



2021/6/27 - B 站直播间
前端早早聊大会福利专场

如何设计和架构渐进式微前端

林长青@RingCentral | 14:00

如何在大型应用里应用微前端

黄烈钦@RingCentral | 15:40



长按扫码报名，进群领取录播 / 讲稿 / PPT

如何设计和架构渐进式微前端

林长青

RingCentral Integration 前端技术负责人

林长青 Michael Lin

.....
RingCentral Integration Tech Leader
.....

曾担任若干公司前端负责人
.....

擅长前端架构设计与JavaScript通用化
.....

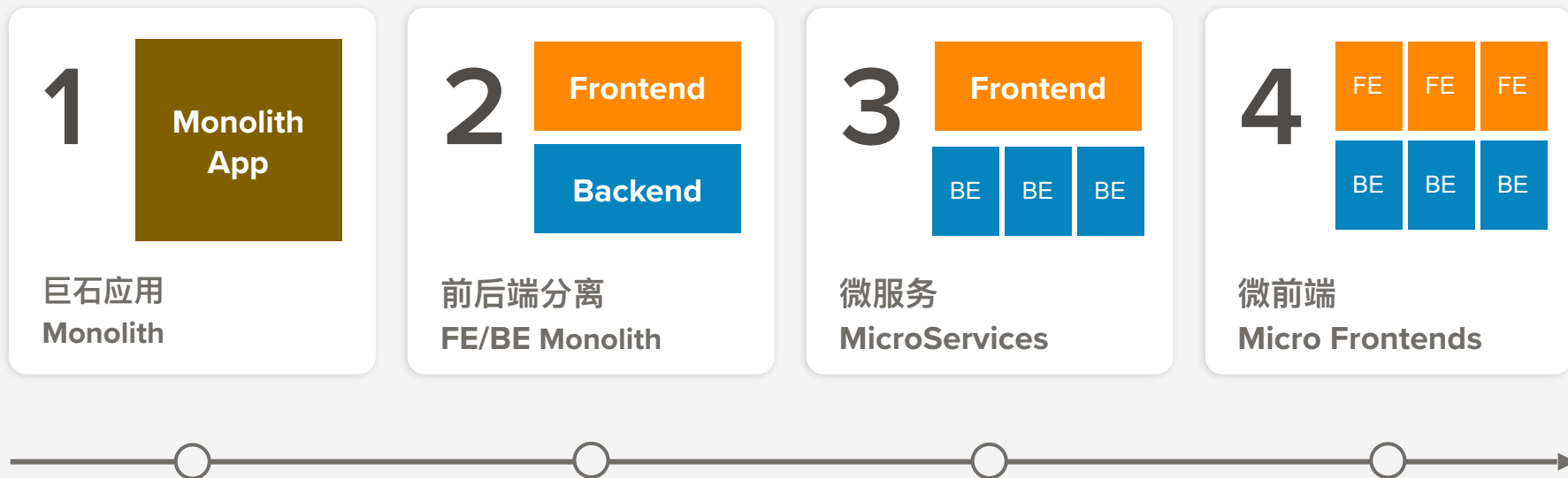
对前端框架设计与自动化测试框架设计有深入研究及实战经验
.....



目录

1. 微前端
2. Module Federation
3. 新的渐进式微前端框架
4. 渐进式微前端架构
5. Demo
6. Q&A

微前端



微前端是一种架构风格，它允许可独立交付的前端应用程序被组合成一个更大的整体。

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
-

微前端的特点

1

独立与自治

- 简化开发
- 新的管理组织形式

2

技术栈无限定

- 技术栈平滑升级
- 架构演化与升级

3

运行时集成

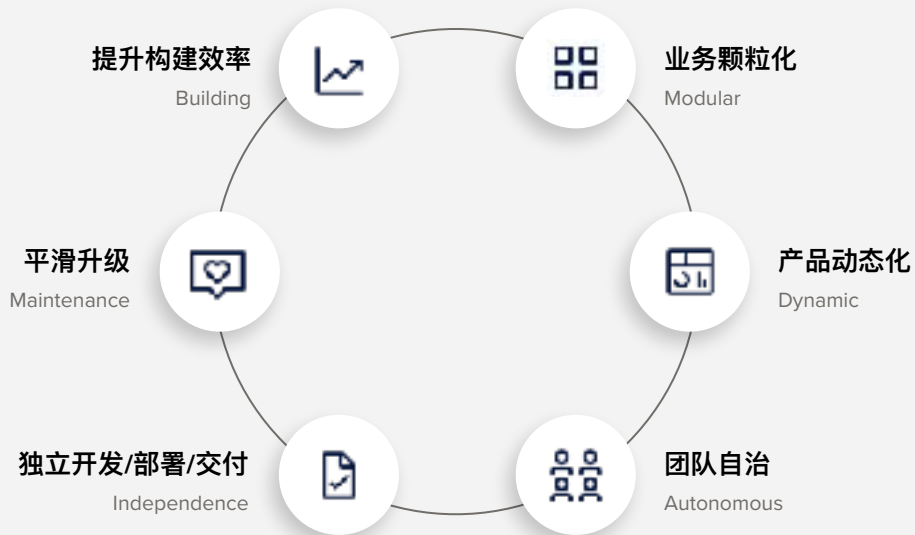
- 依赖控制
- 动态化

4

颗粒化解耦与可组合

- 产品交付的新形式
- 增量更新

微前端的价值



工程 | 商业

微前端的落地需求与目的

新旧子项目混合的大型项目

控制台类型大型项目

平滑升级

无技术栈限制

工程治理

全新大型项目

多种应用类型的大型项目

动态化

独立性

商业/业务价值

微前端的挑战

跨应用通讯
多运行容器间通讯

应用通信

运行环境的差异

开发 / 生产
测试 / 流程

运维 / 治理 - 切分微前端
依赖控制 / 版本控制

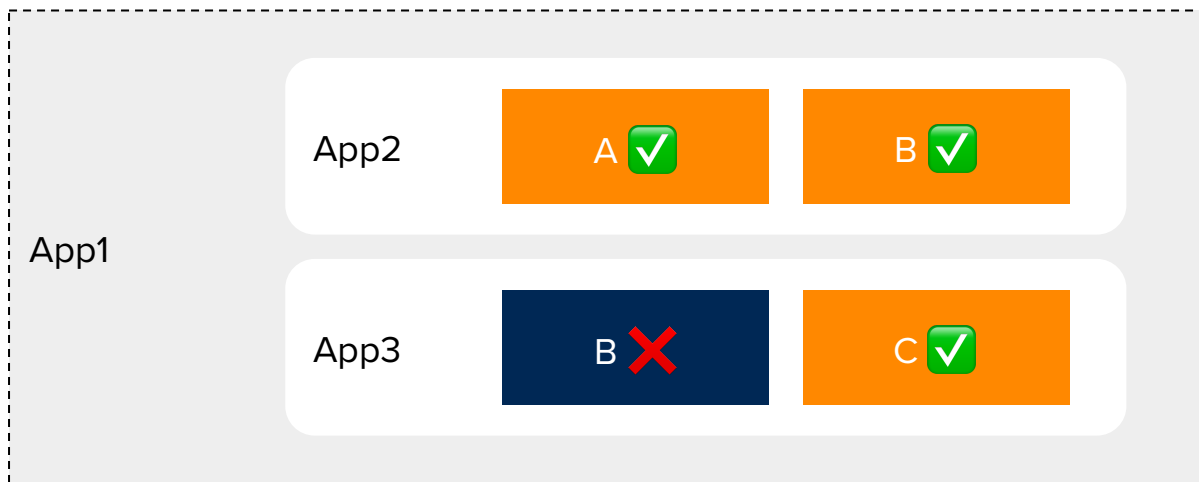
控制架构复杂度

控制微应用包

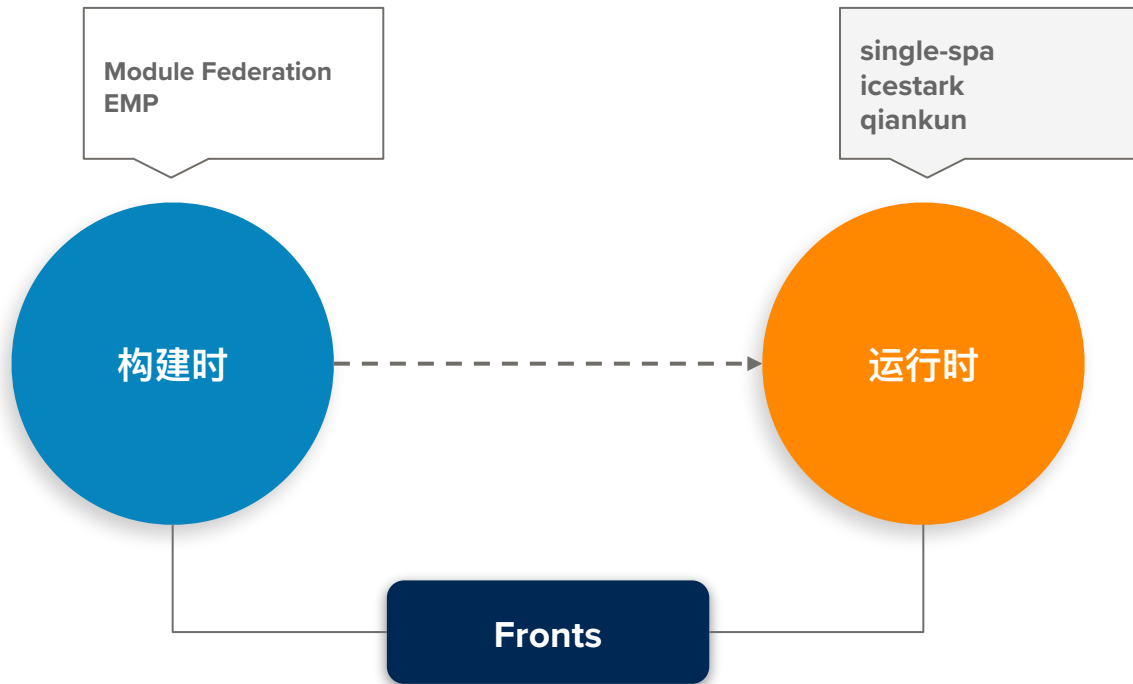
依赖共享 / 依赖管理
共享库的维护

Module Federation

Module Federation允许一个JavaScript应用程序从另一个应用程序动态加载代码，并在这个过程中，共享其依赖关系。



微前端的解决方案

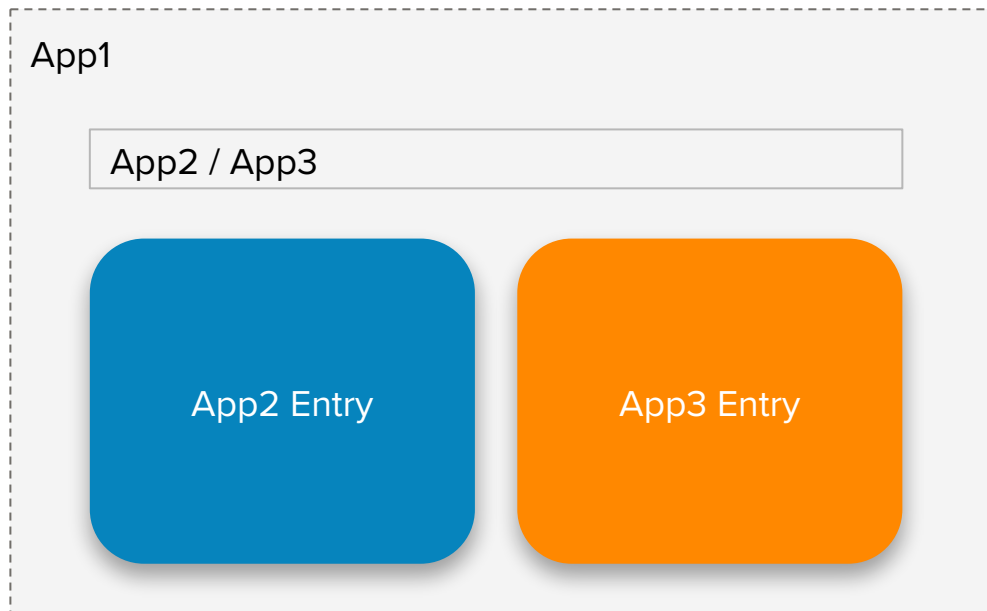


Fronts

Fronts 是一个基于 Webpack 的 Module Federation API 设计的渐进式微前端框架。

它强调微前端颗粒间的去中心化依赖管理，并支持多种运行模式来满足不同的微前端架构需求。

Fronts 范例



Fronts 使用步骤 1

Setup site.json

```
{
  "name": "app1",
  "exports": [],
  "dependencies": {
    "app2": "http://localhost:3002/remoteEntry.js"
  }
}
```

```
{
  "name": "app2",
  "exports": ["/src/bootstrap"],
  "dependencies": {}
}
```

Fronts 使用步骤 2

Setup Webpack
config

```
const { createWebpackConfig } = require('fronts-bundler');  
module.exports = createWebpackConfig(originalWebpackConfig);
```


Fronts 使用步骤 3

Create an entry
in App2

```
import React from 'react';
import ReactDOM from 'react-dom';
import { boot } from 'fronts-react';
import App from './App';

export default function render(element) {
  ReactDOM.render(<App />, element);
  return () => {
    ReactDOM.unmountComponentAtNode(element);
  };
}

boot(render, document.getElementById('root'));
```

Fronts 使用步骤 4

Use the App2 entry
in App1

```
import React from 'react';
import { useApp } from 'fronts-react';

export const App = () => {
  const App2 = useApp([
    name: 'app2',
    loader: () => import('app2/src/bootstrap'),
  ]);
  return <App2 />;
};
```

Fronts 特性

- 支持非 Module Federation
- 去中心化配置
- 跨框架
- 代码分割/懒加载
- CSS 隔离
- 支持快速切换非微前端构建
- 生命周期
- 支持Web Components 和 iFrame
- 支持Monorepo 和 TypeScript
- 版本控制
- 零劫持
- 通用通讯API

微前端框架对比

	single-spa	qiankun2.0	icestark	emp	fronts
1. 无技术栈限制	✓	✓	✓		✓
2. 微应用	✓	✓	✓		✓
3. 微模块	✓		✓	✓	✓
4. CSS隔离		✓	✓		
5. JS隔离		✓	✓		✓
6. 通讯/全局状态	✓	✓	✓		✓
7. 生命周期		✓	✓		
8. HTML入口				✓	✓
9. 去中心化					✓
10. 多种构建类型	✓	✓	✓	✓	✓
11. 版本控制	✓	✓	✓	✓	
12. 依赖共享					
13. CLI					

Fronts API设计与使用

boot() / createWebpackConfig()

useApp()

CSS 隔离*

useWebComponents()

CSS 隔离

useIframe()

JS/CSS 隔离

Fronts 如何解决那些争议问题

颗粒级别划分

入口点JS为主还是HTML为主

模块共享

CSS/JS 隔离利弊

依赖管理和版本控制

颗粒通用性

微应用和微模块

暂不支持HTML Entry

利用Webpack的module federation

CSS隔离在构建时处理 / JS暂不劫持与隔离

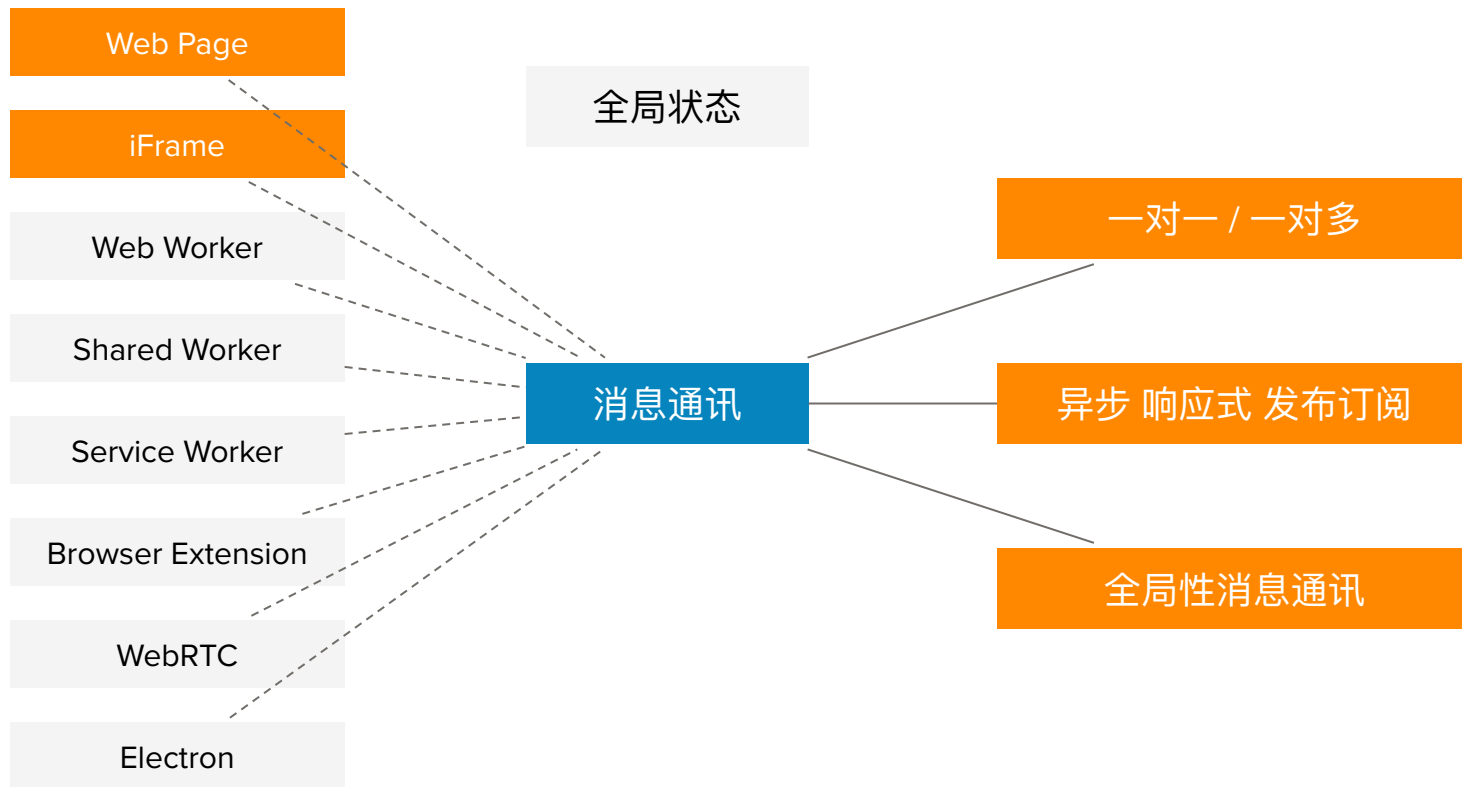
实现构建时依赖管理和运行时版本控制

微应用与微模块运行在多种环境的通用设计

架构 Fronts



Fronts 通讯



Fronts 通讯范例

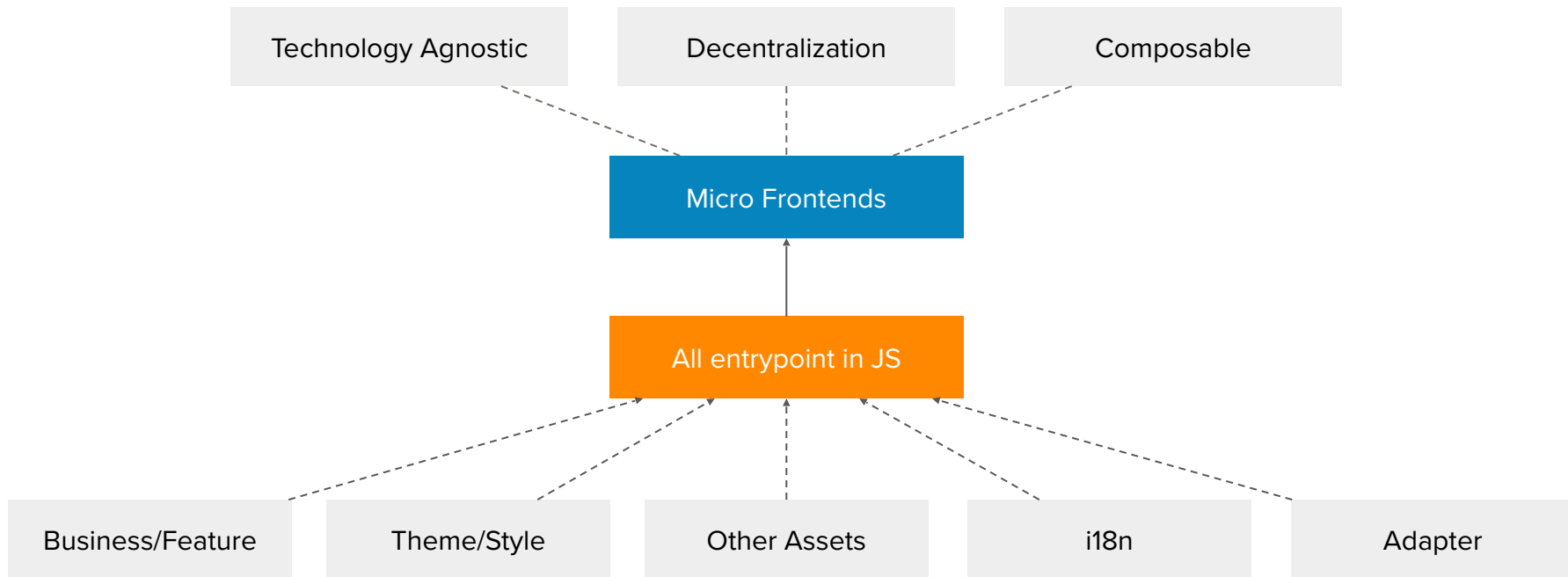


```
import { globalTransport } from "fronts-transport";  
  
globalTransport.listen("increase", (count) => {  
  return count + 1;  
});
```



```
import { globalTransport } from "fronts-transport";  
  
const result = await globalTransport.emit("increase", count);
```

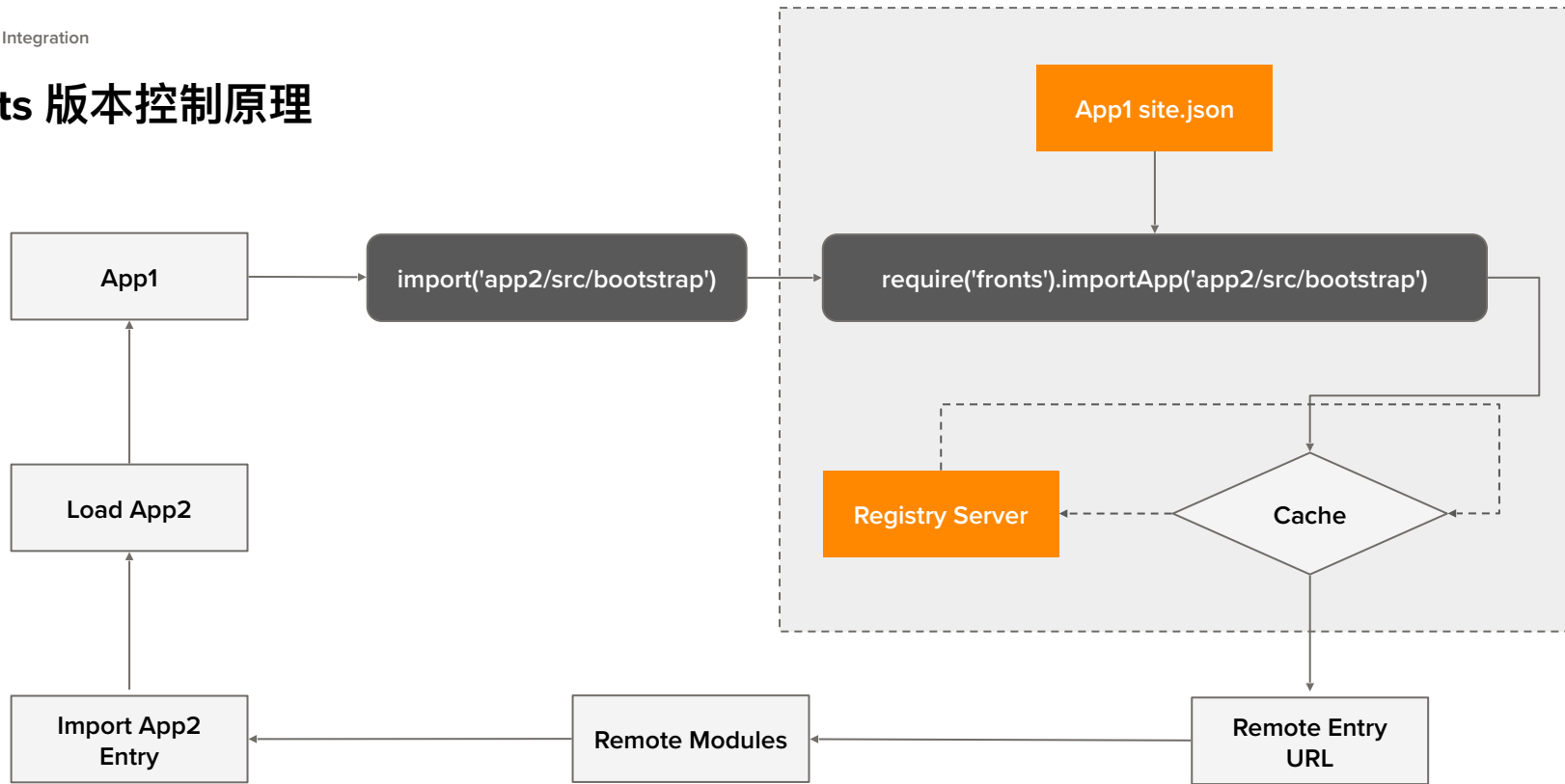
Fronts 核心概念



Fronts 渐进式



Fronts 版本控制原理



Fronts 依赖管理与版本管理

Node

开发与构建

package.json

NPM / Yarn

Fronts

运行时

site.json

Custom Registry Server

Fronts 测试

```
import { run, useContext } from 'fronts-test';

export const addTodo = (todoText) => {
  const { page } = useContext();
  await page.type('.text', todoText);
  await page.click('.add');
};
```

```
import { addTodo } from 'App/test/steps/todo';

const entry = () => {
  await run(addTodo, 'Use Fronts');
};
```

Fronts 调试



```
import { getMeta } from 'fronts';  
console.log(getMeta());
```

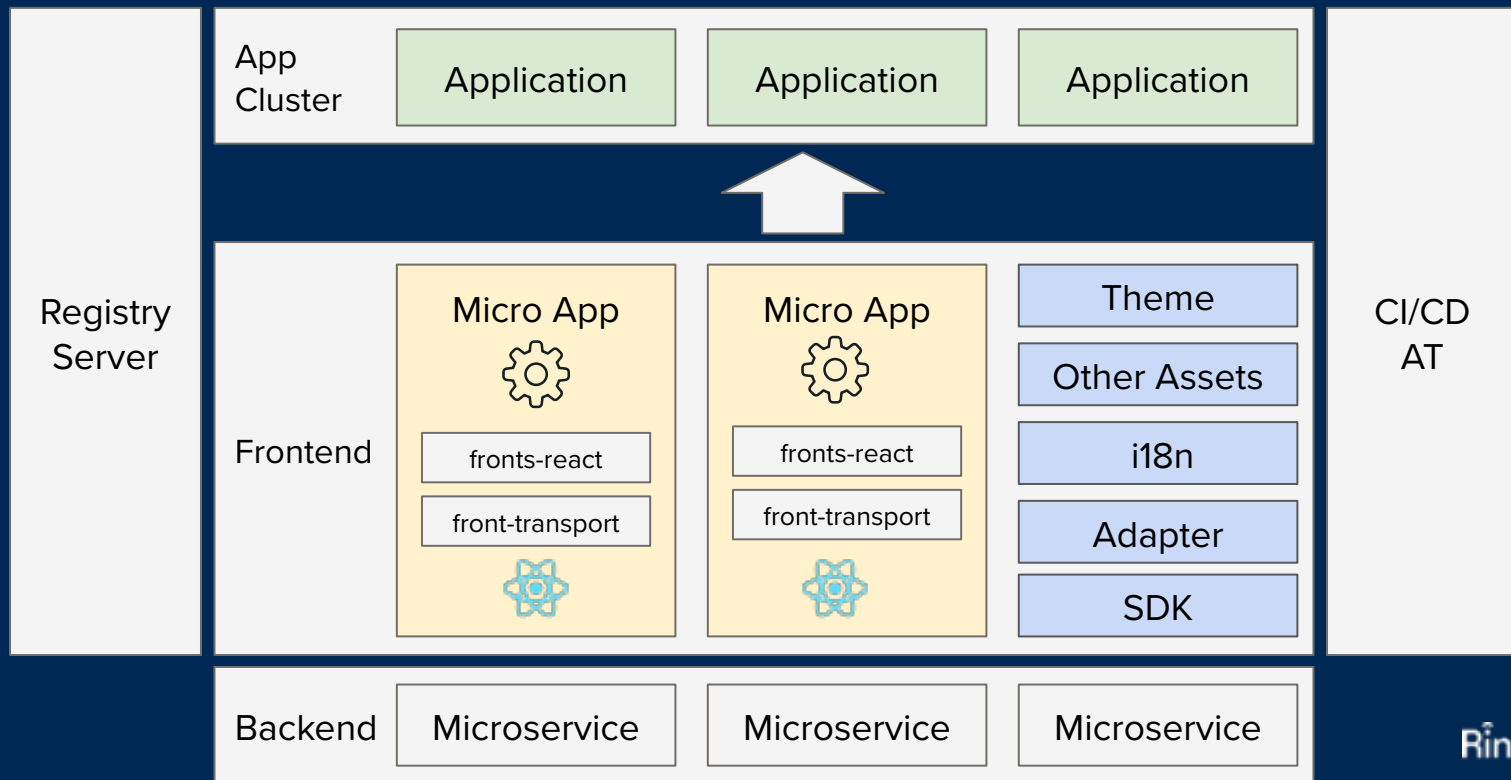


```
{  
  "name": "app3",  
  "meta": {  
    "__main__": "app1",  
    "__entry__": "http://localhost:3001/#/app2",  
    "app2": {  
      "dependencies": {  
        "app3": "http://localhost:3003"  
      }  
    },  
    "app3": {  
      "dependencies": {}  
    },  
    "app1": {  
      "dependencies": {  
        "app2": "http://localhost:3002"  
      }  
    }  
  }  
}
```

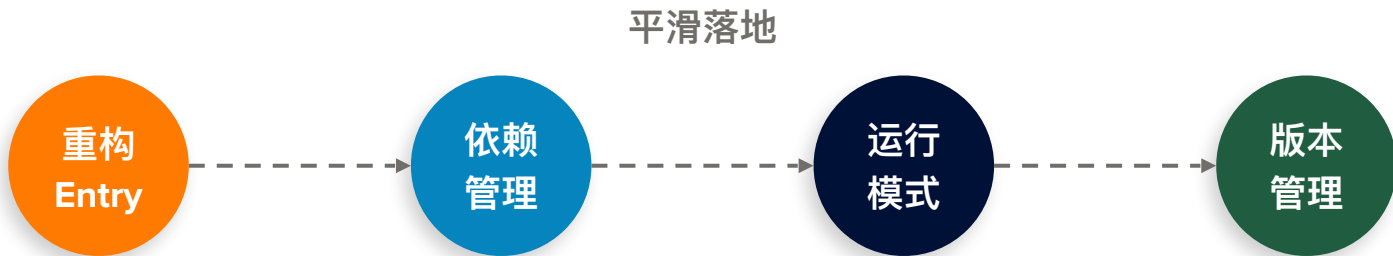
Fronts 分包结构

- ❏ fronts-bundler
- ❏ fronts-react
- ❏ fronts-vue
- ❏ fronts-ng
- ❏ fronts
- ❏ fronts-test
- ❏ fronts-transport
- ❏ fronts-html
- ❏ fronts-sandbox
- ❏ fronts-cli
- ❏ fronts-registry

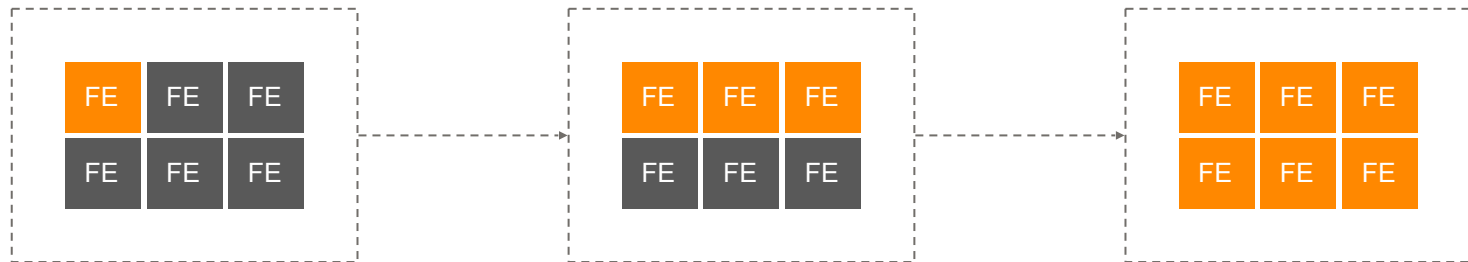
基于Fronts的微前端架构



渐进式微前端

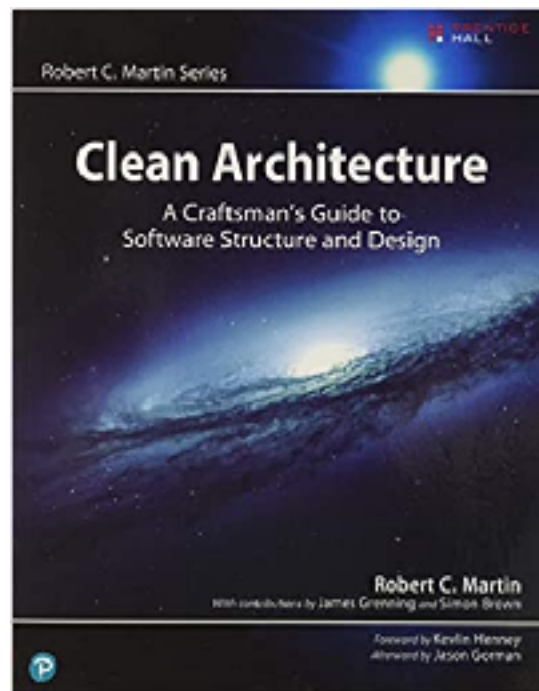


渐进式演进架构



如何回到典型的SPA构建





推荐书籍

Fronts, 不只针对“控制台”等类型工程，更面向多种构建产及产品动态化的大型现代前端工程架构。



Demo



提问互动有礼

答疑环节被选中问题作答的小伙伴，将得到以下礼物（二选一）

RingCentral定制版茶具套装



RingCentral定制版便携充气颈枕



关于RingCentral铃盛

全球UCaaS(统一通信)行业NO.1。总部位于硅谷，分公司遍布全球。目前国内研发中心位于：厦门、杭州、香港。

支持云面试、云入职，部分职位可申请长期在家办公。



加入我们的N个理由

 在家办公	 硅谷文化	 风口行业
 国际平台	 前沿技术	 敏捷开发
 员工持股	 六险一金	 免费外教
 带薪学习	 超长假期	 最佳雇主

如何加入？



微信搜索公众号“RingCentral铃盛”，获取更多前端招聘信息

也可直接发简历到：*Tech.china@ringcentral.com*

Q&A

<https://github.com/unadlib/fronts>



早

第二十九届前端早早聊大会

数据可视化 | 时空可视化 | 大屏 | 搭建 | 弯道超车



掘金

7月17日
全天直播

2021 全年行程

1/9 自由职业/副业

1/23 前端团队管理

2/6 小程序/组件化

2/27 页面搭建与

3/23 前端播客试

4/10 前端 UI/CD

4/24 前端新玩法

5/6 前端播客试

7/15 前端播客试

8/5 跨端 Flutter

8/22 前端播客试

8/25 前端 WebGL

7/17 前端播客试

7/24 前端播客试

8/14 前端播客试

8/28 前端播客试

8/11 Serverless

8/25 前端播客试

10/14 前端播客试

11/20 前端播客试

庀凤	贝壳找房	「基强大前端」 负责人	《前端工程师的可视化修炼之路》	9:00
徐小夕	自由职业	「H5-Deoring」 开源作者	《如何从零搭建全栈可视化大屏编辑器》	9:50
刘茜	奇安信	奇安信雷尔平台 核心开发	《如何打造沉浸式的时空可视化 Web 应用》	10:40
木的树	自由职业	「可视化」 技术专家	《如何从原理层面上手 Three.js》	11:30
芋头	预策科技	技术总监	《地理可视化，不止是炫酷》	13:00
正学	蚂蚁集团	地理可视化引擎 研发负责人	《如何设计与实现 L7 地理可视化引擎》	13:40
泽辉	小米	「体验效能」 前端可视化方向	《如何设计与实现思维导图可视化方案》	14:30
新若	蚂蚁集团	「AntV」 核心开发	《如何构思和开发开箱即用的图表库 - G2Plot》	15:20
逸达	阿里巴巴	图可视化编辑引擎 研发负责人	《如何打造超大规模图可视化画布》	16:10
--	数字冰雹	--	《--》	17:00
--	阿里云	DataV --	《--》	18:00

前端早早聊大会直播

2020 PK 2021

已举办 16 期 100 场

计划举办至少 20 期 140 场

1/11 前端转管理	6/20 前端跨端跨域
2/29 前端搞基建	6/27 前端女生专场
3/28 前端搞择理	7/18 前端搞可视化
4/11 前端搞规划	8/15 前端搞构建
4/25 前端搞造控	8/29 前端搞成长晋升
5/16 Serverless	9/20 前端搞报表
5/30 前端搞微前端	10/17 前端搞组件
5/31 前端搞面试	11/21 前端搞框架
6/13 前端搞文档	12/26 前端搞性能

1/10 前端搞创业	5/20 工程化/Flutter
1/23 前端搞管理	6/05 前端 Flutter
2/06 前端搞小程序	6/19 福利专场
2/27 可视化搭建	6/26 前端搞 WebGL
3/05 前端搞搭建	5/27 前端搞微前端
3/20 前端搞面试	7/17 前端搞可视化
3/27 菜鸟大前端	7/24 前端搞 BFF
4/10 CI/CD	8/28 前端搞安全
4/24 前端搞算法	9/11 Serverless
5/09 前端搞述职	9/25 前端搞 IoT
5/15 前端搞互动	10/16 前端搞造控
11/20 前端搞 IDE	12/11 玩转 Node.js

(以实际举办为准,行程/话题/场次会做动态调整)

早

一箱鲜由天下 重金争得蓝海
单主超5000 年营收300万

前端早早聊大会

2021年票

解锁 2021 年 140 场干货技术直播

单场大会用户

年票 VIP 用户

平均每场 77 元	平均每场 25 元
平均每场 12 元	平均每场 5 元
-	不限次数发布招聘
-	获得优质面试官推荐
-	Scott 简历指导及内推
-	2022 年票 <= 7 折
-	其他神秘福利...

¥660

每个月有直播
每一场有录播



2020 18期 PPT 录播 视频 讲稿

01.11	《前端如何转技术管理路线》	-----	5 位讲师, 5 篇讲稿, 5 份 PPT, 5 小时录播
02.29	《前端团队如何做技术基建》	-----	5 位讲师, 5 篇讲稿, 5 份 PPT, 5 小时录播
03.28	《前端 Lowcode 页面搭建》	-----	7 位讲师, 7 篇讲稿, 7 份 PPT, 7 小时录播
04.11	《前端职业规划与技术规划》	-----	4 位讲师, 4 篇讲稿, 4 份 PPT, 4 小时录播
04.25	《前端如何搭建监控体系》	-----	8 位讲师, 8 篇讲稿, 8 份 PPT, 7 小时录播
05.15	《前端如何借 Serverless 超车》	---	6 位讲师, 6 篇讲稿, 6 份 PPT, 6 小时录播
05.30	《前端如何落地实践微前端架构》	-----	7 位讲师, 7 篇讲稿, 7 份 PPT, 7 小时录播
05.31	《前端面试跳槽大厂的攻略经验》	-----	15 位讲师, 15 篇讲稿/份 PPT, 8 小时录播
06.13	《前端如何搞在线文档技术》	-----	4 位讲师, 4 篇讲稿, 4 份 PPT, 4 小时录播
06.20	《前端的跨端与跨栈》	-----	7 位讲师, 7 篇讲稿, 7 份 PPT, 7 小时录播
06.27	《前端女生的职业发展路线》	-----	10 位讲师, 10 篇讲稿/份 PPT, 10 小时录播
07.18	《前端数据可视化与数据商业化》	-----	10 位讲师, 9 篇讲稿, 9 份 PPT, 9 小时录播
08.15	《前端如何从 0 到 1 搞构建》	-----	8 位讲师, 8 篇讲稿, 8 份 PPT, 8 小时录播
08.29	《前端在大小厂的成长与晋升》	-----	8 位讲师, 8 篇讲稿, 8 份 PPT, 8 小时录播
09.25	《前端报表的快速搭建生产》	-----	5 位讲师, 5 篇讲稿, 5 份 PPT, 5 小时录播
10.17	《前端组件研发的各种玩法》	-----	5 位讲师, 5 篇讲稿, 5 份 PPT, 5 小时录播
11.21	《前端框架在多场景的研发模式》	-----	7 位讲师, 7 篇讲稿, 7 份 PPT, 7 小时录播
12.25	《前端在多端多场景的性能优化》	-----	7 位讲师, 7 篇讲稿, 7 份 PPT, 7 小时录播



2020 加入前端早聊群 提前三星期站在成长的新起点@CAIJIALI

前端早早聊联票上车通道



