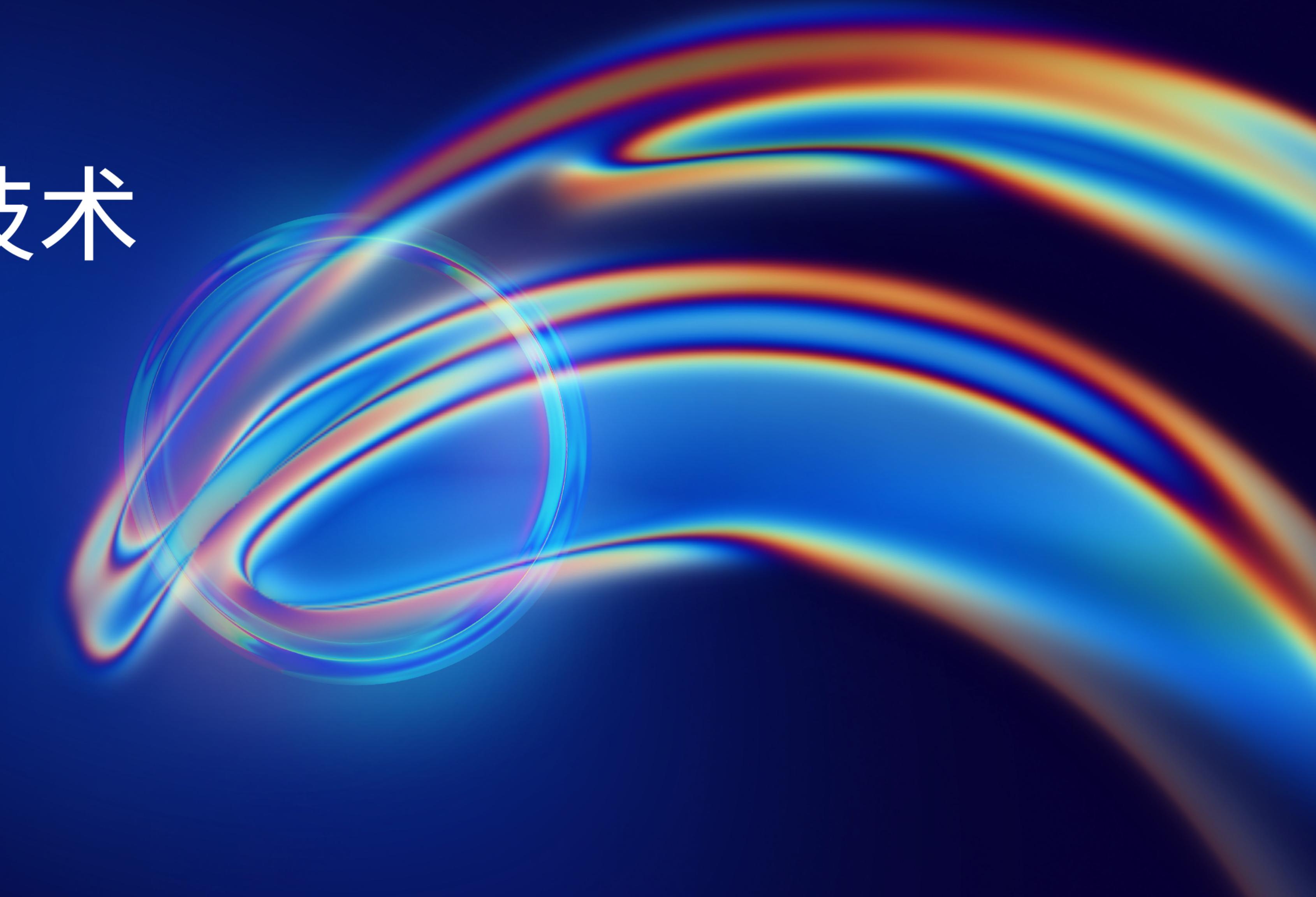


WOT | 51CTO

World Of Tech 2023

WOT全球技术 创新大会

新 技 术 瞰 未 来
2023



字节跳动的前端工程化实践

林宜丙 前端工程师



字节跳动-Web Infra-前端架构工程师，拥有多年前端工程化经验，致力于帮助前端工程师更好地管理和治理工程，目前负责工程治理方向的方案设计及其业务落地工作。

·PART ONE

前端趋势及其新挑战

·PART THREE

整体落地情况

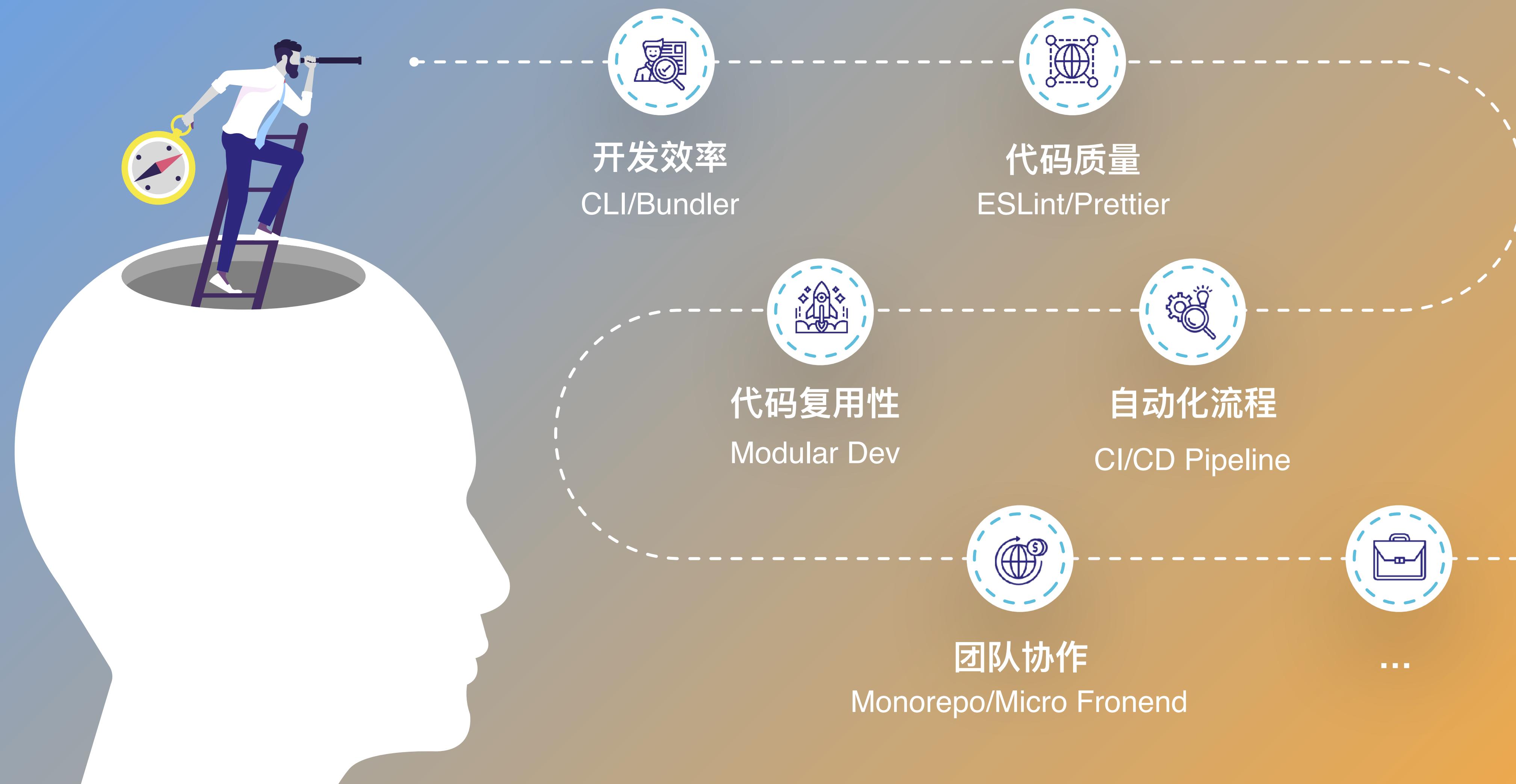
·PART TWO

应对挑战的实践

·PART FOUR

总结与展望

什么是前端工程化?



前端趋势-工种趋势



前端趋势-工程趋势



面临的新挑战



多项目维护成本高

项目基建重复、代码复用困难、工作流程割裂

多人开发协作成本高

相互依赖的流程、级联的依赖升级

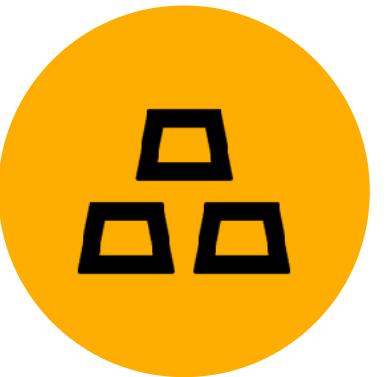
巨型应用构建速度慢

构建耗时随着应用增大而变慢

大型应用劣化速度快

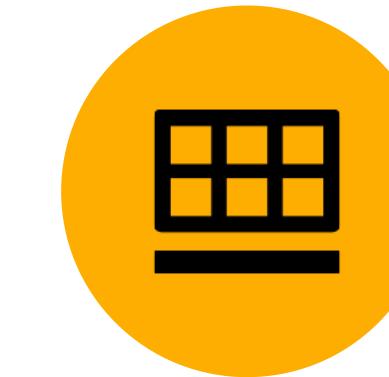
缺乏有效的防劣化手段

应对挑战的实践



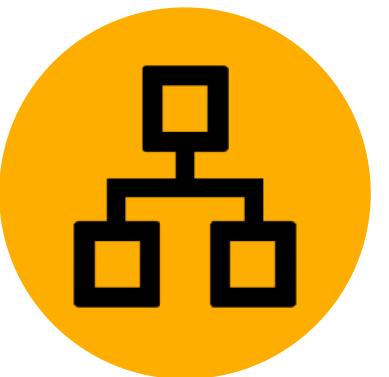
Monorepo

降低多项目的维护成本



Micro Frontend

降低多人开发的协作成本



Bundler & Build System

加快巨型应用的构建速度



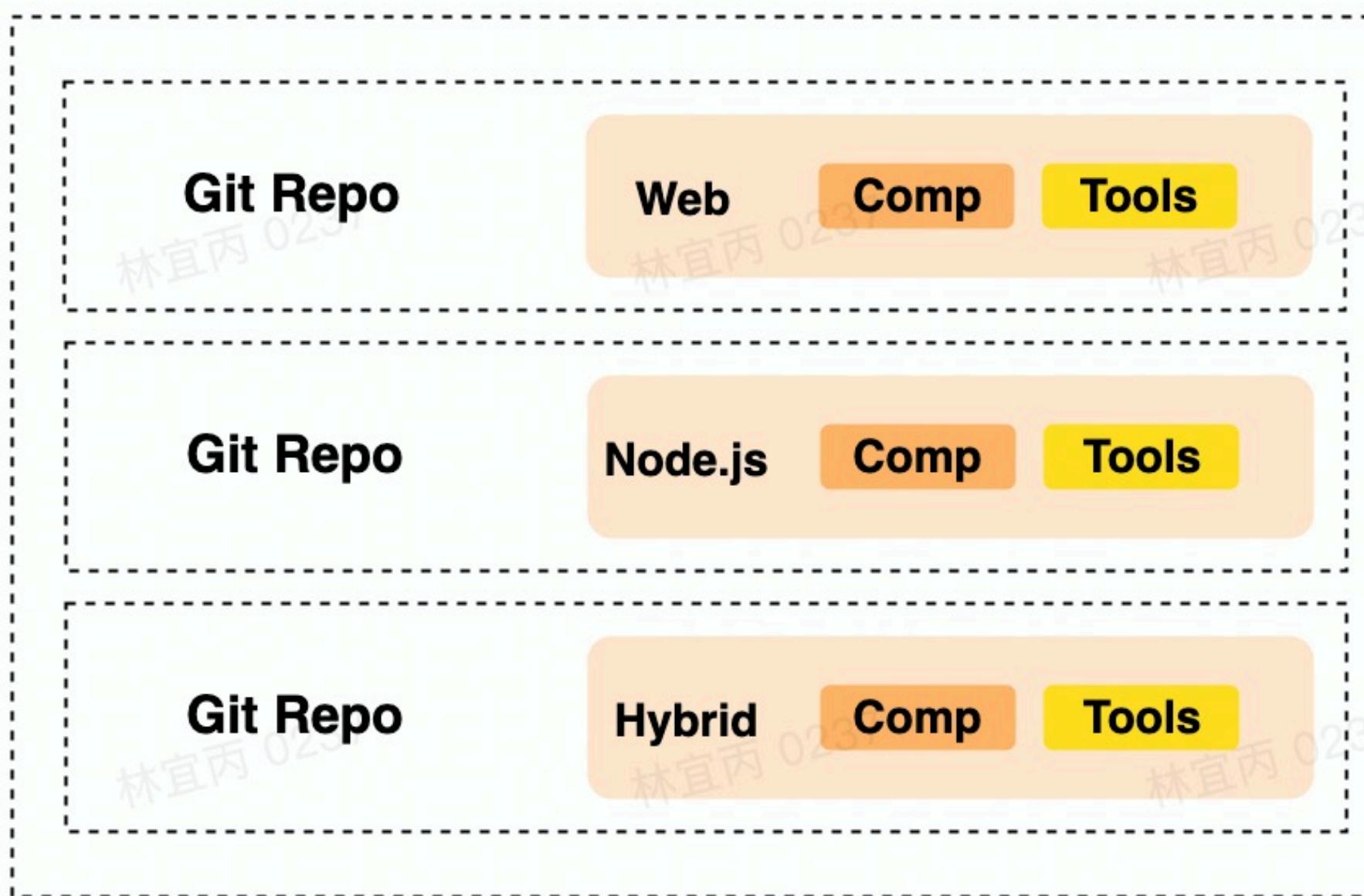
Diagnostics Tool

有效地防止应用劣化

Monorepo 简介

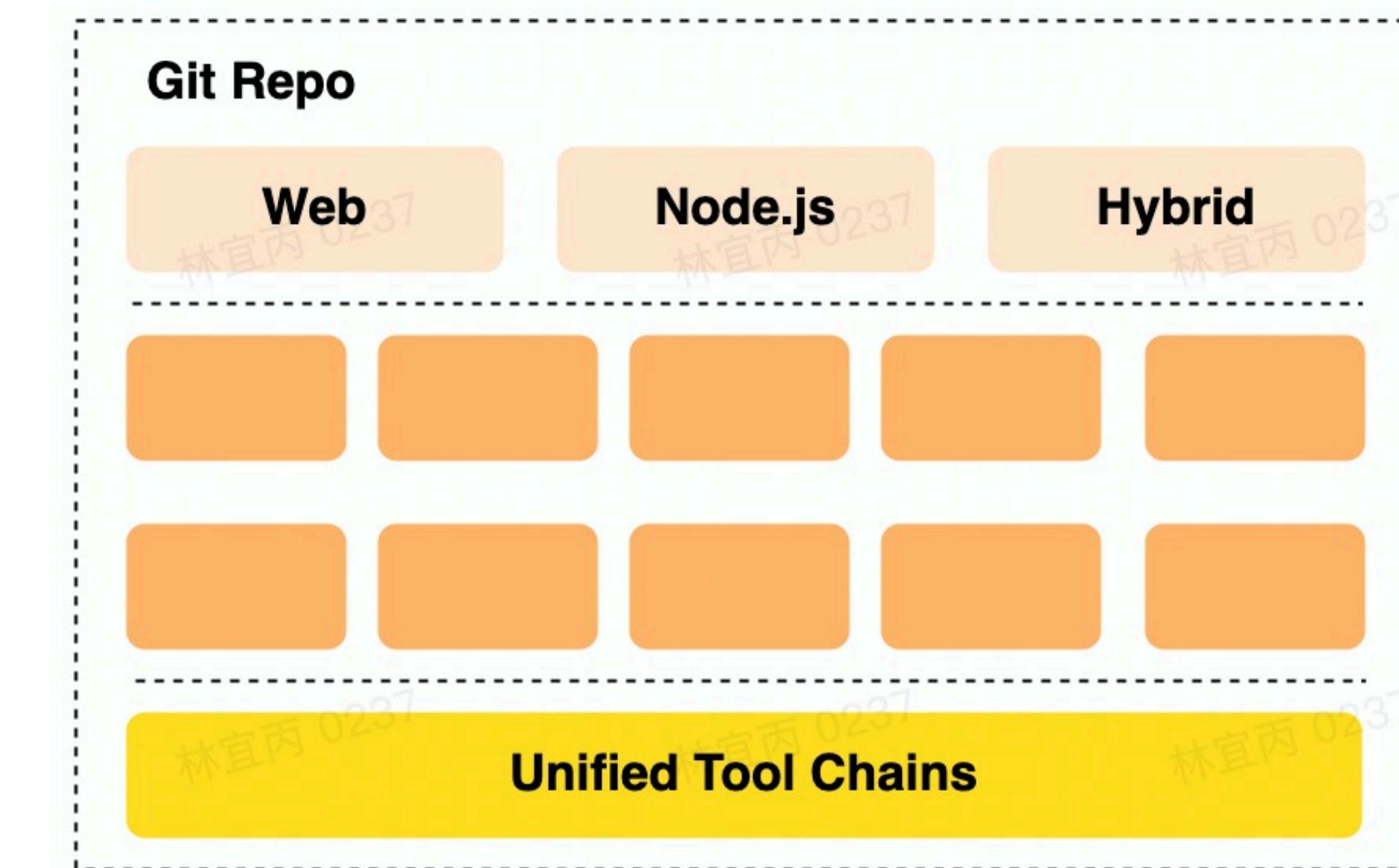
Polyrepo

子项目分布到不同的仓库中管理



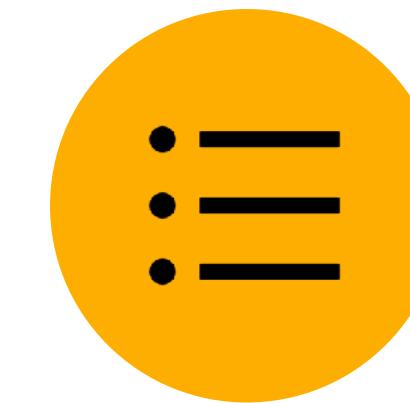
Monorepo

子项目组织到一个仓库中统一管理



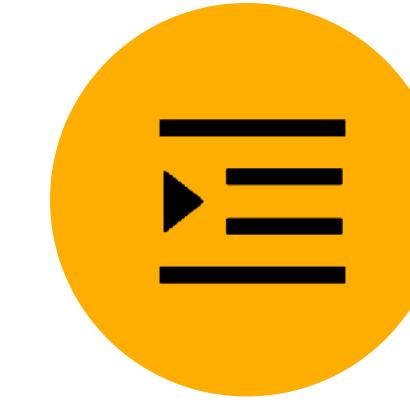
将多个项目组织良好地维护于单个仓库中

如何降低多项 目的维护成 本？



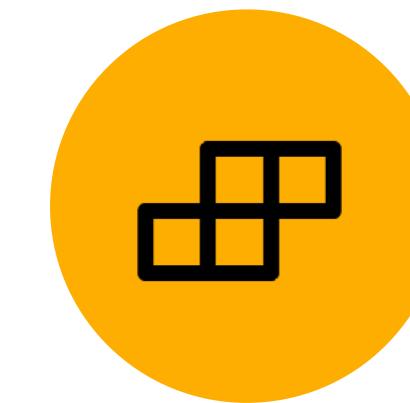
复用基建

让开发人员重新专注于应用本身



代码共享

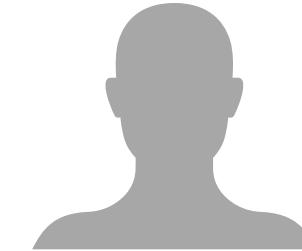
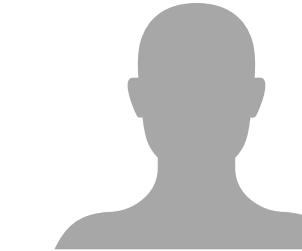
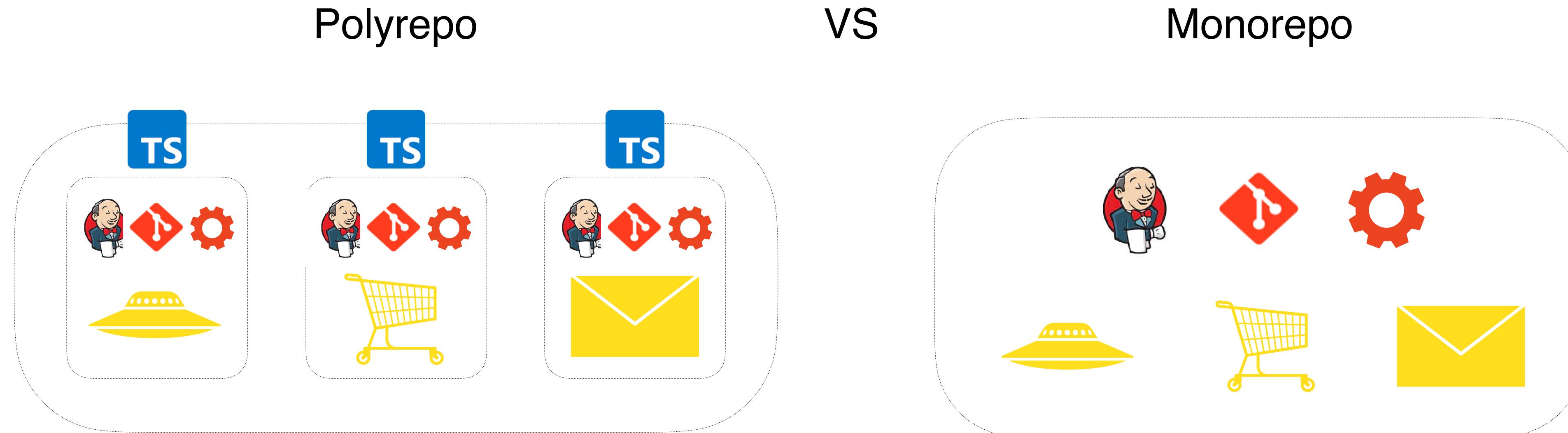
降低代码复用的成本



原子提交

使用自动化的多项目工作流

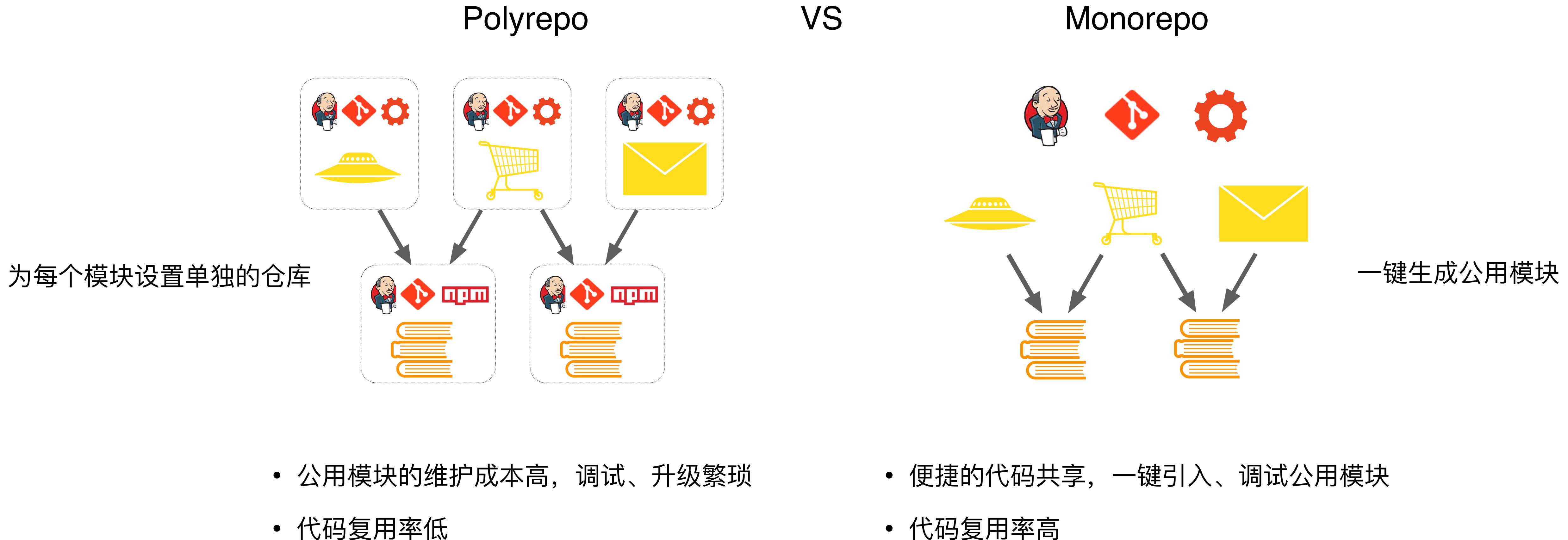
Monorepo - 复用基建



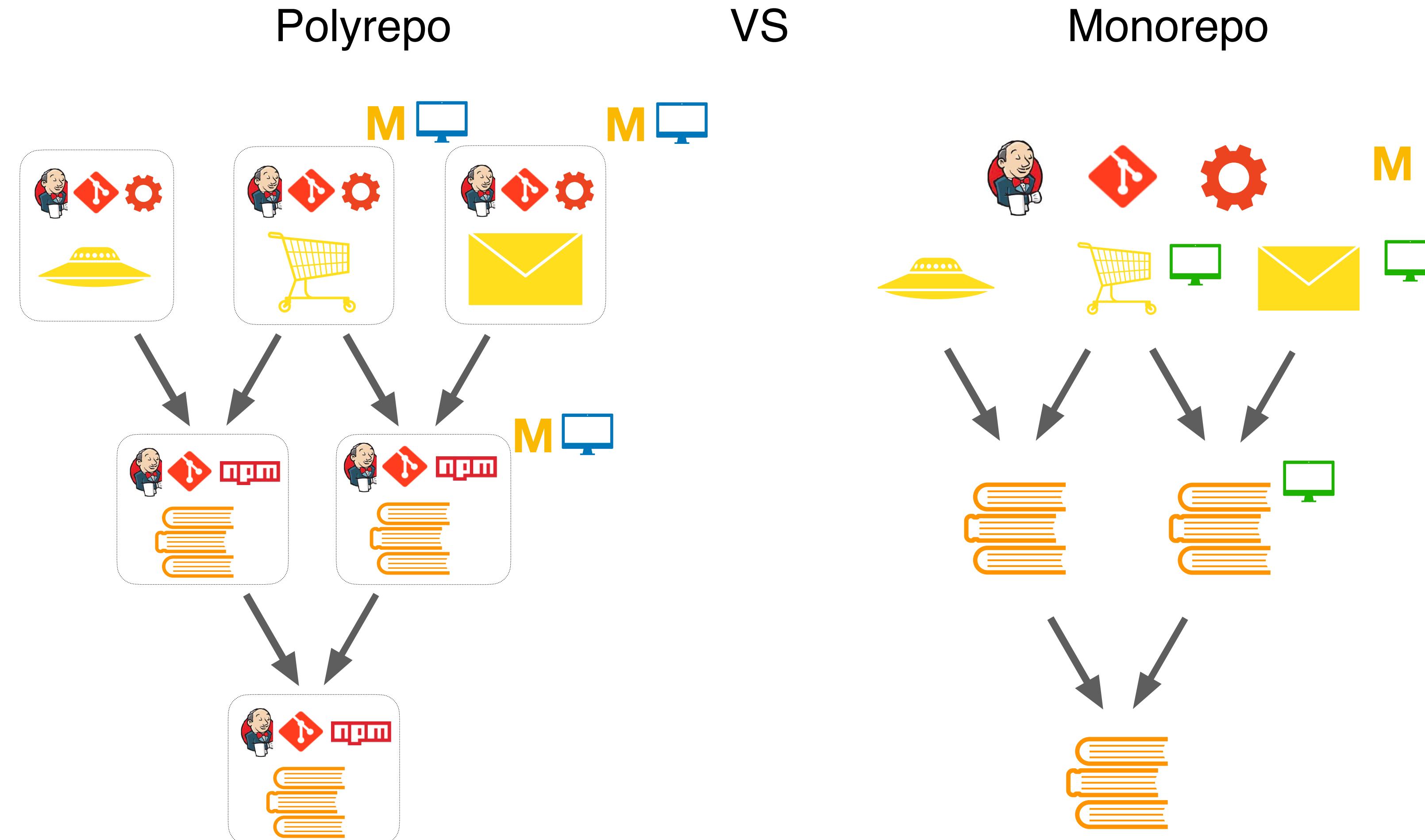
- 每个项目都需要有开发人员创建维护
- 基建重复，调整多个项目基建成本高

- 1-2个开发人员负责所有项目的公共架构维护
- 基建复用，调整多个项目的基建成本低

Monorepo - 代码共享



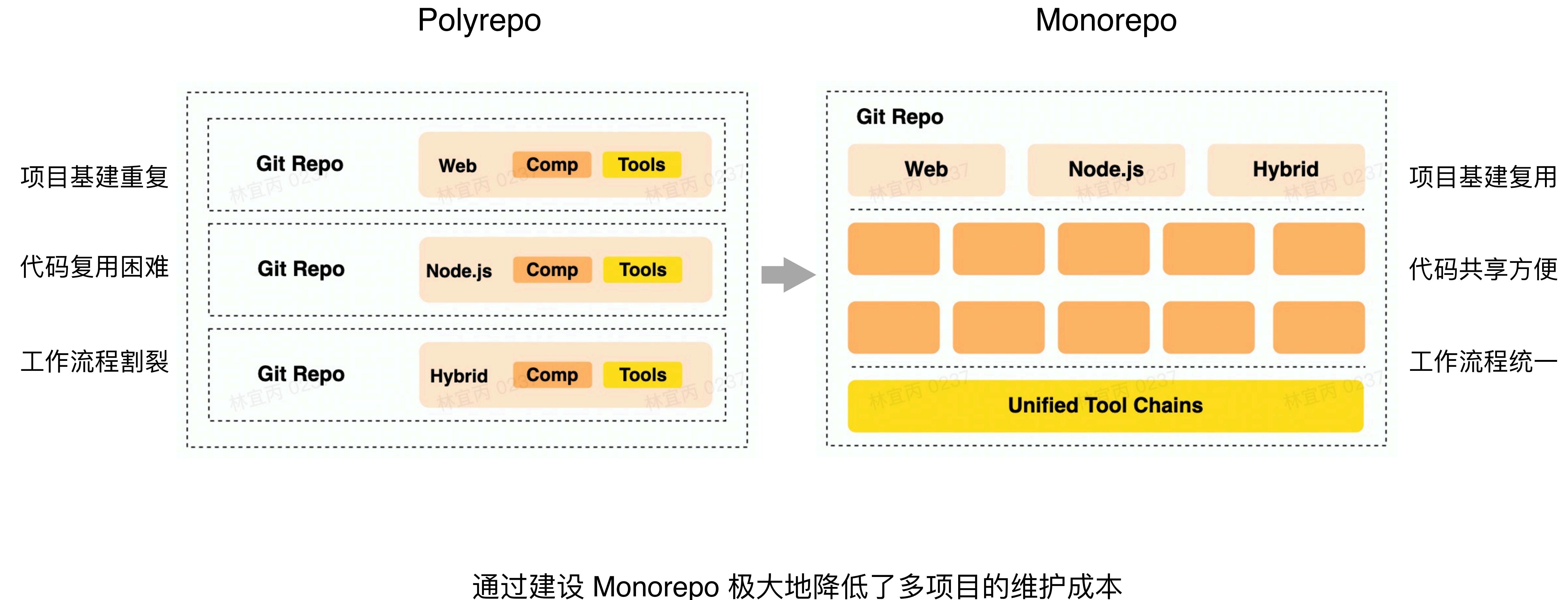
Monorepo - 原子提交



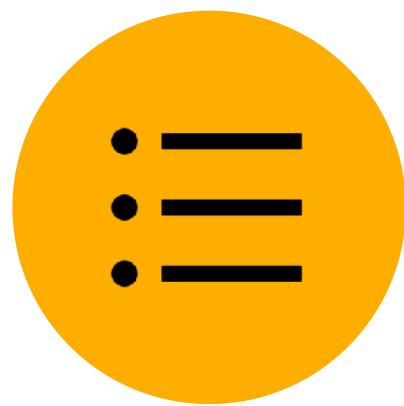
- 业务需求可能涉及多个项目
- 多项目工作流不连续

- 一次性调整并提交多个项目
- 自动、连续的 CI 流程

Monorepo - 总结

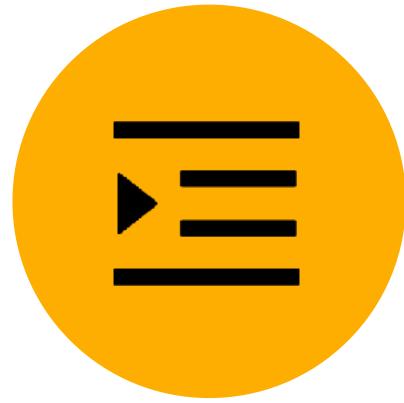


如何加快巨型 应用的构建速 度？



Bundler

加快巨石应用的构建速度

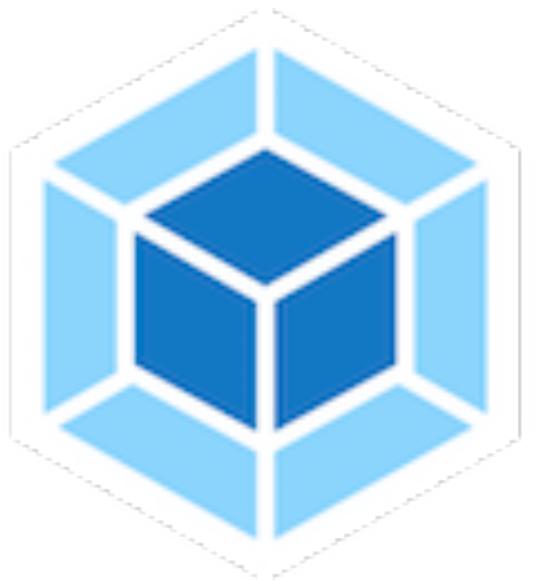


Build System

加快 Monorepo 的构建速度

Bundler 简介

处理文件之间的模块依赖关系图，并将其打包成静态资源



Webpack



Rollup



Vite



Parcel



Esbuild

Bundler 实践

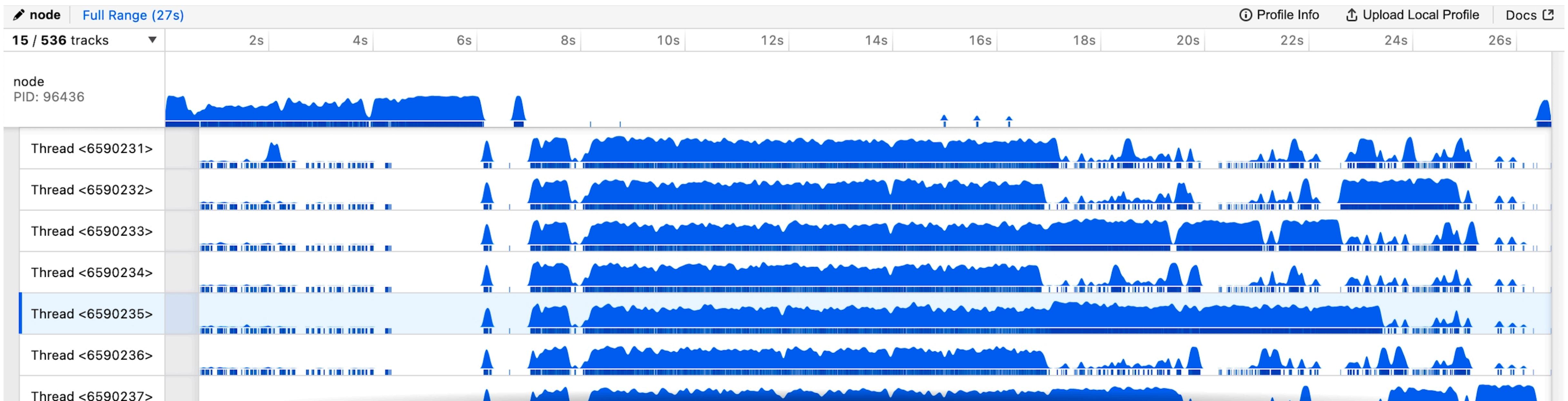
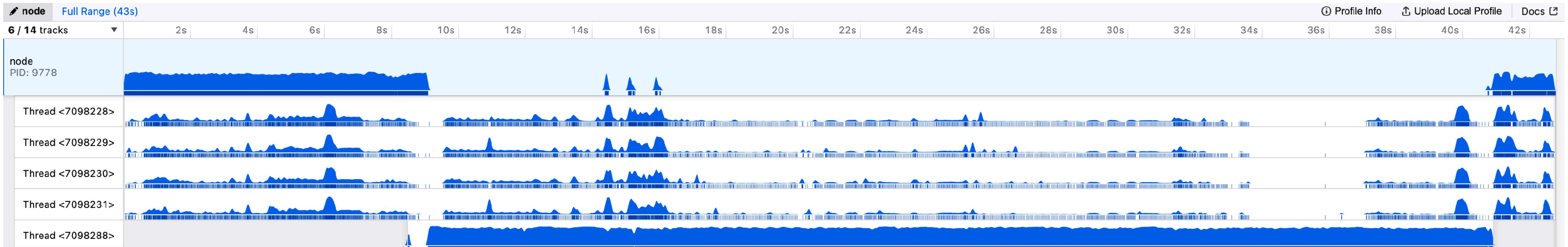
Rspack 是一个基于 Rust 的高性能构建引擎（Bundler），具备与 Webpack 生态系统的互操作性。



<https://www.rspack.dev>

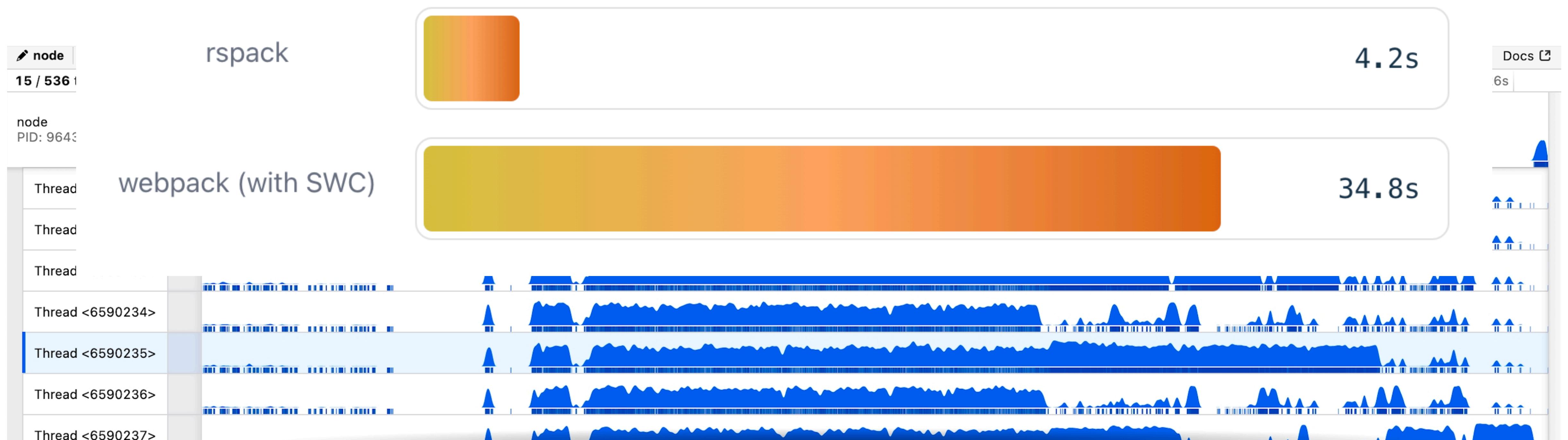
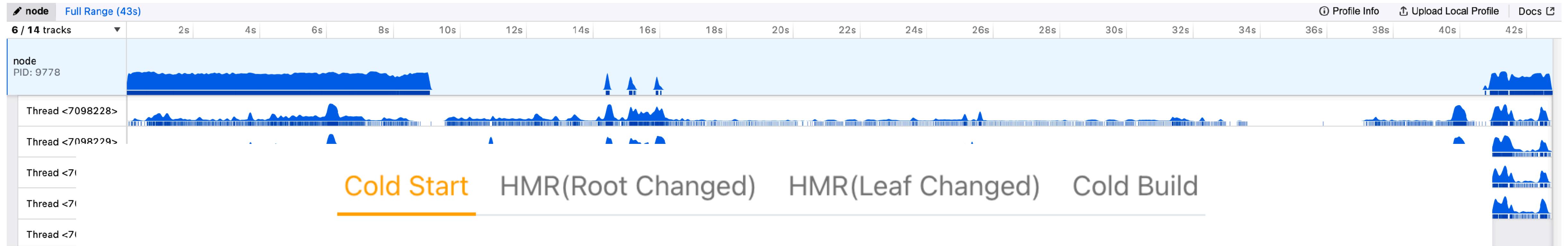
Bundler - Rust 语言实现

🦀 Rust 实现核心部分，充分利用并发特性



Bundler - Rust 语言实现

🦀 Rust 实现核心部分，充分利用并发特性



Bundler - Webpack 生态兼容

📦 与 Webpack 部分兼容

The subset of Webpack!

```
/** @type {import('@rspack/cli').Configuration} */
const config = {
  context: __dirname,
  entry: { main: "./src/index.tsx" },
  devServer: { port: 5555, webSocketServer: "sockjs", historyApiFallback: true },
  mode: prod ? "production" : "development",
  devtool: prod ? false : "source-map",
  cache: false,
  module: ...,
  resolve: { alias: { "@": path.resolve(__dirname, "src") } },
  output: { publicPath: "/", filename: "[name].[contenthash].js" },
  optimization: {
    realContentHash: true,
    splitChunks: { cacheGroups: { someVendor: { chunks: "all", minChunks: 2 } } }
  },
  plugins: [new HtmlWebpackPlugin({ title: "Arco Pro App" })],
  infrastructureLogging: { debug: false },
  builtins: { progress: {}, treeShaking: true, sideEffects: true, noEmitAssets: false }
};
```

- babel-loader
- style-loader
- css-loader
- postcss-loader
- sass-loader
- less-loader
- raw-loader
- file-loader
- url-loader
- svelte-loader
- @mdx-js/loader
- @svgr/webpack
- image-webpack-loader
- thread-loader
- source-map-loader
- node-loader
- vue-loader
-
- html-webpack-plugin => builtins.html
- react-refresh-webpack-plugin => builtins.react.refresh
- webpack.DefinePlugin => builtins.define
- webpack.ProvidePlugin => builtins.provide
- mini-css-extract-plugin => experiments.css
- tsconfig-paths-webpack-plugin => resolve.tsconfigPath
- copy-webpack-plugin => builtins.copy
- webpack-bundle-analyzer
- webpack-stats-plugin
-



<https://www.rspack.dev>

Bundler - 提升效果

	Webpack	Rspack	提升
Build	380s	28s	13 倍
Dev	368s	25s	14 倍
HMR	21s	1s	21 倍

接入业务1

	Webpack	Rspack	提升
Build	508s	59s	9 倍
Dev	243s	24s	10 倍
HMR	18s	1.4s	13 倍

接入业务2



Build System 简介

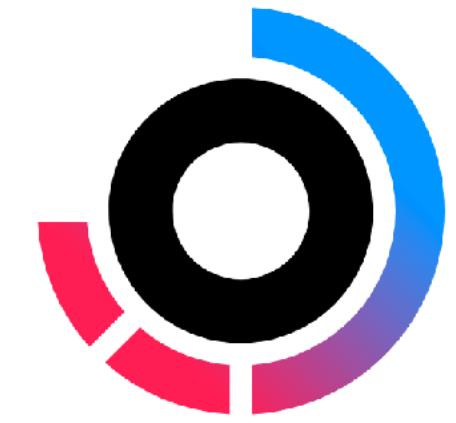
处理 Monorepo 下的项目依赖关系图，并据此调度构建任务



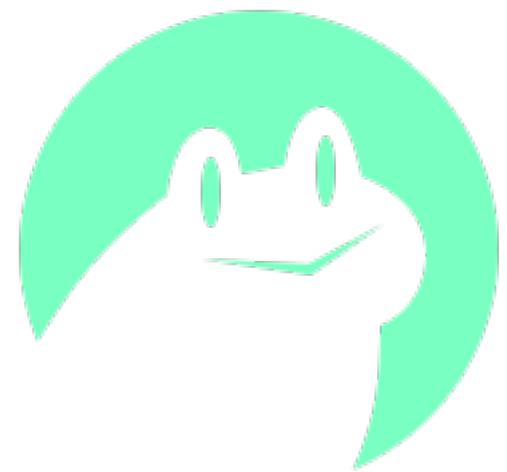
Bazel



NX



Turborepo



Lage

Build System 实践



任务并行能力

采用最大限度的并行任务加速



多级缓存能力

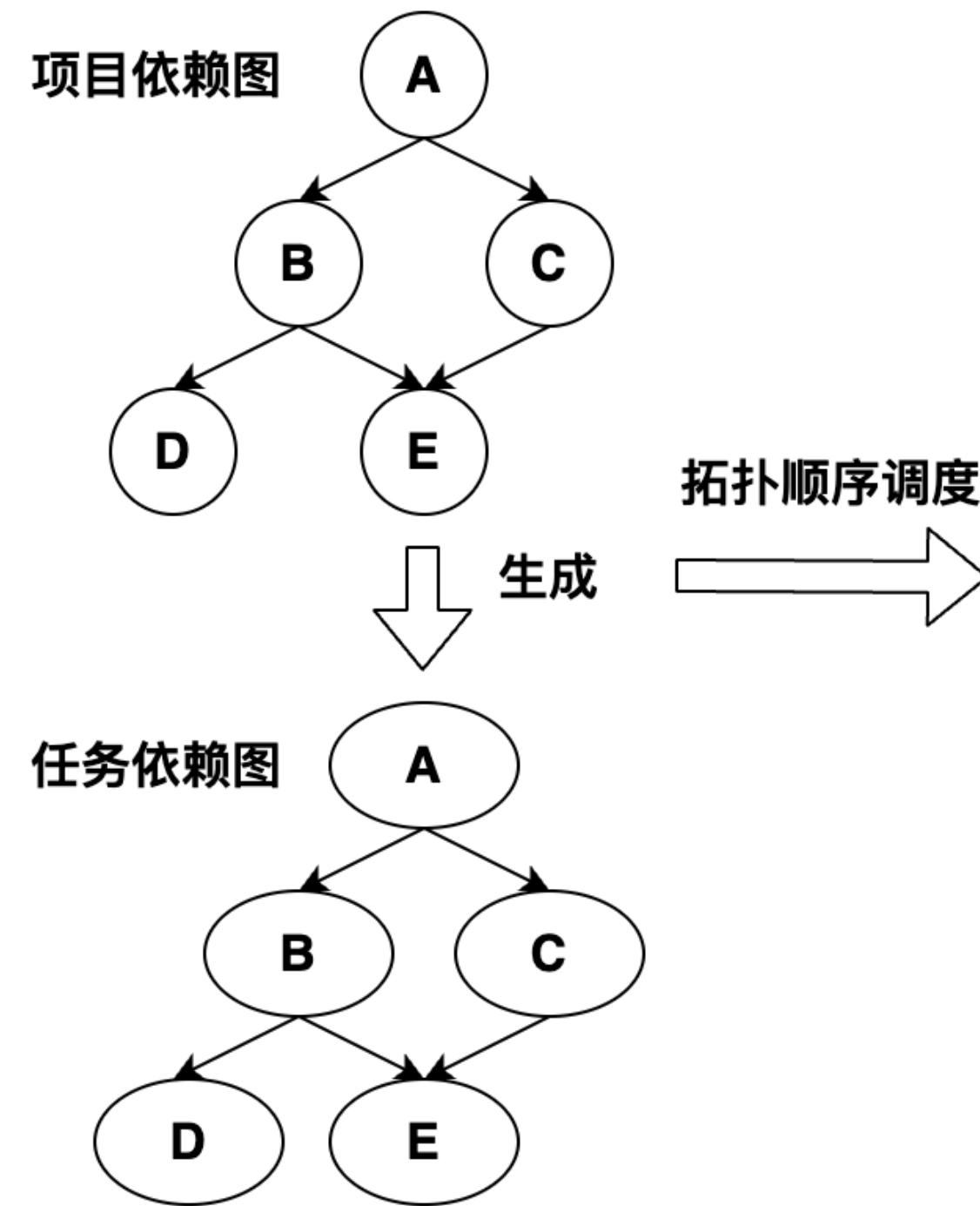
对构建实现了多级缓存



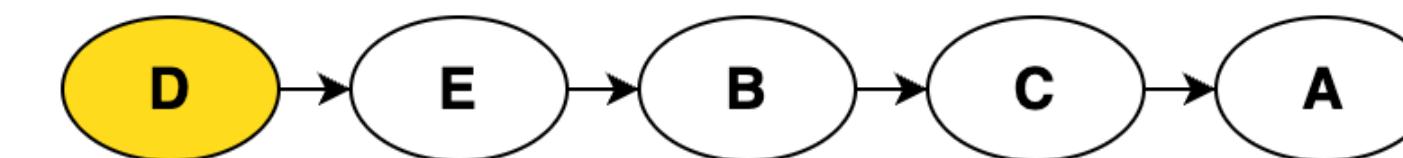
按需构建能力

根据代码更改的影响面来构建

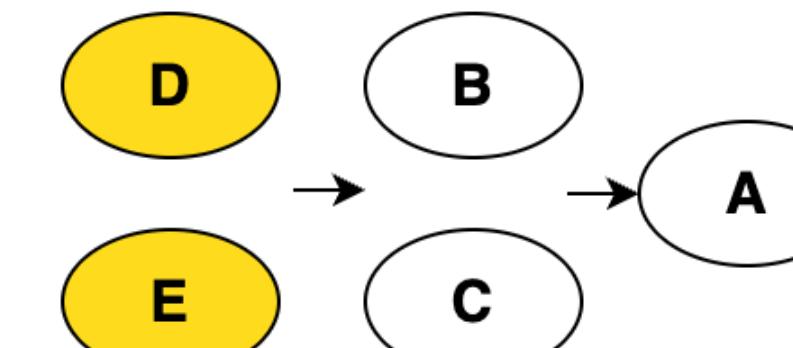
Build System - 任务并行能力



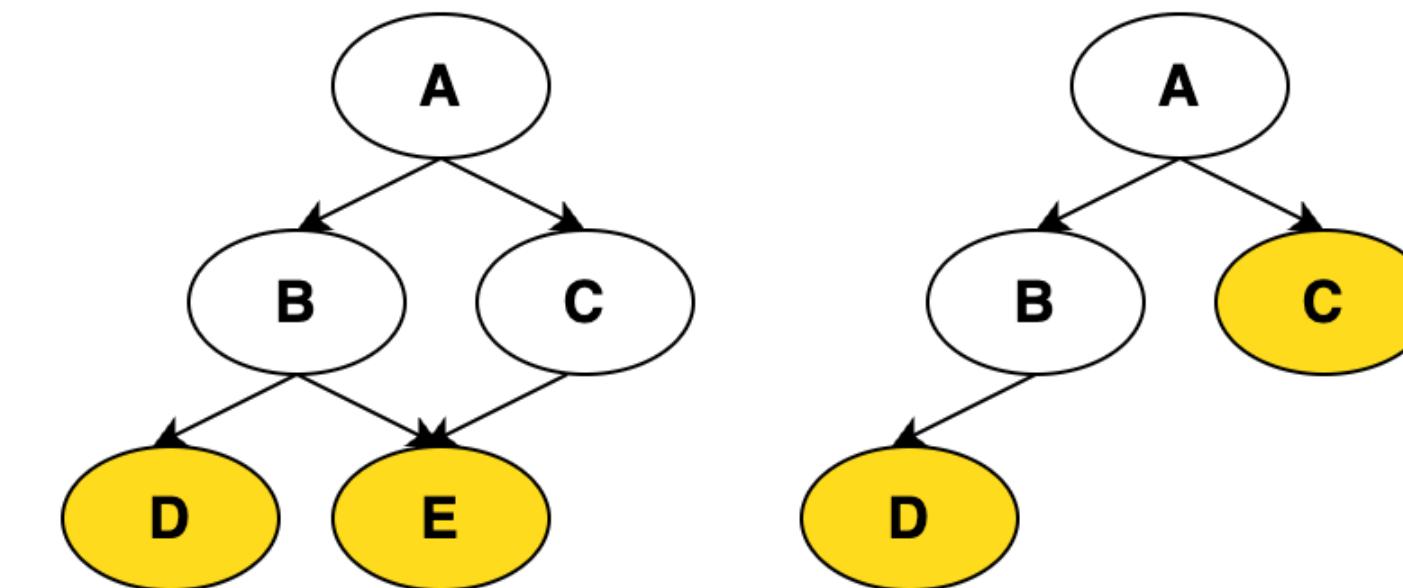
方式1：串行



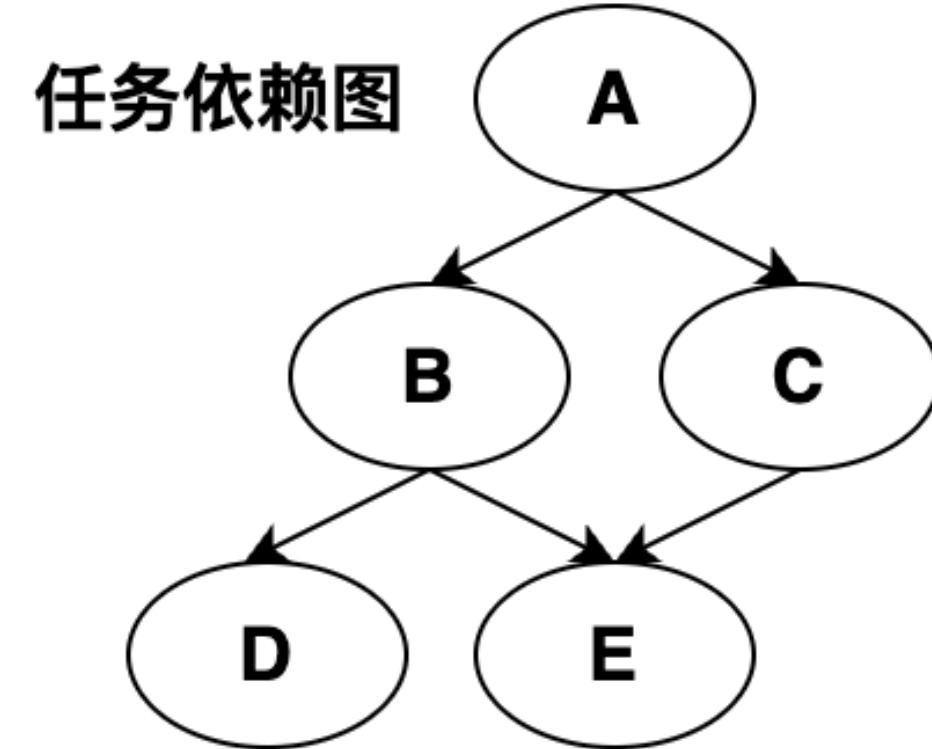
方式2：并行



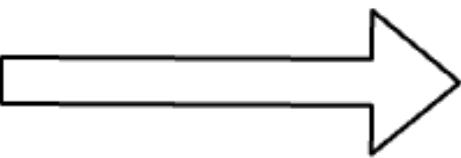
方式3：极致并行



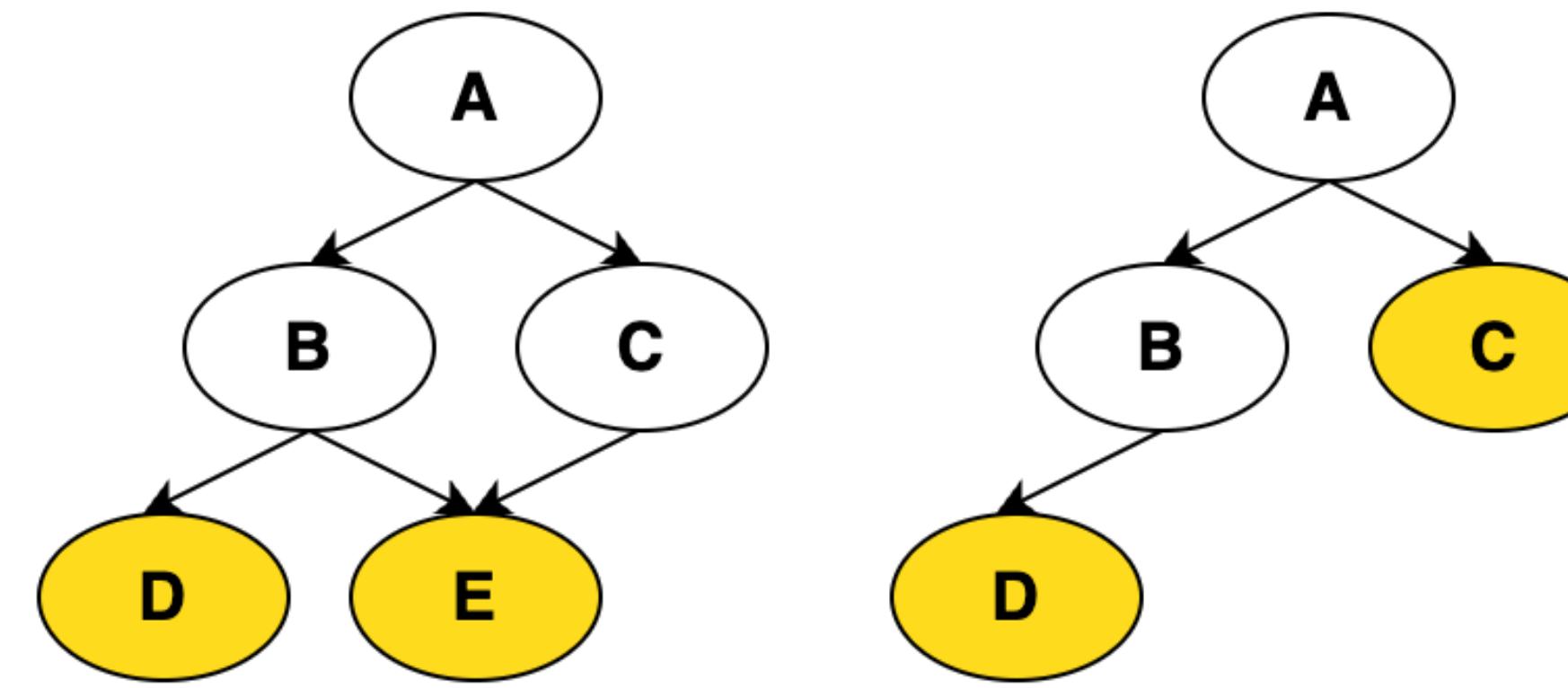
Build System - 多级缓存能力



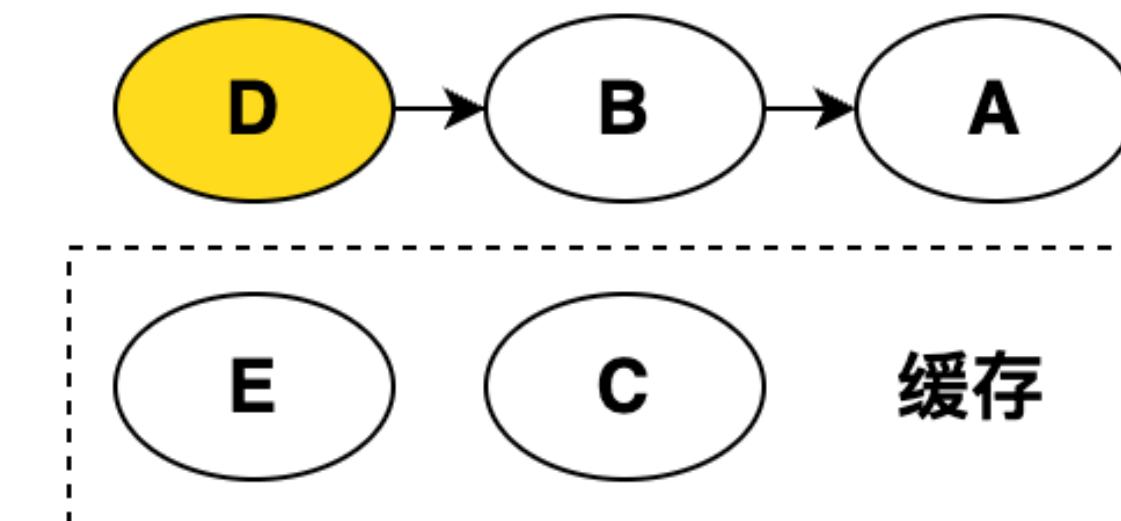
拓扑顺序调度



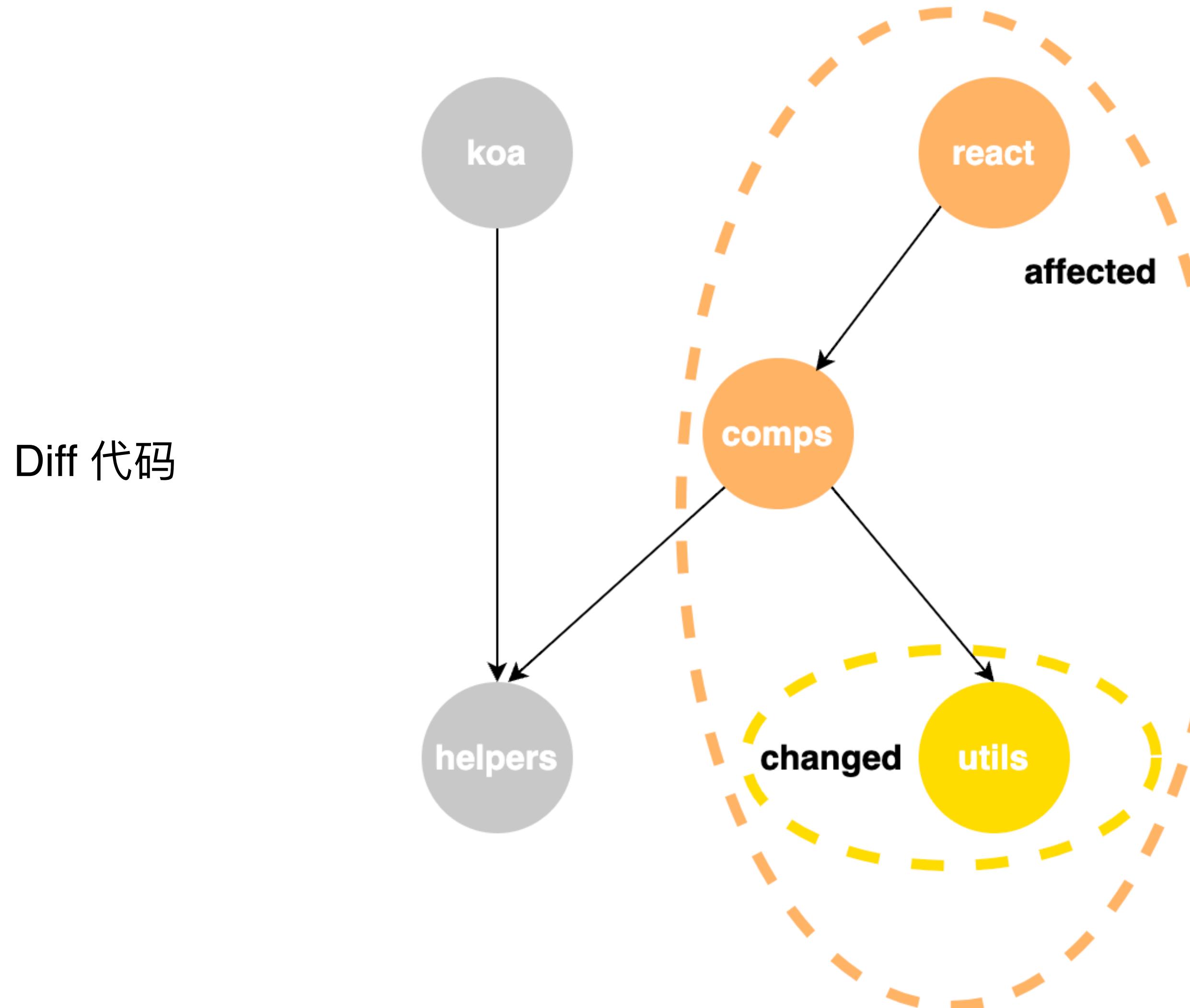
方式3：极致并行



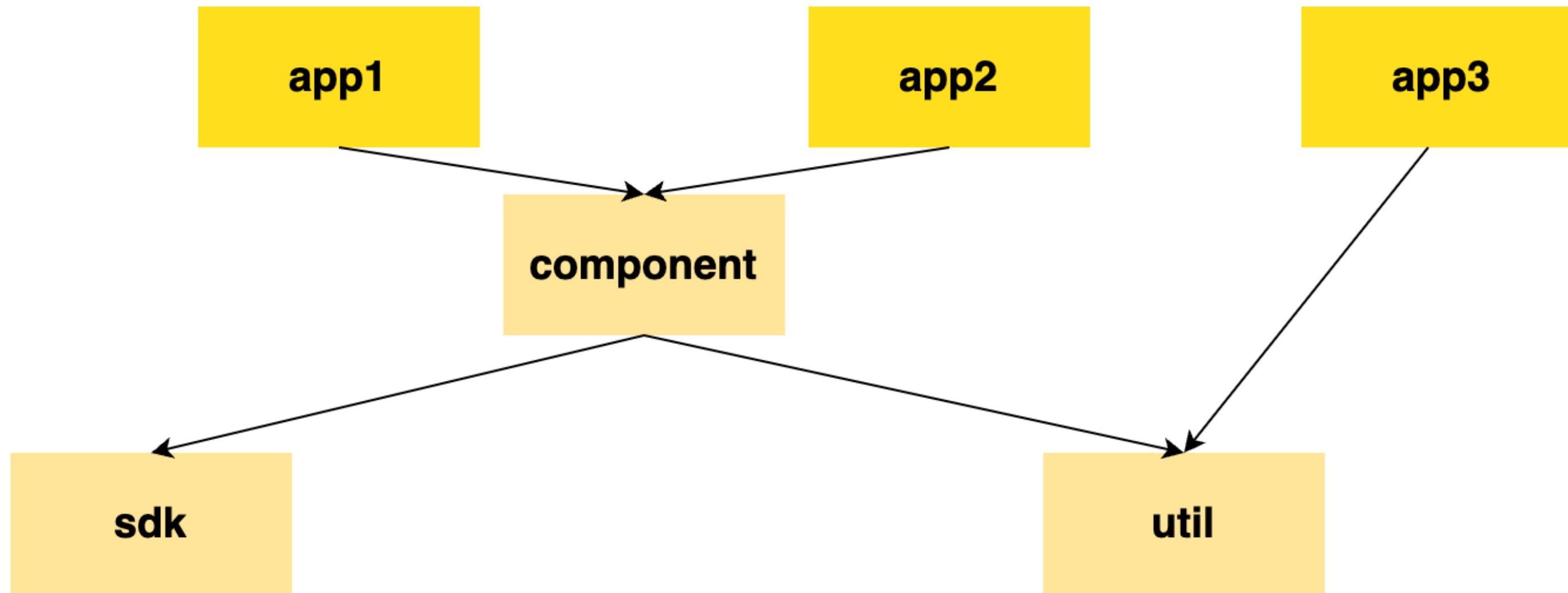
方式4：缓存加速



Build System - 按需构建能力



Build System - 提升效果



	App1	App2	App3
全量		17.72s	
按需		8.94s	
无缓存	16.94s	16.13s	10.77s
有缓存	9.74s	8.94s	7.55s

Bundler & Build System - 总结

拉高应用上限
加快迭代速度



<https://www.rspack.dev>

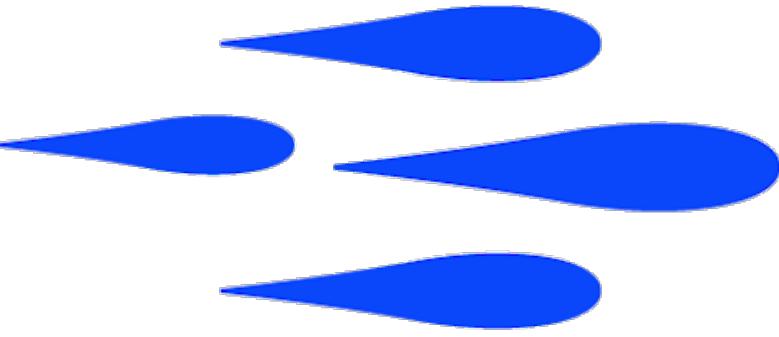
通过建设 Bundler 和 Build System 极大地加速了巨型应用的构建速度

Micro Frontend 简介

前端领域的应用分治解决方案



Qiankun

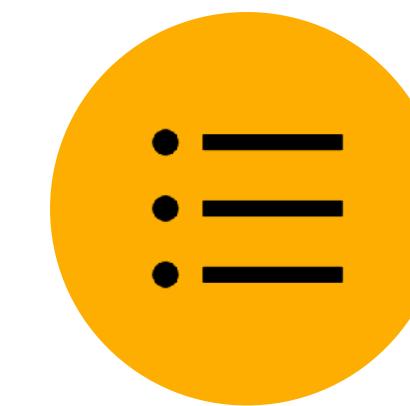


Garfish



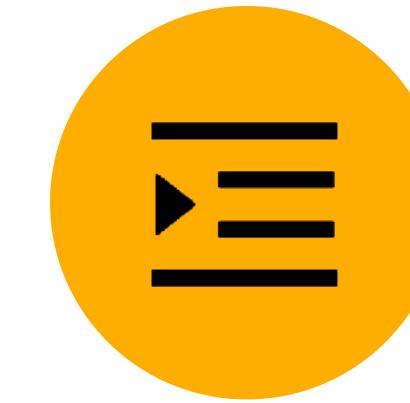
Bit

如何降低多人 开发的协作成 本？



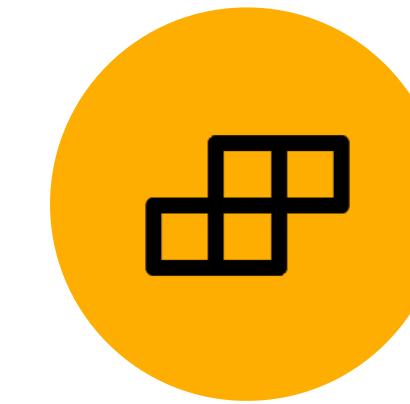
减轻基座负担

基座应用与业务解耦



细粒度的组合

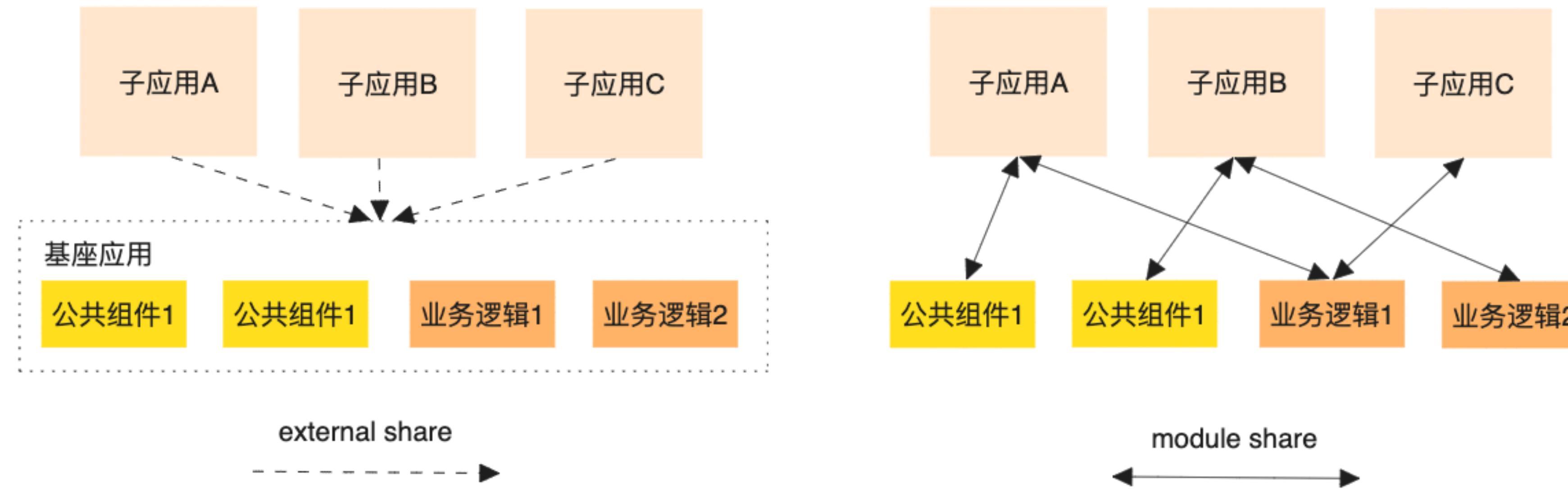
模块级别的独立开发、测试、部署



模块协议标准

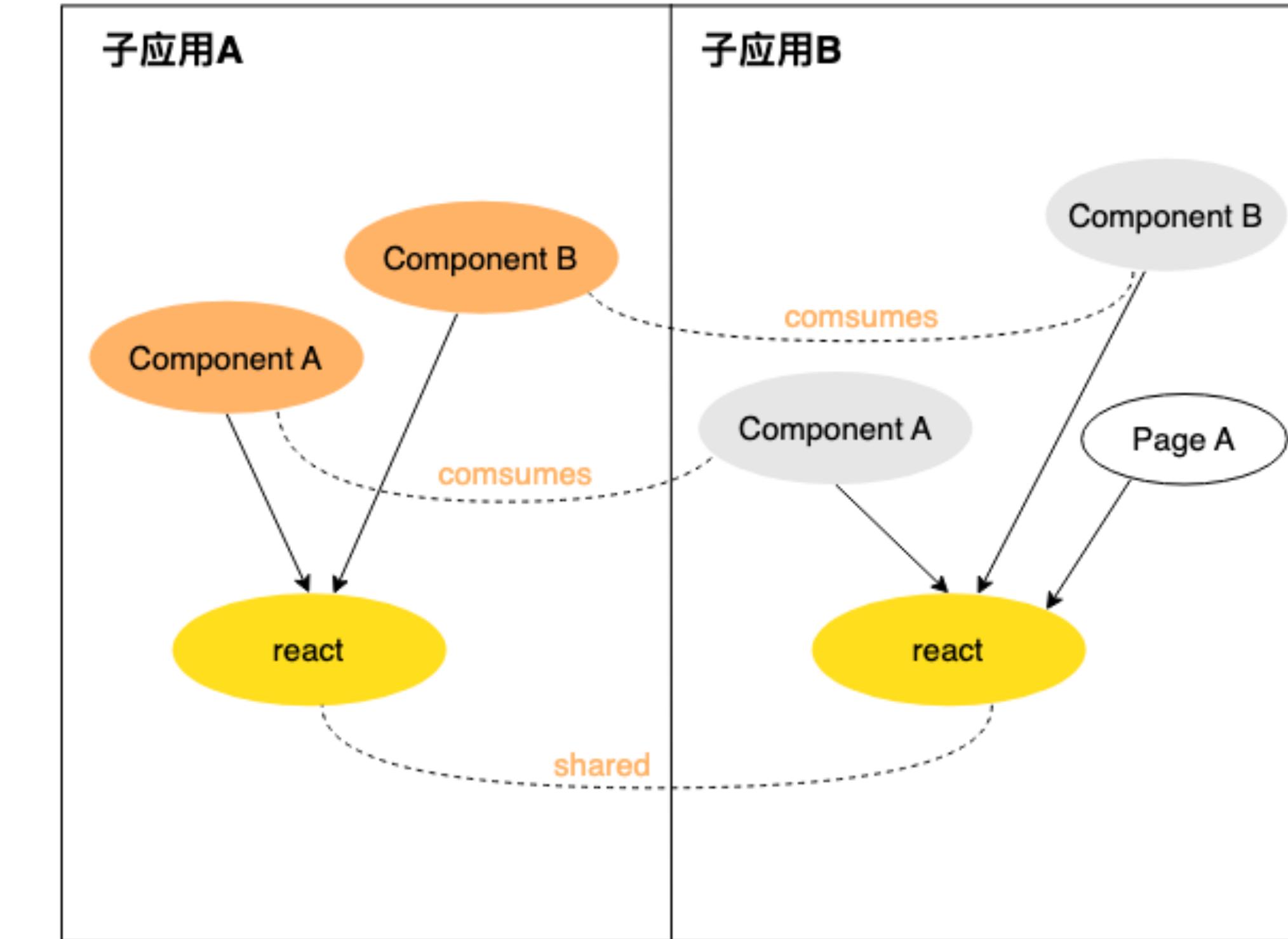
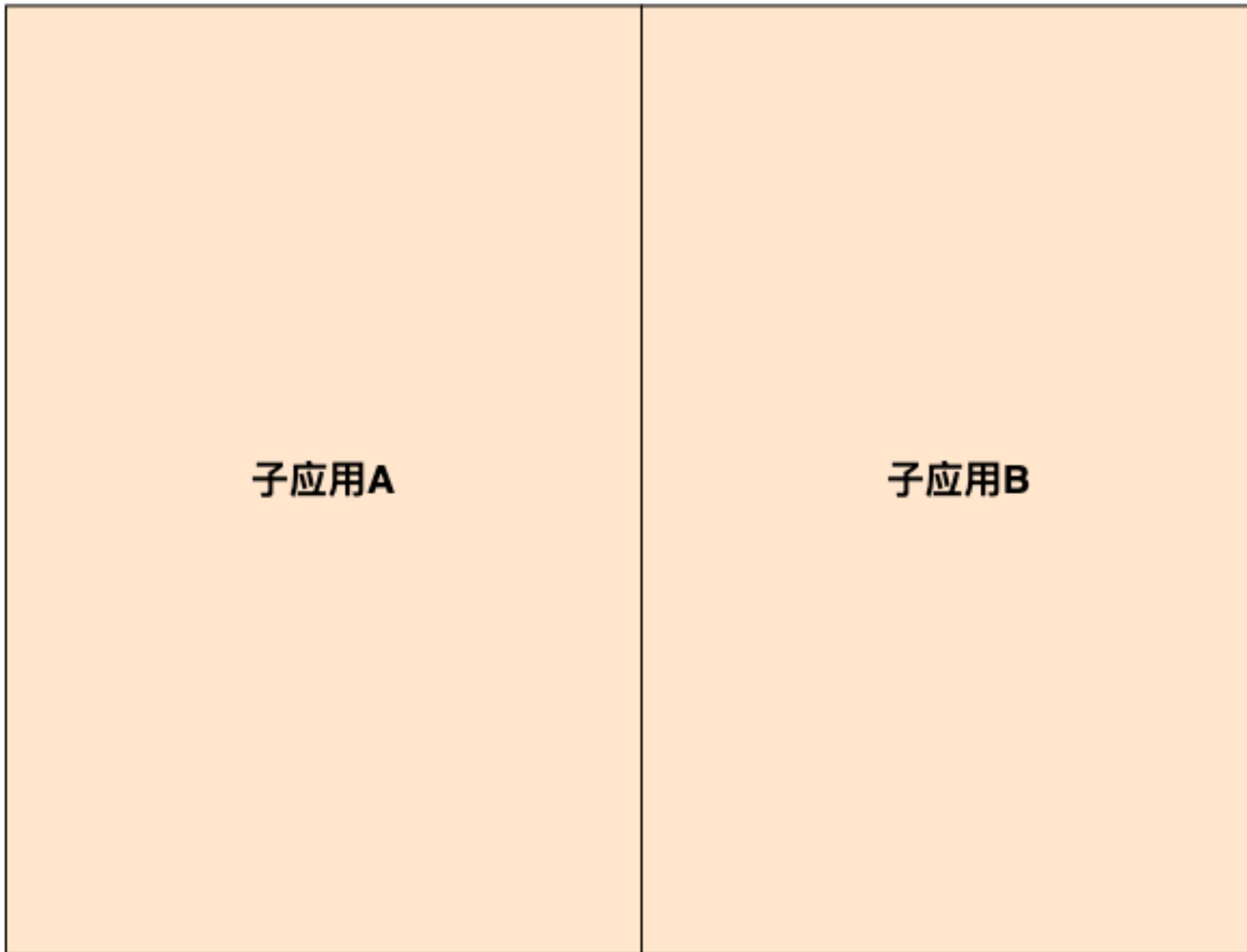
模块中心、结合低码、灰度/AB

Micro Frontend - 减轻基座负担



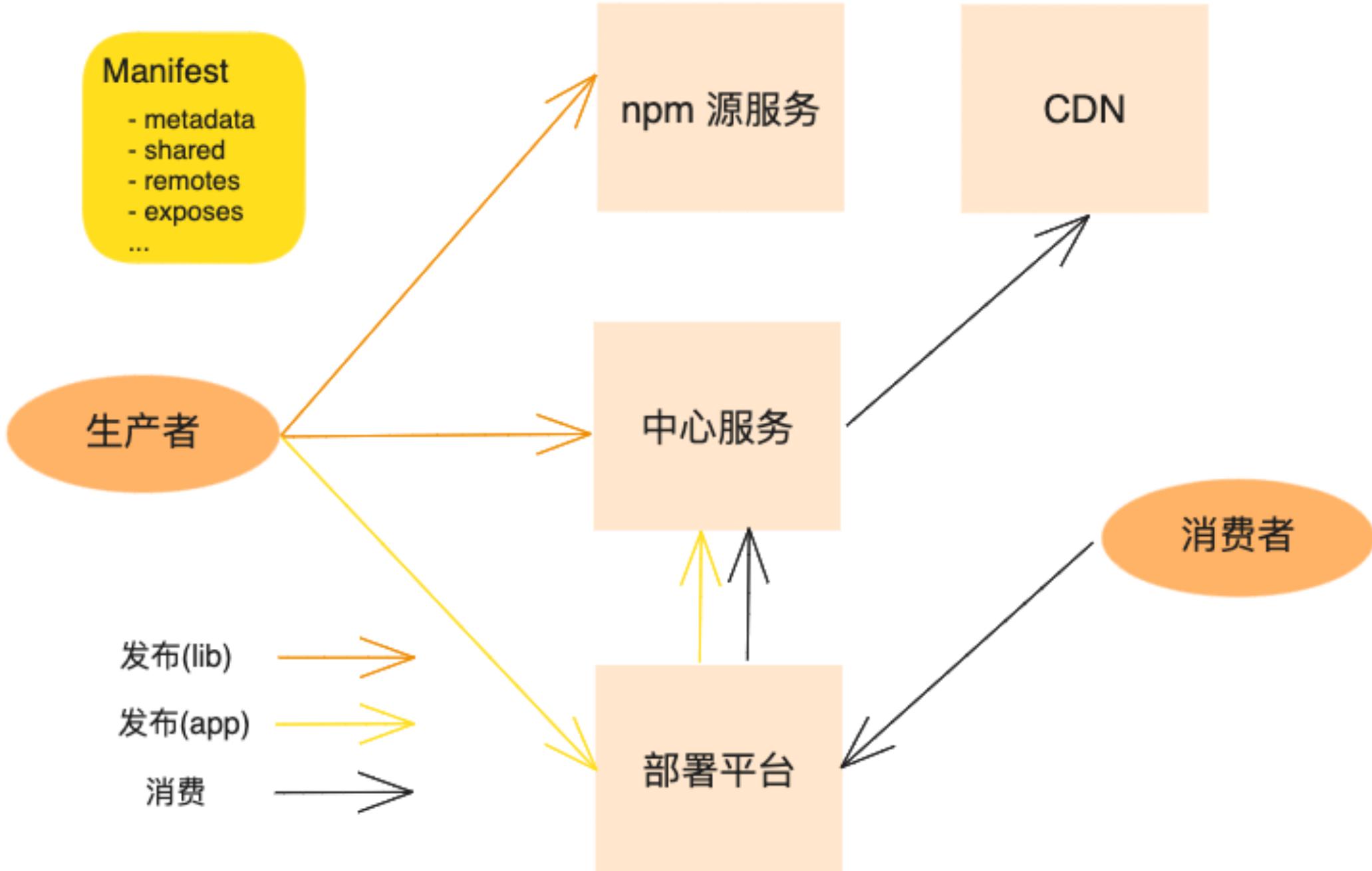
- 业务逻辑耦合到基座
- 故障影响面、失效的缓存
- 造成相互依赖
- 消除公共基座
- 业务逻辑通过消费机制
- 工具库通过共享机制

Micro Frontend - 细粒度的组合



- 沙箱机制保证隔离性
- 制约子应用内外的人员协作
- 提供更细粒度的模块
- 跨子应用消费和共享

Micro Frontend - 模块协议标准



The screenshot shows a module management interface with the following components:

- 模块列表 (Module List)**: A grid of cards showing module details. Each card includes the module name, scope, status, version, and deployment time.
 - benefit-manager-v2: @life-marketing, active, v0.0.428, 2023-05-26 18:00:55
 - freight-subsidy: @life-marketing, active, freight-subsidy, v0.0.128, 2023-05-26 18:00:43
 - benefit-launch-v2: @life-marketing, active, benefit-launch-v2, v0.0.352, 2023-05-26 18:00:42
 - tools-groupmanagement: @arena, tools-groupmanagement, active, v0.1.1, 2023-05-26 17:57:55
- 模块信息 (Module Information) - Detailed View**: Shows information for the selected module **@garfish/micro-app-main**.
 - 基本信息 (Basic Information)**:
 - 模块名称: @garfish/micro-app-main
 - SCM 版本: 1.0.0.2088
 - 发布人: viet.pham
 - 部署来源: goofy
 - 发布日志: (empty)
 - 更新时间: 2023-05-26 16:10:42
 - 依赖 & 模块信息 (Dependencies & Module Information)**:
 - exposes(导出的模块): shared-utils, shared-components, shared-store
 - shared(共享依赖): ./src/shared-utils
 - remotes(消费的模块): ./src/shared-components.tsx
 - consumers(消费者): ./src/store.ts

- 流转于构建平台、部署平台、运行时

Micro Frontend - 总结

	SPA 架构	微前端架构	提升效果
构建耗时	17m43s	8m	50%
部署耗时	10m22s	2m	80%
月均发布	30.6次	77.6次	264%

通过建设 Micro Frontend 降低了多人开发的协作成本

Diagnostics Tool 简介

面向构建过程与构建产物的诊断与分析的一站式工具



Bundle Analyzer

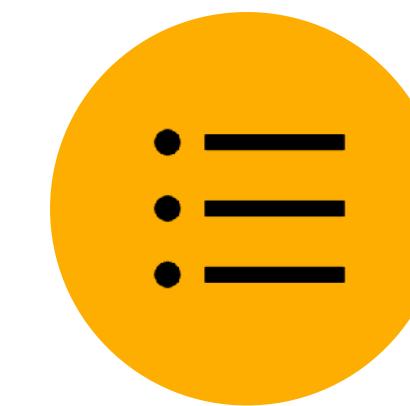


Statoscope



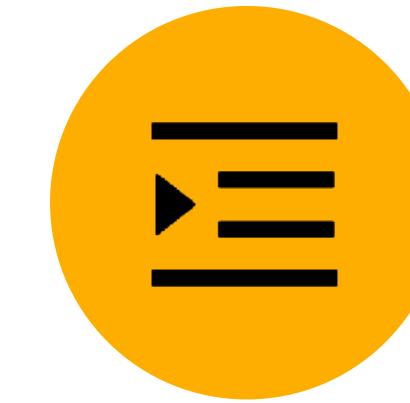
Relative CI

如何有效地防 止应用劣化？



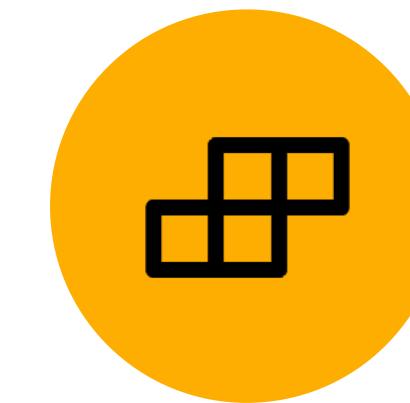
面向构建过程

更细粒度和更丰富的分析能力



可扩展的规则机制

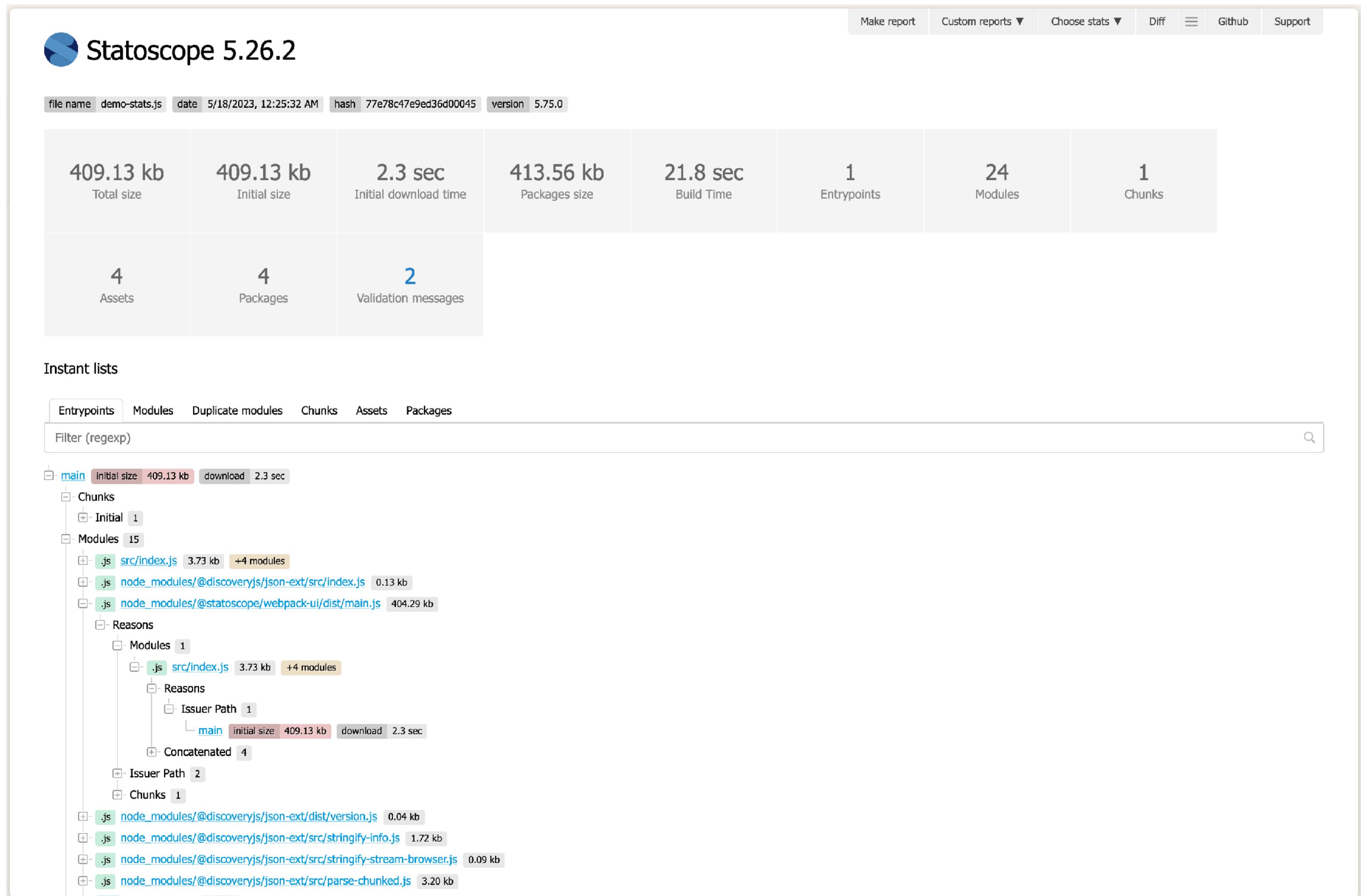
丰富的上下文，强大的扩展能力



与核心研发流程结合

在 CI 中让规则真正发生作用

Diagnostics Tool - 面向构建过程



构建产物 (stats.json)

VS

构建过程

Diagnostics Tool - 面向构建过程



构建产物 (stats.json)

VS

构建过程

Diagnostics Tool - 面向构建过程

The screenshot displays the Diagnostics Tool interface, specifically the "Webpack Loader Overall" section. On the left, there's a detailed tree view of the build process, showing various loaders and their execution times. The main area shows "Executions" and "Loader Details".

Executions:

- 28ms @modern-js/buil... (highlighted)
- 715ms javascript,import "core-js"; @modern-js/buil...
- 164ms javascript,window,__assetPrefix__ = '
- 151ms main.tsx @modern-ts/buil...
- 28ms index.tsx @modern-ts/buil...
- 41ms router.tsx @modern-ts/buil...
- 56ms constants.tsx @modern-ts/buil...
- 2.8ms config.tsx @modern-ts/buil...

Loader Details:

- file path: ./src/index.tsx
- resource path: /opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/index.tsx
- resource query: -
- duration: 28ms
- loader name: @modern-js/builder-shared/compiled/babel-loader
- loader index: 0
- loader path: /opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/eden-mono/temp/node_modules/.pnpm/@modern-js+builder-shared@2.21.1_4asy7drkd73whz3kmjq3plyruij/node_modules/@modern-js/builder-shared/compiled/babel-loader/index.js
- options: {"babelrc":false,"configFile":false,"compact":true,"presets": ["/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/eden-mono/temp/node_modul... more]

The result of [@modern-js/builder-shared/compiled/babel-loader]:

Input	Output
1 // import React from 'react'; 2 // import ReactDOM from 'react-dom/client'; 3- 4 import './i18n'; 5- 6 import App from './main'; 7- 8 import './common/styles/base.scss'; 9 import icon from './common/imgs/icon.svg'; 10- 11 const link = document.createElement('link'); 12 link.setAttribute('type', 'image/x-icon'); 13 link.setAttribute('rel', 'icon'); 14 link.setAttribute('href', icon); 15 document.head.appendChild(link); 16-	1 // import React from 'react'; 2 // import ReactDOM from 'react-dom/client'; 3+ import './i18n'; import App from './main'; import './common/styles/base.scss'; import icon from './common/imgs/icon.svg'; var link=document.createElement('link'); link.setAttribute('type','image/x-icon'); link.setAttribute('rel','icon'); link.setAttribute('href', icon); document.head.appendChild(link); 4+ const root = ReactDOM.createRoot(document.getElementById('root'));

Concatenated: 4 files (0.04 kb, 1.72 kb, 0.09 kb, 3.20 kb)

Issuer Path: 2 chunks (node_modules/@discoveryjs/json-ext/dist/version.js, node_modules/@discoveryjs/json-ext/src/stringify-info.js, node_modules/@discoveryjs/json-ext/src/stringify-stream-browser.js, node_modules/@discoveryjs/json-ext/src/parse-chunked.js)

构建产物 (stats.json)

VS

构建过程

Diagnostics Tool - 面向构建过程

The screenshot displays the Diagnostics Tool interface, specifically the "Webpack Loader Analysis" section. The top navigation bar includes "Make report", "Custom reports", "Choose stats", "Diff", "Github", and "Support". The main area is divided into three main sections:

- Webpack Loader Analysis:** Shows a tree view of files and folders under "data:text". A file named "index.tsx" is selected, highlighted with a blue border.
- Executions:** Displays a "Resolve Diff Viewer" comparing two versions of the code. The left column shows the original code, and the right column shows the modified code. Lines 4 and 6 are highlighted in red, indicating changes. The code snippets are as follows:

Line	Original (Left)	Modified (Right)
1	// import React from 'react';	// import React from 'react';
2	// import ReactDOM from 'react-dom/client';	// import ReactDOM from 'react-dom/client';
3	3-	4+ import "/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/i18n/index.ts";
4	4- import './i18n';	5
5	5- import App from './main';	6+ import App from "/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/main.tsx";
6	6- import './common/styles/base.scss';	7
7	7- import icon from './common/imgs/icon.svg';	8+ import "/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/common/styles/base.scss";
8	8-	9+ import icon from "/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/common/imgs/icon.svg";
9	9-	10
10	10 const link = document.createElement('link');	11 const link = document.createElement('link');
11	11 link.setAttribute('type', 'image/x-icon');	12 link.setAttribute('type', 'image/x-icon');
12	12 link.setAttribute('rel', 'icon');	13 link.setAttribute('rel', 'icon');
13	13 link.setAttribute('href', icon);	14 link.setAttribute('href', icon);
14	14	

- Loader Details:** A table showing resolve details for specific files. The columns are "Source Code", "Duration", and "Resolve Result".

Source Code	Duration	Resolve Result
./main	10ms	/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/main.tsx
./common/styles/base.scss	33ms	/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/common/styles/base.scss
./common/imgs/icon.svg	0.12ms	/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/common/imgs/icon.svg
./i18n	2.2ms	/opt/tiger/compile_path/src/code/byted.org/web-solutions/devtools/packages/client/src/i18n/index.ts

构建产物 (stats.json)

VS

构建过程

Diagnostics Tool - 面向构建过程

The screenshot shows the Diagnostics Tool interface, specifically the "Webpack Plugins Overall" section. The left sidebar displays a tree view of the build process, including "Webpack Loader Overall", "Webpack Loader Analysis", "Webpack Resolver Analysis", and "Webpack Plugins Overall". The main panel shows a table of plugin tap names, their associated hooks, and performance metrics.

Plugin Tap Name	Hook	calls	duration(total)
TerserPlugin	optimizeAssets	1	17.5s
TerserPlugin	processAssets	1	17.5s
WebDoctorWebpackPlugin:moduleGraph	done	1	8.1s
WebDoctorWebpackPlugin:bundle	done	1	2.2s
CssMinimizerPlugin	optimizeAssets	1	943ms
CssMinimizerPlugin	processAssets	1	943ms
RealContentHashPlugin	optimizeAssets	1	854ms
RealContentHashPlugin	processAssets	1	854ms

At the bottom, there is a list of files and their sizes:

- node_modules/@discoveryjs/json-ext/ast/version.js 0.04 kb
- node_modules/@discoveryjs/json-ext/src/stringify-info.js 1.72 kb
- node_modules/@discoveryjs/json-ext/src/stringify-stream-browser.js 0.09 kb
- node_modules/@discoveryjs/json-ext/src/parse-chunked.js 3.20 kb

构建产物 (stats.json)

VS

构建过程

Diagnostics Tool - 面向构建过程

The screenshot shows the Webpack Diagnostics Tool interface. At the top, there are tabs for 'Make report', 'Custom reports ▾', 'Choose stats ▾', 'Diff', 'Github', and 'Support'. Below the tabs, there are sections for 'Webpack Loader Analysis' and 'Webpack Resolver Analysis'. The main area is titled 'Webpack Plugins Overall' and contains several cards: 'Files' (Count: 26, 6.53 MB), 'Total JS / Initial JS' (97.01%, 6.33 MB), 'Total CSS / Initial CSS' (1.69%, 112.64 KB), 'Images / Fonts / Media' (0), 'HTML Files' (0.07%, 4.37 KB), and 'Duplicate Packages' (3). On the left, there is a sidebar with sections like 'Ter', 'Ter', 'We', 'We', 'Cs', 'Cs', 'Re', 'Re', 'Iss', and 'Ch'. The bottom section is titled 'Bundle Analysis' and shows an 'Output Assets List' with 26 assets totaling 6.53 MB. One asset, '130.e564e4eb.js', is highlighted with a size of 1.91 MB. To the right, there is a detailed view of the modules for 'resource/js/130.e564e4eb.js', showing 2708 modules, 1.90 MB total size, and 1 chunk. A search bar and an 'expand all' button are also present.

构建产物 (stats.json)

VS

构建过程

{
configs;
errors;
chunkGraph;
moduleGraph;
packageGraph;
loader;
resolver;
plugin;
code;
sourceMap;
...
}

Diagnostics Tool - 可扩展的规则机制

Bundle Alerts

- E1001 Duplicate Packages tslib 2 versions was found total size 26.44 KB
 - tslib v2.4.1 source size 12.02 KB /opt/tiger/compile_path/src/code.byted.org/web-solutions/devtools/.eden-mono/temp/node_modules/.pnpm/tslib@2.4.1/node_modules/tslib
 - tslib v2.5.0 source size 14.42 KB /opt/tiger/compile_path/src/code.byted.org/web-solutions/devtools/.eden-mono/temp/node_modules/.pnpm/tslib@2.5.0/node_modules/tslib
- E1001 Duplicate Packages @babel/runtime 3 versions was found total size 75.89 KB
 - @babel/runtime v7.20.13 source size 24.20 KB /opt/tiger/compile_path/src/code.byted.org/web-solutions/devtools/.eden-mono/temp/node_modules/.pnpm/@babel+runtime@7.20.13/node_modules/@babel/runtime
 - @babel/runtime v7.21.0 source size 25.09 KB /opt/tiger/compile_path/src/code.byted.org/web-solutions/devtools/.eden-mono/temp/node_modules/.pnpm/@babel+runtime@7.21.0/node_modules/@babel/runtime
 - @babel/runtime v7.22.3 source size 26.59 KB /opt/tiger/compile_path/src/code.byted.org/web-solutions/devtools/.eden-mono/temp/node_modules/.pnpm/@babel+runtime@7.22.3/node_modules/@babel/runtime
- E1001 Duplicate Packages react-is 2 versions was found total size 5.09 KB
 - react-is v16.13.1 source size 2.69 KB /opt/tiger/compile_path/src/code.byted.org/web-solutions/devtools/.eden-mono/temp/node_modules/.pnpm/react-is@16.13.1/node_modules/react-is
 - react-is v18.2.0 source size 2.40 KB /opt/tiger/compile_path/src/code.byted.org/web-solutions/devtools/.eden-mono/temp/node_modules/.pnpm/react-is@18.2.0/node_modules/react-is

默认规则

- Duplicate Packages
- Default Import Check
- Loader Performance Optimization

规则上下文

```
{  
  configs;  
  errors;  
  chunkGraph;  
  moduleGraph;  
  packageGraph;  
  codeAST;  
  ...  
}
```

扩展规则

- 依赖的引用方式检查
- 特定依赖的版本检查
- 禁止使用特定的语句

Diagnostics Tool - 与核心研发流程结合

[CI] Report: Bundle Diff examples/dist/manifest.json Bundle Size Diff Results: 1.96%

Bundle Diff 点击查看详情

Property	Size	Diff	BaseLine
ALL TOTAL Size	474.17 KB	1.96% — 9.13 KB	465.04 KB
JS TOTAL Size	463.27 KB	2.01% — 9.13 KB	454.15 KB
JS Initial Size	463.27 KB	2.01% — 9.13 KB	454.15 KB
CSS TOTAL Size	0 bytes	0% — 0 bytes	0 bytes
CSS Initial Size	0 bytes	0% — 0 bytes	0 bytes
IMGS TOTAL Size	0 bytes	0% — 0 bytes	0 bytes
HTML TOTAL Size	2.45 KB	0% — 0 bytes	2.45 KB
MEDIA TOTAL Size	0 bytes	0% — 0 bytes	0 bytes
FONTS TOTAL Size	0 bytes	0% — 0 bytes	0 bytes
OTHERS TOTAL Size	8.45 KB	0% — 0 bytes	8.45 KB

Web Doctor

Bundle Size

Current: 474.17 KB +1.96%	Baseline: 465.04 KB
---------------------------	---------------------

Total JS +2.01% Initial JS +2.01%

Current: 463.27 KB +2.01%	Baseline: 454.15 KB
---------------------------	---------------------

Total CSS Initial CSS

Current: 0 bytes	Baseline: 0 bytes
------------------	-------------------

Images Fonts Media

Current: 0 bytes	Baseline: 0 bytes
------------------	-------------------

HTML

Current: 2.45 KB	Baseline: 2.45 KB
------------------	-------------------

Others

Current: 8.45 KB	Baseline: 8.45 KB
------------------	-------------------

Duplicate Packages

Current: 6	Baseline: 6
------------	-------------

Modules

Current: 785 +2.88%	Baseline: 763
---------------------	---------------

Packages New 0 Deleted 0 Changed 1

Current: 17	Baseline: 17
-------------	--------------

Overview Assets Modules Packages

Search by name Filter by changed type

Modules	1491/1491 ⓘ	Current: 785 +2.88% ⓘ	Baseline: 763 ⓘ	in Assets	Actions			
		Source Size	Parsed Size	Source Size	Parsed Size			
entities/lib/generated/dec	ode-data-html.js ⓘ	Not Changed node_modules	96.71 KB	78.84 KB	96.71 KB	78.84 KB	Current: static/js/399.033fd63e.js Baseline: static/js/399.8a1adf9b.js	Current Result Viewer
core-	js/modules/web.url.constructor.js ⓘ	New node_modules	34.84 KB +100%	11.90 KB +100%	-	-	Current: static/js/lib-polyfill.affc70ed.js	Current Result Viewer
entities/lib/maps/entities.j	son ⓘ	Not Changed node_modules	32.20 KB	39.50 KB	32.20 KB	39.50 KB	Current: static/js/399.033fd63e.js Baseline: static/js/399.8a1adf9b.js	Current Result Viewer
htmlparser2/lib/Tokenizer.j	s ⓘ	Not Changed node_modules	31.53 KB	12.90 KB	31.53 KB	12.90 KB	Current: static/js/399.033fd63e.js Baseline: static/js/399.8a1adf9b.js	Current Result Viewer
core-js/modules/ush_structured_	...							

Diagnostics Tool - 总结

业务典型收益

-  发现异常 Loader 耗时，减少 50% 的编译耗时
-  构建产物体积减少 14%，首屏加载速度提升 39%
-  推动三方包升级，重复包从 14 个减少到了 6 个
-  多次拦截包体积异常增大的 Merge Request
-  内部移动端框架自定义了典型的诊断规则 6 条

通过建设 Diagnostics Tool 有效地防止了应用劣化

整体落地情况

接入工程
 x个

Monorepo

降低多项目的维护成本

月活用户
 y个

Micro Frontend

降低多人开发的协作成本

GitHub Star
 z个

Bundler & Build System

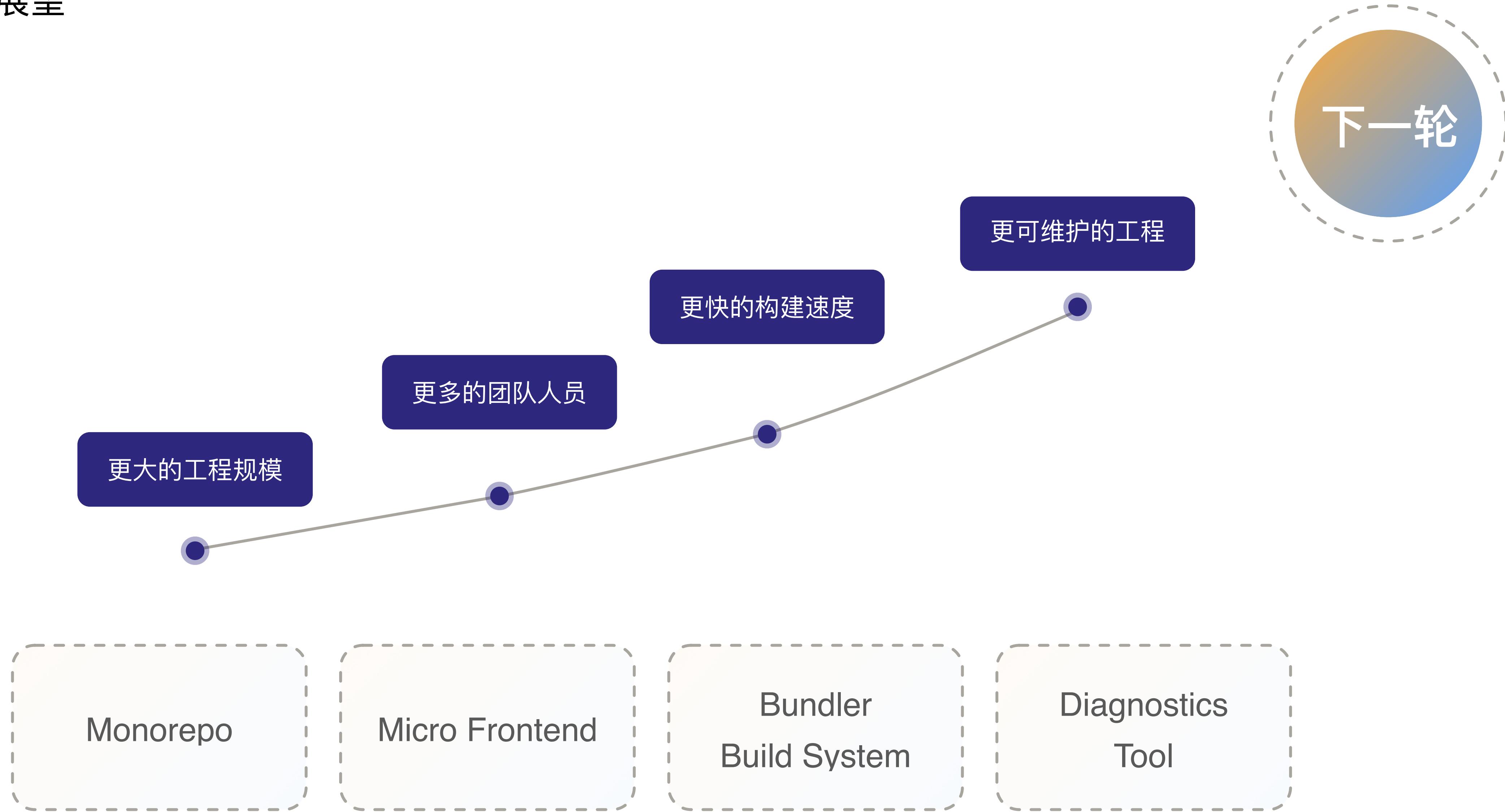
加快巨型应用的构建速度

周下载量
 n次

Diagnostics Tool

有效地防止应用劣化

总结与展望



谢 谢 观 看

thanks