微前端之背景和实践(1)

课程大纲

微前端最早于2016年在Micro-Frontends被提出，并建立了早期的微前端模型。微前端的命名和能力和微服务有类似之处，微服务与微前端，都是希望将某个单一的单体应用，转化为多个可以独立运行、独立开发、独立部署、独立维护的服务或者应用的聚合，从而满足业务快速变化及分布式多团队并行开发的需求。如康威定律(Conway's Law)所言，设计系统的组织，其产生的设计和架构等价于组织间的沟通结构；微服务与微前端不仅仅是技术架构的变化，还包含了组织方式、沟通方式的变化。微服务

与微前端原理和软件工程，面向对象设计中的原理同样相通，都是遵循单一职责(Single Responsibility)、关注分离(Separation of Concerns)、模块化(Modularity)与分而治之(Divide & Conquer)等基本的原则。

本期课程，将会从微前端的背景和意义说起，讲述微前端解决的问题以及如何实现微前端。

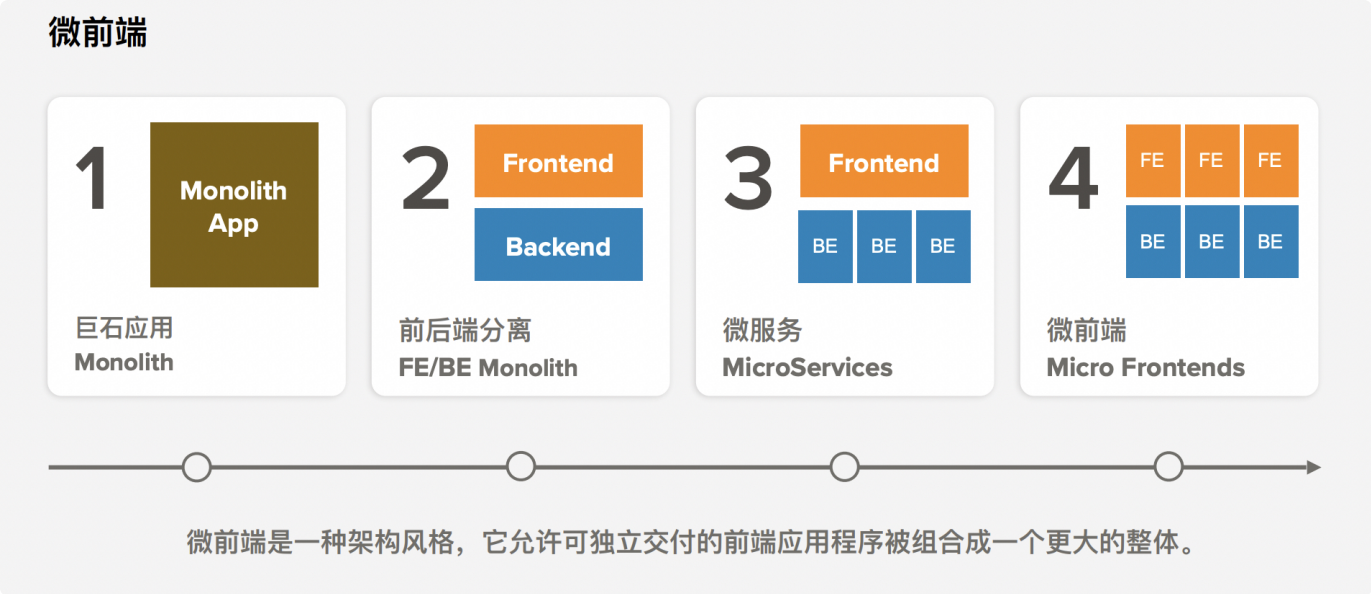
本期课程分两部分，第一部分主要讲述微前端如何解决业务场景的痛点，并以qiankun为例，一步步来给我们的应用接入微前端。

第二部分主要讲解微前端的核心实现原理，并手把手从0到1实现简单的微前端框架。

微前端的背景

微前端是什么

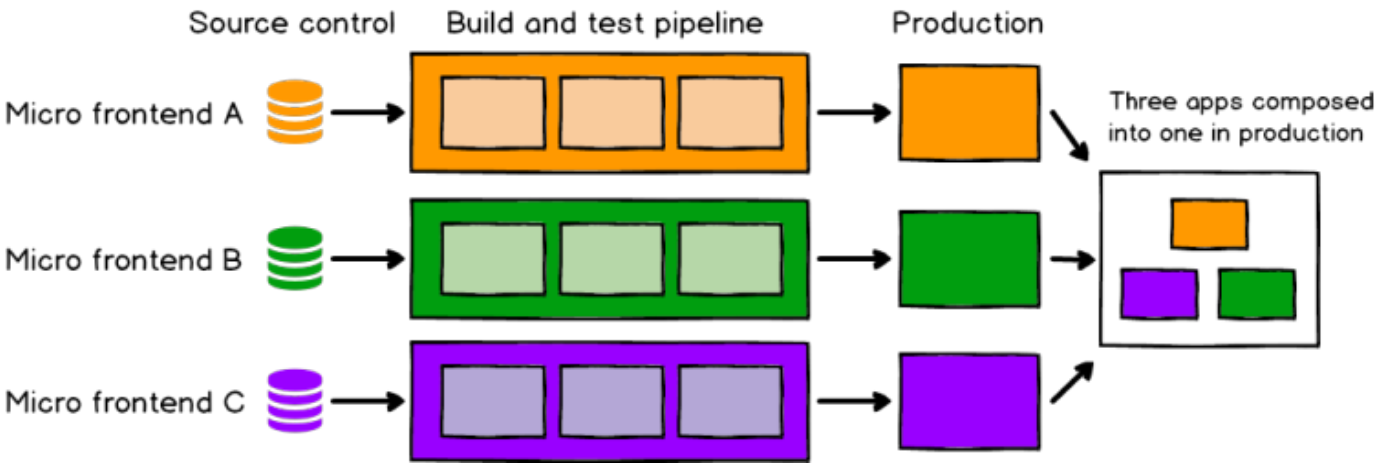
> 一种类似于微服务的架构，是一种由独立交付的多个前端应用组成整体的架构风格，将前端应用分解成一些更小、更简单的能够独立开发、测试、部署的应用，而在用户看来仍然是内聚的单个产品。



现代web应用面临的问题

- DX(developer experience)
  - 多个系统在一个仓库应用中，不同子应用独立SPA模式
  - 系统分为多个仓库，独立上线部署，采用MPA模式
- UX(user experience)
  - 性能体验
  - 页面跳转和用户体验问题

解决方案–微前端



微前端的意义

微前端的主要特点

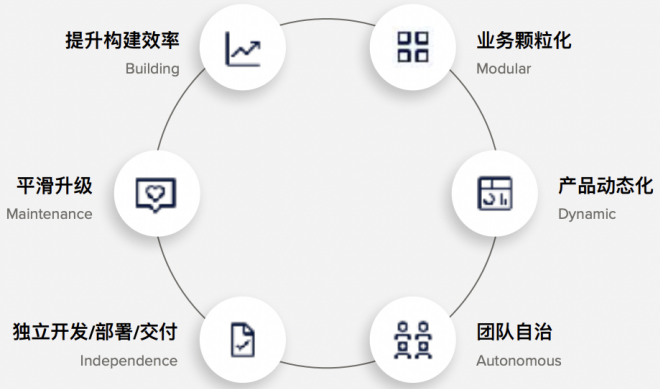
- **低耦合**：当下前端领域，单页面应用（SPA）是非常流行的项目形态之一，而随着时间的推移以及应用功能的丰富，单页应用变得不再单一而是越来越庞大也越来越难以维护，往往是改一处而动全身，由此带来的发版成本也越来越高。微前端的意义就是将这些庞大应用进行拆分，并随之解耦，每个部分可以单独进行维护和部署，提升效率。
- **不限技术栈**：在不少的业务中，或多或少会存在一些历史项目，这些项目大多以采用老框架类似（Backbone.js，Angular.js 1）的B端管理系统为主，介于日常运营，这些系统需要结合到新框架中来使用还不能抛弃，对此我们也没有理由浪费时间和精力重写旧的逻辑。而微前端可以将这些系统进行整合，在基本不修改来逻辑的同时来同时兼容新老两套系统并行运行。

微前端解决的问题

微前端真正解决了什么

- 业务领域的代码库不够独立和高度可重用
- 相同的产品功能由多个团队开发 / 产品功能难以保持统一
- 新的产品理念无法在不同的应用中快速复用 / 实现
- 快速迭代新子业务 / 干净移除将被淘汰的子业务
- 提升构建效率
- 改善交付效率
- 架构渐进升级
- 子团队的独立性
- ……

微前端的价值



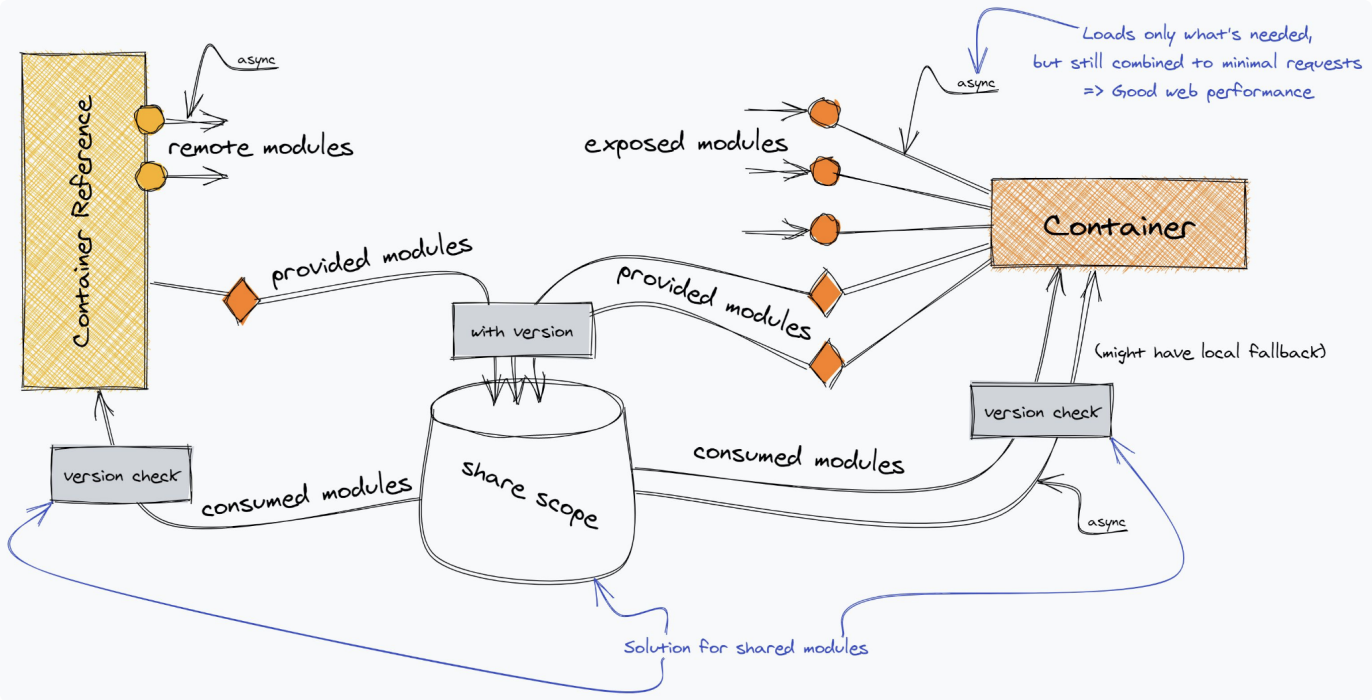
工程 | 商业

微前端的方案

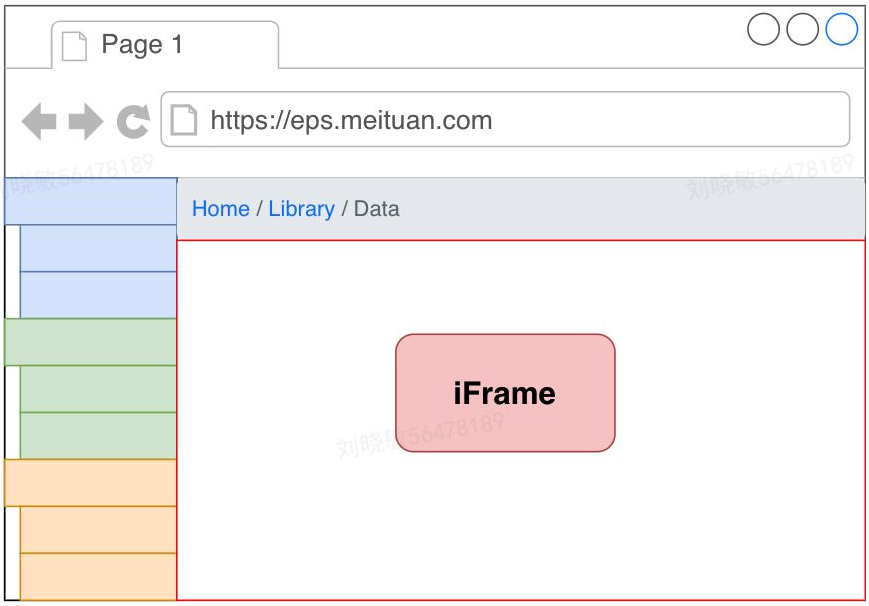
微前端应该具备哪些能力

一些可以实现微前端的方案

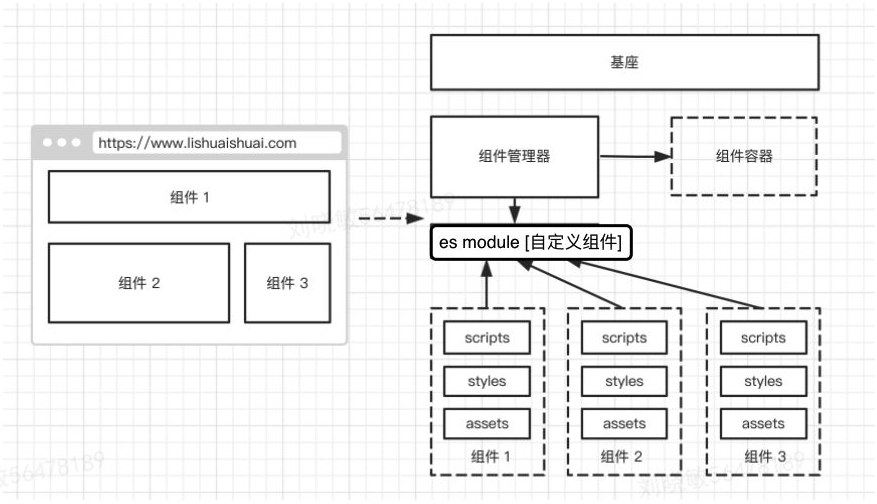
- 使用 HTTP 服务器的路由来重定向多个应用
- **在不同的框架之上设计通讯、加载机制，诸如** Single-SPA qiankun icestark
- **通过组合多个独立应用、组件来构建一个单体应用**
  - 微前端之构建时方案(Module Federation EMP)



- iFrame。使用 iFrame 及自定义消息传递机制



- 使用纯 Web Components 构建应用



- 结合 Web Components 构建
  - stencli

业界主流的微前端框架

- single-spa：社区公认的主流方案，可以基于它做二次开发
- qiankun：基于 single-spa 封装，增加 umi 特色，增加沙箱机制（JS、ShadowDOM 等）
- icestark：类似于 single-spa 实现，React 技术栈友好，阿里的另一个轮子

基于qiankun的微前端实战

创建主应用基坐

基坐主要实现微应用框架的初始化和注册等，通常没有具体的应用业务逻辑在里边

创建主应用

使用 vue-cli 生成一个Vue 的项目，初始化主应用。

创建微应用容器

在主应用中创建微应用的承载容器，这个容器规定了微应用的显示区域，微应用将在该容器内渲染并显示。

注册微应用

构建好了主框架后，需要使用 的 registerMicroApps 方法注册微应用

启动主应用

接入微应用

qiankun 内部通过 import-entry-html 加载微应用，要求微应用需要导出生命周期钩子函数

接入vue微应用

接入react微应用