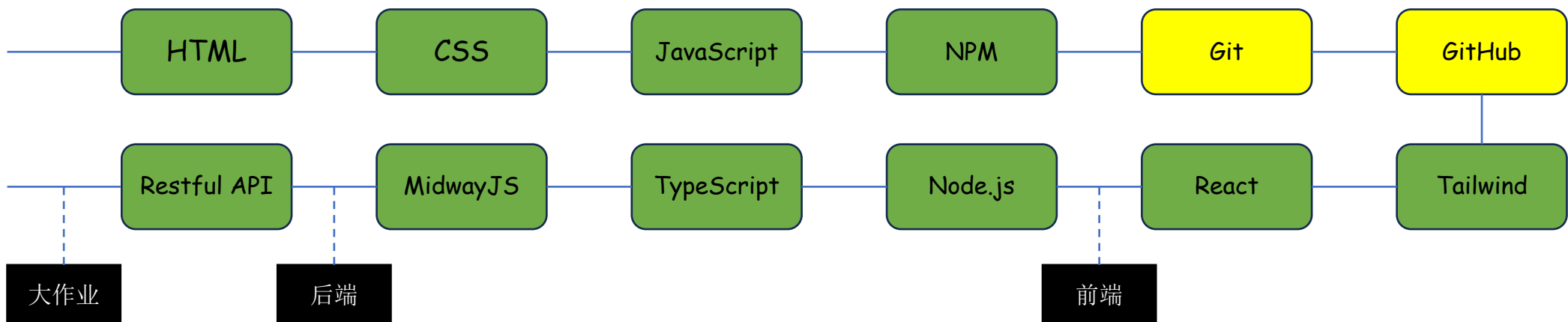


Web 持续集成、部署与性能优化

章许帆



代码是如何部署成一个服务的？

持续集成 GitHub Actions

<https://docs.github.com/zh/actions/automating-builds-and-tests/building-and-testing-nodejs>

```
.github > workflows > ! node.js.yml
1  name: Node.js CI
2  on:
3    push:
4      branches: [ "main" ]
5    pull_request:
6      branches: [ "main" ]
7  jobs:
8    build:
9      runs-on: ubuntu-latest
10     strategy:
11       matrix:
12         node-version: [20.x]
13     steps:
14       - uses: actions/checkout@v4
15       - name: Use Node.js ${ matrix.node-version }
16         uses: actions/setup-node@v3
17         with:
18           node-version: ${ matrix.node-version }
19           cache: 'npm'
20       - run: npm ci
21       - run: npm run build --if-present
22       - run: npm test
```

GitHub Actions

build (20.x)
Started 1m 7s ago

Search logs

> ✓ Set up job1s

> ✓ Run actions/checkout@v41s

> ✓ Use Node.js 20.x1s

> ✓ Run npm ci19s

> ✓ Run npm run build --if-present3s

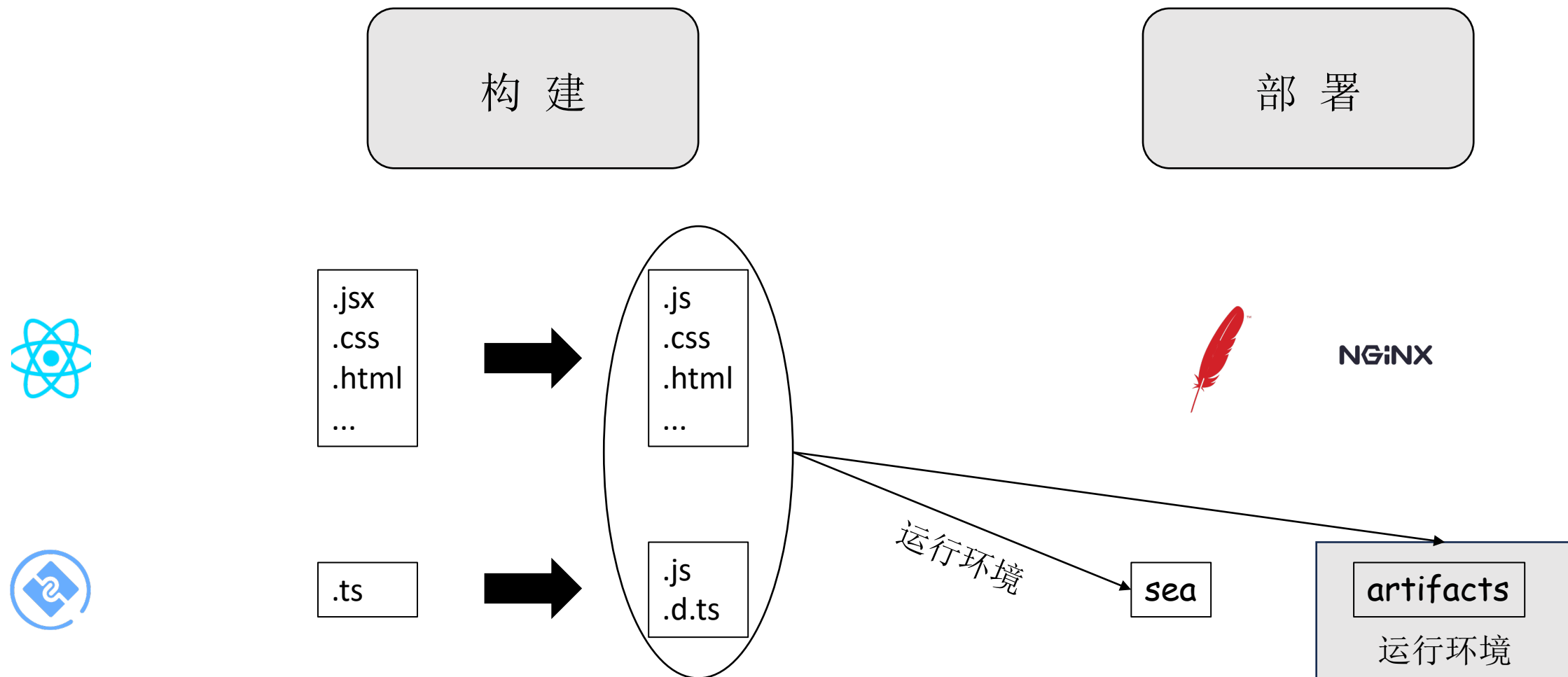
> ✓ Run npm test5s

● Post Use Node.js 20.x38s

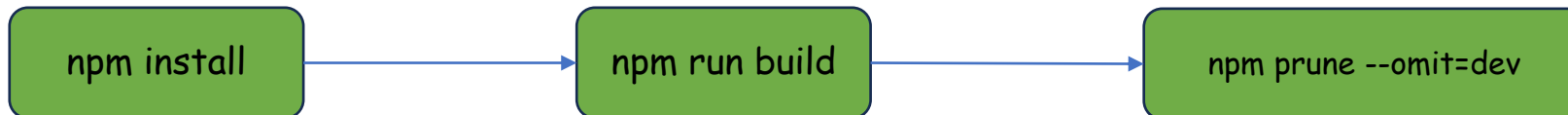
```
1 Post job cleanup.
2 /usr/bin/tar --posix -cf cache.tzst --exclude cache.tzst -P -C /home/runner/work/node-backend/node-backend --files-from manifest.txt
  --use-compress-program zstdmt
3 Cache Size: ~26 MB (27516331 B)
4 Cache saved successfully
5 Cache saved with the key: node-cache-Linux-npm-e97a8b9a9761f0c59d0f79c5f6f7bff5c68dc910d0ec25f83b9bb4c8bab16b7b
```

○ Post Run actions/checkout@v4

软件构建与部署

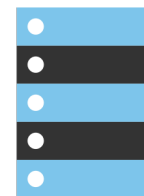


软件构建



```
> dist
> logs
> node_modules
> src
> test
JS bootstrap.js
JS jest.config.js
{} package-lock.json
{} package.json
i README.md
⬇ README.zh-CN.md
TS tsconfig.json
```

- dist
- node_modules
- bootstrap.js
- package.json



Server

软件部署-pm2



```
$ pm2 start      # 启动一个服务
$ pm2 list       # 列出当前的服务
$ pm2 stop       # 停止某个服务
$ pm2 restart    # 重启某个服务
$ pm2 delete     # 删除某个服务
$ pm2 logs       # 查看服务的输出日志
```

```
pm2 start ./bootstrap.js --name [backend]
```

```
● fan@Finleys-MacBook-Pro backend % pm2 list
```

id	name	mode	↺	status	cpu	memory
0	backend	fork	0	online	0%	57.6mb

软件部署-pm2

简单易用

进程管理

自动重启

运行诊断

资源限制

软件部署-pkg

```
38     "lint": "mwtsc check",
39     "lint:fix": "mwtsc fix",
40     "ci": "npm run cov",
41     "build": "mwtsc --cleanOutDir",
42     "bundle": "bundle && npm run build",
43     "pkg": "pkg . -t node18-macos-x64 --out-path ./output"
44   },
45   "bin": "./bootstrap.js",
46   "pkg": {
47     "scripts": "dist/**/*.js"
48   },
49   "repository": {
50     "type": "git",
51     "url": ""
52   },
```

软件部署-pkg

部署简洁

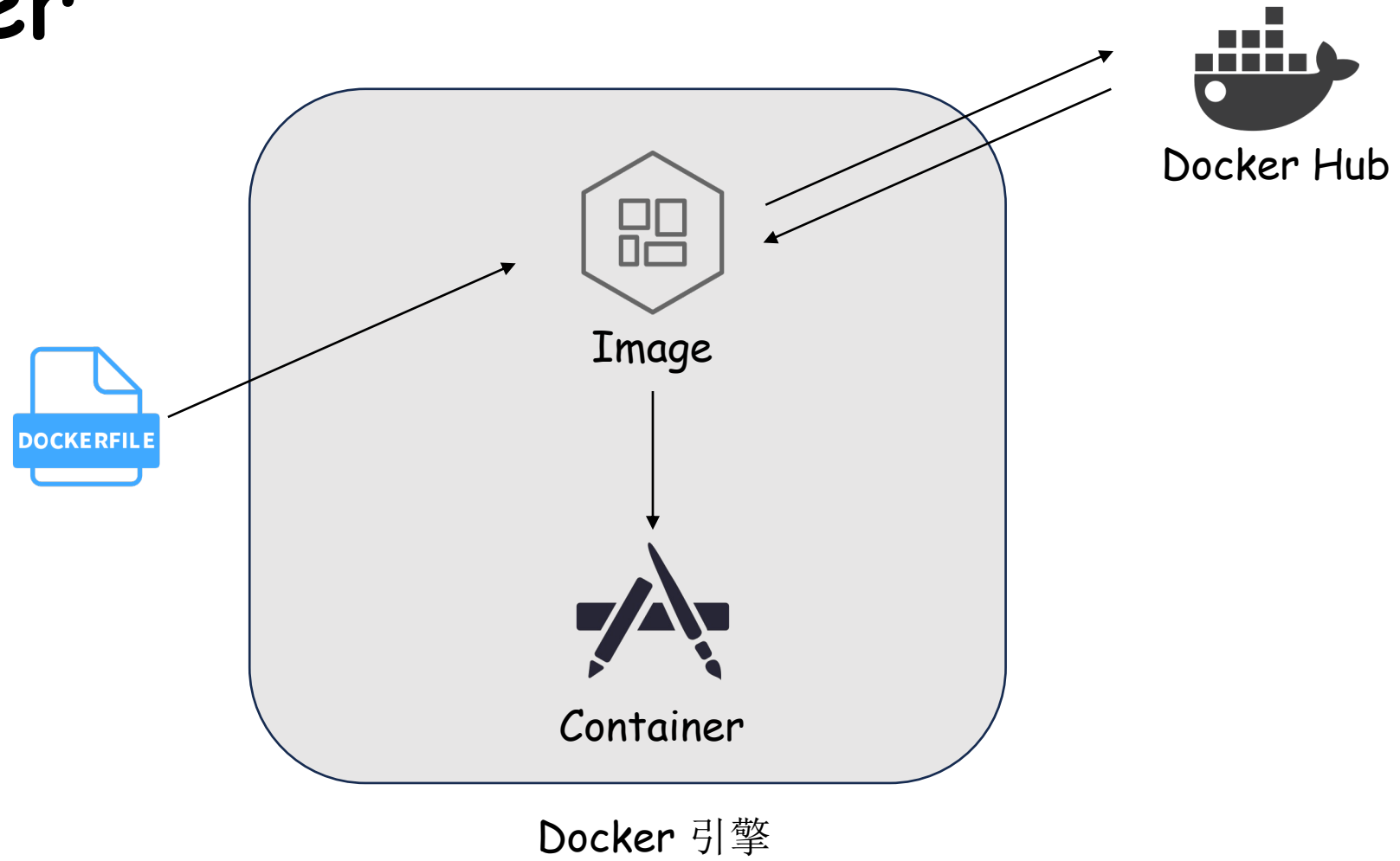
无需环境

易于分发

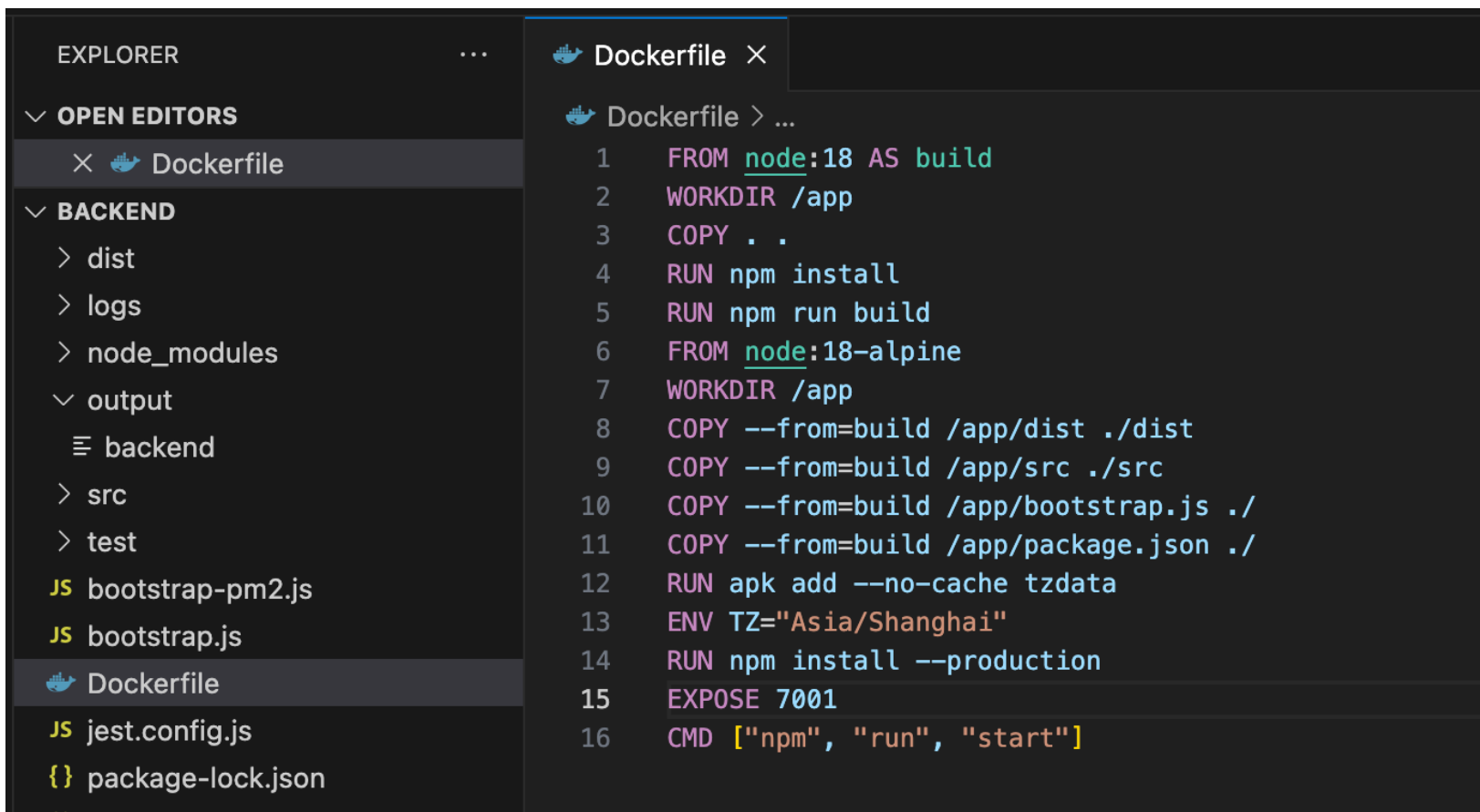
平台依赖

全量更新

docker



软件部署-docker



The image shows a screenshot of the Visual Studio Code interface. On the left, the Explorer sidebar is open, showing a project structure with folders like 'dist', 'logs', 'node_modules', 'output', 'src', and 'test'. A file named 'Dockerfile' is selected and highlighted. The main editor area on the right displays the content of the 'Dockerfile' file, which contains 16 lines of Docker build instructions. The instructions start with 'FROM node:18 AS build' and end with 'CMD ["npm", "run", "start"]'.

```
1 FROM node:18 AS build
2 WORKDIR /app
3 COPY . .
4 RUN npm install
5 RUN npm run build
6 FROM node:18-alpine
7 WORKDIR /app
8 COPY --from=build /app/dist ./dist
9 COPY --from=build /app/src ./src
10 COPY --from=build /app/bootstrap.js ./
11 COPY --from=build /app/package.json ./
12 RUN apk add --no-cache tzdata
13 ENV TZ="Asia/Shanghai"
14 RUN npm install --production
15 EXPOSE 7001
16 CMD ["npm", "run", "start"]
```

软件部署-docker

环境隔离

轻量

可移植

版本控制

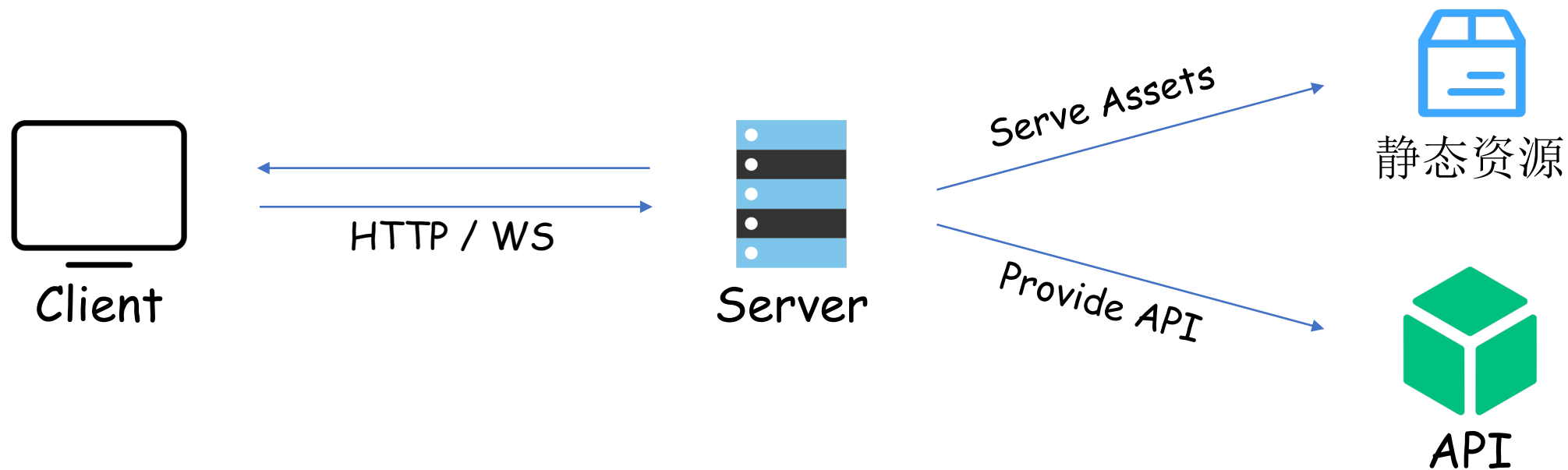
可复现

便于扩展

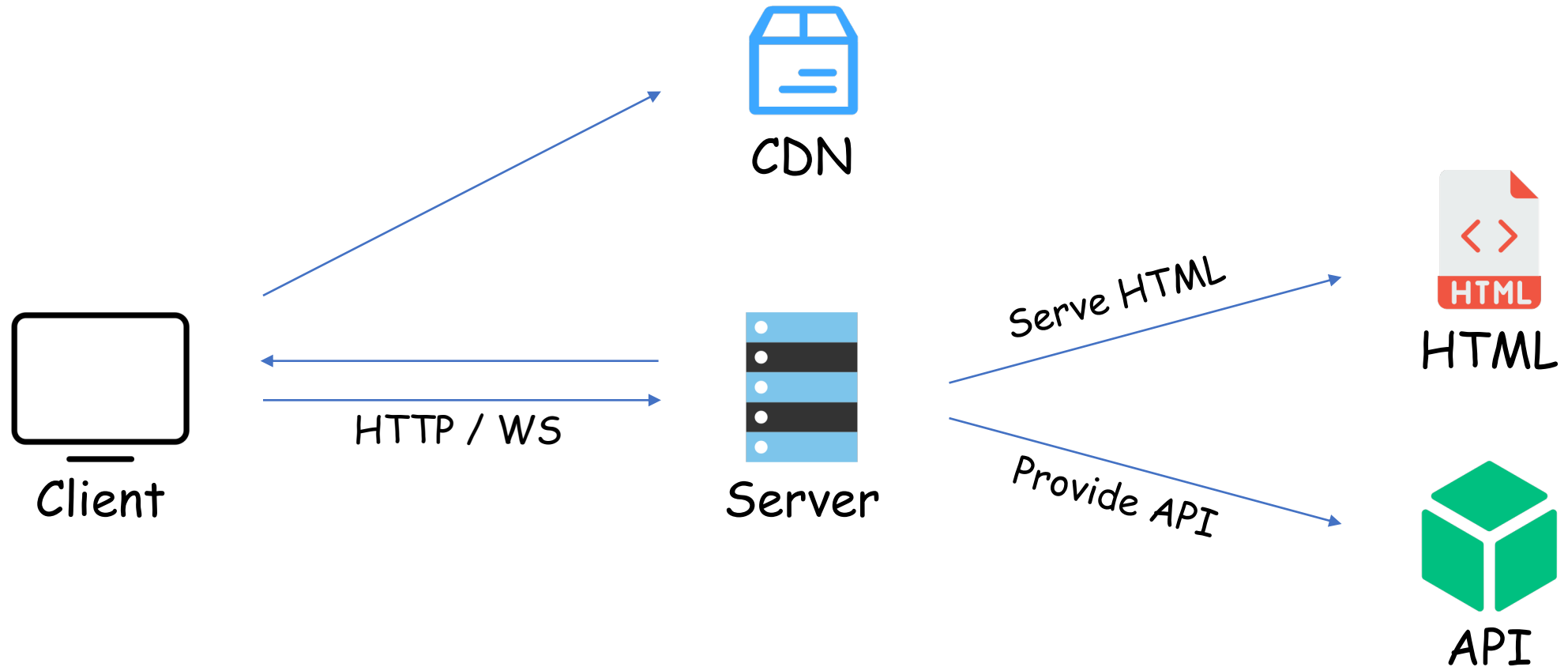
资源消耗

维护复杂

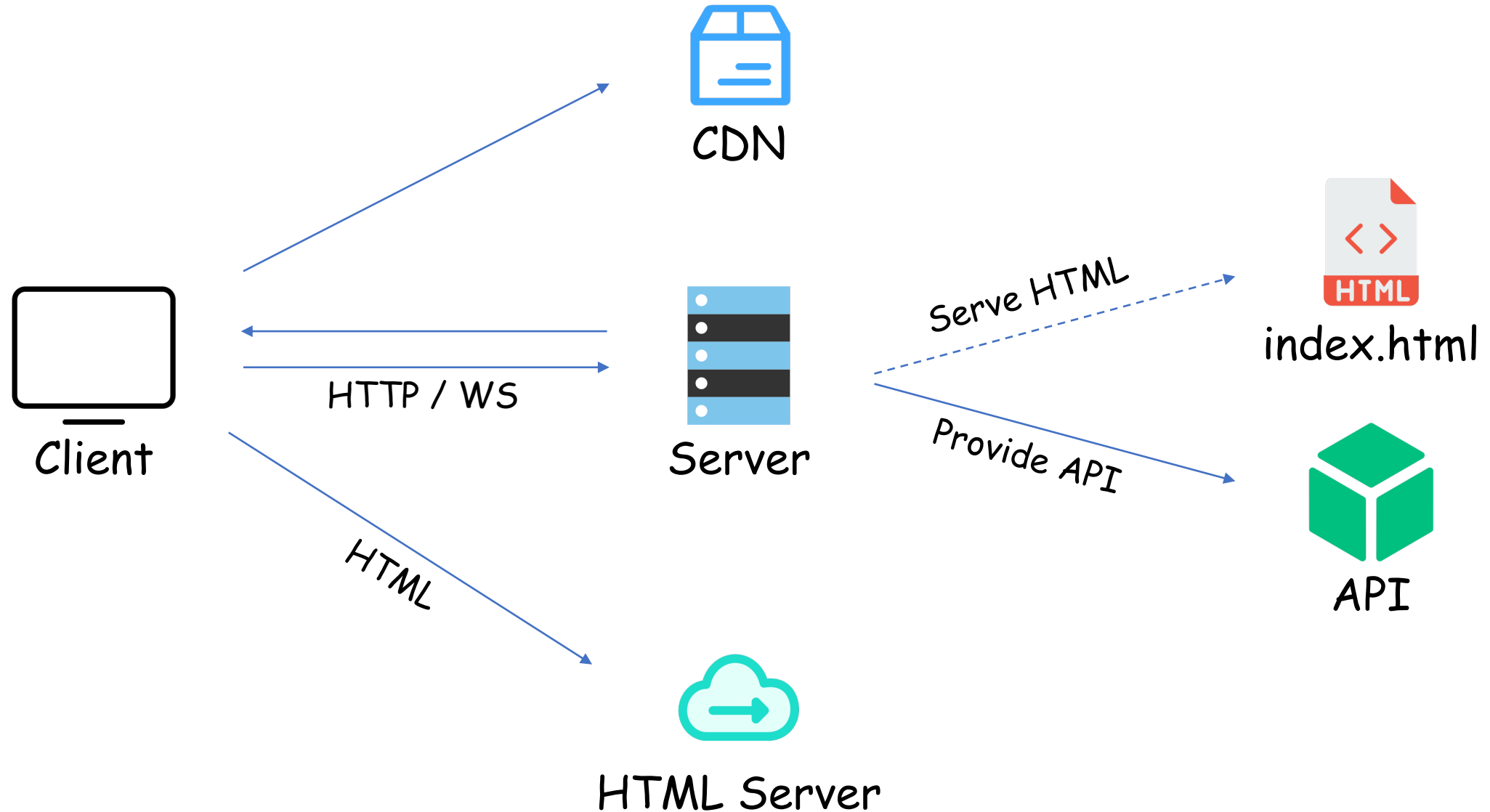
一体化部署



CDN + Server



CDN + HTML Server + Web Server



案例分析

一个有时间显示、有数据展示的 **Web** 界面，如果它响应用户的界面点击操作存在较大的延迟，结合本门课程中提到的**Web**应用相关的技术基础，分析一下，可能是由哪些原因造成的？

相关技术栈

<https://nodejs.org/>

<https://vitejs.dev/guide/>

<https://react.dev/>

<https://tailwindcss.com/>

<https://midwayjs.org/>

<https://vitest.dev/guide/>

<https://jestjs.io/>

<https://developer.mozilla.org/en-US/docs/Web/API/WebSocket>

<https://pm2.io/>

谢 谢