



创心 第十四届
D2前端技术论坛

云时代的前端架构

微前端架构体系

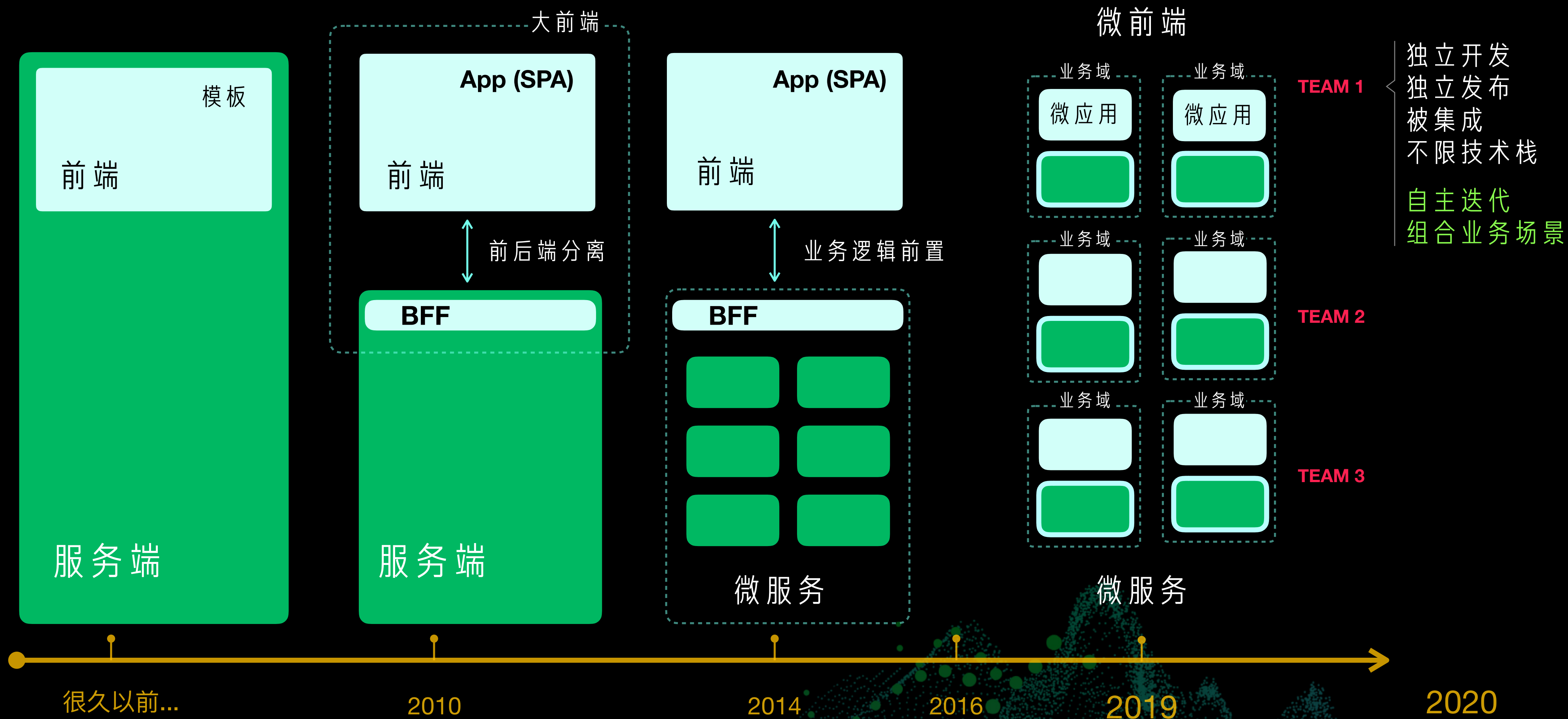


克军

阿里云智能安全前端团队负责人



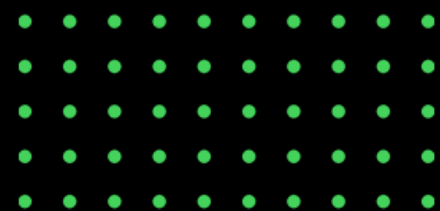
微前端（它是什么、从哪来、到哪去（怎么做））







不吹捧





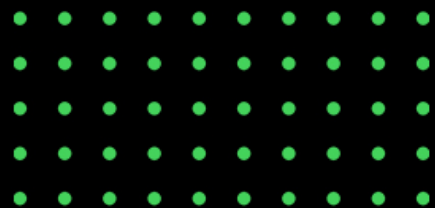
?

跟iFrame、Web Components、NPM包、BigPipe、插件有什么区别？
多页应用需要微前端吗？为什么非要做成单页应用？



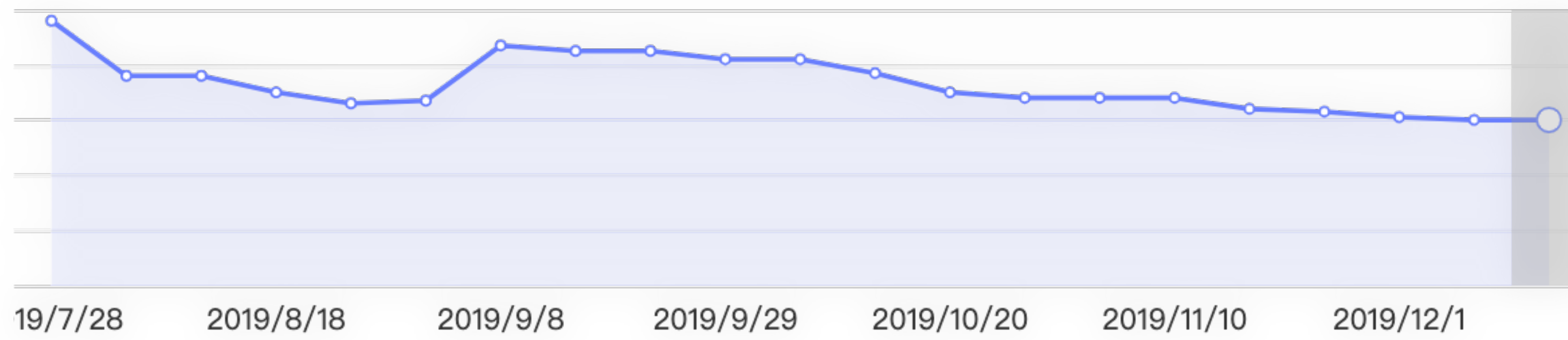
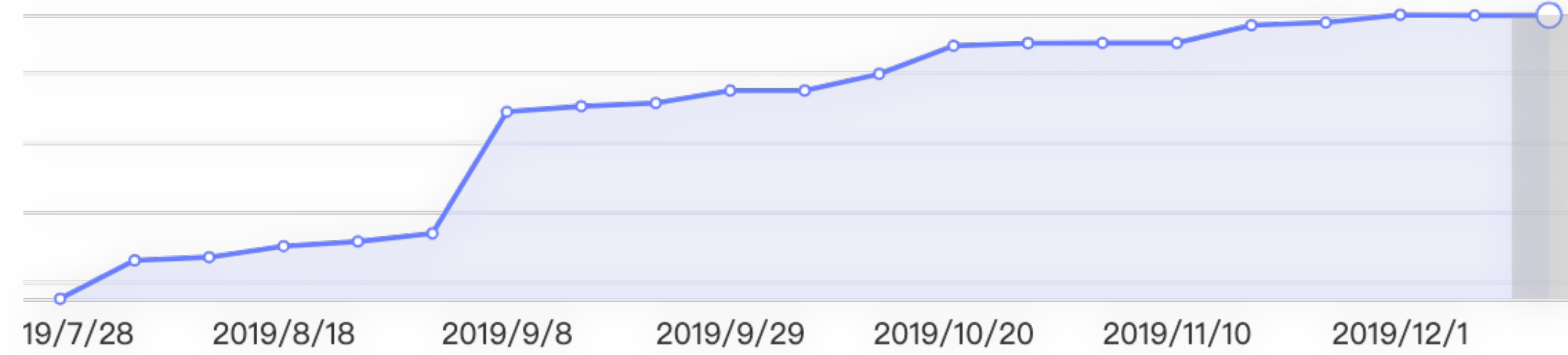


微前端	Widget / 业务组件
架构体系。用来实现大型Web应用	以库（外联/npm）的形式实现复用
生产方式	生产工具
通过隔离机制实现技术栈无关	需要人工解决依赖和冲突问题
单独构建\单独发布\热升级	整体构建\整体发布
体系化治理，可控性强	可控性差
主从关系（路由映射、消息机制）	相互无关
微应用是产品的子集（粒度大）	通用功能（粒度小）
变化快	变化小
若干微应用的组合	“外挂”



400万个零件、30个国家：空客A380是怎么造出来的？







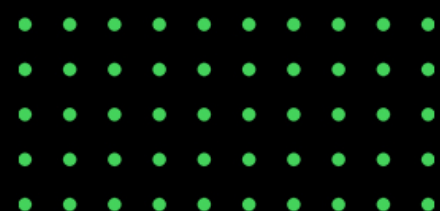
微前端的工程价值

优点

1. 独立开发和部署
2. 大型单页应用无限扩展
3. 不限技术栈
4. 快速整合业务
5. 多团队协作

缺点（微前端之“熵”）

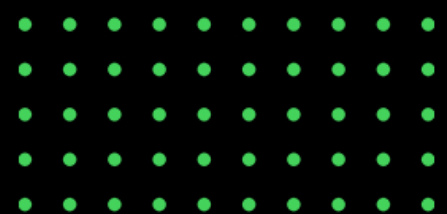
1. 体验有折损
2. 维护成本高
3. 管理版本复杂、依赖复杂
4. 开发体验不太友好
5. 粒度不宜太小





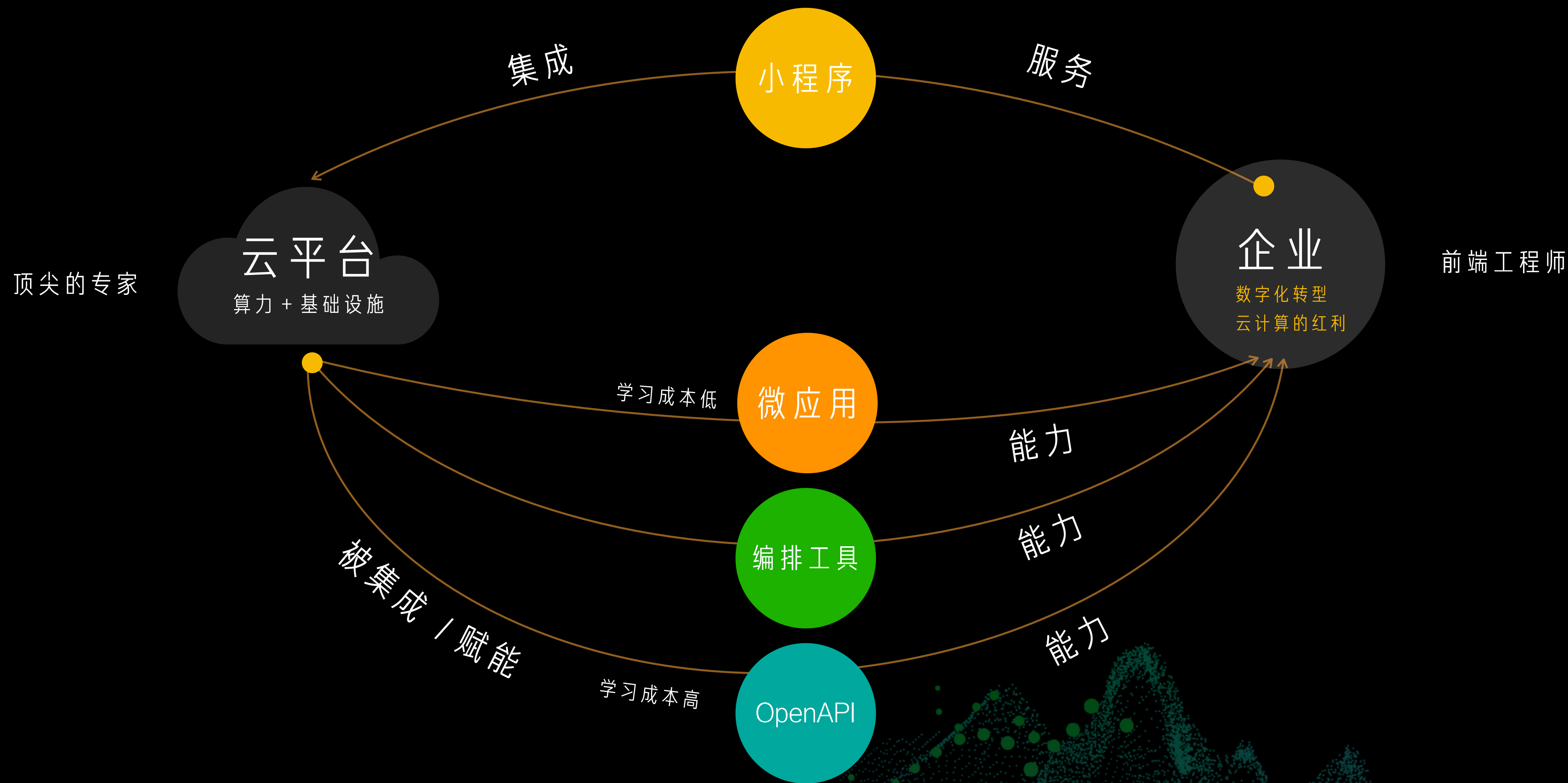
微前端的业务价值

1. 产品“原子化”：扩展性、组合性、局部迭代
2. 解决能力输出“最后一公里”的价值
3. 云生态中的“新物种”——微应用



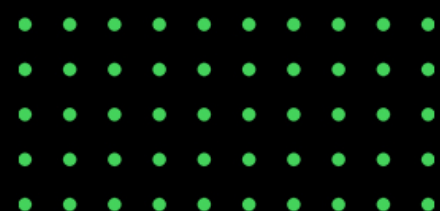
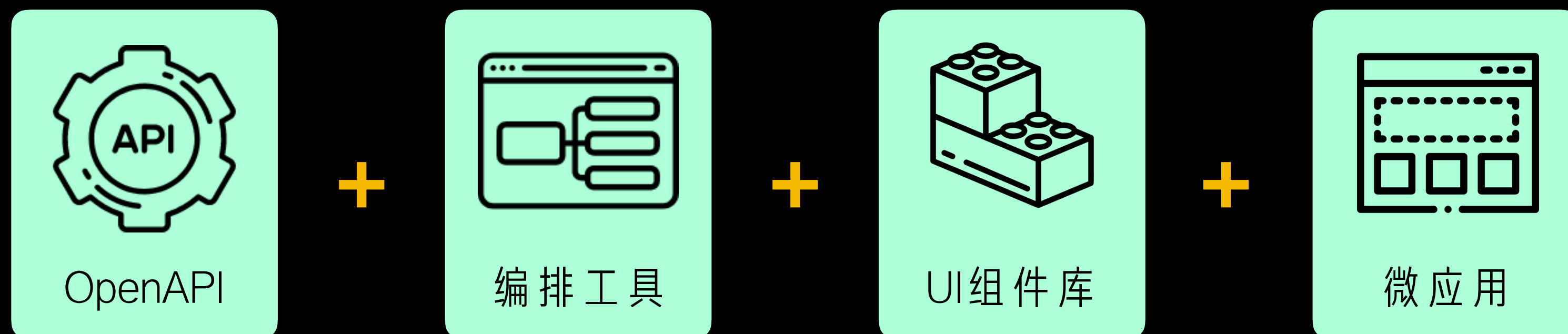


云生态 × 微前端





idea 1: 云时代的企业开发套件

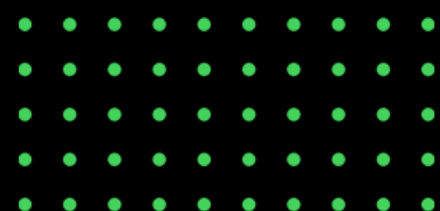




idea 2: OpenComponents (2014)

<https://opencomponents.github.io>

开放的微应用市场

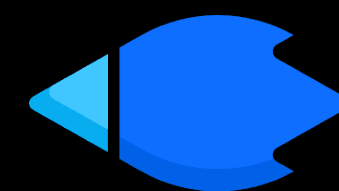




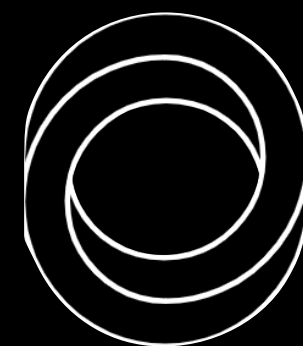
MicroX

敬请期待

M A G I X



icestark



WidgetJS

qiankun
乾坤



single-spa



Ara Framework

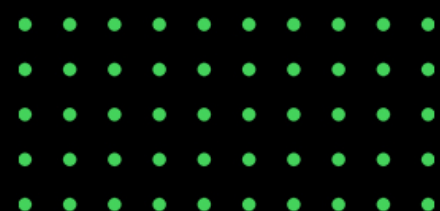


Luigi



建体系：

微前端不是一个框架 / 工具
而是一套架构体系

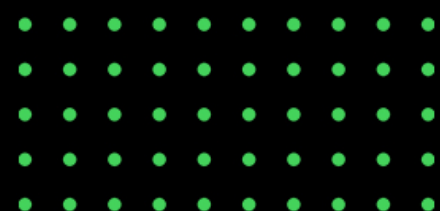




微前端体系

(分布式体系的中心化治理很重要)

1. 微前端基础设施 (No Framework特点)
2. 微前端配置中心 (版本管理、发布策略、动态构建)
3. 微前端观察工具 (运维职能: 可见、可控性)





微前端

- | | |
|--------|------|
| 微应用列表 | 版本管理 |
| 依赖管理 | 访问策略 |
| 灰度、A/B | 参数配置 |

配置中心

平台

- | | |
|------|-------|
| 异常监控 | 代码健康度 |
| 行为日志 | 性能指标 |
| 安全检测 | 运维数据 |

观察工具

平台

微应用市场

平台



配套设施

- | | | |
|-------|------|------|
| 项目脚手架 | 动态构建 | 一键部署 |
| 依赖分析 | 代码质量 | 访问控制 |
| 开发环境 | 集成测试 | 国际化 |

本地开发工作台

工具集

- | | |
|------|------|
| 异步加载 | 生命周期 |
| 沙盒隔离 | 异常处理 |
| 路由映射 | 消息机制 |

微应用容器

库

对接层

- | | | |
|--------|------|---------|
| Mock数据 | 接口编排 | GraphQL |
|--------|------|---------|

数据网关

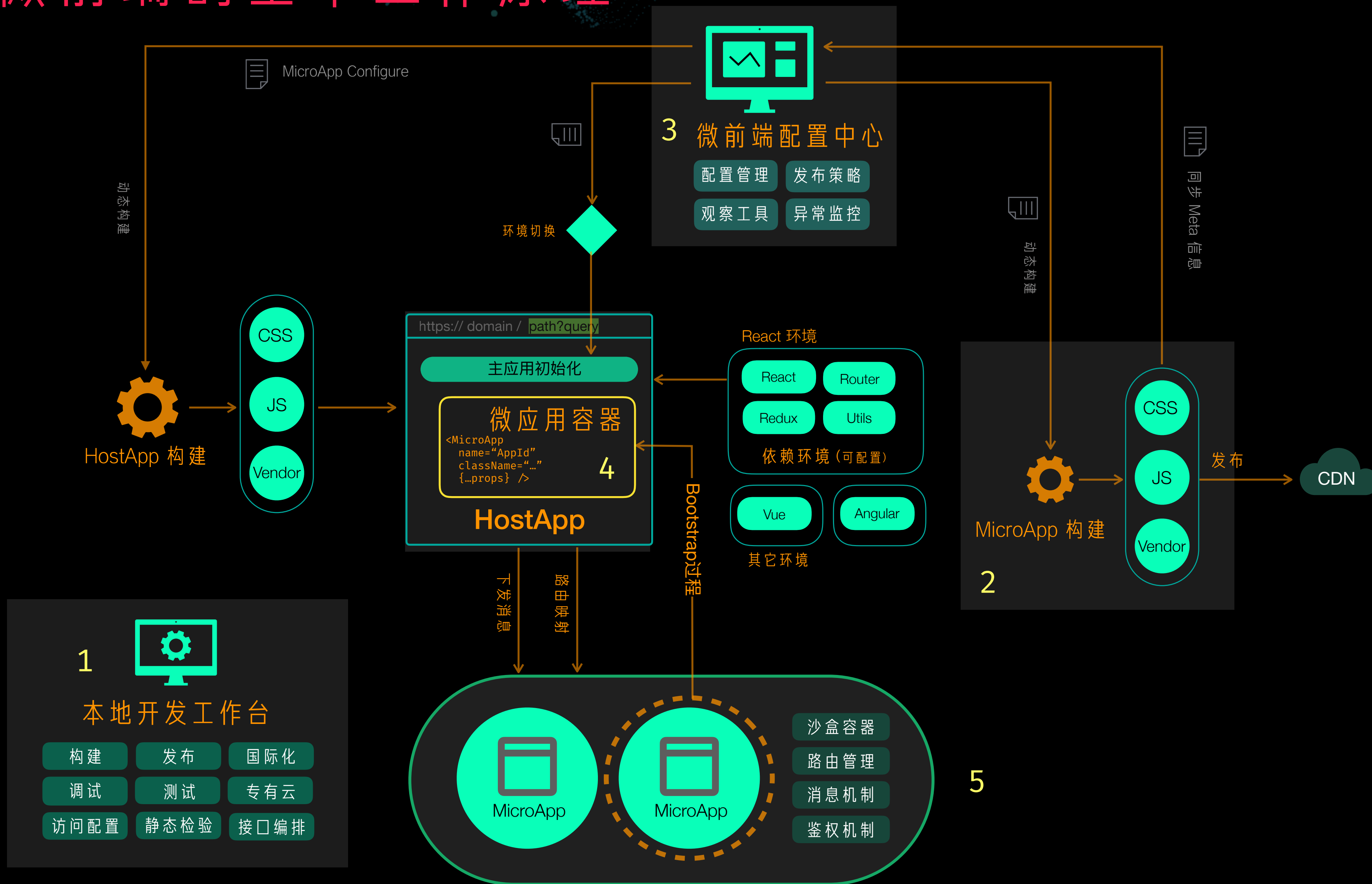
平台 / 库



Serverless服务



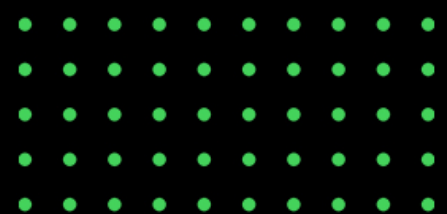
微前端的基本工作原理





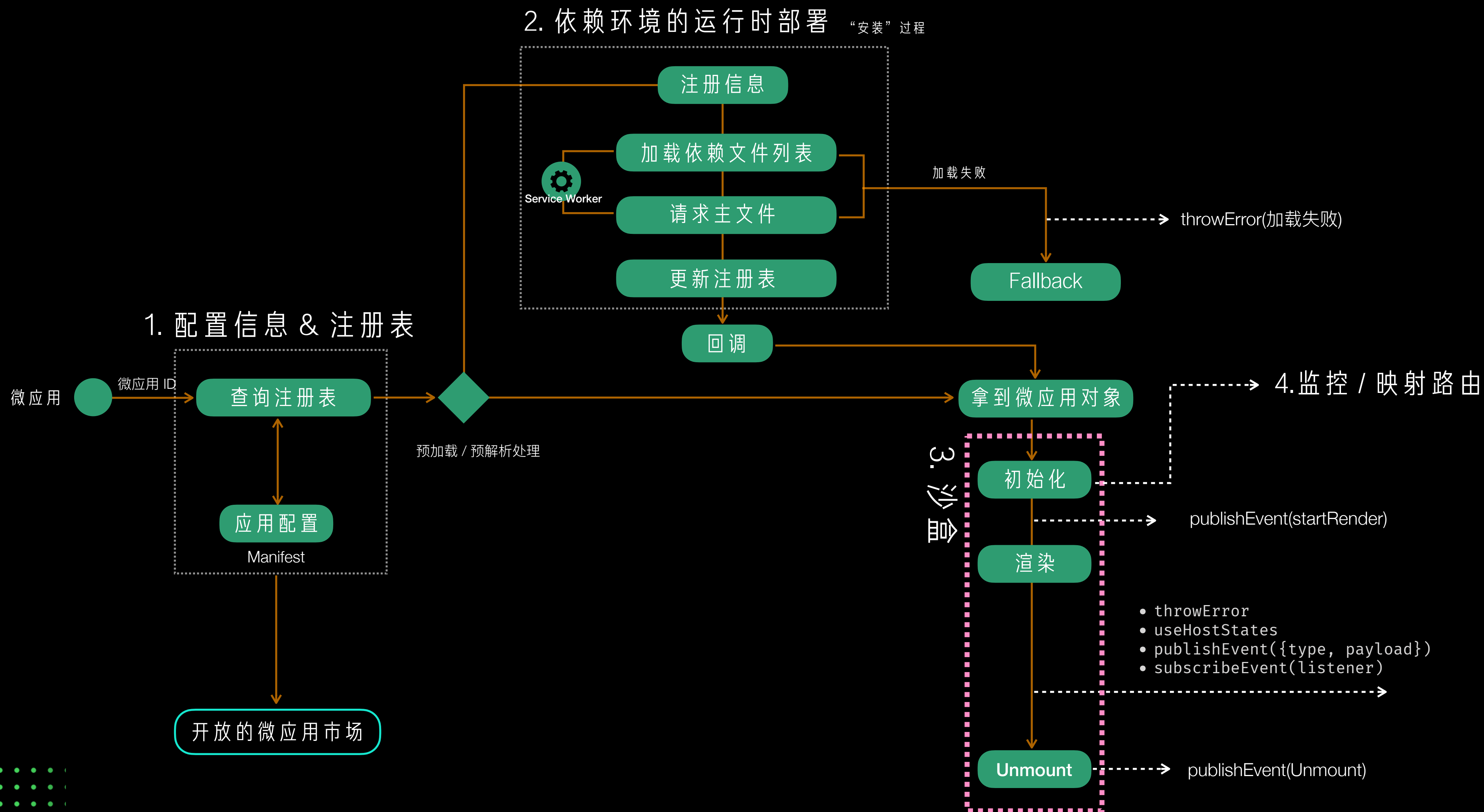
微前端的10个问题 (10-factor)

1. 微应用的注册、异步加载和生命周期管理
2. 微应用之间、主从之间的消息机制
3. 微应用之间的隔离措施
4. 微应用的框架无关、版本无关
5. 公共依赖的库、业务逻辑(utils)以及版本管理
6. 微应用独立调试、和主应用联调
7. 微应用的发布流程
8. 微应用打包优化问题
9. 微应用专有云场景的出包方案
10. 渐进升级：平滑改造老项目





微应用的加载和运行机制





微应用容器能力

安全

分级

- ◆可信（一 / 二方 / 可信任的平台）
- ◆不可信（三方 / ISV）

- ◆iFrame方案
- ◆前置的静态检查
- ◆安全的DSL

隔离

- ◆隔离样式: ShadowDOM
- ◆隔离 JavaScript / DOM 侵入性

回弹

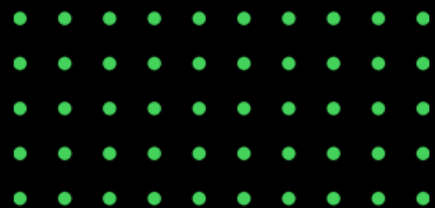
- ◆加载失败的Fallback
- ◆容错(Error Boundaries)、上报错误信息

可见

- ◆埋点监控
- ◆性能、健康度
- ◆Inspector

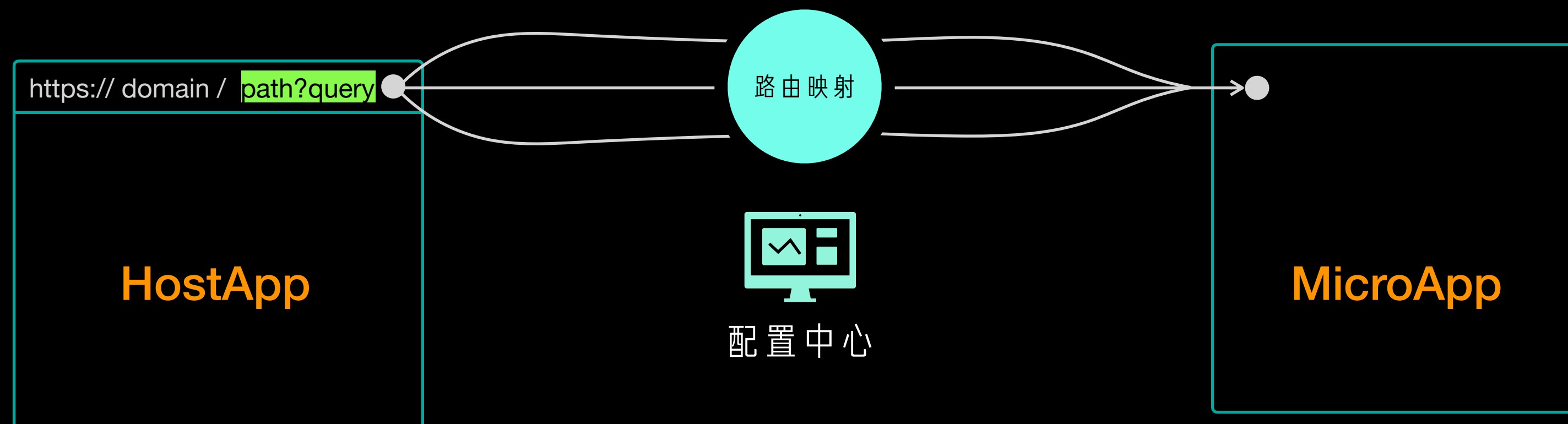
通讯

- ◆消息机制



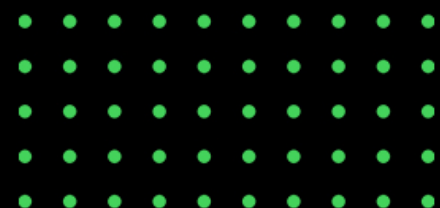


路由管理



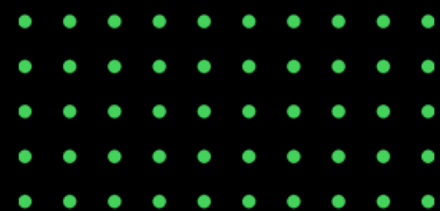
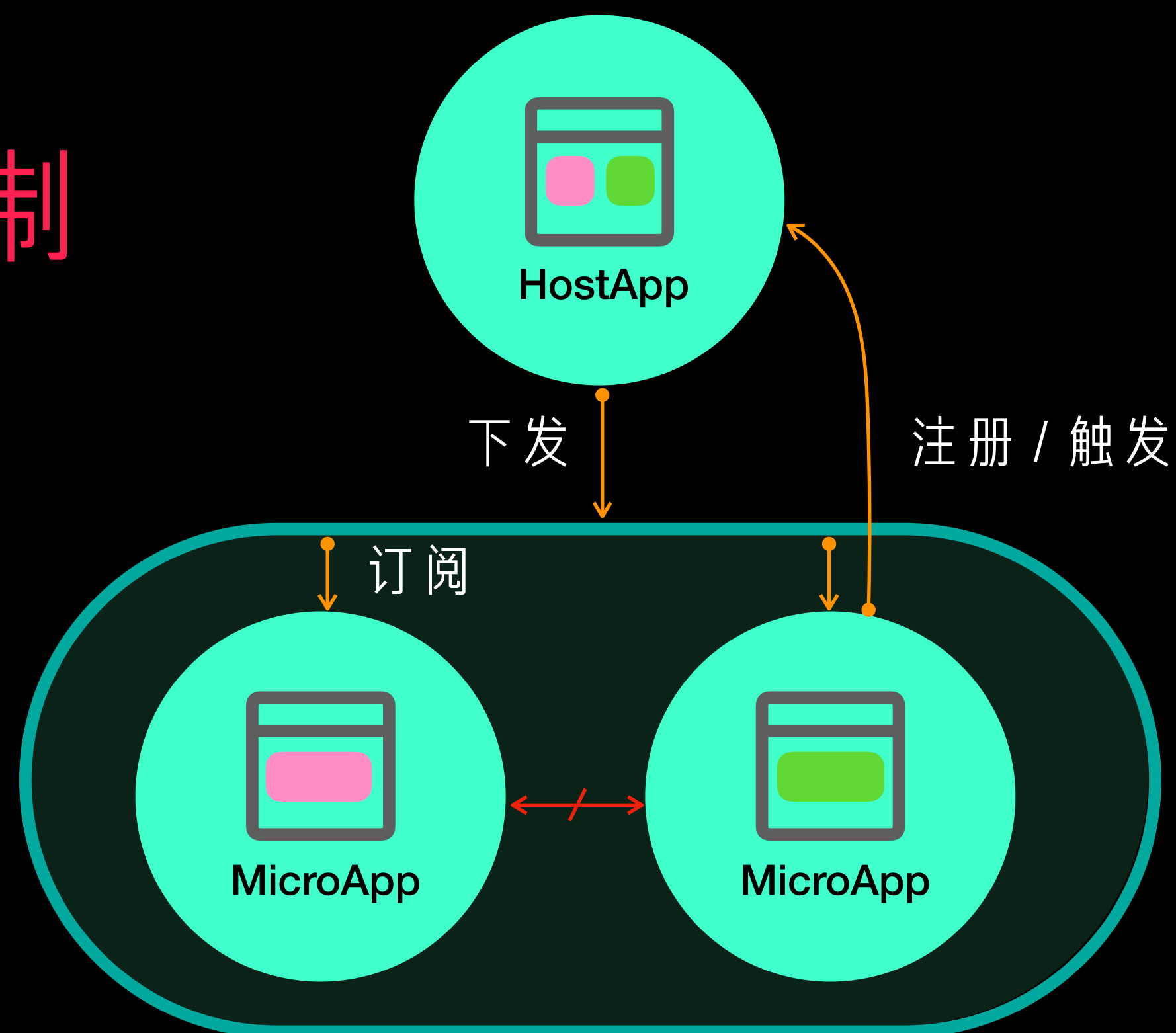
路由配置:

```
{
  routeMapping: {
    '/vulManage': '<AppId>', // 默认前缀
    '/vulManage/linux': '<AppId>linux',
    '/vulManage/window': '<AppId>window',
    '/vulManage/cve/:id': '<AppId>detail/:id',
    .....
  }
}
```





消息机制



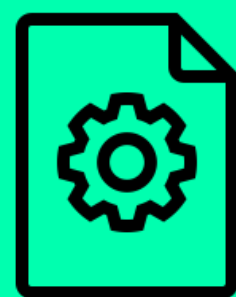
望楼系统





微前端的“望楼系统”（可观察性工具）

除了监控（系统健康度指标、告警和阈值），系统运行状态和可控性（观察工具、跟踪、调试…）



配置

- 版本灰度
- 版本回滚
- A/B测试
- 访问控制



代码质量

- 健康状况
- 安全检测
- 维护者



观察

- 微应用列表
- 微应用参数
- 公共依赖
- 公共组件
- 异常监控
- 行为日志



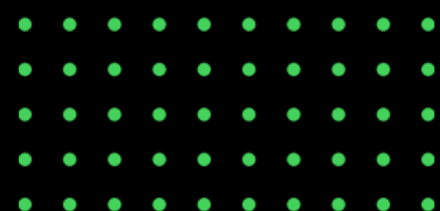
数据服务

- 接口编排
- GraphQL



度量

- 加载速度（性能指标）
- 访问量（PV/UV）





前端架构体系不可忽略运维能力

运维系统的设计原则

WHEN DESIGNING FOR OPERABILITY REMEMBER: C O D S

Controlability 可控	Observability 可观察	Decomposability 分拆	Simplicity 自动化
<u>CONTROL RISK EXPOSURE</u> - Blue-Green Deploys - Feature Toggles - Canary Releases - A/B testing - Shadowing	<u>OBSERVE SYSTEM BEHAVIOUR</u> - Instrumentation - Logging - Monitoring - Alerts - Dashboards - Synthetic Tests - Data Analytics - Tracing - Profiling	<u>DECOMPOSE DEPLOYS</u> - Single changeset per deploy	<u>SIMPLIFY DEPLOYS</u> - Single click of a button to deploy or Roll-back

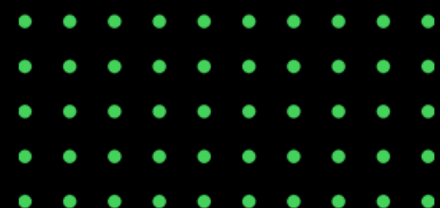
来自 Rob Meaney

<https://twitter.com/RobMeaney/status/1177010238307934210/photo/1>



微前端配置中心方案

可灰度、可回滚、中心化管理





MicroX - 微前端

首页公共微应用UI基础组件开发文档设置

云防火墙

项目发布

微应用

VM页面

项目质量

代码质量

错误监控

文件引用分析

首页 / 微应用管理

发布状态: 全部 启用状态: 全部 微应用名称或描述信息 搜索 添加微应用

主应用: root 项目主应用 发布 取消

当前版本: 1.0.0

灰度比例: 测试账号

灰度账号: 12121212

引用资源: http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.js

依赖资源: http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.js

是否启用: ☒

type微应用 version1.0.12 50%

appid (MicroApp ID)sas-asset desc安全中心-资产中心

errors3 maintainability82 complexity17 jscpd2.4% eslint0 / 11

resourcesstorybook

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.js

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.css

type微应用 version1.0.12 50%

appid (MicroApp ID)sas-asset desc安全中心-资产中心

errors3 maintainability82 complexity17 jscpd2.4% eslint0 / 11

resourcesstorybook

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.js

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.css

type微应用 version1.0.12 50%

appid (MicroApp ID)sas-asset desc安全中心-资产中心

errors3 maintainability82 complexity17 jscpd2.4% eslint0 / 11

resourcesstorybook

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.js

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.css

type微应用 version1.0.12 50%

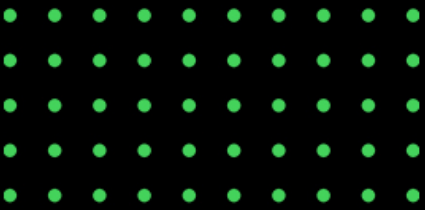
appid (MicroApp ID)sas-vulmanage desc安全中心-漏洞管理

errors3 maintainability82 complexity17 jscpd2.4% eslint0 / 11

resourcesstorybook

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.js

http://g.alicdn.com/[group.name]/[project.id]/[app.version]/index.min.css





基本信息

异常监控

资源文件

type

微应用

version

1.0.12

50%

appid (MicroApp ID)

sas-asset

desc

安全中心-资产中心

errors

3

maintainability

82

complexity

17

jscpd

2.4%

eslint

0 / 11

resources

+

storybook

编辑、隐藏等操作

当前版本 & 灰发控制

代码健康度

演示、文档



钉钉自助建群

?

有问题，找专家

Knobs

Apis

用法

<UI.MicroApp name="ddg" />

Copy

属性

参数	说明	类型	默认值
config	配置项	object	见config配置
handler	从外部控制浮层弹出	function	
haselIrl	ani的hasel IRI	string	



总结 微前端是云时代的前端架构体系

建体系

基础设施

全链路机制

核心能力

治理

配置中心

运维能力

术语

微前端

微应用

公共依赖

微应用容器

微应用配置文件

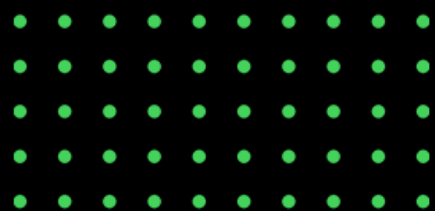
引导 (Bootstrap)

本地工作台

观察性

微前端治理

配置中心





Q & A

谢谢！



扫码参与调查

