

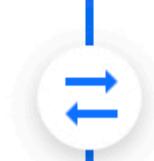
Семирульник Валентин

@7rulnik



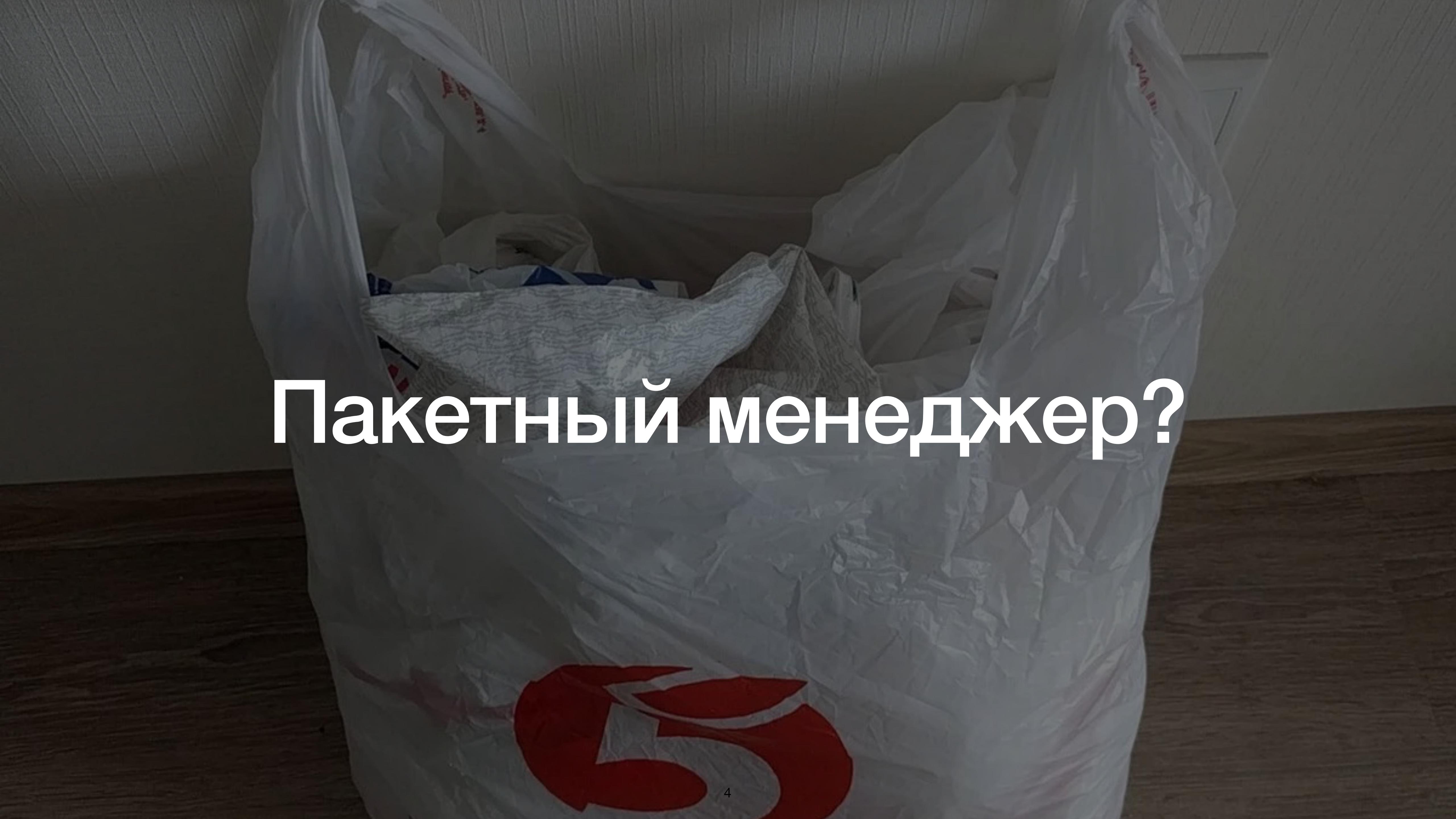
Поиск дешёвых авиабилетов

Лучший способ купить авиабилеты дёшево

[Авиабилеты](#)[Отели](#)[Для бизнеса ↗](#)[Откуда](#)[Куда](#)[Когда](#)[Обратно](#)

ПАССАЖИРЫ И КЛАСС

1 пассажир
эконом[Найти билеты](#)[Открыть Booking.com в новом окне](#)[Составить сложный маршрут](#)

A large, crumpled white plastic bag is shown from a top-down perspective. Inside the bag, a red graduation cap (mortarboard) is visible, resting on a blue and white striped fabric, likely a gown. The bag is tied at the top with a red ribbon.

Пакетный менеджер?

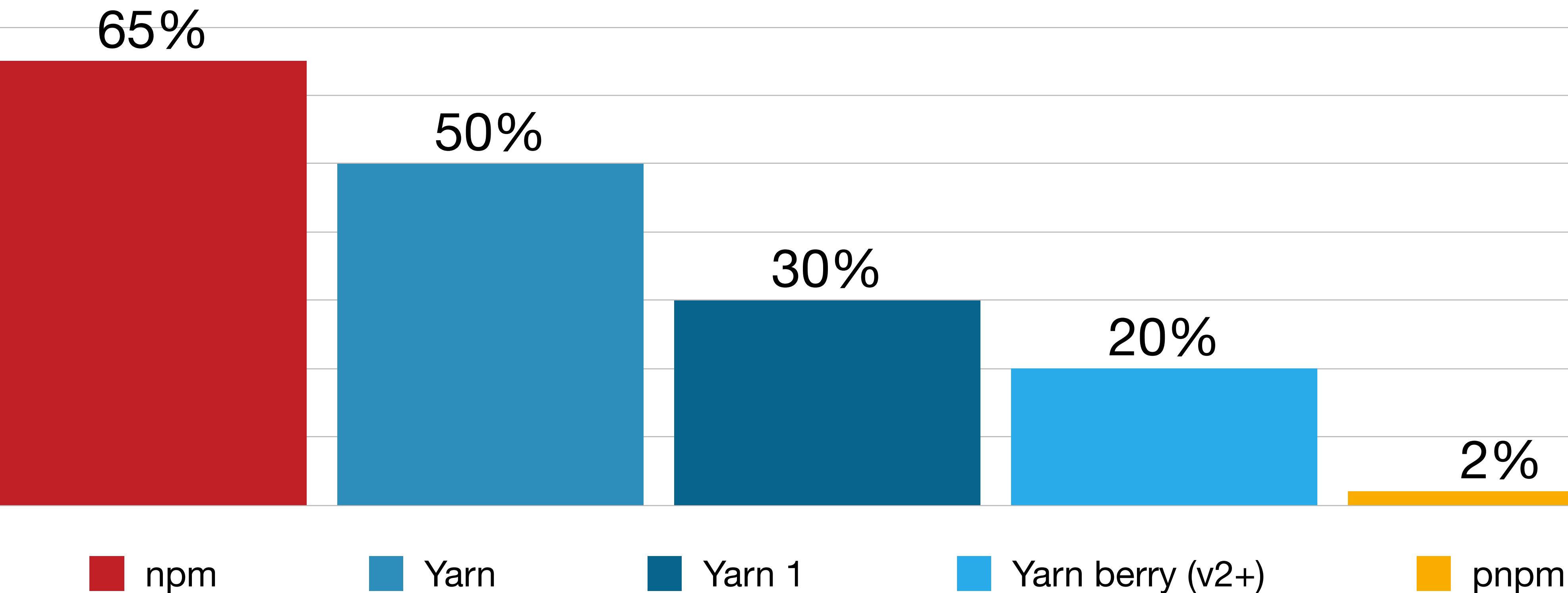


yarn

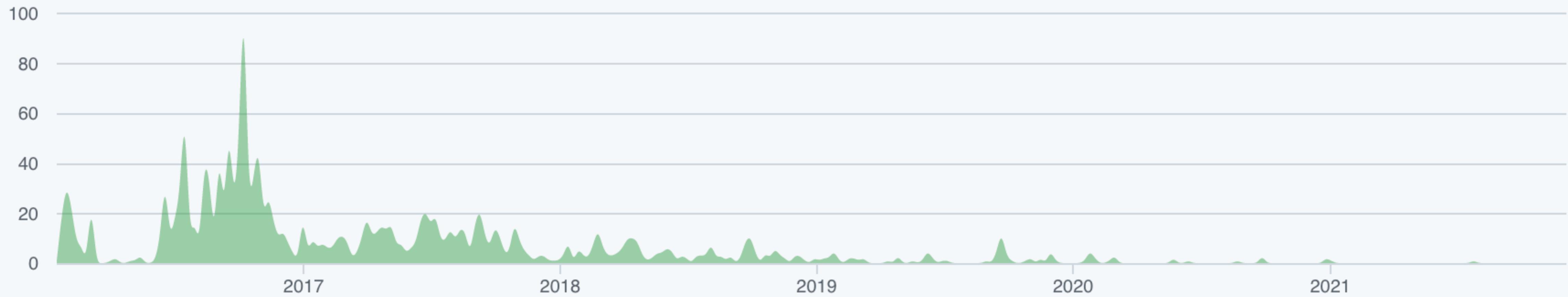
npm



Какой пакетный менеджер используют на работе?



Yarn 1 is dead



**Пакетные менеджеры
развиваются медленно**

**А мы ничего нового от них
и не ждём**

21 ноября 2011
npm включен в Node.js

github.com/nodejs/node/commit/b488be12

npm

!=



**Конкуренция – двигатель
прогресса**

Лучшее – враг хорошего

**Действителъ ли прт –
хороший пакетный менеджер?**

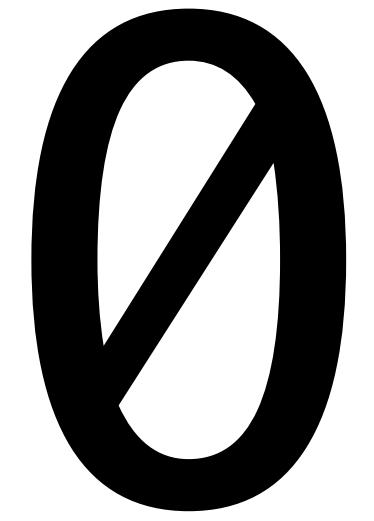
**npm – плохой пакетный
менеджер**

Есть ли жизнь за пределами прт?

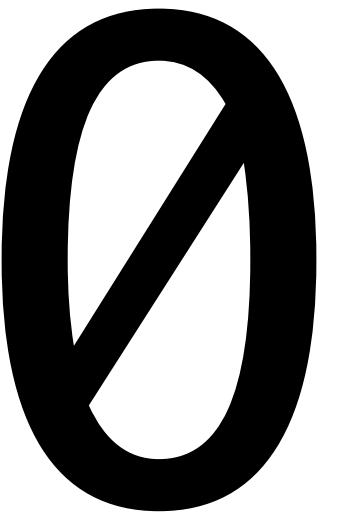
Критерии

1. Быстродействие
2. Структура node_modules
3. Модель управления разработкой
4. Поддержка монорепозиториев
5. Безопасность
6. Расширяемость

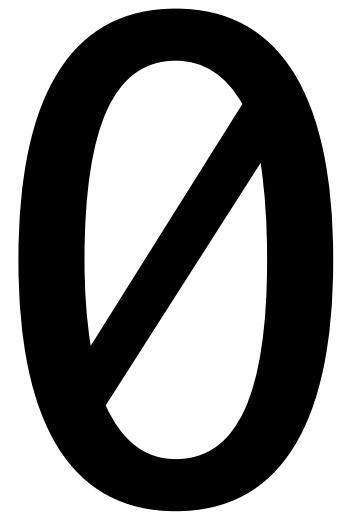
Ни слова про Yarn 1



npm



pnpm



Yarn

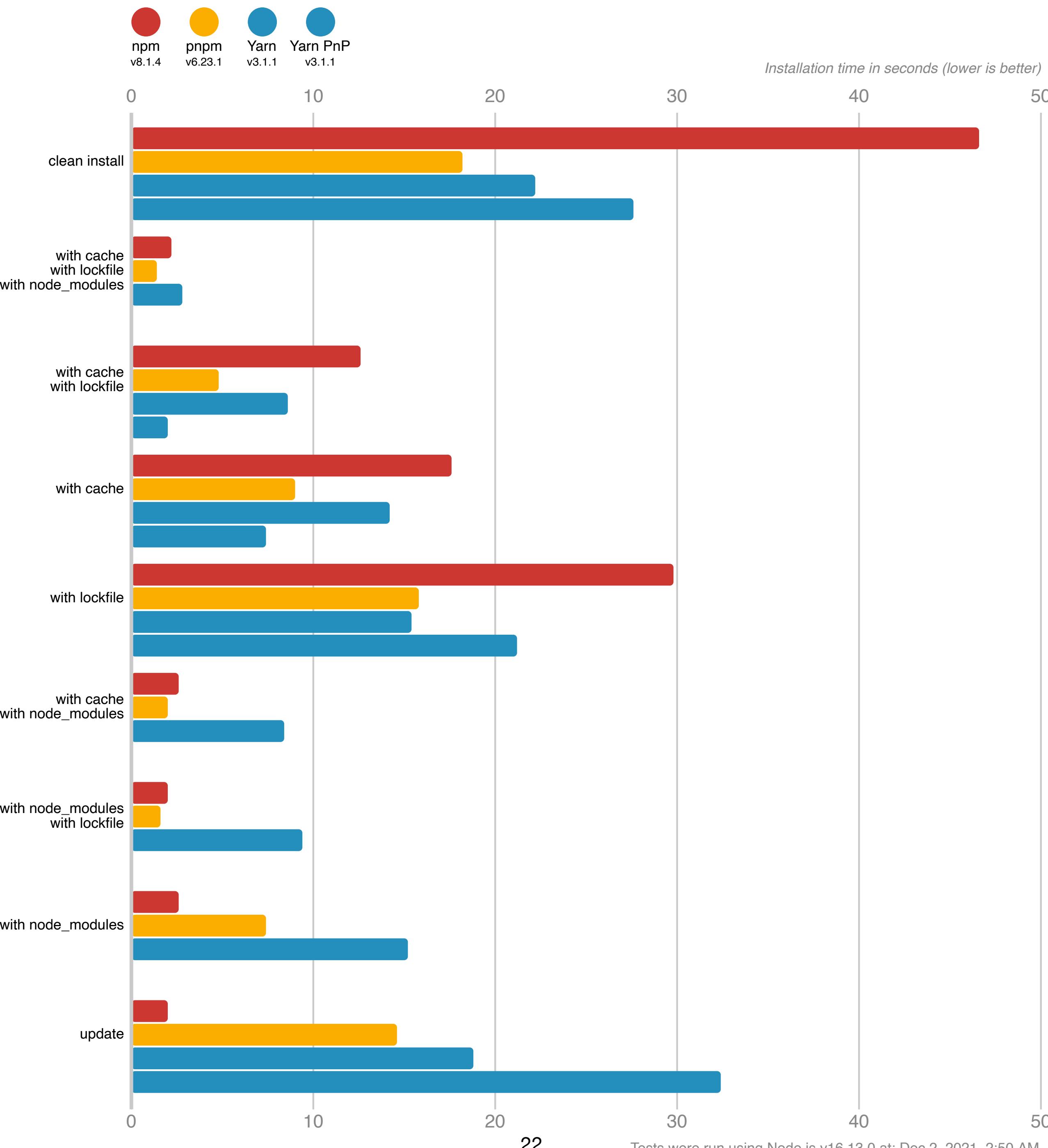


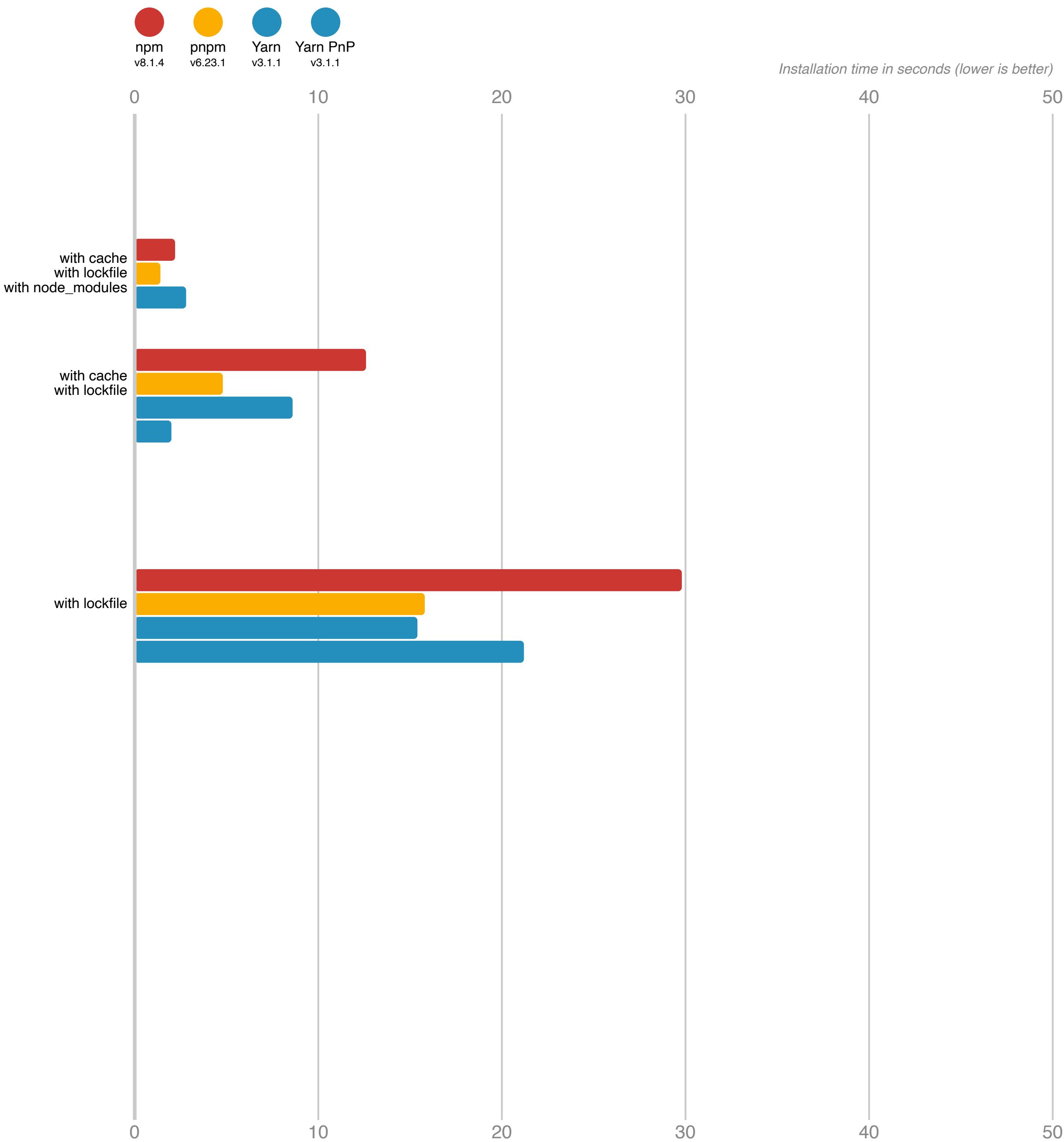
1. Самый быстрый

pnpm.io/benchmarks

yarnpkg.com/benchmarks

github.com/npm/benchmarks





медленнее != медленно

Быстродействие

1. Lockfile – **pnpm** и **yarn**
2. Cache и lockfile – **pnpm**
3. Cache, lockfile и node_modules – **все равны**

1

npm

3

pnpm

2

Yarn

A large outdoor storage area filled with numerous cardboard boxes and packages, all heavily covered in a thick layer of white snow. The items are stacked in various piles of different heights, some reaching up to 10 feet. The scene is set against a backdrop of industrial buildings and power lines under a clear sky.

2. Packages tree layout

Как Node.js резолвит модули?

В файле `/home/7rulnik/projects/talk/index.js` вызовем `require('react')`

1. `./home/7rulnik/projects/talk/node_modules/react`
2. `./home/7rulnik/projects/node_modules/react`
3. `./home/7rulnik/node_modules/react`
4. `./home/node_modules/react`
5. `/node_modules/react`

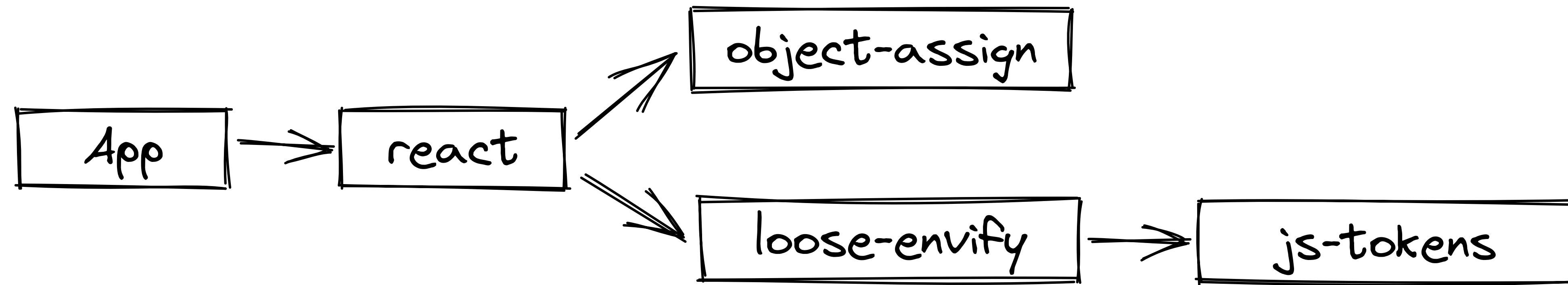
nodejs.org/dist/latest/docs/api/modules.html#all-together

Как было раньше?



npm v2

npm v1



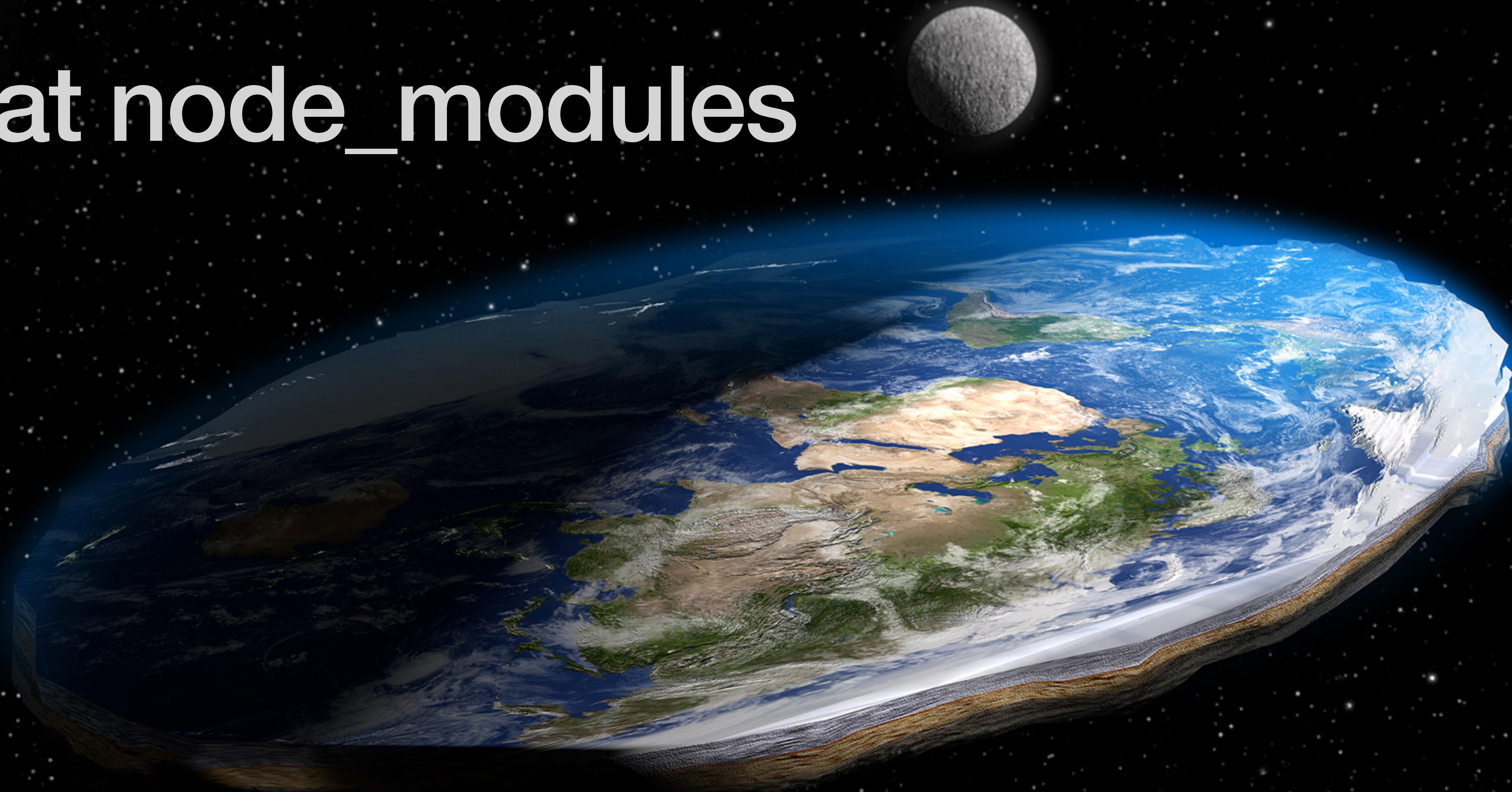
```
node_modules
└ react
    └ node_modules
        ├ loose-envify
        └ node_modules
            └ js-tokens
        └ object-assign
```

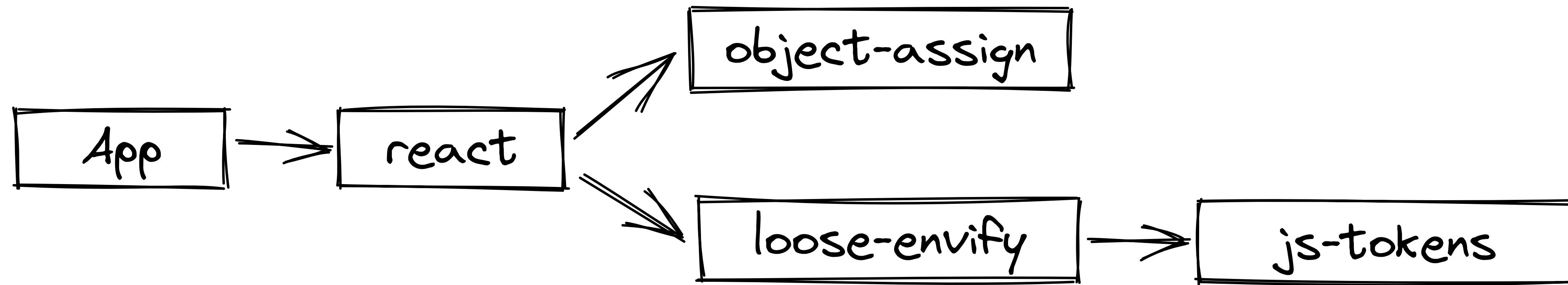

node_modules/jest/node_modules/@jest/
core/node_modules/@jest/transform/
node_modules/@babel/core/node_modules/
@babel/helper-module-transforms/
node_modules/@babel/helper-replace-
supers/node_modules/@babel/helper-
member-expression-to-functions/
node_modules/@babel/types/package.json

260

docs.microsoft.com/en-us/windows/win32/fileio/maximum-file-path-limitation

Flat node_modules

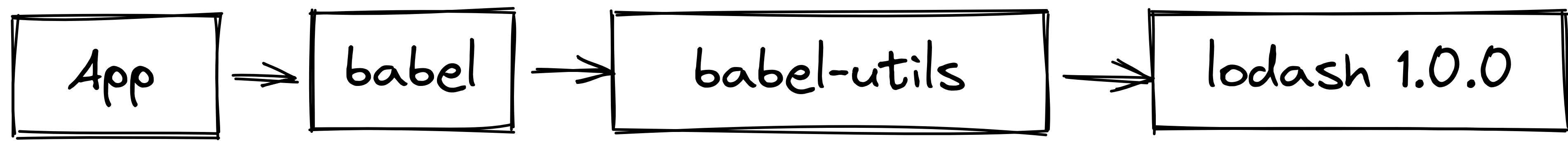




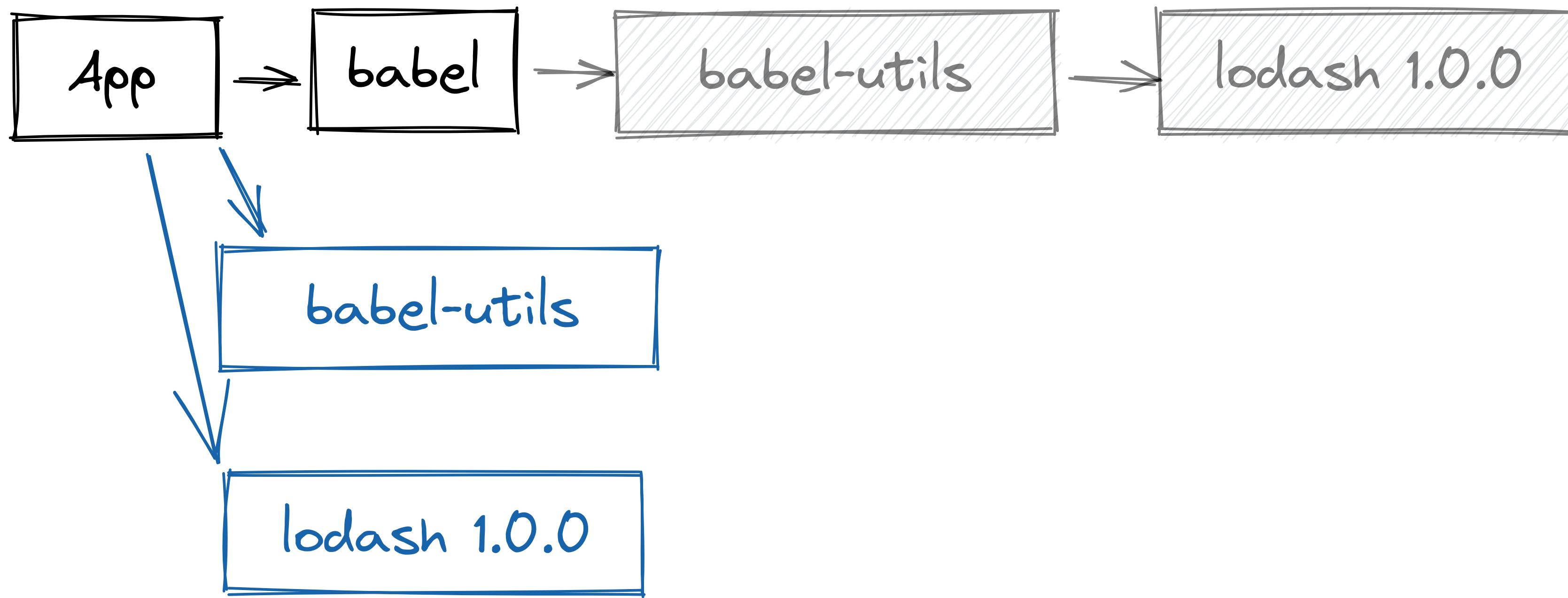
```
node_modules
  └── js-tokens
  └── loose-envify
  └── object-assign
      └── react
```

hoisting

[həʊstɪŋ]

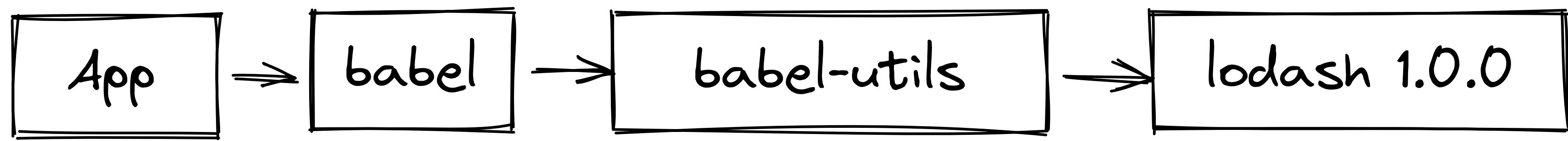


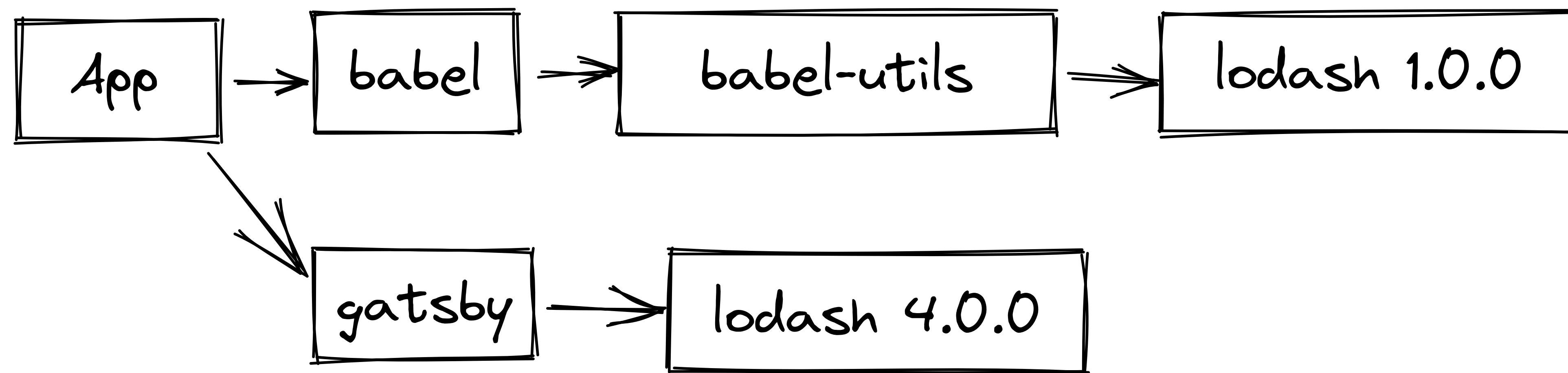
`flatten(tree)`

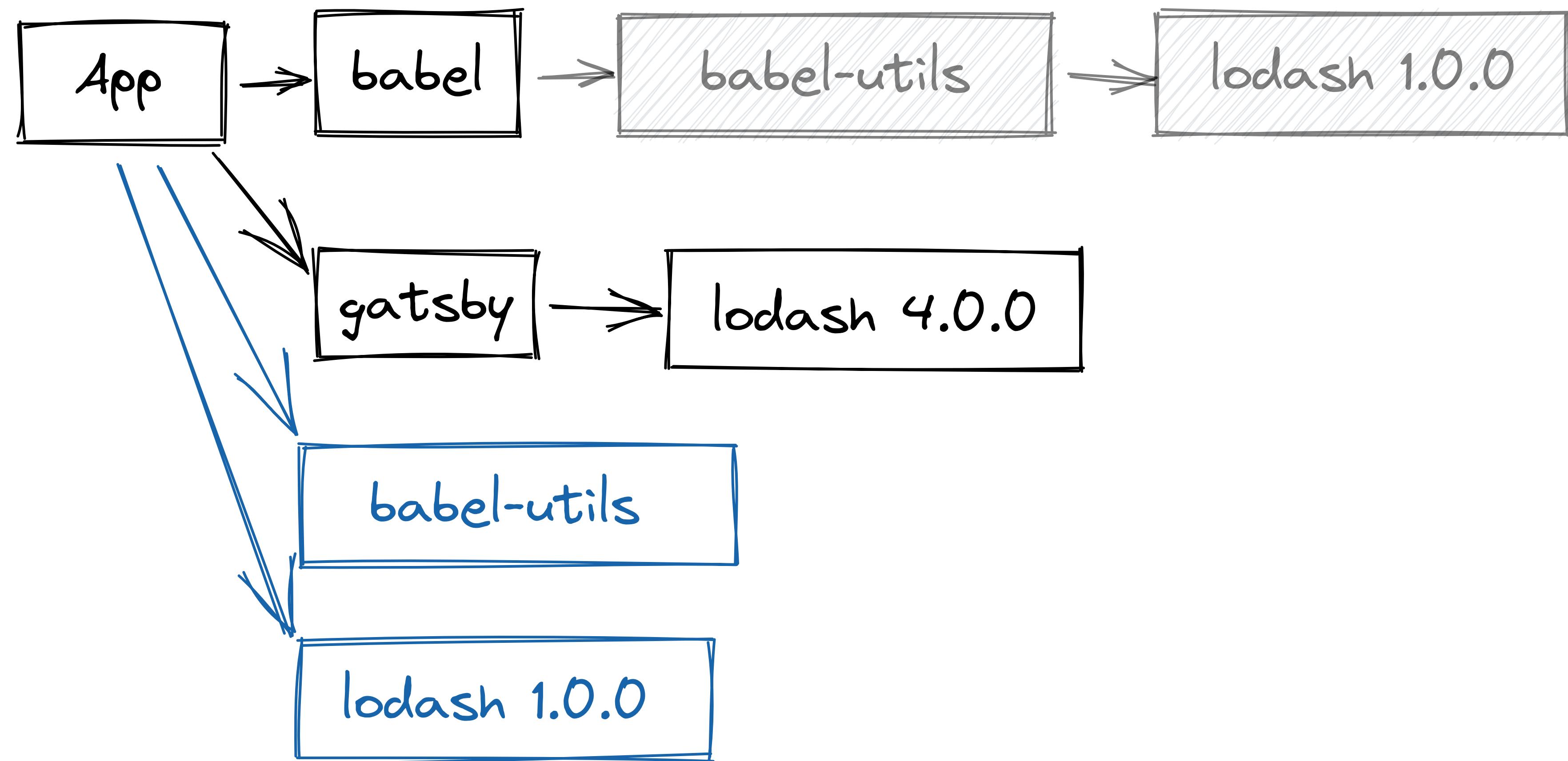


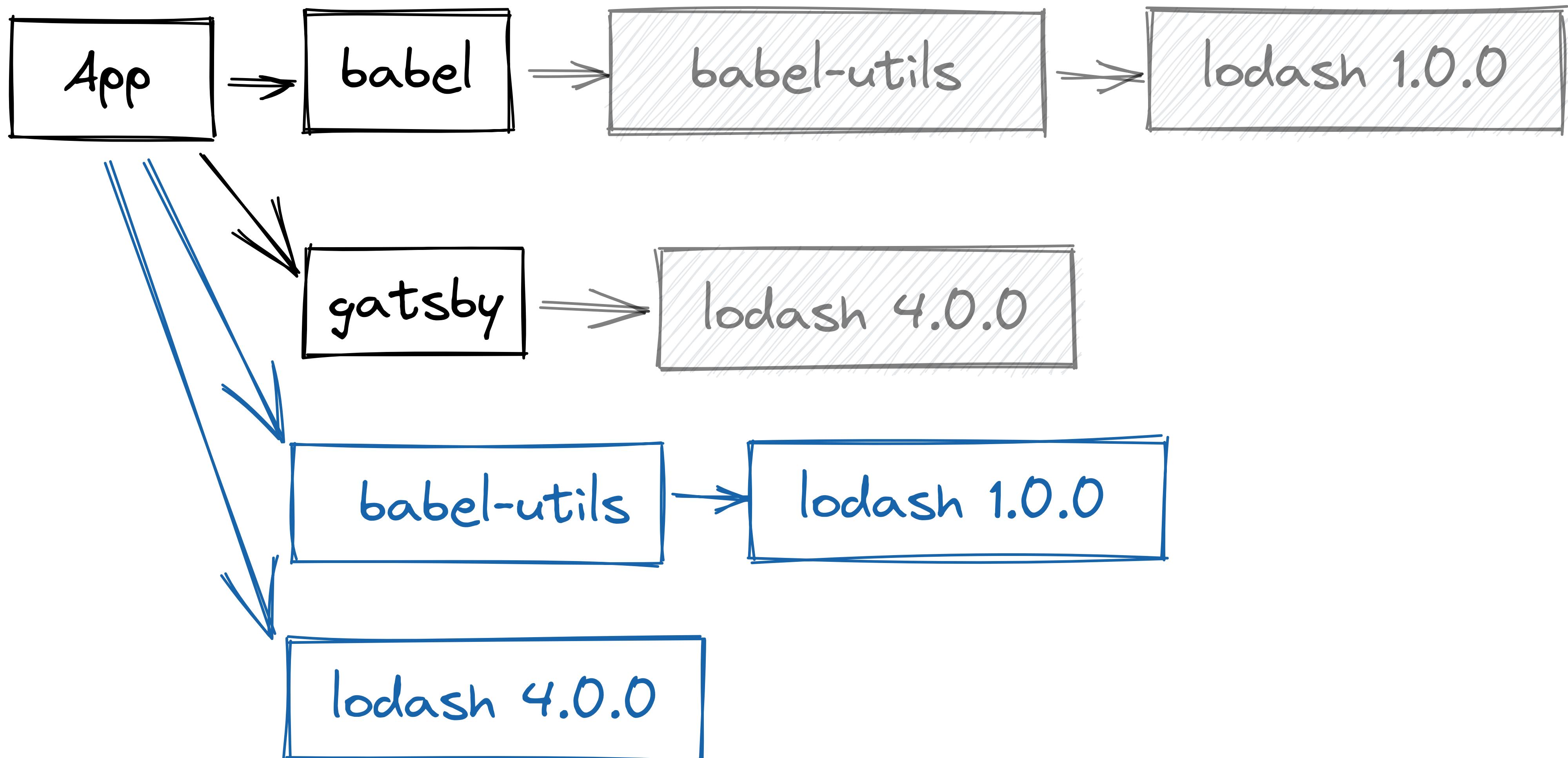
```
node_modules
  └── babel
      ├── babel-utils
      └── lodash
```

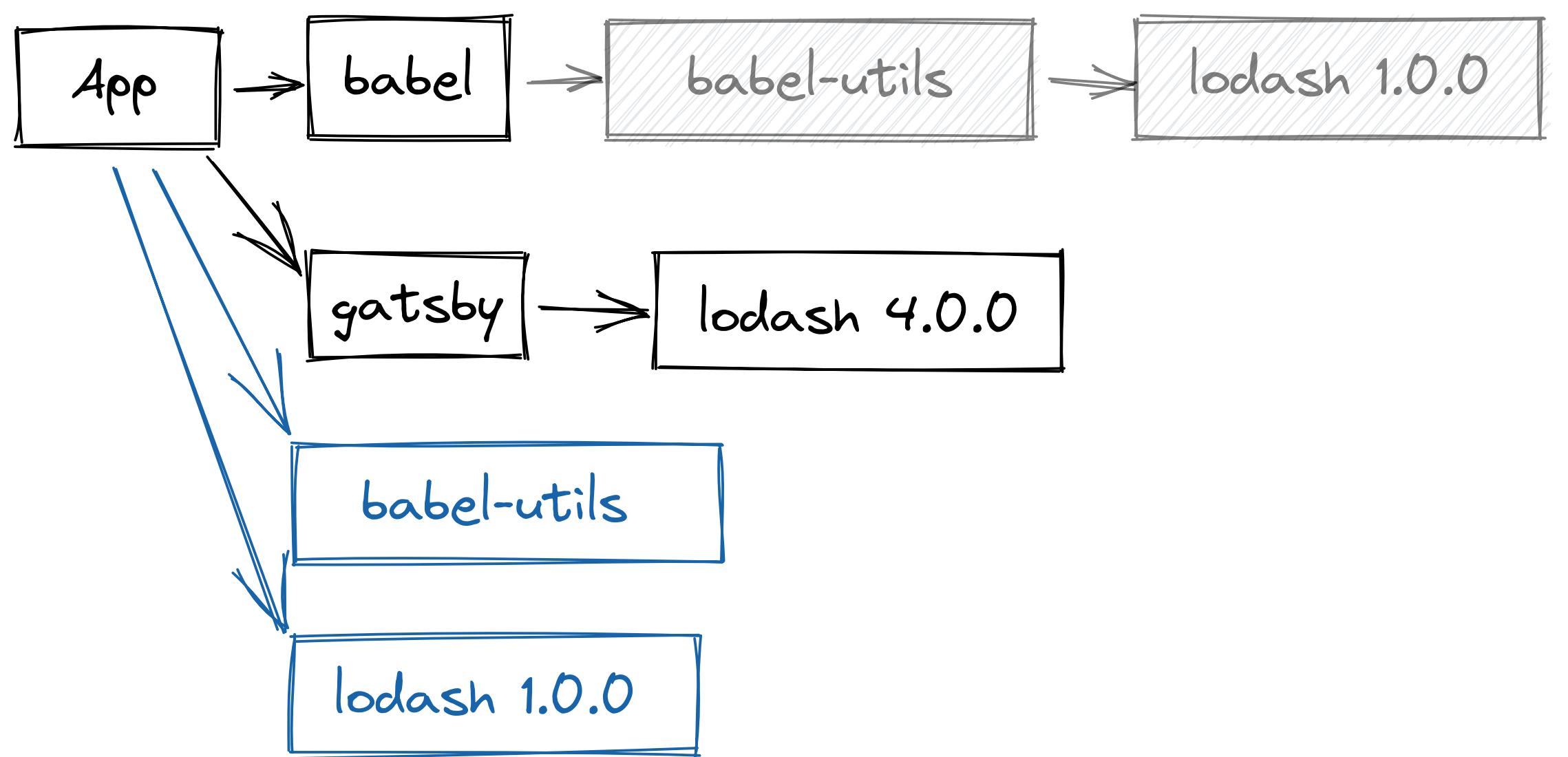
Phantom dependencies



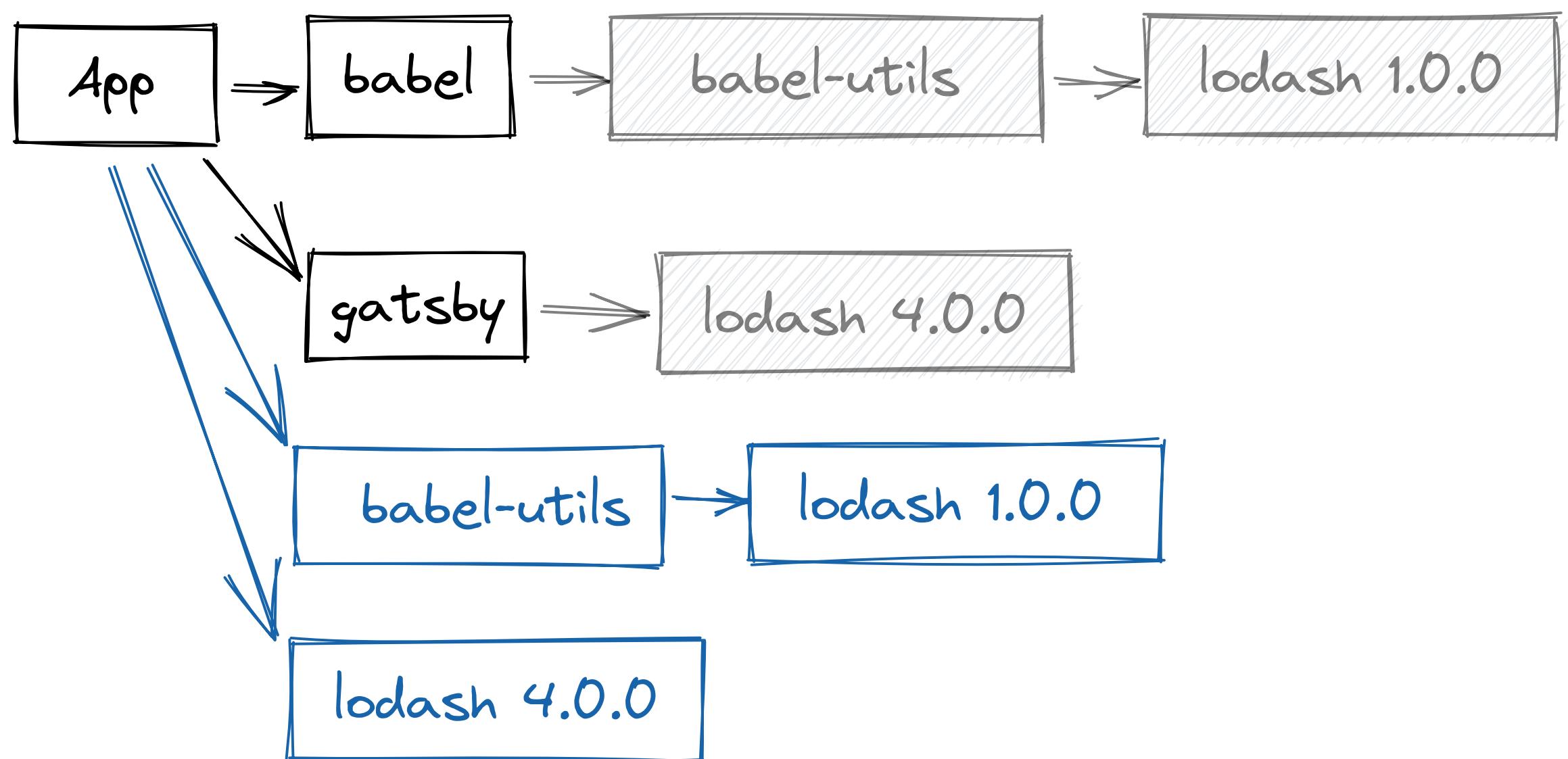






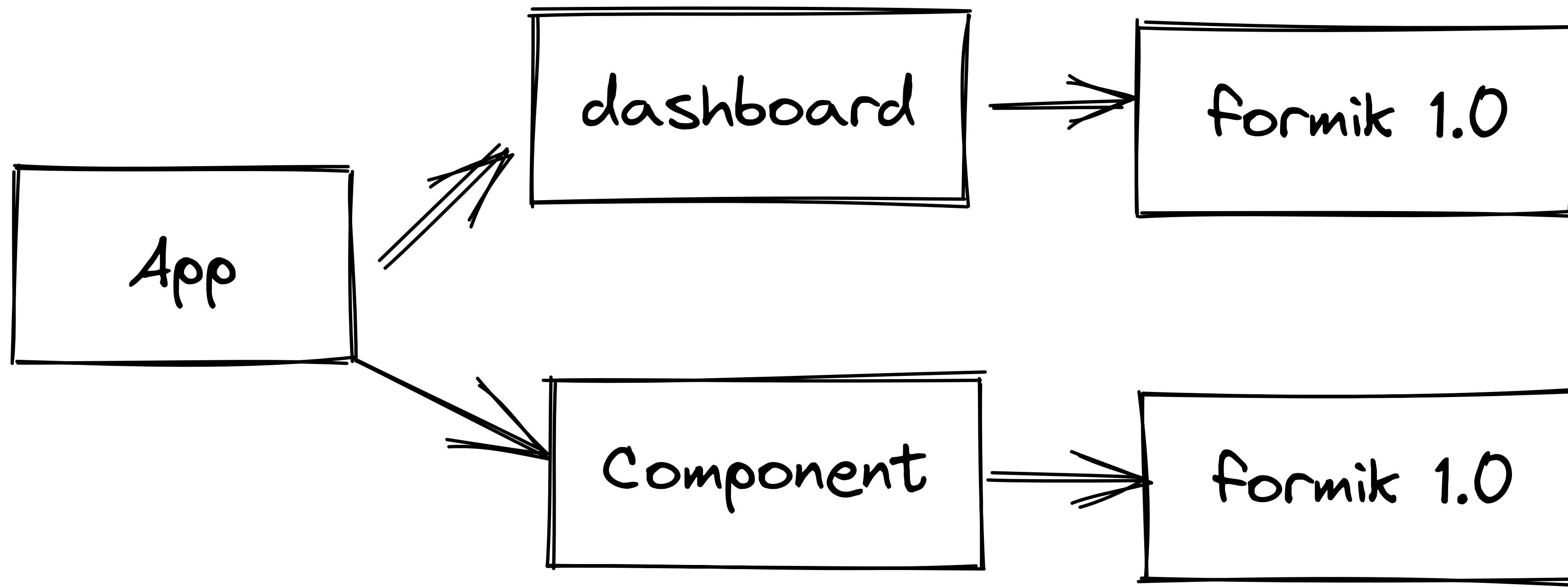


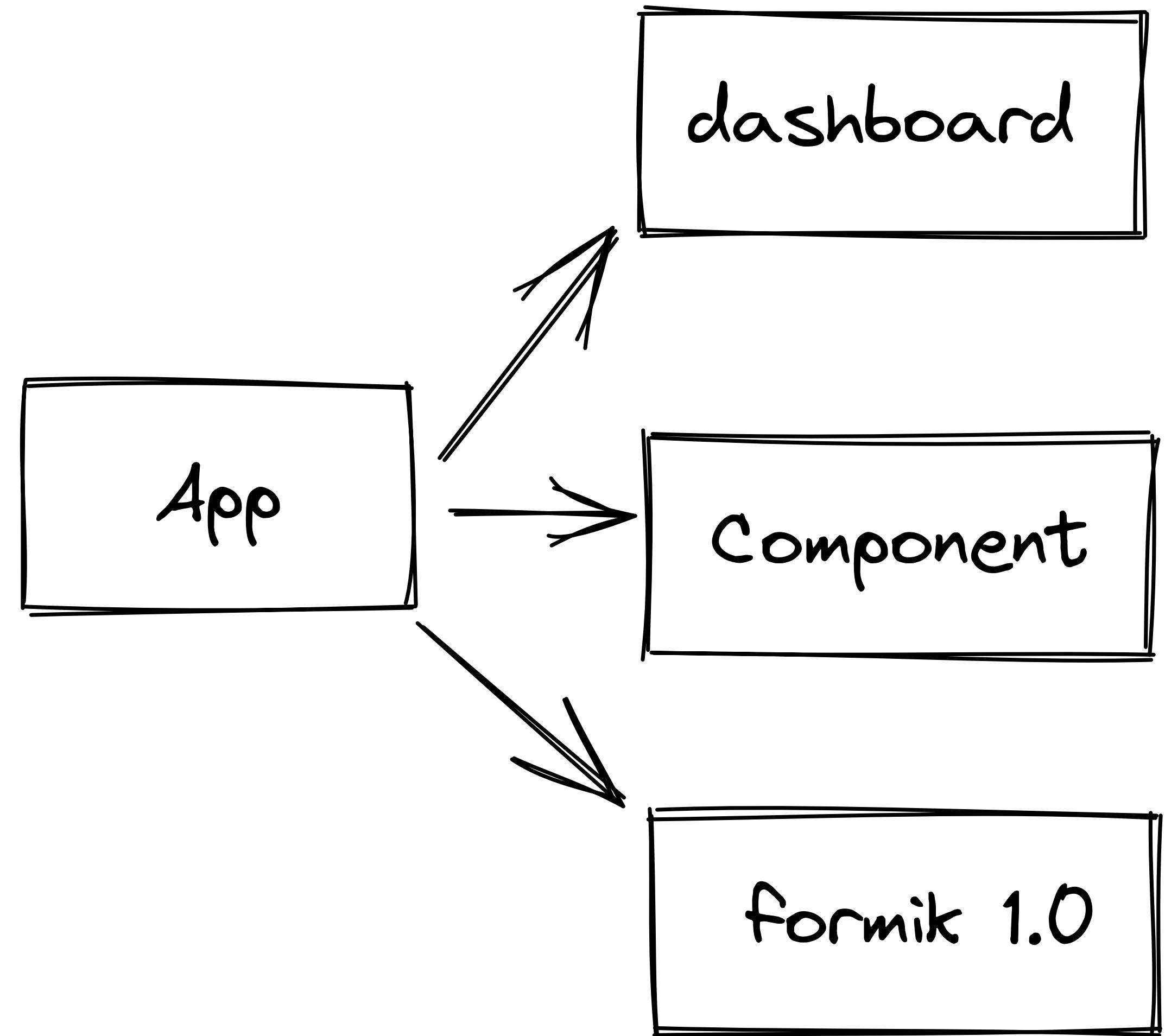
VS

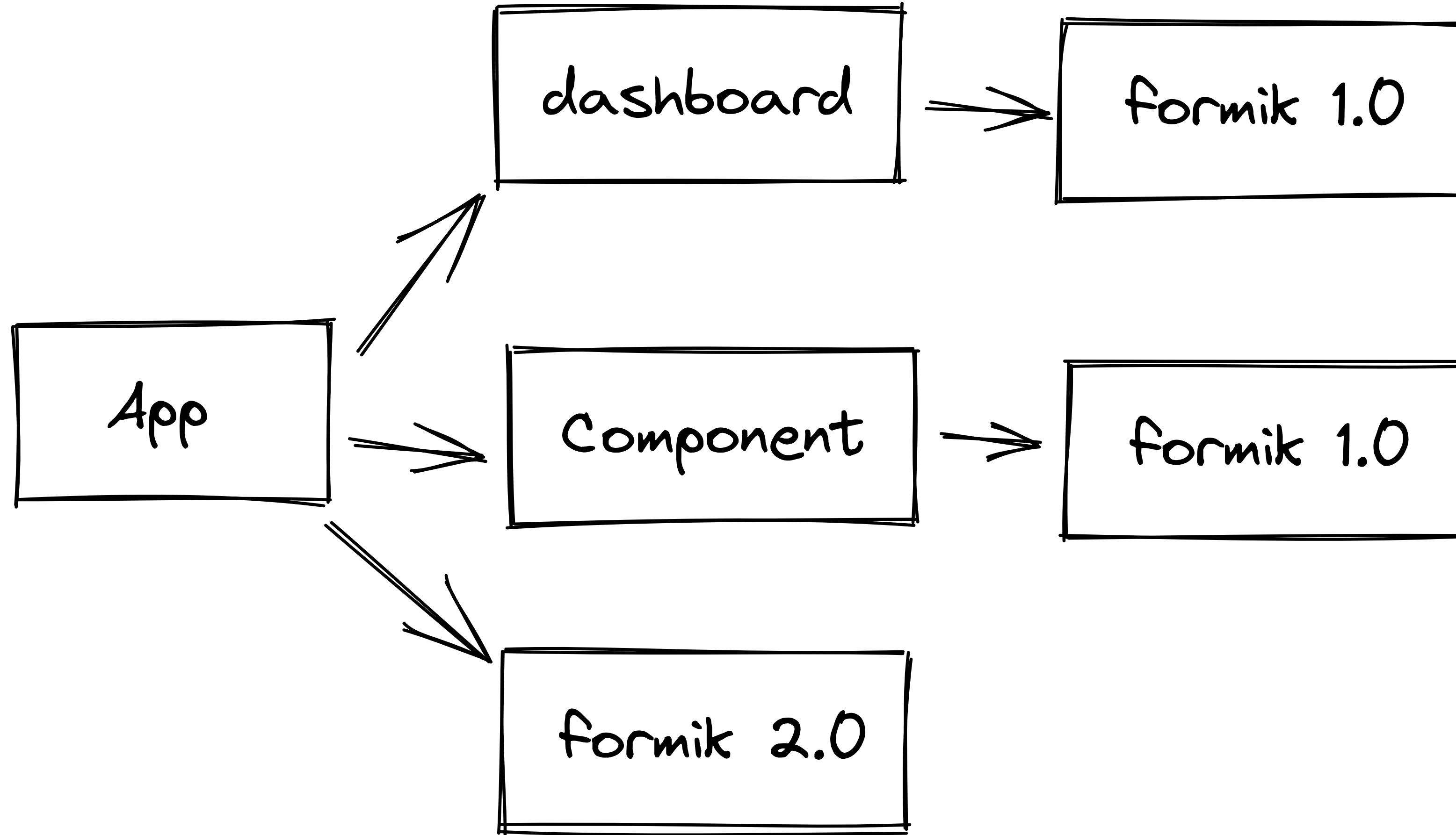


Doppelgangers









Flat node_modules

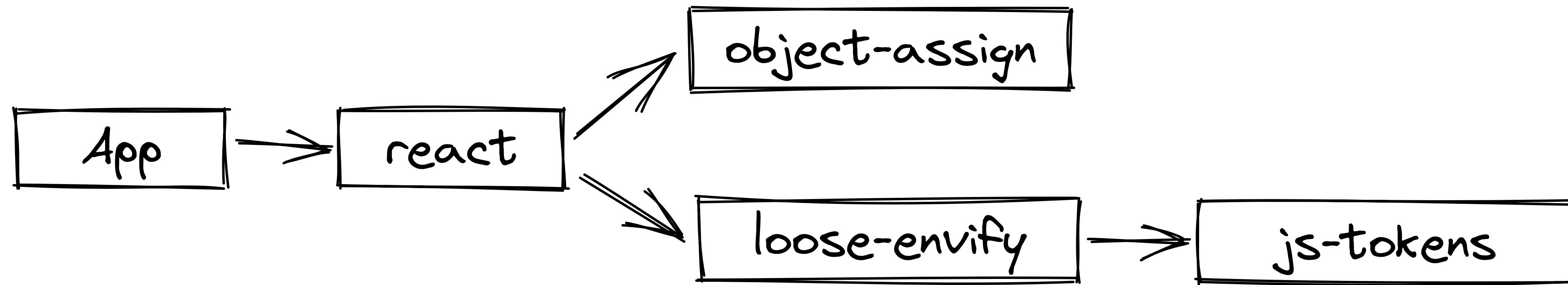
Плюсы:

???

Минусы:

1. Сложный недетерминированный алгоритм хоистинга
2. Фантомные зависимости
3. Дубликаты пакетов

2. Symlink way



```
node_modules
├── pnpm
│   └── js-tokens@4.0.0
│       └── node_modules
│           └── js-tokens
└── loose-envify@1.4.0
    └── node_modules
        ├── js-tokens -> ../../js-tokens@4.0.0/node_modules/js-tokens
        └── loose-envify
└── object-assign@4.1.1
    └── node_modules
        └── object-assign
└── react@17.0.2
    └── node_modules
        ├── loose-envify -> ../../loose-envify@1.4.0/node_modules/loose-envify
        ├── object-assign -> ../../object-assign@4.1.1/node_modules/object-assign
        └── react
└── react -> .pnpm/react@17.0.2/node_modules/react
```

Symlink

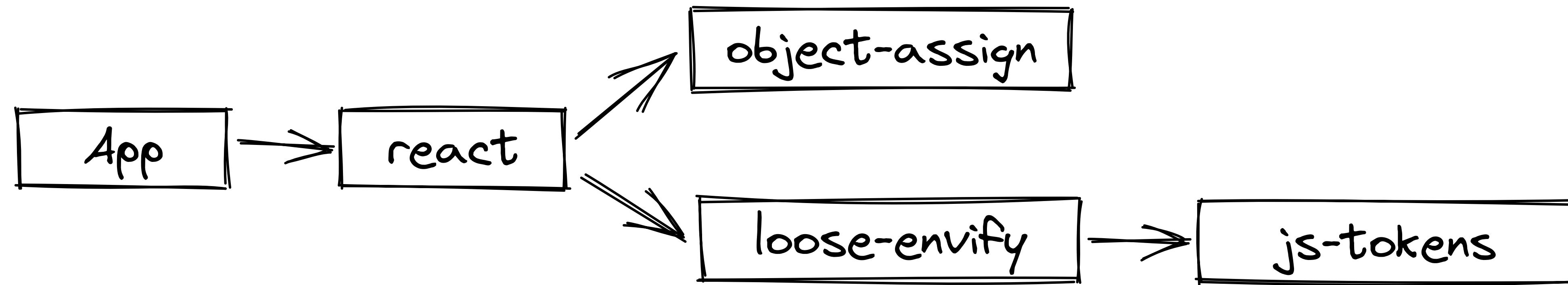
Плюсы:

1. Детерминированный и простой алгоритм
2. Строже чем flat node_modules
3. Появление дубликатов невозможно

Минусы:

1. Зависимости могут обращаться к основным зависимостям проекта

3. Plug N' Play





node_modules?

~~node_modules?~~

- 1. Скачиваем архив**
- 2. Парсим и распаковываем**
- 3. Копируем файлы в node_modules**

- 1. Скачиваем архив**
- 2. Парсим и распаковываем**
- 3. Копируем файлы в node_modules**

```
• yarn/cache/
  └── js-tokens-npm-4.0.0-0ac852e9e2-8a95213a5a.zip
  └── loose-envify-npm-1.4.0-6307b72ccf-6517e24e0c.zip
  └── object-assign-npm-4.1.1-1004ad6dec-fcc6e4ea8c.zip
  └── react-npm-17.0.2-99ba37d931-b254cc17ce.zip
```

Runtime

.pnp.cjs + .pnp.loader.mjs

Runtime

1. Переопределение поведения FS и require
2. Чтение из архива
3. Валидация связей между модулями
4. Кастомизация поиска модуля для ESM (import)

Plug N' Play

Плюсы:

1. Детерминированный
2. Самый строгий
3. Двойников не существует
4. Сильно экономит время при установке

Минусы:

1. Манкипатчинг
2. Node.js стартует дольше, чем обычно
3. Не все инструменты поддерживают официально (например, TypeScript)

	npm	pnpm	Yarn
Flat node_modules	+	+	+
Symlink	-	+	+
Plug'N'Play	-	+	+

2

npm

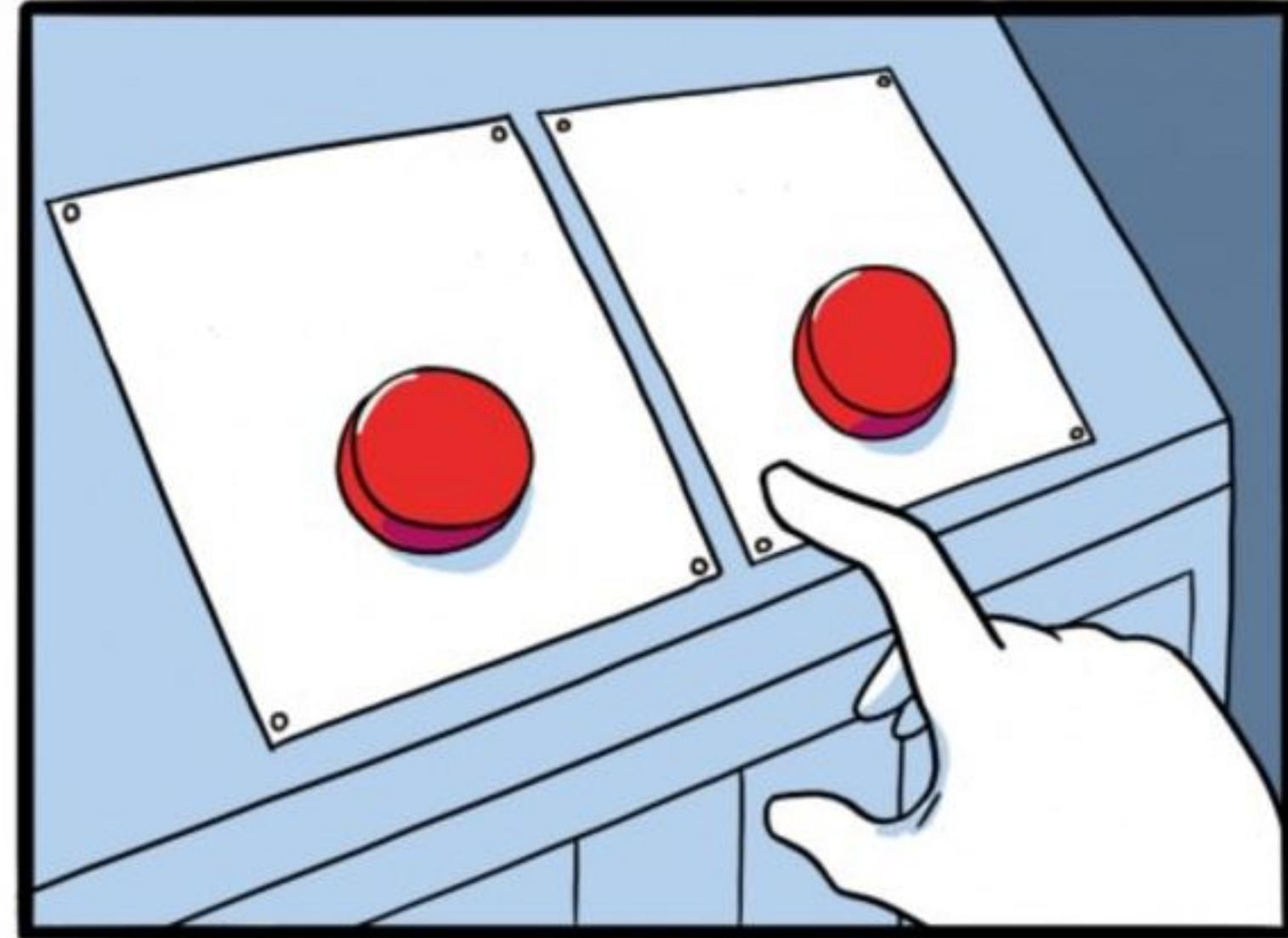
6

pnpm

5

Yarn

3. Governance model



JAKE-CLARK.TUMBLR

Нет стандарта
Нужен контроль над разработкой

npm 7+ автоматически устанавливает peerDependencies

github.blog/2021-02-02-npm-7-is-now-generally-available/#peer-dependencies
github.com/npm/feedback/discussions/610
github.com/npm/feedback/discussions/615

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

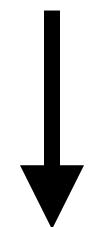
SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.

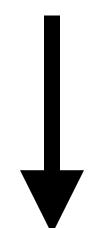


SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.



GitHub





yarn

pnpm

	npm	pnpm	Yarn
Roadmap	+	+	+
Публичный чат	-	+	+
Прозрачный процесс принятия решений	+	-	+
Понятный состав команды	-	-	+
Большой бюджет на разработку	+	-	-

5

npm

8

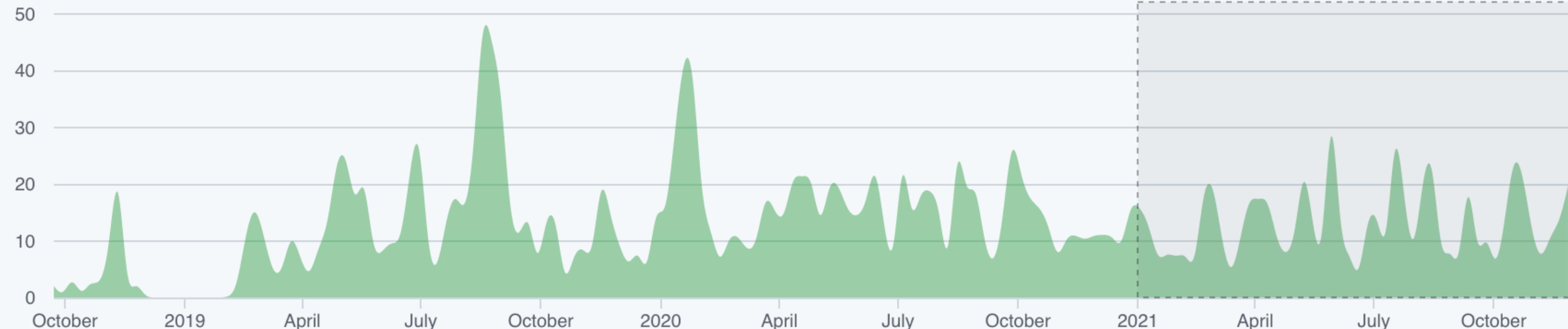
pnpm

9

Yarn

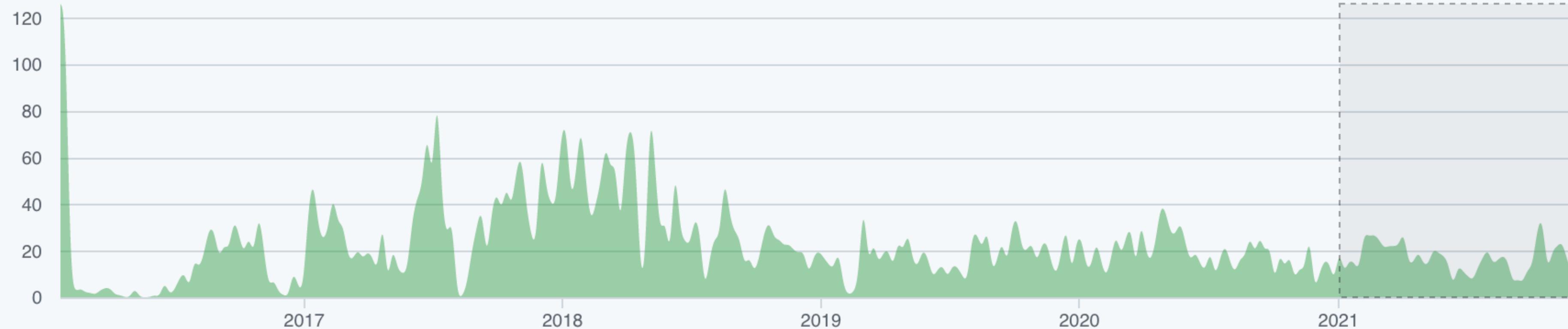
Bonus:
Вовлечённость коммьюнити

yarn berry



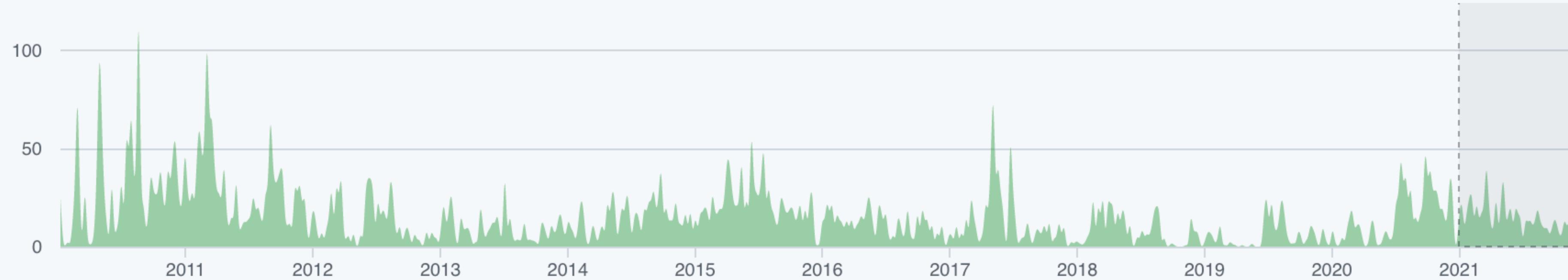
48
контрибьюторов

pnpm



37
контрибьюторов

npm



24
контрибьютора

5

npm

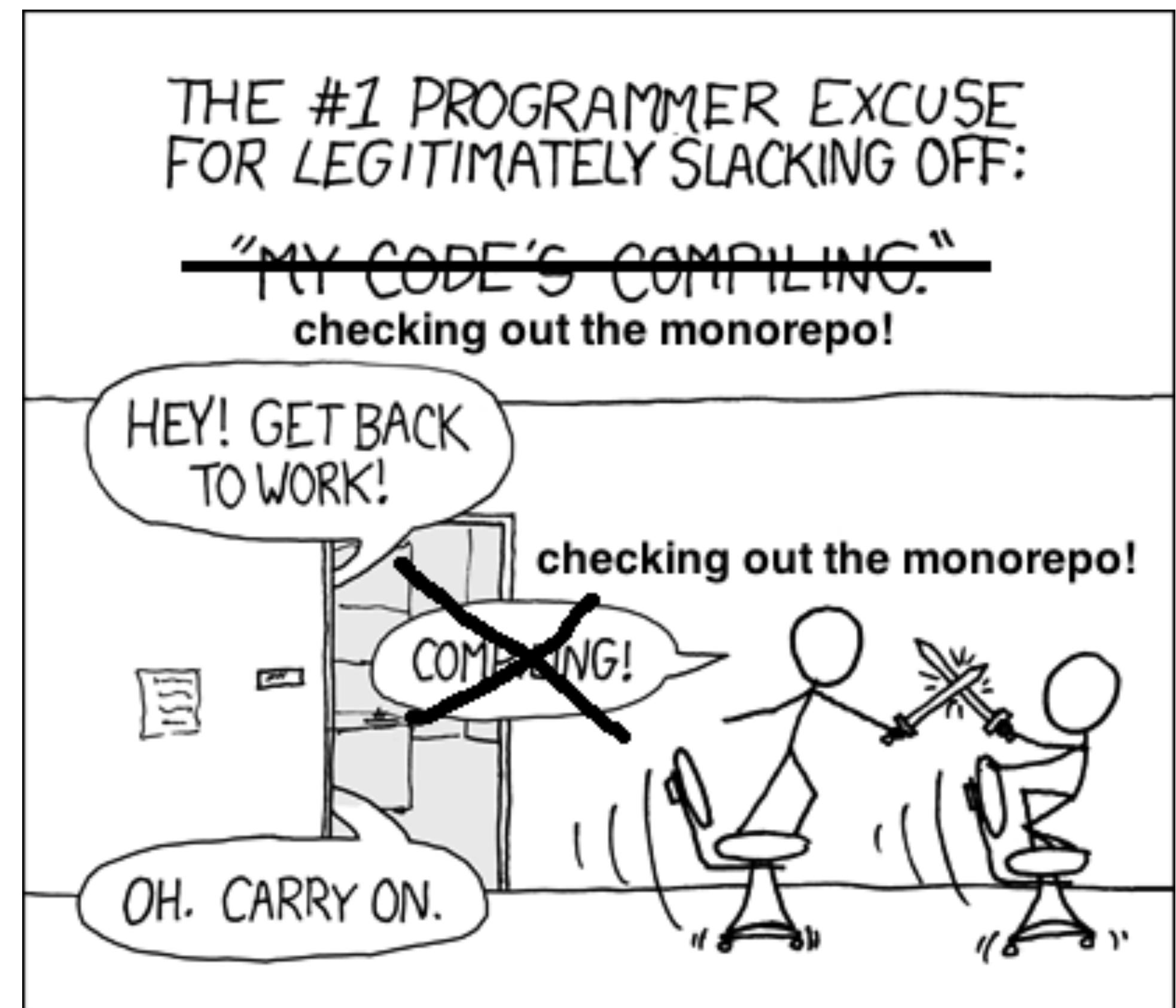
8

pnpm

10

Yarn

4. Monorepo (workspaces)



4.1 Как связываются воркспейсы между собой?

```
{  
  "name": "my-super-big-monorepo",  
  "workspaces": [  
    "packages/workspace-a",  
    "packages/workspace-b"  
  ]  
}
```

```
{  
  "name": "workspace-a",  
  "version": "1.0.0",  
  "dependencies": {  
    "workspace-b": "2.0.0"  
  }  
}
```



```
{  
  "name": "workspace-b",  
  "version": "2.0.0"  
}
```

```
{  
  "name": "workspace-a",  
  "version": "1.0.0",  
  "dependencies": {  
    "workspace-b": "2.0.0"  
  }  
}
```

```
{  
  "name": "workspace-b",  
  "version": "3.0.0"  
}
```

HTTP

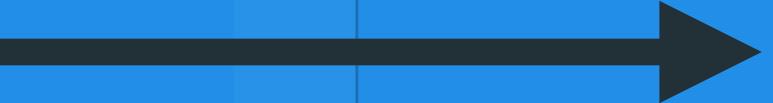
Package Registry

```
{  
  "name": "workspace-a",  
  "version": "1.0.0",  
  "dependencies": {  
    "workspace-b": "2.0.0"  
  }  
}
```

```
{  
  "name": "workspace-b",  
  "version": "3.0.0"  
}
```

workspace:

```
{  
  "name": "workspace-a",  
  "version": "1.0.0",  
  "dependencies": {  
    "workspace-b": "workspace:/*"  
  }  
}
```



```
{  
  "name": "workspace-b",  
  "version": "2.0.0"  
}
```

```
{  
  "dependencies": {  
    "star": "workspace:*",  
    "caret": "workspace:^",  
    "tilde": "workspace:~",  
    "range": "workspace:^1.2.3",  
    "path": "workspace:path/to/baz"  
  }  
}
```

```
{  
  "dependencies": {  
    "star": "1.5.0",  
    "caret": "^1.5.0",  
    "tilde": "~1.5.0",  
    "range": "^1.2.3",  
    "path": "1.5.0"  
  }  
}
```

5

npm

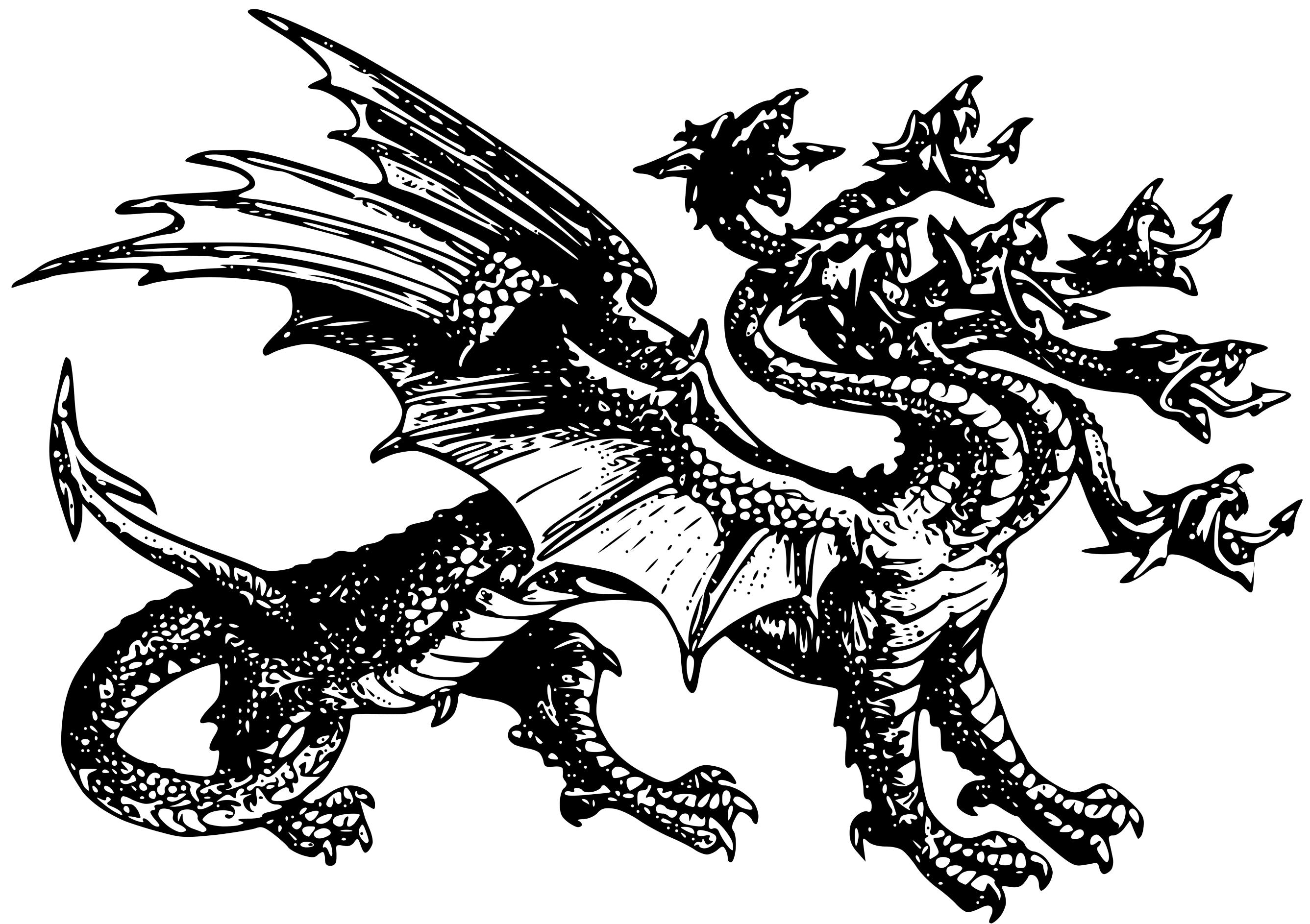
9

pnpm

11

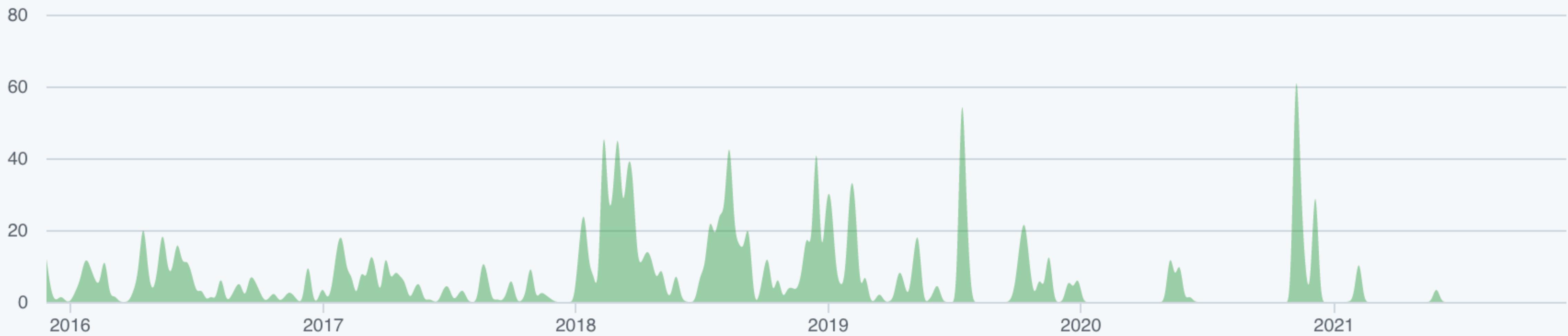
Yarn

4.2 Запуск команд в монорепозитории



Lerna

Lerna is dead too



Запуск команд в моно-репозитории

	npm	pnpm	yarn
Параллельность	-	--workspace-concurrency	--jobs
Фильтрация по GIT	-	--filter "[<since>]"	--since
Белый список	--workspace=a	--filter	--include
Черный список	-	--filter	--exclude

npm run / yarn run / pnpm run

npm

6

pnpm

13

Yarn

14

5. Security

preinstall

install

postinstall



```
{  
  "name": "core-js",  
  "description": "Standard library",  
  "version": "3.19.2",  
  "repository": {  
    "type": "git",  
    "url": "https://github.com/zloirock/core-js.git"  
  },  
  "main": "index.js",  
  "license": "MIT",  
  "scripts": {  
    "postinstall": "node -e \"try{require('./postinstall')}catch(e){}\""  
  }  
}
```

Yarn: yarnrc#enableScripts

pnpm: --ignore-scripts + pnp.neverBuiltDependencies

npm: github.com/npm/rfcs/pull/488 :(

npm

6

pnpm

14

Yarn

15

6. Расширяемость

yarnpkg.com/features/plugins

npm

6

pnpm

14

Yarn

16

Критерии

1. Быстродействие
2. Структура node_modules
3. Модель управления разработкой
4. Поддержка монорепозиториев
5. Безопасность
6. Расширяемость

npm

6

pnpm

14

Yarn

16

Выводы

1. npm сильно отстает по совокупности факторов
2. Ни в одном из кейсов не является лучшим
3. Худший с точки зрения структуры node_modules – основной задачи пакетного менеджера

**Окей, убедил.
Как договориться?**

Corepack

Node.js 16.9.0+

nodejs.org/api/corepack

```
{  
  "name": "corepack-example",  
  "version": "1.0.0",  
  "main": "index.js",  
  "license": "MIT",  
  "packageManager": "yarn@3.1.1"  
}
```

```
{  
  "name": "corepack-example",  
  "version": "1.0.0",  
  "main": "index.js",  
  "license": "MIT",  
  "packageManager": "pnpm@6.22.2"  
}
```

github.com/pnpm/only-allow

```
{  
  "scripts": {  
    "preinstall": "npx only-allow pnpm"  
  }  
}
```

```
{  
  "scripts": {  
    "preinstall": "npx only-allow yarn"  
  }  
}
```

The End

CI/CD Consulting

Семирульник Валентин

@7rulnik

t.me/valya_reads_issue