

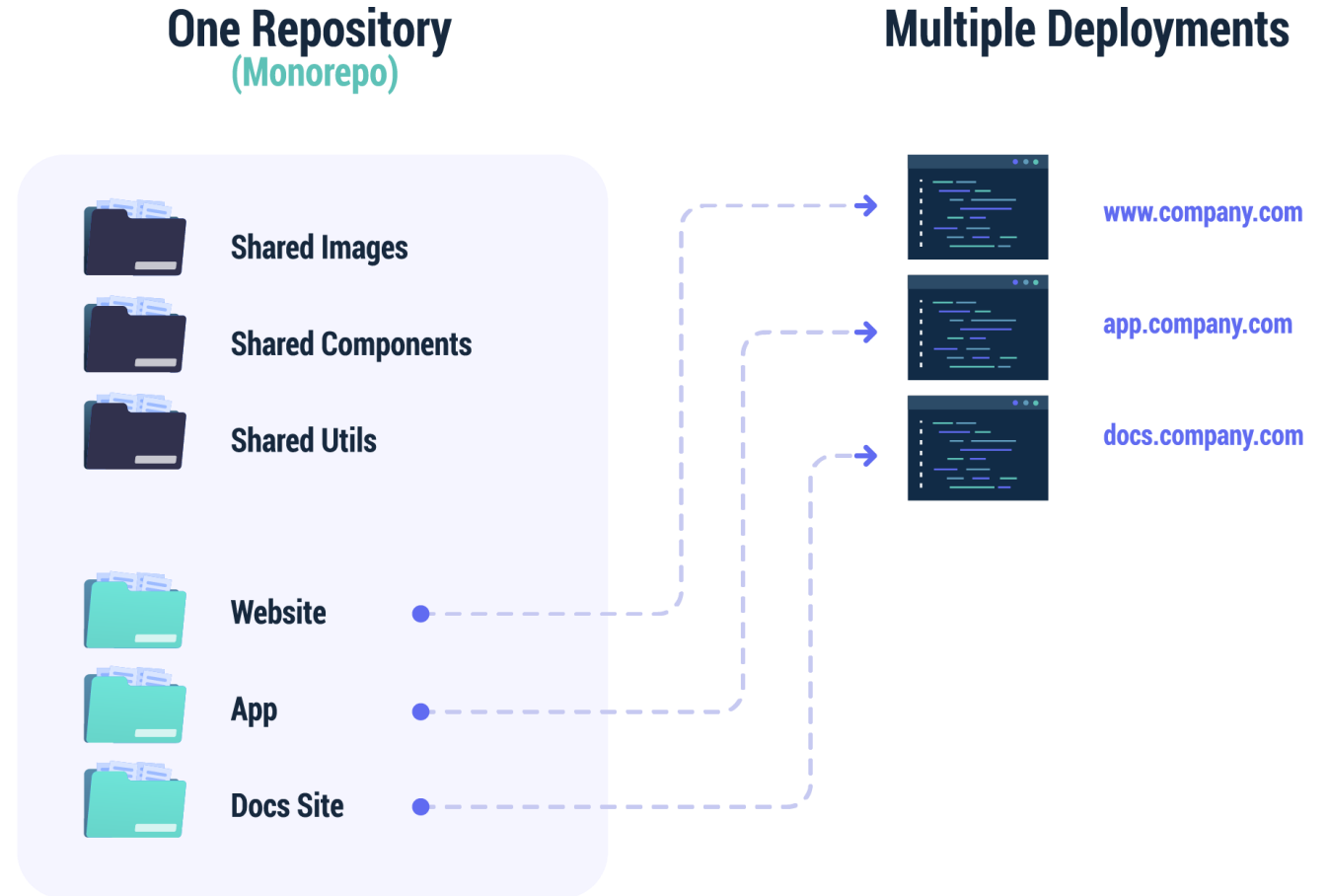


Monorepos

Accenture

Sofention

Monolithic Repository approach,
to keep the code for multiple projects
stored under in a single repository,
facilitating the ci-cd practices,
With a well-defined relationships.



About me...

Arturs Jermolickis

Sr. Backend engineer at
Accenture Baltics



- Based in Latvia
Visiting around the world.
- Available at:
arturs.jermolickis@accenture.com

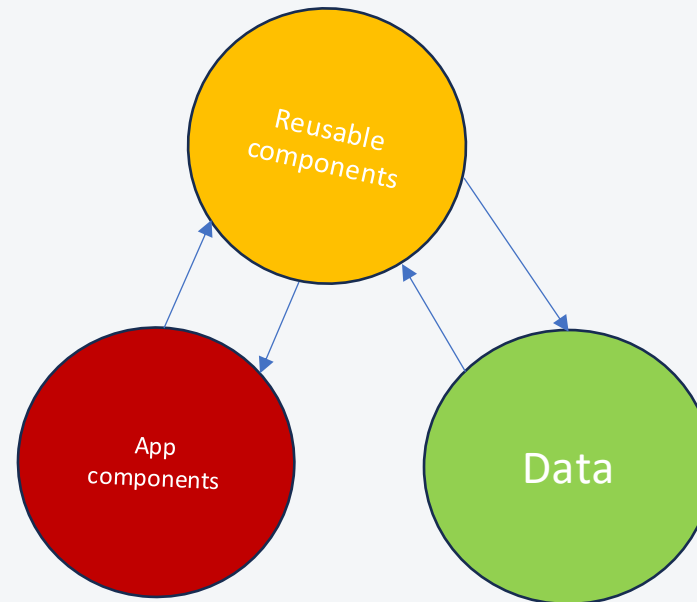
Monorepo is monolith?

Monolith is not Monorepo

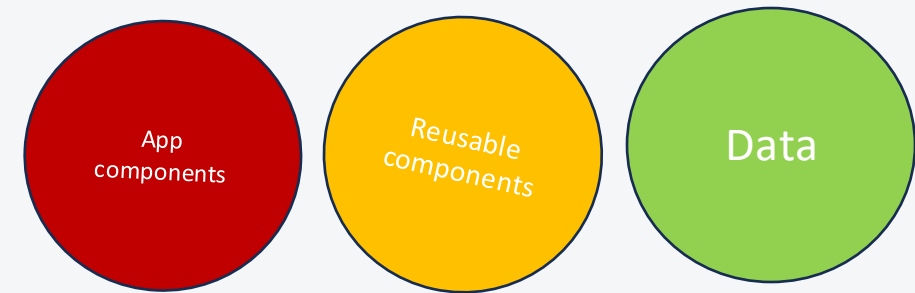
- Monolith is the way to deploy and link the code. In a monolithic architecture, components of the program are interconnected and interdependent.
- Monorepo is the way of code organisation. This is a code organization strategy where multiple projects are stored in a single repository (repo).

Monolith App repo and Modular App repo

Monolith App Repo:



Modular App Repo:



Relationships

- Key element for the monorepo
- Achieved by tooling

NX

- Nx is a fast, smart and extensible build system
- Helps teams develop applications at any scale.
- Integrates with modern frameworks and libraries
- Provides
 - computation caching and smart rebuilds
 - code generators.

Prerequisites

1. Node.js

- Nx requires a recent version of Node.js.

2. npm, Yarn, or pnpm

- These are package managers for JavaScript

3. Git

- Nx leverages Git for managing projects within a monorepo.

4. Knowledge of JavaScript/TypeScript

- Nx is commonly used for JavaScript and TypeScript projects.
- knowledge in the specific framework you plan to use (like Angular, React, or Node.js for backend applications) is also beneficial.

5. Understanding of Monorepos

- Nx is particularly powerful for managing monorepos

6. IDE/Text Editor

Creating a workspace

Quick guide on How to start using NX

Creating a first app.

What benefits are visible from the beginning

```
arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
? Which stack do you want to use? ... █
```

None:	Configures a TypeScript/JavaScript project with minimal structure.
<u>React:</u>	<u>Configures a React application with your framework of choice.</u>
Vue:	Configures a Vue application with modern tooling.
Angular:	Configures a Angular application with modern tooling.
Node:	Configures a Node API application with your framework of choice.

```
arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
✓ Which stack do you want to use? · react
```

```
? What framework would you like to use? ... █
```

```
None I only want react and react-dom
```

```
Next.js [ https://nextjs.org/ ]
```

```
Expo [ https://expo.io/ ]
```

```
React Native [ https://reactnative.dev/ ]
```

```
[arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
[✓ Which stack do you want to use? · react
```

```
[✓ What framework would you like to use? · none
```

```
? Integrated monorepo, or standalone project? ... █
```

Integrated Monorepo: Nx creates a monorepo that contains multiple projects.

Standalone: Nx creates a single project and makes it fast.

```
arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
✓ Which stack do you want to use? · react
```

```
✓ What framework would you like to use? · none
```

```
✓ Integrated monorepo, or standalone project? · integrated
```

```
? Application name > jsapp
```

```
arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
✓ Which stack do you want to use? · react
```

```
✓ What framework would you like to use? · none
```

```
✓ Integrated monorepo, or standalone project? · integrated
```

```
✓ Application name · jsapp
```

```
? Which bundler would you like to use? ... █
```

```
Vite [ https://vitejs.dev/ ]
```

```
Webpack [ https://webpack.js.org/ ]
```

```
Rspack [ https://www.rspack.dev/ ]
```



```
[arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
[✓ Which stack do you want to use? · react
```

```
[✓ What framework would you like to use? · none
```

```
[✓ Integrated monorepo, or standalone project? · integrated
```

```
✓ Application name · jsapp
```

```
[✓ Which bundler would you like to use? · webpack
```

```
? Test runner to use for end to end (E2E) tests ... █
```

```
Cypress [ https://www.cypress.io/ ]
```

```
Playwright [ https://playwright.dev/ ]
```

```
None
```



```
arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
✓ Which stack do you want to use? · react
✓ What framework would you like to use? · none
✓ Integrated monorepo, or standalone project? · integrated
✓ Application name · jsapp
✓ Which bundler would you like to use? · webpack
✓ Test runner to use for end to end (E2E) tests · cypress
? Default stylesheet format ...
```

CSS

```
SASS(.scss)      [ http://sass-lang.com      ]
LESS              [ http://lesscss.org        ]
styled-components [ https://styled-components.com      ]
emotion           [ https://emotion.sh          ]
styled-jsx        [ https://www.npmjs.com/package/styled-jsx ]
```

```
[arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

```
[✓ Which stack do you want to use? · react
```

```
[✓ What framework would you like to use? · none
```

```
[✓ Integrated monorepo, or standalone project? · integrated
```

```
✓ Application name · jsapp
```

```
[✓ Which bundler would you like to use? · webpack
```

```
[✓ Test runner to use for end to end (E2E) tests · cypress
```

```
[✓ Default stylesheet format · css
```

```
? Enable distributed caching to make your CI faster ... █
```

```
Yes I want faster builds
```

```
No
```

```
[arturs.jermolickis@AMAC02FTEJLMD6M ~ % npx create-nx-workspace jsconf
```

```
> NX Let's create a new workspace [https://nx.dev/getting-started/intro]
```

- ✓ Which stack do you want to use? · [react](#)
- ✓ What framework would you like to use? · [none](#)
- ✓ Integrated monorepo, or standalone project? · [integrated](#)
- ✓ Application name · [jsapp](#)
- ✓ Which bundler would you like to use? · [webpack](#)
- ✓ Test runner to use for end to end (E2E) tests · [cypress](#)
- ✓ Default stylesheet format · [css](#)
- ✓ Enable distributed caching to make your CI faster · [Yes](#)

```
> NX Creating your v17.2.3 workspace.
```

To make sure the command works reliably in all environments, and that the preset is applied correctly, Nx will run "npm install" several times. Please wait.

```
⌄ Installing dependencies with npm
```

```
✓ Which stack do you want to use? · react
✓ What framework would you like to use? · none
✓ Integrated monorepo, or standalone project? · integrated
✓ Application name · jsapp
✓ Which bundler would you like to use? · webpack
✓ Test runner to use for end to end (E2E) tests · cypress
✓ Default stylesheet format · css
✓ Enable distributed caching to make your CI faster · Yes

> NX Creating your v17.2.3 workspace.

To make sure the command works reliably in all environments, and that the preset is applied correctly,
Nx will run "npm install" several times. Please wait.

✓ Installing dependencies with npm
✓ Successfully created the workspace: jsconf.
✓ NxCloud has been set up successfully
Python Validation Successful
cwd /Users/arturs.jermolickis/jsconf
config_ini_file /Users/arturs.jermolickis/.acn-security-tools/config/private/private_secrets_dictionary.ini
[WARN] File apps/jsapp/src/favicon.ico is encoded in an unsupported format and was not checked! utf-8
[WARN] Tier 2 secret(s) found in file 'apps/jsapp-e2e/src/support/commands.ts' - check lines [17, 22, 23]
```

```
> NX Nx CLI is not installed globally.

This means that you might have to use "yarn nx" or "npx nx" to execute commands in the workspace.
Run "yarn global add nx" or "npm install -g nx" to be able to execute command directly.

> NX Distributed caching via Nx Cloud has been enabled

In addition to the caching, Nx Cloud provides config-free distributed execution,
UI for viewing complex runs and GitHub integration. Learn more at https://nx.app

Your workspace is currently unclaimed. Run details from unclaimed workspaces can be viewed on cloud.nx.app by anyone
with the link. Claim your workspace at the following link to restrict access.

https://cloud.nx.app/orgs/workspace-setup?accessToken=NDdhMmQ0MGQtMzE0My00NWJhLWI4N2UtZDA0NzhjMDNhODYzfHJlYWQtd3JpdGU=
```

```
> NX First time using Nx? Check out this interactive Nx tutorial.

https://nx.dev/react-tutorial/1-code-generation

arturs.jermolickis@AMAC02FTEJLMD6M ~ %
```

Structure Created:

```

  JSCONF
  .nx
  .vscode
  apps
  jsapp
  jsapp-e2e
  node_modules
  .editorconfig
  .eslintignore
  .eslintrc.json
  .gitignore
  .prettierignore
  .prettierrc
  jest.config.ts
  jest.preset.js
  nx.json
  package-lock.json
  package.json
  README.md
  tsconfig.base.json
```

Creating a new library and app

```
/Users/arturs.jermolickis/.zshrc:31: command not found: pass  
○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx generate @nrwl/react:lib shared-ui
```

CSS

SASS (.scss)	[http://sass-lang.com]
LESS	[http://lesscss.org]
styled-components	[https://styled-components.com]
emotion	[https://emotion.sh]
styled-jsx	[https://www.npmjs.com/package/styled-jsx]
None	

✓ Which stylesheet format would you like to use? · [css](#)
? What unit test runner should be used? ... ☐
[jest](#)
vitest
none

✓ Which stylesheet format would you like to use? · `css`

✓ What unit test runner should be used? · `jest`

? Which bundler would you like to use to build the library? Choose 'none' to skip build setup. ... 

`none`

`vite`

`rollup`

Creating a new app

```
○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx generate @nrwl/react:app newjsapp
```

```
○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx generate @nrwl/react:app newjsapp
```

```
> NX Generating @nrwl/react:application
```

```
? Which stylesheet format would you like to use? ...
```

CSS

SASS(.scss)	[http://sass-lang.com]
-------------	---	---	---

LESS	[http://lesscss.org]
------	---	---	---

styled-components	[https://styled-components.com]
-------------------	---	---	---

emotion	[https://emotion.sh]
---------	---	---	---

styled-jsx	[https://www.npmjs.com/package/styled-jsx]
------------	---	---	---

None

```
○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx generate @nrwl/react:app newjsapp
```

```
> NX Generating @nrwl/react:application
```

```
✓ Which stylesheet format would you like to use? · css  
✓ Would you like to add React Router to this application? (y/N) · true  
✓ Which E2E test runner would you like to use? · cypress  
? Which bundler do you want to use to build the application? ... ■  
vite  
webpack  
rspack
```

```
○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx generate @nrwl/react:app newjsapp
```

```
> NX Generating @nrwl/react:application
```

- ✓ Which stylesheet format would you like to use? · **css**
- ✓ Would you like to add React Router to this application? (y/N) · **true**
- ✓ Which E2E test runner would you like to use? · **cypress**
- ✓ Which bundler do you want to use to build the application? · **webpack**
- ? What should be the project name and where should it be generated? ...

```
> As provided:
```

```
  Name: newjsapp
```

```
  Root: newjsapp
```

```
Derived:
```

```
  Name: newjsapp
```

```
  Root: apps/newjsapp
```

Import SharedUi into newjsapp

TS app.tsx U X

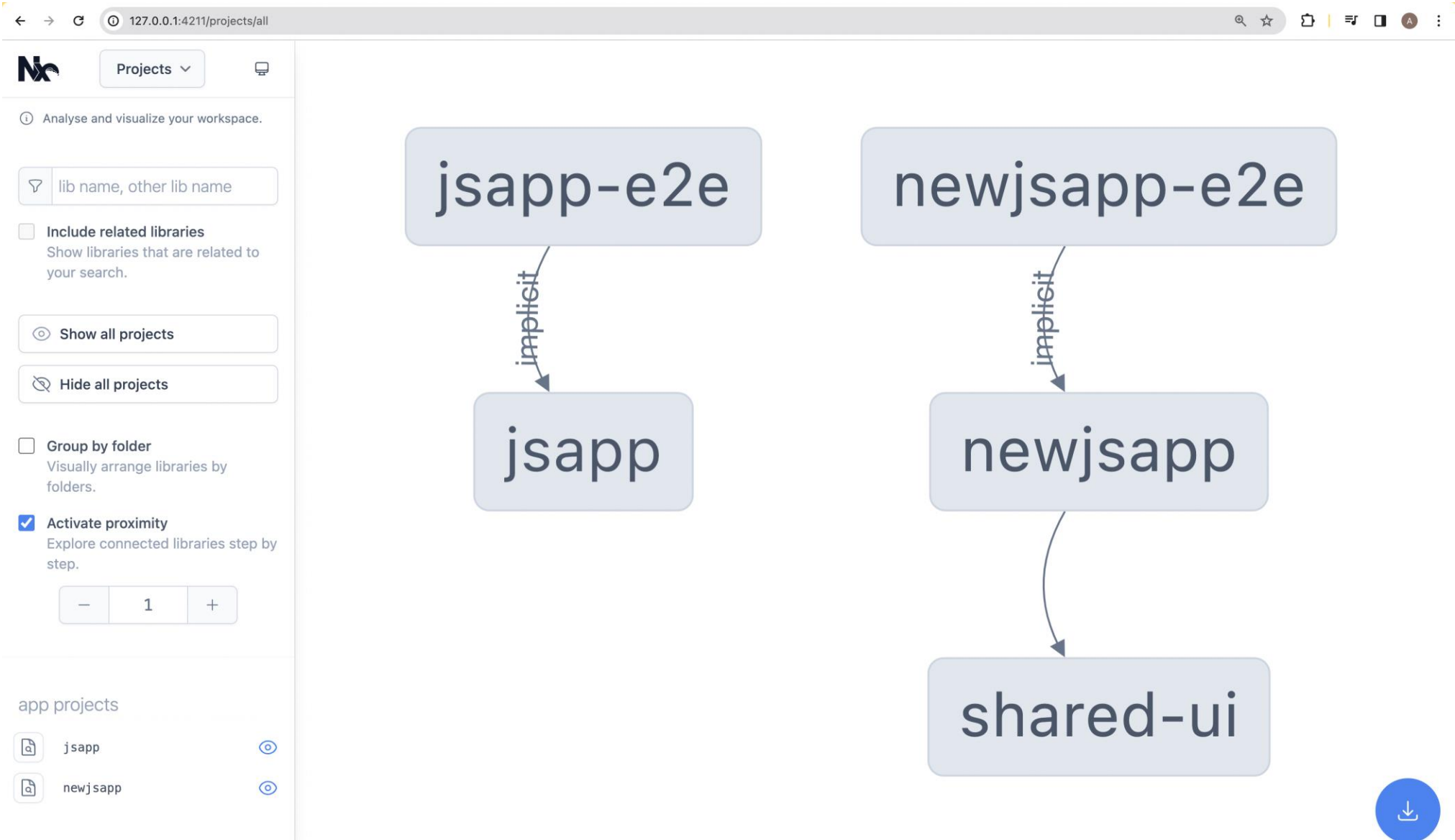
TS shared-ui.tsx U

newjsapp > src > app > TS app.tsx >  App

```
1 // eslint-disable-next-line @typescript-eslint/no-unused-vars
2 import styles from './app.module.css';
3
4 import NxWelcome from './nx-welcome';
5
6 import { Route, Routes, Link } from 'react-router-dom';
7
8 import { SharedUi } from '@jsconf/shared-ui';
9
```

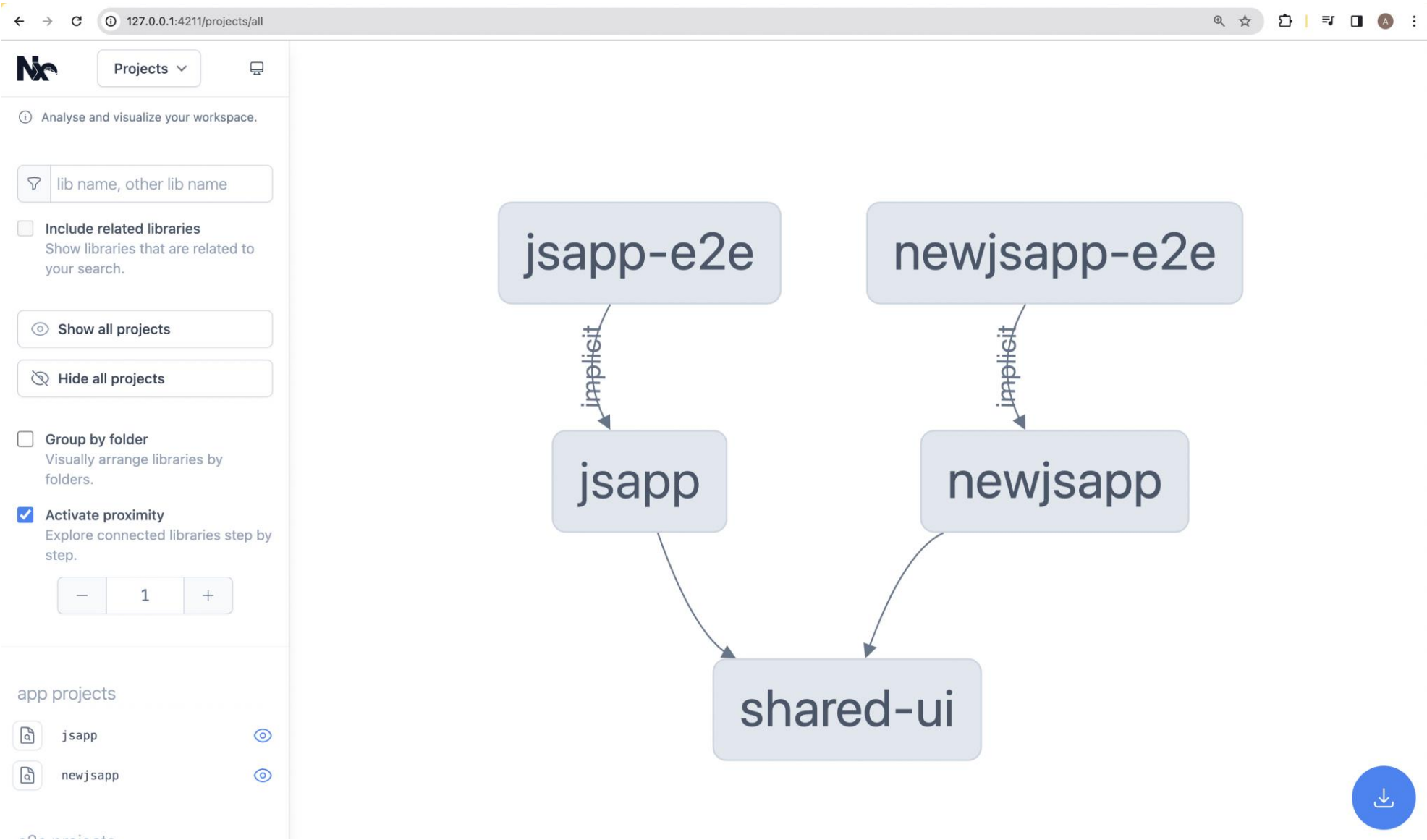
Running a graph to display the structure

```
○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx dep-graph
```



Importing SharedUi into a newjsapp

```
TS app.tsx newjsapp/... U TS app.tsx apps/... M X
apps > jsapp > src > app > TS app.tsx > ...
1 // eslint-disable-next-line @typescript-eslint/no-unused-vars
2 import styles from './app.module.css';
3
4 import NxWelcome from './nx-welcome';
5
6 import { SharedUi } from '@jsconf/shared-ui';
7
8 export function App() {
9   return (
10     <div>
11       <NxWelcome title="jsapp" />
12     </div>
13   );
14 }
15
16 export default App;
17
```



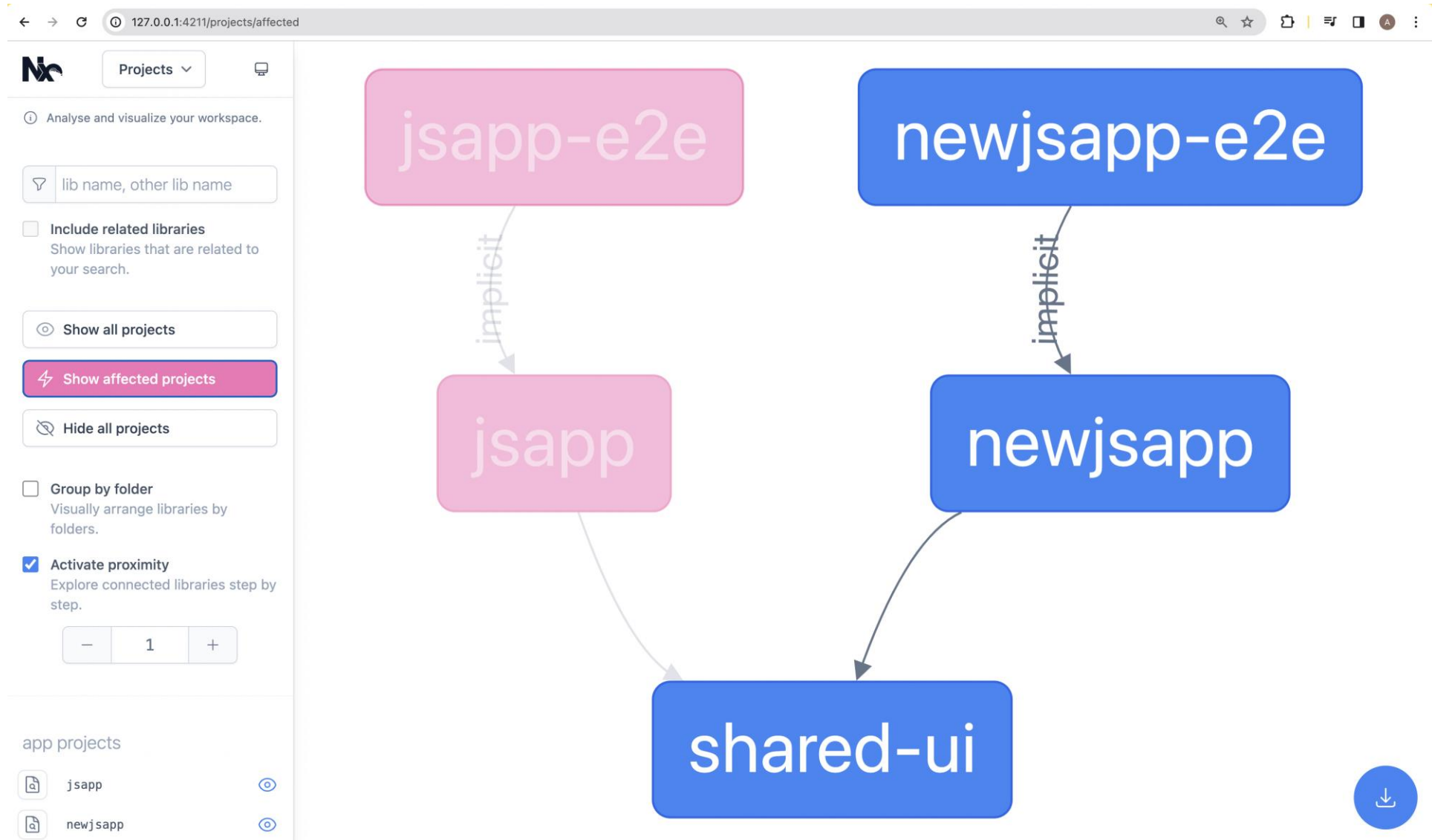
Adding some changes to newjsapp

```
newjsapp > src > app > TS app.tsx > ...
```

```
1  // eslint-disable-next-line @typescript-eslint/no-unused-vars
2  import styles from './app.module.css';
3
4  import NxWelcome from './nx-welcome';
5
6  import { Route, Routes, Link } from 'react-router-dom';
7
8  import { SharedUi } from '@jsconf/shared-ui';
9  console.log('hi');
```

Running graph on affected elements

```
arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx affected:dep-graph
```



Running test on affected

```
○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf % npx nx affected:test
```

> **NX** Running target **test** for 3 projects

> **NX** Running target **test** for 3 projects

> **NX** Running target **test** for 3 projects

> **NX** Running target **test** for 3 projects

→ Executing 3/3 remaining tasks in parallel...

⋮ nx run jsapp:test

⋮ nx run newjsapp:test

⋮ nx run shared-ui:test

> **NX** Running target **test** for 3 projects

- ✓ nx run shared-ui:test (22s)
- ✓ nx run jsapp:test (22s)
- ✓ nx run newjsapp:test (22s)

> **NX** Successfully ran target **test** for 3 projects (29s)

View logs and investigate cache misses at <https://nx.app/runs/9XJfqCnYcF>

○ arturs.jermolickis@AMAC02FTEJLMD6M jsconf %

Pros



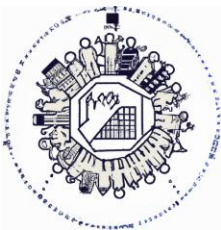
all the necessary dependencies and configuration files are stored in a single location



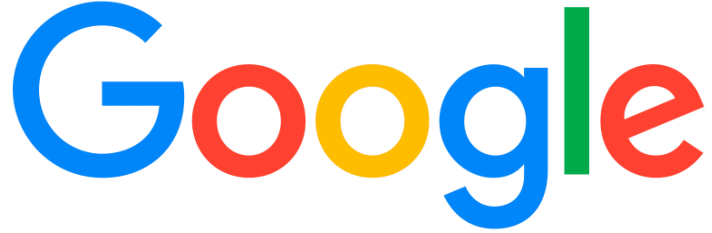
easier to automate the build, testing, and deployment processes



If the dependency is updated, all the dependency – related code is immediately visible and can be modified accordingly



Very useful for the large projects and large organisations.
High level overview is easier.



Monorepo



BigSur

Uber

Codebase



Monorail



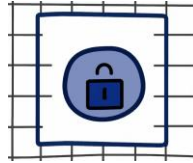
Microsoft

One Engineering System

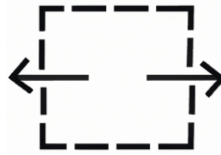
Cons



Size - Repositories of a Huge size, sometimes not possible to have them cloned



Security - Whole code is at one place, and often Online, there is always a Security risk, even if it's hugely small, therefore it requires complex access management



Flexibility – less flexible then poly-repo architecture.



Tooling – Some of the features, such as tests, will need a different tools, to be used with monorepos.



Licensing – Different packages might enforce different licenses



During **updates** the new tag is generated, and there is a possibility to have an “empty updates” if the setup is wrong.

Other approaches

- **Multi repo architecture** - independent services that communicate with each other through APIs.
 - Each service has its own codebase and repository
 - easier to manage and scale the system.
 - reduce complexity and improve modularity
 - requires more coordination and communication between teams.
- **Modular monorepo** - hybrid approach that combines monorepo and poly-repo architectures.
 - Smaller modules that have their own codebase and repository, but managed within a single monorepo.
 - Approach can help reduce complexity
 - Enabling code reuse and easy collaboration.
- **Component-based** architecture: reusable components.
 - Each component has its own codebase and repository
 - easily integrated into a larger system.
 - modularity and code reuse
 - require more work upfront to define and create the components.

There is no straight answer on which approach to use. Depending on requirements of the project and the way that project is going to be managed, and correct approach has to be chosen based on this decision.

Regarding migration from monorepo to multirepo or vice versa, it is generally easier to migrate from a monorepo to a multirepo than the other way around.

Thank you

you can find arturs.jermolickis@Accenture.com