

基于Electron Vue的桌面应用开发实践





赵 帅
前端工程师 @美团点评-到店事业群

负责-度假业务系统工程化
瓦匠桌面应用开发者

01 瓦匠应用的介绍

02 桌面应用的技术选型

03 Electron Vue项目实践

04 桌面应用开发总结

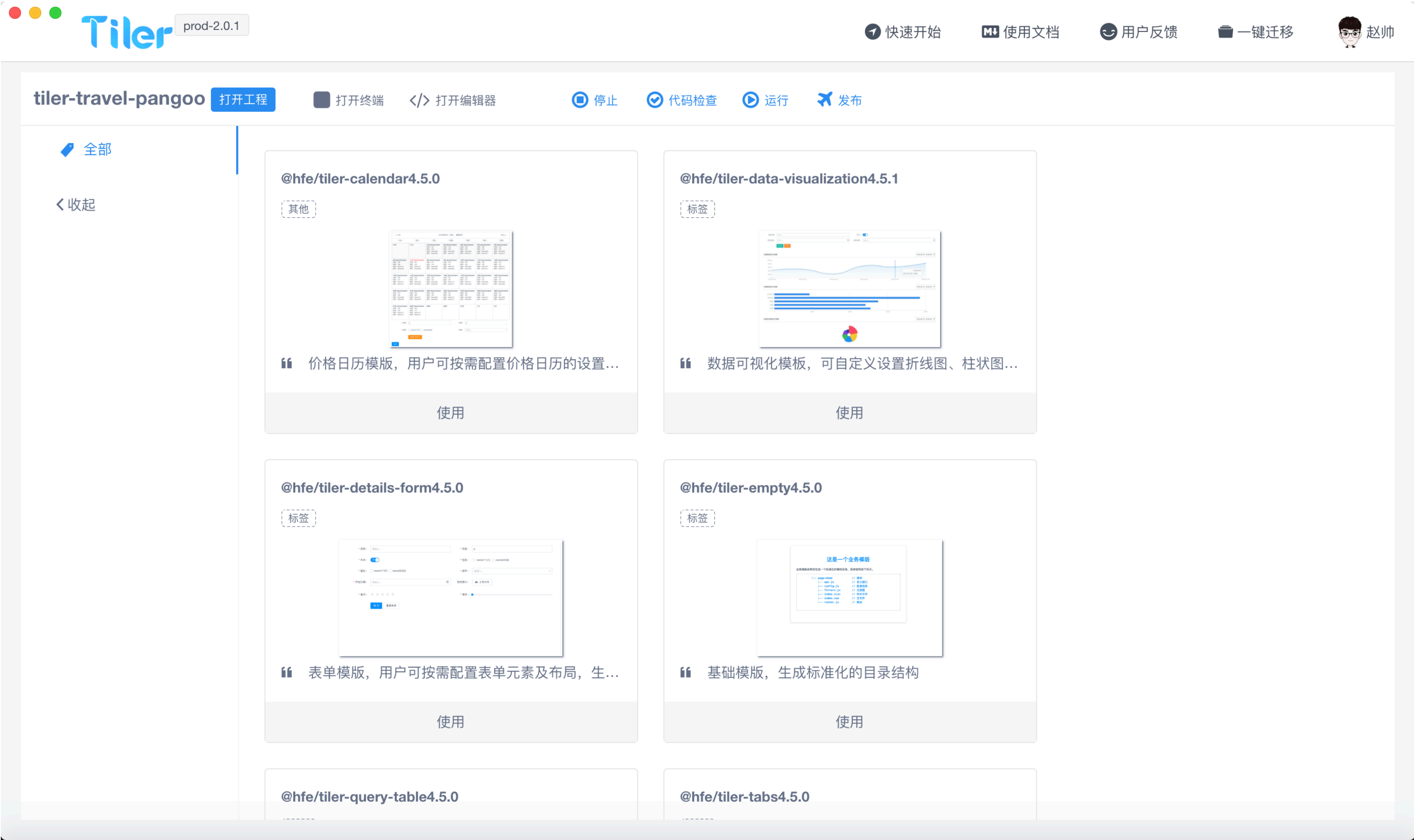
应用介绍

瓦匠

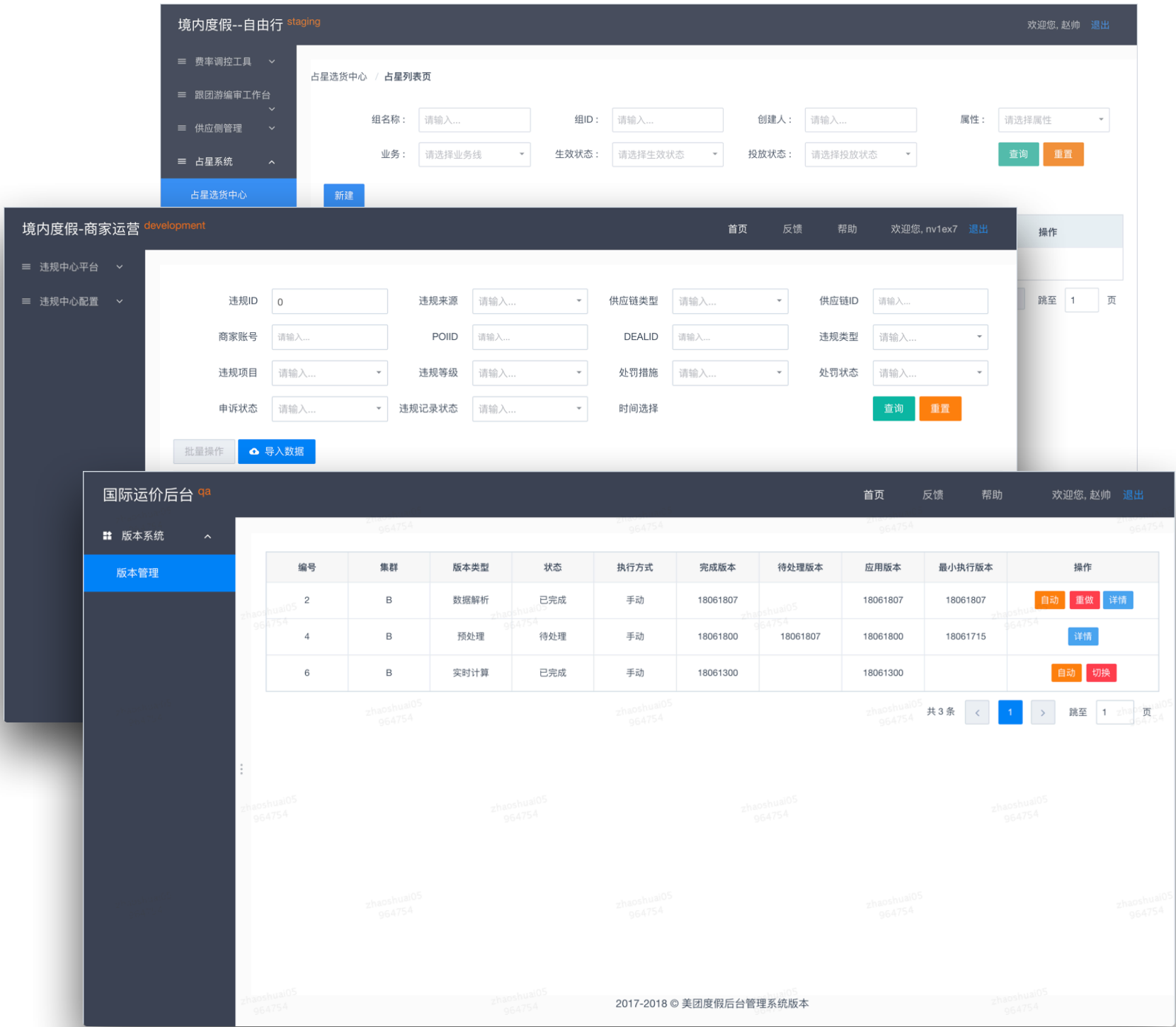
瓦匠工作台

一体化解决方案

提高开发效率



应用介绍
背景概述



需求类型

CRUD页面
场景化页面

新项目

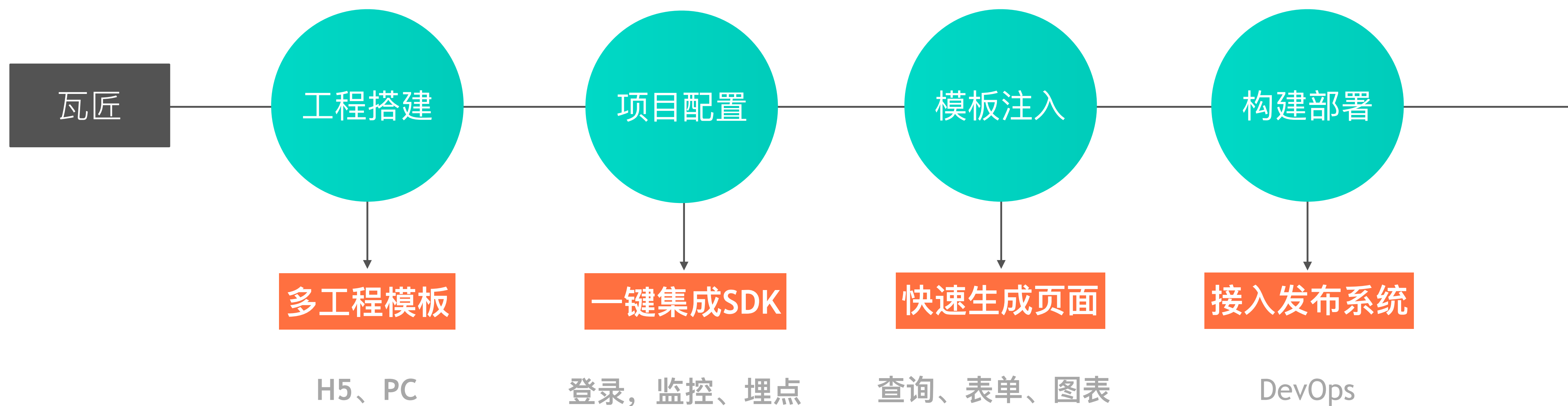
登录SDK，数据上报，权限系统，发布系统等.....



效率低，同质化，重复成本

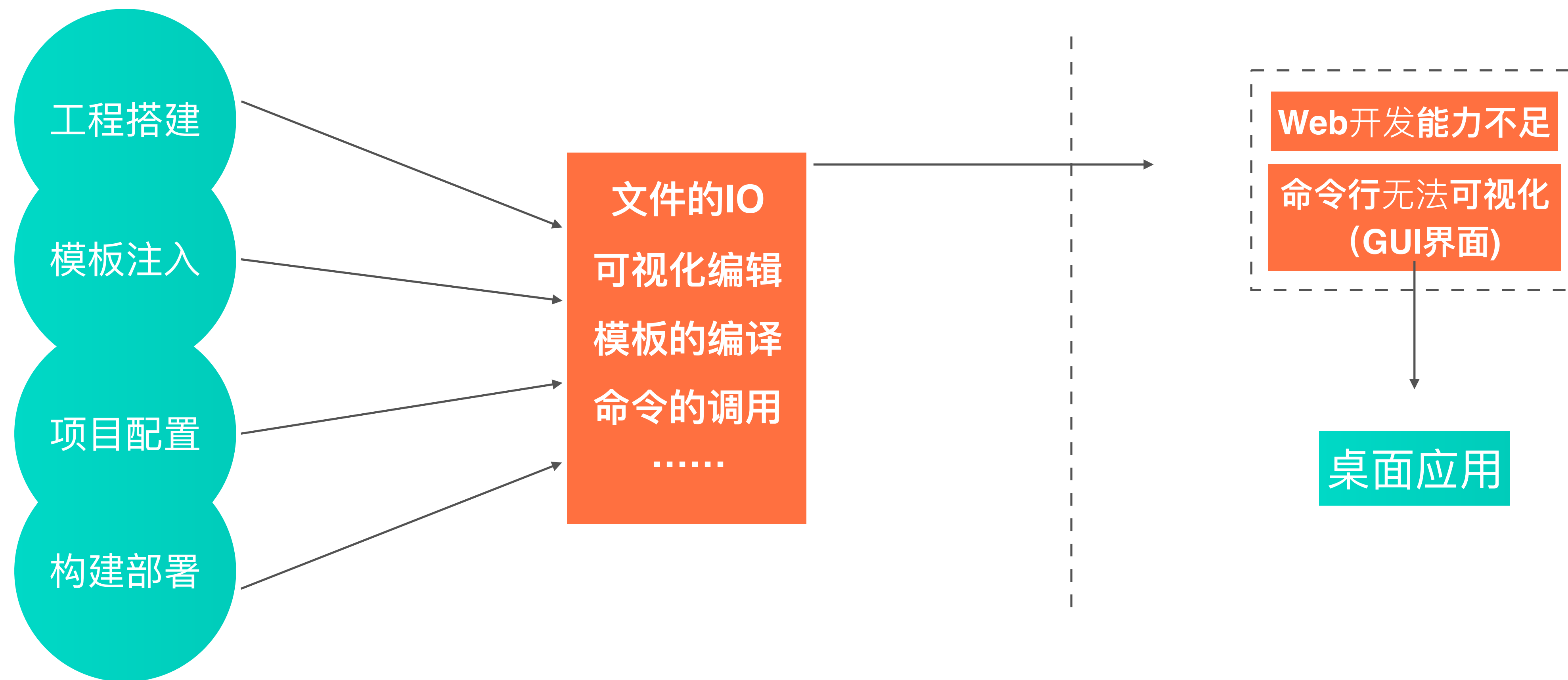
应用介绍

核心功能



应用介绍

技术挑战



01 瓦匠应用的介绍

02 桌面应用的技术选型

03 Electron Vue项目实践

04 桌面应用开发总结

技术选型

桌面应用开发选型



纯Native开发

C++/C#/C+.....(Windows)
Objective-C/Swift(Mac)

优点：性能好，安装包小
缺点：开发周期长，无法跨平台



开发框架

Qt: (Windows/Mac)
NW.js/Electron(Windows/Mac)

优点：接近Native性能，功能丰富，跨平台
缺点：库太大，编译慢，开发周期长

优点：开发速度快，跨平台，更新迭代快
缺点：集成Chromium，安装包大

技术选型

Electron应用



Skype



GitHub Desktop



Figma



Flow



Beaker Browser



Hyper



Kap



Now Desktop



Insomnia



SvgSus



Simplenote

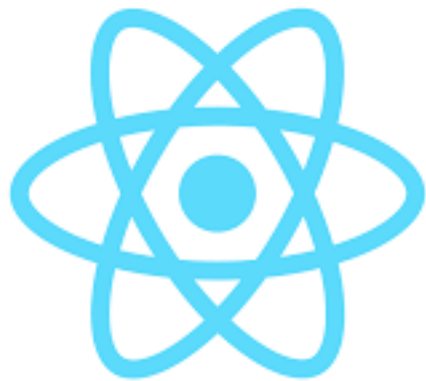


Collectie

VS Code, POSTMAN

技术选型

前端选型



体积小、轻量化
健全的生态
优秀的组件化机制



01 瓦匠应用的介绍

02 桌面应用的技术选型

03 Electron Vue项目实践

04 桌面应用开发总结

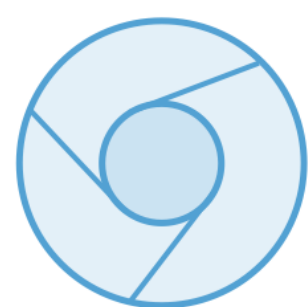
项目实践

Electron原理

两个核心

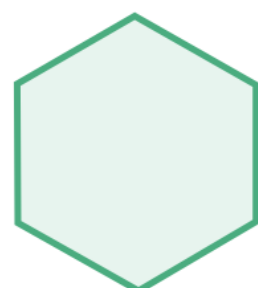
Main-process

Render-process



Chromium
for making
web pages

+

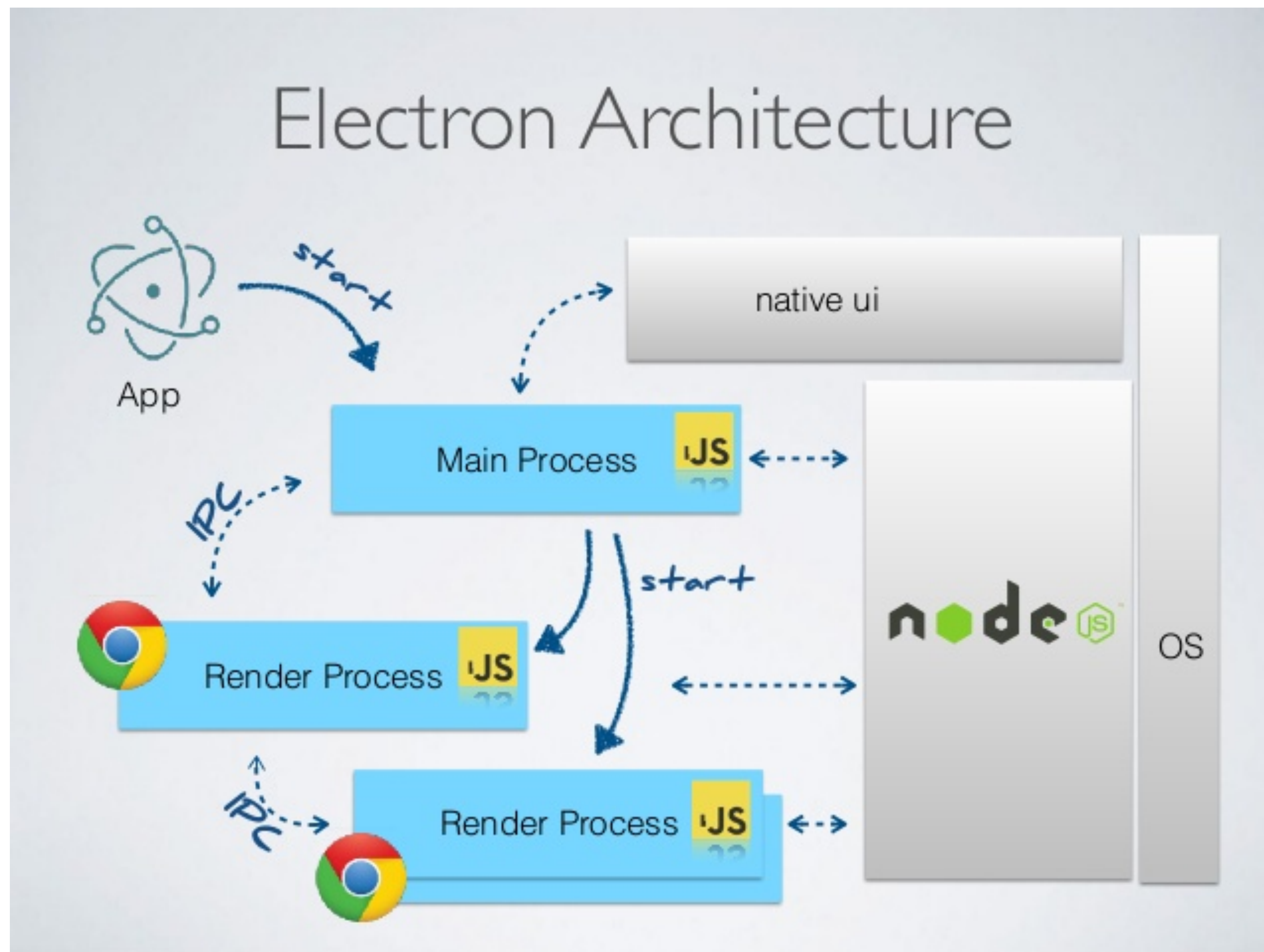


Node.js
for filesystems
and networks

+



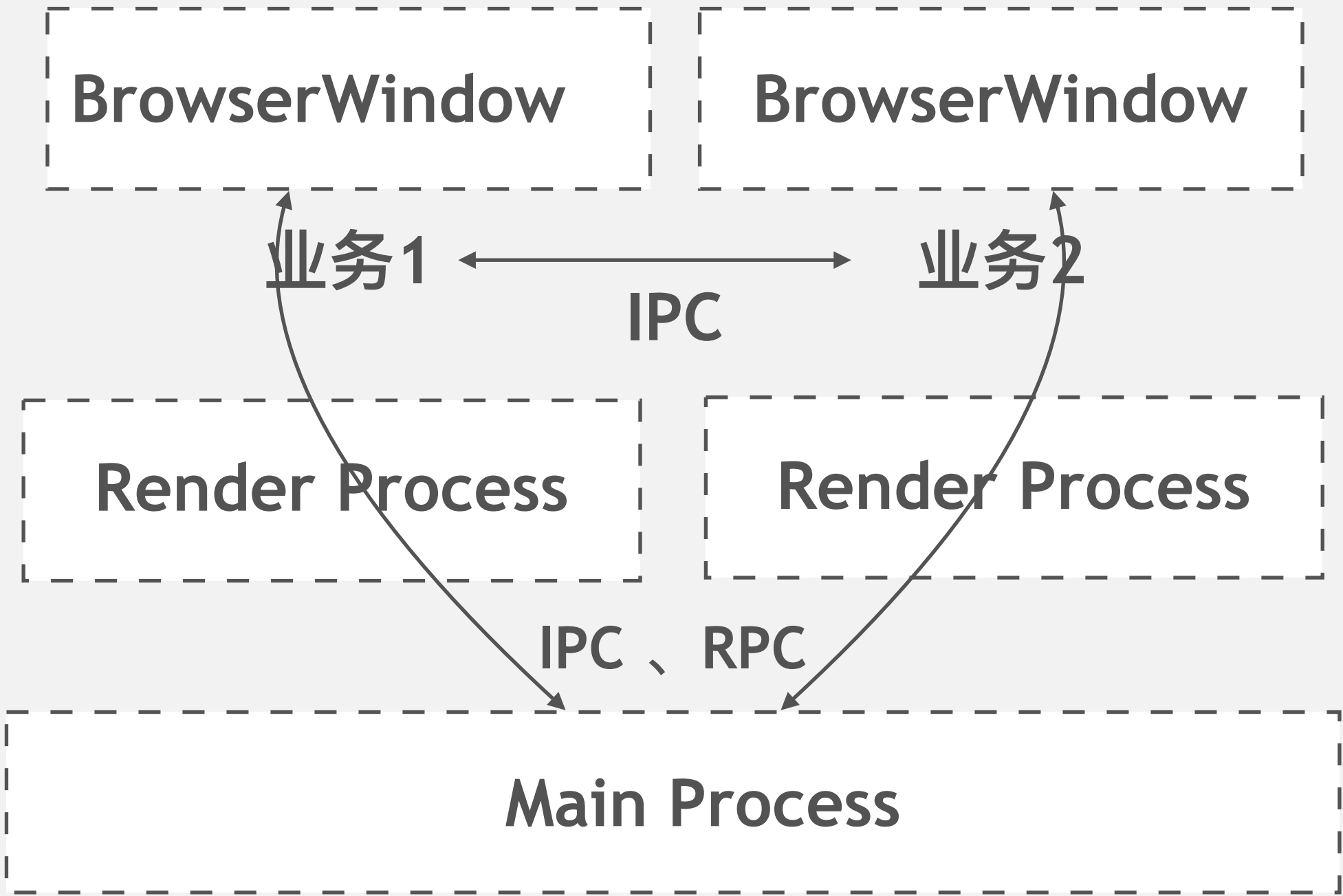
Native APIs
for three
systems



项目实践

应用架构选型

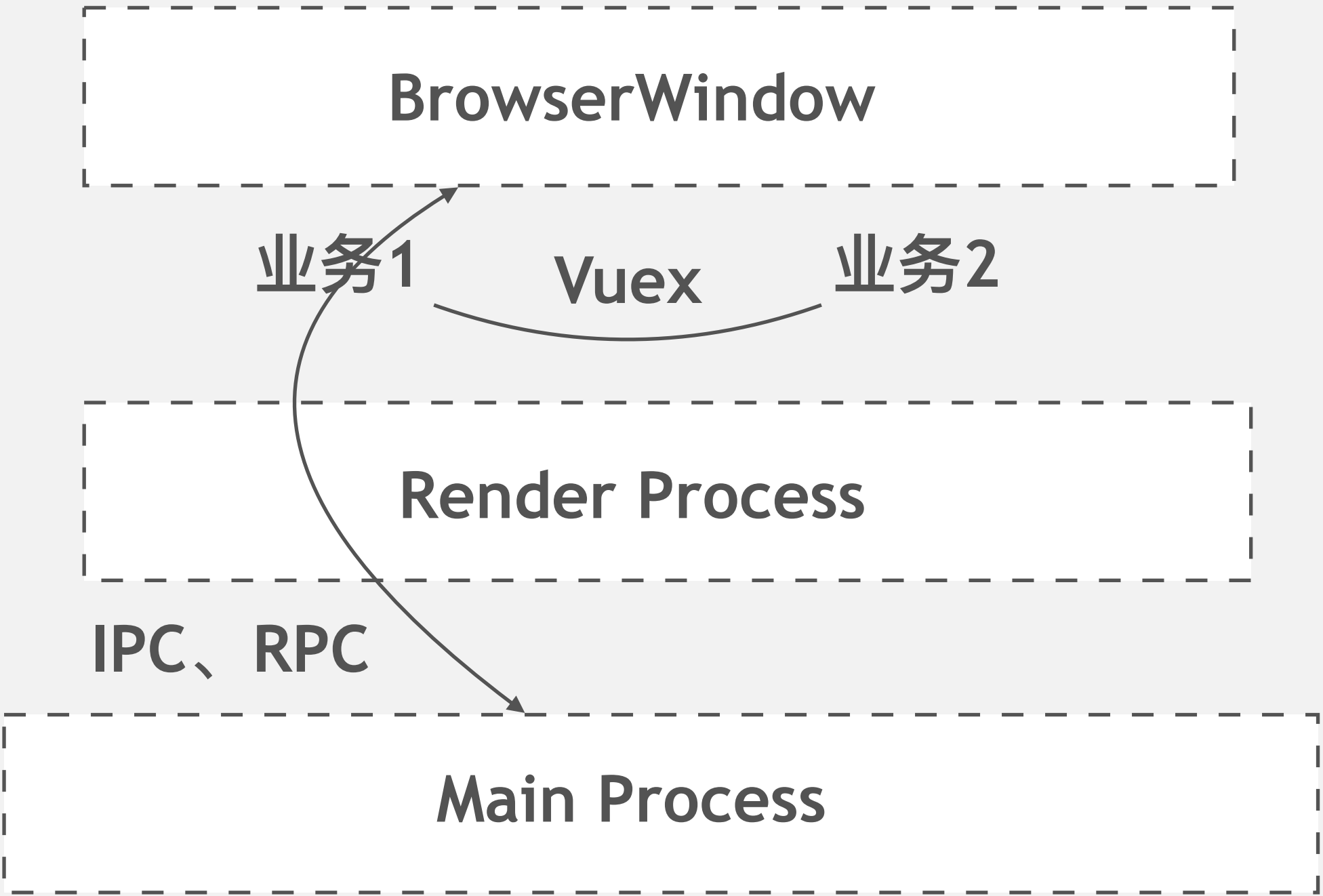
多窗口



适用场景 IDE, IM等

特点：项目协同开发，多任务处理

单窗口



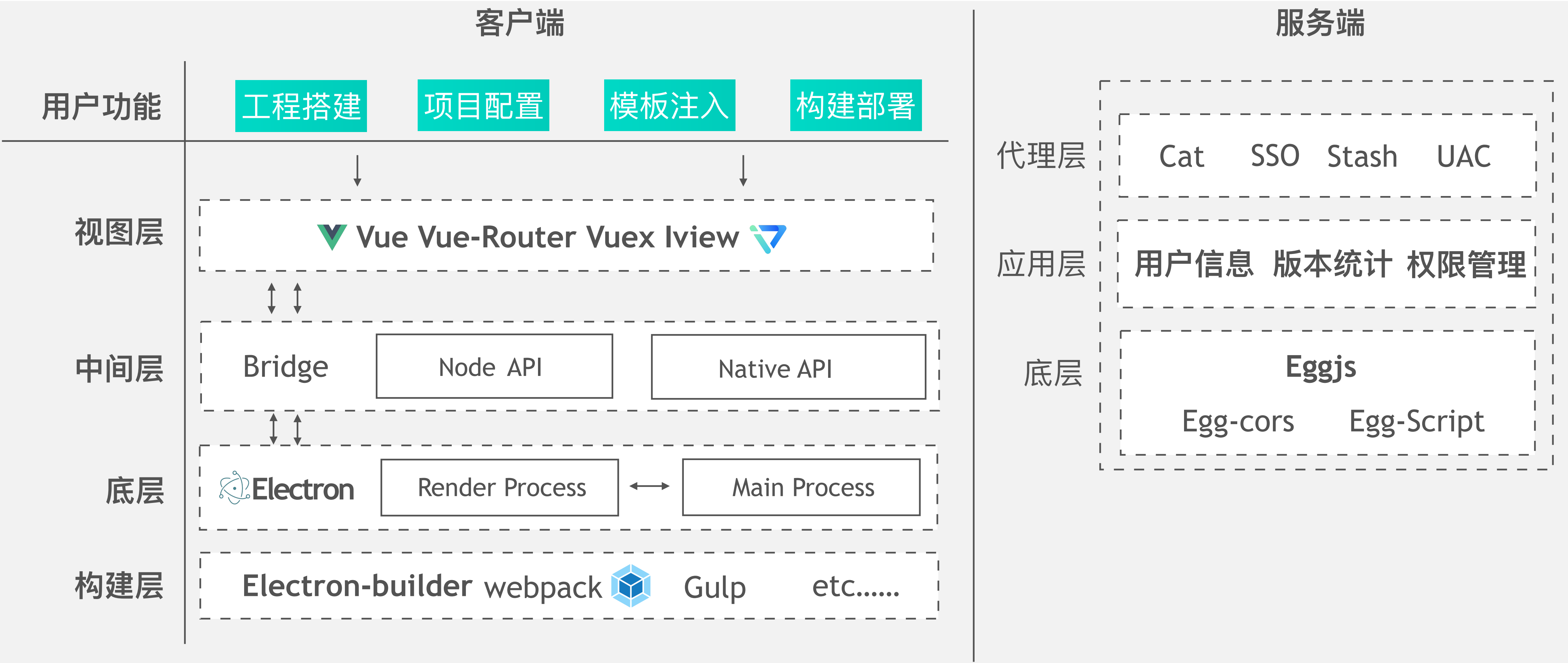
适用场景 工具，流程

特点：交互简单，通信方便



项目实践

应用架构



项目实践

工程目录结构

构建打包

构建配置

主进程

渲染进程

Electron SWA 应用

```
├── bin
│   ├── jenkins.sh
│   └── upload.sh
├── tasks
│   ├── gulp.dev.js
│   ├── gulp.prod.js
│   └── release.js
├── conf
│   ├── webpack.renderer.conf.js
│   ├── webpack.main.conf.js
│   └── release.conf.js
├── src
│   ├── main
│   │   ├── index.js
│   │   └── utils.js
│   ├── renderer
│   │   ├── components
│   │   ├── pages
│   │   ├── store
│   │   ├── router
│   │   ├── App.vue
│   │   └── index.js
│   └── package.json
```

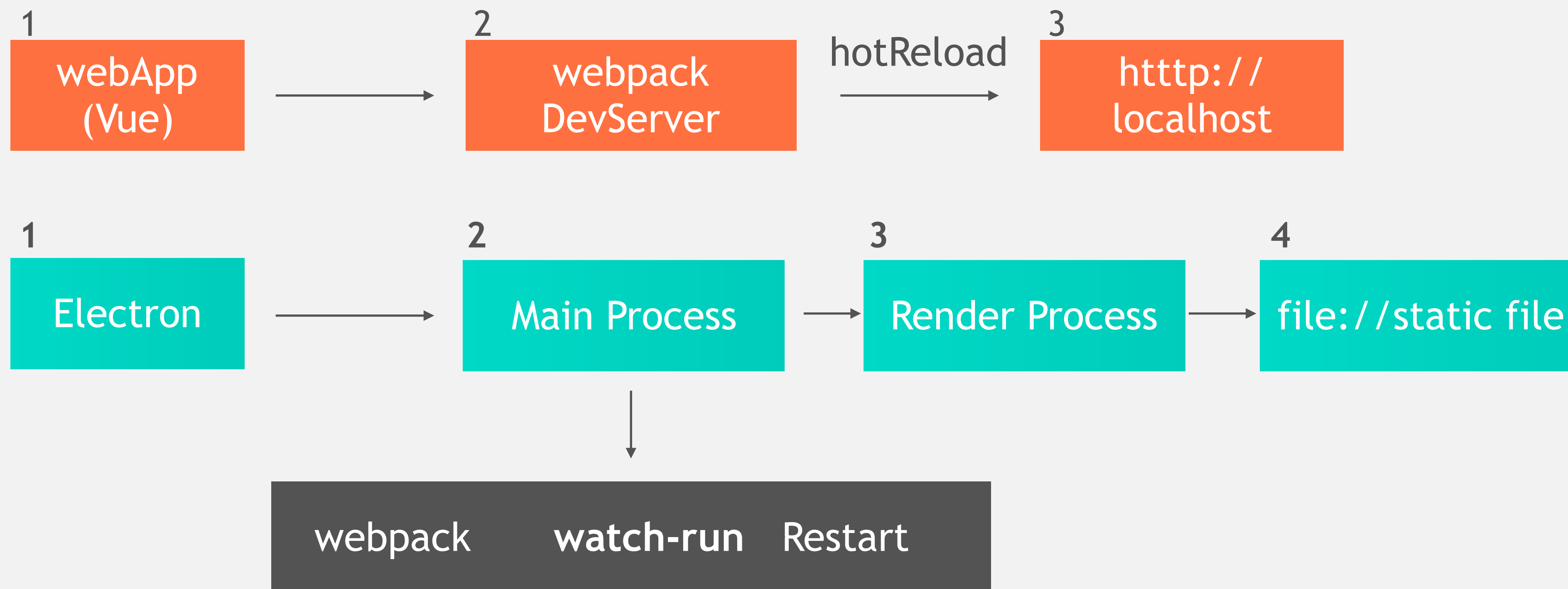
Web SPA 应用

```
├── bin
│   ├── jenkins.sh
│   └── run-test.sh
├── tasks
│   ├── gulp.dev.js
│   └── gulp.prod.js
├── conf
│   ├── gulpfile.babel.js
│   ├── webpack.dev.conf.js
│   ├── webpack.prod.conf.js
│   └── postcss.config.js
├── src
│   ├── components
│   ├── pages
│   ├── store
│   ├── router
│   ├── styles
│   ├── App.vue
│   └── index.js
└── package.json
```


Electron应用的开发调试、构建发布

项目实践

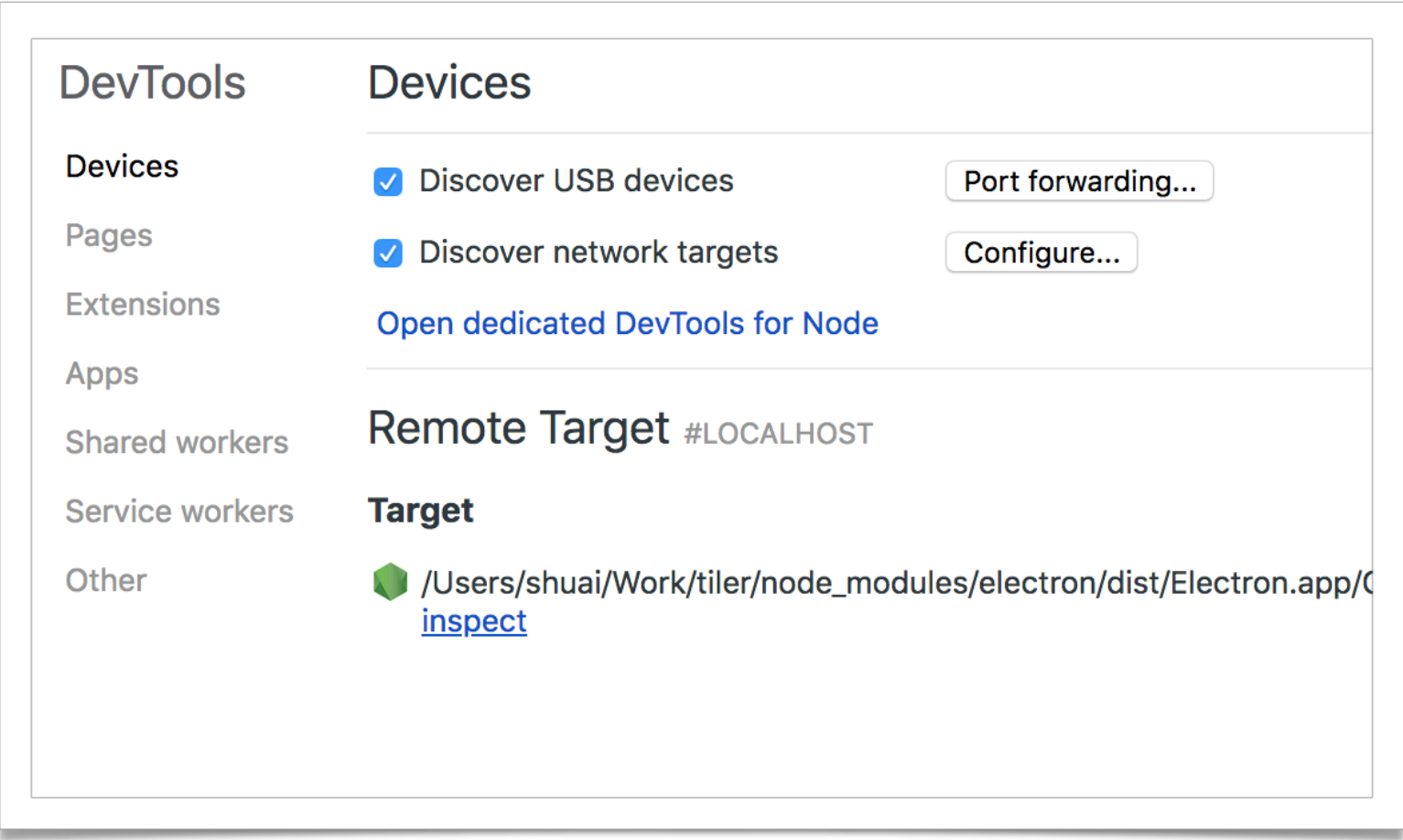
开发和调试



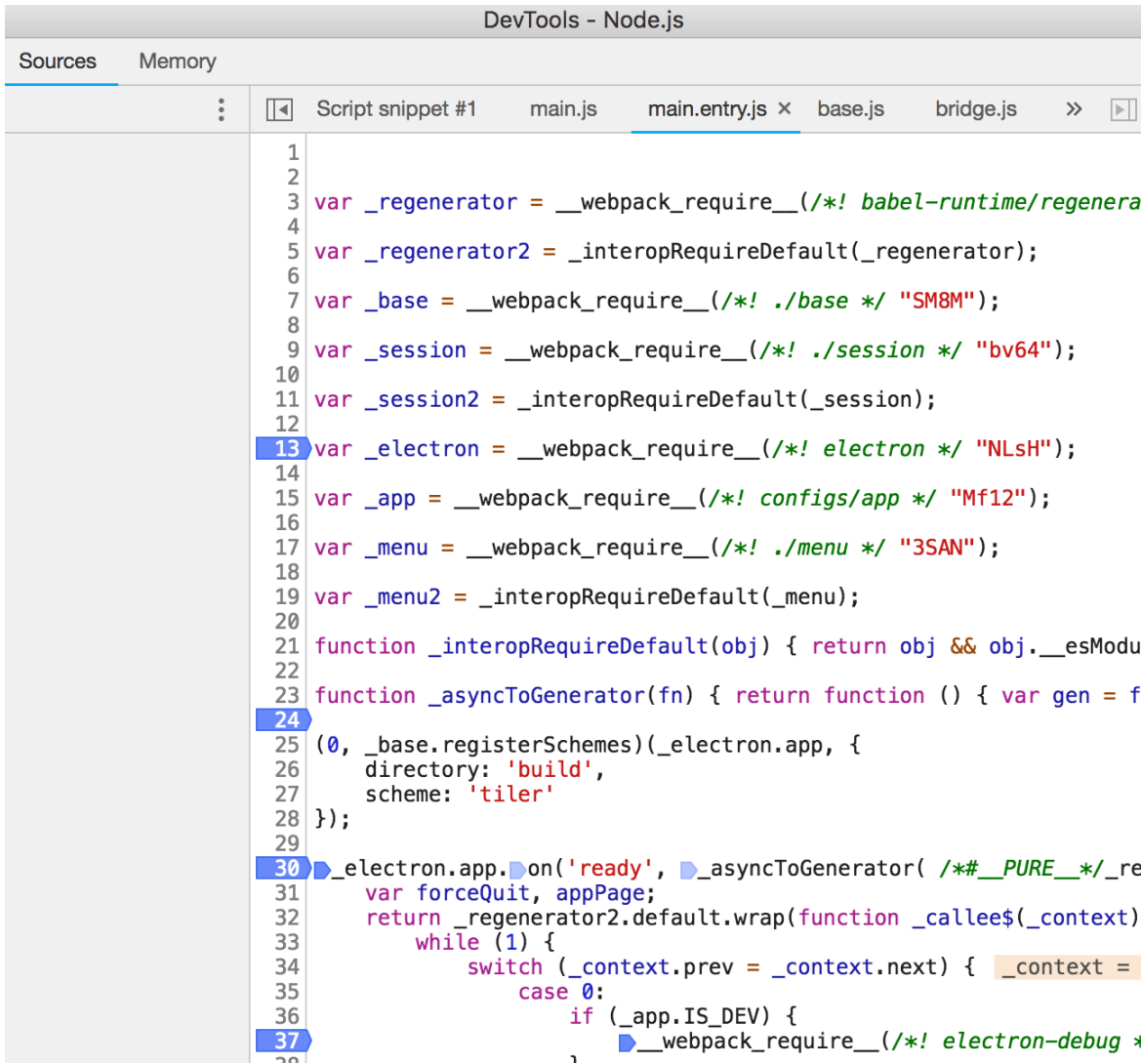
项目实践
开发和调试

- 渲染进程、主进程
 - 该如何调试?
- 渲染进程: vue-devtools
- 主进程: Electron —inspect

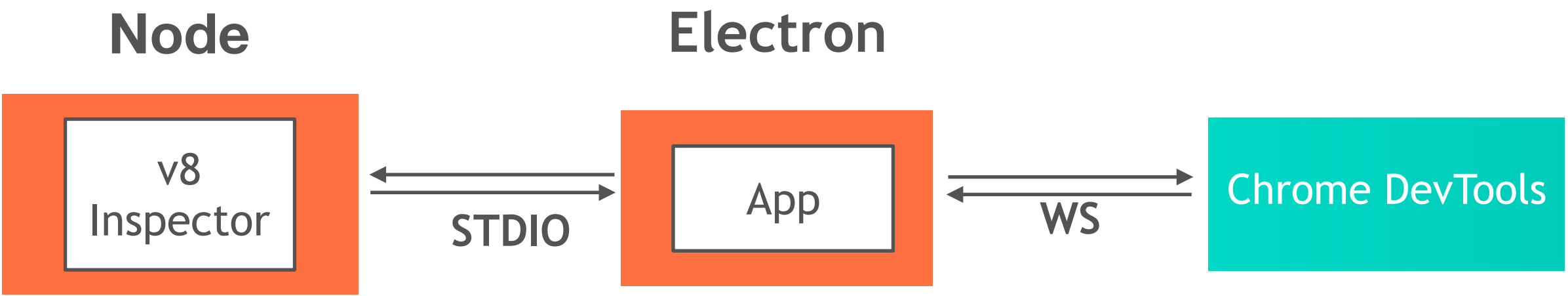
远程调试



chrome://inspect/#devices



主进程Debug

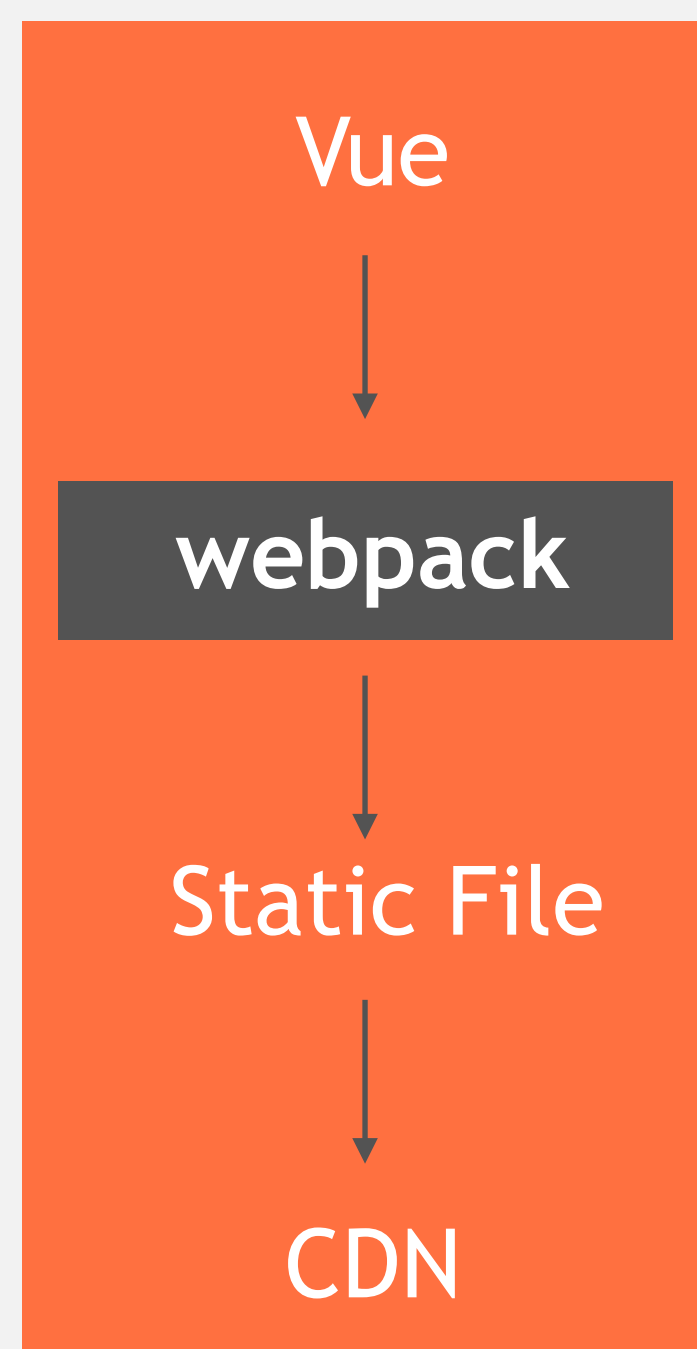


Chrome DevTools Protocol

项目实践

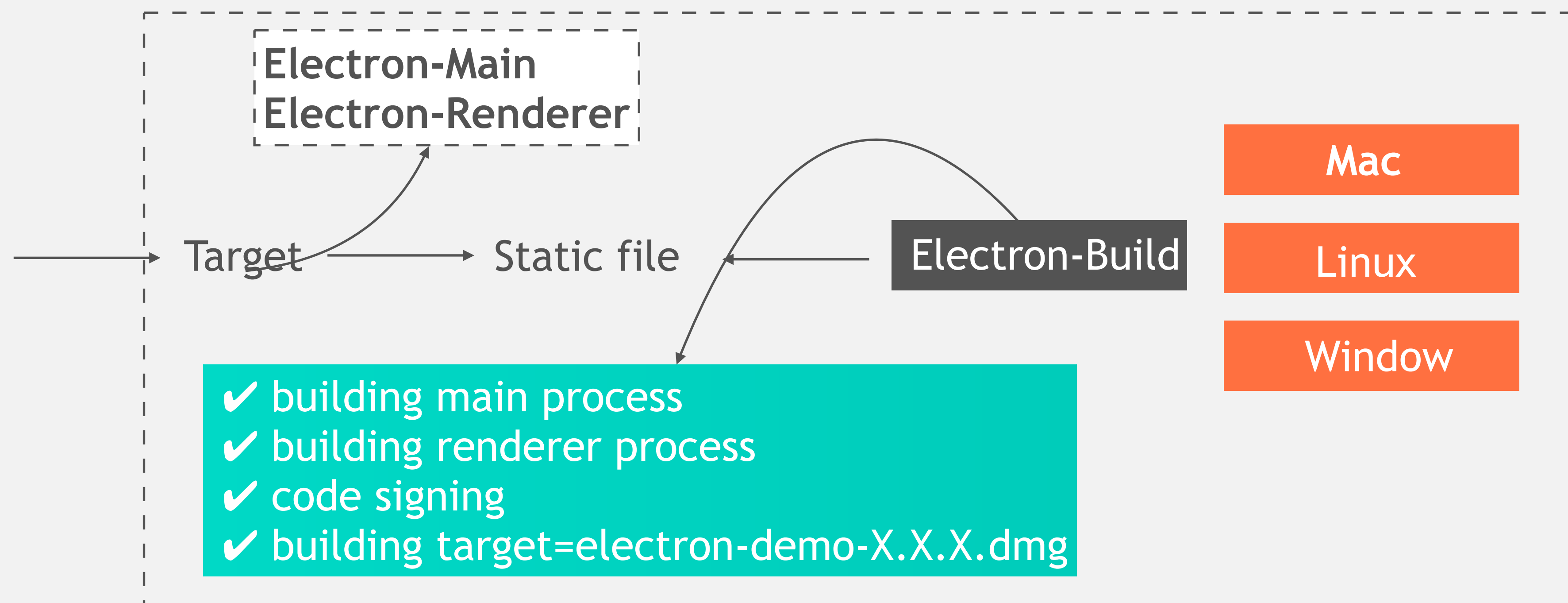
构建和发布

webApp本地构建



Web

Electron本地构建



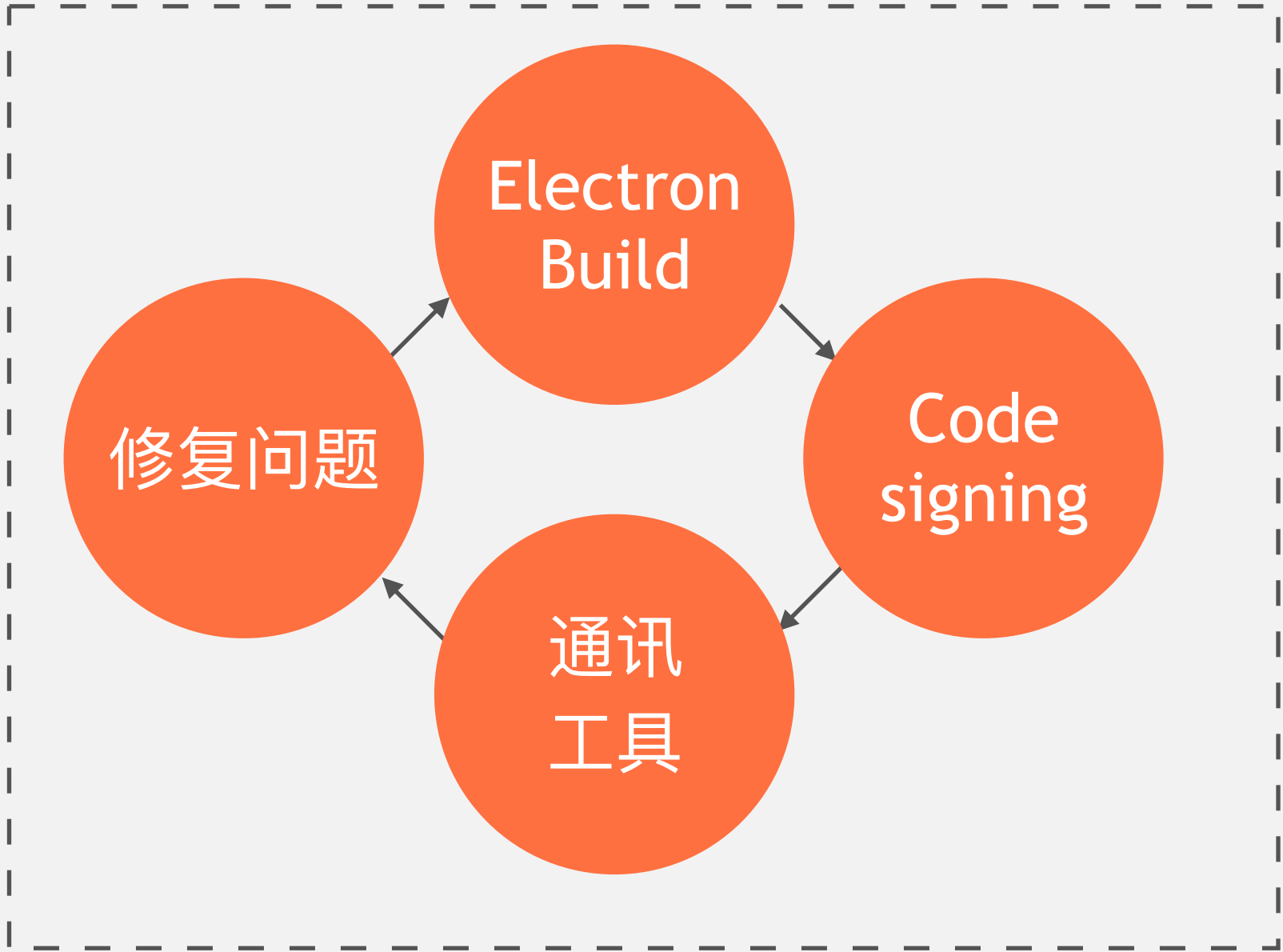
Native



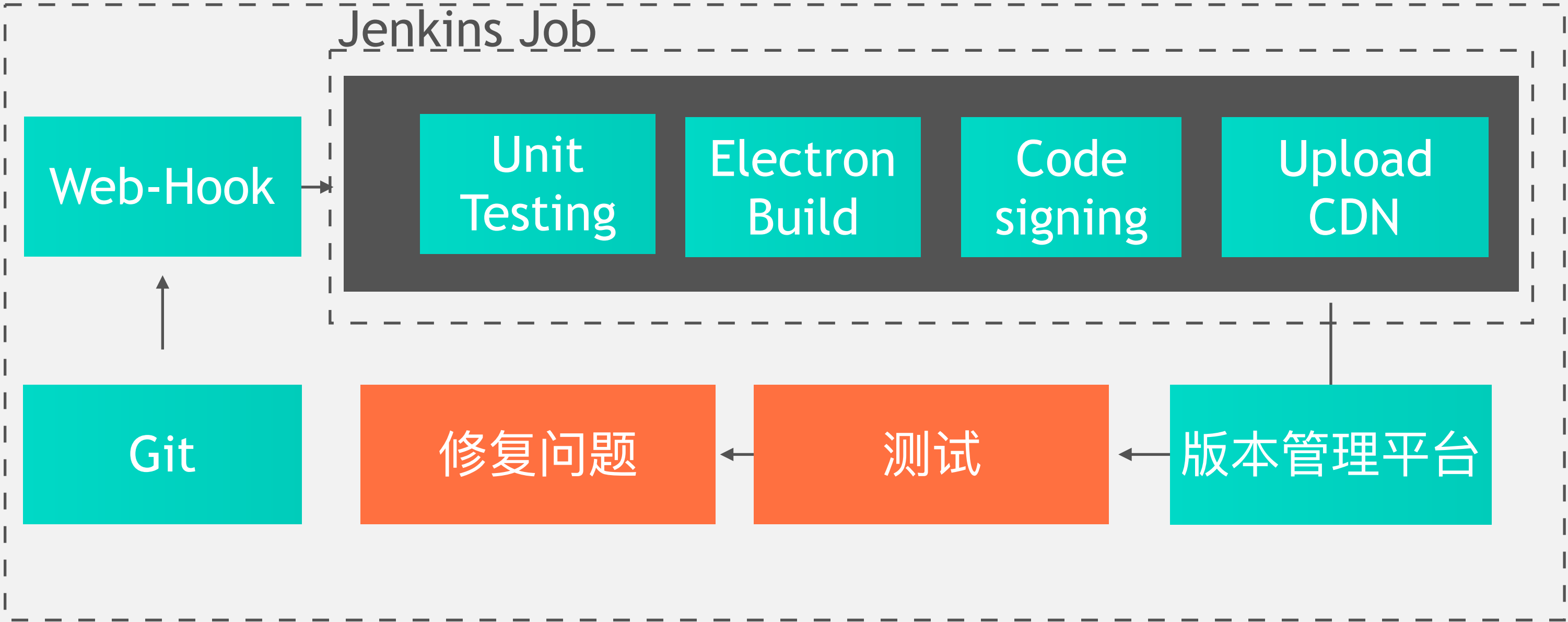
项目实践

构建和发布

手动构建、发布



自动化构建、发布

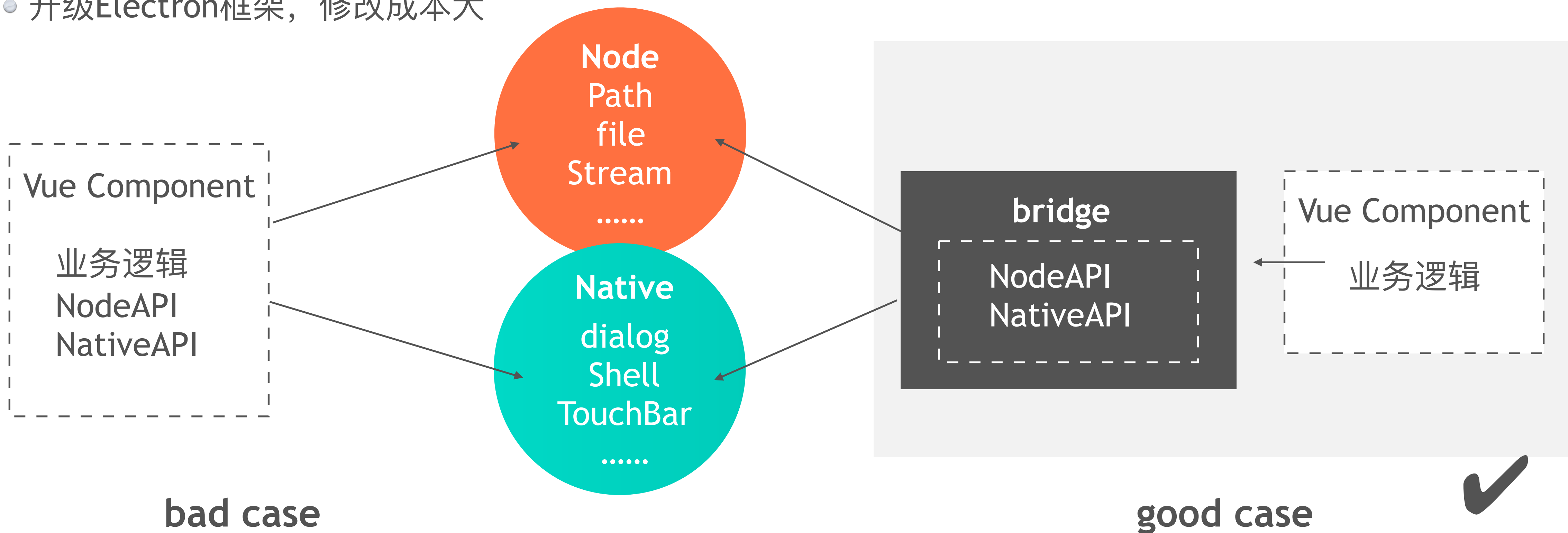


Electron应用中遇到的问题^{问题}和^{解决方案}解决方案

项目实践

#1 如何解耦业务逻辑

- 页面代码与业务逻辑混写，功能重复
- 升级Electron框架，修改成本大

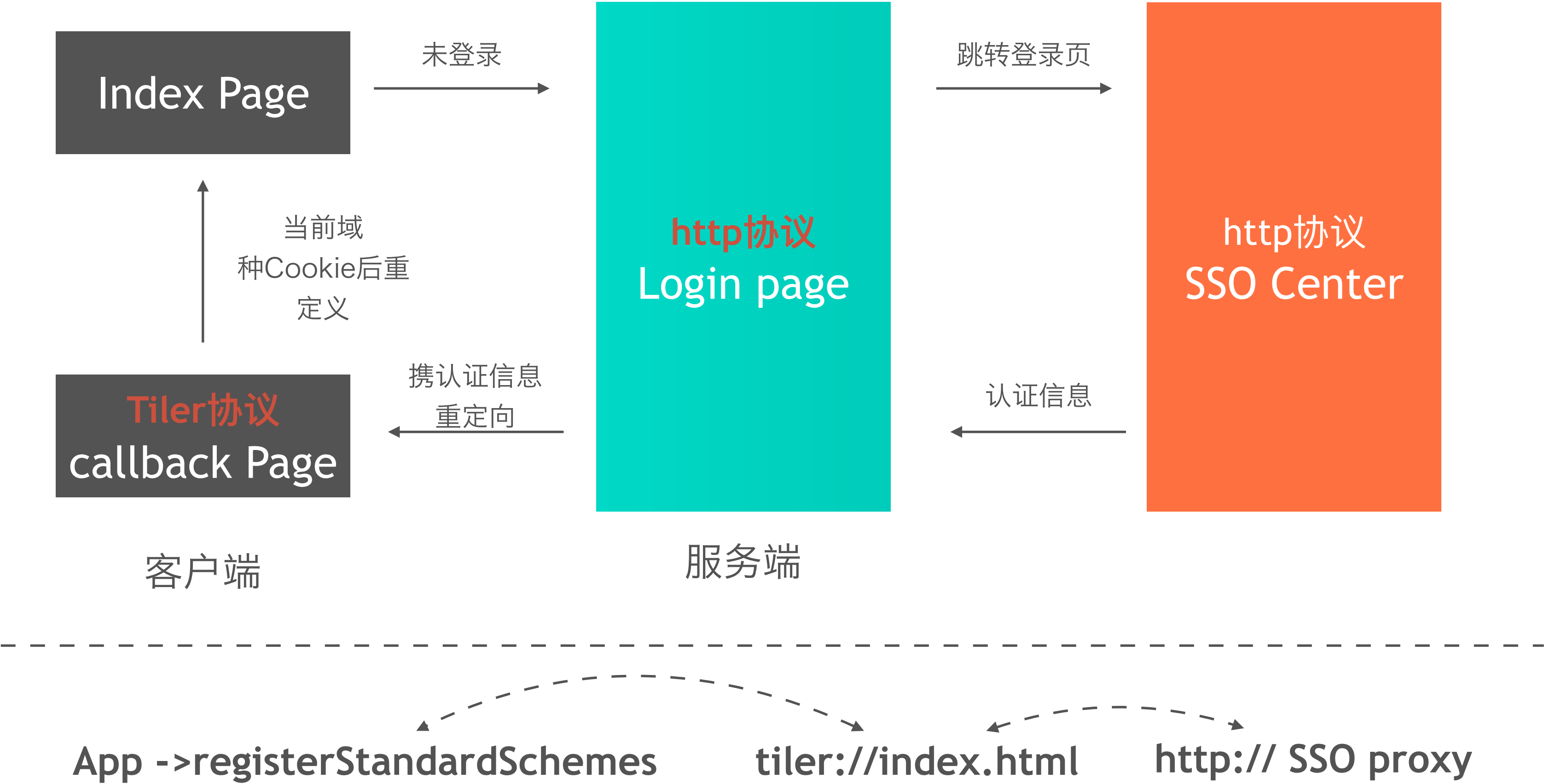


项目实践

#2如何实现登录认证

- Electron跨域如何解决?
- File协议与HTTP如何交互?
- 自定义协议是什么?

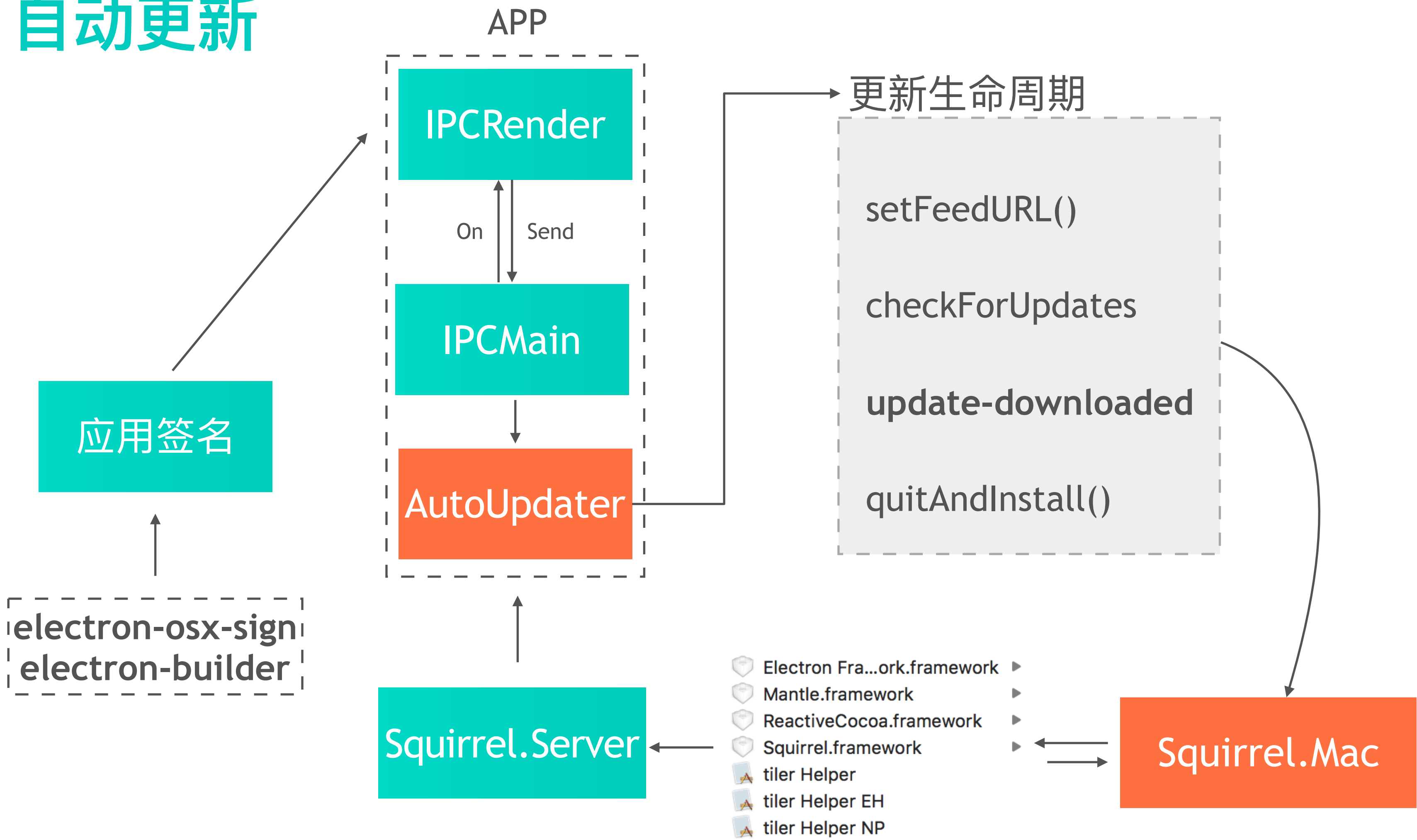
RFC3986



项目实践

#3如何完成签名、自动更新

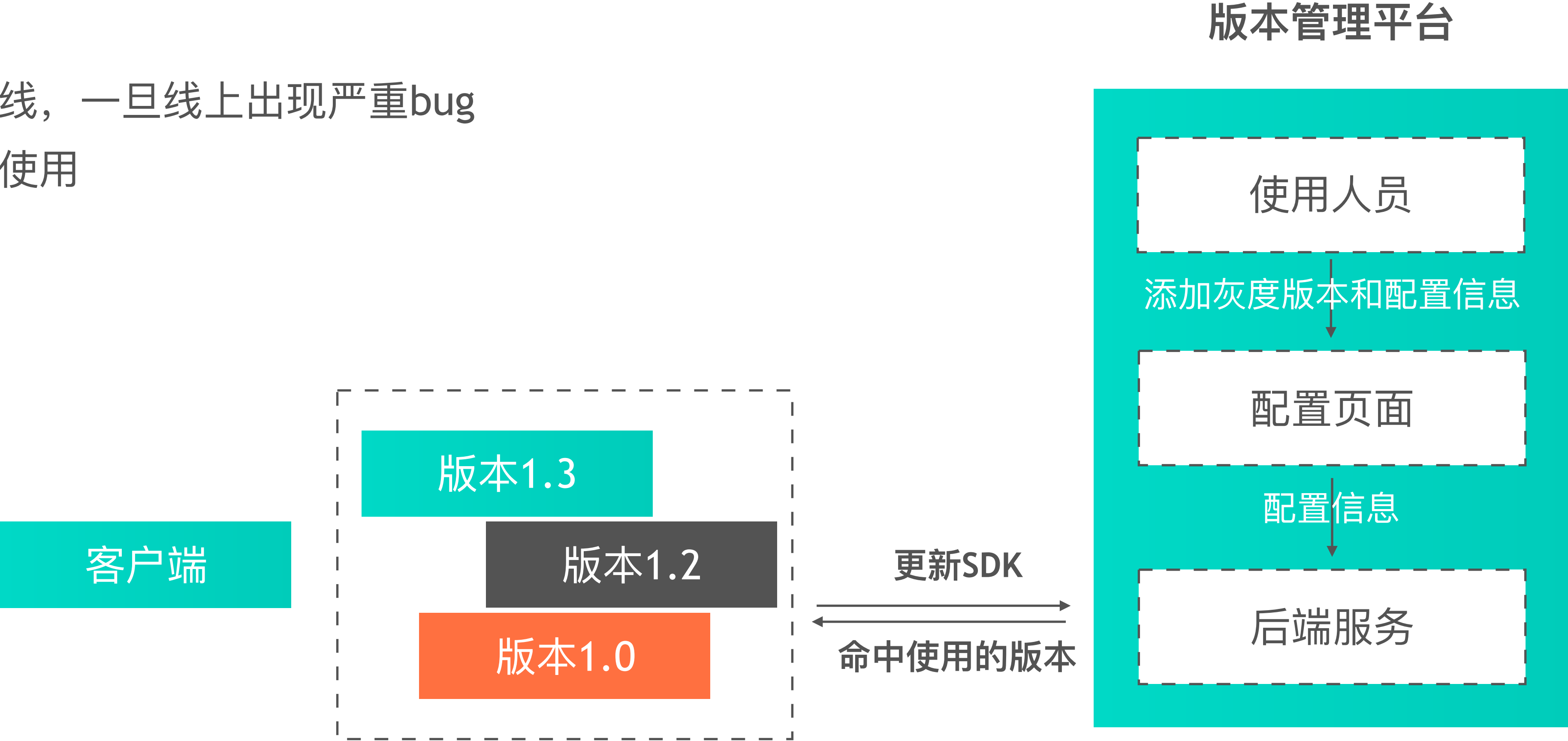
- 用户每次都要重新下载
- 下载的应用不被信任
- 网络请求被询问.....



项目实践

#3如何完成签名、自动更新

- 每次新版本上线，一旦线上出现严重bug
- 影响所有用户使用



项目实践

#3如何完成签名、自动更新

- 灰度发布
- 指定人员发布
- 指定版本发布

灰度发布动态配置项目设置

赵帅

灰度版本列表灰度设置

添加灰度版本

只匹配状态开启且在有效期内的版本，匹配优先级为：排序在前 > 排序在后

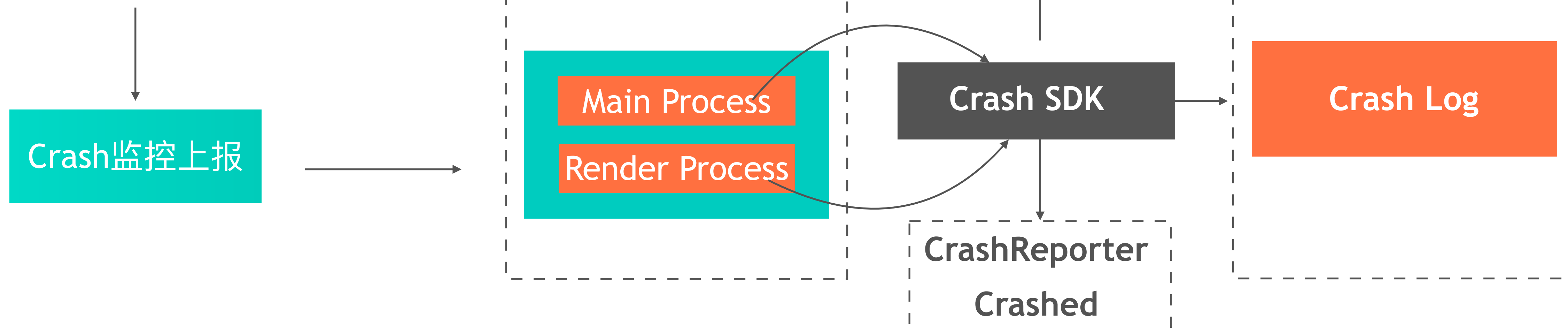
接口调试

灰度版本号	描述	状态	创建时间	操作
1.0.9	dev版本:1.0.9	<div></div>	2018-10-29 11:29	<div>其他设置复制更多</div>
1.0.8	dev版本:1.0.8	<div></div>	2018-10-11 15:51	<div>其他设置复制更多</div>
1.0.7	dev版本:1.0.7	<div></div>	2018-10-10 18:10	<div>其他设置复制更多</div>
1.0.6	dev版本:1.0.6	<div></div>	2018-10-10 11:36	<div>其他设置复制更多</div>
1.0.5	dev版本:1.0.5	<div></div>	2018-08-09 19:24	<div>其他设置复制更多</div>

项目实践

#4如何定位和收集问题

- 客户端版本多，用户崩溃信息收集困难
- 崩溃比率和出错版本，咨询用户，提供信息较少



项目实践

#4如何定位和收集问题

共5行

操作系统版本	请输入内容	进程类型	请输入内容	uid	请输入内容
uid	崩溃原因	版本号	崩溃时间		
47725	EXC_BAD_INSTRUCTION / EXC_I386_INVOP	1.8.7	2018-08-21 21:46:45.916		
2069853	EXC_BAD_INSTRUCTION / EXC_I386_INVOP	1.8.7	2018-08-06 12:02:23.709		
2069853	EXC_BAD_INSTRUCTION / EXC_I386_INVOP	1.8.7	2018-08-06 10:56:31.26		
2069853	EXC_BAD_INSTRUCTION / EXC_I386_INVOP	1.8.7	2018-08-06 10:55:49.74		

错误信息

Home

Crash

基本信息

id: 2736178

用户id: 47725

版本id: 221

内核版本: 1.8.7

启动时间: 1534858719651

崩溃时间: 1534859205916

运行时间: 486265

自定义信息:

guid: 8b5853b4-28c9-461c-acd7-4611bca93b8a

操作系统: darwin

操作系统版本: 17.7.0

崩溃原因: EXC_BAD_INSTRUCTION / EXC_I386_INVOP

下载崩溃日志: http://s3.meituan.net/v1/mss_de9bbaffb6644e81b118621ea8fc00

下载原始dmp: http://s3.meituan.net/v1/mss_de9bbaffb6644e81b118621ea8fc00

版本信息: 1.8.7

版本id: 221

业务: tiler-dev

- 01 瓦匠应用的介绍
- 02 桌面应用的技术选型
- 03 Electron Vue项目实践
- 04 桌面应用开发总结

开发总结

Electron开发和Web开发差异

	Electron应用	Web应用
设计	单窗口、多窗口	单页面、多页面
开发	html CSS JS Node.js OS API	html CSS JS
调试	主进程，渲染进程	渲染进程
构建	资源文件，安装包	资源文件
发布	Mac / Window / Linux	Nginx / CDN
关注点	进程通信，内存管理，版本管理，性能及Crash监控.....	兼容，DOM，组件、性能，

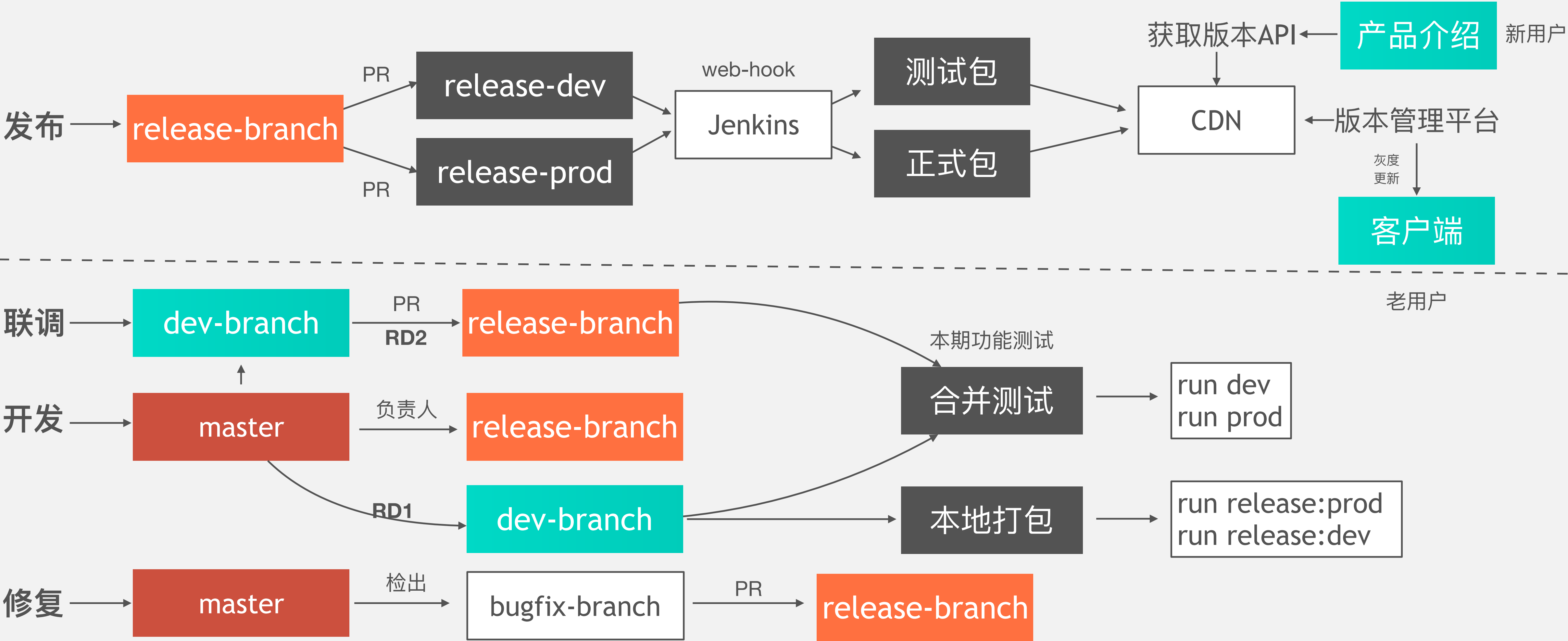
开发总结

Electron应用技术体系推荐

应用架构	前端框架	Vue	Vuex Vue-Router		UI 组件库
	能力SDK	更新 SDK	性能 SDK	监控 SDK	登录 SDK
	基础框架	Electron	Node RunTime		Native API
工程化	开发调试	Vue-devtools Electron –inspect			devtron
	打包构建	webpack	Electron-build		babel
	检查\测试	ESLint			Spectron + Mocha
基础设施		监控平台	版本管理平台	数据统计平台	DevOps平台

开发总结

Electron应用开发流程推荐



CODE A BETTER LIFE

一行代码 亿万生活



Thanks

邮箱: zhaoshuai05@meituan.com

微信: z416043658