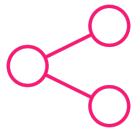
Enforcing Boundaries with Nx and Tooling

Zachary Bennett
Lead Software Developer

@z_bennett_ | https://www.linkedin.com/in/zbennett10

Nx is a popular choice for managing multiple Angular apps and libraries!

Nx Monorepo Benefits at Scale



Code sharing

Optimized build and development performance

Enhanced maintainability

Demo: Working in an Angular Monorepo

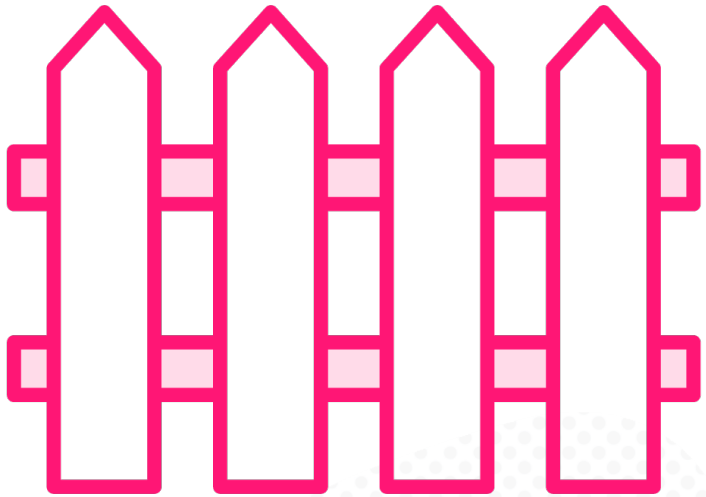
Defining Architectural Boundaries

Multi-team domains

Dependency rules

Module boundaries

Tags and linting errors



Multiple-team Project Support

Scale well by using tags associated different modules/projects

nx.json

```
{
  "projects": {
    "@sleek-store/sleek-store": { "tags": ["scope:app", "type:app"] },
    "@sleek-store/user": { "tags": ["scope:feature-user", "type:lib"] },
    "@sleek-store/cart": { "tags": ["scope:shared", "type:lib"] }
  },

  <snip>
}
```

Enforce Module Boundaries with ESLint

ESLint integrates seamlessly with Nx to enforce module boundaries for apps/libs

.eslintrc.json

```
"@nx/enforce-module-boundaries":

  <snip>

  "depConstraints": [
    {
      "sourceTag": "scope:shared",
      "onlyDependOnLibsWithTags": ["scope:shared"]
    }

   <snip>
  ]
```

Configuring Nx Module Boundaries

Maintaining Encapsulation with Tooling

Fixing module boundary violations is important for scalability!

Maintaining Module Boundaries