

美团金融 前端工程化 最佳实践

分享人 田泱





田泱

美团金融服务平台 前端工程师

主要研究方向：前端工程化、离线化

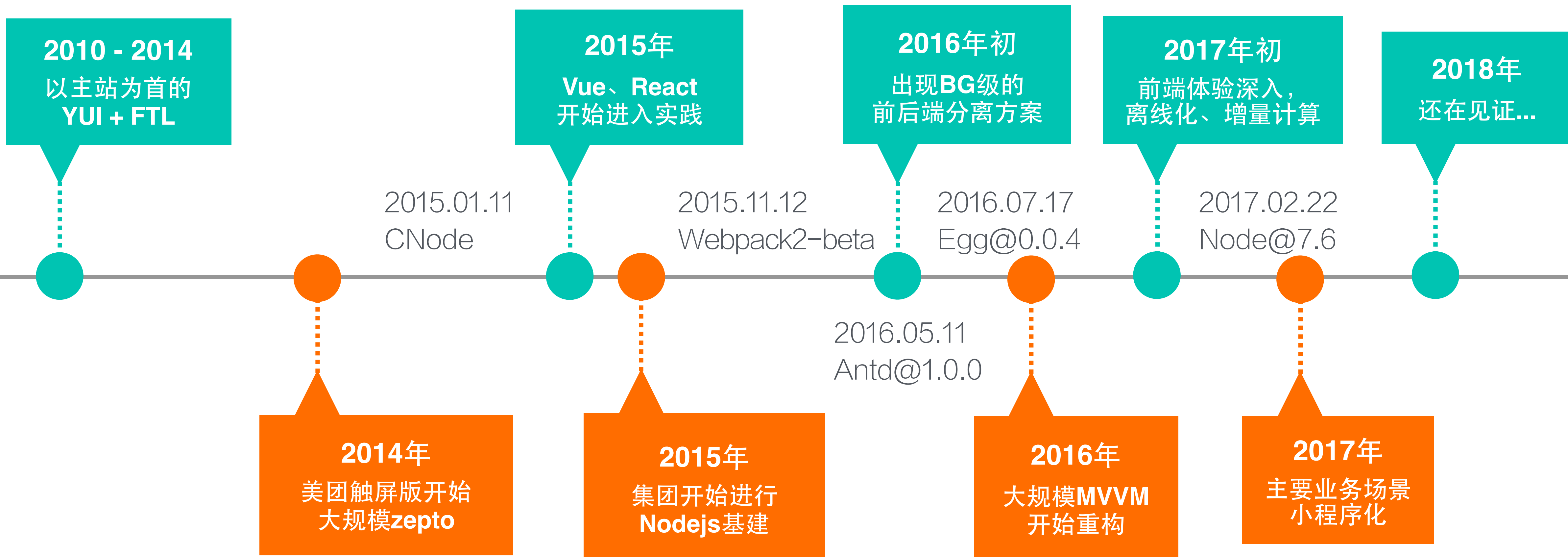
2017年1月，开始负责技术平台前端基建工作

2016年5月，加入金融服务平台，负责扫码付

2015年7月，加入美团

美团金融 前端工程化 最佳实践

开篇 - 回顾美团前端发展史 (个人视角)



美团金融 前端工程化 最佳实践

开篇 - 有没有银弹

Thanos?

新时代的

开放式、全自动、一站式、一体化

前端开发解决方案



美团金融 前端工程化 最佳实践

开篇

不讲什么

美团金融的前端解决方案
是什么，有多少功能，他怎么用

要讲什么

- 工程化就是一个脚手架吗？
- 做一个脚手架，要考虑哪些问题？
- 脚手架的问题，我们要注意什么？
- 脚手架的下一步是什么？

美团金融 前端工程化 最佳实践

工程化解决的是什么问题

PM: “你明天能不能上线”



美团金融 前端工程化 最佳实践

工程化解决的是什么问题

PM: “你明天能不能上线”

我: “我要...”

创建项目

框架选型

业务逻辑

配置打点

收集日志

接口代理

构建测试

申请服务

上线检查

“我用vue还是react?”

“我用不用eslint?”

“redux太复杂了，要不就别用了吧”

“好的，开始撸代码，3，2，1写完了”

“我还要，埋点，监控，联调，搭环境”

“... ..”

美团金融 前端工程化 最佳实践

工程化解决的是什么问题

PM: “你明天能不能上线”

我: “我要...”

创建项目

框架选型

业务逻辑

配置打点

收集日志

接口代理

构建测试

申请服务

上线检查

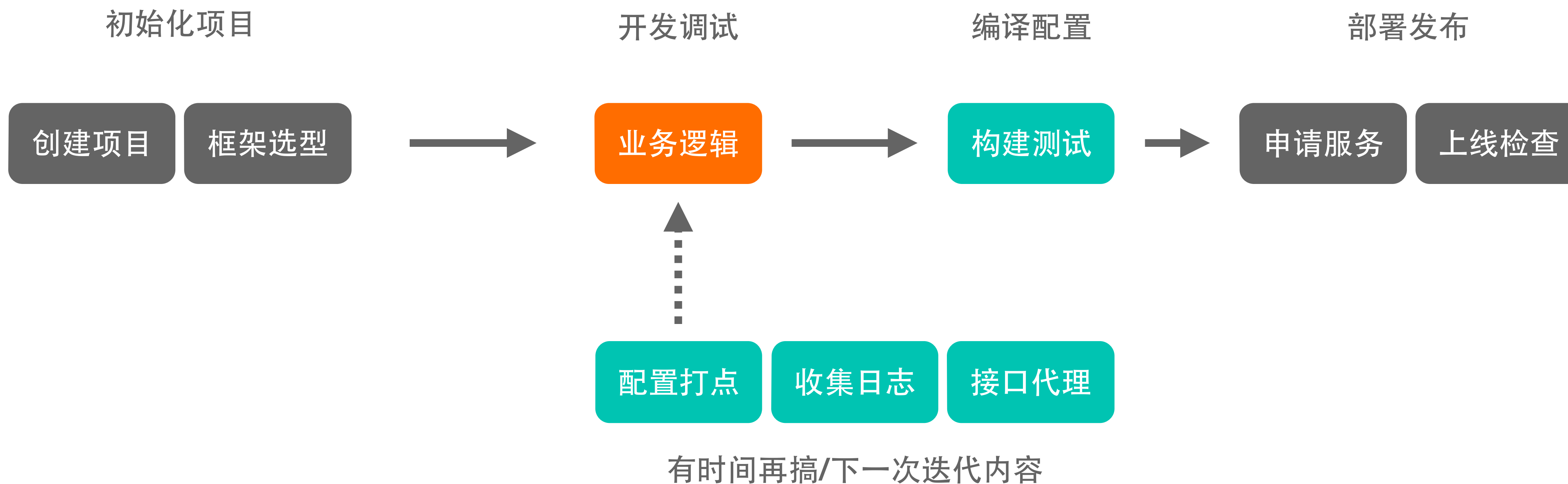
两天过后，

我: “好了我做不完，我们还是打一架吧”

美团金融 前端工程化 最佳实践

工程化也就这几件事

分析一下，有些是非必须的（重要不紧急）直接砍掉



美团金融 前端工程化 最佳实践

工程化就是一个脚手架

前端架构师 = 创建文件夹工程师

脚手架 = 工程模板

- 初始化项目 = CLI
- 开发调试 = Dev-Server
- 编译构建 = Webpack
- 部署发布 = release.sh



美团金融 前端工程化 最佳实践

工程化就是一个脚手架

“ 脚手架搞好了，业务团队去用吧，有问题找我。 ”

初始化项目

CLI

react/vue?
eslint?
redux/redux?

开发调试

业务逻辑

编译配置

Webpack

内置babel/eslint
配置最佳实践
版本升级

部署发布

Node Server:
生成发布脚本

纯静态: 对接CDN

“企业级”前端工程化解决方案

集团
服务



初始化项目

CLI
react/vue?
eslint?
redux/redux?

开发调试

PluginA PluginB

业务逻辑

PluginC PluginD

编译配置

Webpack
内置babel/eslint
配置最佳实践
版本升级

部署发布

Node Server:
生成发布脚本

纯静态: 对接CDN

美团金融 前端工程化 最佳实践

一劳永逸 的脚手架

“ 脚手架的约束形成了标准，保证团队可以在规范内工作 ”

组织结构变动、业务形态变化 => 约束力 变成了 限制

基础设施的变化碾压了团队信心 => 限制变成了 累赘

缺乏信心，各自为战 => 累赘 无法流通

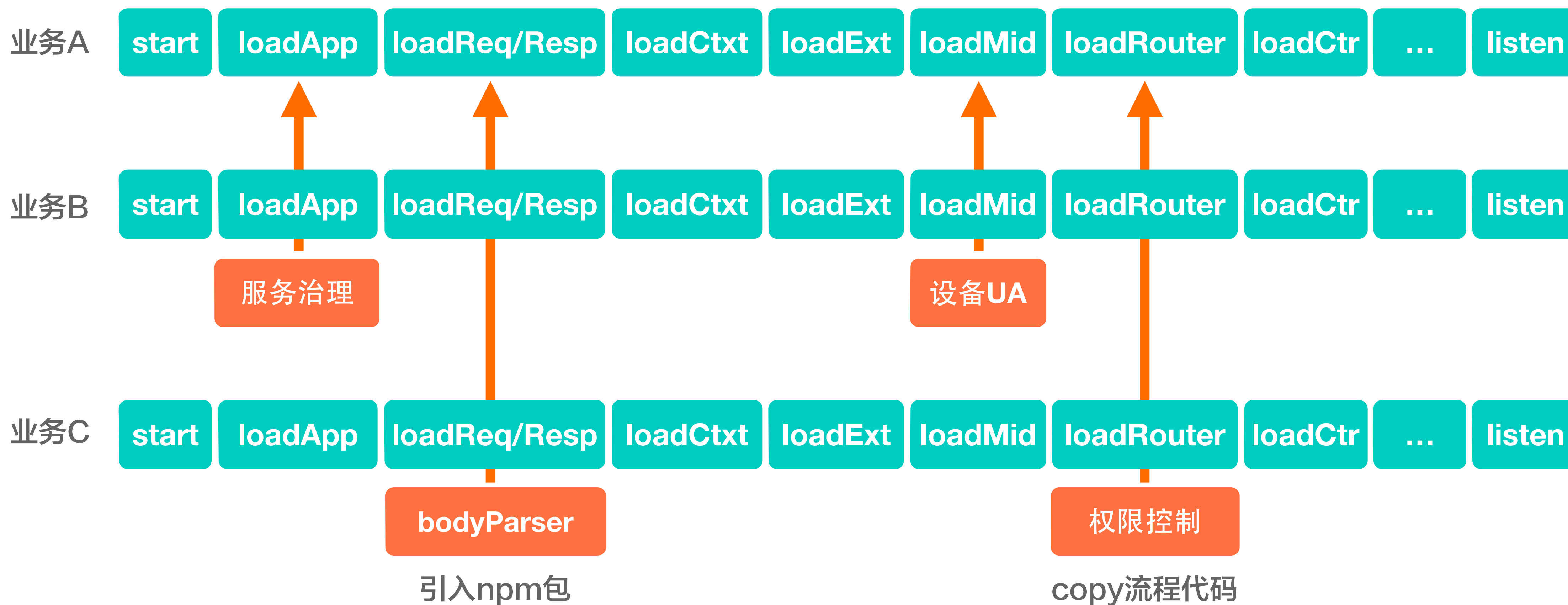
团队变成天花板 => 诉求无法得到满足

缺少了 **灵活性**，也就少了 **技术成长空间**



美团金融 前端工程化 最佳实践

脚手架如何提供灵活性?

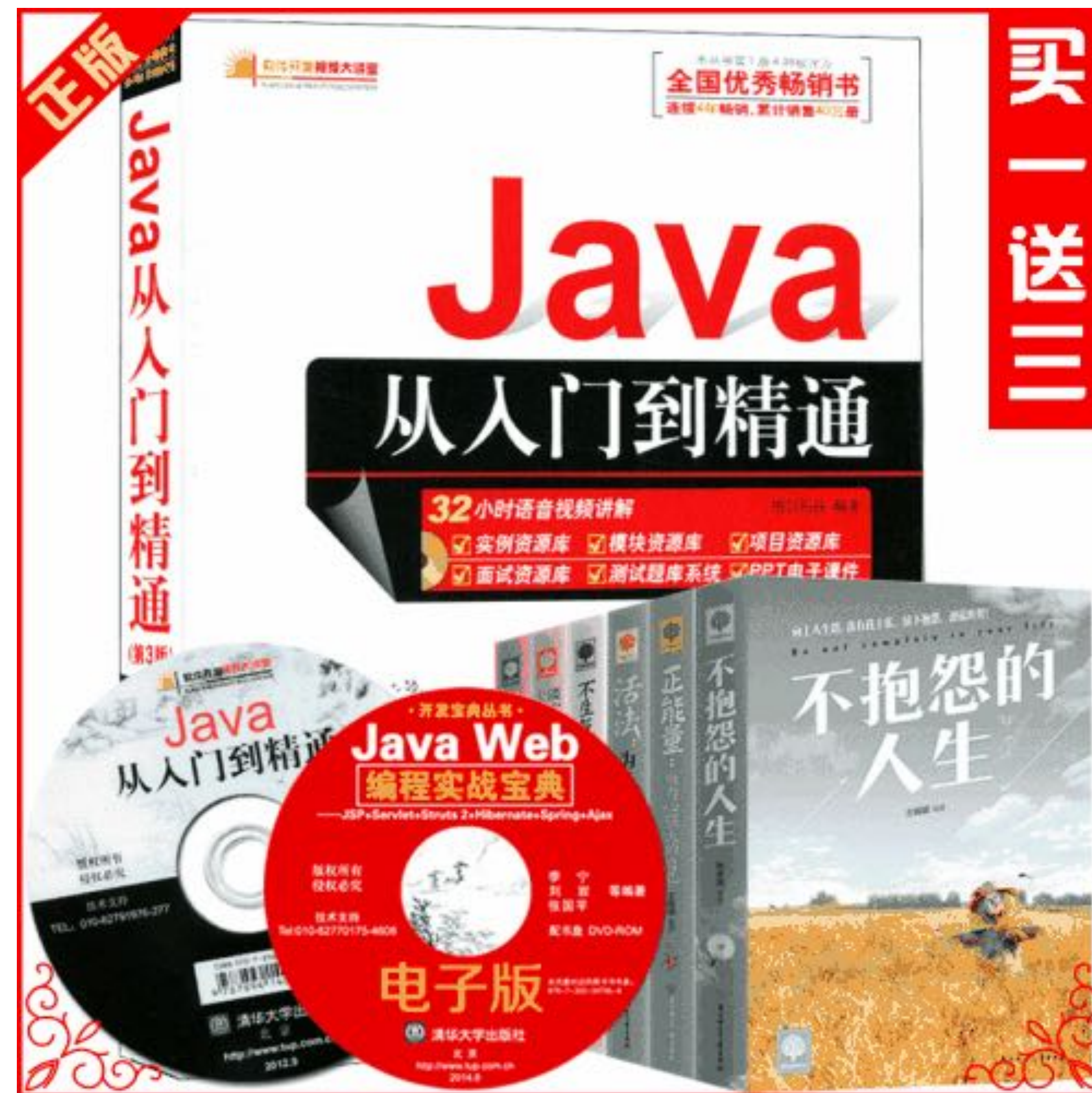


美团金融 前端工程化 最佳实践

我们能不能获得别人的能力？

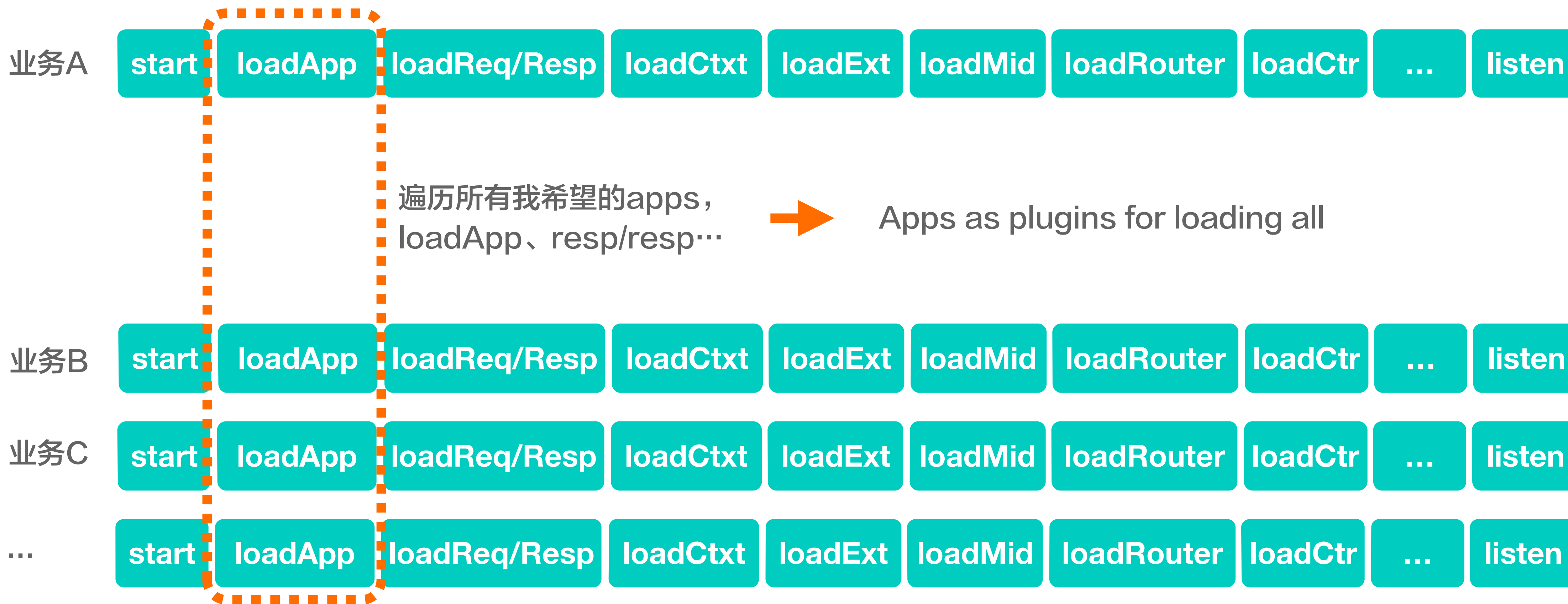
我想一睁眼，就能：

- 瞬间移动
- 影分身
- 隔空取物
- 冻结时间



美团金融 前端工程化 最佳实践

大刀阔斧改Dev-Server



遍历所有我希望的apps,
loadApp、resp/resp...

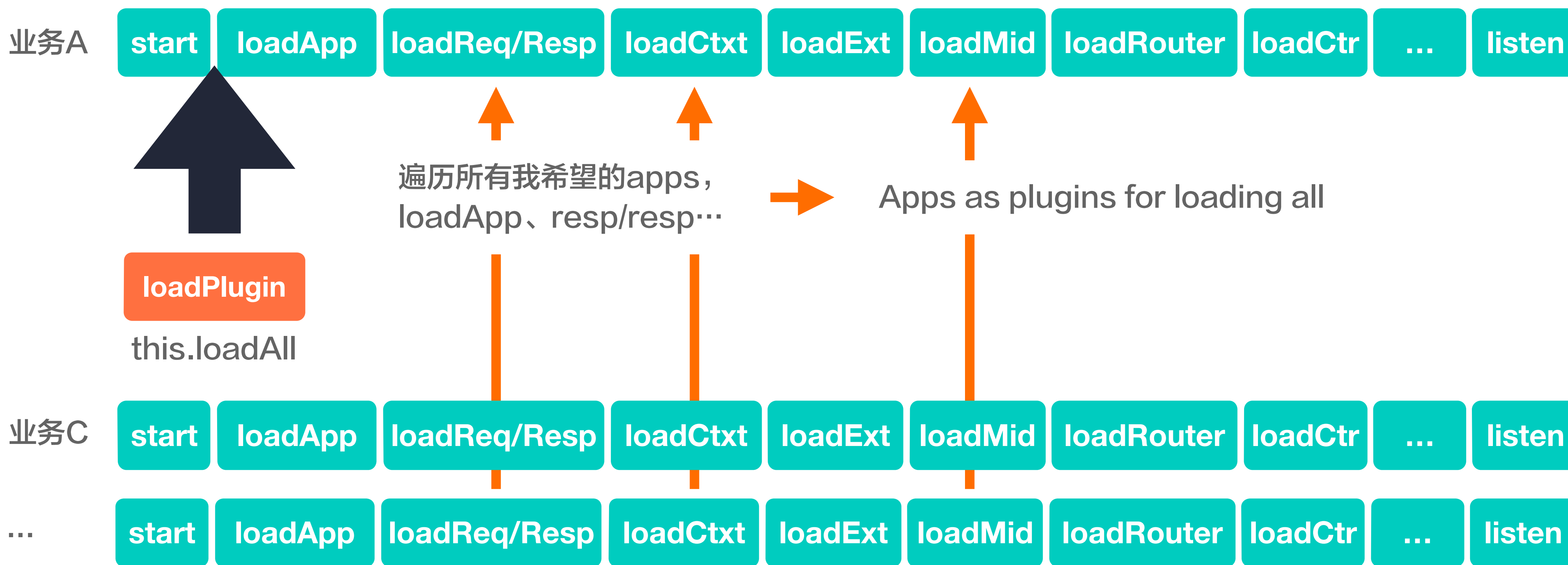


Apps as plugins for loading all

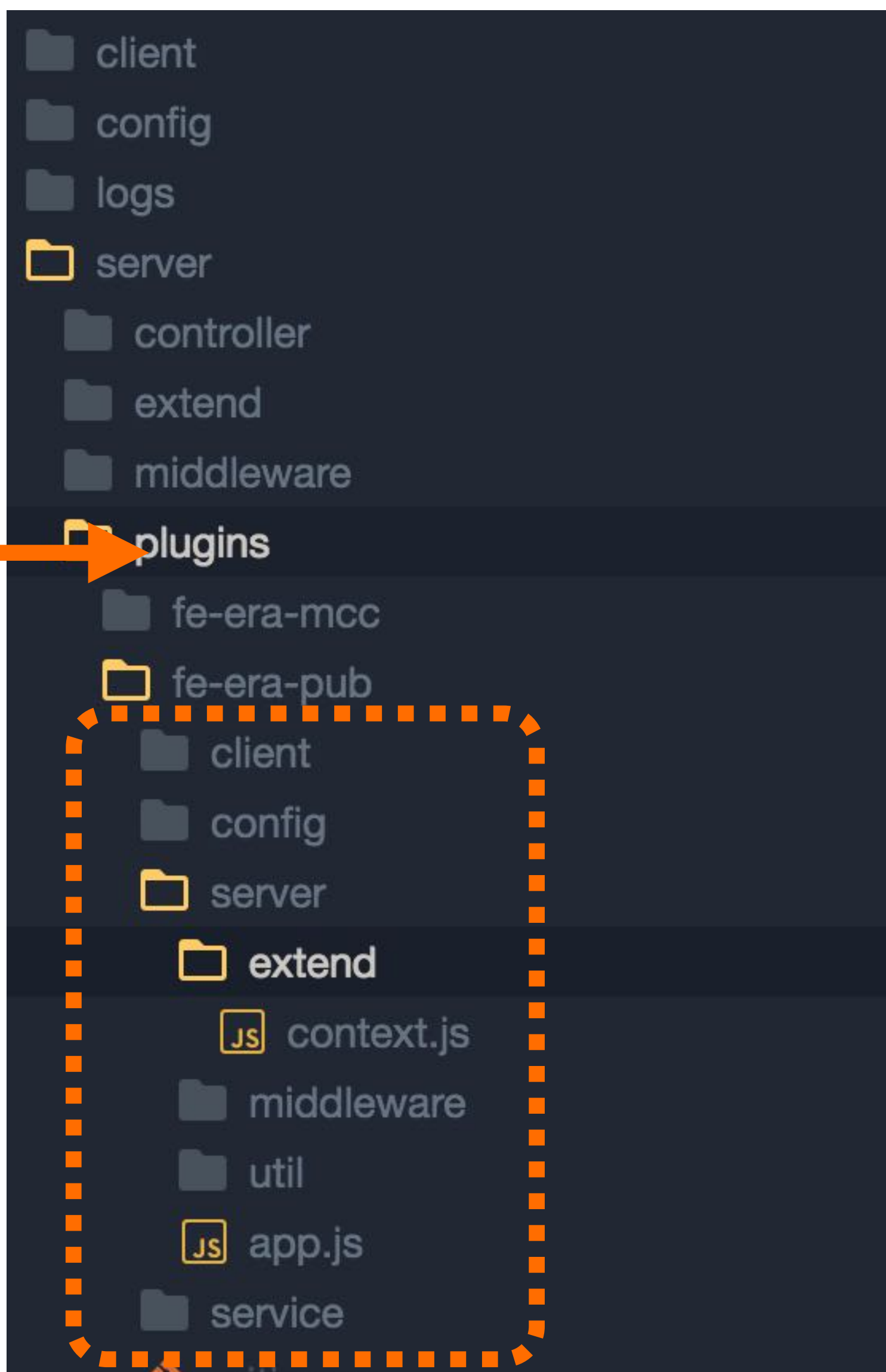
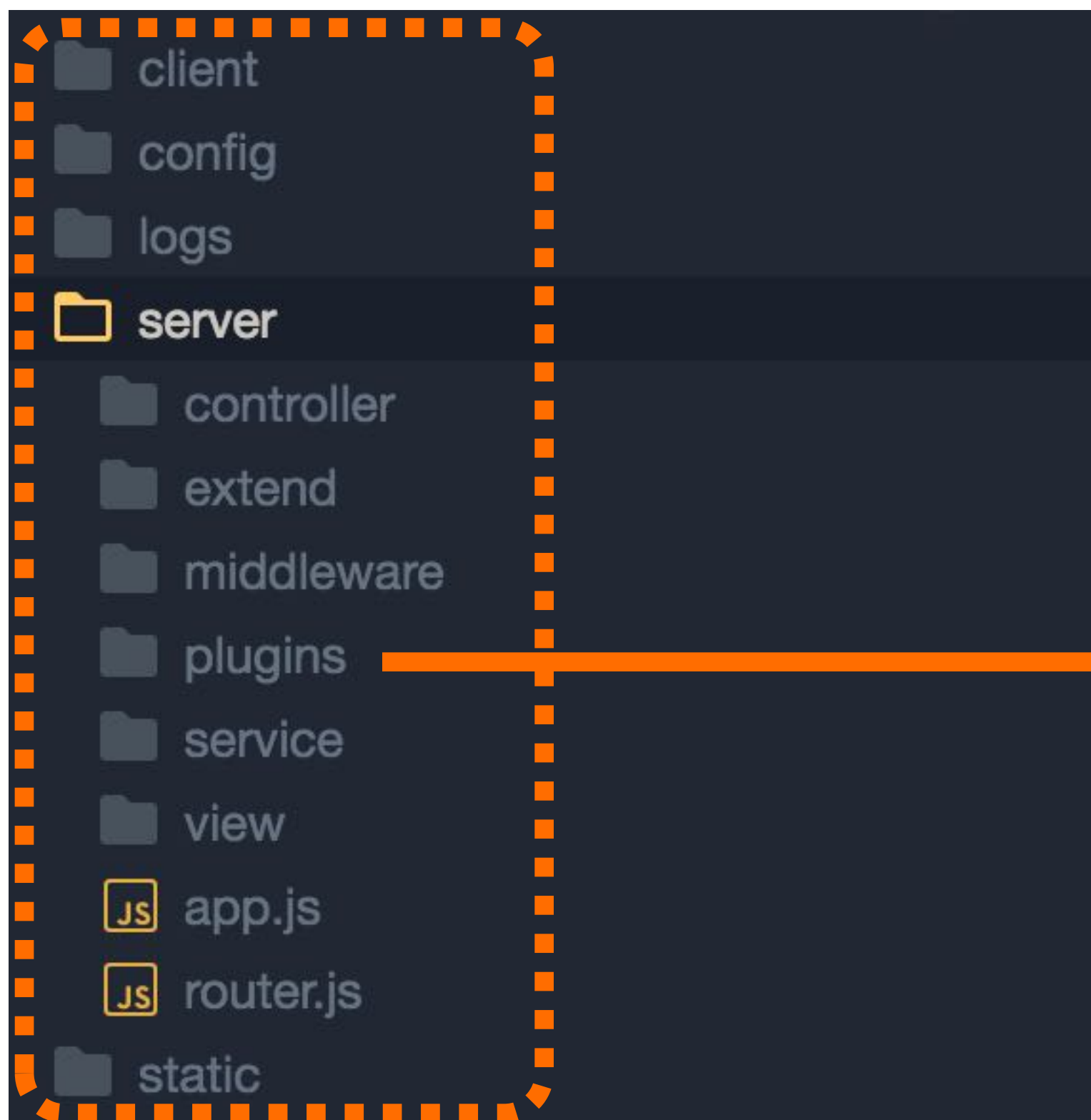
```
appList.push([a, b, c]).map(app => loadApp)
```


美团金融 前端工程化 最佳实践

大刀阔斧改Dev-Server



Everything is plugin



App = Plugin

- 业务B作为业务A的插件
- 目录结构可插拔
- 能力流程复用

美团金融 前端工程化 最佳实践

前端工程化的收益

脚手架的收益



1. 精力收敛

美团金融 前端工程化 最佳实践

前端工程化的收益

业务能力插件化 的收益



1. 精力收敛

2. 能力托管



美团金融 前端工程化 最佳实践

前端工程化的收益

业务能力插件化 的收益



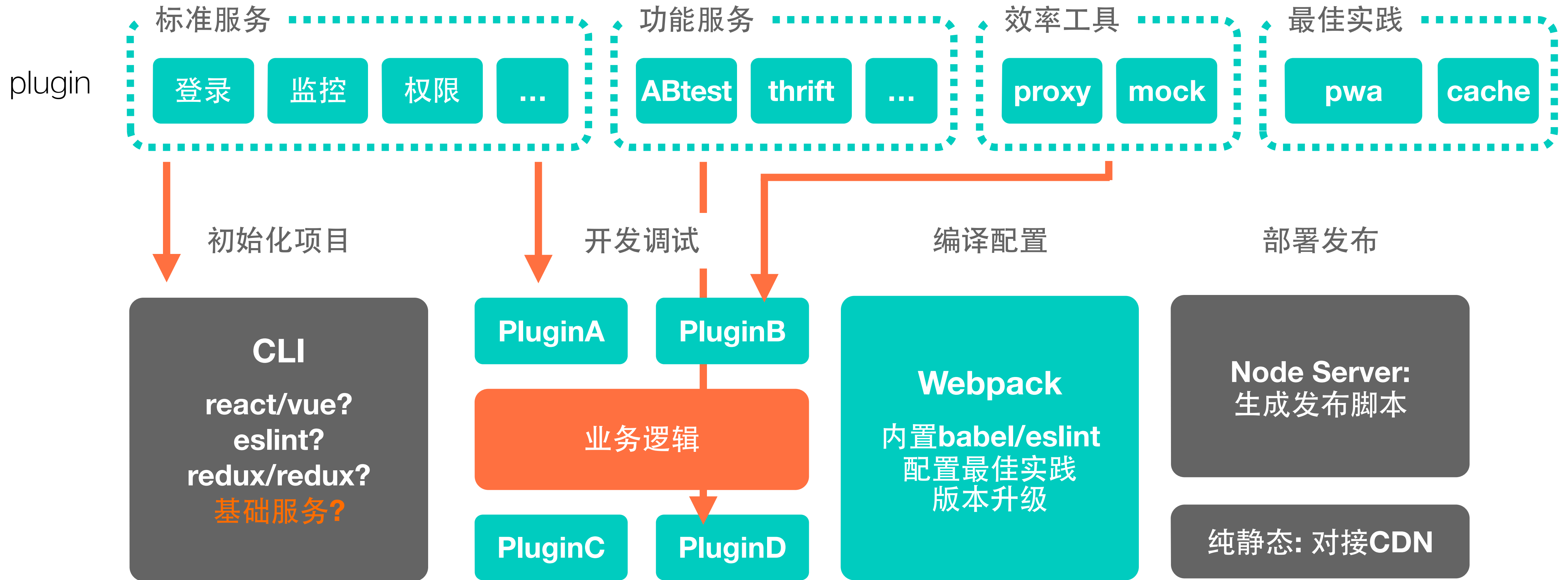
1. 精力收敛

2. 能力托管

3. 便于复用



终于等到你，Everything is plugin



提供插件化的能力，引导业务能力自增

美团金融 前端工程化 最佳实践

如何论证有效性? - 工程师两大自我安慰

“ 没人反馈问题，代表没问题。 ”

- 查阅业务邮件
- 参加业务周会
- 学习业务Case study

“ 提升人效100% ”

- 加强流程管理
- 监控流程数据
- 回收研发数据

美团金融 前端工程化 最佳实践

研发数据中的故事

2: 4: 1: 1: 3

周一到周五的输出效率比例

17%

前端编译EB比 (error / build)

70%

开发过程中, 服务启动失败率

Top3

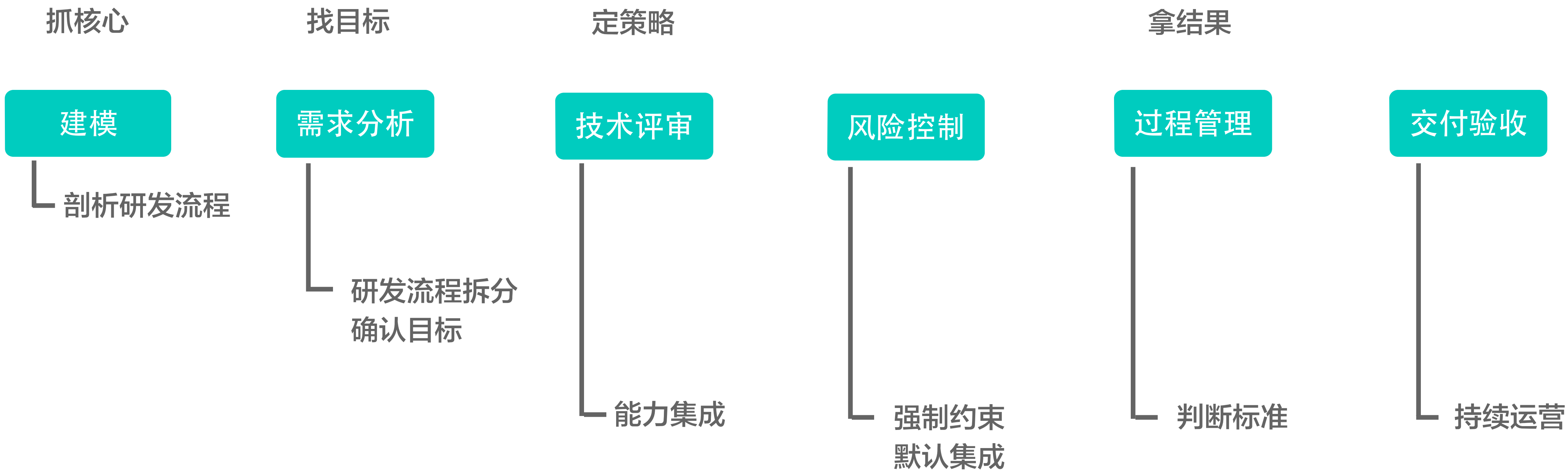
代码检查: Eslint Warning

模板编译失败: Error compiling template

找不到模块: Can't resolve Module

美团金融 前端工程化 最佳实践

如果你要做工程化，要怎么做？



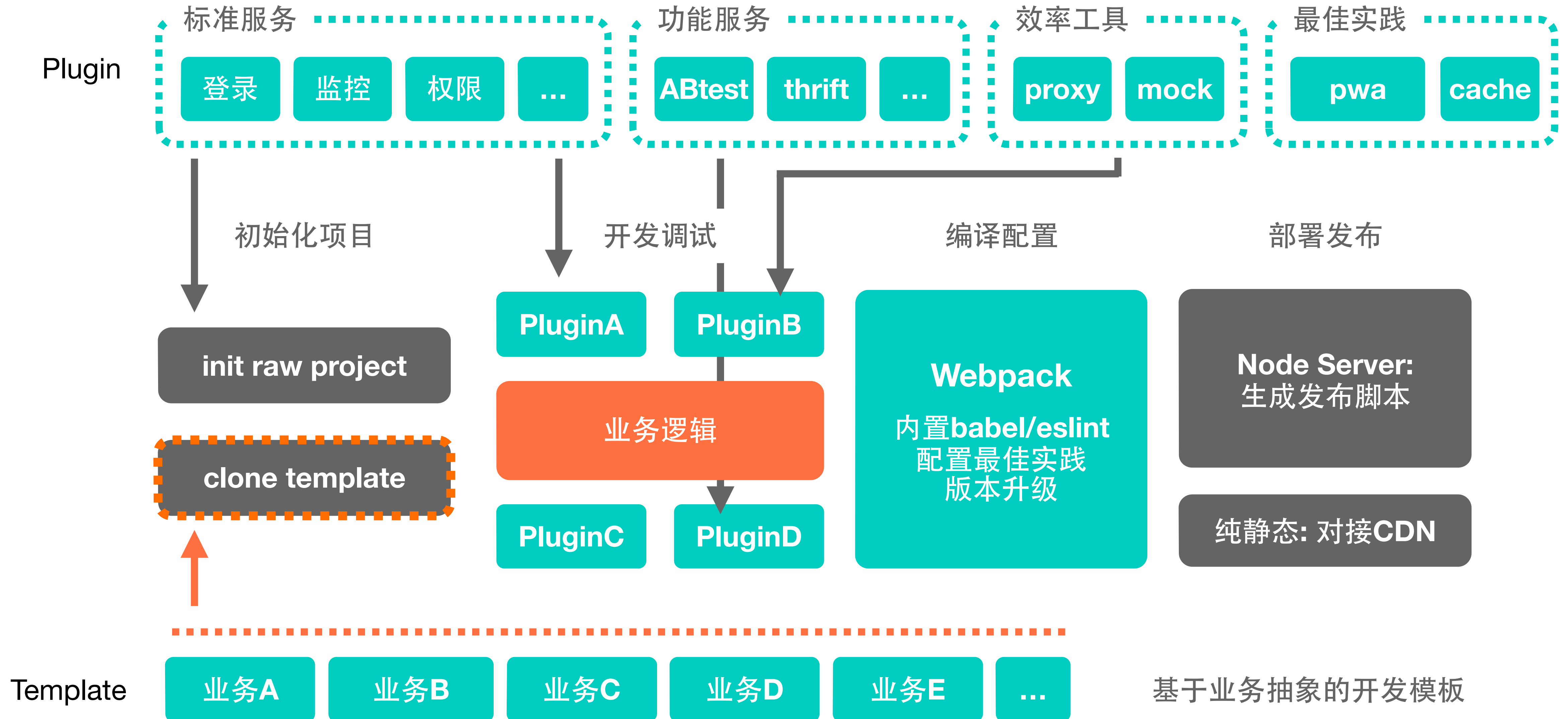
美团金融 前端工程化 最佳实践

回顾与总结

- **Q: 前端工程化，就是一个脚手架吗？**
A: 不是。脚手架只是工程化的第一步
- **Q: 前端工程化，要考虑哪些问题？**
A: 同时需要 **约束力** 和 **灵活性**，缺乏灵活性的脚手架无法长远
- **Q: 前端工程化，我们还需要注意什么？**
A: 提供 **业务能力自增** 和 **效果回收反馈**

工程化的下一步是什么？ DevOps? GUI? CMS?

继续加速，业务与基建的联动



基于业务抽象的开发模板

美团金融 前端工程化 最佳实践

前端工程化下一步 的思考

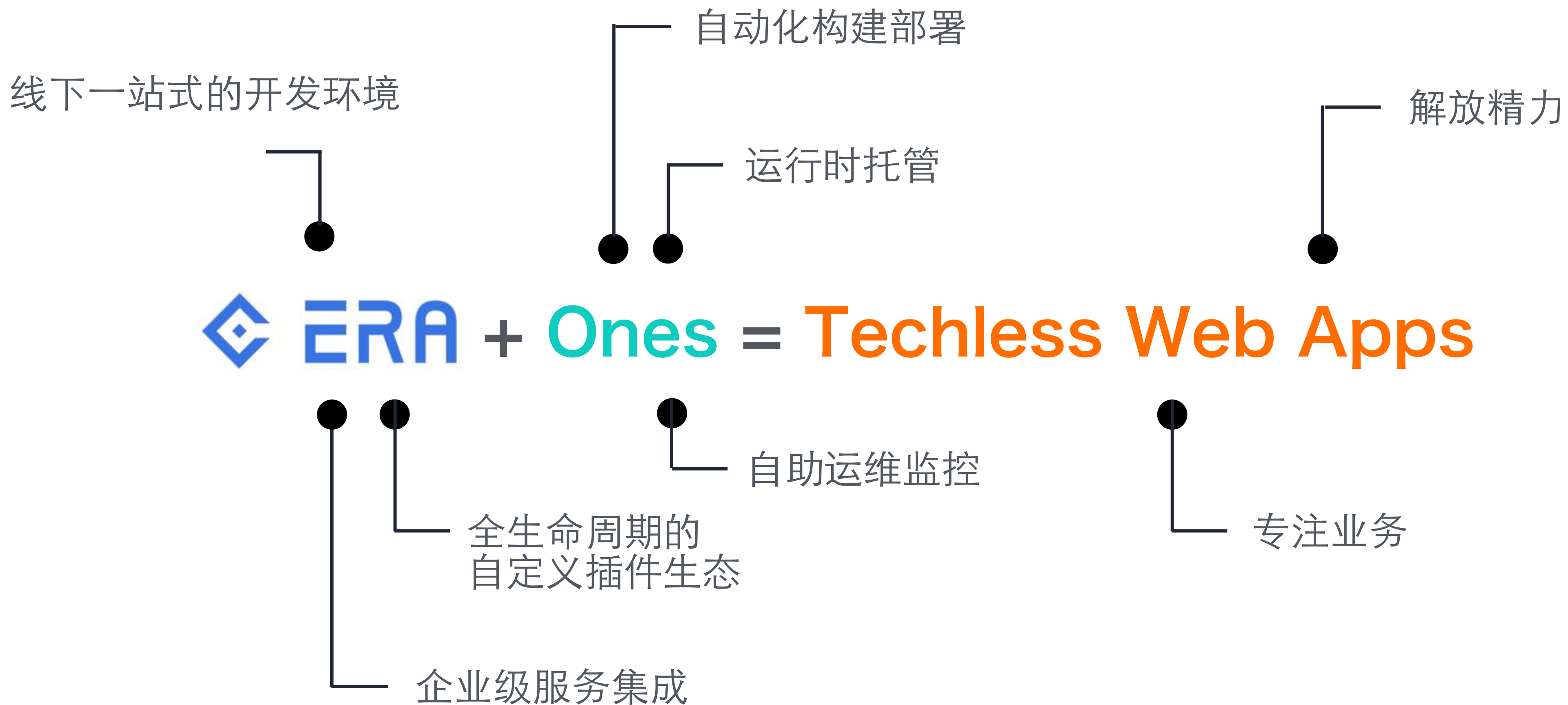
“ 多形态业务场景的精准匹配，快速进入开发 ”



美团金融 前端工程化 最佳实践

我们在做的事情

“ 让开发变得更简单。 ”



Q&A





扫码关注美团点评技术团队公众号
获取最 IN 的技术资讯

招聘：高级前端工程师 / 前端技术专家
邮箱：tianyang02@meituan.com