Micro-Frontend

精致化的微前端开发之旅

01 微前端概念解读
02 微前端快速入门
03 Web Components
04 三大框架微前端
05 关于未来的思考

01

微前端概念解读

微前端（Micro-Frontend），是将微服务（Micro-Services）理念应用于前端技术后的相关实践，使得一个前端项目能够经由多个团队独立开发以及独立部署。

01 微前端开发的特性

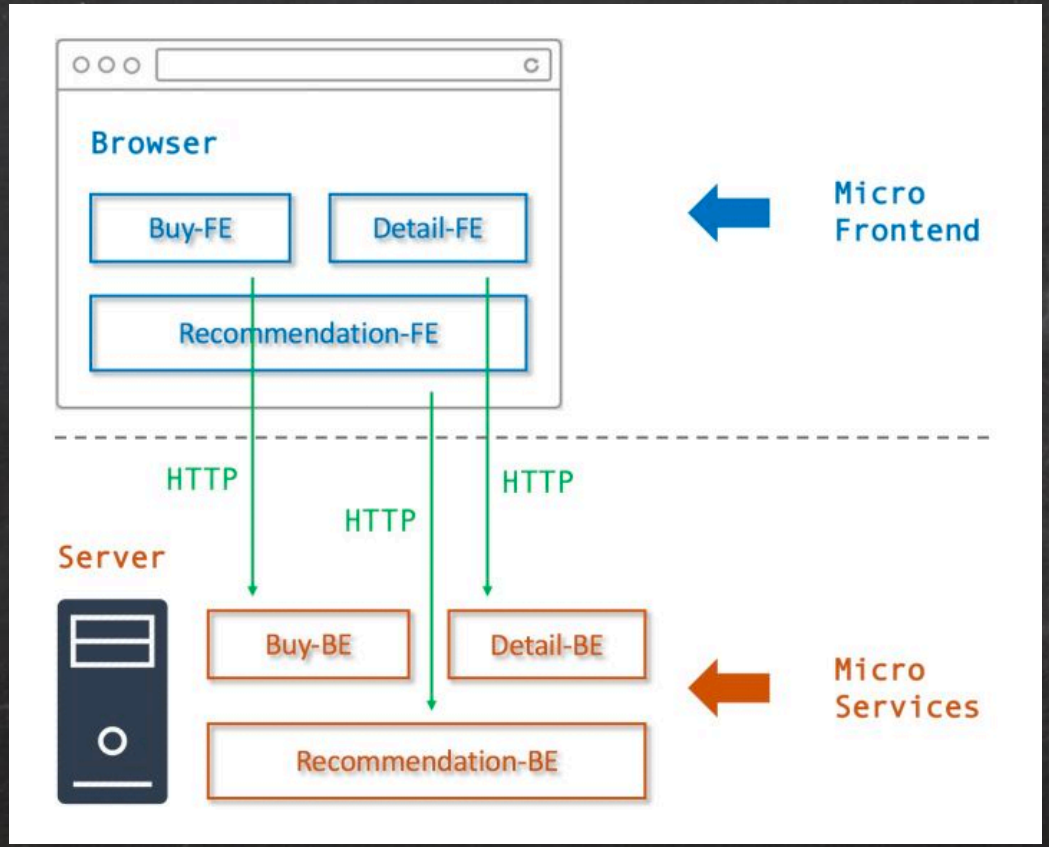
各个开发团队都可以自行选择技术栈，不受同一项目中其它团队影响；

各个交付产物都可以被独立使用，避免和其它交付产物耦合；

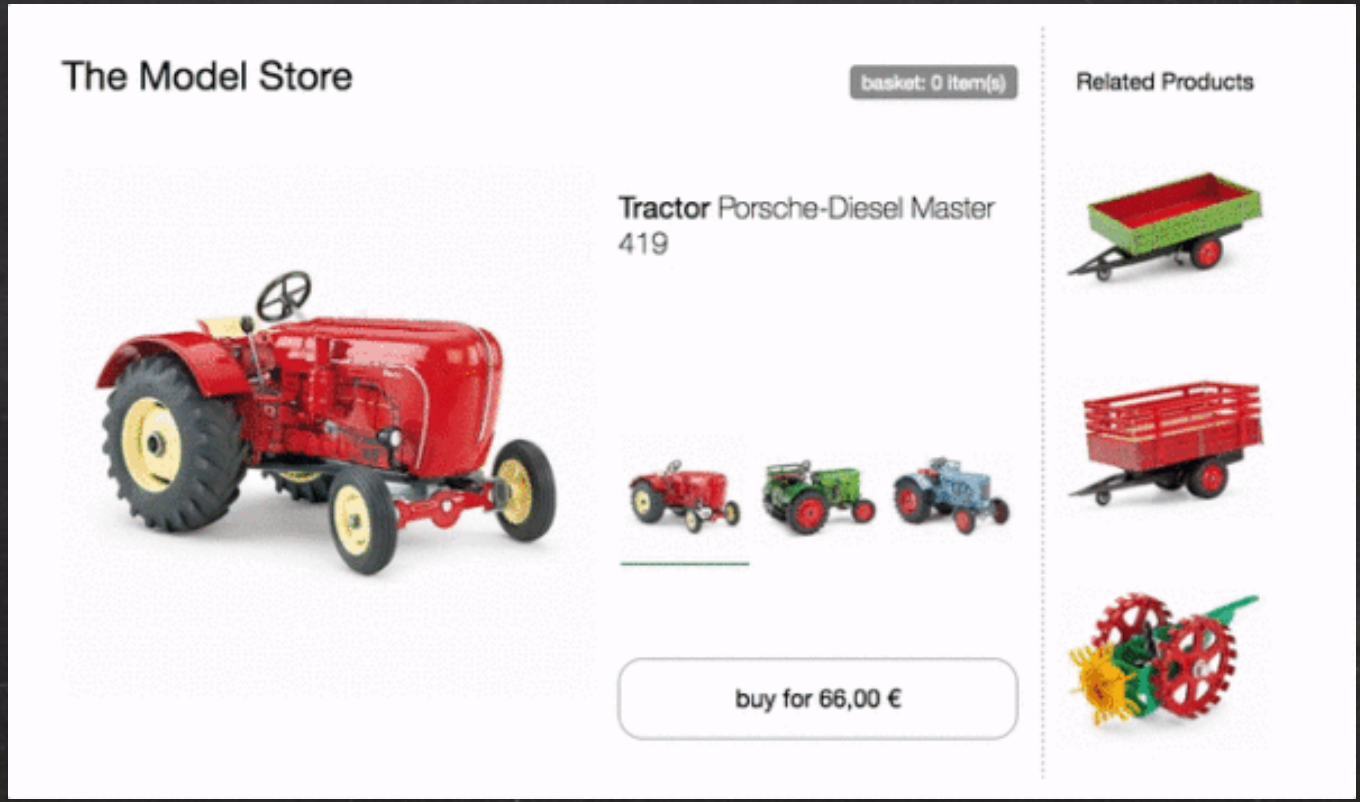
各个交付产物中的样式不会污染到其它组件；

各个交付产物都可以自由使用浏览器原生 API，而非要求使用封装后的 API；

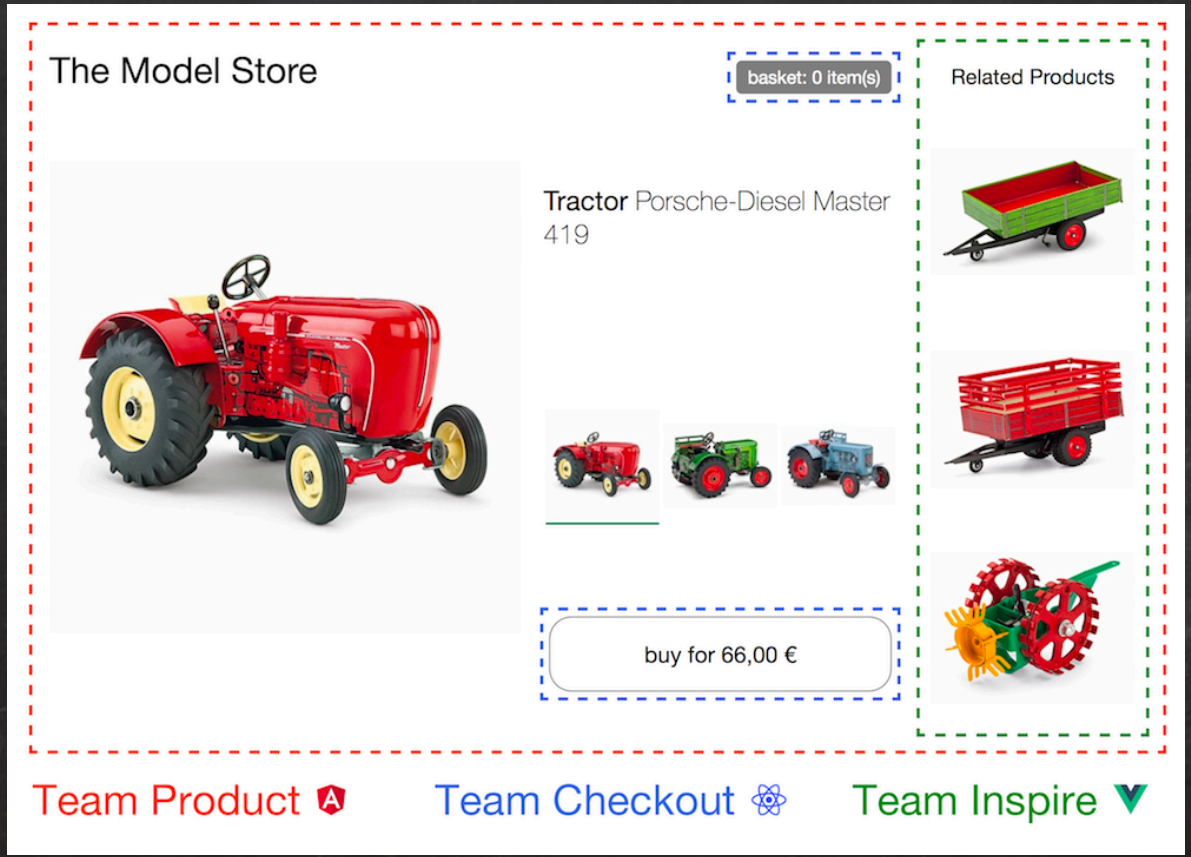
02 开发示意图


Browser
Buy-FE
Detail-FE
Recommendation-FE
Micro Frontend
HTTP
HTTP
HTTP
Server
Buy-BE
Detail-BE
Recommendation-BE
Micro Services

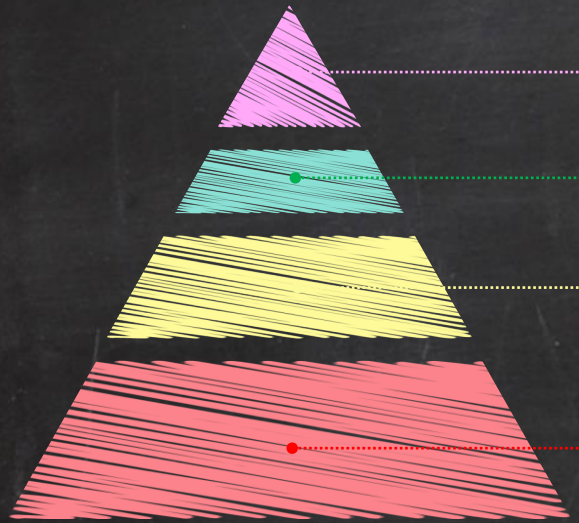
02-1 开发效果图


The Model Store
basket: 0 item(s)
Related Products
Tractor Porsche-Diesel Master 419
buy for 66,00 €

02-1 开发效果图


The Model Store
basket: 0 item(s)
Related Products
Tractor Porsche-Diesel Master 419
buy for 66,00 €
Team Product
Team Checkout
Team Inspire

03 iframe现存问题



不可控制

iframe嵌入的显示区大小不容易控制，存在一定局限性。

bfcache！

URL的记录完全无效，页面刷新不能够被记忆，刷新会返回首页，iframe功能之间跳转也无效

兼容性坑

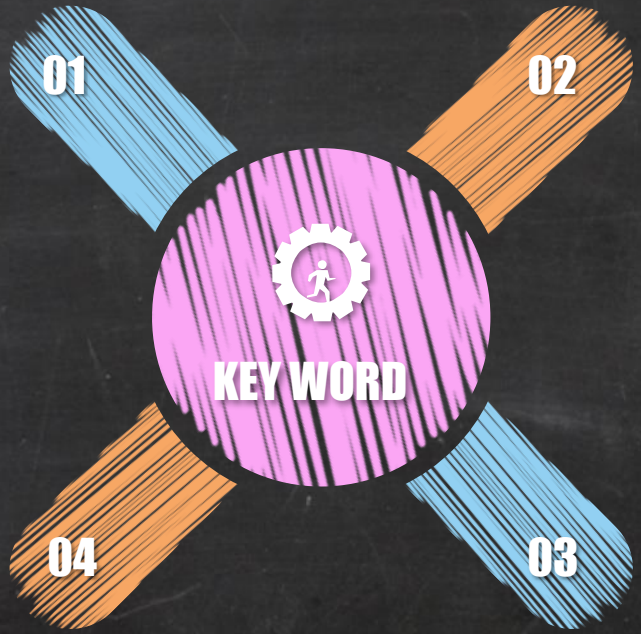
iframe的样式显示、兼容性等都具有局限性

性能开销

iframe 阻塞 onload、占用连接池、多层嵌套页面崩溃。。

04 必须要解决的问题

一个前端需要对应多个后端



提供一套应用注册机制，完成应用的无缝整合

在应用之前团队开发者要制定好使用CSR或SSR的技术方案

构建时集成应用和应用独立发布部署

05 微前端交付产物

发布静态资源+后台路由和服务

发布组件启动时机全由父级决定

发布局部应用配置过程由自身决定

02

前端静态+后端

每个项目独立通过代码版本管理库独立分组、统一技术方案，合成整体技术架构。

https://micro-frontends.org/

01 FIS从入门到放弃

FIS

FIS3 , 为你定制的前端工程构建工具

解决前端开发中自动化工具、性能优化、模块化框架、开发规范、代码部署、开发流程等问题.我们已FIS为栗子🌰切入进微前端。

03

前端独立发包

通过A站点请求B站点独立的组件，全部控制权在A站点内。

Vue

一套用于构建用户界面的渐进式框架。Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，容易于上手。

React

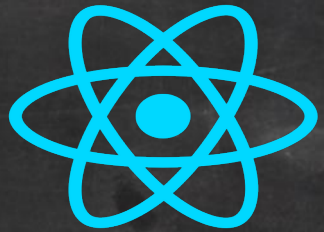
在数据改变时 React 也可以高效地更新渲染界面。以声明式编写 UI，可以让你的代码更加可靠，且方便调试。

Web Components

它由四项主要技术组成，它们可以一起使用来创建封装功能的定制元素，可以在你喜欢的任何地方重用，不必担心代码冲突。

Angular

有着诸多特性，最为核心的是 MVW、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。



Webpack

JS应用程序的静态模块打包器。当处理应用程序时，会递归地构建一个依赖关系图，将所有这些模块打包成一个或多个 bundle。

深入理解Webpack

systemjs 是一个最小系统加载工具，用来创建插件来处理可替代的场景加载过程，包括加载 CSS 场景和图片，主要运行在浏览器和 NodeJS 中。它是 ES6 浏览器加载程序的的扩展，将应用在本地浏览器中。通常创建的插件名称是模块本身，要是没有特意指定用途，则默认插件名是模块的扩展名称

04

总线注册机制

通过开发期、构建期、部署期、运行期完善整体系统

01 微前端构建类单页的业务系统

父项目：映射多个后端服务、统一登录鉴权、获取全局菜单树

子项目：项目菜单、权限、角色隔离

父项目：重新生成引用文件、更新全局入口文件、重启服务

子项目：替换资源文件、调用主项目更新

父项目：生成全局路由、生成全局入口引用

子项目：替换公用库依赖、CSS添加作用域、生成项目静态资源

父项目：路由控制 、模块加载、JS公用库替代、数据流注册、作用域

子项目：注册项目作用域、输出数据流、输出路由、输出功能

05

关于未来的思考

一切不已浏览器为目标的框架都是耍流氓！

01 新的前端框架一览

去万物糟粕

基于 Web Components 并使用 omio 兼容老浏览器(IE8+)，很容易实现 MVVM 架构

JSX使用web-component

详情见官网

详情见官网

Vue使用web-component

详情见官网

Angular使用web-component

3ks!

精致化的微前端开发之旅