

# 阿里跨境供应链前端架构演进与 Serverless 实践

繆传杰 (樵枫)

阿里巴巴 高级前端技术专家

# 樵枫 / 缪传杰

- ICBU 深圳多元化履约前端团队负责人
- 参与推进集团 FaaS 产品 0→1 建设
- 爱好喝酒、洗脚、吃生蚝

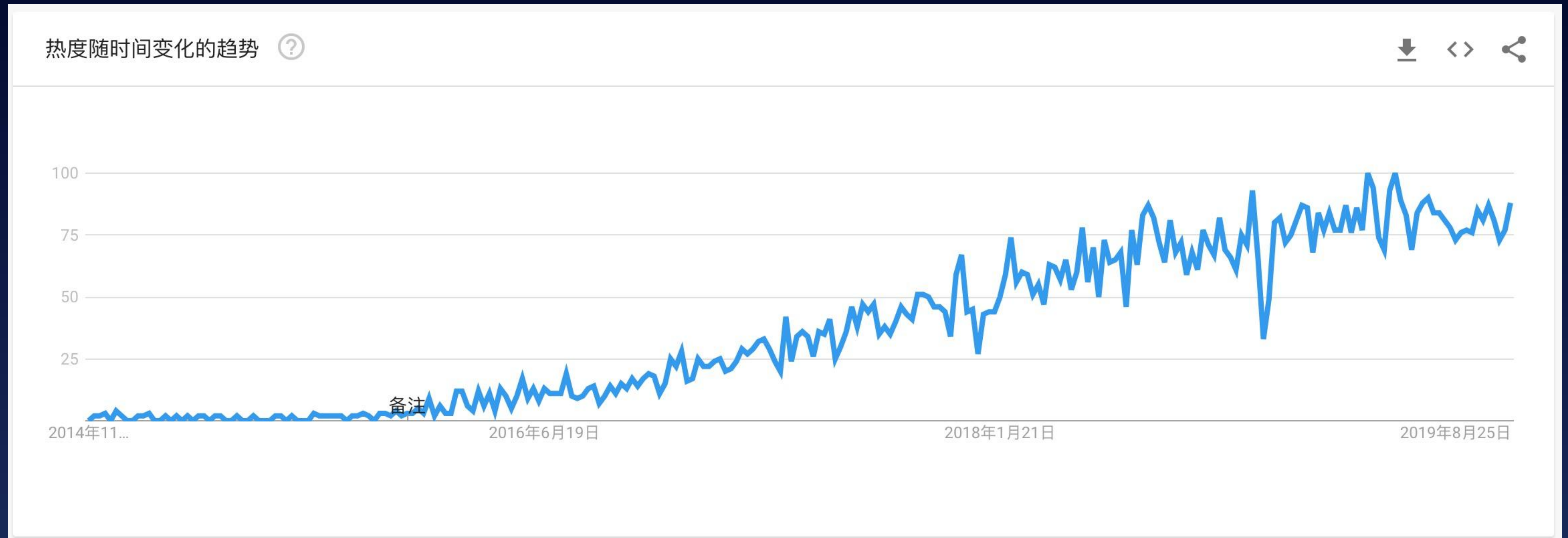


# 目录

- 1、Serverless 技术浪潮
- 2、ICBU 跨境供应链业务背景与前端架构演进
- 3、Serverless 实践与配套工程、监控能力介绍
- 4、阶段结果&展望

# Serverless 技术浪潮

# Serverless 技术浪潮



“Serverless is like teenage sex: Everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it.”

—— Juven Xu

Serverless = 广义 FaaS + BaaS

高效能

按量付费

免运维

聚焦业务逻辑，减少工程链路消耗和运维成本  
用最小的成本透出业务领域能力，赋能业务增值

# ICBU 跨境供应链业务介绍与前端架构演进

# ICBU 跨境供应链业务





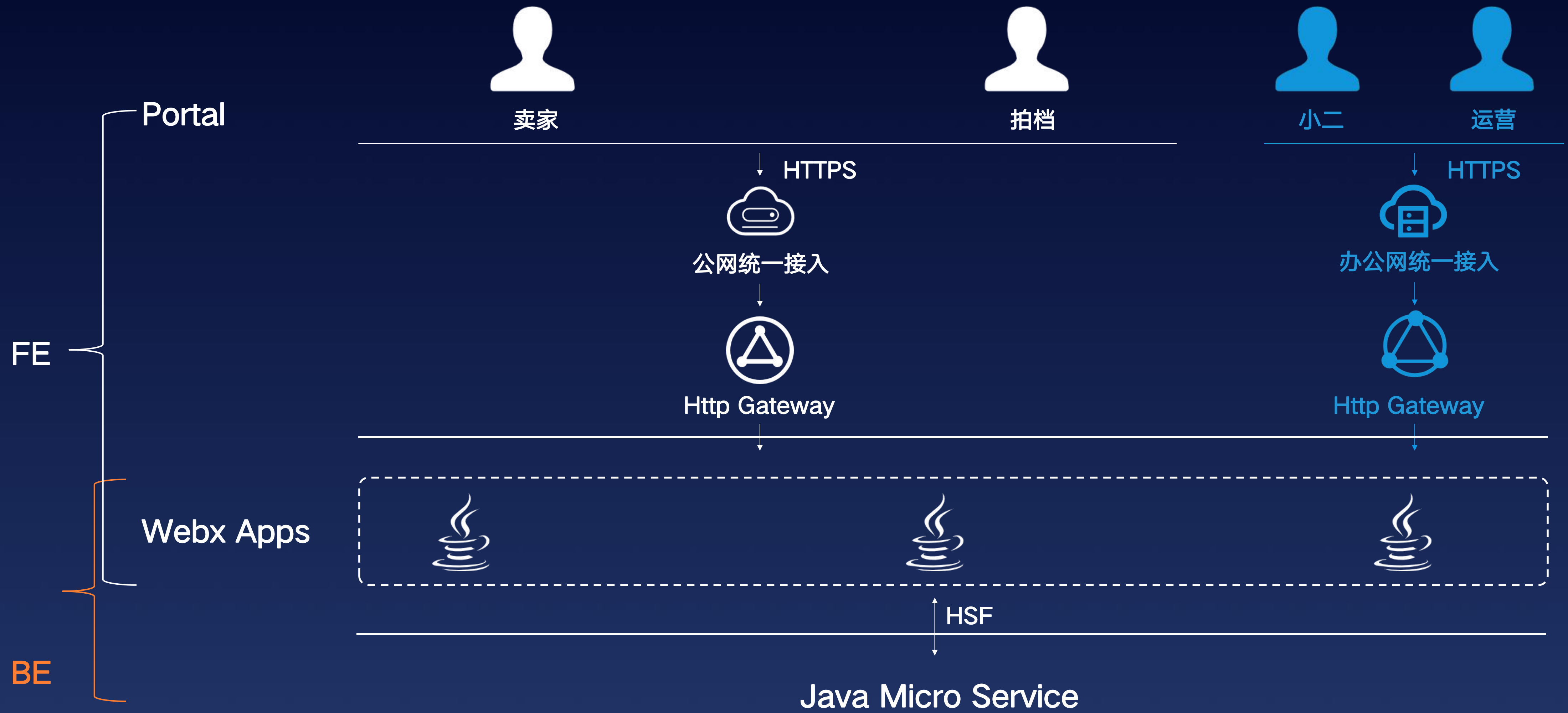
# ICBU 跨境供应链业务关键词

跨境 B2B

专业性极强

大量的中后台

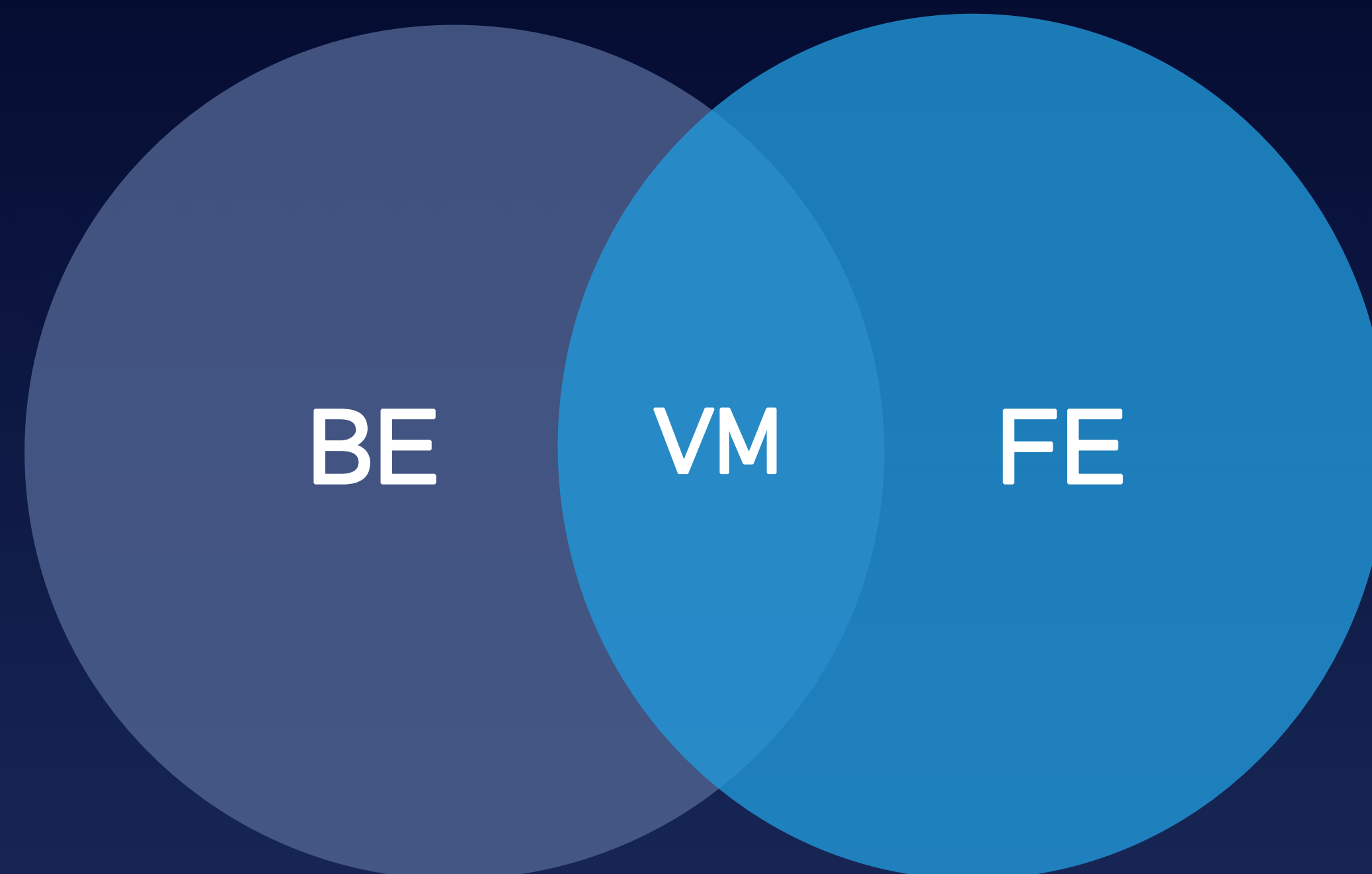
# 青铜时代 - 2016 年前



## 痛点 —— VM 层的耦合带来协作、效率、质量问题

- 前后端职责不清，容易扯皮，不利专业发展；
- 前端重度依赖后端环境，开发、联调成本高，效率低；
- 浏览器端和 Web 层采用不同技术栈，组件无法复用；
- 前后端耦合暴露了很多质量问题

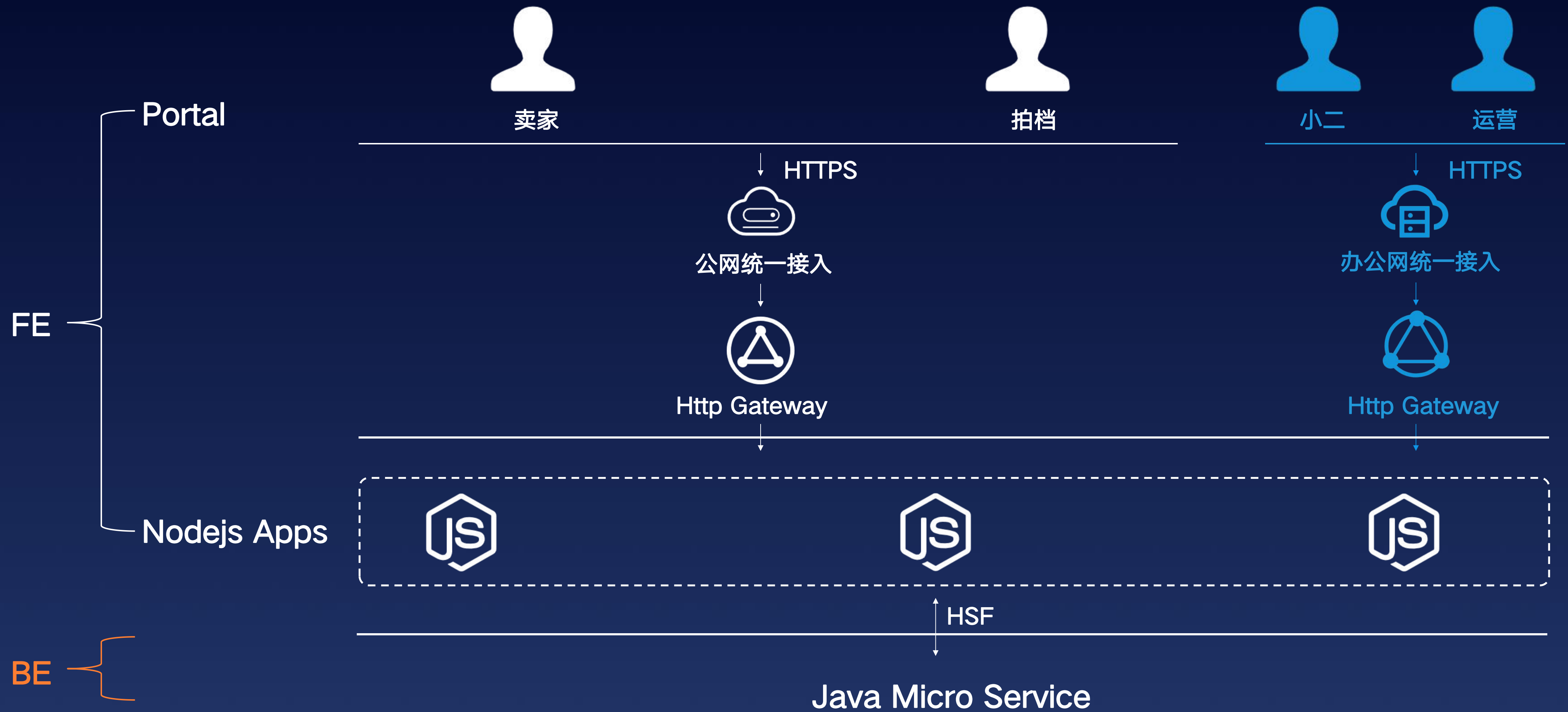
## 痛点 —— 技术栈严重限制了团队发展和开发幸福感



“吾日三省吾身：表单做好了吗？列表做好了吗？详情做好了吗？”

—— 樵枫

# 白银时代 — 2016 至 2018 年



# 白银时代 — 2016 至 2018 年



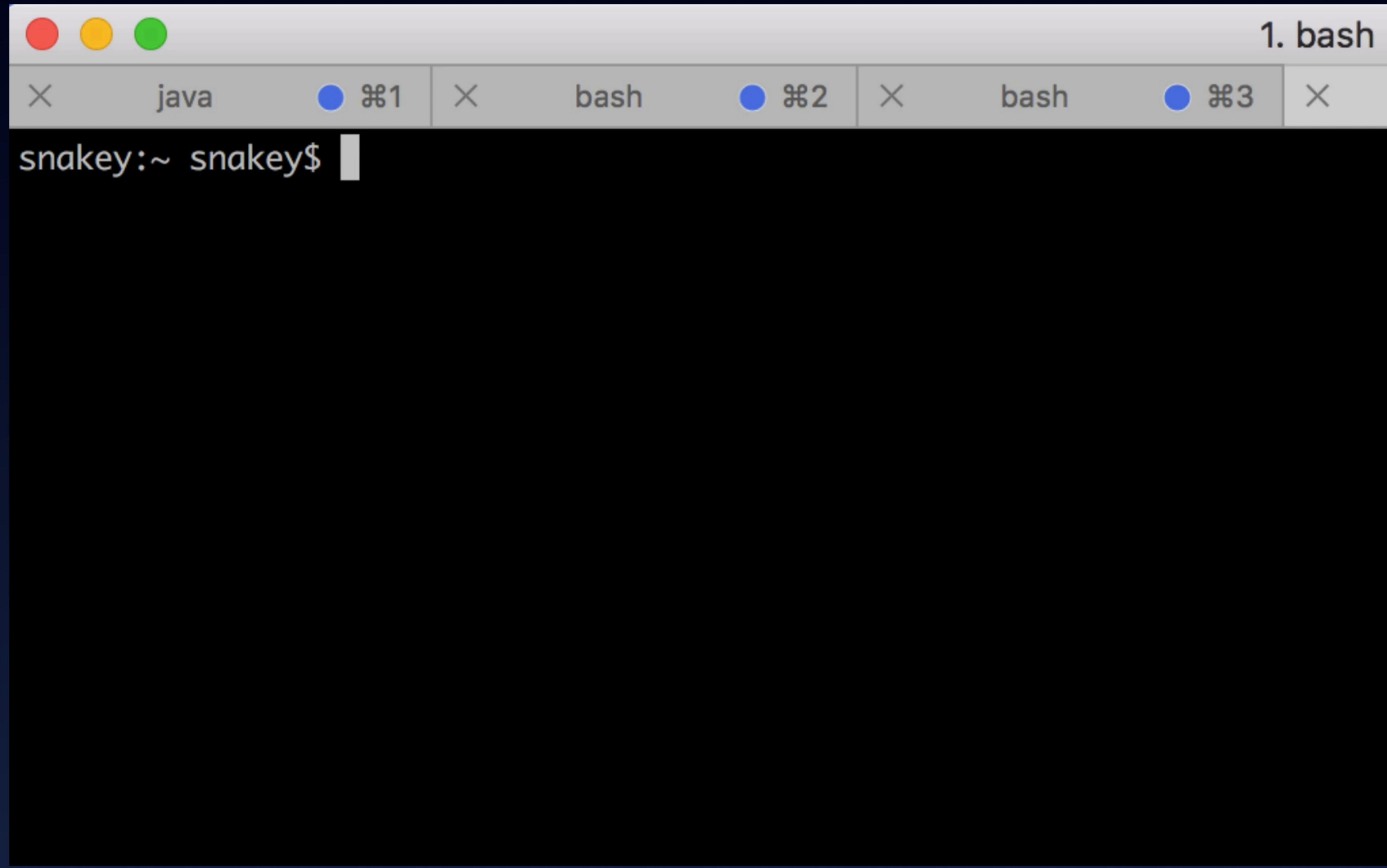
集团约有 Node.js 应用 1600+, BFF 应用数量约占 70%  
常年水位在 10% 以下

# BFF Next 路在何方...

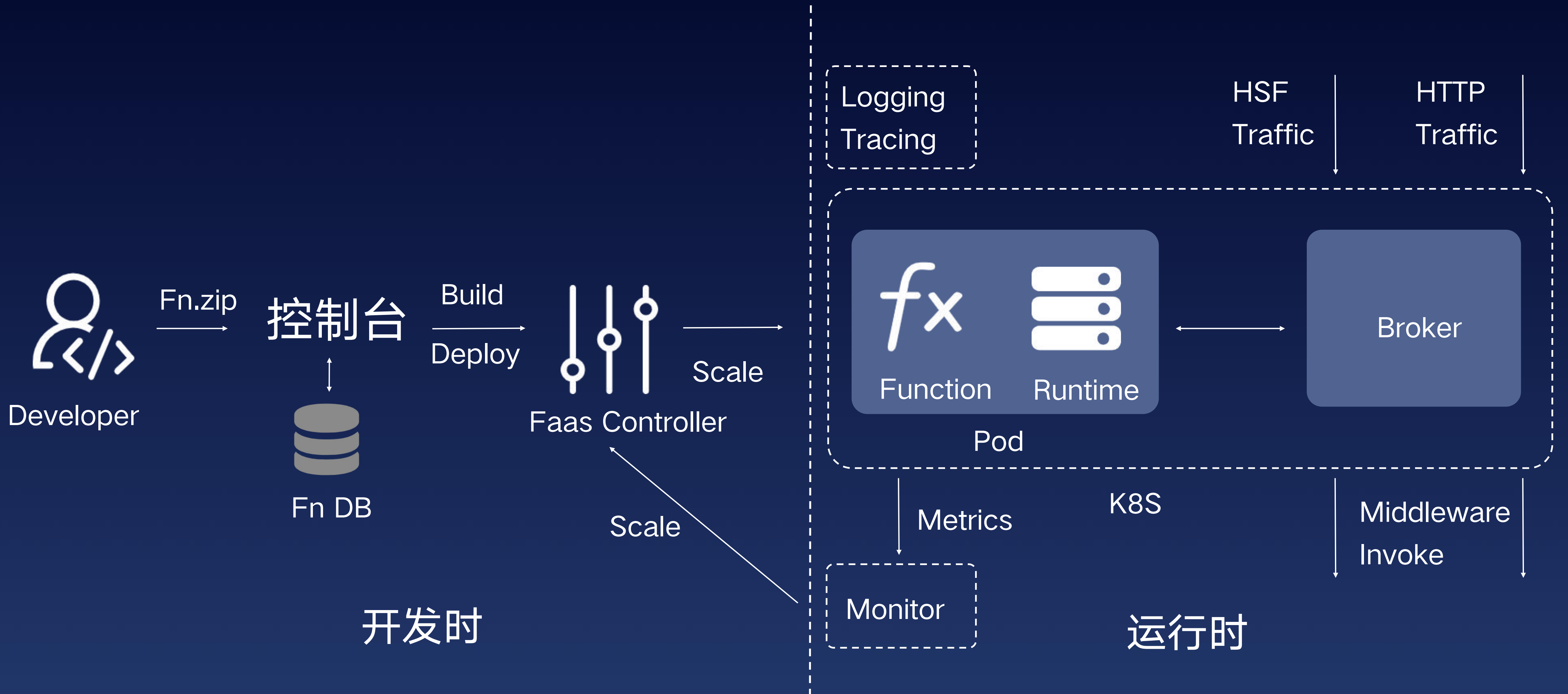
# Serverless 实践与配套工程、监控能力



体感



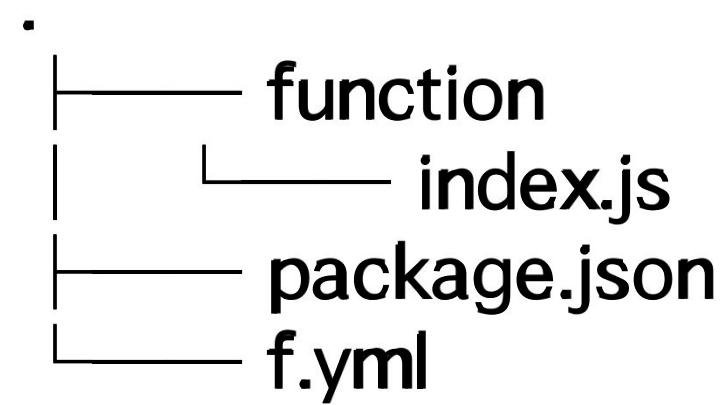
# FaaS 整体架构



# FaaS Nodejs 生态



# 函数规范



handler.js

```
exports.handler = async (ctx, param1, param2) => {  
}
```

f.yml

```
service: my-test  
provider:  
  name: ginkgo  
  runtime: nodejs10  
functions:  
  main:  
    handler: index.handler  
    events:  
      - http: true  
    layers:  
      - "egg-layer"  
      - "ginkgo-layer-onetouch"  
      - "bianque-layer"  
layers:  
  egg-layer:  
    path: oss:@ali/egg-layer@latest  
  ginkgo-layer-onetouch:  
    path: oss:@ali/ginkgo-layer-onetouch@latest  
  bianque-layer:  
    path: oss:@ali/egg-bianque@latest
```

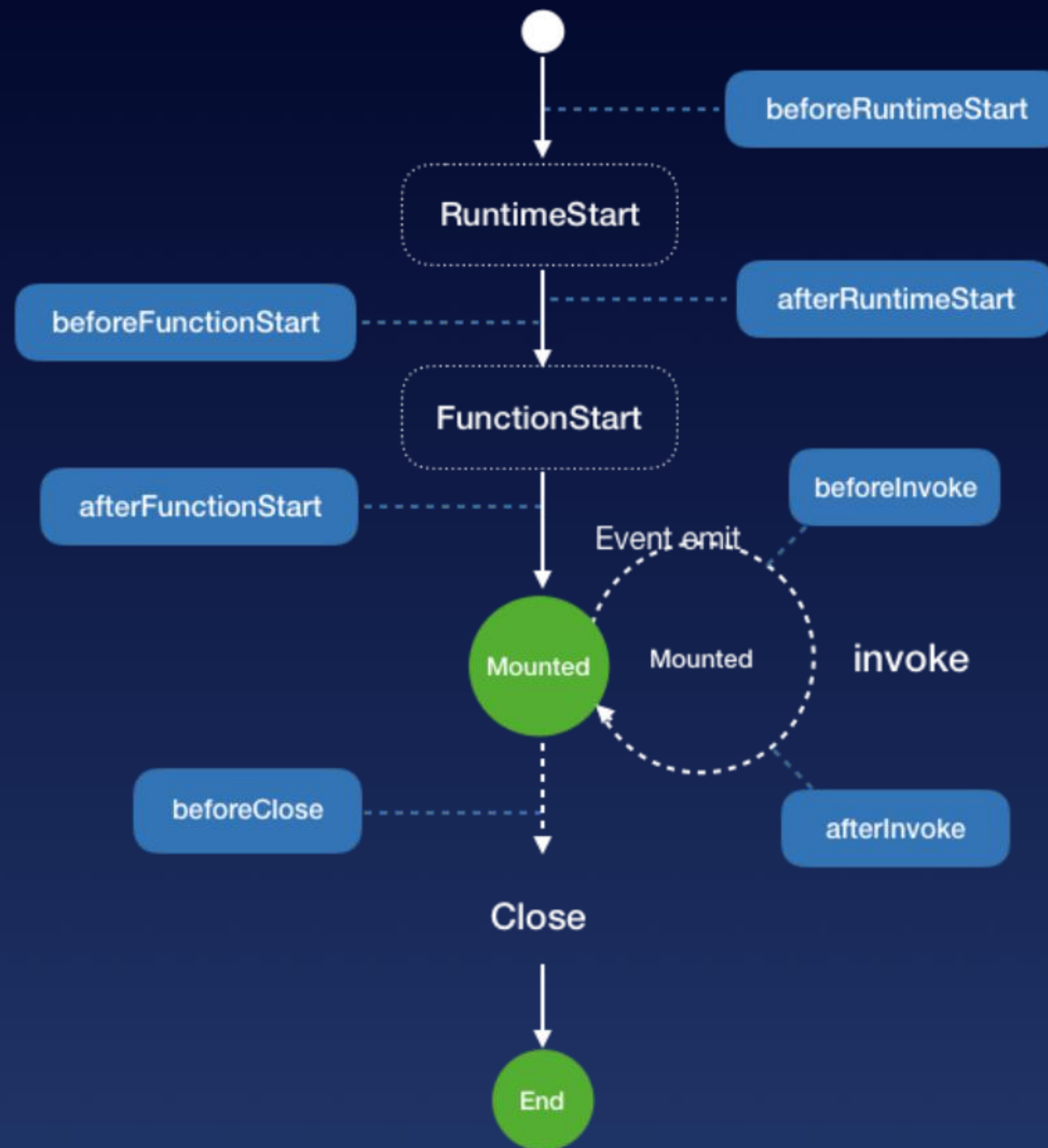
## 微应用例子: Egg As a Service

```
snakey:artist-home snakey$ tree
.
├── app
│   ├── controller
│   │   └── index.js
│   ├── router.js
│   ├── service
│   │   └── my_service.js
│   └── view
│       └── success.htm
├── artist-home.release
├── config
│   ├── config.default.js
│   └── plugin.js
├── f.yml
├── package.json
└── test
    └── fn.test.js
```

EaaS = Web.md + FaaS Infra

# 微应用例子: Egg As a Service

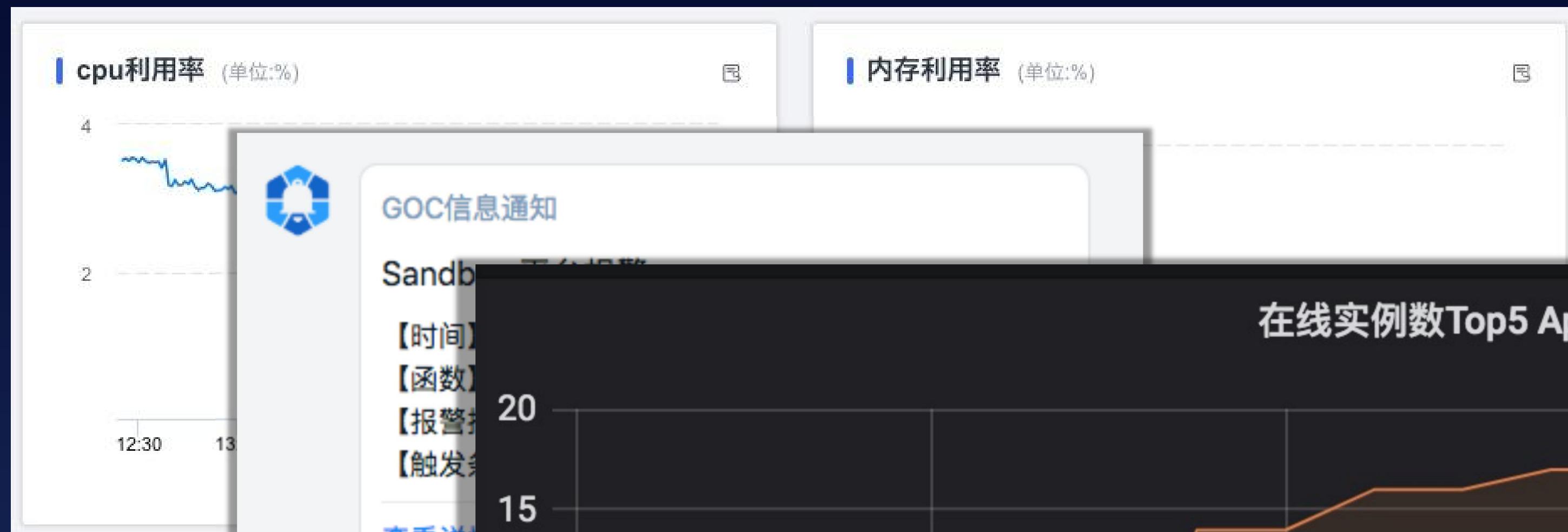
```
snakey:artist-home snakey$ tree
.
├── app
│   ├── controller
│   │   └── index.js
│   ├── router.js
│   ├── service
│   │   └── my_service.js
│   └── view
│       └── success.htm
├── artist-home.release
├── config
│   ├── config.default.js
│   └── plugin.js
├── f.yml
├── package.json
├── test
│   └── fn.test.js
```



# 工程闭环



# 监控体系



**GOC信息通知**

Sandb

【时间】

【函数】

【报警】

【触发】

[查看详情](#)





# 扁鹊：Serverless 时代面向业务的监控

传统监控：关注机器CPU、内存、磁盘...

Serverless 时代的监控：关注业务层异常

**用户**  
什么人

**时间**  
在什么时间

**地点**  
在哪个函数

**行动**  
做了什么事情

**结果**  
结果如何

**背后的原因**  
什么原因导致的

域名: [输入框]    应用: [输入框]    用户名: 用户名    入口 @: [输入框] queryAdditionalInfo.json    搜索

Trace Id: Trace Id    返回码: 返回码    关键字: 例如 (订单号/客户Id...)

2019-12-11 00:00:00 - 2019-12-12 00:00:00    添加日常机器     耗时按降序排列    日常 预发 beta 安全生产 线上

请求时间	应用名	用户名	Host	Entry	返回码	鹰眼 ID	耗时	操作
2019-12-11	[模糊]	[模糊]	[模糊]	[模糊]	200	[模糊]	8 ms	诊断
2019-12-11	[模糊]	[模糊]	[模糊]	[模糊]	200	[模糊]	6 ms	诊断

# 扁鹊：Serverless 时代面向业务的监控

### 链路详情

0 ms  
15:42:27 133

2 ms  
15:42:27 135

方法名: `getUserInfo`

调用参数

参数 1: `"123456"`

3 ms  
15:42:27 136

方法名: `getUserInfo`

展开应用 `xxx` 的调用记录

12 ms  
15:42:27 145

服务名: `xxx`

方法名: `getUserInfo`

返回结果: Object

17 ms  
16:59:29 701

10026 ms  
16:59:39 710

10055 ms  
16:59:39 739

```
nodejs.ResponseTimeoutError: Server no response in 10000ms, address#127.0.0.1:32867  
at Timeout._invokes.set.timer.setTimeout [as _onTimeout] (...)
```

nginx日志告警 机器人

[线上] `xxx` 异常更新

HSF 提供方 `xxx` 线程池满 (HSF 应用: `xxx`)

异常数量: 17 -> 61

影响用户: 12 -> 40

近一小时: 16

查看

参数 1: `"123456"`

# 逻辑编排：陆游平台



The screenshot shows the '浏览器传参' (browser parameters) section of the Lu You platform. It displays a sequence of three processors:

- 处理器0: Function**: A function that takes browser parameters and returns a name and text.

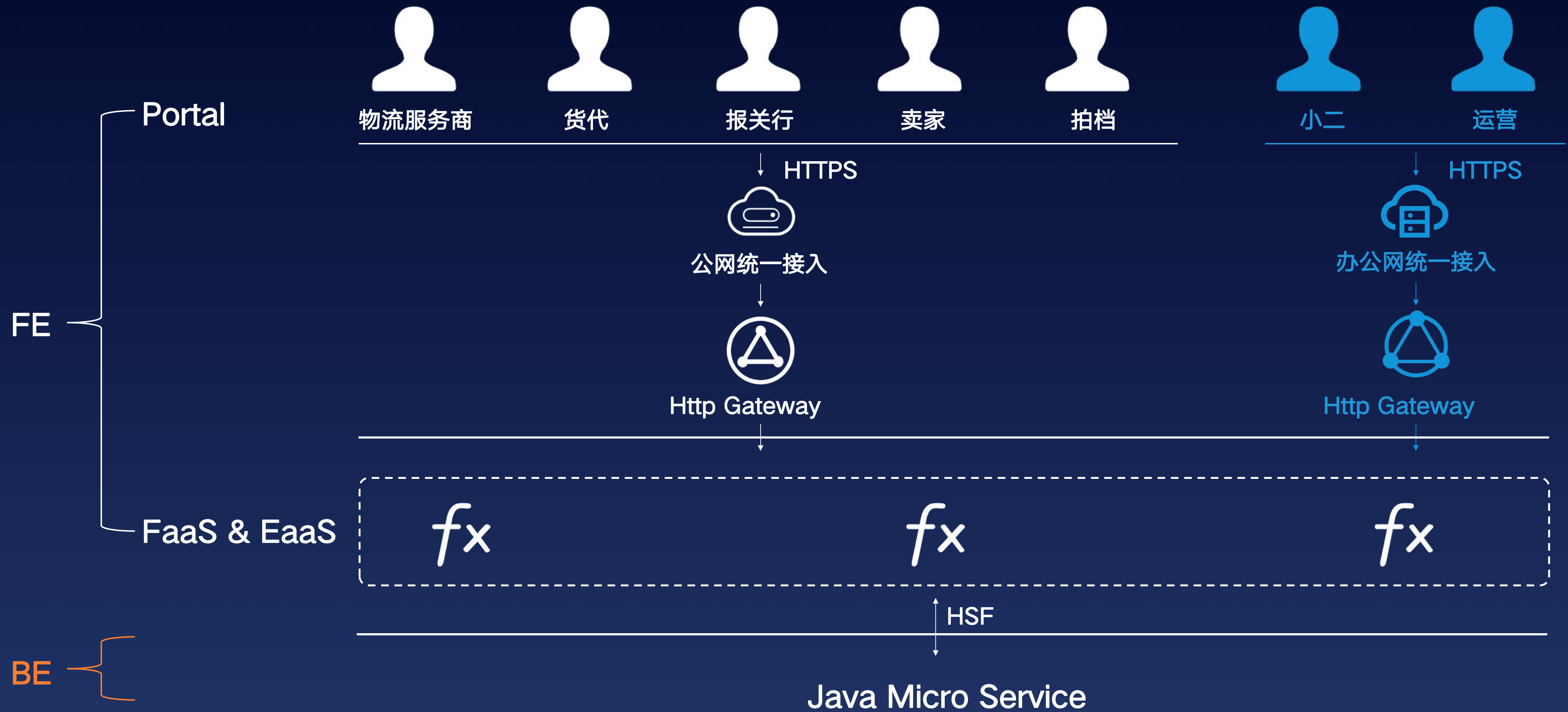
```
function ( $browserParams, $params, $results, $ctx, $util ){  
  const { name = '', text = '' } = $browserParams;  
  return [name, text];  
}
```
- 处理器1: HSF**: An HSF (Service Framework) processor with a method 'format(String, String)'. It includes a search bar and a dropdown menu.
- 处理器2: Function**: A function that takes browser parameters and returns a success code, data, message, and code.

```
function ( $browserParams, $params, $results, $ctx, $util ){  
  const { success, data, message, code } = $params;  
  return {  
    code: success ? 0 : code,  
    data: {  
      message,  
      code,  
      ...data,  
    }  
  }  
}
```

Each processor is connected to the next by a '+' icon and a downward arrow. A vertical label '处理器管理' (processor management) is visible on the right side of the interface.

# 阶段结果&展望

# 黄金时代 — 2018 年至今



## 效果 — 降本提效

研发提效 (数  
据脱敏)

机器成本降低  
(数据脱敏)

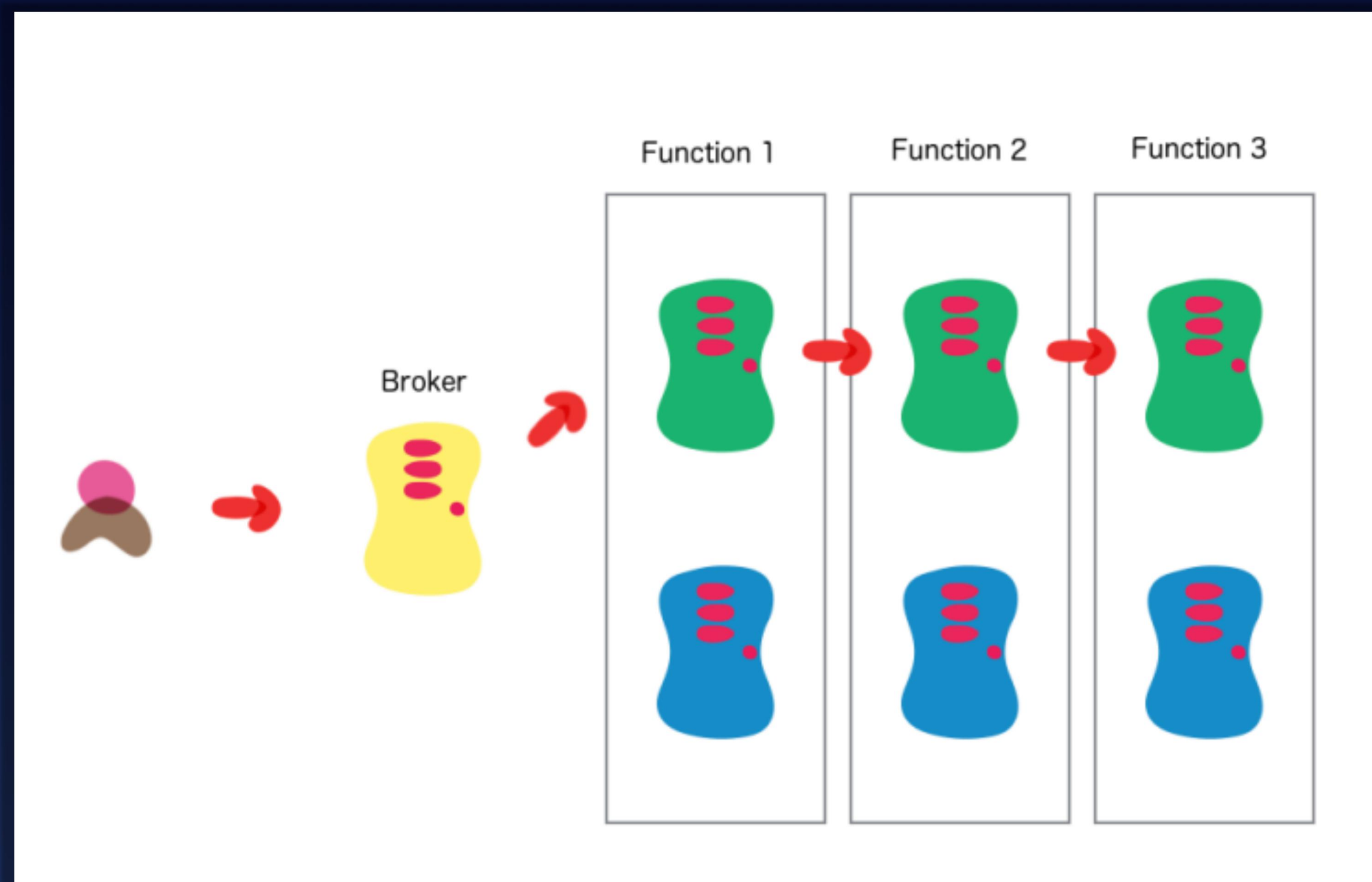
运维成本降低  
(数据脱敏)

# 黄金时代 — 稳定性提升

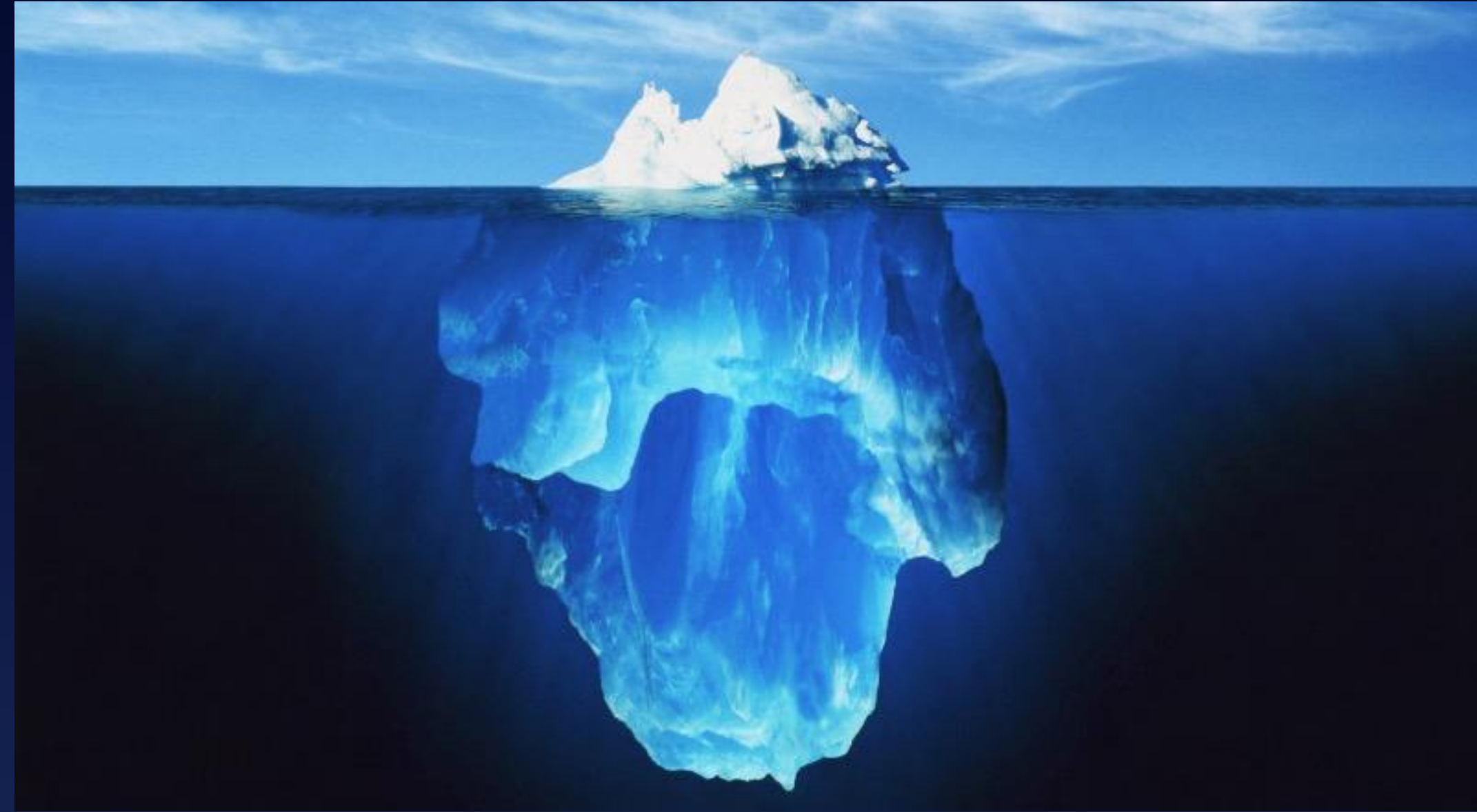
灵活灰度

1 分钟问题发现

秒级回滚



## 再谈 Serverless 价值



“前端价值 = ( 对业务认知 + 对用户理解 ) + ( 前端角色的优势 + 前端技术优势 )”

—— 赵振宇

B2B 业务下，Serverless 将是撬起前端价值的重要支点



# 展望

## 平台完善

- 启动提速
- 扩容提速
- 国际化

## 社区共建

- 开源

## Serverless +

- 技术赋能业务
- Serverless + 其他前端领域

# Midway FaaS 开源



[midwayjs/midway-faaS](https://github.com/midwayjs/midway-faaS)

多云部署

- Avoid Vendor Lock-in

灵活性

- Flexibility

生命周期  
可扩展

- Lifecycle Extension

# 总结

- 1、Serverless 还没有事实标准
- 2、Serverless 有着美好未来
- 3、Serverless 更大的价值在于撬动业务价值和更多创新思维

Nodejs Serverless, 为前端的 自由 而生

THANKS

GMTC  
全球大前端技术大会