

褚煦露 Chu Xulu

4+ 前端工作经验（抖音、科大讯飞）

北方工业大学（机械）GPA: 3.68/4.0





褚煦露

FRONT-END 前端工程师

前端基础 / CSS 工具



HTML 5



CSS 3



JavaScript



TypeScript



Tailwind CSS

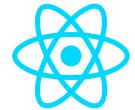


LESS



Sass

JavaScript 框架 & 库



React



Vue



Lodash



jQuery



Element UI



Ant Design

前端模块化



Webpack



Gulp.js



Rollup.js



Babel.js



npm



yarn

后端技术



Node.js



Next.js



Express.js



koa



MySQL



Docker



Nginx





- 1 抖音 - 项目经历
电子牵、乐小活、FollowMe
- 6 工程化思维
注重文档构建和输出，减少沟通成本
- 2 抖音 - 技术
前端架构
- 7 抖音 - FOLLOW ME
担任后端开发工程师
- 3 研发流程的进化
物料包 -> 微前端 -> Monorepo
- 8 科大讯飞
讯飞新高考门户
- 4 研发流程的规范
gitflow、CI CD
- 9 CONTRIBUTING TO DEPAUL
As a Graduate Student
- 5 提效工具
工具的解耦和统一管理

目录





用科技助力灵活就业

赋能用工平台、税酬机构、自由职业者。

[Website](#)





电子牵

合同签约

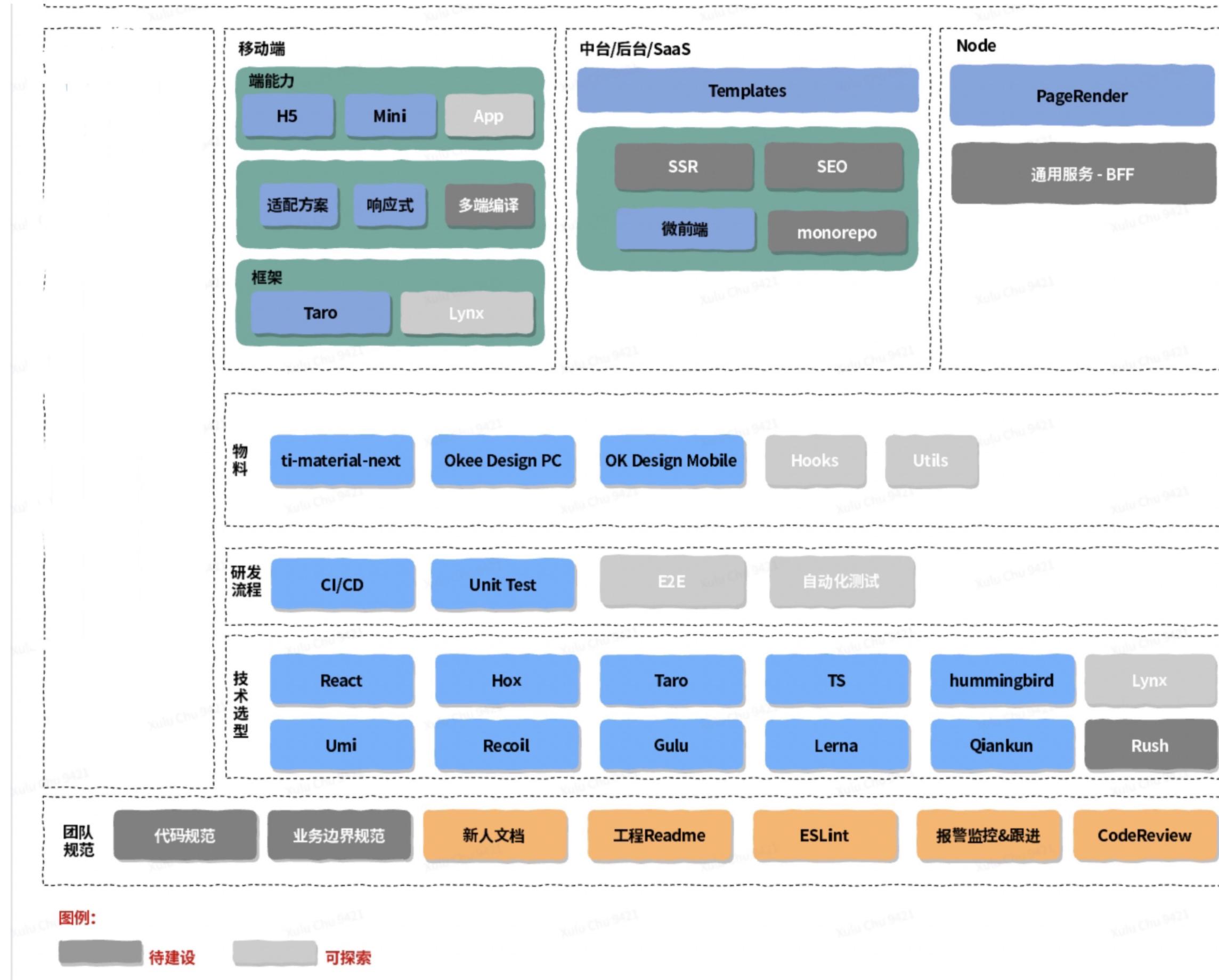
面向企业和个人的一站式文件签署&管理系统平台

[Website](#)





前端架构





研发流程的进化

物料包 -> 微前端 -> MONOREPO

背景

- 在多个项目中很多重复的代码片段...

✓ 解决方案: 物料包

- 使用 **lerna** 与 **dumi** 建立起 PC、移动端的物料包。

```
material-package/
├── package.json          # 根 package.json 文件
├── dist/                  # 打包生成的组件
├── mock/                 # Mock 数据
├── src/
│   ├── components/        # 组件
│   │   ├── Button/         # Button 组件
│   │   │   ├── index.tsx
│   │   │   └── .storybook/
│   │   └── Checkbox
│   ├── hooks/              # Hook 函数
│   ├── utils/              # 工具函数
│   ├── styles/             # 样式文件
│   └── index.ts            # 导出文件
└── .umirc.ts              # Dumi 配置
└── test/                  # 单元测试
└── jest.config.js         # Jest 配置
└── package.json           # 子 package.json 文件
```





研发流程的进化

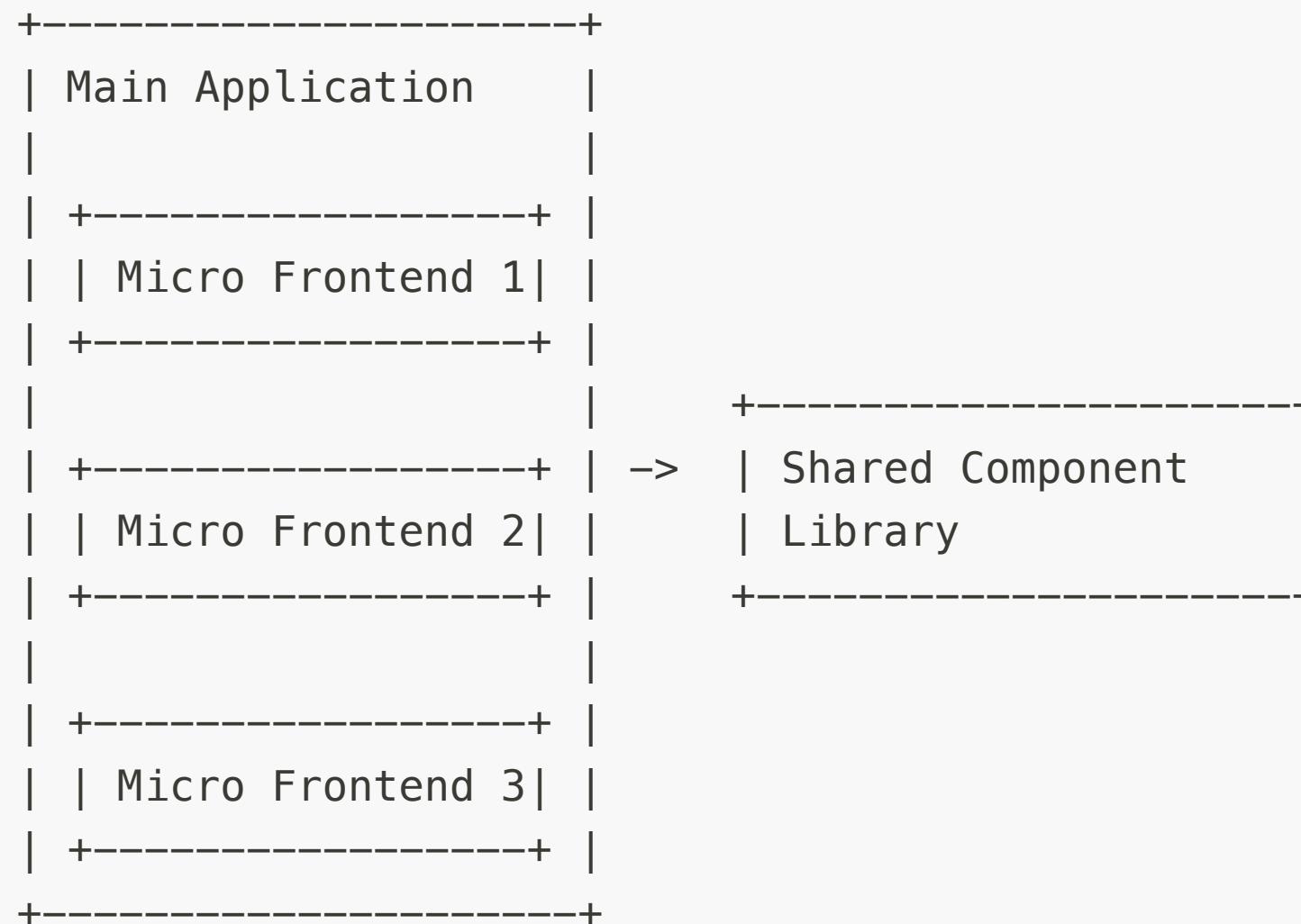
物料包 -> 微前端 -> MONOREPO

背景

- 我们的仓库，变得越来越庞大，越来越难以维护，成为“巨石项目”。

✓ 解决方案: 微前端

- 使用 qiankun，把一个项目拆分成不同的模块，模块与模块之间互相独立，每个模块一个 git 仓库。不同的模块交给不同的团队维护，减轻每个程序员的负担。





研发流程的进化

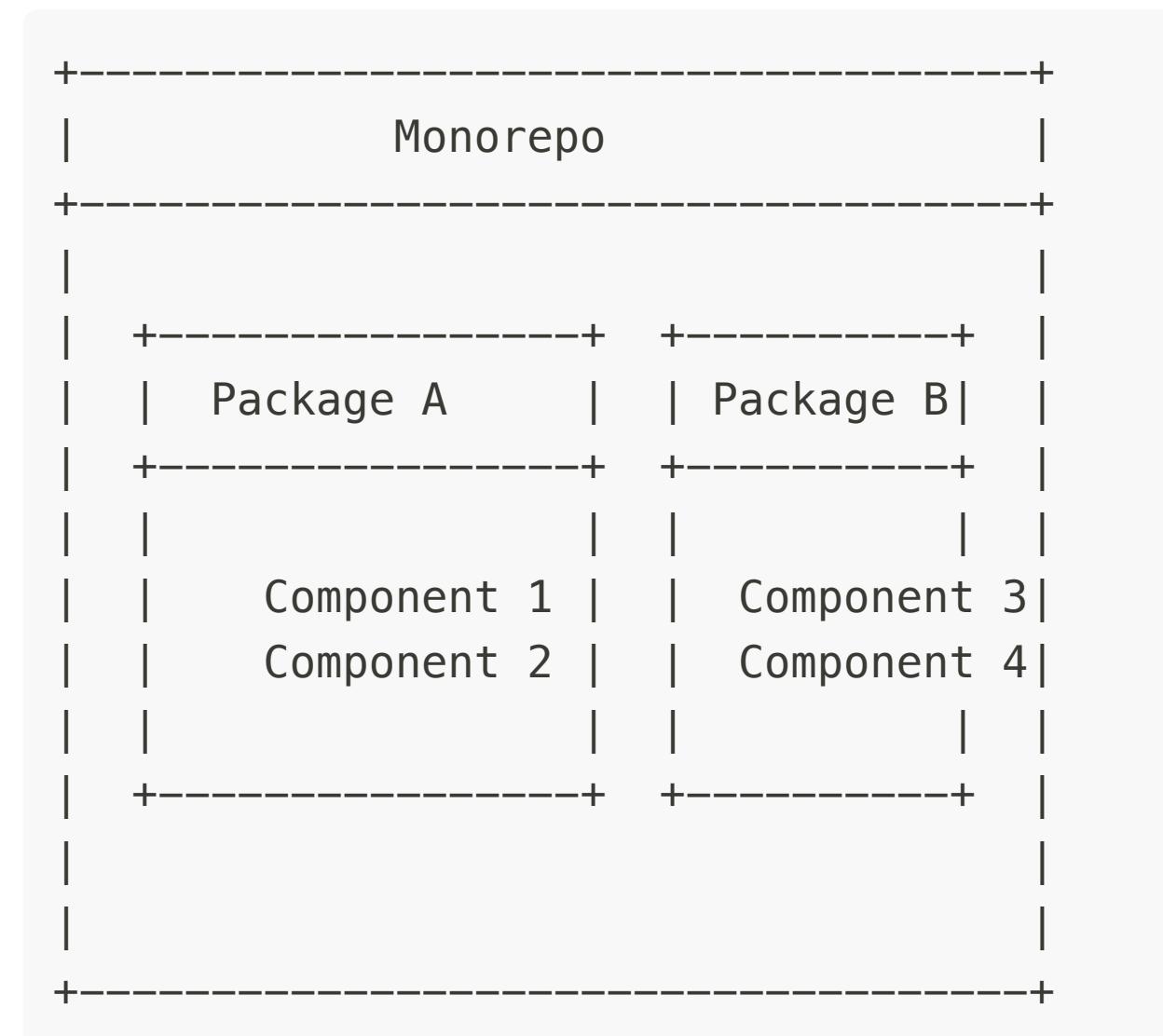
物料包 -> 微前端 -> MONOREPO

背景

- 随着微前端在组内的应用，项目中 **git 仓库**越来越多。
- 随着物料包的增大，使用**基础库的层级越来越深**。如果想修改一个基础库的属性，往往需要修改多个仓库，并进行多次发包。

✓ 解决方案: 迁移到 Monorepo

- 使用 **Rush.js** 把各个分散的 git 仓库聚合在一起。
- 方便变更与升级、提高研发效率





规范研发流程：

GITFLOW、CI CD...

背景

- 随着团队的不断扩大，每个小组有了一套自己的 Workflow，一些研发流程未标准化。

✓ 解决方案

- 如今，我们已经将很多流程标准化，并集成到 **CI/CD** 中，让开发者关注更少的细节。
 1. 物料包的 publish，交给 pipeline，而不是占用本地资源发包。
 2. 业务工程接入 pipeline，**lint / ut / e2e** 等环节由 pipeline 集群完成校验。

Code Review → 物料包发布 → Lint/UT/E2E → Build → 小流量发布 → 全流量发布



工具的解耦和统一管理：

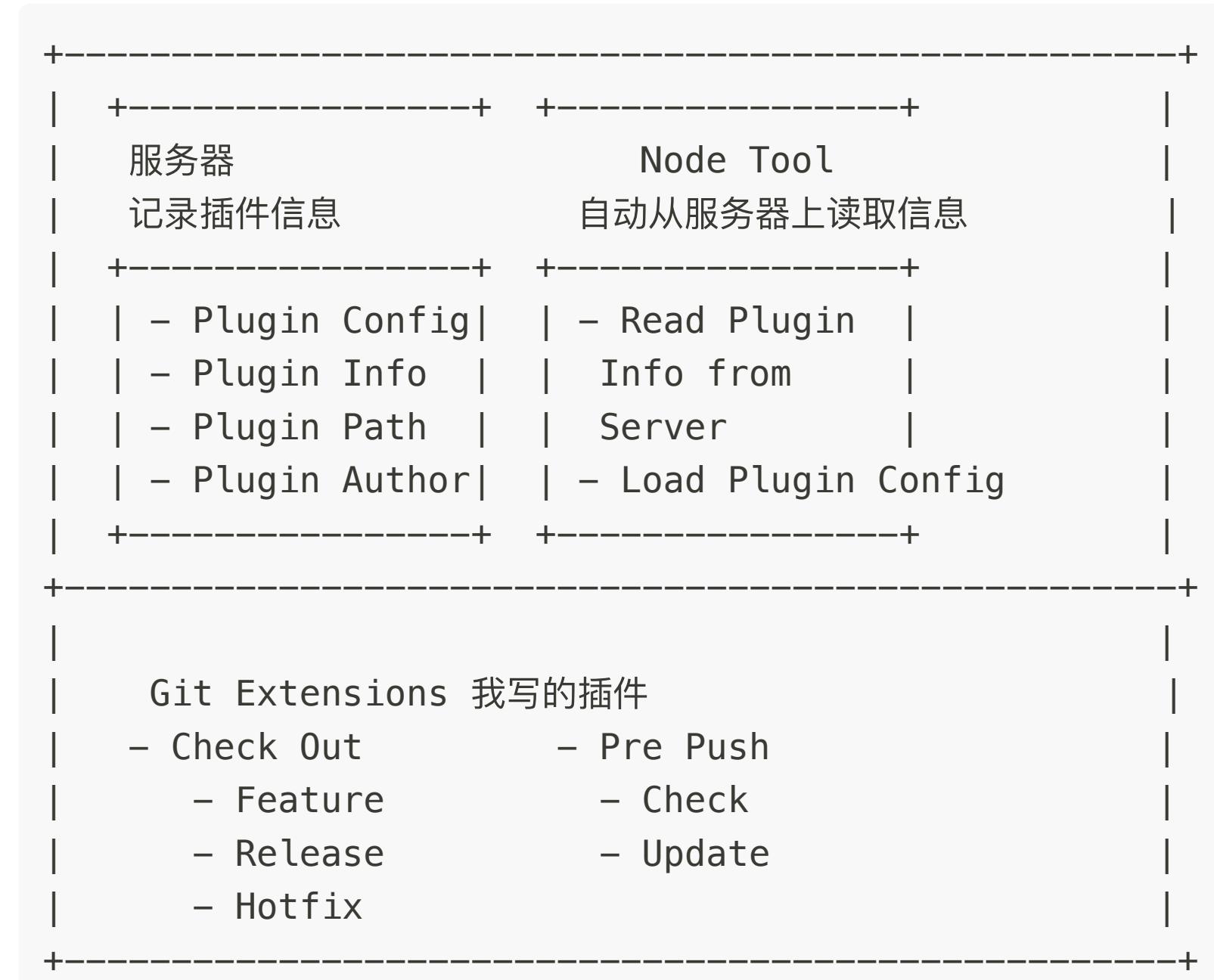
SHELL

背景

- 有一些不影响具体业务的工具比较分散，我们需要一个地方统一管理它们。

解决方案

- 为了解决这个问题，我们可以在一台服务器上集中管理这些工具。





工程化思维，注重文档构建和输出，减少沟通成本：

README

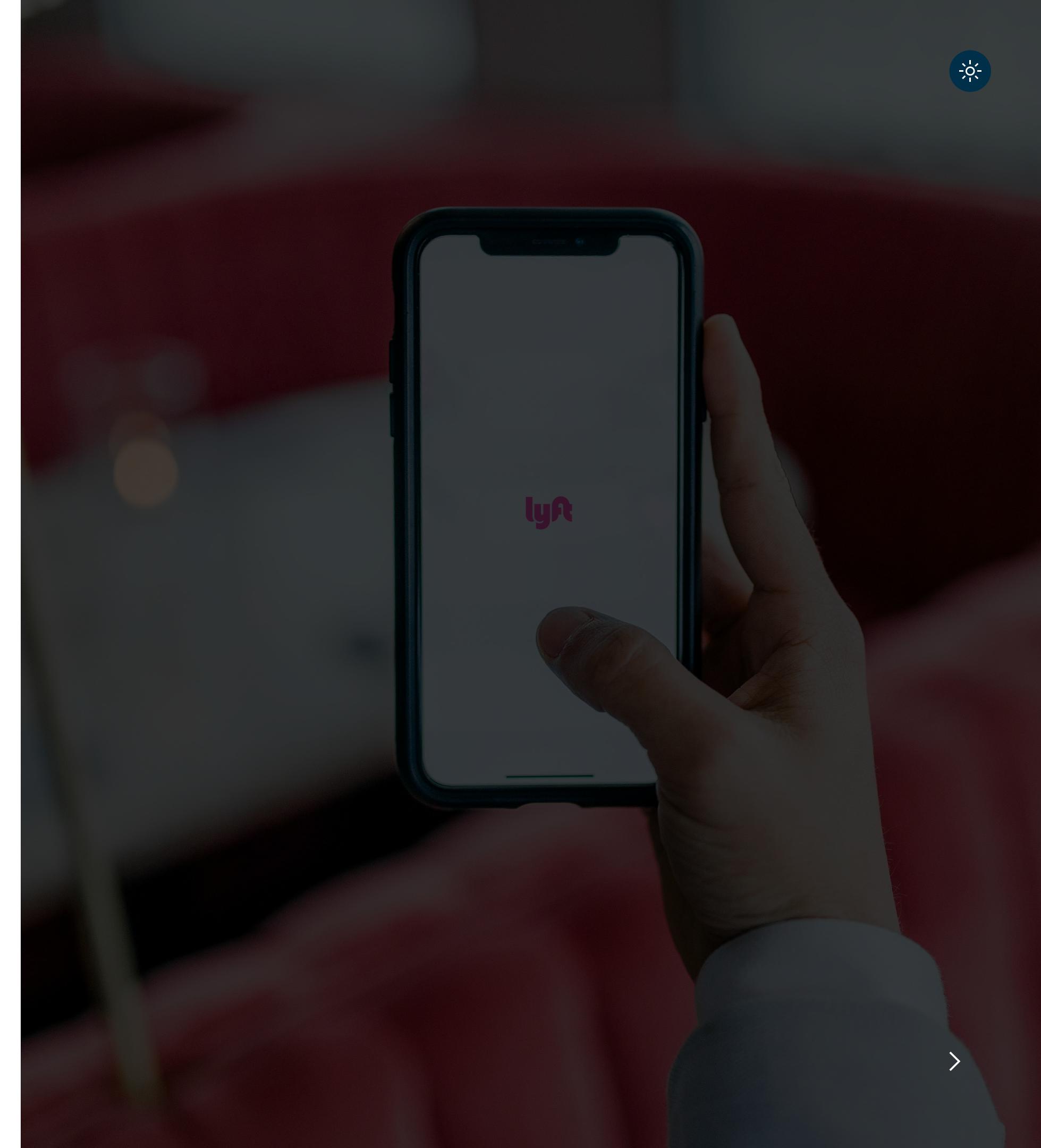
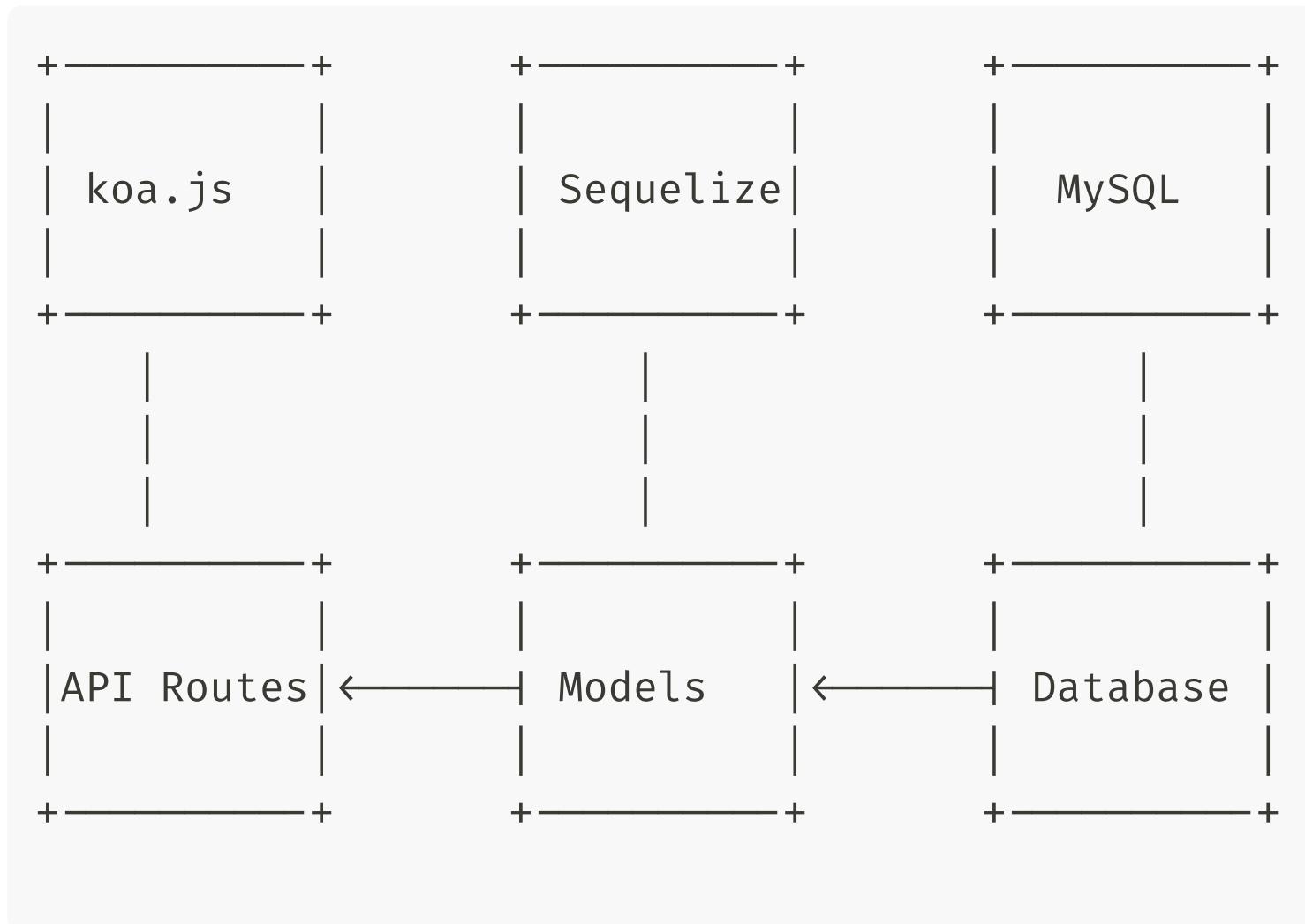
- 提供新人 onboarding 文档
- 结构化 readme 文件
 - 减少沟通：新同事先查看 readme 解决疑问
 - 提取 readme 模版：避免重复，有需要直接 copy 模板并修改相应内容。
- 总结小程序开发经验：记录问题与解决方案，帮助新人避坑。
- 分享项目经验：被同事评价为，见过最详细、有用的业务分享。





抖音 Follow Me

让非开发人员也可轻松测量和提高您产品的在线使用率。





讯飞新高考门户

高考服务平台，包括智能排班、智能排考。



Vue



Webpack



Gulp.js





1. React

- React Hooks API: <https://juejin.cn/post/6847902224144662541>
- useState 原理: <https://juejin.cn/post/6847902217475735560>
- react + Typescript 的记账系统: <https://github.com/Maricaya/h5-accounting>

2. Node.js 框架和库、RESTful API

- 使用 Nextjs + TypeORM + TypeScript 开发的博客系统: <https://github.com/Maricaya/nextjs-blog>

3. npm yarn webpack gulp

- webpack 性能优化: https://maricaya.github.io/interview_blog/webpackInfo/compile.html#webpack-构建优化
- 使用 webpack4.0 从零搭建 React + TypeScript 应用: <https://juejin.cn/post/6850418115189309447>
- Create React App 中配置 CSS (css normalize、scss、styled-components): <https://juejin.cn/post/6850418120386052103>
- 使用 svg-sprite-loader、svgo-loader 优化项目中的 Icon: <https://juejin.cn/post/6854573215646875655>
- 如何编写一个 Vite 插件: <https://juejin.cn/post/6876812524338216973>

4. CSS

- Tailwind CSS: <https://www.zhihu.com/question/506073568/answer/293472301>

