

Monorepo 在字节跳动前端团队的实践

liyubei@bytedance.com

About me

2018年加入字节跳动，目前负责字节跳动前端基础工程团队主要的方向包括：前端工程化解决方案、Node.js Infra、跨平台解决方案、性能监控解决方案等等。



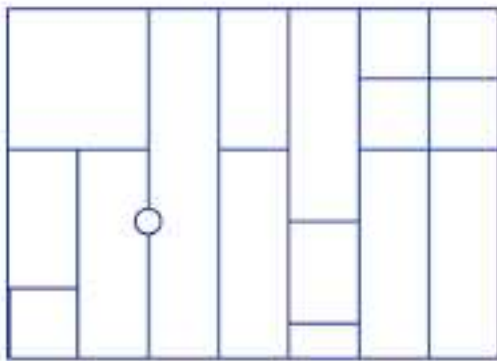
Outline

- What's monorepo
- Why monorepo
- How to use it
- Q & A

What's monorepo

The monorepo is **a software development strategy** where code for many projects is stored in the same repository. **It creates a single source of truth.**

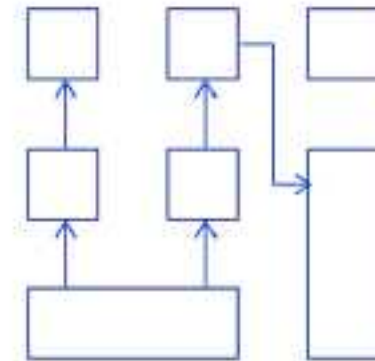
-- from wikipedia



Monorepo



Single-repo Monolith



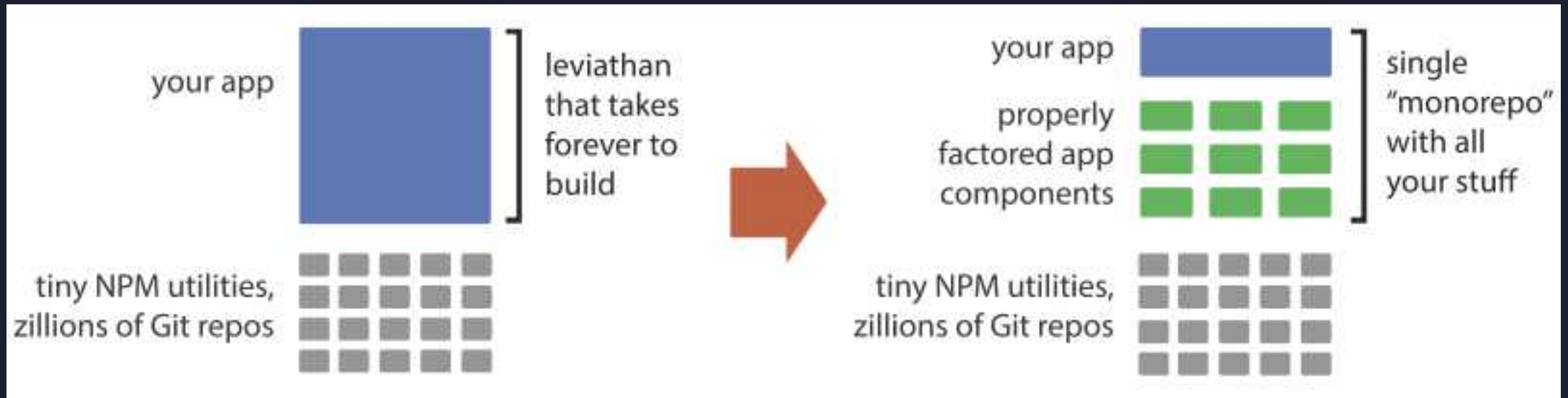
Multi-repo

Why monorepo?

Let us assume a possible situation...

-- *from rushjs.org*

Why monorepo?



-- from *rushjs.org*

Why monorepo?

Why did Google choose a monorepo and stick with it?
Because using a monorepo is **key to an open and collaborative culture**.

-- *Why Google Stores Billions of Lines of Code in a Single Repository*

Who is using monorepo?

BABEL



Google



Uber

Advantages

1. One place to store all configs and tests
2. Simplified package publishing
3. Re-use code with shared packages while still keeping them isolated
4. Simplified dependency management
5. Atomic commits & Speed
6. Large-scale code refactoring
7. Simplified organization & collaboration across teams
8. Security audit

Limitations

1. Loss of version information
2. Lack of per-project security
3. More storage needed
4. No way to restrict access only to some parts of the app
5. Poor Git performance when working on large-scale projects
6. Higher build time
7. ...

Outline

- What's monorepo
- Why monorepo
- How to use it
- Q & A

How to use it?

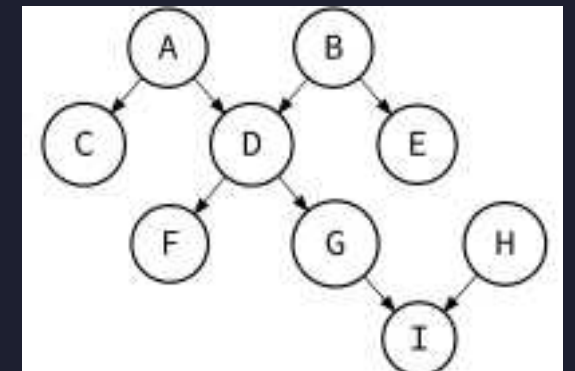
1. Initialization
2. Resolve the dependency issues
3. Provide better developer experience
4. Deployment

Initialization

- A simple package → `clone & yarn install & [postinstall]`

Initialization

- The monorepo
 - There are lots of packages
 - Packages have dependencies with each other
- Solution
 - **monorepo manager** + clone & yarn install



Initialization — monorepo manager



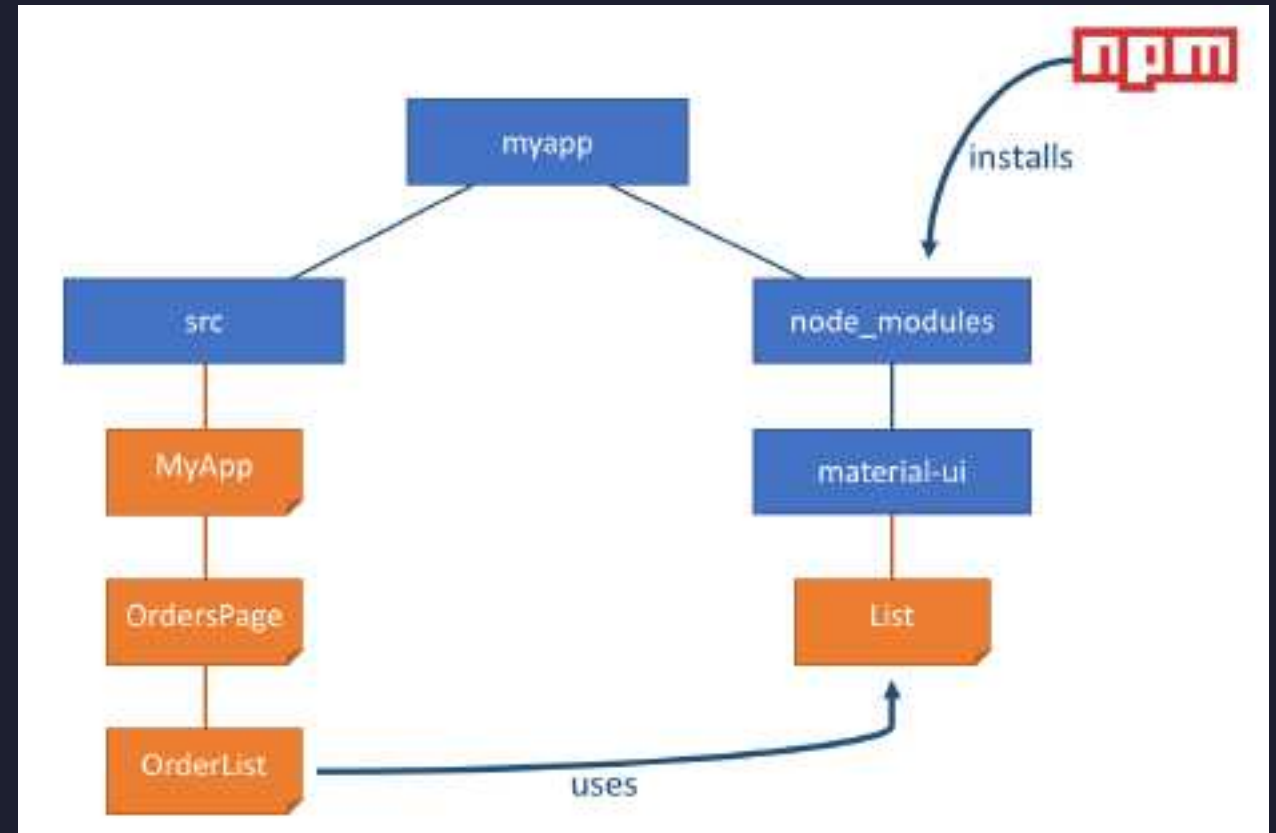
Initialization – yarn workspace

```
// package.json & yarn workspace
{
  ...
  "private": true,
  "workspaces": [
    "packages/*",
    "apps/*",
  ]
  ...
}
```



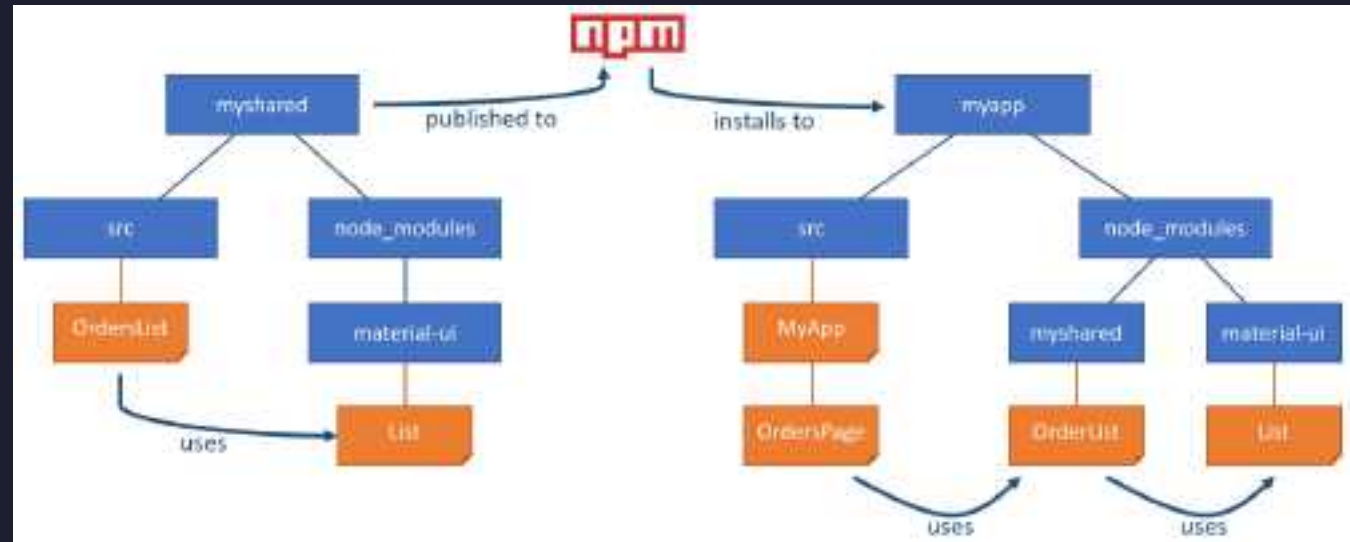
Initialization

```
// package.json
{
  "name": "@scope/MyApp",
  "version": "1.0.0",
  "dependencies": {
    "material-ui": "0.20.2"
  }
}
```



Initialization

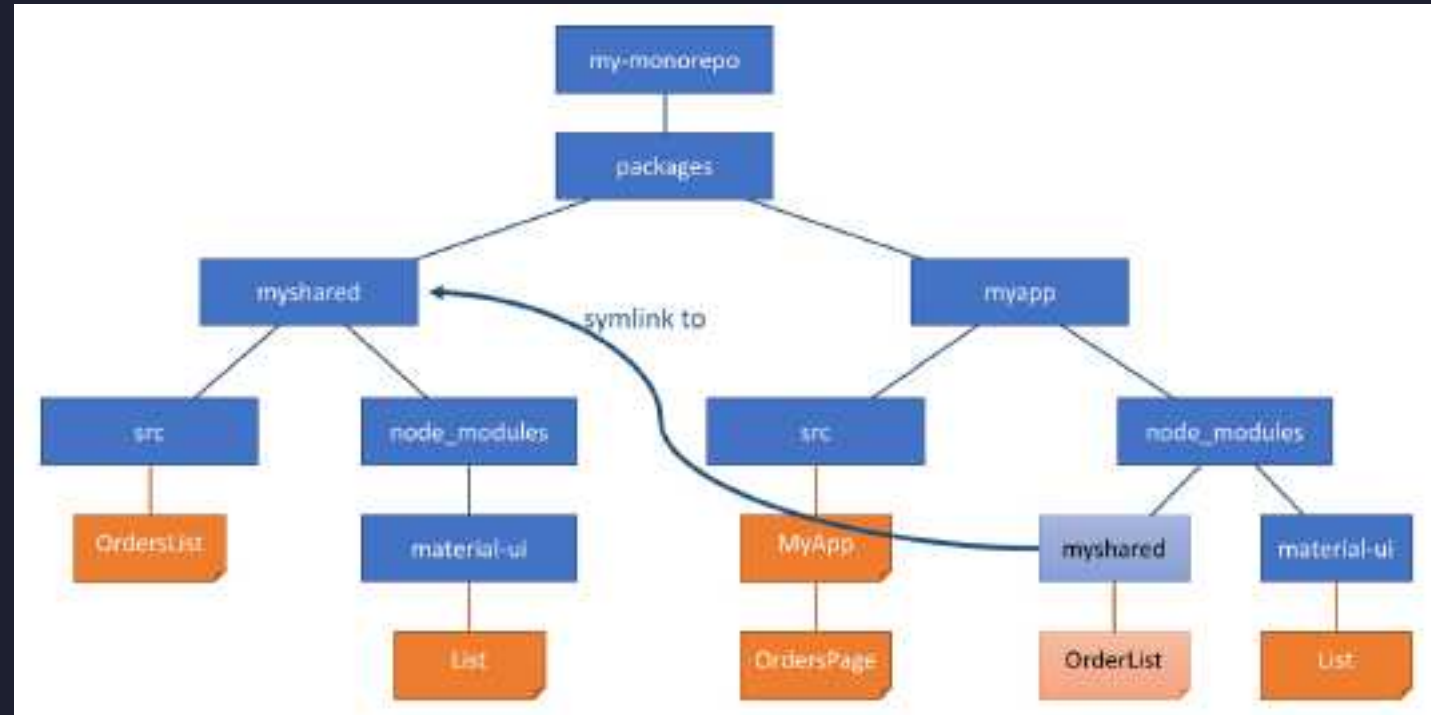
```
// package.json
{
  "name": "@scope/MyApp",
  "version": "1.0.0",
  "dependencies": {
    "material-ui": "0.20.2",
    "@scope/my-shared": "0.0.1"
  }
}
```



Initialization

```
// package.json
{
  "name": "@scope/MyApp",
  "version": "1.0.0",
  "dependencies": {
    "material-ui": "0.20.2",
    "@scope/my-shared": "0.0.1"
  }
}

// package.json & yarn workspace
{
  ...
  "private": true,
  "workspaces": [
    "packages/*"
  ]
  ...
}
```



Summary: yarn workspace

- Set “private”: true
- Add workspace packages
- Optionally you can add nohoist list of packages as well
- Run yarn at the root of monorepo directory

```
{  
  "name": "my-monorepo",  
  "version": "1.0.0",  
  "private": true,  
  "main": "index.js",  
  "workspaces": [  
    "packages/*"  
  ],  
  "nohoist": [  
    "packages/pkg-c",  
    "packages/pkg-g"  
  ],  
  "license": "MIT"  
}
```

But

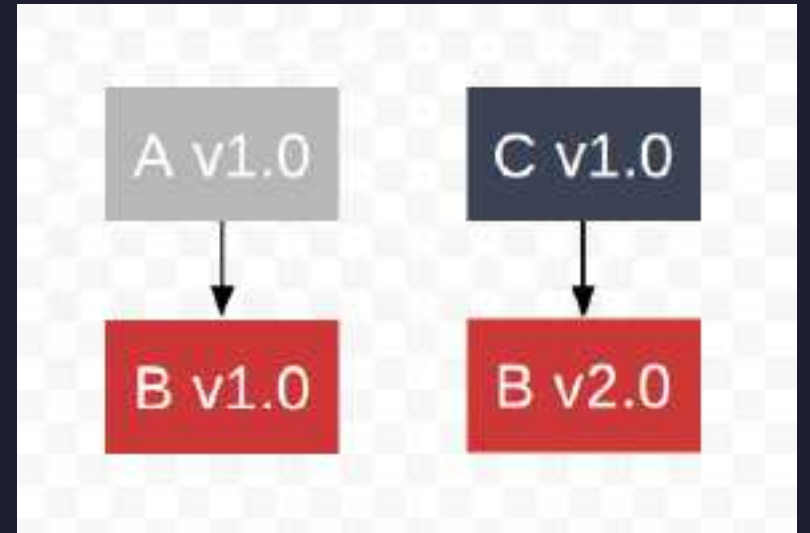
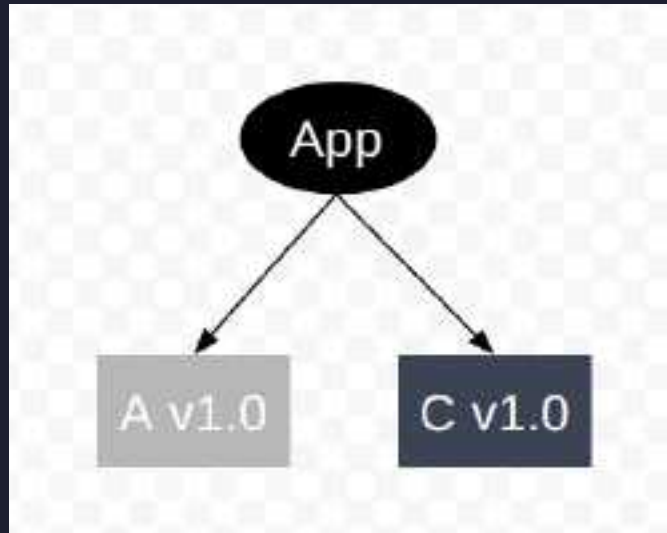
the real world is more complicated...

- Dependencies
- Performance
- Developer Experience
- Release & Deployment
-

-- we need to upgrade lots of toolchain and infrastructure

Dependencies

In the computer field, once it is scaled up, any small problems will be magnified.



Node.js resolve module strategy



Regret: node_modules

It massively complicates the module resolution algorithm.

vendor-by-default has good intentions, but in practice just using \$NODE_PATH wouldn't have precluded that.

Deviates greatly from browser semantics.

It's my fault and I'm very sorry.
Unfortunately it's impossible to undo now.

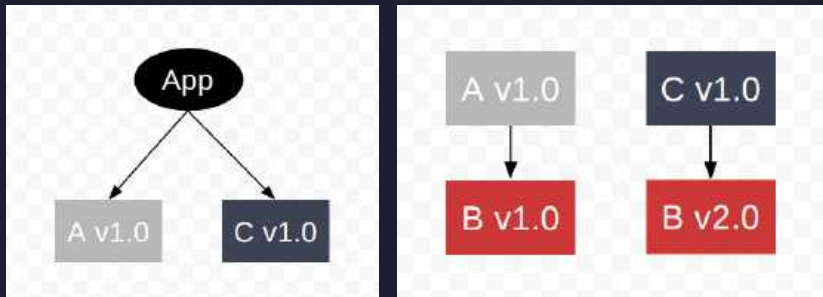
Node.js resolve module strategy

Recursively look up the package in `node_modules`. If `require('bar.js')` is called in the `'/home/ry/projects/foo.js'` file, Node.js will search in the following order:

- `/home/ry/projects/node_modules/bar.js`
- `/home/ry/node_modules/bar.js`
- `/home/node_modules/bar.js`
- `/node_modules/bar.js`

node_modules (nested mode)

A simple solution.



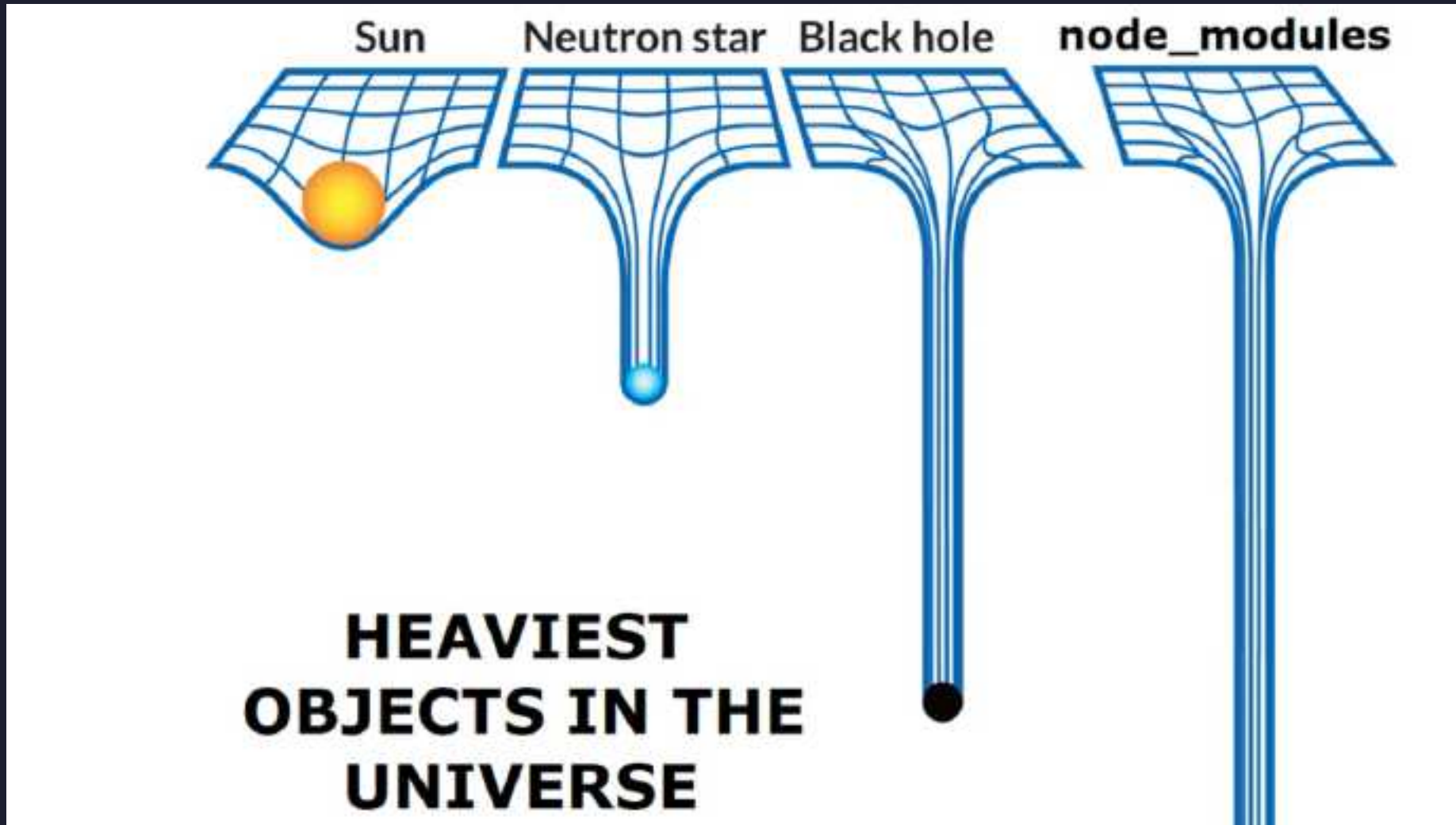
```
→ project tree .  
.  
├── node_modules  
│   ├── mod-a  
│   │   └── node_modules  
│   │       └── mod-b@1.0  
│   └── mod-c  
│       └── node_modules  
│           └── mod-b@2.0
```

But...

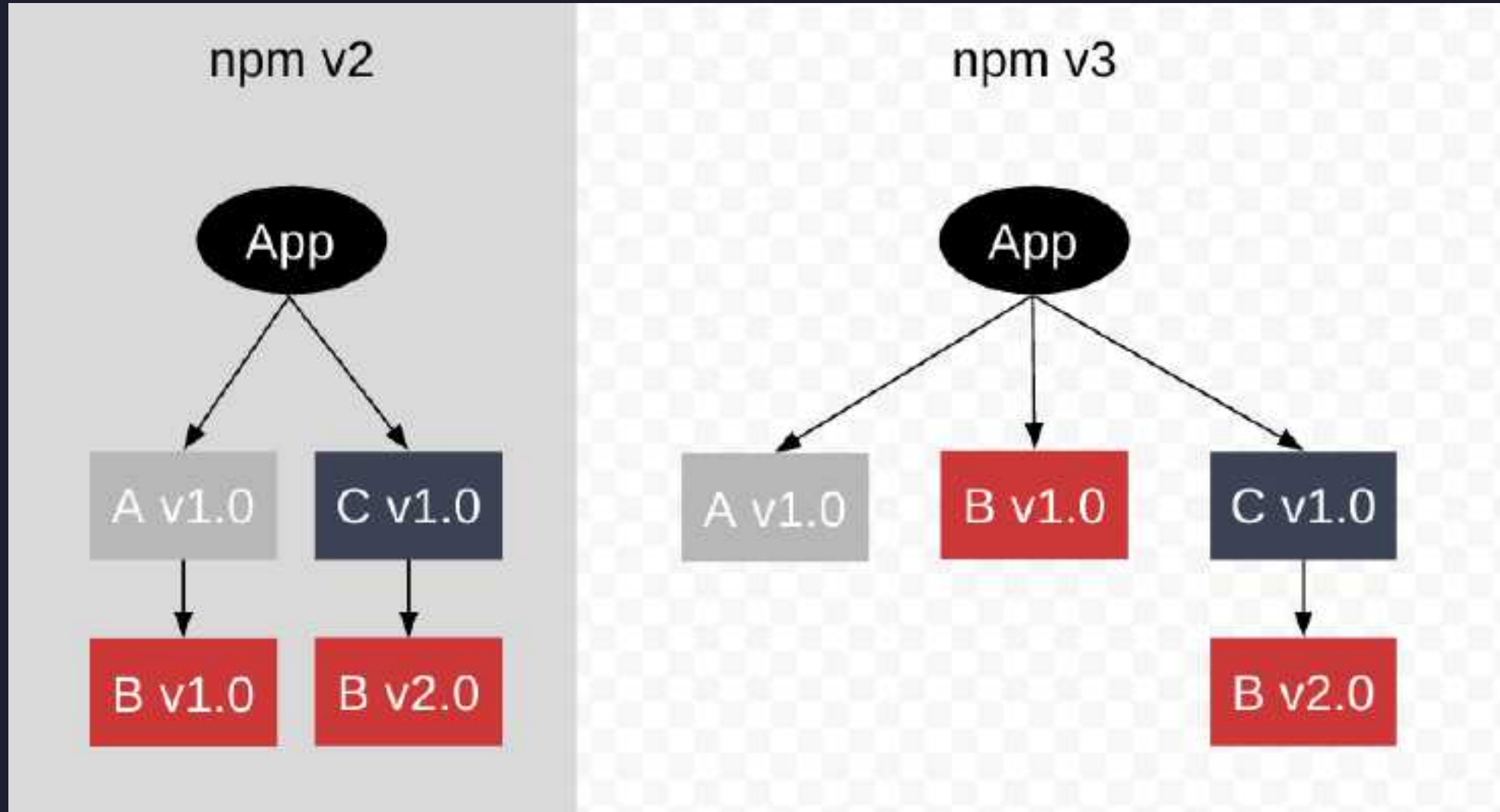
Lots of duplicated packages

```
→ project tree .  
.  
└─ node_modules  
    └─ mod-a  
        └─ node_modules  
            └─ mod-b@1.0  
    └─ mod-c  
        └─ node_modules  
            └─ mod-b@2.0  
    └─ mod-d  
        └─ node_modules  
            └─ mod-b@2.0
```

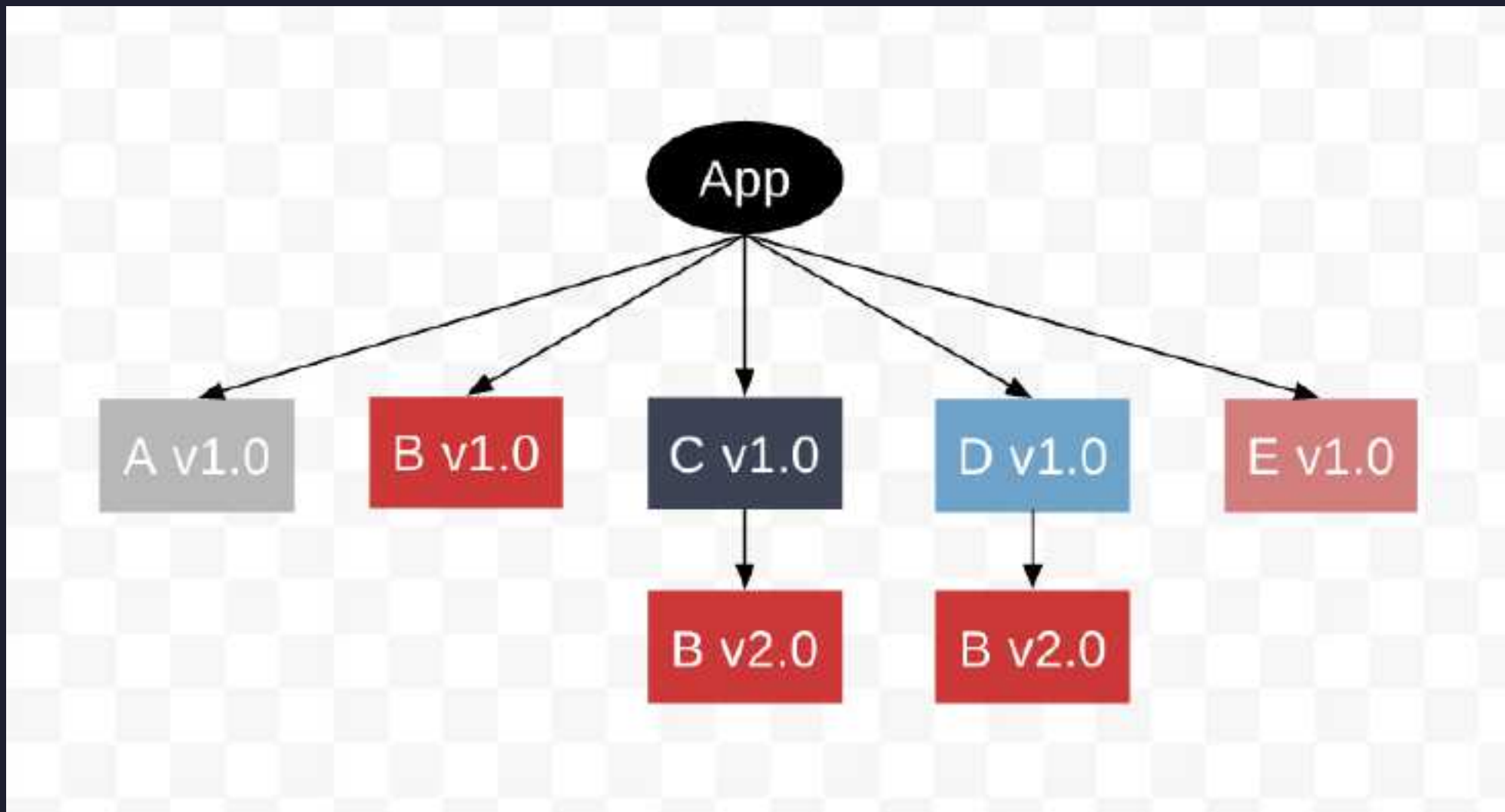
The dependency hell



node_modules (flat mode)



doppelgangers (多党派)



Phantom dependency

```
{  
  "name": "my-library",  
  "version": "1.0.0",  
  "main": "lib/index.js",  
  "dependencies": {  
    "minimatch": "^3.0.4"  
  },  
  "devDependencies": {  
    "rimraf": "^2.6.2"  
  }  
}
```

Phantom dependency

```
const minimatch = require('minimatch');  
const expand = require('brace-expansion'); // ???  
const glob = require('glob'); // ???|
```

monorepo: link & phantom dependency

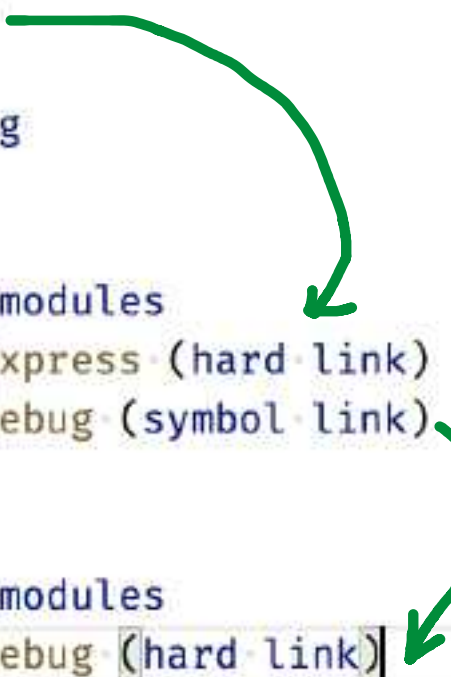
- Problem: third_party dependencies && link hell
- Solution (lerna && yarn && ...)
 - flat mode / hoist (default)
 - auto link to the top level <ROOT>/node_modules
 - auto link <PKG>/node_modules/.bin to <ROOT>/node_modules
- But phantom dependency will be more serious
- Better Solution (pnpm)

pnpm: explicit is better than implicit

```
// Default (nested mode)
myProject
|--node_modules
|  |--express
|  |  |--node_modules
|  |  |--debug
```

```
// Hoist (flat mode)
myProject
|--node_modules
|  |--express
|  |--debug
```

```
myProject
|--node_modules
|  |--express (symbol link)
|  |--.pnpm
|  |  |--registry.npmjs.org
|  |  |  |--express
|  |  |  |  |--4.0.0
|  |  |  |  |--node_modules
|  |  |  |  |  |--express (hard link)
|  |  |  |  |  |--debug (symbol link)
|  |  |  |--debug
|  |  |  |  |--1.0.0
|  |  |  |  |--node_modules
|  |  |  |  |  |--debug (hard link)
```

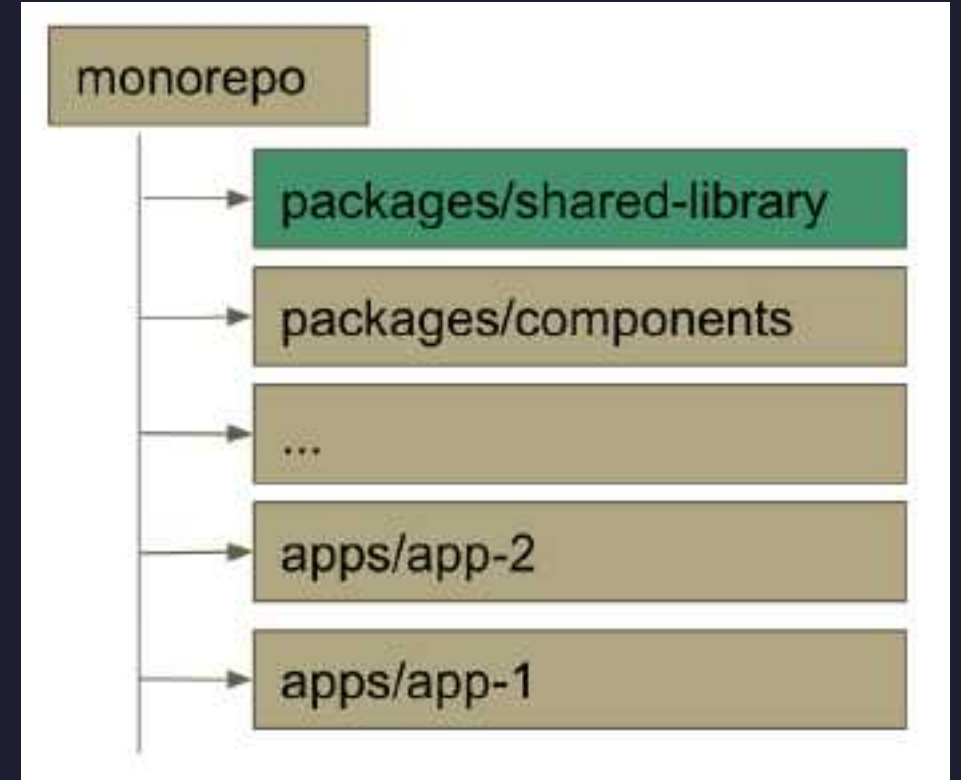


pnpm: explicit is better than implicit

```
+ jupiter-tools-monorepo git:(master) ✗ rm -rf node_modules && time yarn install 1>/dev/null 2>1 && du -d 0 -h node_modules
yarn install > /dev/null 2> 1 173.55s user 83.60s system 213% cpu 2:00.42 total
1.2G    node_modules
+ jupiter-tools-monorepo git:(master) ✗ rm -rf node_modules && time pnpm install 1>/dev/null 2>1 && du -d 0 -h node_modules
pnpm install > /dev/null 2> 1 20.23s user 26.39s system 203% cpu 22.935 total
359M    node_modules
+ jupiter-tools-monorepo git:(master) ✗
```

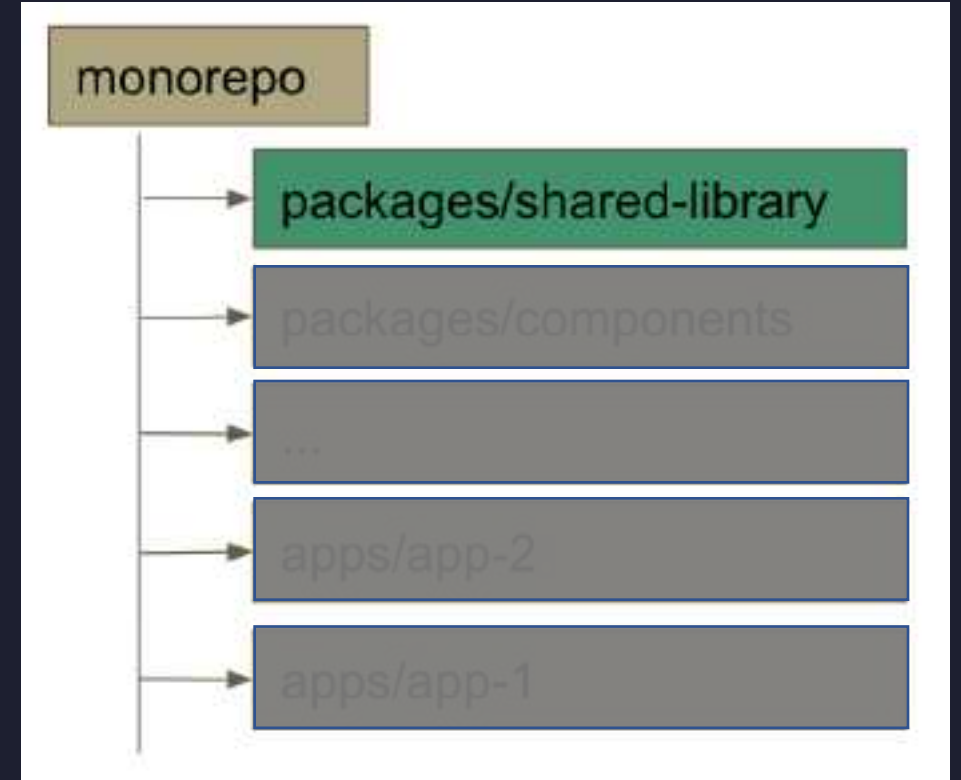
Tip: Improve the install speed

Auto remove unnecessary `package.json` and decrease the `node_module` size



Tip: Improve the install speed

Auto remove unnecessary `package.json` and decrease the `node_module` size



Performance

Webpack HMR took lots of time to rebuild the project as the project code size increase

包名	平均 rebuild 耗时	平均单 rebuild 次数
webpack-dev-server	8503.50ms	74
webpack	3631.90ms	147
webpack-cli	2208.97ms	163
webpack-merge	555.57ms	293
webpack-hot-middleware	2063.49ms	437
webpack-hot-dev-server	2219.68ms	166
webpack-hot-middleware-webpack-dev-server	2673.22ms	294
webpack-hot-middleware-webpack-dev-server	4102.48ms	400
webpack-hot-middleware-webpack-dev-server	442.75ms	164
webpack-hot-middleware-webpack-dev-server	1271.59ms	412
webpack-hot-middleware-webpack-dev-server	6847.76ms	125
webpack-hot-middleware-webpack-dev-server	4440.03ms	124
webpack-hot-middleware-webpack-dev-server	3784.64ms	15
webpack-hot-middleware-webpack-dev-server	2869.28ms	90
webpack-hot-middleware-webpack-dev-server	11285.34ms	100

Performance – Webpack

Bundled (ex: webpack)



Unbundled (ex: snowpack)



Dev Server Performance – eds

esm dev server (eds) aims to decrease the rebuild time 90% to 200ms

- No bundle
- H2 (Server Push) to decrease the http requests ...
- Filepath resolver
 - workspace / typescript / node_modules / customized
- Compiler pipeline

Dev Server Performance – eds

- Bare Module Support
- TypeScript Module Resolve
- ...

Bare Module

```
import React from 'react';
import { Spin } from 'antd';
import './index.less';
interface PropsType {
  title?: string;
}
export default function({ title }: PropsType) {
  return (
    <div className="loading-page">
      <Spin tip={title} />
    </div>
  );
}
```

```
import * as __EDS_HMR__ from '/__eds__/hmr.js';
import.meta.hot = __EDS_HMR__.createHotContext(import.meta.url);

/** React Refresh: Setup */
if (import.meta.hot) {
  var prevRefreshReg = window.$RefreshReg$;
  var prevRefreshSig = window.$RefreshSig$;
  window.$RefreshReg$ = (type, id) => {
    window.$RefreshRuntime$.register(type, `/_dist_/app/components/Loading/index.js` + ` ` + id);
  };
  window.$RefreshSig$ = window.$RefreshRuntime$.createSignatureFunctionForTransform;
}
import React from '/web_modules/react.js';
import { Spin } from '/web_modules/antd.js';
import './_dist_/app/components/Loading/index.less._proxy_.js';
export default function ({
  title
}) {
  return React.createElement("div", {
    className: "loading-page"
  }, React.createElement(Spin, {
    tip: title
  }));
}

/** React Refresh: Connect */
if (import.meta.hot) {
  window.$RefreshReg$ = prevRefreshReg;
  window.$RefreshSig$ = prevRefreshSig;
  import.meta.hot.accept(() => {
    window.$RefreshRuntime$.performReactRefresh();
  });
}
```

Bare Module

```
.loading-page {  
  width: 100%;  
  height: 100%;  
  text-align: center;  
  background-color: rgba(255, 0.6);  
  
  .ant-spin {  
    position: absolute;  
    top: 50%;  
  }  
}
```

```
import * as __EDS_HMR__ from '/__eds__/hmr.js';  
import.meta.hot = __EDS_HMR__.createHotContext(import.meta.url);  
  
import * as __EDS_HMR_API__ from '/__eds__/hmr.js';  
import.meta.hot = __EDS_HMR_API__.createHotContext(import.meta.url);  
import.meta.hot.accept();  
import.meta.hot.dispose(() => {  
  document.head.removeChild(styleEl);  
});  
const code = ".loading-page {\n  width: 100%;\n  height: 100%;\n  text-align: center;\n  background-color: rgba(255, 0.6);\n  .ant-spin {\n    position: absolute;\n    top: 50%;\n  }\n}";  
const styleEl = document.createElement("style");  
const codeEl = document.createTextNode(code);  
styleEl.type = 'text/css';  
  
styleEl.appendChild(codeEl);  
document.head.appendChild(styleEl);  
// Exports  
var __CSS_LOADER_EXPORT__ = {};  
export default __CSS_LOADER_EXPORT__;
```

TypeScript Module Resolve

```
// @scope/mod-a tsconfig.json
{
  "baseUrl": "./src",
  "paths": {
    "@common/*": [ "../common/*" ],
    "@utils/*": [ "../utils/*" ]
  }
}

// @scope/mod-a/src/index.ts
import { add } from '@utils/helper';
```



```
// @scope/mod-b tsconfig.json
{
  "baseUrl": "./src",
  "paths": {
    "@common/*": [ "../common/*" ],
    "@utils/*": [ "../utils/*" ]
  }
}

// @scope/mod-b/src/index.ts
import { xxxx } from '@scope/mod-a/src/index';
```

Dev Server Performance – eds

- Dev Server startup time: 150s → 1s
- HMR rebuild time: 5s ~ 6s → 100ms

Version bumping

- Out of dated version or security concerns
 - Version conflicts
 -
-
- The open source community tools: `npm-check-updates` / `npm-check`

Version bumping – taze

```
+ taze usage -r
```

@jupiter/ui-tools	8 in use / 7 versions	*, ^2.0.0, ^2.1.1, ^2.1.1-alpha.2, ^2.1.6, ^2.14.0, ^2.9.8 +	2.15.14
react	30 in use / 6 versions	^16.8.6, ^16.12.0, ^16.13.1, ^16.5.2, ^16.8.1, ^16.8.6 +	16.13.1
@jupiter-block/runtime	7 in use / 6 versions	^1.0.0, ^1.0.11, ^1.1.0, ^1.2.1, ^1.2.24, ^1.3.12 +	1.4.7
@bytedesign/web-react	20 in use / 5 versions	^1.10.1, ^1.13.0-beta.4, ^1.15.2, ^1.17.2, ^1.17.3 +	1.21.5
@jupiter/plugin-runtime	8 in use / 5 versions	^2.0.0, ^3, ^3.0.0, ^3.10.6, ^3.4.1 +	3.11.2
react-dom	27 in use / 5 versions	^16.12.0, ^16.13.1, ^16.5.2, ^16.8.1, ^16.8.6 +	16.13.1
@types/node	4 in use / 4 versions	^12.7.5, ^13.7.6, ^9.3.0, ^9.4.6 +	14.8.26
core-js	12 in use / 4 versions	^3.0.0, ^3.0.4, ^3.0.5, @npmcore-js@3.0.0 +	3.6.5
@babel/core	5 in use / 4 versions	^7.4.5, ^7.5.5, ^7.7.5, ^7.8.4 +	7.10.5
codesandbox-api	3 in use / 3 versions	^0.0.19, ^0.0.20, ^0.0.24 +	0.0.24
styled-components	12 in use / 3 versions	^4.2.0, ^4.3.2, ^5.0.1 +	5.1.1
@jupiter/plugin-css	3 in use / 3 versions	^1.0.0, ^1.0.13, ^1.2.1 +	1.2.7
typescript	10 in use / 3 versions	^2.6.2, ^3.7.4, ^3.8.3 +	3.9.7
@storybook/react	3 in use / 3 versions	^3.1.14, ^3.2.8, ^3.2.13 +	5.3.19
@types/jest	9 in use / 3 versions	^22.0.0, ^24.0.10, ^25.2.3 +	26.0.7
ts-jest	9 in use / 3 versions	^22.0.0, ^24.0.2, ^25.0.0 +	26.1.3
qs	4 in use / 3 versions	^6.7.0, ^6.9.3, ^6.9.4 +	6.9.4
@babel/preset-env	5 in use / 3 versions	^7.4.5, ^7.5.5, ^7.9.0 +	7.10.4
@byted/tsdx	7 in use / 3 versions	0.11.0-beta.11, ^0.11.0-beta.20, ^0.11.0-beta.21 +	0.11.0-beta.21
@babel/parser	3 in use / 3 versions	^7.10.5, ^7.7.5, ^7.9.4 +	7.10.5
@redux/core	5 in use / 2 versions	^4.0.0, ^4.1.12 +	4.3.1
@redux/plugin-immr	5 in use / 2 versions	^3.0.0, ^4.1.1 +	4.1.9
classnames	6 in use / 2 versions	^2.3.5, ^2.2.6 +	2.2.6
dayjs	2 in use / 2 versions	1.8.29, ^1.8.23 +	1.8.30
react-codemirror2	2 in use / 2 versions	^4.0.1, ^7.2.1 +	7.2.1
react-router-dom	4 in use / 2 versions	^5.0.1, ^5.1.2 +	5.2.0
@jupiter/app-tools	8 in use / 2 versions	^3.0.0, ^3.0.1 +	3.15.1
@types/react	2 in use / 2 versions	^16.7.18, ^16.9.13 +	16.9.43

Version bumping – taze

```
* taze usage -r -a
```

```
@jupiter/ui-tools • 7 versions • latest: 2.15.14
```

```
^2.1.1-alpha.2 fes-tools-example
  *
    ^2.1.1 @jupiter-block/ui-tools-plugin-demo
    ^2.1.6 @jupiter-block/ui-tools-plugin-demo
    ^2.9.8 @fes/layout
    ^2.14.0 @fes/editor-sdk
    ^2.0.0 @byted-blocks/simple-comp-demo-for-upload-test
```

```
react • 6 versions • latest: 16.13.1
```

```
^16.12.0 butter, fes-app, block-demo, block-panel-demo, drag-layout-example, editor-demo,
  editor-sdk-demo, fes-tools-example, runtime2-block,
  @jupiter-block/ui-tools-plugin-demo, @jupiter-block/ui-tools-example-simple,
  @jupiter-block/ui-tools-example-simple, @jupiter-block/runtime,
  @jupiter-block/runtime, @jupiter-source/runtime, @fes/editor-sdk,
  @fes/editor-types, @fes/geo-service, @fes/block-card, @fes/block-card,
  @bytedesign-blocks/tabs, @bytedesign-blocks/tabs,
  @byted-blocks/simple-comp-demo-for-upload-test
^16.8.1 sandpack-example
^16.13.1 simple-block, @jupiter/ui-tools
^16.5.2 react-smooshpack, react-smooshpack
16.8.6 @jupiter-block/core
^16.8.6 @fes/editor
```

```
@jupiter-block/runtime • 6 versions • latest: 1.4.7
```

```
^1.0.11 fes-tools-example
^1.0.0 simple-block
```

Release & Deployment

- Hoist vs Vendor
- The challenge: how to keep a clean node_modules (*Workspaces are awesome, but they're inconvenient for deployment.*)
 - The yarnpkg issue opened since 2017: Feature Request: mechanism to "vendor" a workspace module's node_modules
#4521

Release & Deployment

- Create a bundle and reduce the runtime dependency.
- Analyze dependencies and copyTo package/node_modules

-- Tip: Carefully handle the phantom dependencies

Summing it up...

We upgrade the toolchain to improve the:

- Resolve the dependency hell
- Dev server startup performance & HMR reload performance (ESM)
- Add vendor support the specified packages
- Tweak the build time via decrease the dependencies
- Support the version bumping
- And more....

THANKS

QCon⁺ 案例研习社