冯子龙

来自考拉前端公共技术组

考拉公技组发起人

主要负责考拉基建方面以及通用解决方案的落地
考拉小程序框架发起者，mpregular、megalo 核心开发
...

GitHub：https://github.com/fengzilong

NUT - born for microfrontends

by fengzilong @2019/06/29

说到微服务

有哪些优点？

1

编程语言 / 技术框架 无关

比如 A 服务用 Java, B 服务用 Python

技术选型灵活、技术栈迁移更平滑

2

独立开发、测试、部署

加速构建，容错

当 前端 遇上 “微服务”

微前端

前端 单体 到 微前端

定义

The term **Micro Frontends** first came up in ThoughtWorks Technology Radar at the end of 2016. It extends the concepts of micro services to the frontend world. The current trend is to build a feature-rich and powerful browser application, aka single page app, which sits on top of a micro service architecture. Over time the frontend layer, often developed by a separate team, grows and gets more difficult to maintain. That's what we call a Frontend Monolith.

The idea behind Micro Frontends is to think about a website or web app as **a composition of features** which are owned by **independent teams**. Each team has a **distinct area of business** or **mission** it cares about and specialises in. A team is **cross functional** and develops its features **end-to-end**, from database to user interface.

将微服务的概念延伸到了前端领域
吸收了微服务的优点

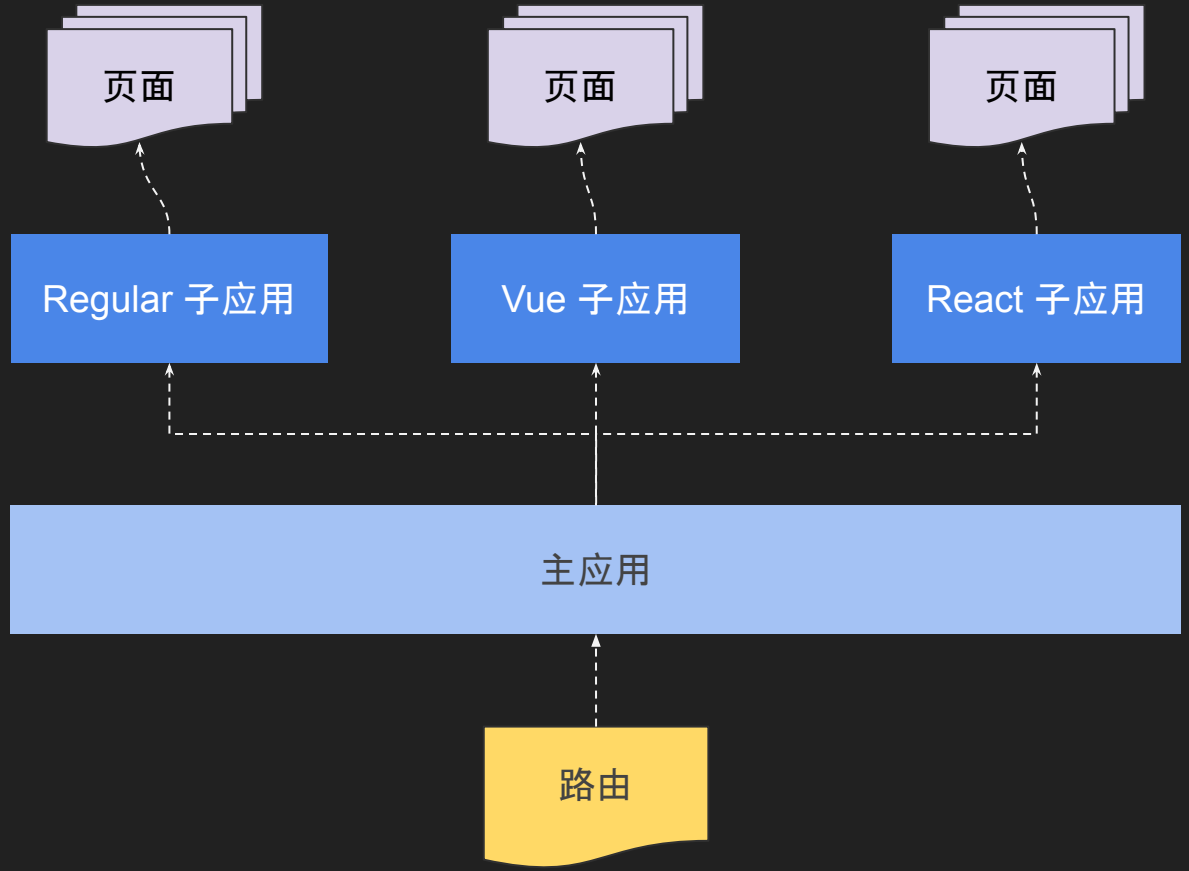
多框架 / 语言支持

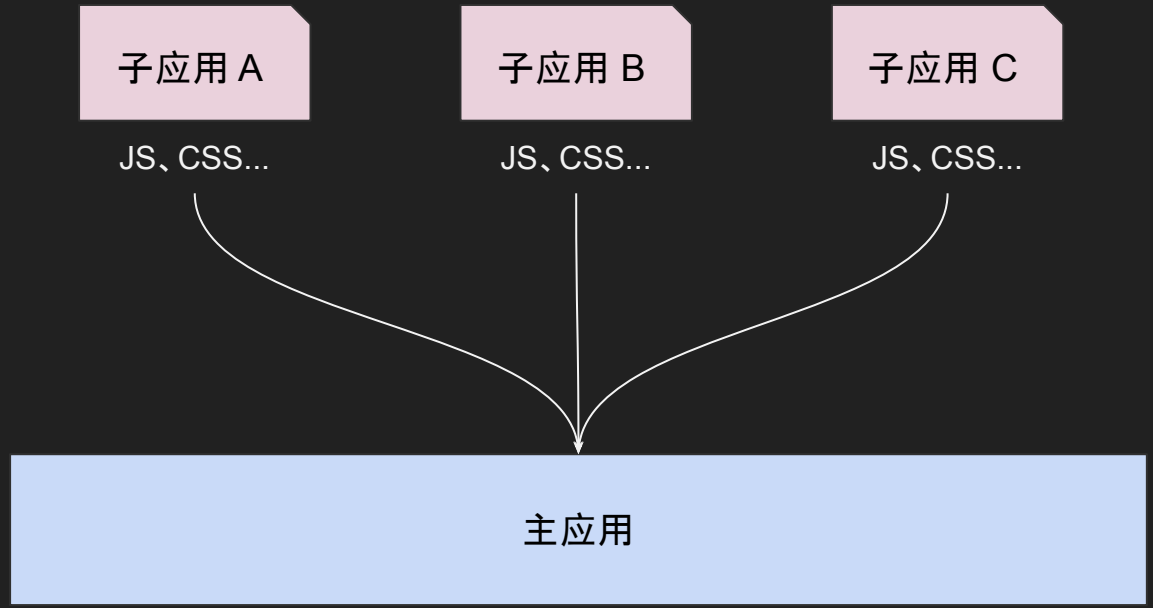
Vue、React、Markdown、Vanilla JS

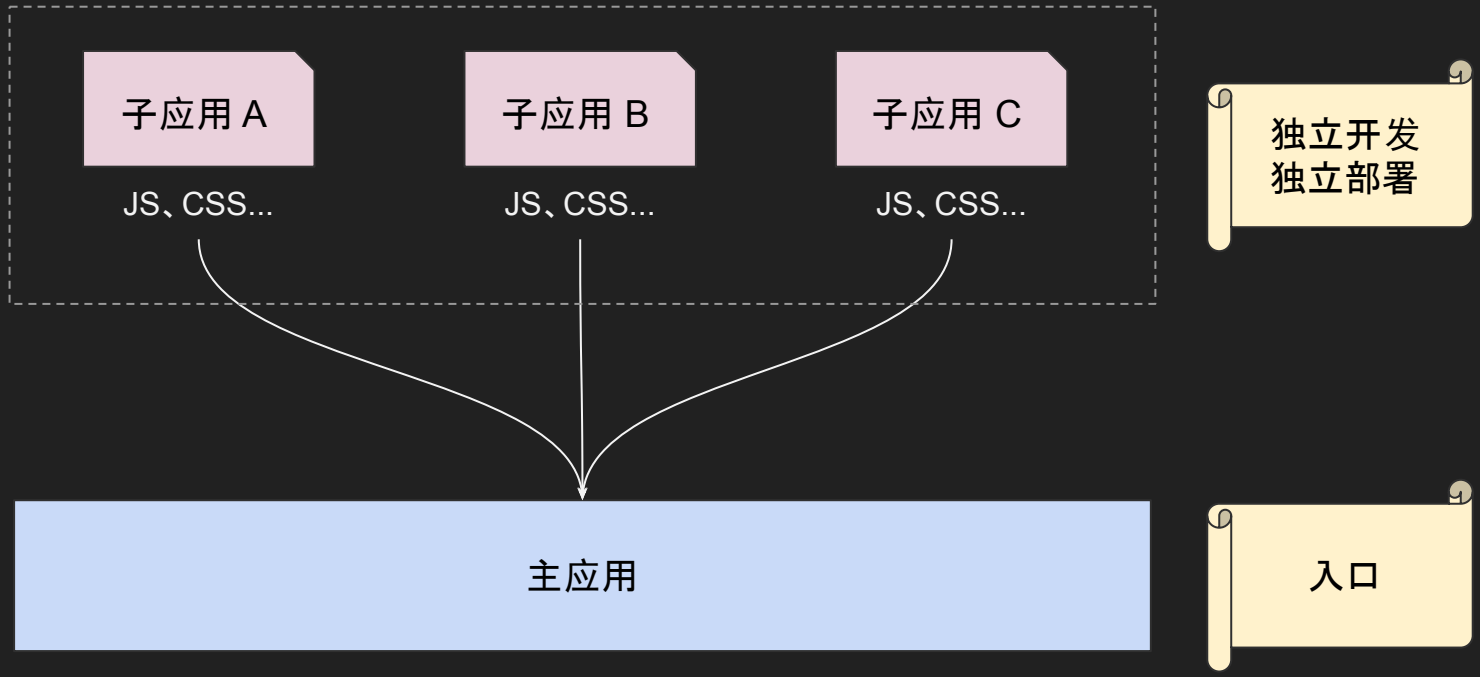
more...

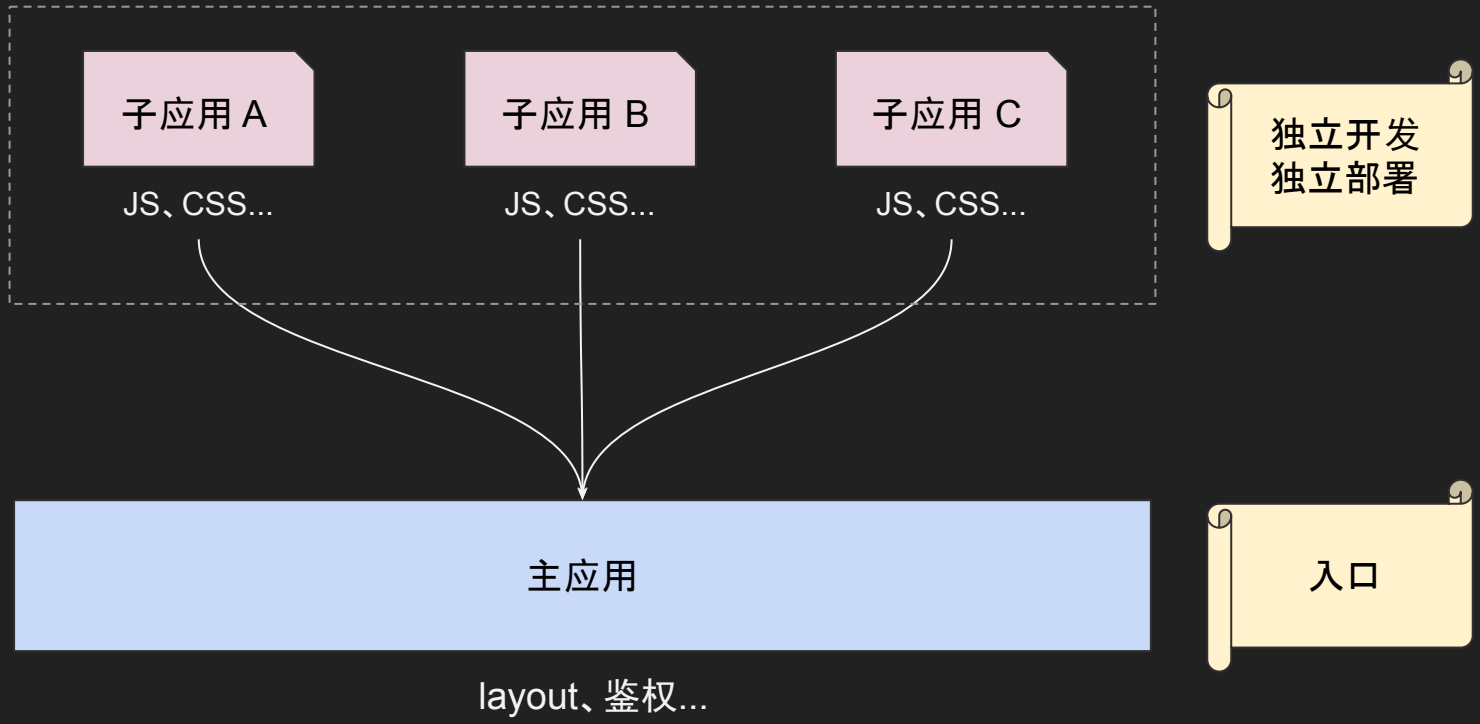
独立开发、部署

代表：Single-SPA / mooa


页面
页面
页面
Regular 子应用
Vue 子应用
React 子应用
主应用
路由


子应用 A
子应用 B
子应用 C
JS、CSS...
JS、CSS...
JS、CSS...
主应用


子应用 A
子应用 B
子应用 C
JS、CSS...
JS、CSS...
JS、CSS...
独立开发
独立部署
主应用
入口


子应用 A
子应用 B
子应用 C
JS、CSS...
JS、CSS...
JS、CSS...
独立开发
独立部署
主应用
入口
layout、鉴权...

关键词提取

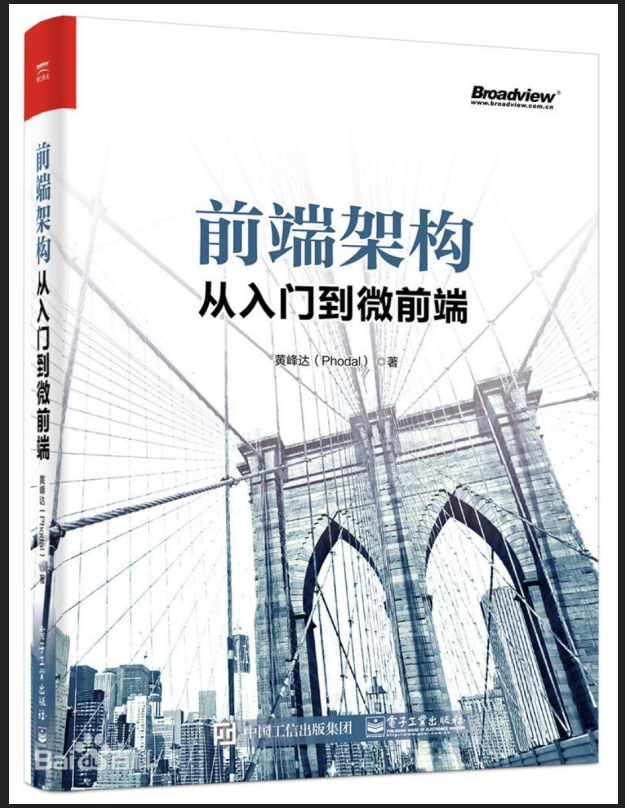
框架解耦 / 拆分 / 聚合

应用 / 模块拆分（架构上）

解耦，独立开发部署，加速编译，技术选型灵活

应用 / 模块聚合（使用上）

做到最后可能就变成了 平台化，如 小程序 / App Store...



phodal

sorrycc

美团技术团队

kuitos

蚂蚁在 GMTC 2019 评价 **微前端**

蚂蚁在 GMTC 2019 评价 **微前端**

在生产环境有 大量应用

关键词：技术架构、分工协作、组织架构

直观了解

compose

来自微前端的挑战

多框架并存方案

样式隔离

静态资源

加载性能

公共runtime、公共依赖

NUT：Born for microfrontends

为微前端而生

创建 NUT 应用有多简单

演示

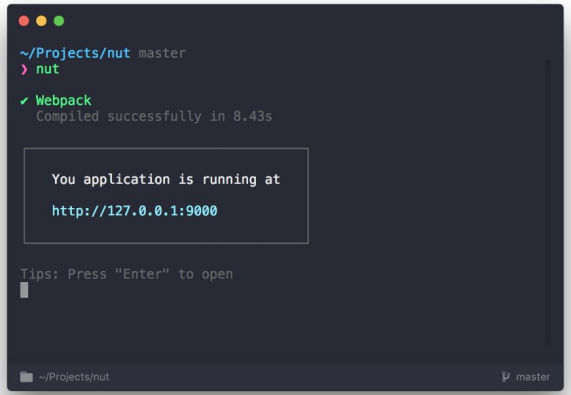
（本地 + 在线）

NUT 的插件体系

Layout

NUT，中后台系统的前端解决方案。

从广义的角度来说，nut 是一个静态站点生成工具。但是他本身提供的诸多特性，更专注于提供一套中后台系统的解决方案



引导

bilibili 创作中心

投稿

首页管理

内容管理

数据中心

粉丝管理

互动管理

收益管理

创作学院

个人设置

申诉管理

视频管理　专栏管理　音频管理

搜索稿件

创作中心开启投稿入口 1 / 6

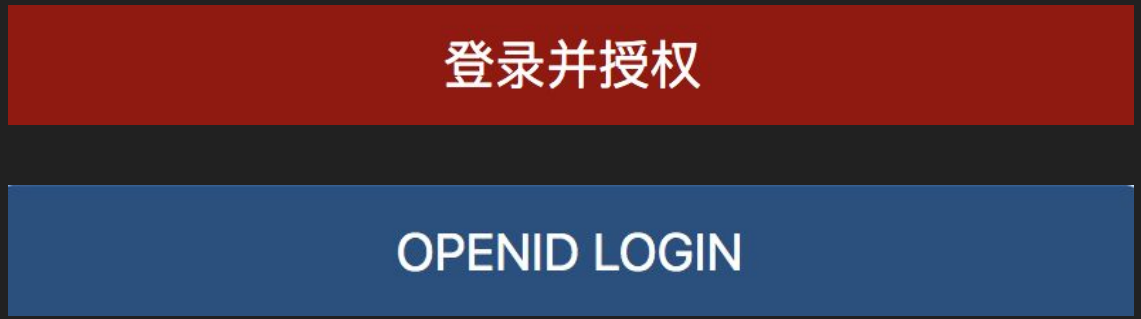
视频、音频、专栏、相簿投稿入口统一移动到创作中心“投稿”，支持鼠标移动到按钮上的悬浮按钮选择、同时也有顶部菜单可以直接切换～

下一步　朕知道了

没有投过一个稿件(ˉ▽ˉ)

立即投稿

登录鉴权



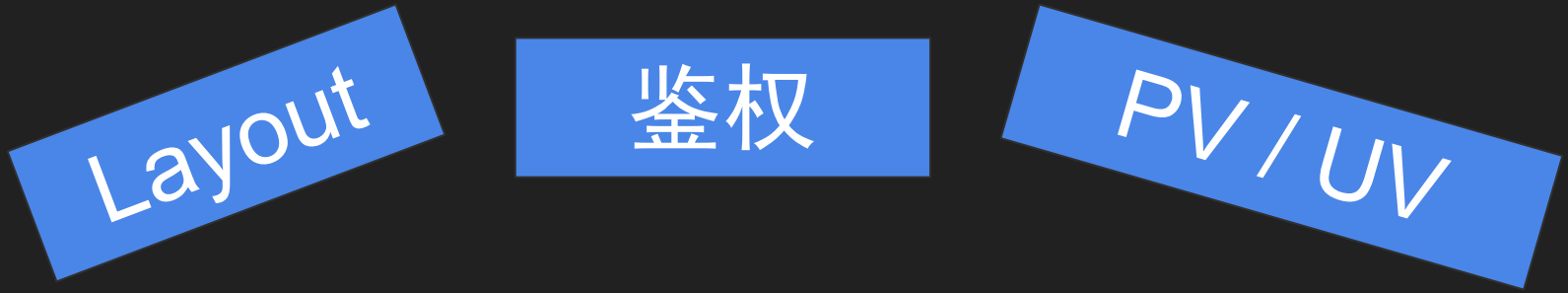
GitLab 授权登录

...

理论上

非 pages 逻辑 -> 插件

这种设计也是实施 微前端 的前提



重复的工作为什么要继续重复？

“AOP”

基于 NPM 包的 require
基于 NUT 插件的 enhance

基于 NPM 包的 require
基于 NUT 插件的 enhance

学习成本
使用成本
移除成本

原理剖析

多框架支持

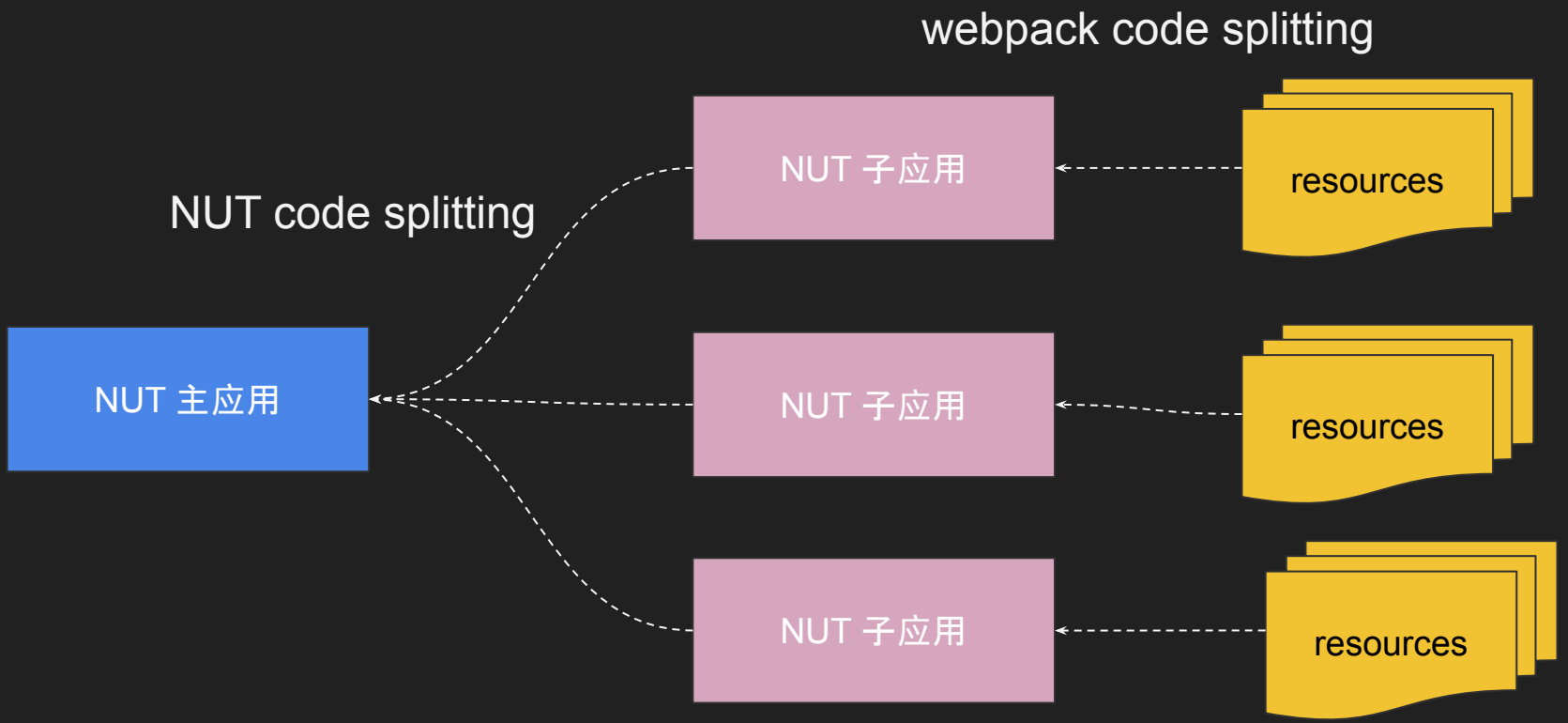
```
mount( node ) {
  node.innerHTML = source
},

unmount( node ) {
  node.innerHTML = ''
},
```

重新定义生命周期
抹平框架差异

微前端模式

```
if ( window.nutJsonp ) {
  nutJsonp( {
    pages,
    config,
    routes,
  } )
}
```


webpack code splitting
NUT code splitting
NUT 主应用
NUT 子应用
NUT 子应用
NUT 子应用
resources
resources
resources

NUT 相关

GitHub 仓库

文档站点

Q&A

谢谢观看