
早 RingCentral
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
前
长按扫码报名，进群领取录播/讲稿/PPT

RingCentral

如何在大型应用里应用微前端

黄烈钦

RingCentral APP前端架构师

黄烈钦

RingCentral APP前端架构师

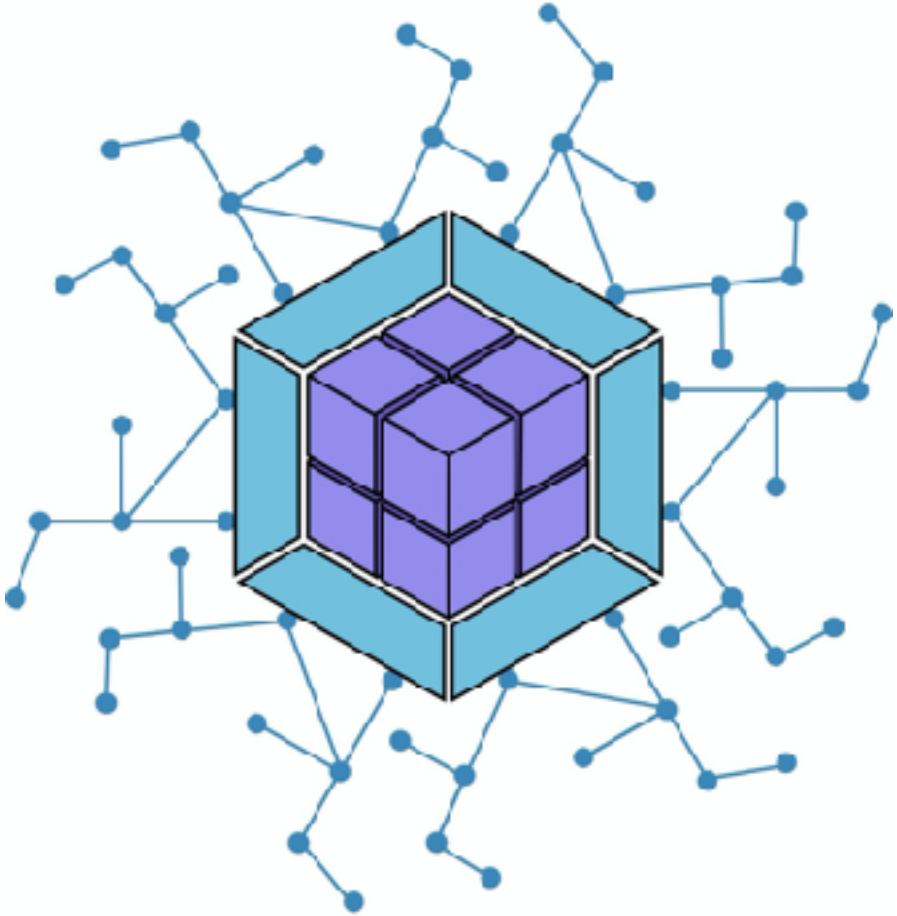
具有丰富的企业及系统架构设计经验

主导了模块化与微前端架构的设计与落地

RingCentral

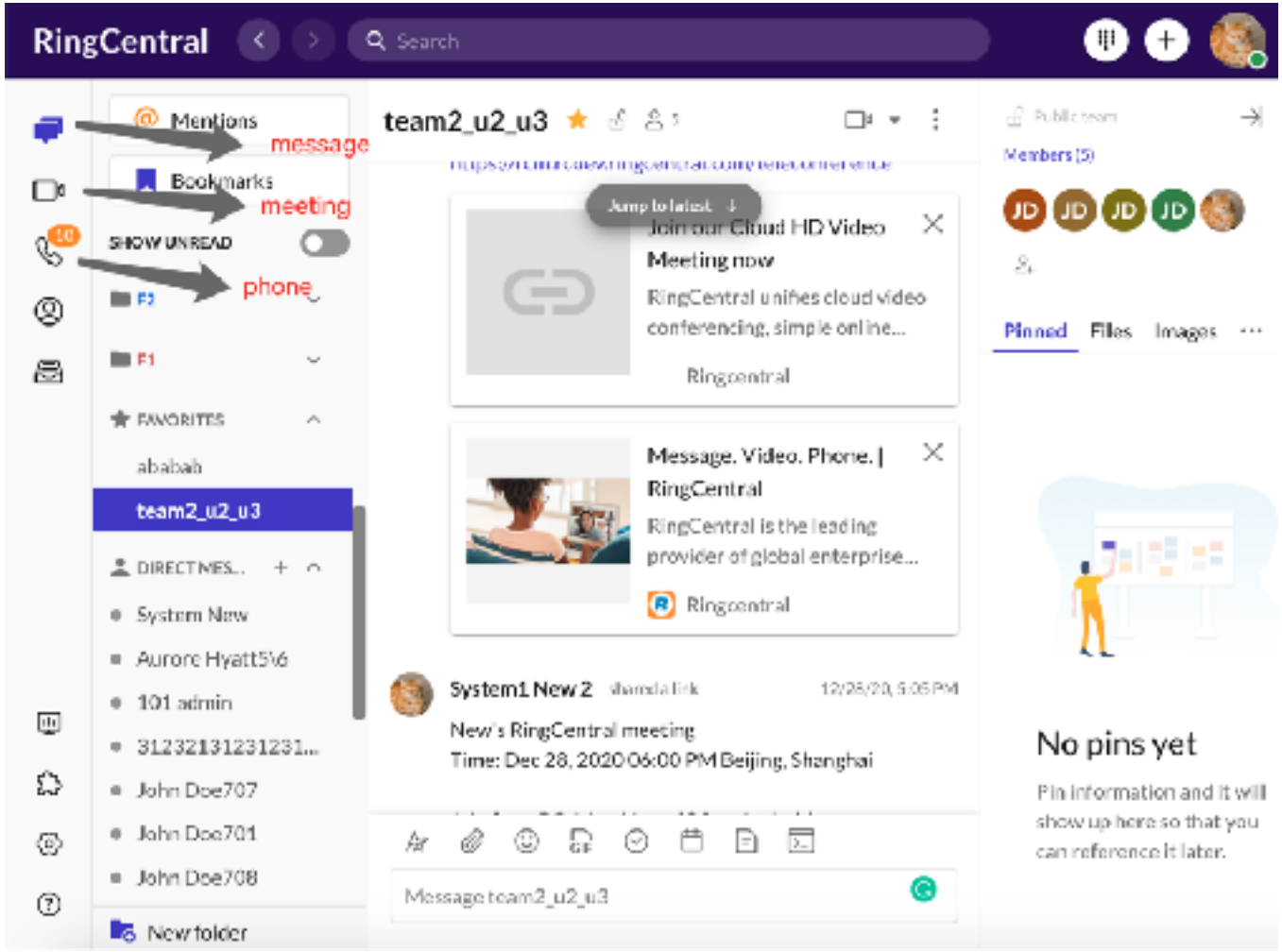
导航

1. 项目背景
2. 微前端方案
3. 微应用集成
4. 微前端架构背后的主要概念与解决方案



RingCentral

1.项目背景


RingCentral
Search
Mentions
message
Bookmarks
meeting
SHOW UNREAD
phone
F2
F1
FAVORITES
ababab
team2_u2_u3
DIRECT MES...
System New
Aurore Hyatt5\6
101 admin
31232131231231...
John Doe707
John Doe701
John Doe708
New folder
team2_u2_u3
Jump to latest
Join our Cloud HD Video Meeting now
RingCentral unifies cloud video conferencing, simple online...
Ringcentral
Message. Video. Phone. | RingCentral
RingCentral is the leading provider of global enterprise...
Ringcentral
System1 New 2
12/28/20, 5:05 PM
New s RingCentral meeting
Time: Dec 28, 2020 06:00 PM Beijing, Shanghai
Message team2_u2_u3
Members (5)
JD
Pinned
Files
Images
No pins yet
Pin information and it will show up here so that you can reference it later.

大型单体应用

开发效率
底

技术栈迁
移成本高

二方接入
成本高

建立体验良好、可持续维护迭代的系统

解决方案	简介	针对场景	优点	缺点	代表方案
iframe	不考虑体验的问题，基本上是最完美的微前端解决方安	第三方应用的接入	完美的硬隔离	1. URI 不同步 2. DOM 结构不共享 3. 慢	Luigi
基于路由的基座解决方案	基于路由分发，由一个基座应用来监听路由，按路由规则来加载不同的应用，以实现应用间解耦。	1. 多应用集成，如一体化工作台。 2. 大型单体应用拆分，包括应用/模块粒度	技术栈无关	1. 一定的接入改造成本 2. 没有好的依赖共享机制	Single-spa / Qiankun / Icestark

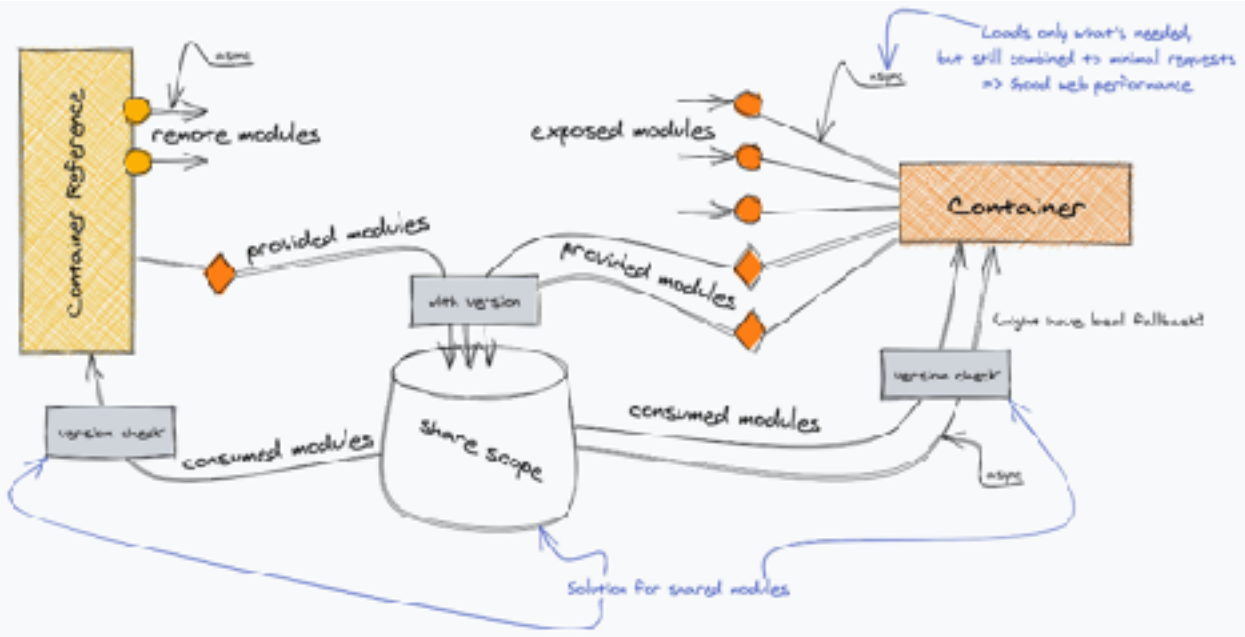
RingCentral

2.微前端方案

RingCentral

基于Module Federation

一种去中心化的方案，每个应用之间都可以彼此分享资源。

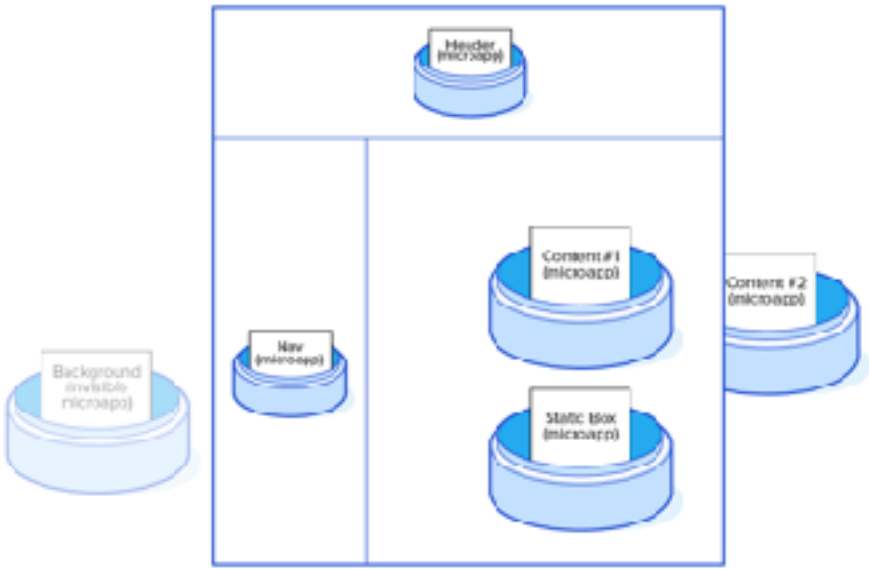


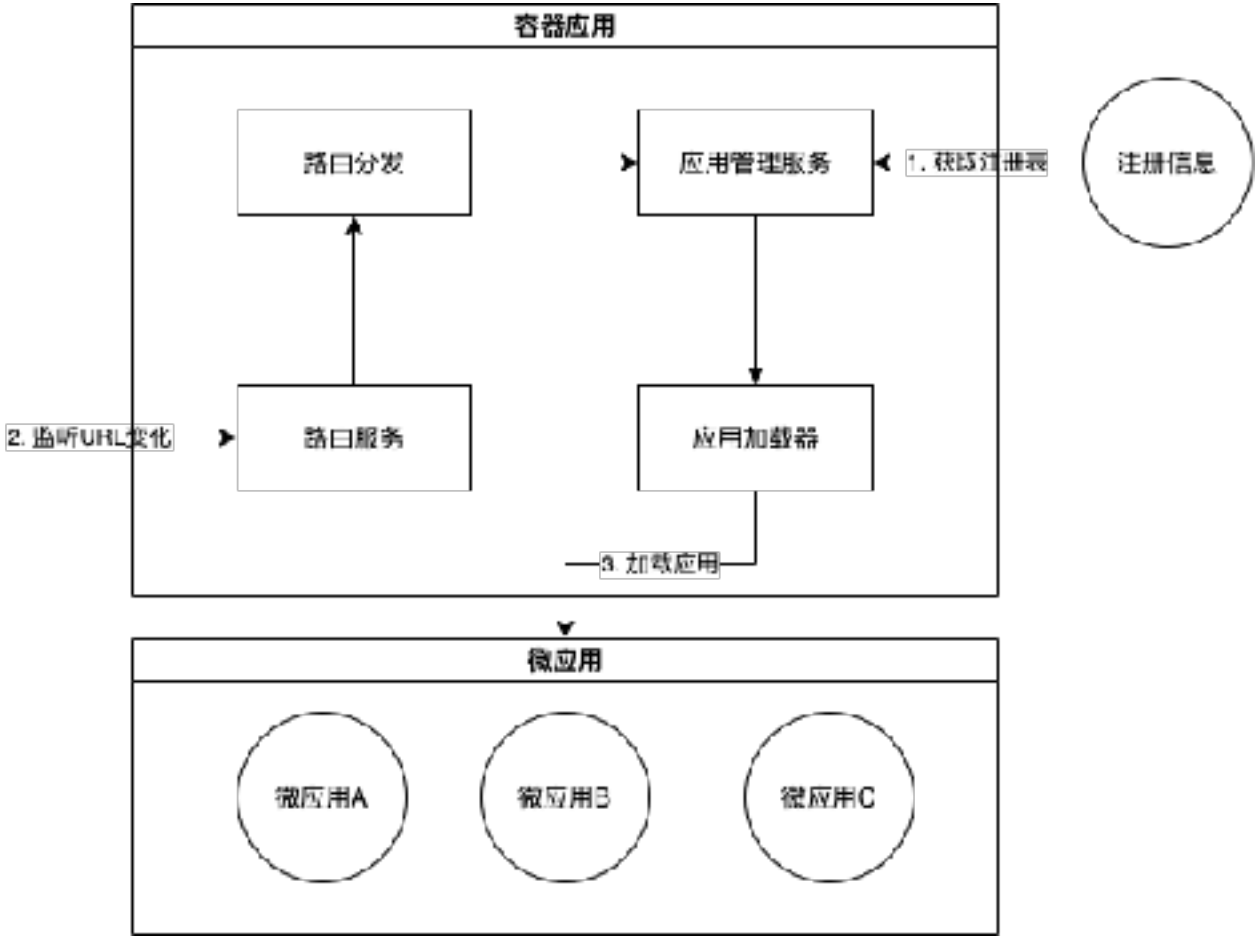
RingCentral

3.微应用集成

容器应用

- 微应用集成
- 微应用管理
- 认证管理
- 路由管理
- 渲染共用的页面元素，如header与footer




容器应用
路由分发
应用管理服务
1. 获取注册表
注册信息
2. 监听URL变化
路由服务
应用加载器
3. 加载应用
微应用
微应用A
微应用B
微应用C

集成协议

微应用对外提供expose模块。expose模块可以export下面三项

01	entry	• 可选 • 微应用视图入口
02	API	• 可选 • 微应用对外提供的公共接口
03	effect	• 可选 • 加载微应用时的副作用

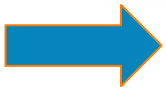
```
// can use dynamic import()
funcation entry() {
    return <div>sub app entry</div>;
}

const API = {
    doSomeing() {
        return 'do someing';
    }
}

function effect() {
    function focusListener() {};
    window.addEventListener('focus', focusListener);
    return () => {
        window.removEventListener('focus', focusListener);
    };
}

export { entry, API, effect };
```

依赖前置 - Webpack

同步转异步问题

```
// before
import Button from 'Mui/Button'

function ContainedButtons() {
    return <Button />;
}

// after: option one
const RemoteButton = React.lazy(() => import('remote/Mui/Button'));

const ContainedButtons = () => (
    <React.Suspense fallback="Loading Button">
      <RemoteButton />
    </React.Suspense>
)

// after: option two
// app.js
import Button from 'remote/Mui/Button'

function ContainedButtons() {
    return <Button />;
}

// index.js
import('./app');
```

```
__webpack_require__.e = function requireEnsure(chunkId) {
    ...
    // start chunk loading
    var head = document.getElementsByTagName("head")[0];
    var script = document.createElement('script');
    script.type = 'text/javascript';
    script.charset = 'utf-8';
    script.async = true;
    script.timeout = 120000;

    ...

    head.appendChild(script);
    return promise;
}
```



```
__webpack_require__.e = (chunkId) => {
    Promise
        .all(Object.keys(__webpack_require__.f)
        .reduce((promises, key) => {
            __webpack_require__.f[key](chunkId, promises);
            return promises;
        }, []));
};
```

```
// overridables 可覆盖的, shared配置有关
__webpack_require__.f.overridables = (chunkId, promises) => {}

// remotes 远程的, remotes配置有关
__webpack_require__.f.remotes = (chunkId, promises) => {}

// jsonp加载chunk函数
__webpack_require__.f.j = (chunkId, promises) => []
```

4.微前端架构背后的主要概念与解决方案

RingCentral

两大共性问题

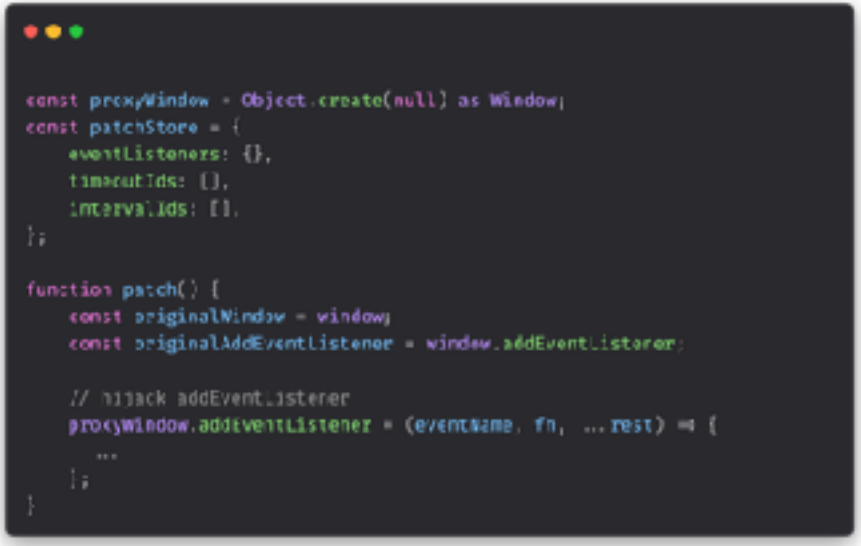
- 应用的加载与切换
  - 路由/手动
  - 应用加载
  - 应用入口
- 应用的隔离与通信
  - JS 隔离
  - 样式隔离
  - 父子、子子之间的通信

脚本隔离


code
创建沙箱
const sandbox = new Proxy(Object.create(null), {
get(){},
set(){},
has(){},
...
});
执行代码
const execScript = `with (sandbox) {;${script}\n}`;
const execCode = new Function('sandbox', execScript);
execCode(sandbox);
Window
快照记录
与恢复
window

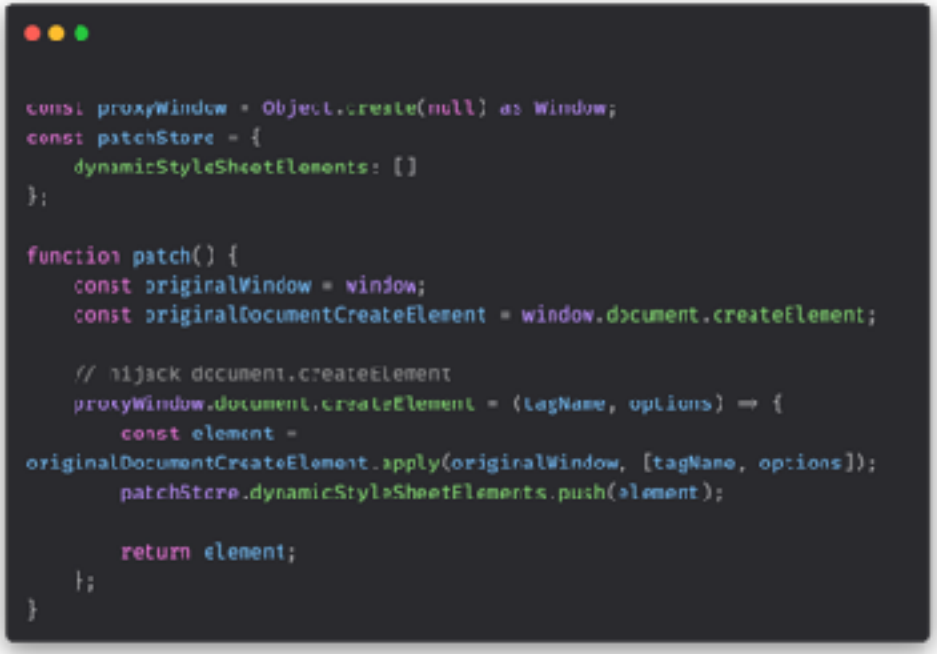
脚本隔离

- addEventListener/removeEventListener
- setTimeout/setInterval
- document.createElement



样式隔离

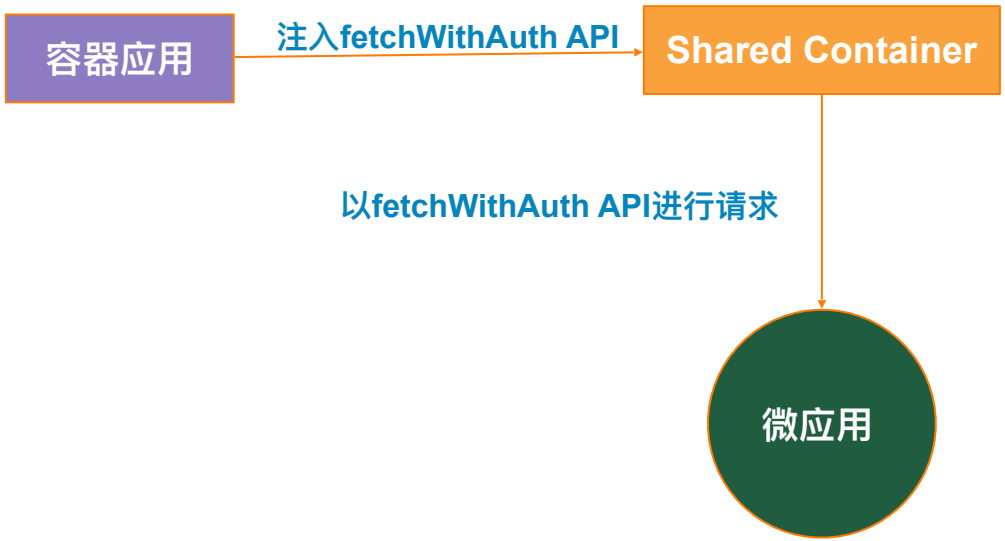
- BEM
- CSS Module
- CSS-in-JS (styled-components)



```
const proxyWindow = Object.create(null) as Window;
const patchStore = {
    dynamicStyleSheetElements: []
};

function patch() {
    const originalWindow = window;
    const originalDocumentCreateElement = window.document.createElement;

    // hijack document.createElement
    proxyWindow.document.createElement = (tagName, options) => {
        const element =
originalDocumentCreateElement.apply(originalWindow, [tagName, options]);
        patchStore.dynamicStyleSheetElements.push(element);

        return element;
    };
}
```

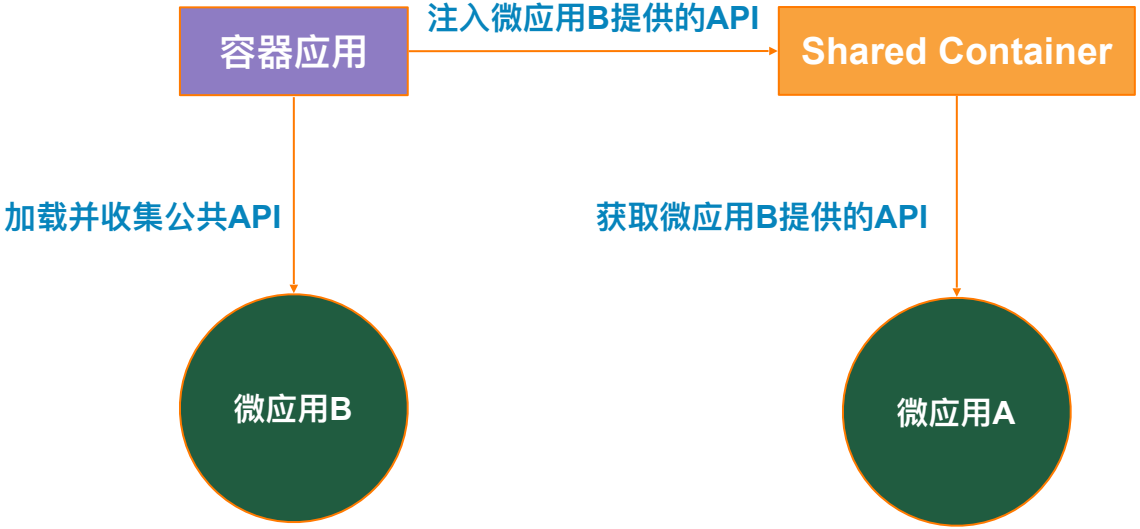
通信 - 后端

容器应用注入网络请求接口到Shared Container。微应用可以从Shared Container拿到网络请求接口



通信 - 微应用

容器应用加载微应用并收集其提供的API注入到Shared Container



Shared Container

- 独立Repo
- 单例

```
// micro app a
// api.js
let status = '';

const setStatus = (newStatus) => status = newStatus;
const getStatus = () => status;

export { setStatus, getStatus }

// somefile.js
import { setStatus, getStatus } from './api';

setStatus('anyStatus');

import('appB/api');

// micro app b
// api.js
import { getStatus } from 'appA/api';

console.log(getStatus());
```

RingCentral

```
const installedModules = {}

const modules = {
    './src/main.js': (function (module, __webpack_exports__,
__webpack_require__) {}),
}

function __webpack_require__(moduleId) {
    // Check if module is in cache
    if (installedModules[moduleId]) {
        return installedModules[moduleId].exports;
    }
    // Create a new module (and put it into the cache)
    var module = installedModules[moduleId] = {
        exports: {}
    };
    // Execute the module function
    modules[moduleId].call(module.exports, module, module.exports,
__webpack_require__);
    // Return the exports of the module
    return module.exports;
}
```

代码共享

基于ModuleFederationPlugin

- exposes：指明了这个容器的消费者将可以获取什么模块
- shared：指明了那些依赖包是可以共享的

原则

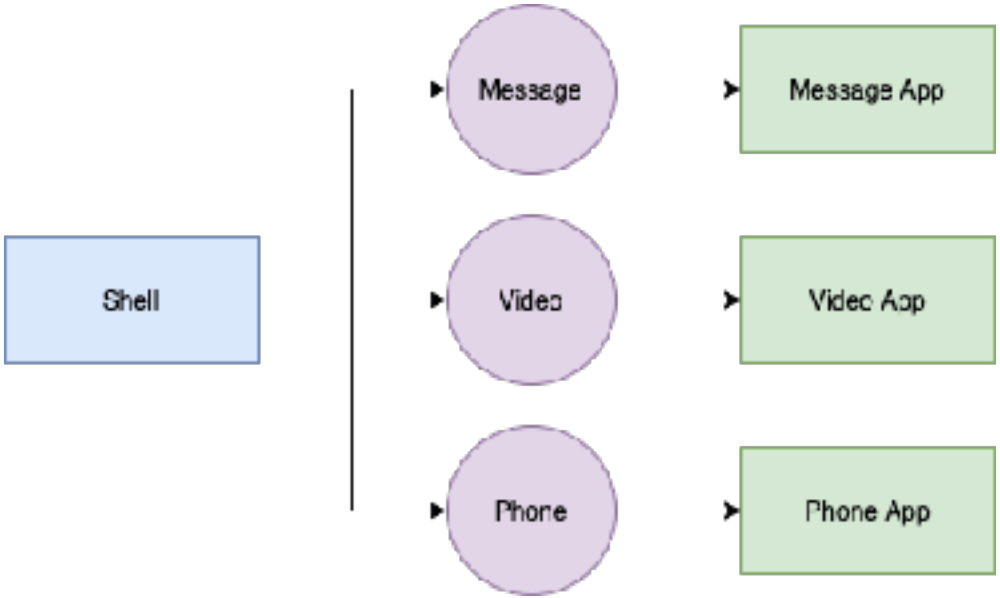
- exposes: 有状态的组件
- shared: 无状态的组件

路由

- 容器应用基于React Router触发加载和渲染微应用
- 容器应用指定微应用的一级路由
- 微应用有自己的理由管理

测试 - E2E

容器应用与微应用集成后(自动化)就是一个可以独立运行的应用



部署

- 所有的微应用共享同一个域名（https://app.ringcentral.com）
- 以path来区别不同的微应用（https://app.ringecntral.com/{appName}）

目录结构

- {version}
  - index.html
  - remoteEntry.js
- release
  - index.html
  - remoteEntry.js

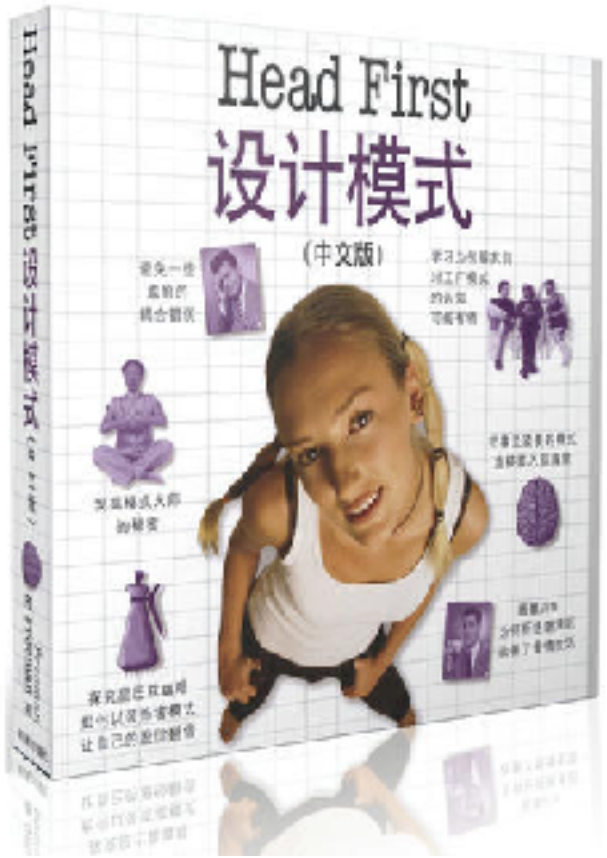
版本管理

01 容器应用自动集成微应用的最新版本

02 微应用自己确保与容器应用的兼容性

03 微应用的入口文件不缓存或采用stale-while-revalidate缓存策略(remoteEntry.js)

书籍推荐



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

RingCentral

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。

Forbes 2020 GLOBAL 2000 WORLD'S LARGEST PUBLIC COMPANIES

HR Asia BEST COMPANIES TO WORK FOR IN ASIA 2020

美世 中国卓越健康雇主 CHINA HEALTHIEST WORKPLACE

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
- 超长年假
- 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*

RingCentral

Q&A

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

7 月 17 日 全天直播

2021 全年行程

庞凤	贝壳找房	「基础大前端」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-Dooring」开源作者	《如何从零搭建全栈可视化大屏编辑器》	9:50
刘芃	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

1/11 前端转管理	6/20 前端跨端跨栈
2/29 前端搞基建	6/27 前端女生专场
3/28 前端搞搭建	7/18 前端搞可视化
4/11 前端搞规划	8/15 前端搞构建
4/25 前端搞监控	8/29 前端成长晋升
5/16 Serverless	9/26 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

计划举办至少 20 期 140 场

1/10 前端搞副业	5/29 工程化/Flutter
1/23 前端搞管理	6/05 跨端 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 前端搞监控
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)

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
平均每期 77 元	平均每期 25 元
平均每场 12 元	平均每场 5 元
-	不限次数发布招聘
-	获得优质简历投递
-	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 《前端如何落建监控体系》 ---------- 8 位讲师，8 篇讲稿，8 份 PPT，7 小时录播

05.16 《前端如何借 Serverless 起车》 --- 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

09.26 《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

前端早早聊联票上车通道

2020 加入前端早早聊，提前三星期站在成长的新起点

2020 全年录播视频 + PPT

前端早早聊联票上车通道

2020 加入前端早早聊，提前三星期，站在成长的起点 @ZAOTALK

2020

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
01.25	《前端如何搭建垂控体系》	8 位讲师，8 篇讲稿，8 份 PPT，7 小时录播
05.16	《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，15 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，10 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 搭构建》	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
09.26	《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，6 小时录播
12.16	《前端如何做性能优化》	7 位讲师，7 篇讲稿，7 份 PPT，6 小时录播