



WAKEDATA

惟客数据

微前端：子应用开发

入门+实战

Content

/00

基本概念

/01

子应用创建、开发、调试

/03

部署

/04

已有项目、外部应用集成

/01

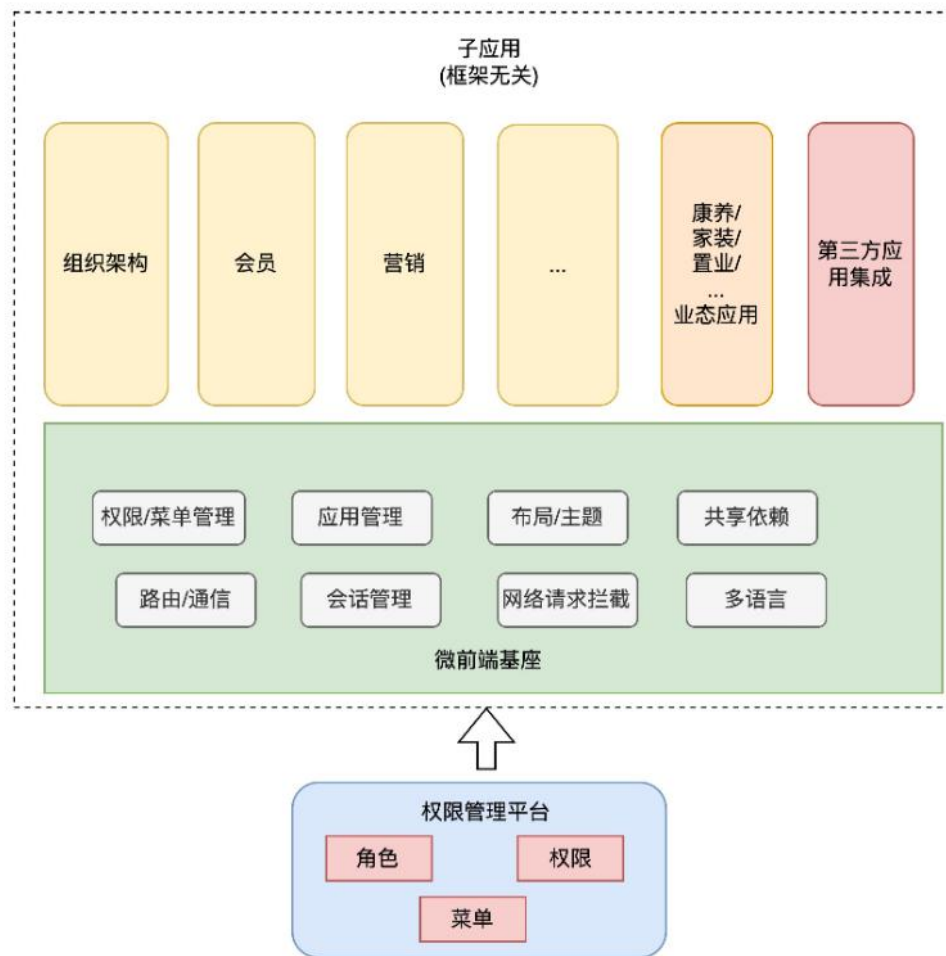
基本概念

客户经营，找惟客



基座

我们采用了基于路由分发 + 基座形式的微前端方案



顶部扩展区域

二级菜单/三级菜单

底部扩展插槽

微应用入口

运营中心 会员中心 客户中心 营销中心 广告中心 企微助手

头部扩展区域

业务中台-测试

四级菜单

消耗动作规则 发放动作规则

规则名称: 请输入规则名称 积分动作: 请选择积分动作 状态: 请选择状态 重置

规则名称	积分动作	适用范围	有效时间	状态	创建人	操作
暂无数据						

< 1 > 前往 1 页 10条/页 共 0 条

微应用渲染区域

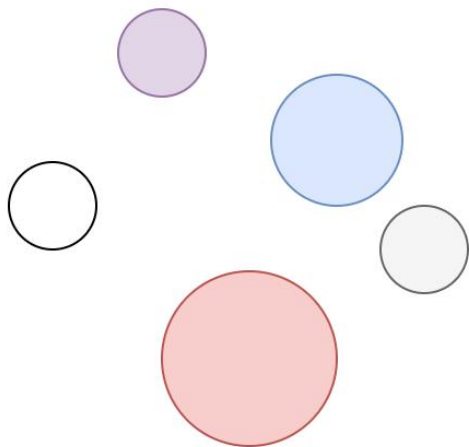
菜单扩展区域

多语言 切换应用 设置中心 退出登录

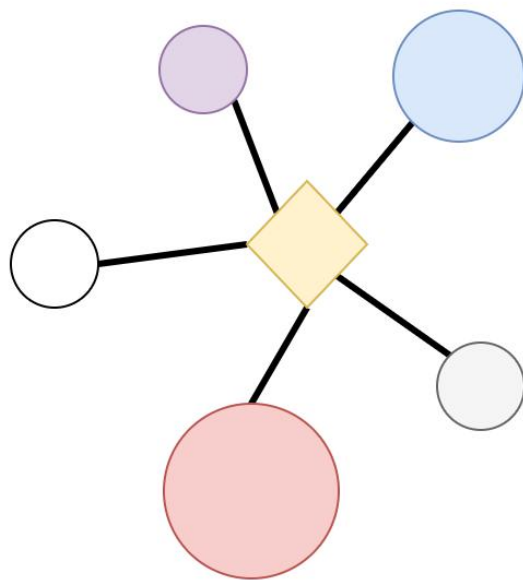


微前端？基座中心化？

为了解决什么问题？



离散的应用



基座聚合的应用

1. ‘中央集权’

我们的多页应用本身就支持独立开发、独立部署、而且是天然的隔离。那微前端还有什么用？

基座是‘子应用’的‘管理者’：

- 统一管理子应用
- 统一规范、标准、视觉
- 资源共享



微前端？基座中心化？

2. 提供了一种更友好的应用集成方式

[为什么不使用 iframe ？](#)

微前端相比多页应用、iframe 有更好的性能优势，也能保证用户体验的连贯性

同时，集成过程也非常简单



微前端？基座中心化？

3. 提供更灵活的服务聚合方式

很多典型的应用都是按菜单聚合的。但菜单会跟随业务的需求变化

场景1：‘运营中心’的营销工具需要放到‘营销中心’

场景2：惟数平台 DataOps 菜单重组

与菜单解耦后，子应用可以根据业务、职能等合适的方式进行聚合

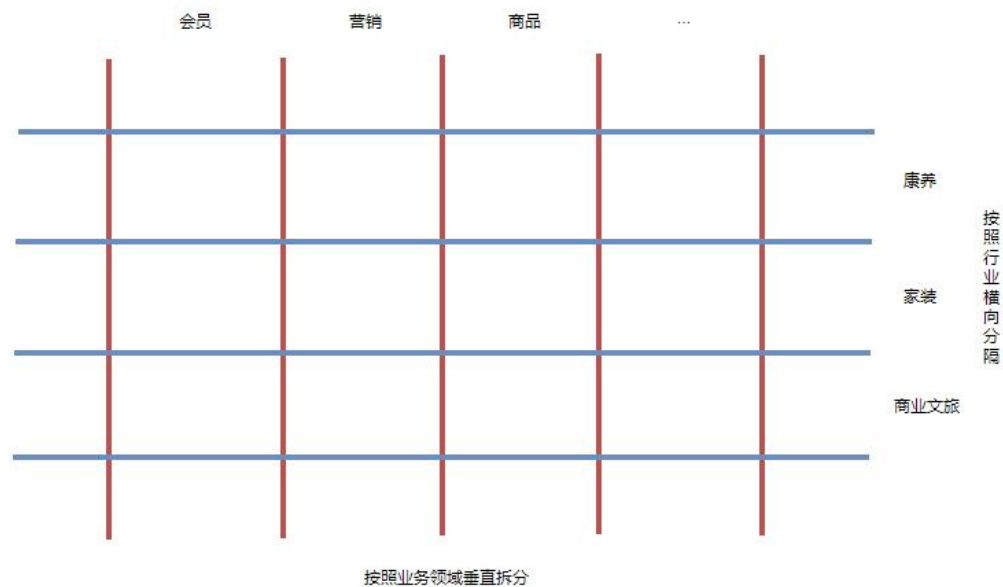
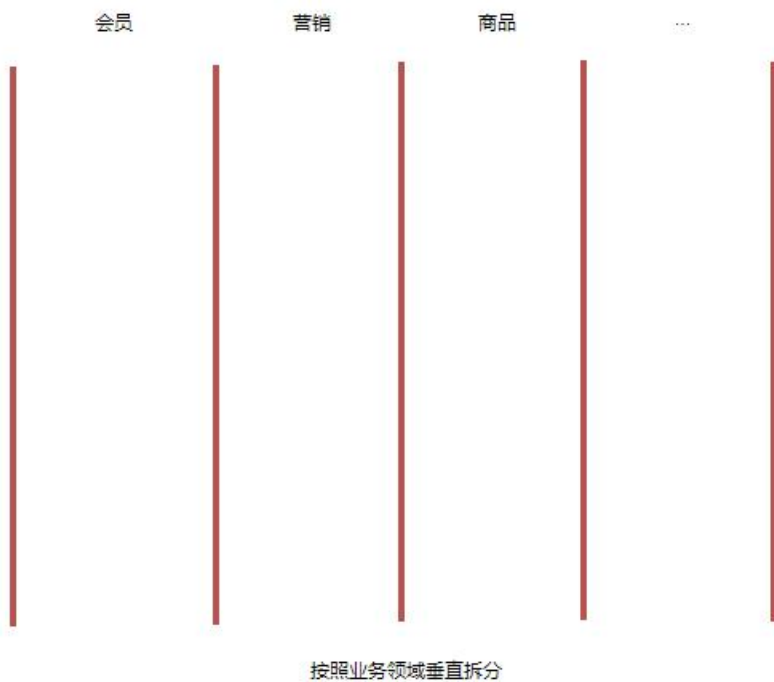
3. 还有很多

- 技术栈无关
- 简化新项目的开发和集成
- 简化了第三方应用的集成
-



微前端？基座中心化？

不能解决什么问题？



微前端可以很好地解决垂直模块拆分问题，但是无法解决行业多态问题

一些基本概念

- `name: string` 子应用的名称, 全局唯一
- `activeRule: string` 子应用的路径, 当路由匹配到该路径时, 加载子应用并渲染
- `entry: string` 子应用 html 入口

/01

子应用创建

客户经营，找惟客



微前端环境准备

1. [正式部署微前端环境](#)
2. 使用[游乐场](#)

```
docker run -it --rm -p 7112:80 wkfe/playground
```



基于游乐场写个 hello world 看看

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <script>
      window.hello = {
        bootstrap: async () => {
          console.log('hello bootstrap');
        },
        mount: async props => {
          console.log('hello mount');
          props.container.innerHTML = '<div>hello world</div>';
        },
        unmount: async () => {
          console.log('hello unmount');
        },
      };
    </script>
  </body>
</html>
```



```
$ npx http-server --cors .
```



hello world

open localhost:7112/__debug__

已注册微应用

基本信息

- baseUrl: /

微应用

本地	name	activeRule	independent	routeMode	状态	操作
No Data						

* 名称(name)

* 入口(entry)

* 路由(activeRule)

独立模式(independent) ☐

路由模式(routeMode)

添加

调试脚本注入

调试脚本链接, 多个脚本以 ';' 分隔, 重启后生效



hello world



鹅城

你好世界

设置中心

营销中心

广告中心

运营中心

会员中心

DEMO

分组



张麻子



查看



编辑

hello world



hello world

部署?

1. 进入容器内部
2. 将 `index.html` 放到 `/data/source/__apps__/hello` 下



权限管理菜单配置

游乐场模式跳过(硬编码)

终端管理

资源管理

接口管理

菜单管理

角色管理

角色类型管理

模块管理

数据字典

租户管理

企业管理

租户管理

消息通知

系统设置

账号管理

IPAAS账号管理

日志管理

事件中心

平台: wkb 菜单名称: 输入菜单名称查询

所属终端: 员工B端 查询

+ 新增菜单 批量导入 批量导出

菜单测试 地产康养 地产333 写字楼 家居建材 企业服务 商超零售 酒店文旅 购物中心 商业中心 地产置业 大会员集团

名称	菜单类型	所属终端	权限标识	路由地址	icon	序号	状态	创建时间	操作
你好世界	菜单	员工B端	hello	/hello	Add	1000	<div></div>	2022-08-15 11:24:59	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>
> 运营中心	菜单	员工B端	wkb	/wkb.html		1001	<div></div>	2022-02-22 18:23:42	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>
> 销售中心	菜单	员工B端	scrm	/scrm.html		1002	<div></div>	2022-04-06 17:01:27	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>
> 营销中心	菜单	员工B端	marketing	/dsp.html		1003	<div></div>	2022-02-22 18:23:42	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>
> 会员中心	菜单	员工B端	customer	/dmp.html		1004	<div></div>	2022-02-22 18:23:42	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>
> 客户中心	菜单	员工B端	cdp	/cdp.html		1005	<div></div>	2022-02-28 15:36:42	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>
> 企微助手	菜单	员工B端	wecom	/wecom.html		1006	<div></div>	2022-04-14 14:15:37	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>
广告中心	菜单	员工B端	ad	/ad.html		1007	<div></div>	2022-03-01 19:42:26	<div>修改 复制 删除</div> <div>加入黑名单 取消黑名单</div>

菜单管理 -> 模块管理 -> 账号管理



hello work

我们也提供了脚手架，用来快速创建一个子应用

```
npm init wakeadmin my-project

Last login: Sun Jul 31 20:58:42 on ttys003
→ ~ npm init wakeadmin my-project
? 选择项目类型
> vue 微前端子应用
  主题包
  Vue 应用
```



hello work

目录结构以及一些简单的配置

```
// vue.config.js
module.exports = defineConfig({
  parallel: false,
  transpileDependencies: [/(wakeapp|wakeadmin)/],
  pluginOptions: {
    // 详见子应用集成: https://wakeadmin.wakedata.com/mapp/integration.html
    ...defineMappChild({
      mapp: {
        - activeRule: '/demo',
        + activeRule: '/wkb.html',
      },
    }),
    // 共享依赖: 详见: https://wakeadmin.wakedata.com/mapp/advanced/vendors.html
    ...defineVendors({
      modules: {},
    }),
    ...defineCE({ customElement: /wkc-/ }),
  },
})
```



hello work

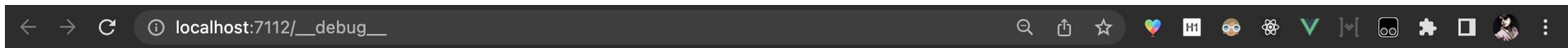
启动

```
$ pnpm serve
```

pnpm serve	
demo@0.1.0 serve /Users/ivan/my-project	
vue-cli-service serve	
mode	子应用
terminalMode	false
name	demo
publicPath	http://localhost:61950/
port	61950
assetsDir	demo
outputDir	dist
shared	<input type="checkbox"/>
mapp	{ "name": "demo", "activeRule": "/wkb.html", "entry": "http://localhost:61950/", "version": "0.1.0", "description": "Wakeadmin 子应用" }



本地开发调试 __debug__



已注册微应用

基本信息

- baseUrl: /

微应用

* 名称(name) demo

* 入口(entry) http://localhost:61950/

* 路由(activeRule) /wkb.html

独立模式(independent) ☐

路由模式(routeMode) hash

添加

调试脚本注入

调试脚本链接，多个脚本以 ';' 分隔，重启后生效



hello work

基座 API 调用

- 1. 安装和使用
- 2. 组件
 - wkc-allows
 - wkc-allows-page
 - wkc-breadcrumb
 - wkc-content
 - wkc-float-footer
 - wkc-fullscreen
 - wkc-menuless
 - wkc-header
 - wkc-header-slot
 - wkc-header-dropdown-menu
 - wkc-sidebar-top-slot / wkc-sidebar-bottom-slot
 - wkc-icon
 - wkc-error-page
 - wkc-error-page-not-found
 - wkc-error-forbidden
- 3. API
 - 3.1 eventBus
 - 3.2 权限检查
 - 3.3 侧边栏、菜单栏操作
 - 3.4 路由
 - 3.5 会话操作
 - 3.6 多语言
 - 3.7 assets
 - 3.8 基座



hello work

配置一个 404 页面试试

```
// src/router/index.ts
import { h } from 'vue';
import { RouteRecordRaw } from 'vue-router';
import HomeView from '../views/HomeView.vue';

export const routes: RouteRecordRaw[] = [
  {
    path: '/',
    name: 'home',
    component: HomeView,
  },
  {
    path: '/about',
    name: 'about',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/AboutView.vue'),
  },
  {
    path: '/*:pathMatch(.*)',
    name: 'notFound',
    component: () => h('wkc-error-page-not-found'),
  },
];
```


/03

部署

客户经营，找惟客



```
# Dockerfile
FROM wkfe/mapp-child

# 拷贝 dist 目录下的文件到 /data
COPY dist /data
```



Jenkins、Rancher...

跳过，我们在上一次分享已经介绍过



挂载到 wakeadmin-apps

🐮

wkb-k8s-cluster1
bizpf-test

资源

应用商店

命名空间

成员

工具

仪表盘

+

升级 wakeadmin-apps 的 Sidecar

名称
app-web

Sidecar 类型
☐ 标准容器
☒ Init 容器

Docker 镜像 *
wkfe/my-app

端口映射
+ 添加规则

全部展开

环境变量
设置容器可见的环境变量，包括从其他资源(如密文等)注入的值。

健康检查
周期性地向容器发出请求，以检测其健康状态。默认设置下，readiness 和 liveness 使用相同的配置参数。对于应用初始化较长的容器，需要增加就绪检测时间。

数据卷
持久化和共享独立于单个容器生命周期的数据

卷名
vol1

卷类型
PVC

删除卷

PVC *
wakeadmin

容器路径 *
/mnt

子路径(仅适用于数据卷挂载，不适用于主机路径映射)

只读
☐

删除

+ 添加映射

添加卷

/04

已有项目、外部系统集成



最小化集成

```
import { createApp } from 'vue';
import App from './App.vue';
import router from './router';

const APP_NAME = 'APP_NAME'; // 必须唯一
let app;

async function bootstrap() {
  console.log('bootstrapped');
}

async function mount(props) {
  app = createApp(App);
  app.use(router);
  app.mount(props.container ? props.container.querySelector('#app') : '#app');
}

async function unmount(props) {
  app.unmount();
}

async function update(props) {
  console.log('update props', props);
}

window[APP_NAME] = {
  bootstrap,
  mount,
  unmount,
  update,
};

// 不在微前端框架下，直接渲染
if (typeof window.__MAP_BAY__ === 'undefined') {
  mount();
}
```

```
// 以 vue-cli 为例子:
const { defineConfig } = require('@vue/cli-service');

// 开发服务器端口号
const port = 9999;

module.exports = defineConfig({
  // 以下配置只用于本地开发
  // 保证基座能够正常加载
  publicPath: process.env.NODE_ENV === 'production' ? '<应用的部署路径、或者 CDN 地址>' : `http://localhost:${port}/`,
  devServer: {
    port,
    https: true,
    // 支持跨域
    headers: { 'Access-Control-Allow-Origin': '*' },
  },
});
```

基座提供的能力对子应用来说是可选的。

1. 添加生命周期钩子
2. 如果要本地开发，需要配置 `publicPath`，并支持跨域
3. 隐藏菜单、调整布局



已有应用改造

复用基座的能力、做减法

- 我们提供了 `vue-cli` 插件
- 减法：
 - 移除侧边栏、顶部菜单栏
 - 移除组件库样式
 - 移除 `polyfill` 代码
- 使用基座 API
 - 组件库
 - 权限管理
 - 会话管理
- 最佳实践
 - 统一使用 `vue-cli`
 - 统一使用主题变量
 - 不要硬编码配置信息
-

详见：[子应用集成指南](#)



WAKEDATA

惟 客 数 据

Thanks



客户经营，找惟客