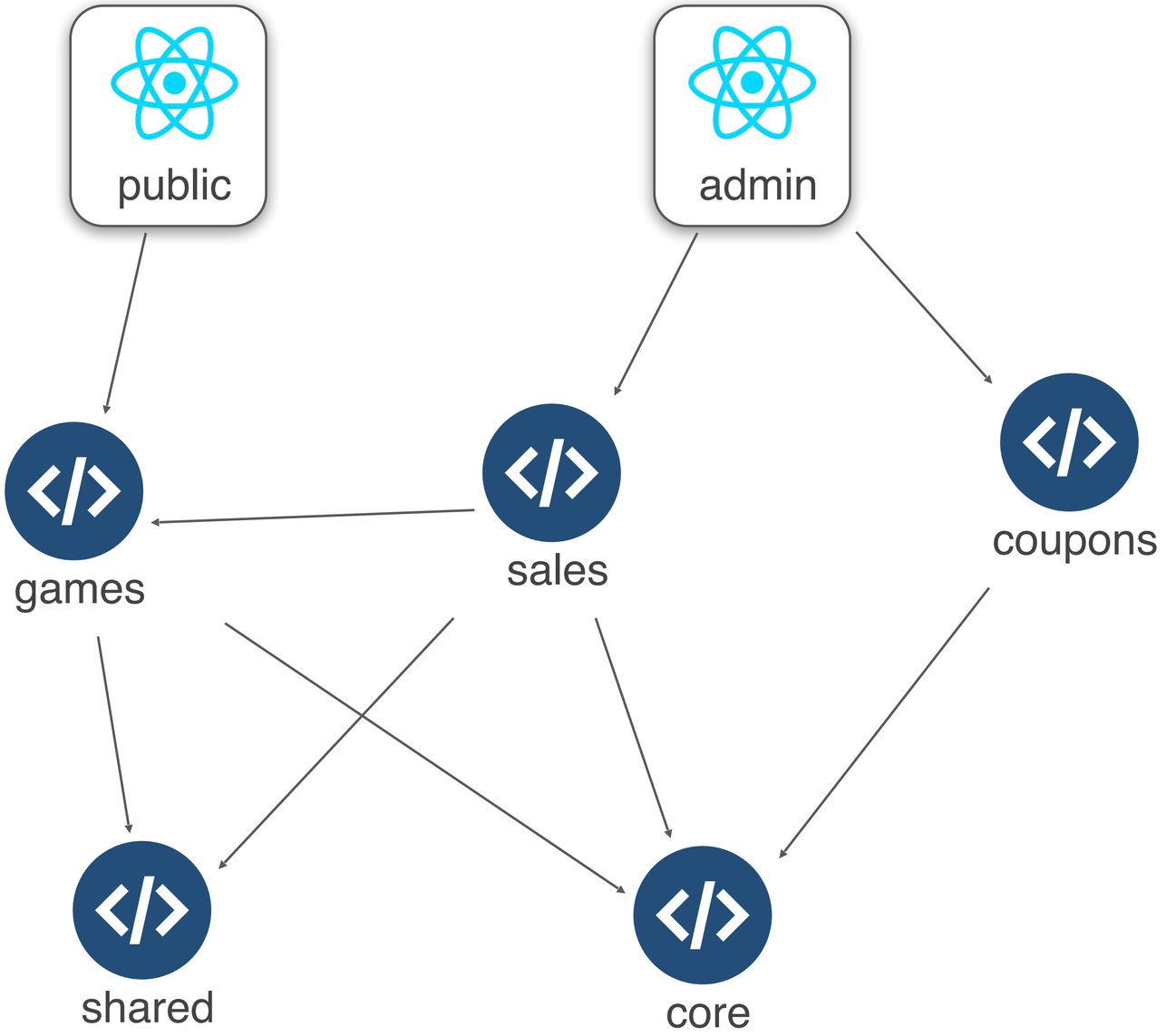
Day 1

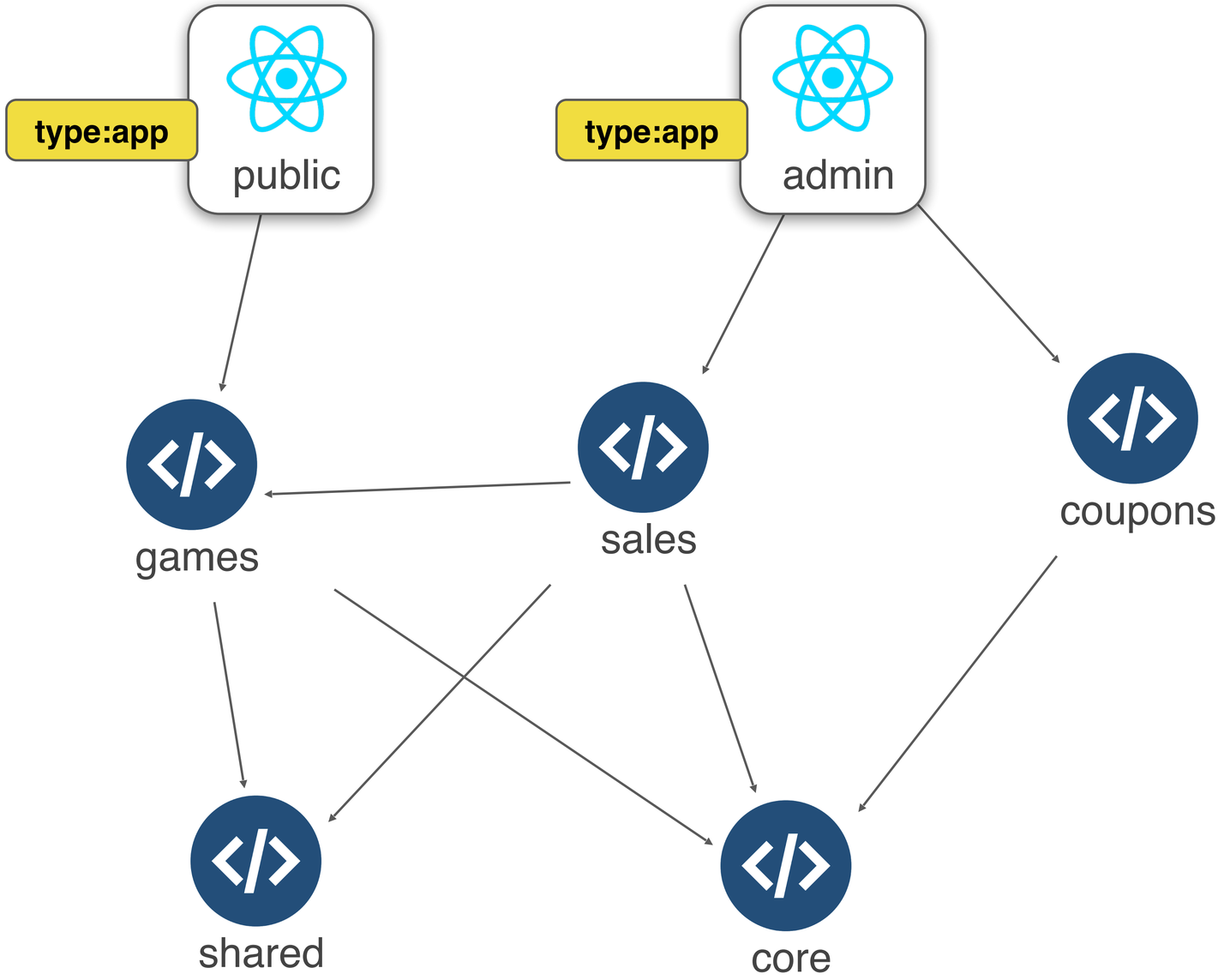
- 🔬 Lab 1 - Generate an empty workspace
- 🤿 Lab 2 - Generate an React app
- 🧪 Lab 3 - Executors
- 🔭 Lab 4 - Generate a component lib
- 🧬 Lab 5 - Generate a utility lib
- 🧱 Lab 6 - Generate a route lib
- 🤖 Lab 7 - Add an Express API
- 📐 Lab 8 - Displaying a full game in the routed game-detail component
- 💻 Lab 9 - Generate a type lib that the API and frontend can share
- 🤷‍♀️ Lab 10 - Generate Storybook stories for the shared ui component
- ⌨️ Lab 11 - E2E test the shared component

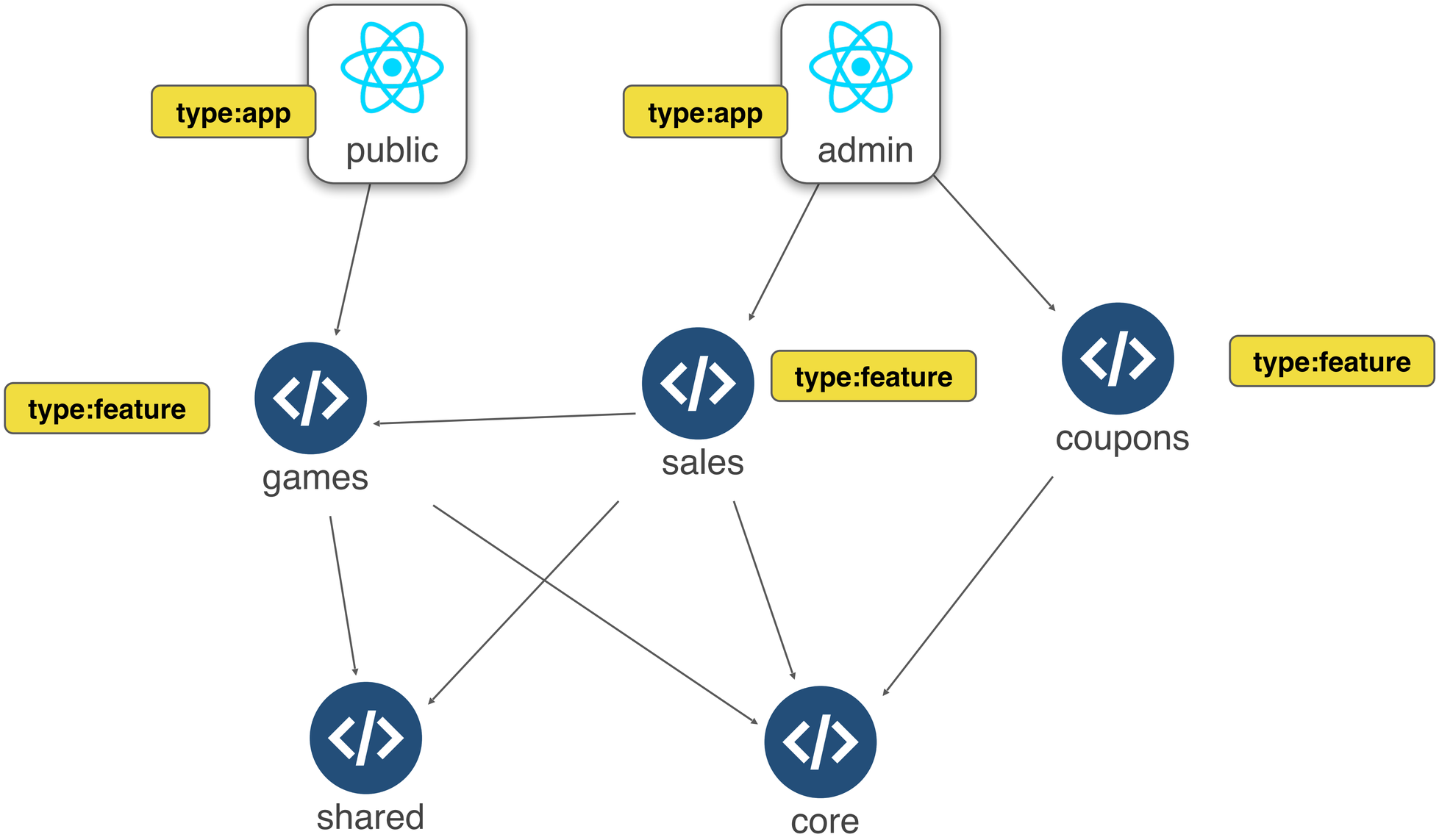
Day 2

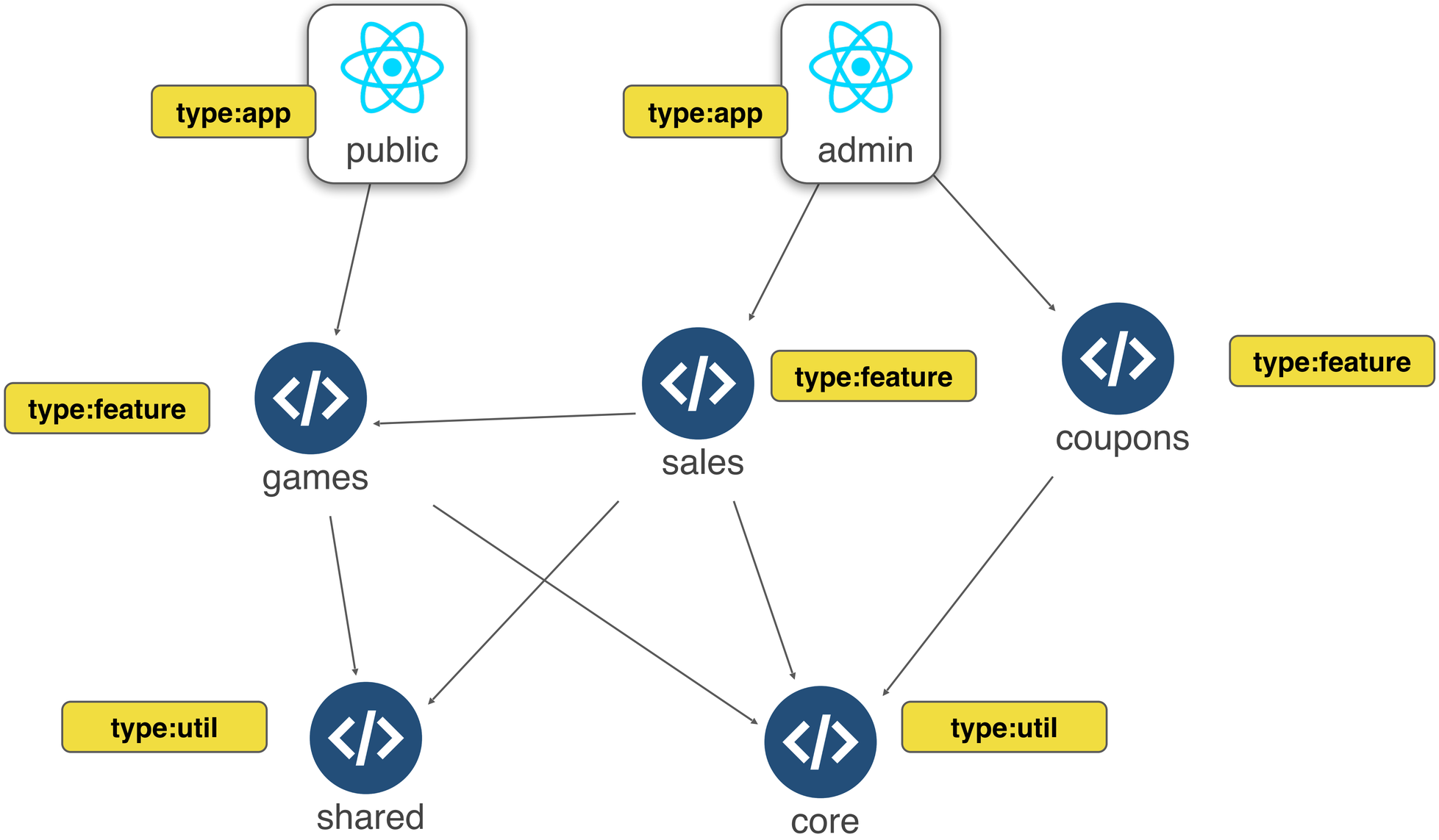
- 💡Lab 12 - Module boundaries
- 🧸Lab 13 - Workspace Generators - Intro
- 🧵Lab 14 - Workspace Generators - Modifying files
- 💎Lab 15 - Setting up CI
- 🔌Lab 16 - Distributed caching
- 🔍Lab 17 - NxCloud GitHub bot
- 📎Lab 18 - Run-Commands and deploying the frontend
- 🧲Lab 19 - Deploying the API
- 🎸Lab 20 - Connecting the frontend and backend
- 🎈Lab 21 - Setting up CD for automatic deployment
- 💈Lab 22 - Deploying only what changed

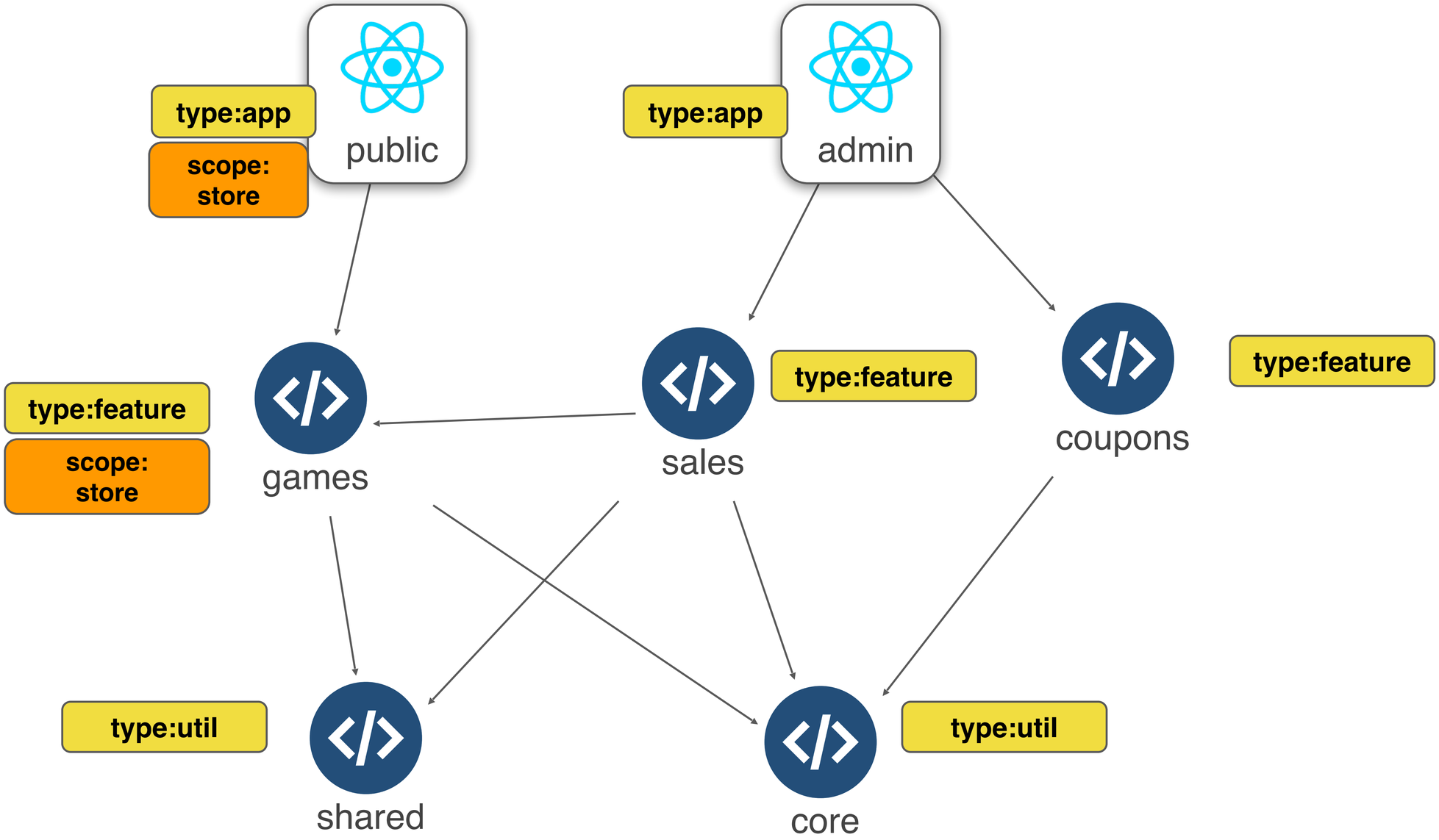

public
admin
games
sales
coupons
shared
core

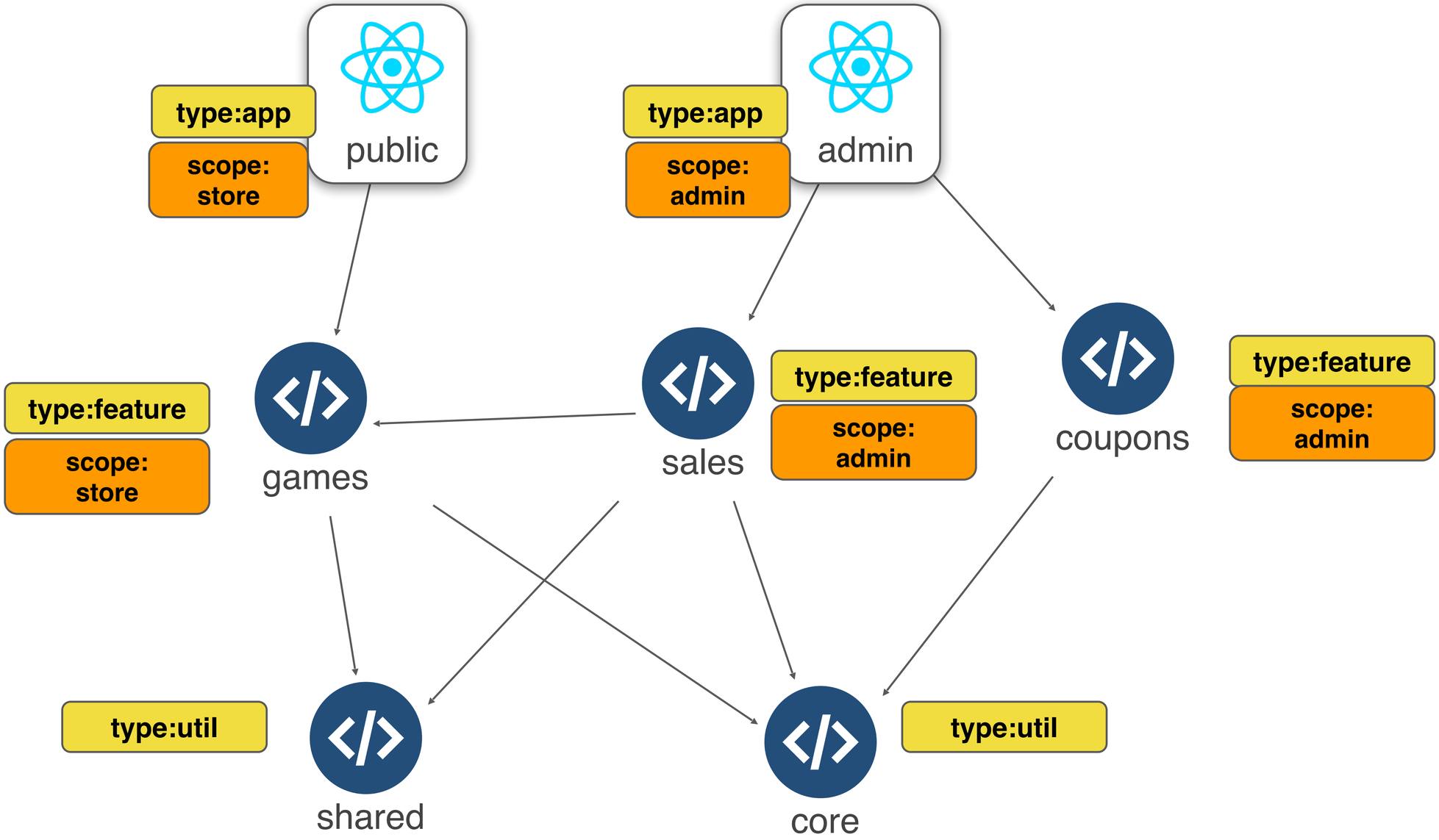

public
admin
games
sales
coupons
shared
core

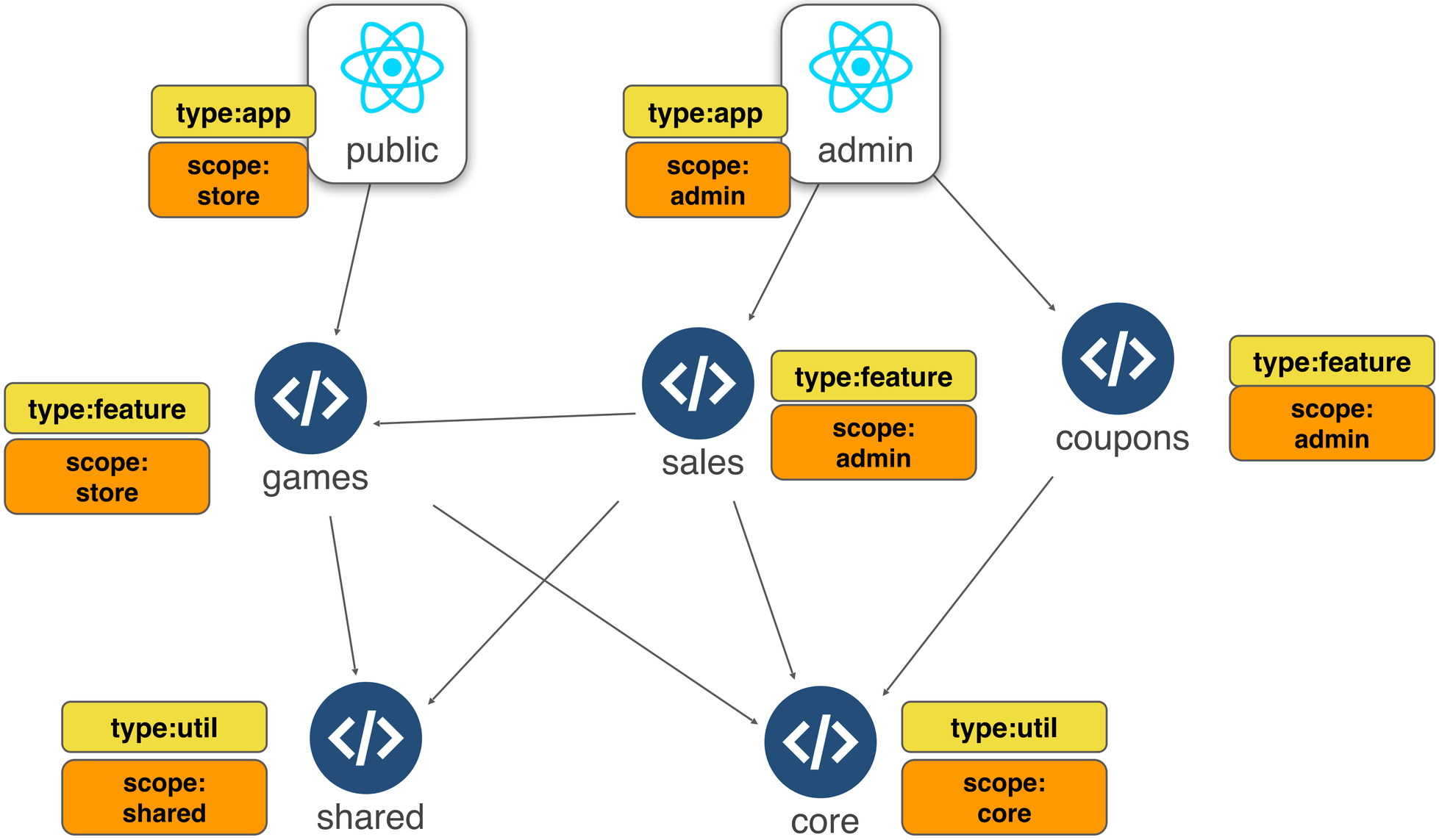

type:app
public
type:app
admin
games
sales
coupons
shared
core


type:app
public
type:app
admin
type:feature
games
sales
type:feature
coupons
type:feature
shared
core


type:app
public
type:app
admin
type:feature
games
type:feature
sales
type:feature
coupons
type:util
shared
type:util
core

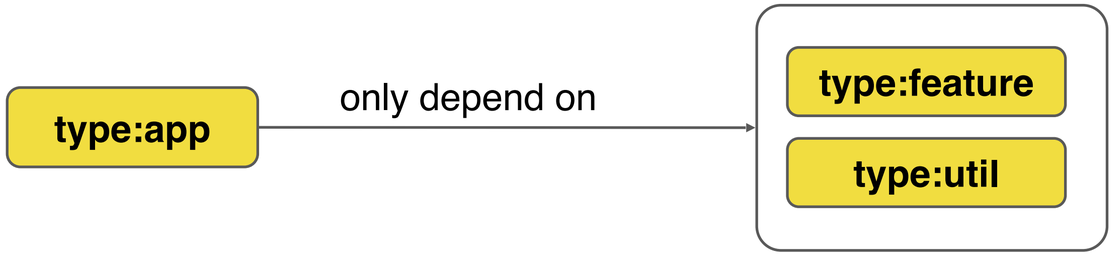

type:app
scope:
store
public
type:app
admin
type:feature
scope:
store
games
sales
type:feature
coupons
type:feature
type:util
shared
core
type:util

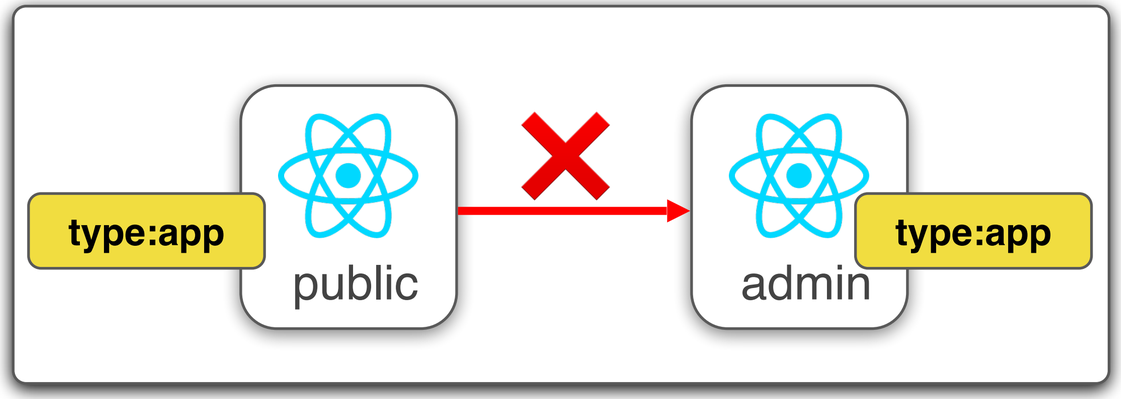

type:app
scope:
store
public
type:app
scope:
admin
admin
type:feature
scope:
store
games
sales
type:feature
scope:
admin
coupons
type:feature
scope:
admin
type:util
shared
core
type:util


type:app
scope:
store
public
type:app
scope:
admin
admin
type:feature
scope:
store
games
sales
type:feature
scope:
admin
coupons
type:feature
scope:
admin
type:util
scope:
shared
shared
core
type:util
scope:
core

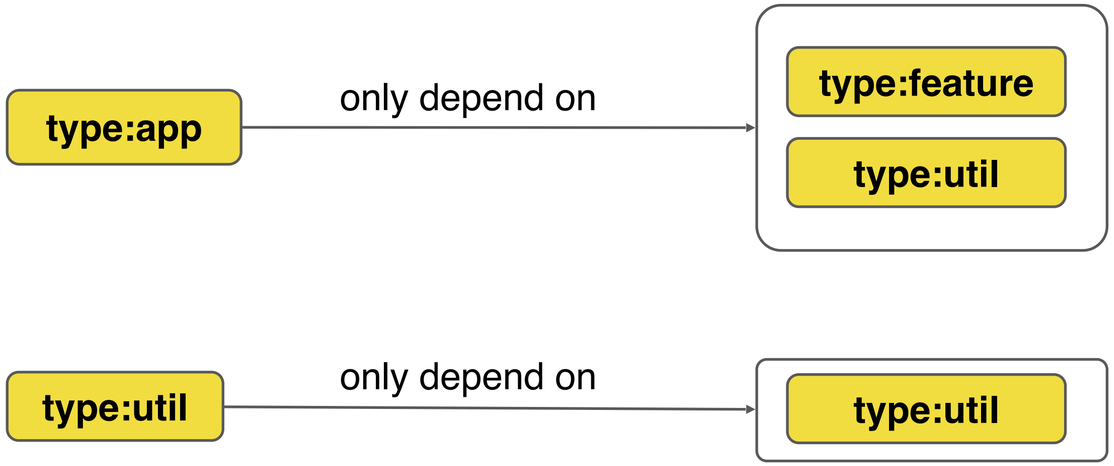
Defining **Rules**

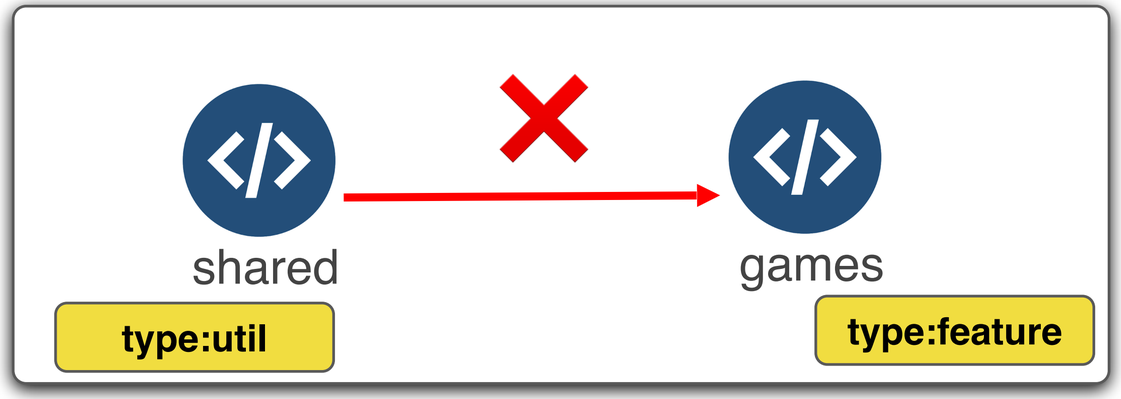
Restricting dependencies





Restricting dependencies



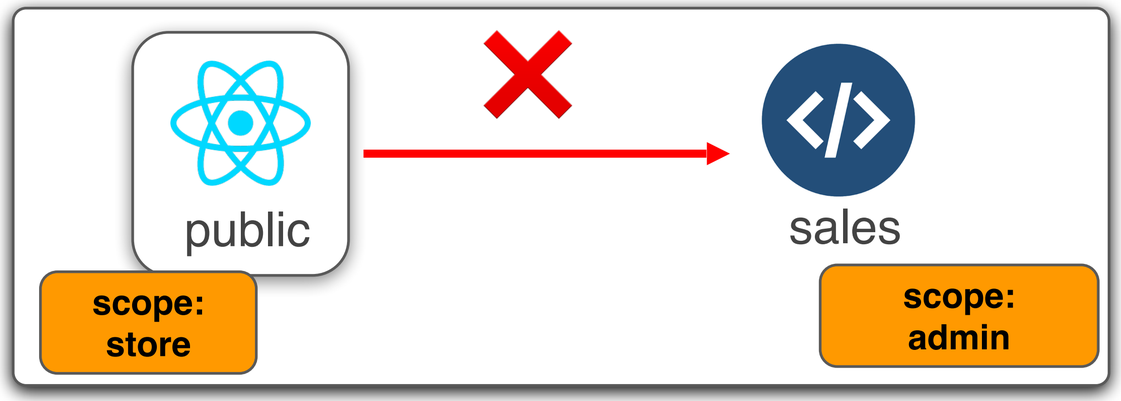


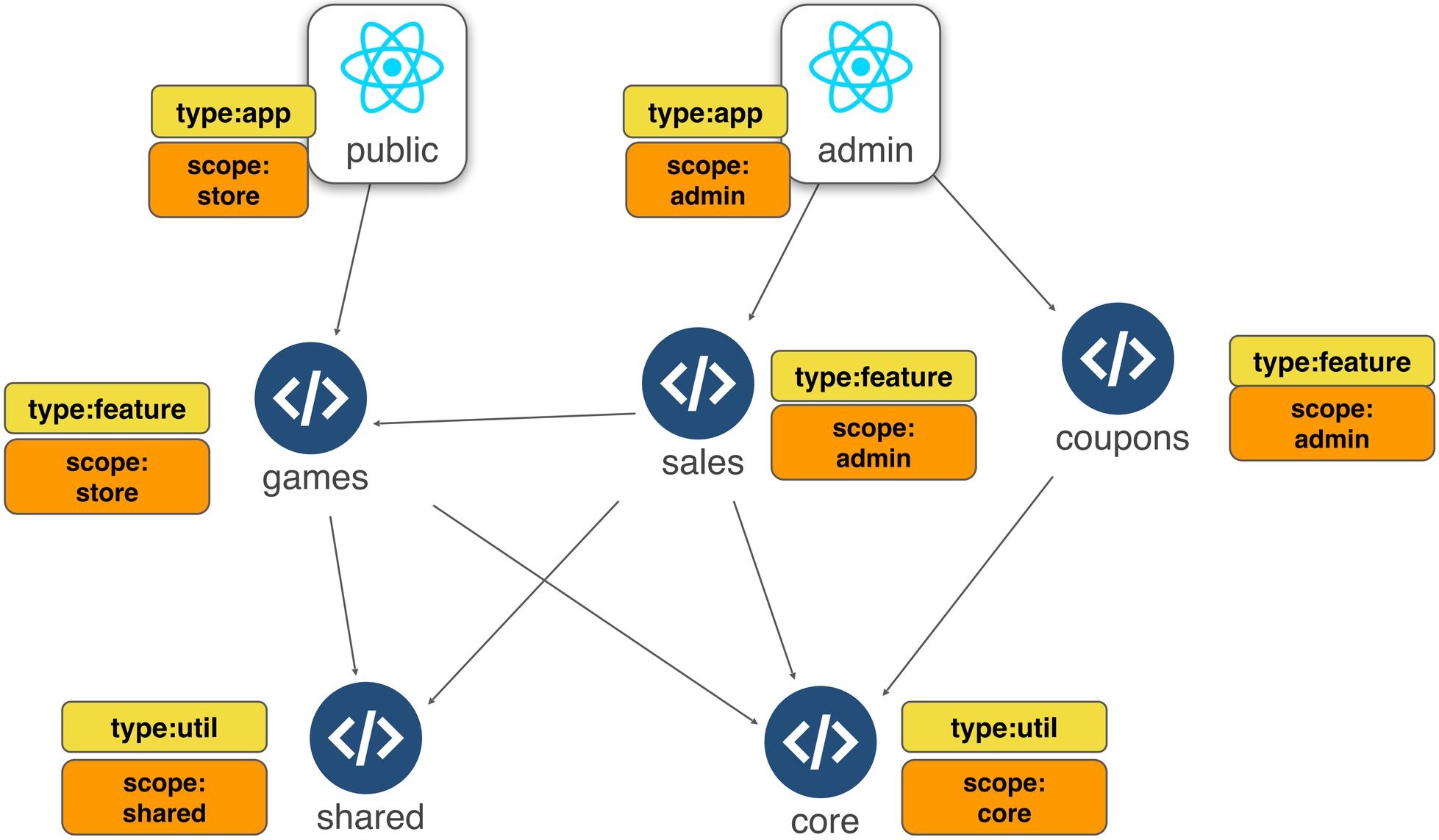
Restricting dependencies

type:app → only depend on → type:feature, type:util

type:util → only depend on → type:util

scope: store → only depend on → scope: store, scope: shared, scope: core




type:app
scope:
store
public
type:app
scope:
admin
admin
type:feature
scope:
store
games
sales
type:feature
scope:
admin
coupons
type:feature
scope:
admin
type:util
scope:
shared
shared
core
type:util
scope:
core

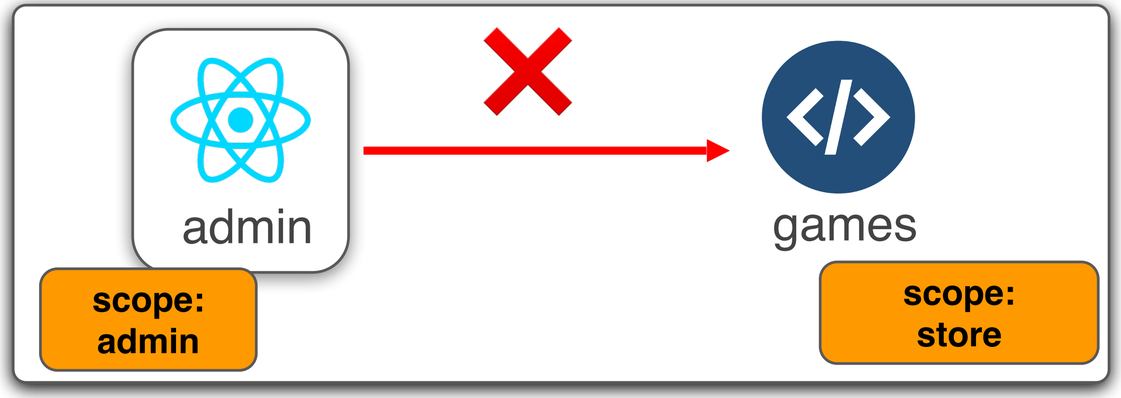
Restricting dependencies

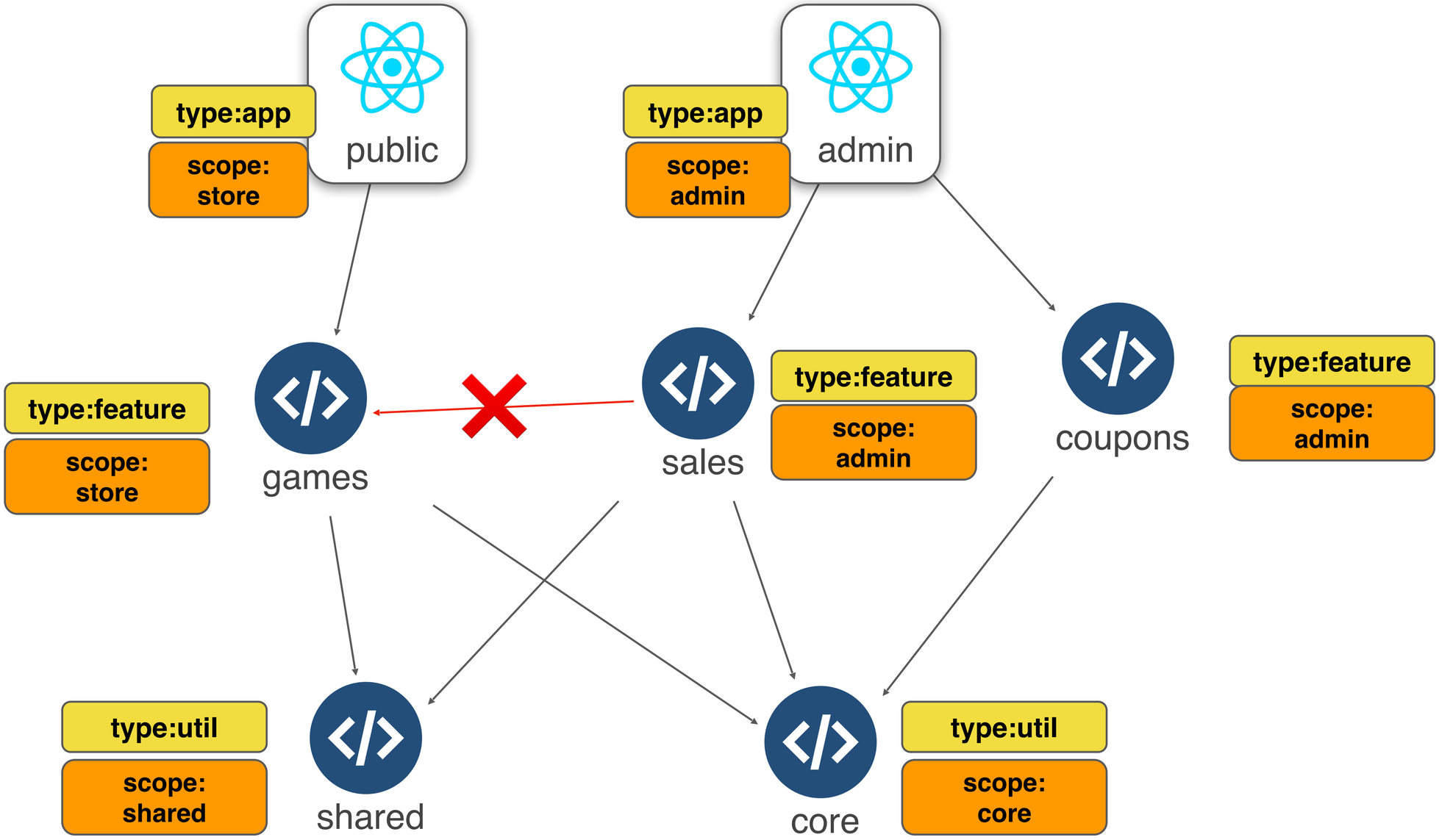
type:app only depend on type:feature, type:util

type:util only depend on type:util

scope: store only depend on scope: store, scope: shared, scope: core

scope: admin only depend on scope: admin, scope: shared, scope: core




type:app
scope:
store
public
type:app
scope:
admin
admin
type:feature
scope:
store
games
sales
type:feature
scope:
admin
coupons
type:feature
scope:
admin
type:util
scope:
shared
shared
core
type:util
scope:
core

Lab 12

Enforce Library Boundaries with Tags

Workspace Generators

Code example

Lab 13

Create Workspace Generators

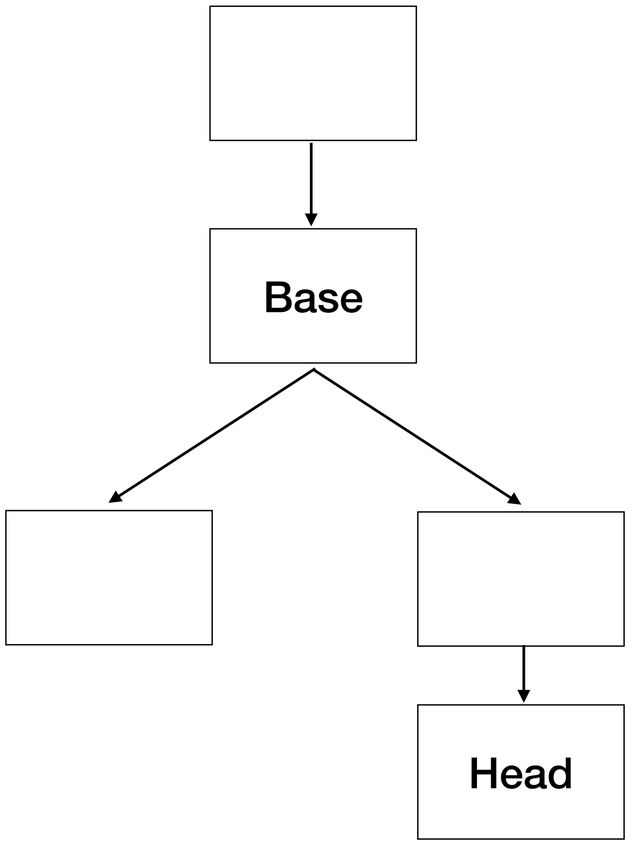
Lab 14

Modifying source code with workspace generators

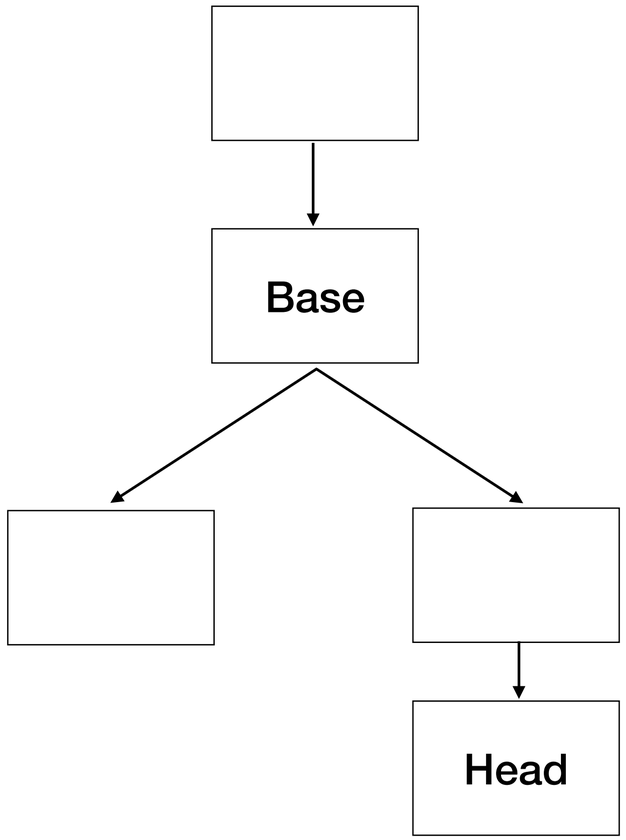
Lab 15

Setting up Github Workflows for Continuous Integration

Commits



Changed File List



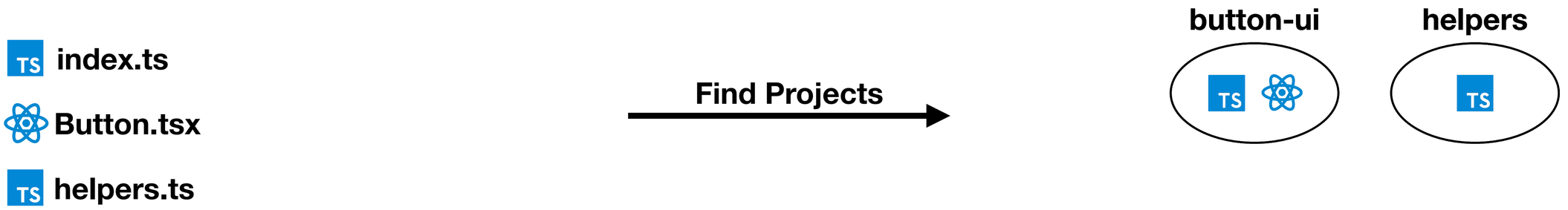
git diff

index.ts

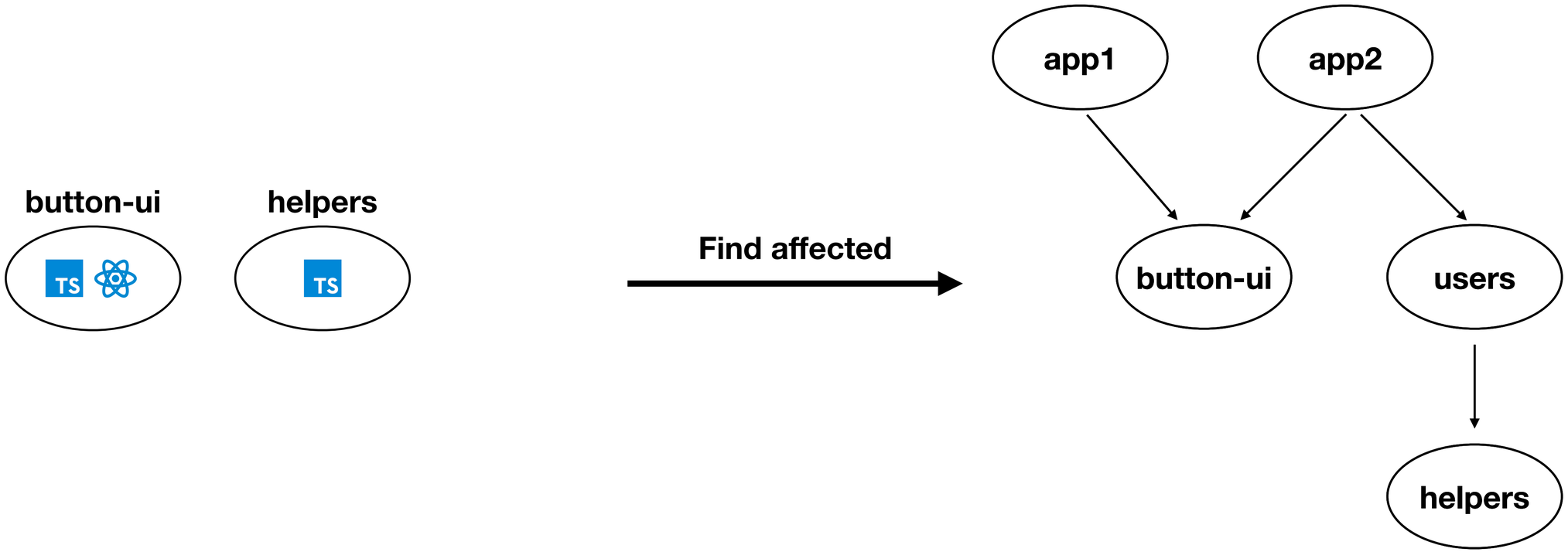
Button.tsx

helpers.ts

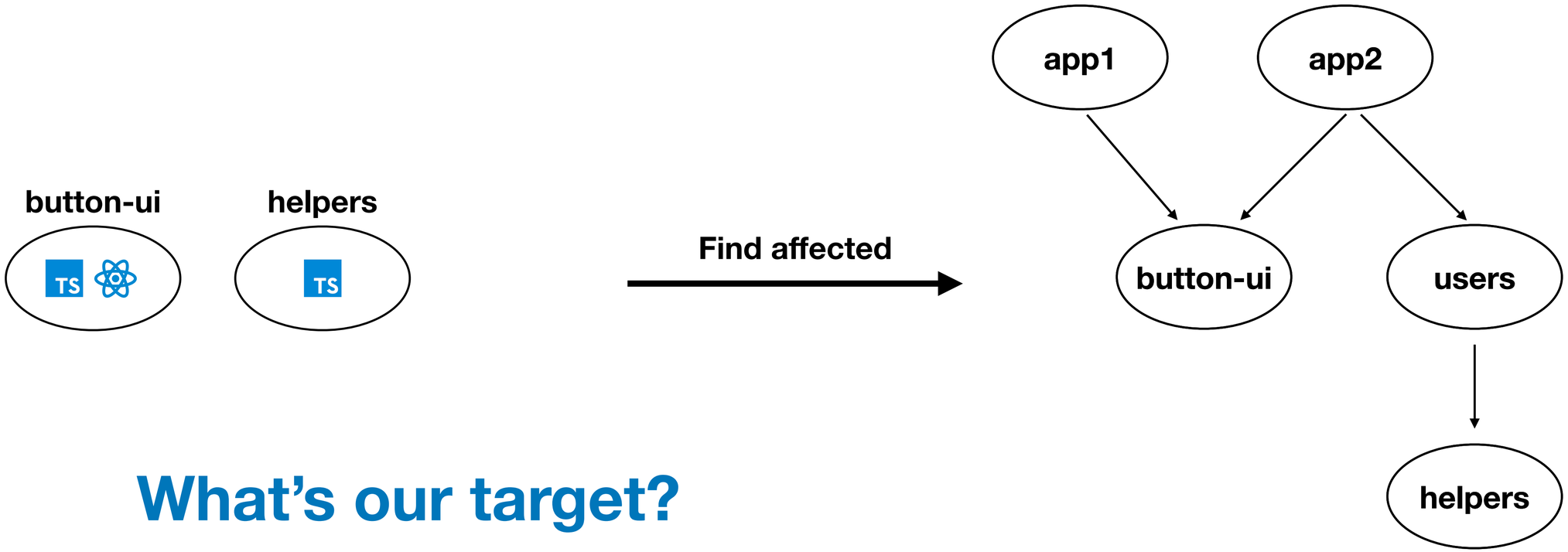
Changed Projects



Affected Projects

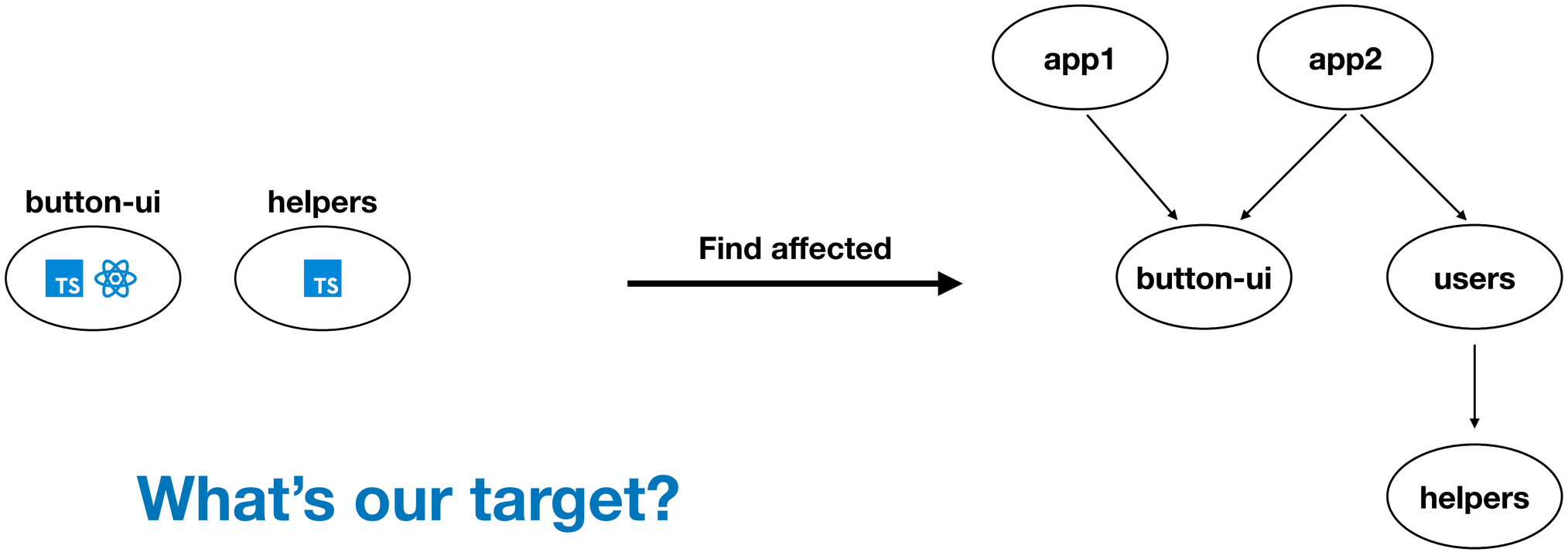


Affected Projects



What's our target?

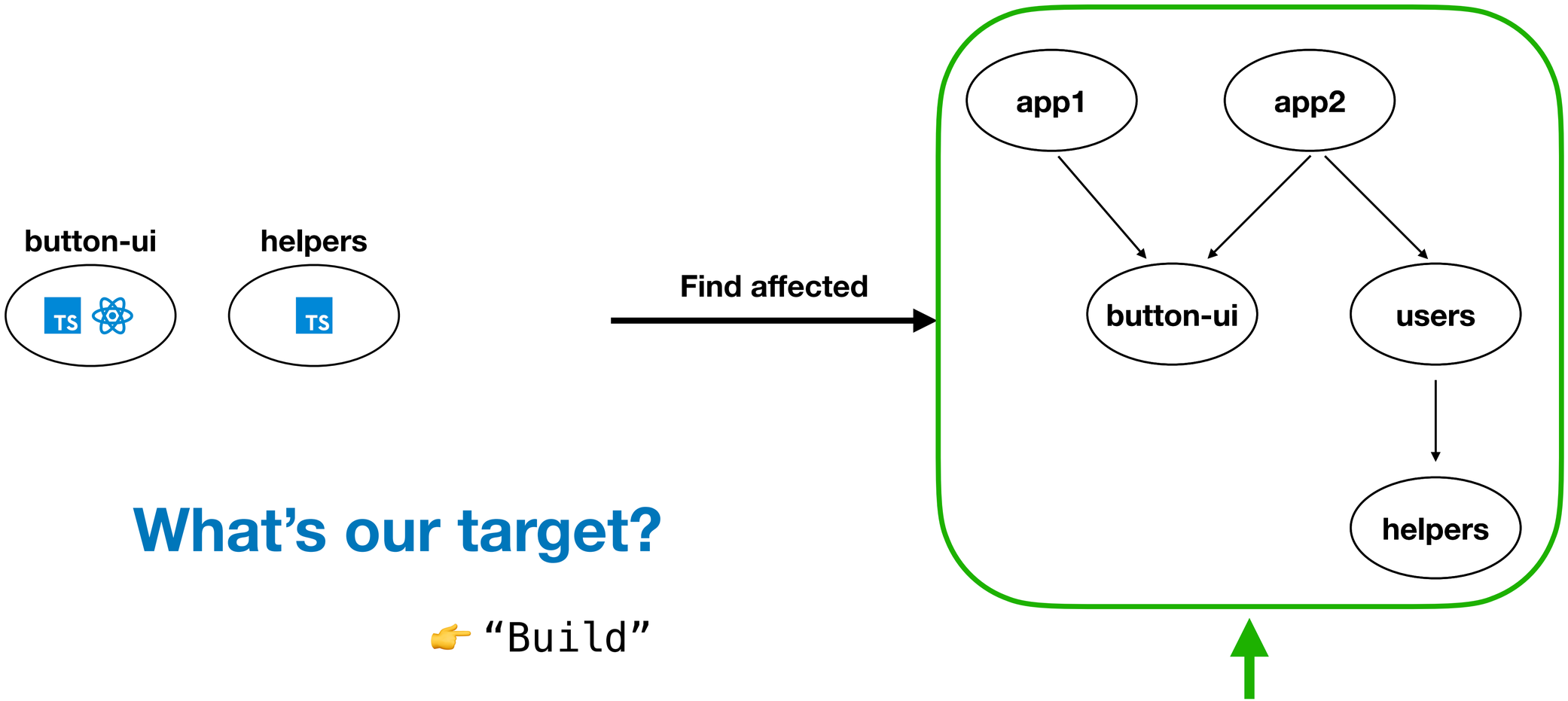
Affected Projects



What's our target?

👉 "Build"

Affected Projects

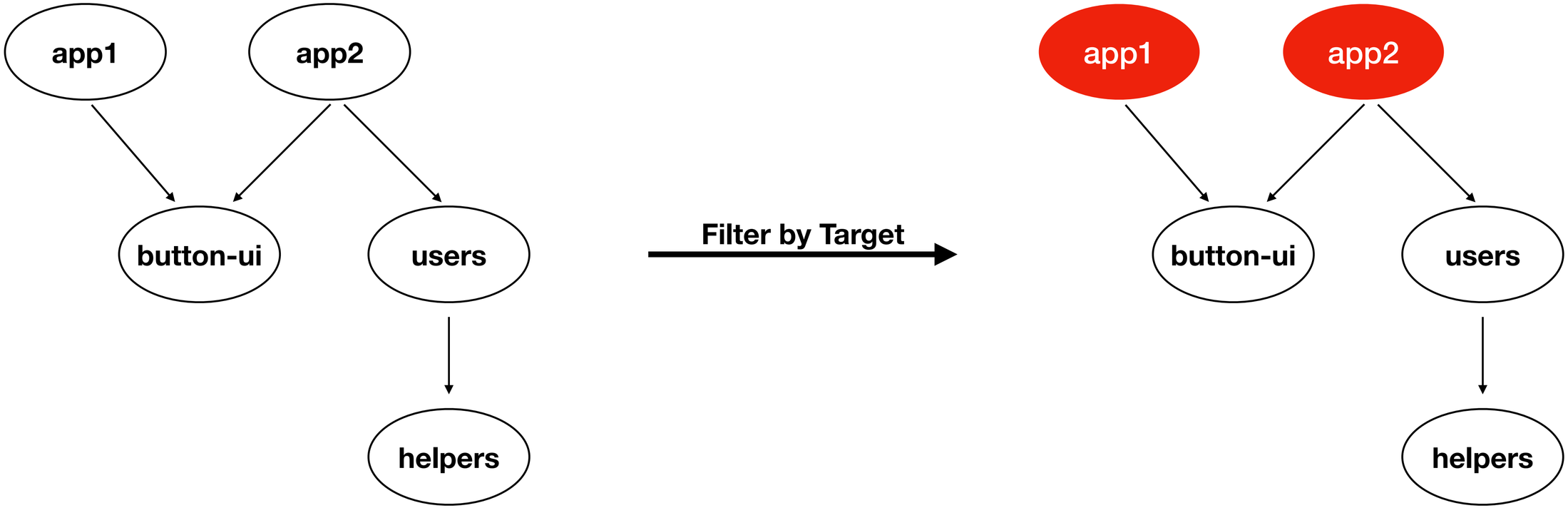


What's our target?

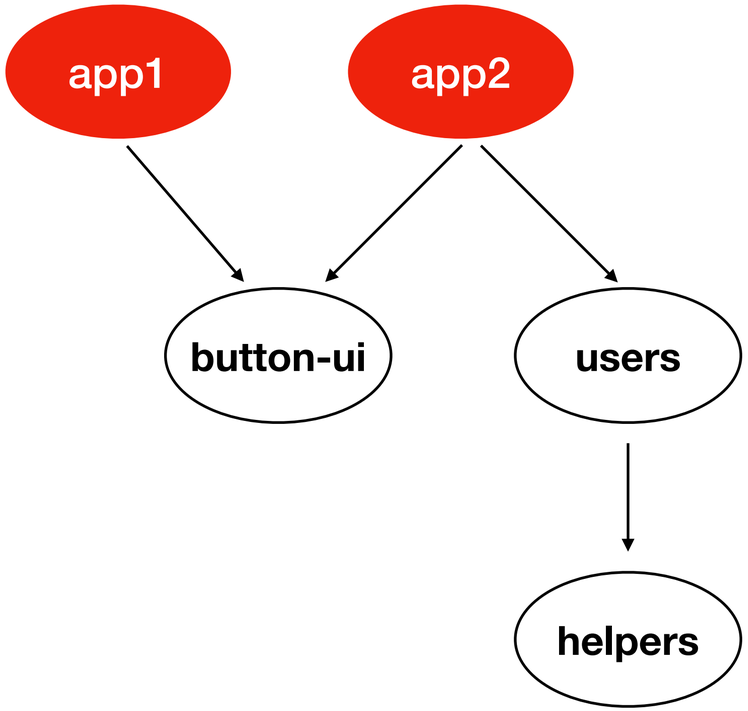
👉 "Build"

Which projects can be built?

Affected Projects with Target



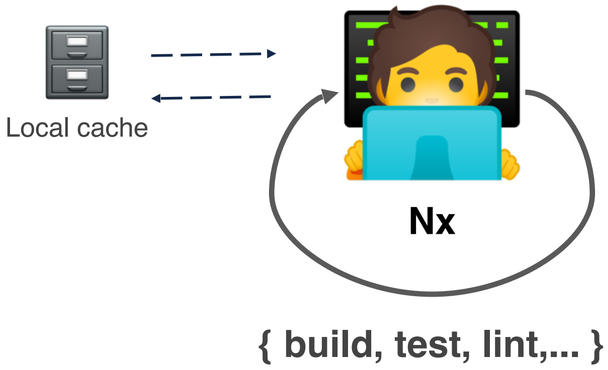
CLI Commands

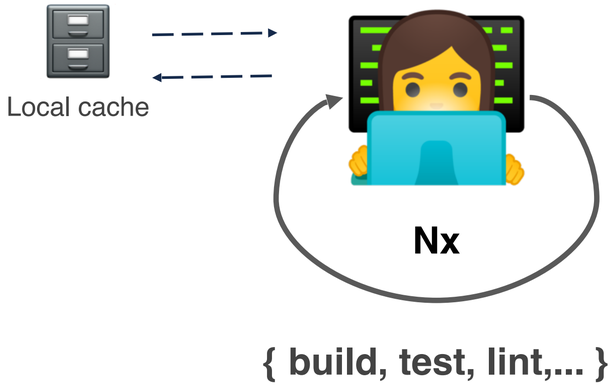


Run Commands

nx run app1:build
nx run app2:build

Computational Cache



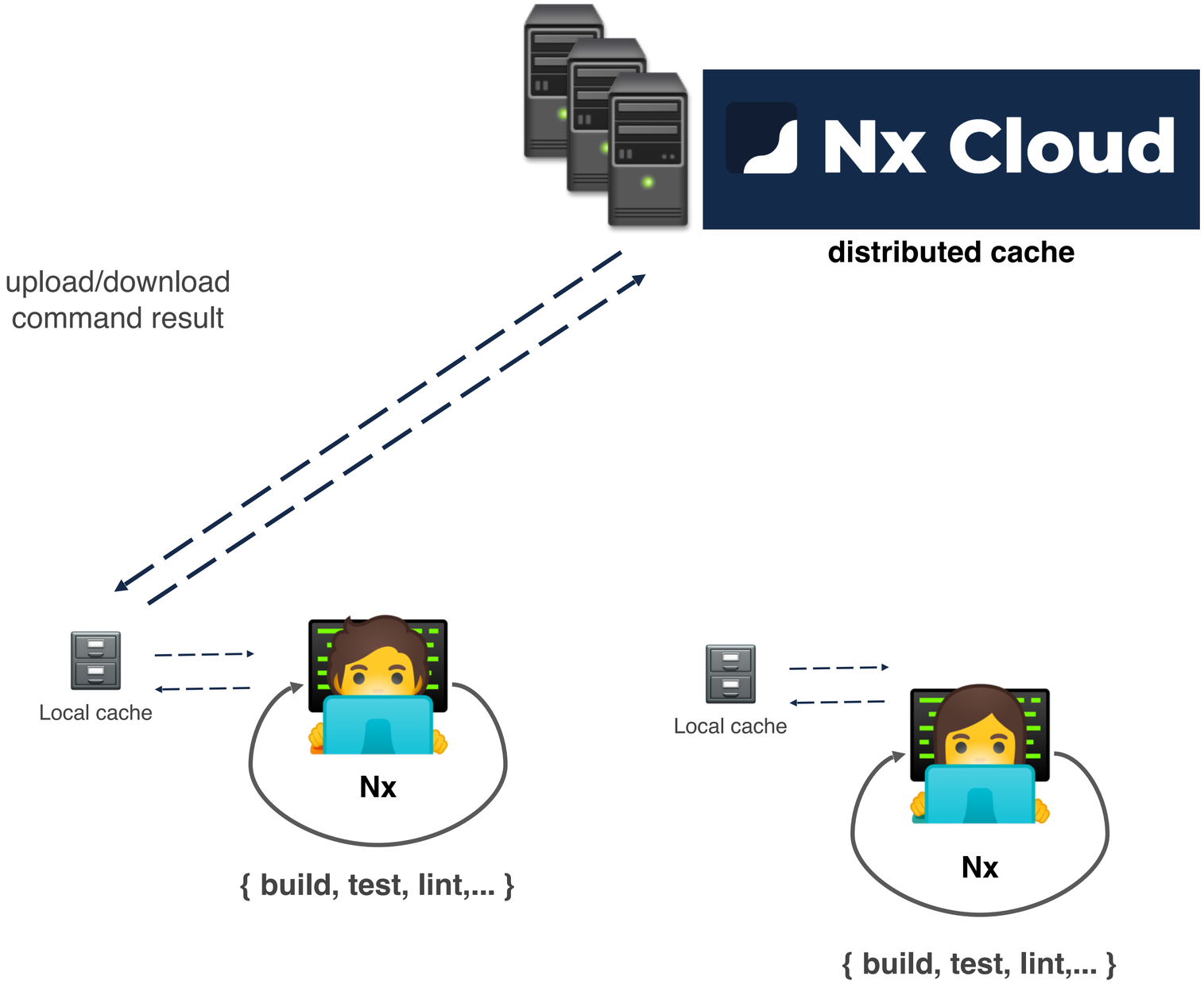


Nx Cloud

Never {build, test,...} twice

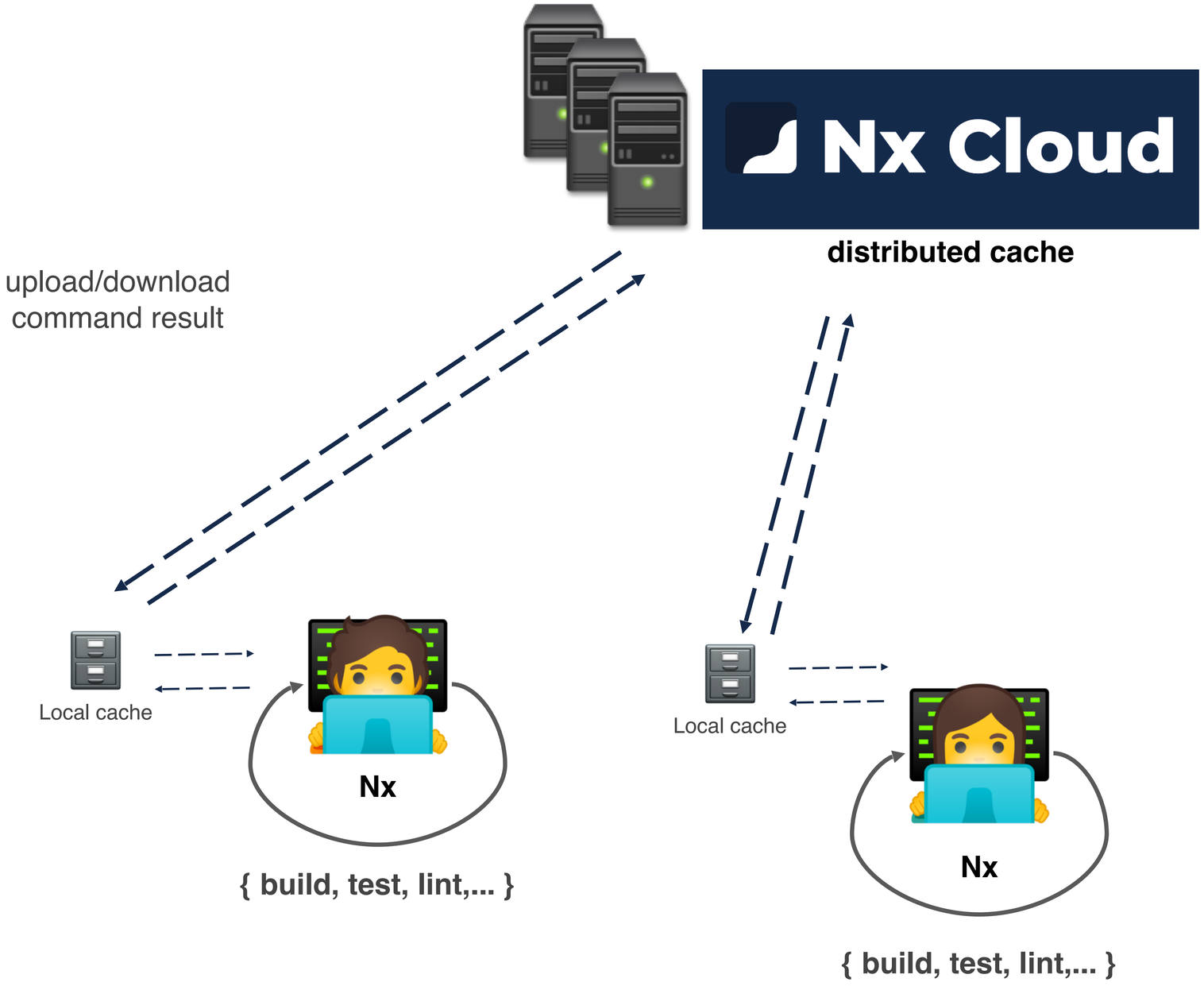
nx.app

Distributed computational cache



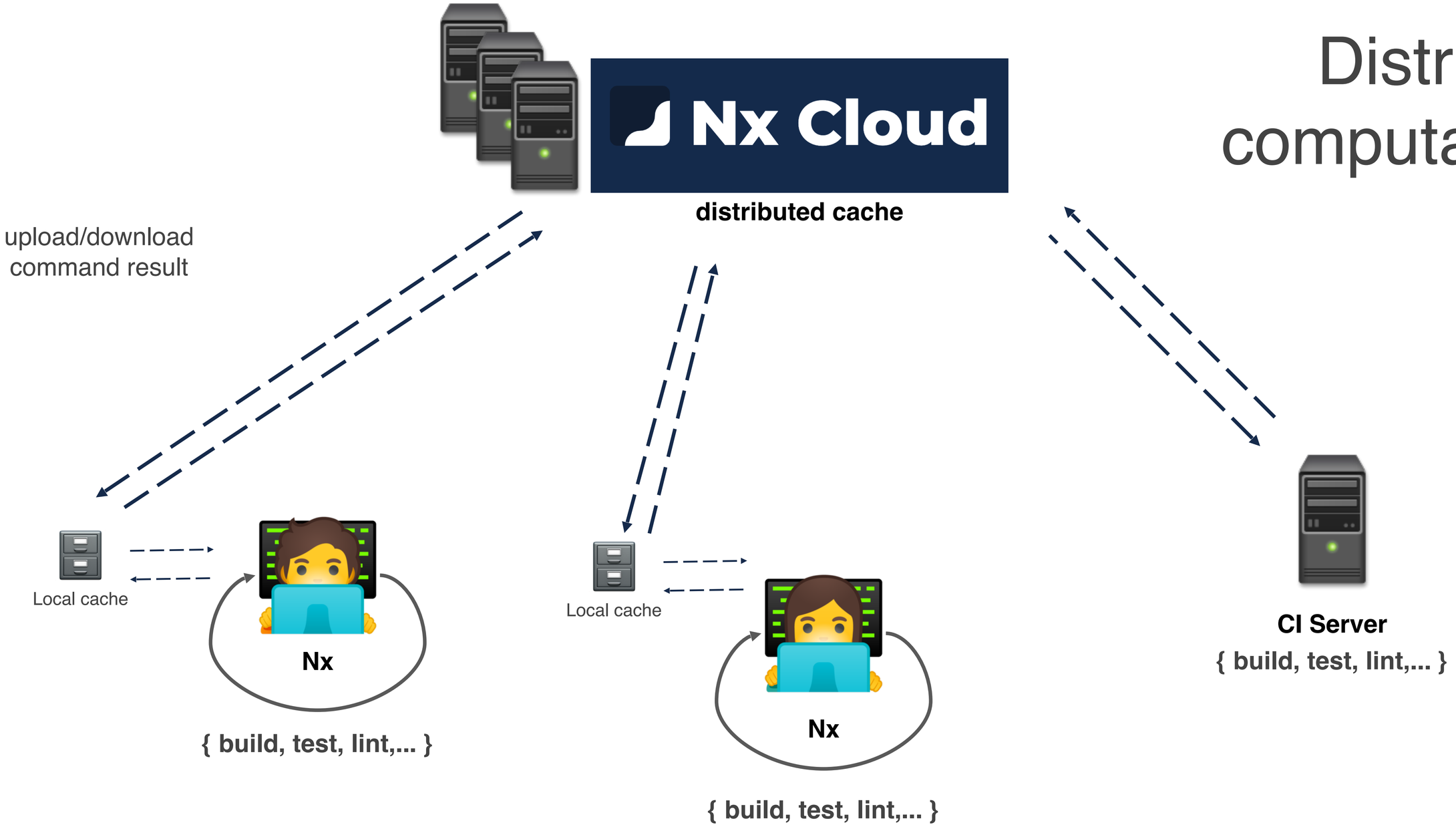
Victor Savkin: How to never build or test the same code twice

Distributed computational cache



Victor Savkin: How to never build or test the same code twice

Distributed computational cache



Victor Savkin: How to never build or test the same code twice

Lab 16

Local Caching and Nx Cloud’s Distributed Caching

Nx Cloud Github Bot

nx-cloud bot commented 7 days ago • edited

Nx Cloud Report

CI ran the following commands for commit `f6b966f`. Click to see the status, the terminal output, and the build insights.

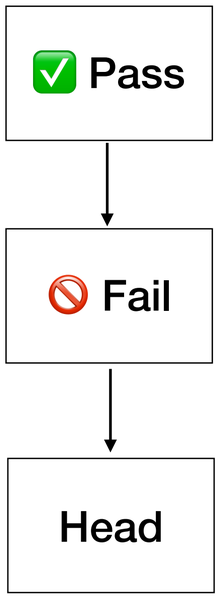
Status	Command	Start Time
✅	`nx build-base express`	11/8/2020, 3:48:54 PM
✅	`nx build-base nest`	11/8/2020, 3:48:54 PM
✅	`nx build-base next`	11/8/2020, 3:48:49 PM
✅	`nx build-base react`	11/8/2020, 3:48:49 PM
✅	`nx run-many --target=build --all --parallel`	11/8/2020, 3:47:34 PM
✅	`nx run-many --target=e2e --projects=e2e-angular`	11/8/2020, 3:48:49 PM
✅	`nx run-many --target=e2e --projects=e2e-cypress,e2e-jest,e2e-nx-plugin`	11/8/2020, 3:48:59 PM
✅	`nx run-many --target=e2e --projects=e2e-next`	11/8/2020, 3:50:10 PM
✅	`nx run-many --target=e2e --projects=e2e-node`	11/8/2020, 3:48:39 PM
✅	`nx run-many --target=e2e --projects=e2e-react`	11/8/2020, 3:48:41 PM
✅	`nx run-many --target=e2e --projects=e2e-web,e2e-linter,e2e-storybook`	11/8/2020, 3:48:36 PM
✅	`nx run-many --target=e2e --projects=e2e-workspace,e2e-cli`	11/8/2020, 3:49:05 PM
❌	`nx run-many --target=test --all --parallel`	11/8/2020, 3:47:38 PM

Sent with ❤️ from NxCloud.

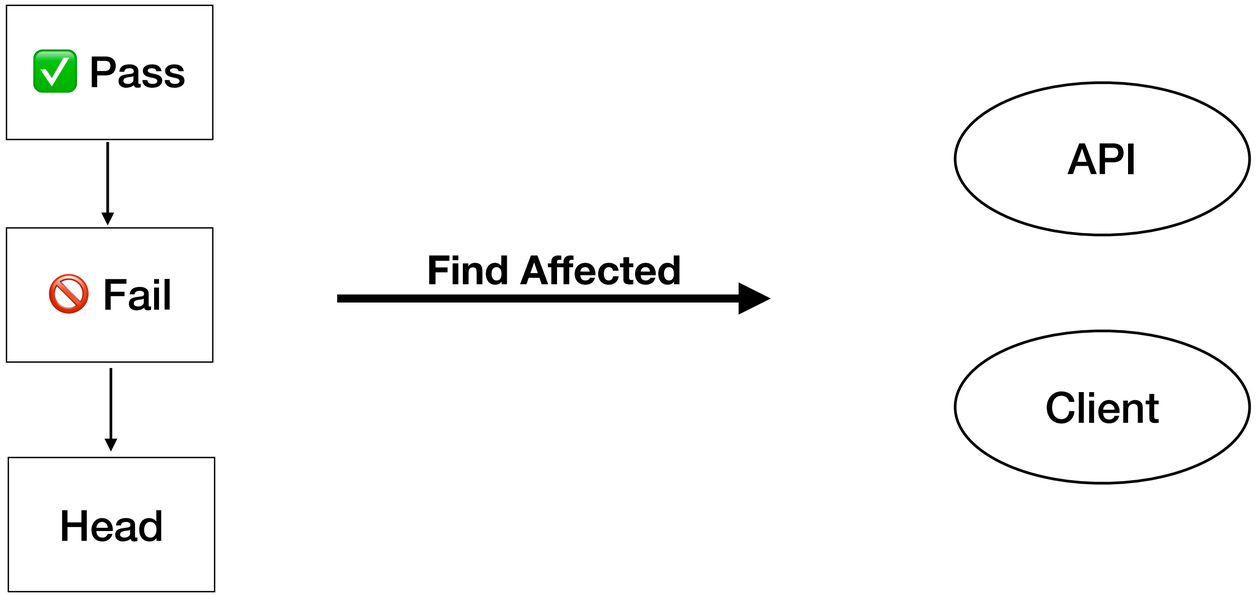
Lab 17

Nx Cloud Github Bot

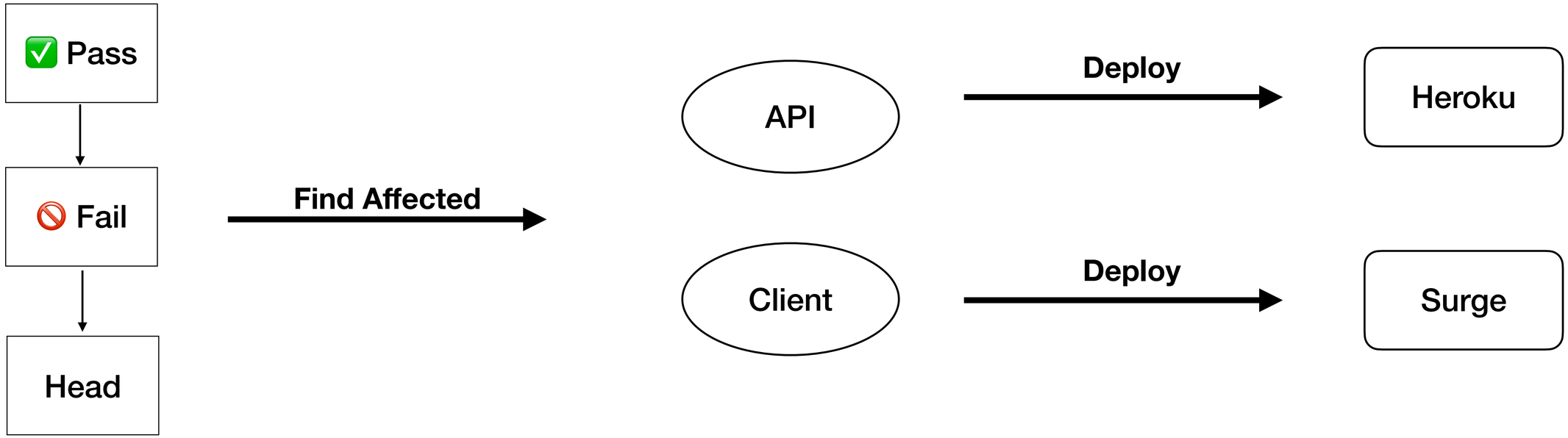
Deploying to Staging



Deploying to Staging



Deploying to Staging



Lab 18

Custom Executors with Run Commands

Lab 19

Deploying the API

Lab 20

Connecting the frontend and backend

Lab 21

Setting up automated continuous deployment

Lab 22

Deploying only what’s been affected

Poll / Q&A