

超越边界：FaaS 的应用实践和未来展望

蚂蚁集团 邢奇 2023/07

FaaS 概述



ChatGPT3.5

FaaS (Function-as-a-Service) 是一种云计算服务模型，它允许开发者编写和部署函数，而不需要管理底层基础设施的运行。在 FaaS 中，开发人员只需将他们的代码上传到云平台，云平台会自动运行和扩展这些代码。这样，开发者可以专注于编写应用程序的逻辑而不用担心基础设施细节。FaaS 常用于构建事件驱动的应用程序，如 Webhook，数据处理等。常见的 FaaS 平台包括 AWS Lambda，Azure Functions，Google Cloud Functions 等。

```
1 import json
2
3 def lambda_handler(event, context):
4     # 处理事件数据
5     name = event['name']
6     message = 'Hello, ' + name + '!'
7
8     # 构造响应数据
9     response = {
10         'statusCode': 200,
11         'body': json.dumps(message)
12     }
13
14     return response
```

上述代码实现了一个简单的 Lambda 函数，接收一个 JSON 格式的事件数据，从中提取出 name 字段，然后返回一个包含欢迎消息的 JSON 格式响应数据。

FaaS 的崛起

01

应用研发模式面临的困境

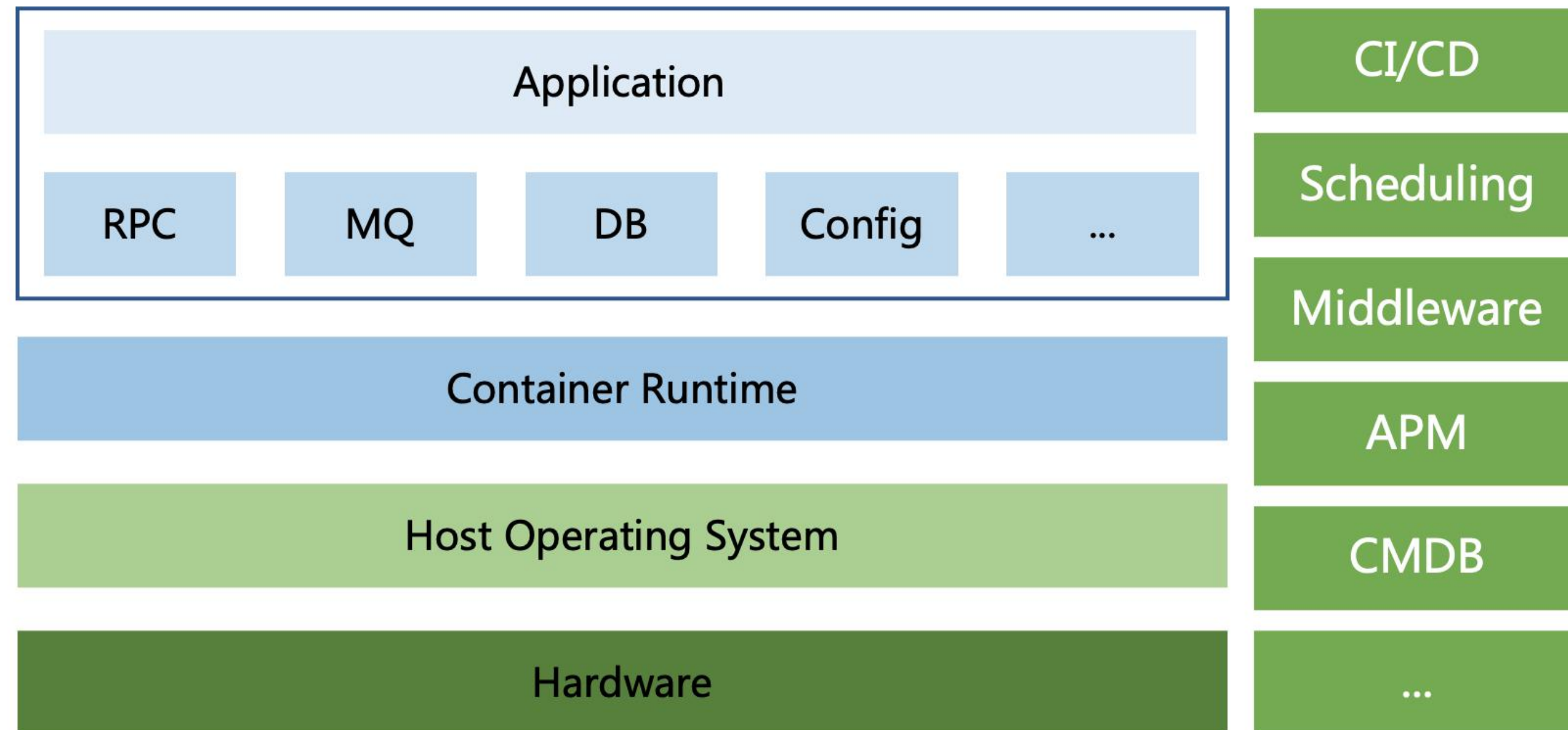
02

FaaS 的适用场景和价值

03

FaaS 落地面临的问题和挑战

应用研发模式面临的困境



研发态负担

- 中间件 SDK 升级
- 多云差异感知
- ...

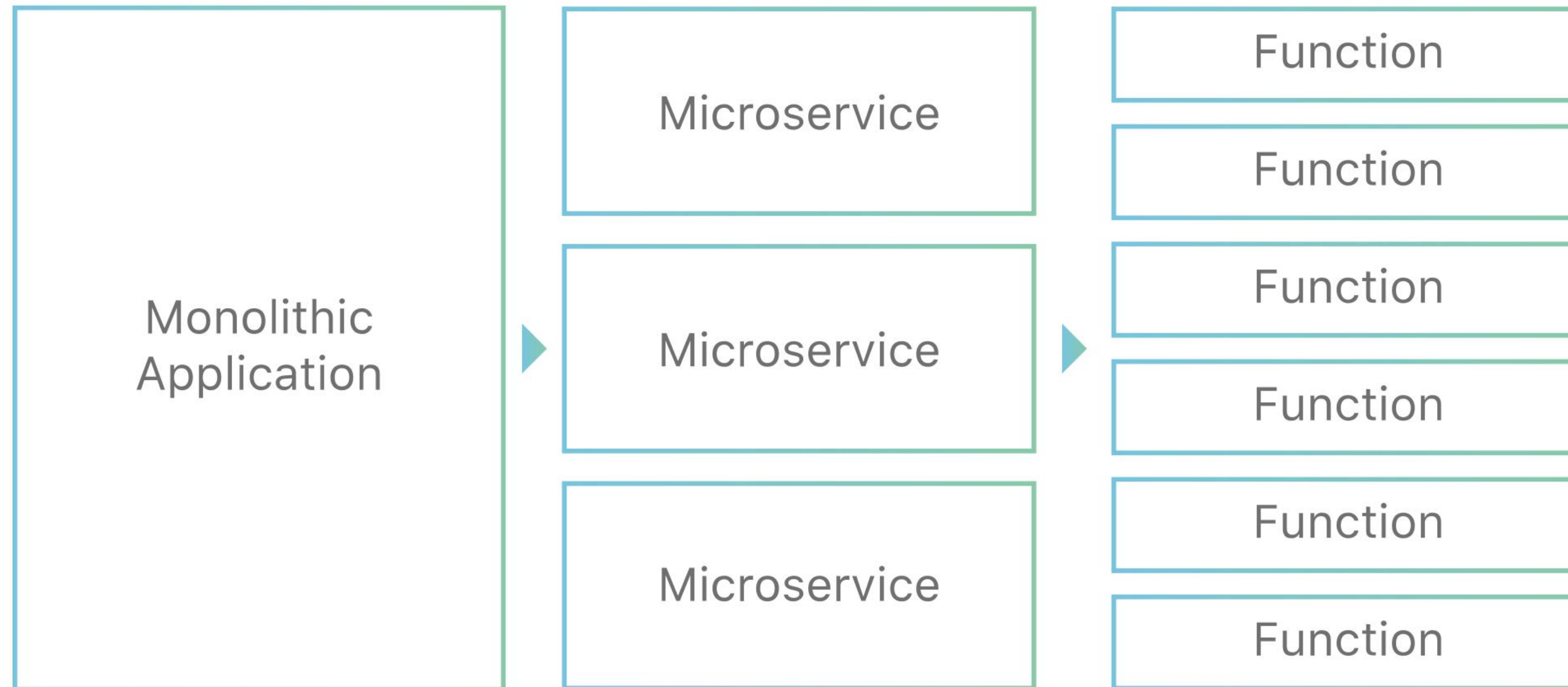


运维态负担

- 发布流程
- 资源申请
- 容量评估
- ...

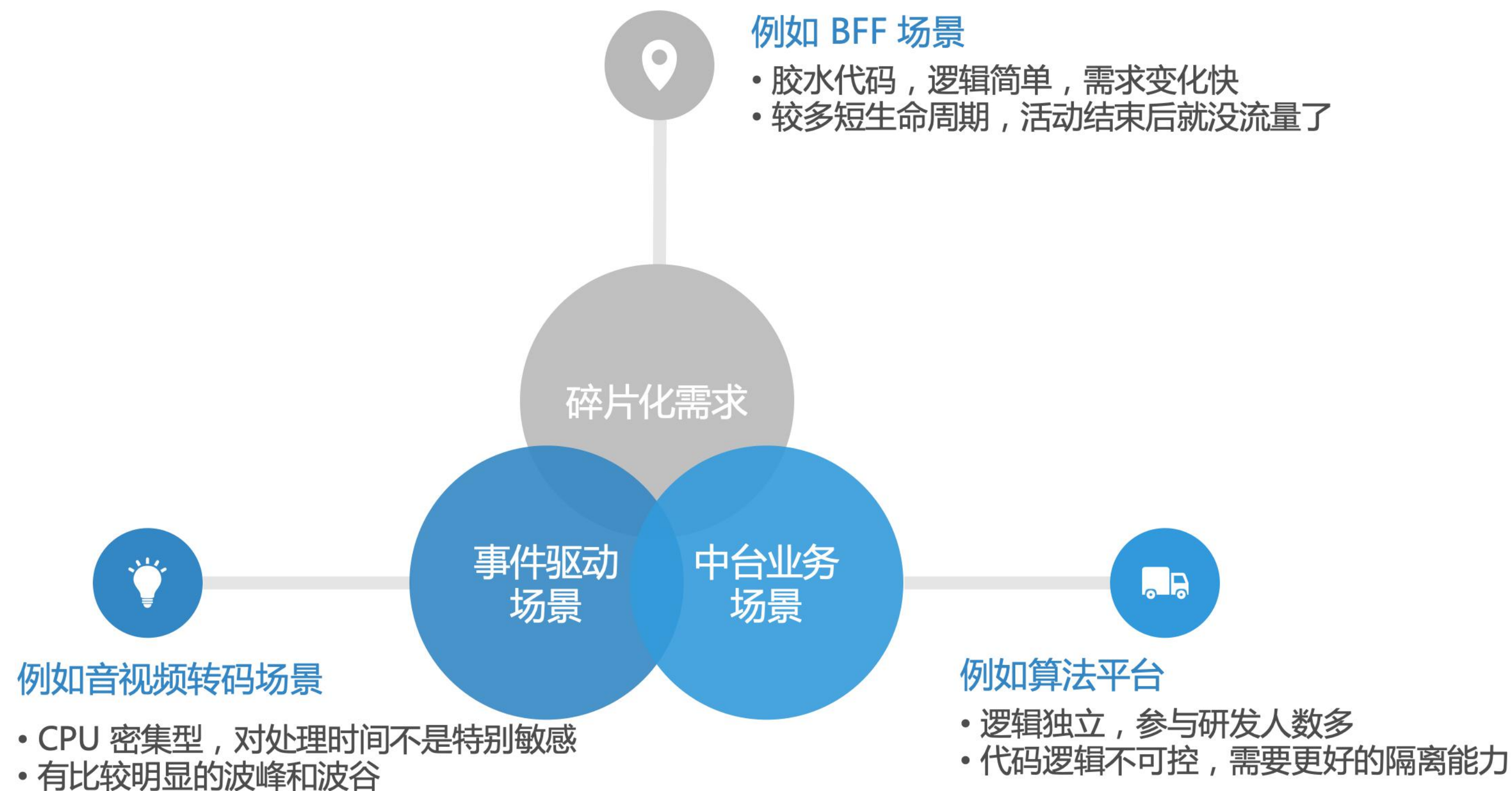
代码耦合严重，复杂度高；运维流程繁琐，效率低；资源利用率低，存在浪费

FaaS 研发模式的优势



- 提升研发效能：快写快发
- 提高运维效率：免运维
- 降低成本和资源消耗

FaaS 的适用场景和价值



让研发者更专注于业务

适合场景

- 胶水代码
- 短生命周期
- 事件驱动
- 波峰波谷明显
- 多语言
- 高隔离性

适合业务

- H5/小程序
- 聚合服务/BFF
- 推荐/营销平台
- 音/视频转码平台
- 算法/ AI

FaaS 落地面临的问题和挑战

性能

- 函数调用的性能问题
- 弹性扩缩容反应时间慢
- 容器启动速度慢

安全

- 资源如何隔离
- 容器逃逸如何防御
- 免鉴权怎么支持

成本

- 资源利用率如何提升
- 用户成本如何降低

体验

- 函数研发体验优化
- 函数如何运维

深入蚂蚁 FaaS 技术架构

01

蚂蚁FaaS技术实践说明

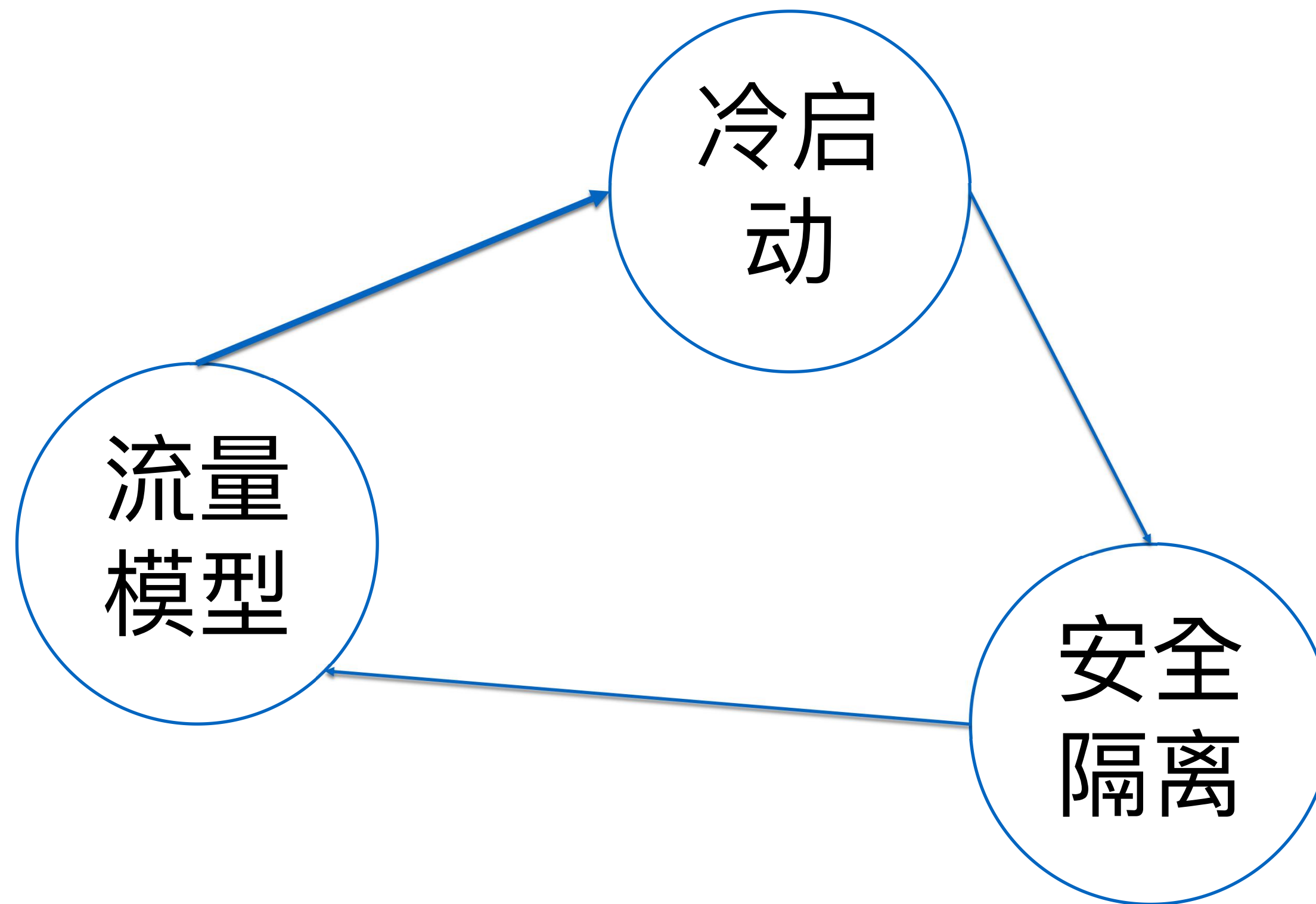
02

蚂蚁FaaS高性能实践详解

03

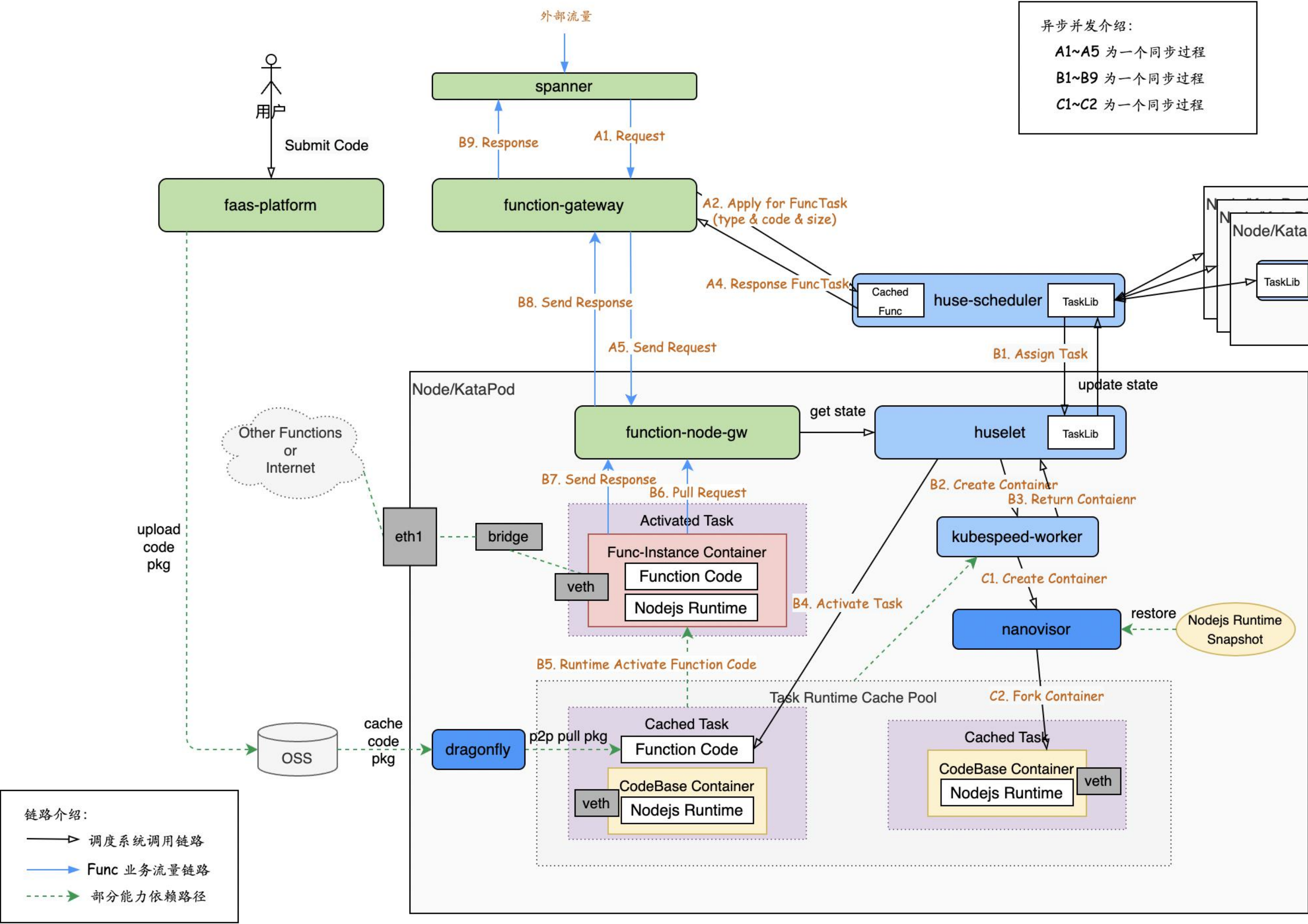
蚂蚁FaaS安全能力建设

蚂蚁 FaaS 技术实践原则



one request per instance

蚂蚁 FaaS 技术实践说明



函数网关

负责函数请求的转发和控制，并且为每个请求发起一次容器调度任务

容器调度引擎（huse）

负责容器的调度，对容器的生命周期进行管理，并且可以控制函数的并发度和复用等状态。并且对函数pod资源池进行管理

函数Pod资源池

函数容器运行的环境，一个集群有N个Pod资源

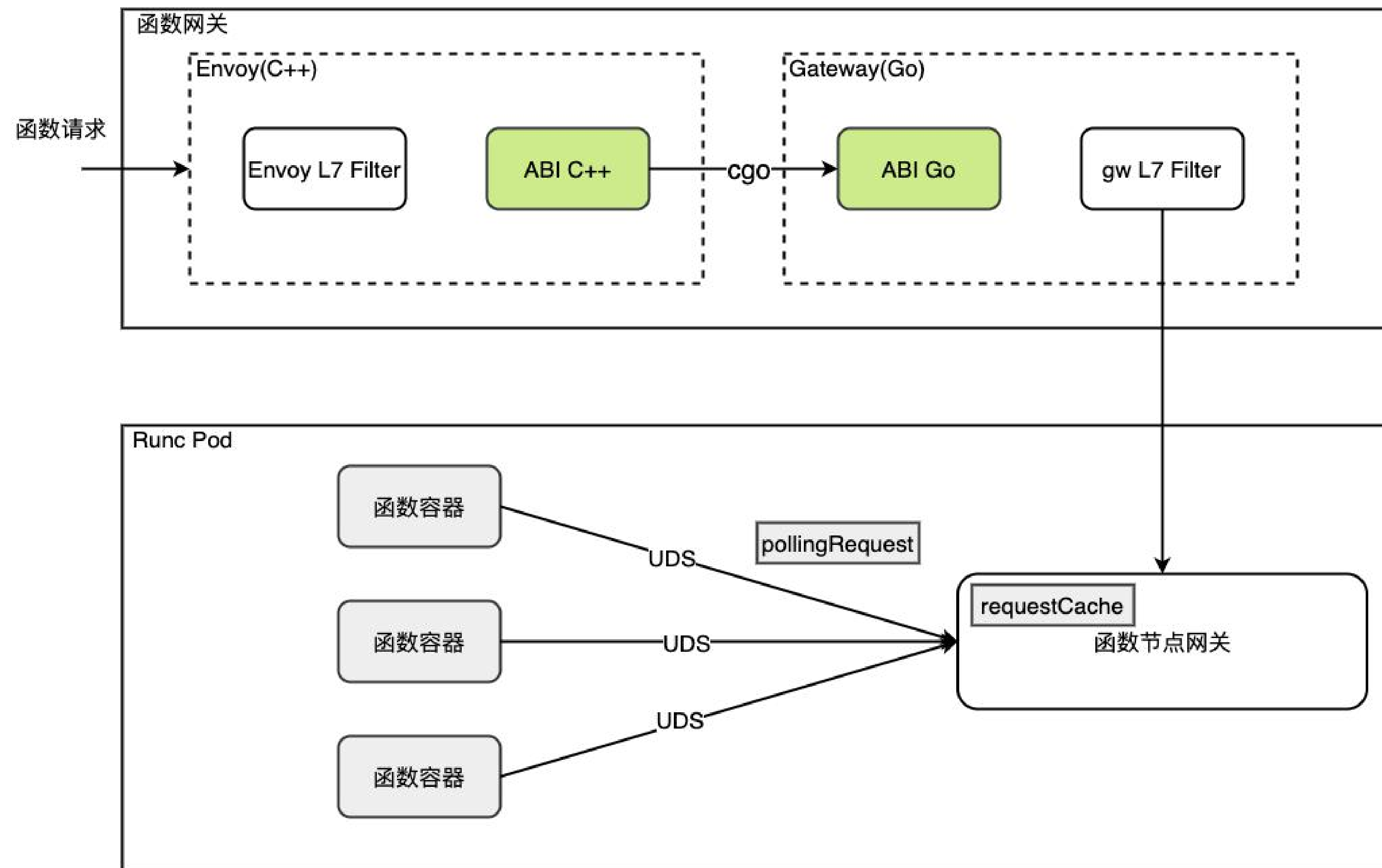
容器运行时

负责实现OCI标准，快速启动函数容器，对容器的runtime进行有效控制

函数容器

用户的函数代码运行在函数容器中，仅作为客户端拉取请求并处理，返回结果

蚂蚁 FaaS 性能优化实践 函数网关



函数网关

负责统一接收函数请求，进行认证授权，并且发起容器调度，最后将请求转发给函数节点网关

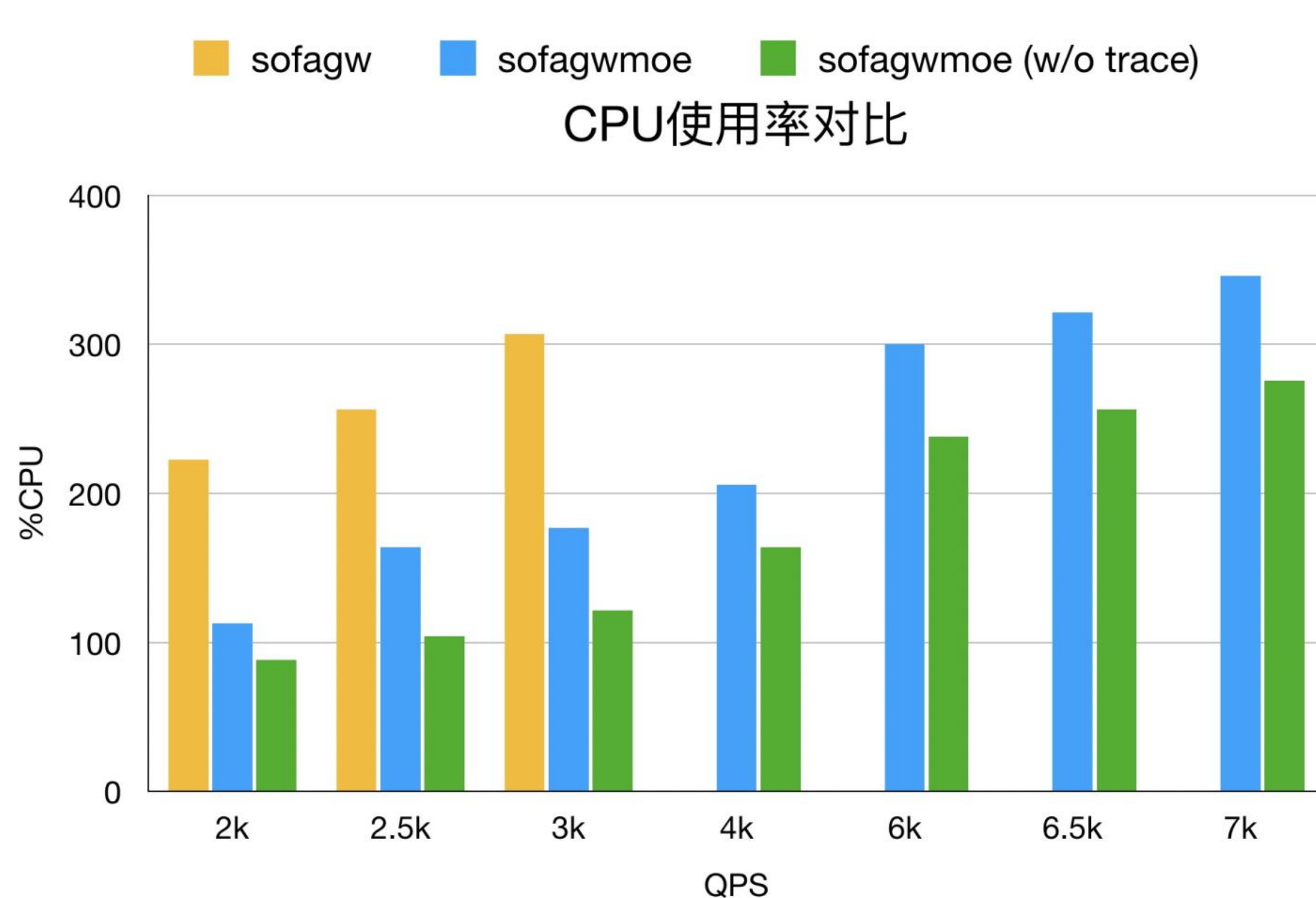
函数节点网关

接收并缓存函数请求，并且将请求进行实例级别的分发，同时等待函数容器启动并开始处理请求

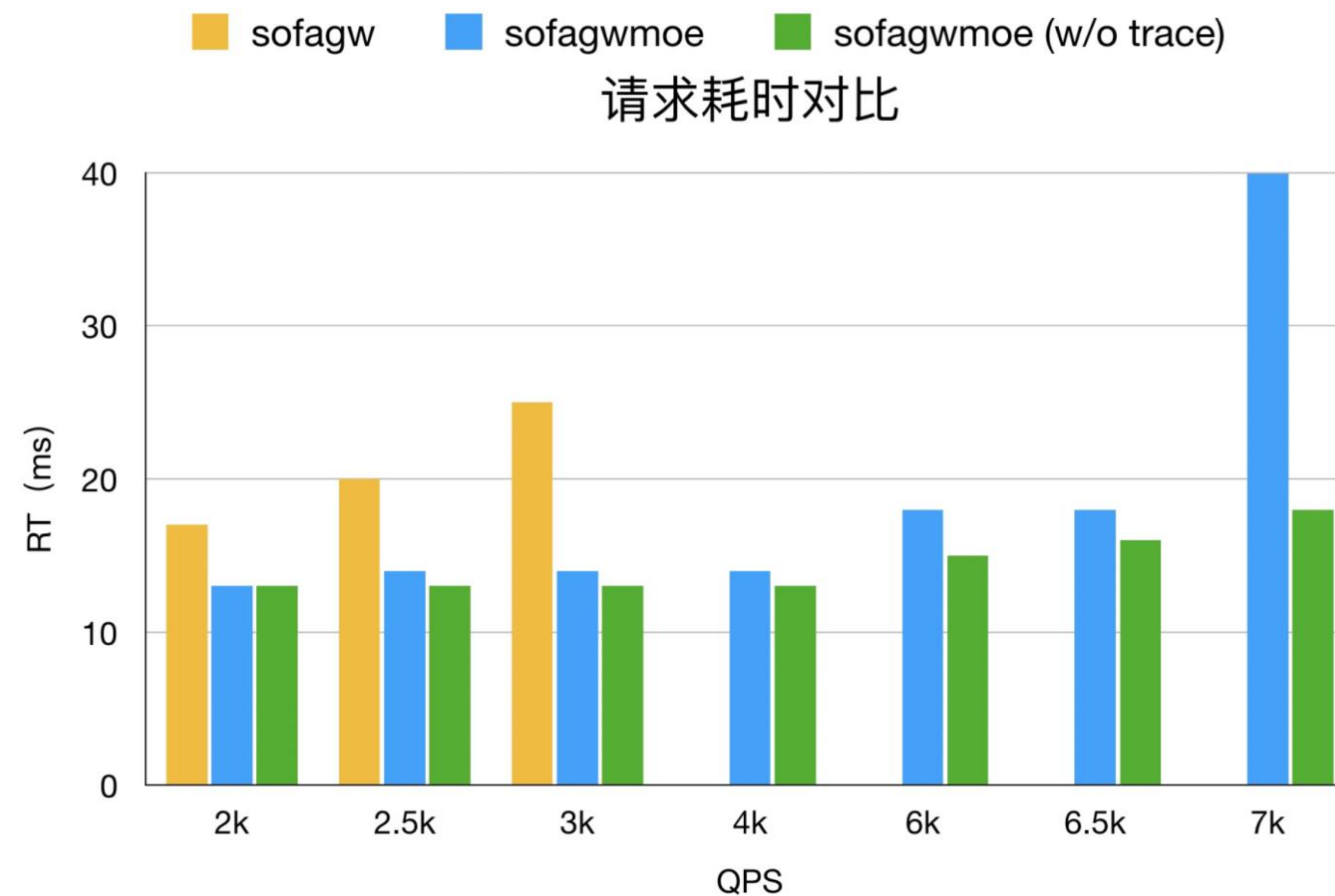
函数容器

函数代码运行的容器，启动之后会作为客户端拉取请求并进行业务处理

蚂蚁 FaaS 性能优化实践 函数网关



CPU资源消耗下降了50%以上



请求耗时下降了30%左右

蚂蚁 FaaS 性能优化实践 容器调度引擎



HUSE调度引擎，即Hyper-elastic Unified Serverless Engine，**蚂蚁容器调度引擎的下一代架构**

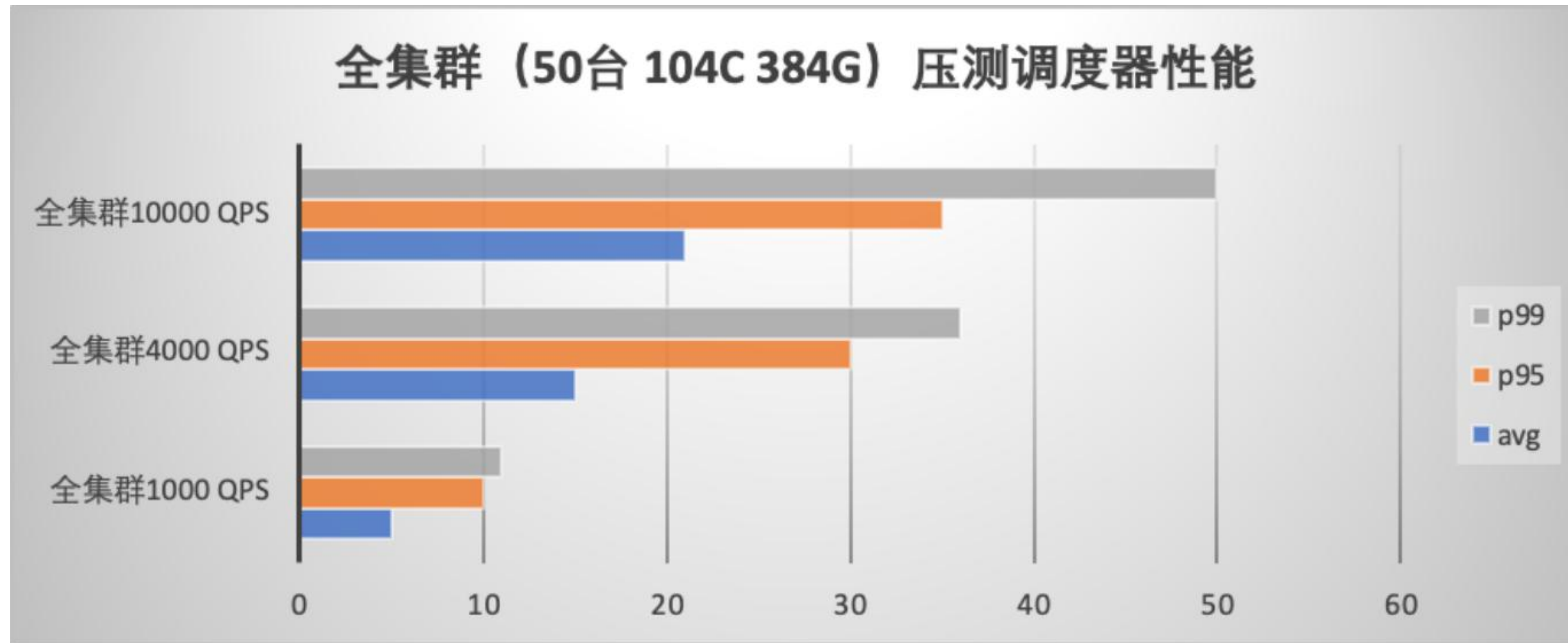
特性

蚂蚁容器调度引擎的下一代架构，是专门面向高吞吐低延迟、低成本、极速启动的Serverless场景而设计的。

场景

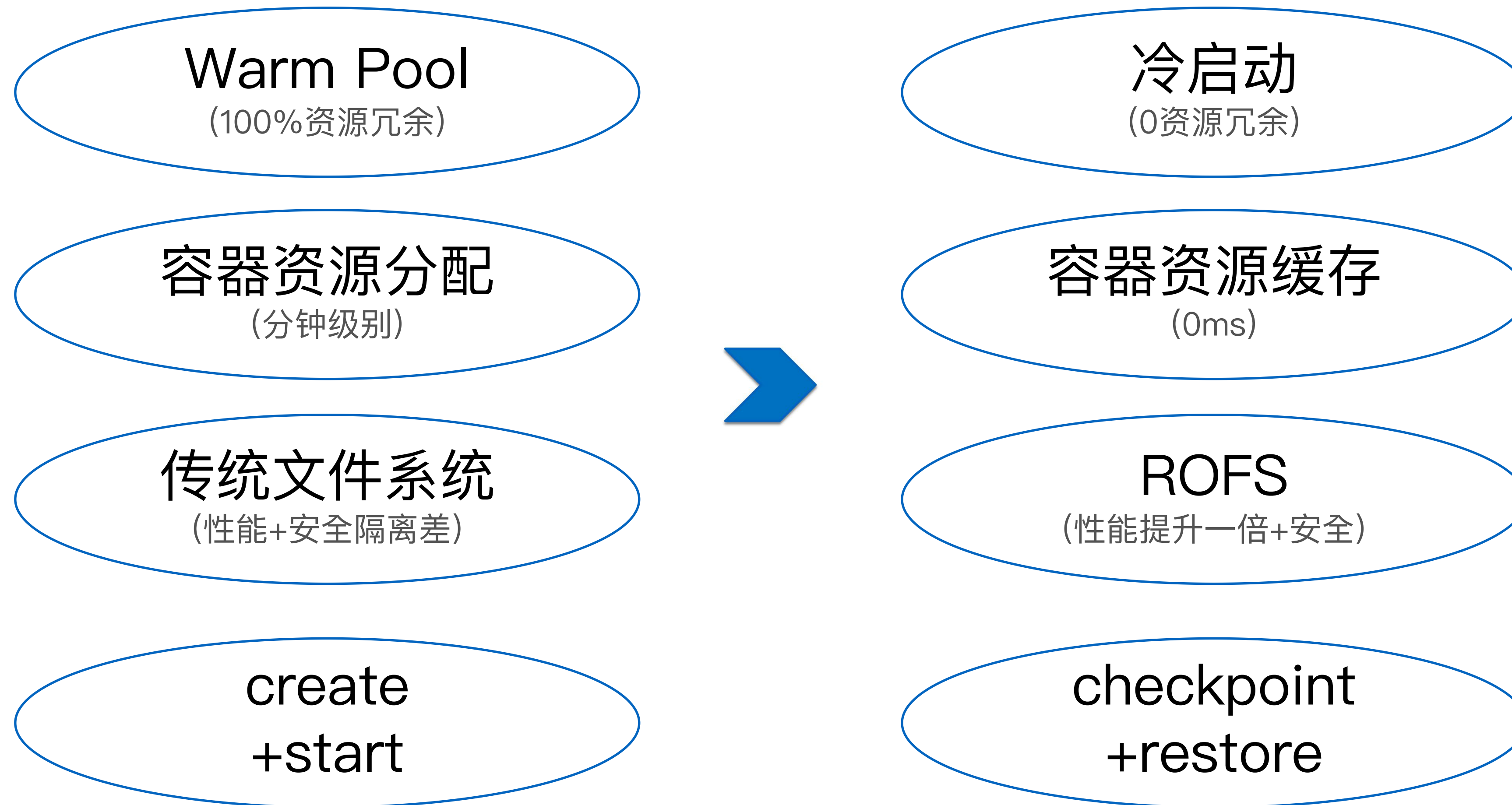
目前HUSE不仅在FaaS场景进行实践，同时也应用在batch job等场景，是一个通用的serverless任务调度引擎。

蚂蚁 FaaS 性能优化实践 容器调度引擎

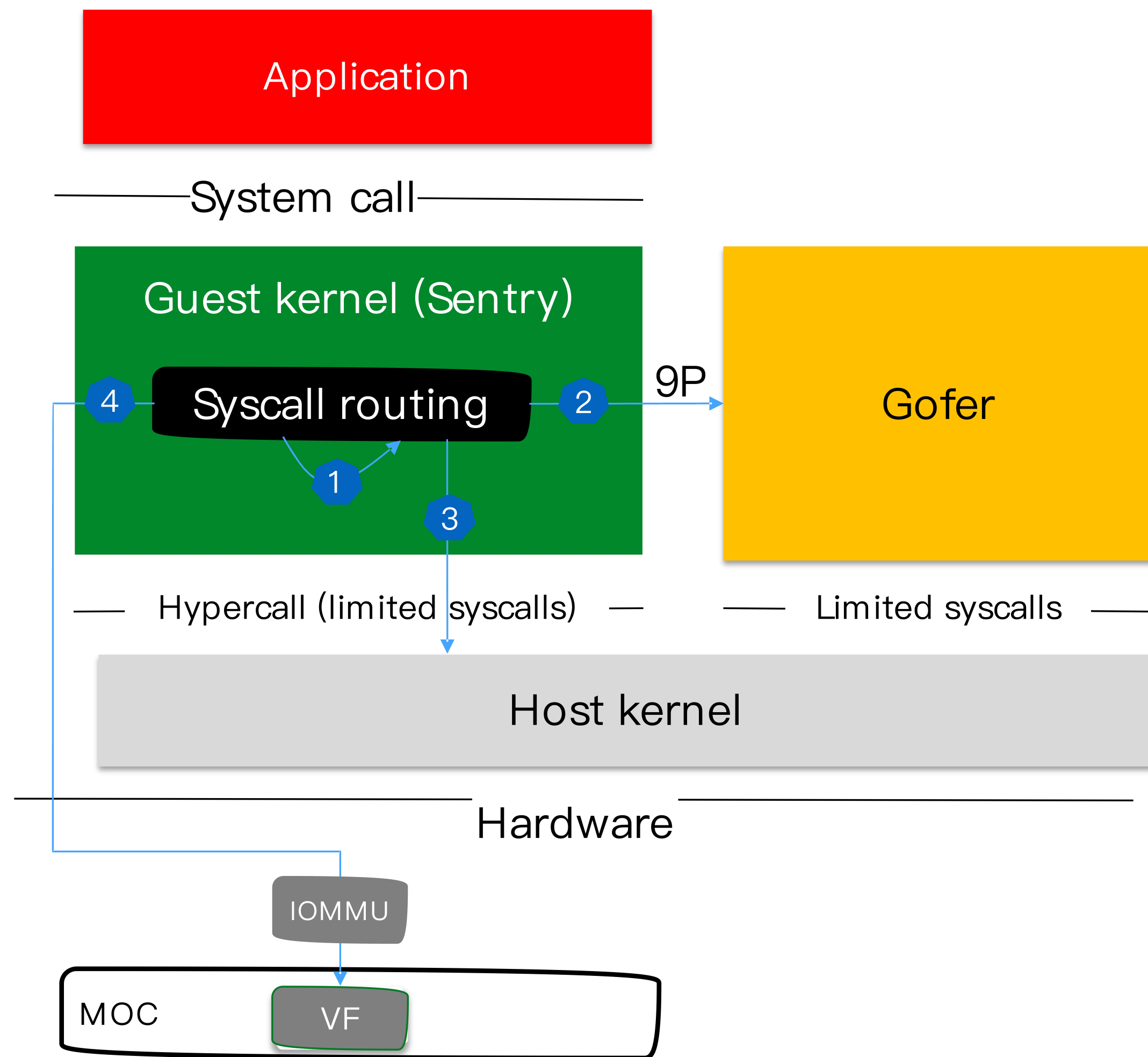


全集群10000 QPS调度吞吐下，HUSE可以实现平均21ms的容器调度延迟

蚂蚁 FaaS 性能优化实践 容器冷启动



蚂蚁 FaaS 性能优化实践 容器冷启动



NanoVisor

轻量级Hypervisor，进行syscall interception和host syscall加速

多平台支持

ARM（鲲鹏）、Hygon/AMD

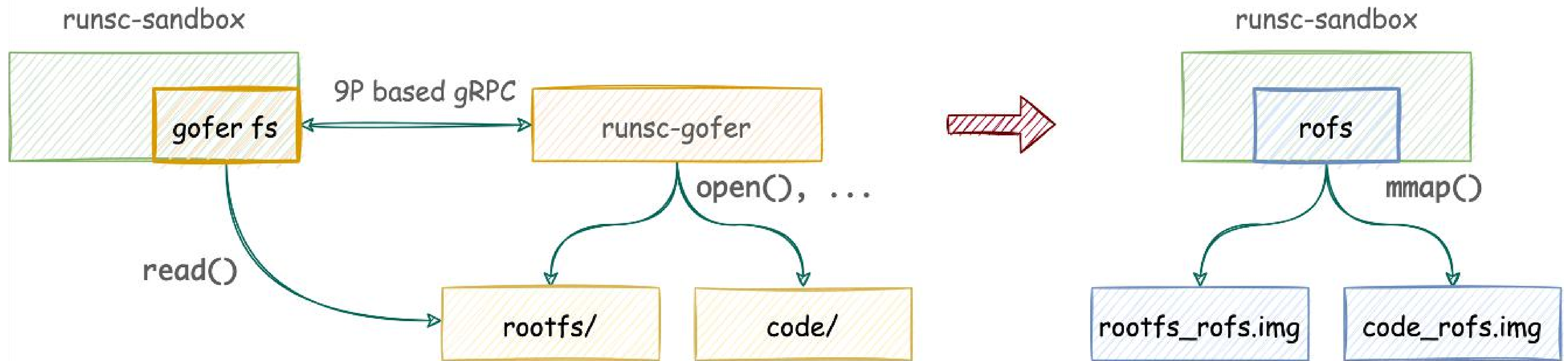
性能

火焰图热点分析、Go runtime增强、引入高性能用户态协议栈（TLDK/DPDK，引入CGO）

应用场景

FaaS、增强基础设施安全、增强容器平台安全、快速扩缩容

蚂蚁 FaaS 性能优化实践 容器冷启动

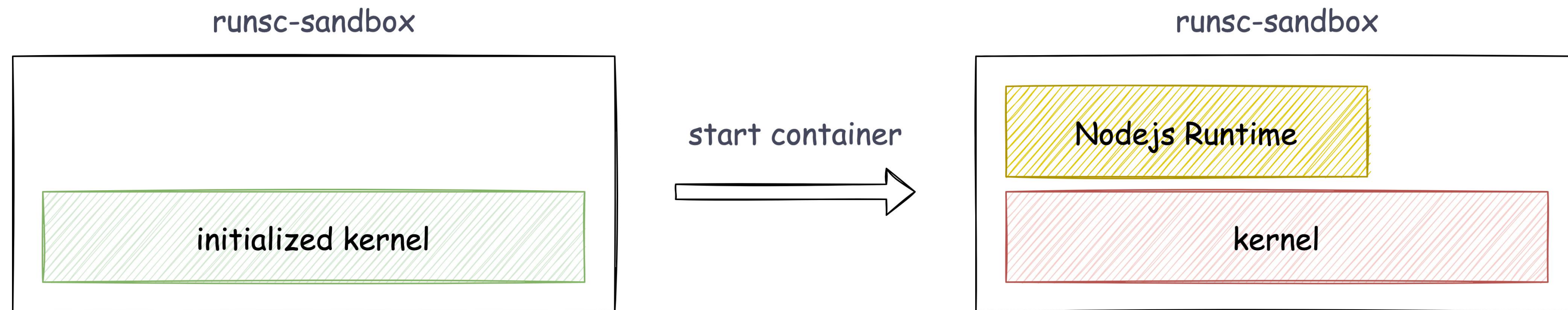


Read Only File System (ROFS)

容器镜像文件不再需要解压为一个 host 上的目录并 bindmount 到容器内，而是使用 ROFS 可以解析的格式。

容器创建的时候，镜像文件在 `runsc-sandbox` 进程之外打开，并在 `sandbox` 进程内直接 `mmap`，对镜像中文件的所有操作直接变成内存操作，而不需要调用系统调用操作 host 上的文件。

蚂蚁 FaaS 性能优化实践 容器冷启动

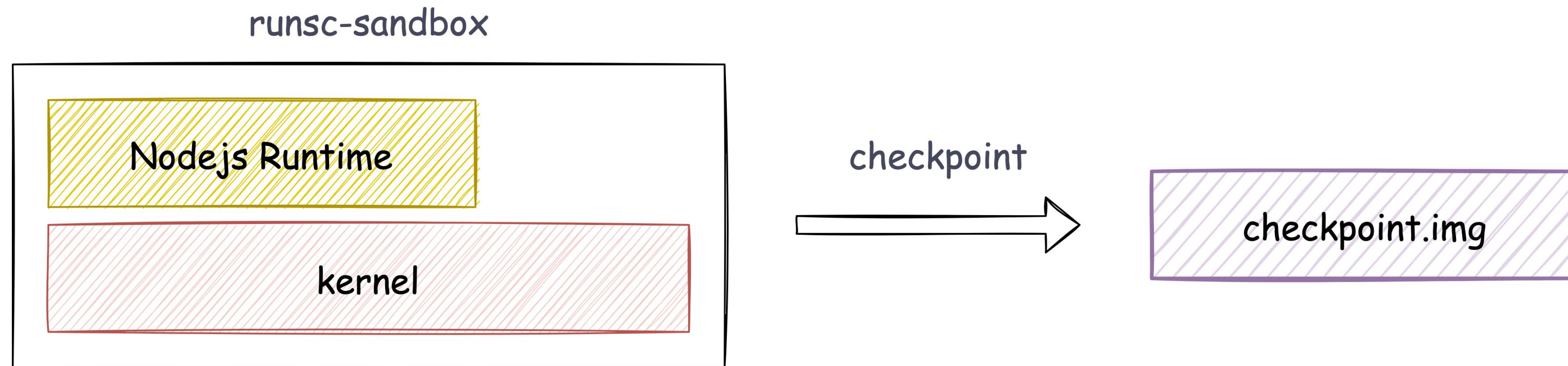


Create+Start容器

符合 OCI 标准的容器运行时包含 create 和 start 两个容器启动相关的接口

create 是根据容器镜像和配置文件创建容器运行的环境，对应 NanoVisor 容器沙箱的创建和应用程序内核的初始化；start 则是启动容器，对应 NanoVisor 容器沙箱内一号进程（Nodejs Runtime）的创建和启动。

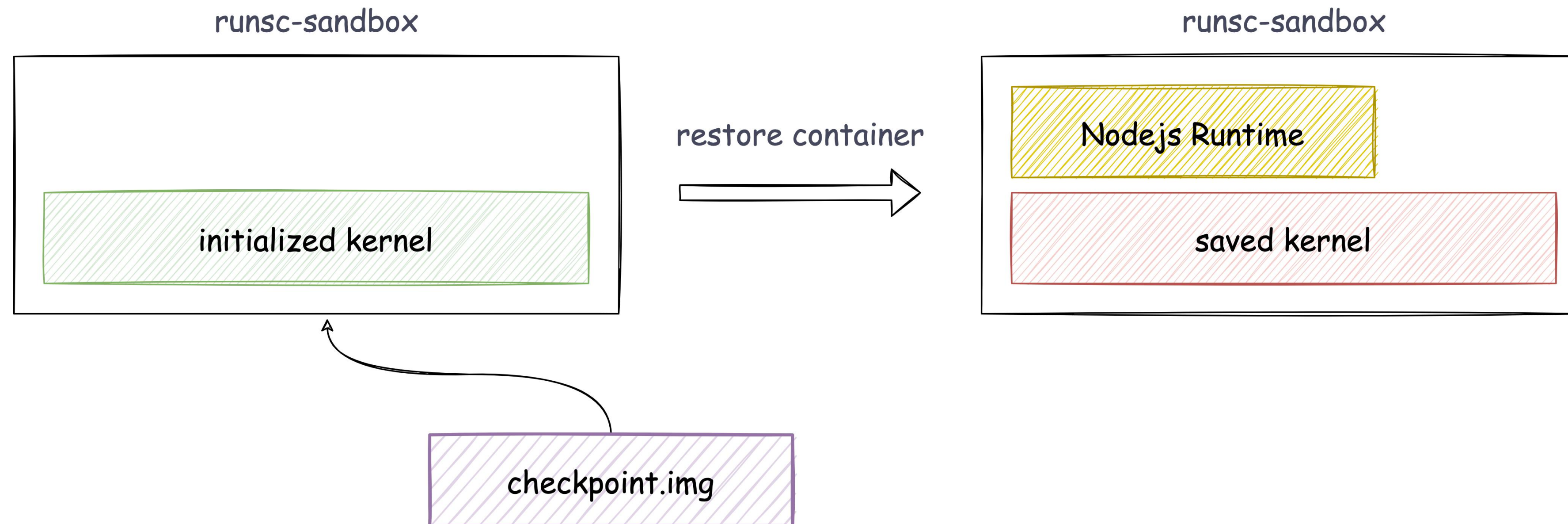
蚂蚁 FaaS 性能优化实践 容器冷启动



制作checkpoint种子

首先按照传统方式创建、启动一个容器，等待容器内的 Nodejs Runtime 初始化完成之后，使用 checkpoint 技术对 Nodejs Runtime 进程和应用程序内核 sentry 进行状态、数据的保存。

蚂蚁 FaaS 性能优化实践 容器冷启动



Restore容器

在容器启动阶段，我们使用 restore 来进行容器启动，直接利用之前保存好的进程、内核状态和数据进行恢复，不再需要重新初始化 Nodejs Runtime。

在恢复完成之后，Nodejs Runtime 就可以立刻进行业务逻辑处理。

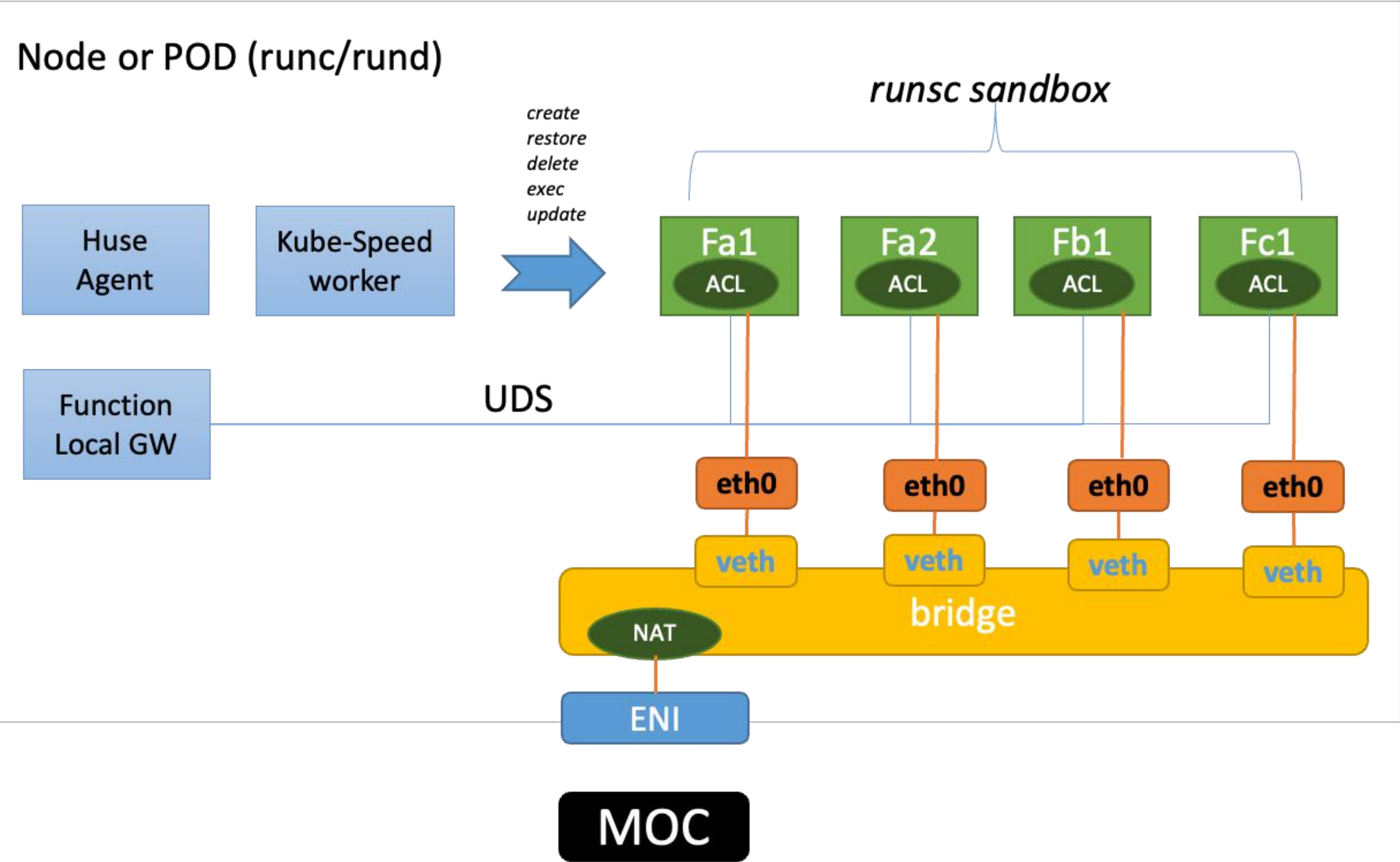
蚂蚁 FaaS 性能优化实践 容器冷启动

FaaS函数容器落地开销

- 启动速度 < 90ms
- 内存开销 < 1MB

蚂蚁对第三代容器运行时的期待：安全容器发于安全，而归于性能成本——10x

蚂蚁 FaaS 安全建设 容器和网络架构



Runsc (NanoVisor)

负责实现函数生命周期的管理，快速启动函数容器

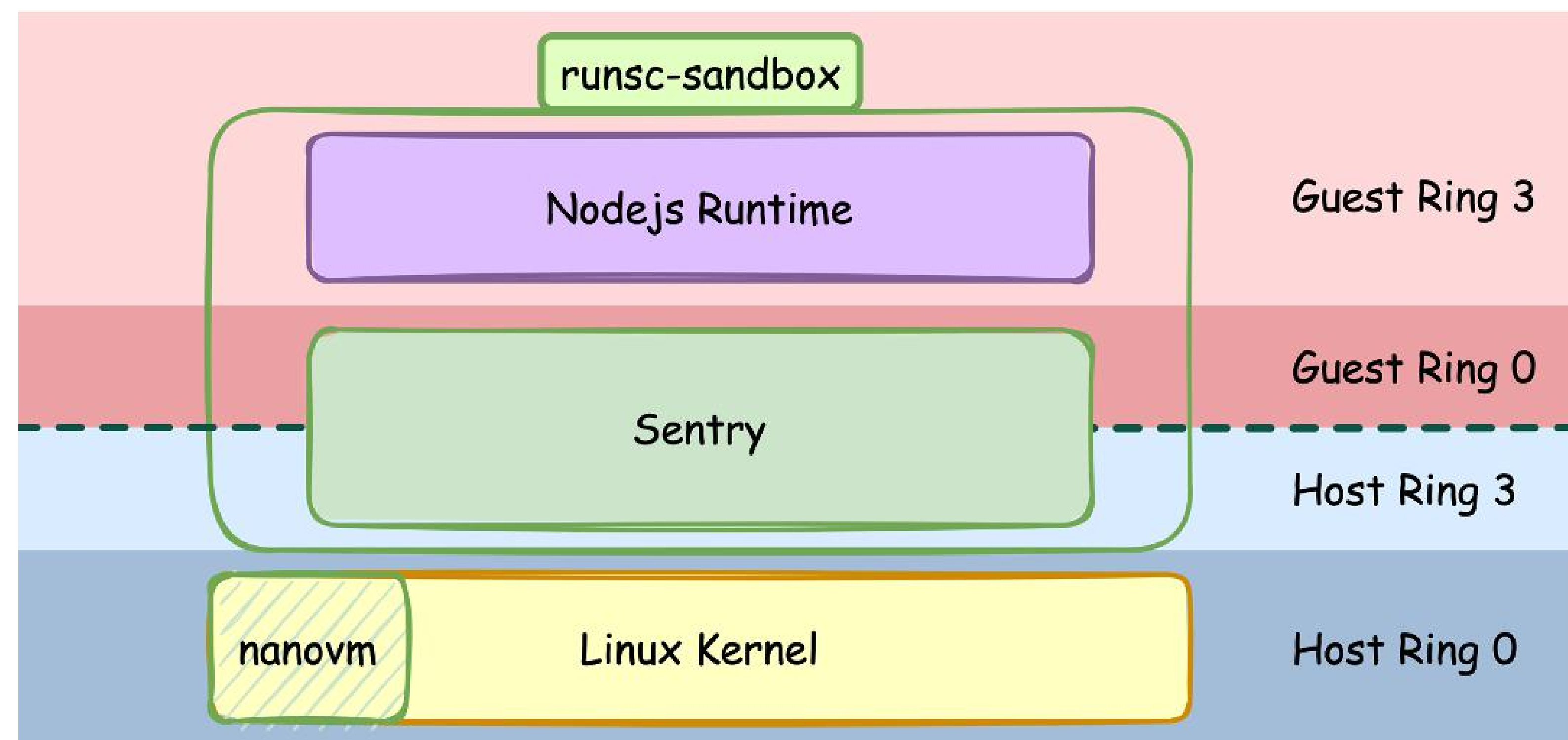
函数容器 (runsc sandbox)

函数容器是一个完全隔离的runsc沙箱环境，配置有ACL规则和虚拟的veth pair网卡

Bridge+eBPF

函数容器的veth pair挂在bridge上，通过eBPF进行高效网络过滤、控制和转发

蚂蚁 FaaS 安全建设 容器纵向安全防逃逸



函数程序

运行在虚拟化的 guest 态，它的系统调用会被 sentry（运行在 guest kernel 态和 host kernel 态）处理和响应。

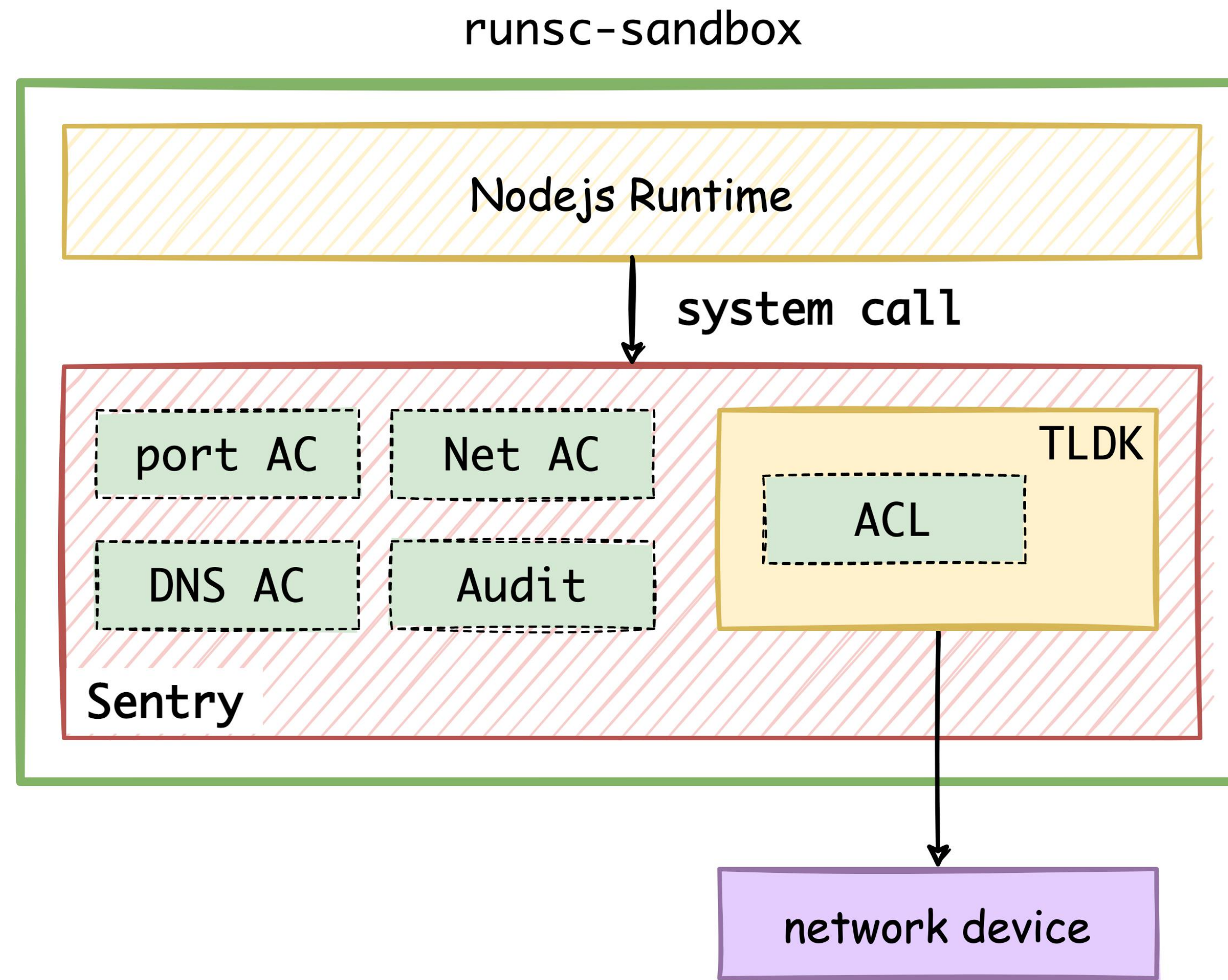
NanoVisor

提供了一个 Go 语言编写的应用程序内核 sentry，它会处理所有函数实例的系统调用。sentry 本身的系统调用由 seccomp 进行限制。

nanovm虚拟化

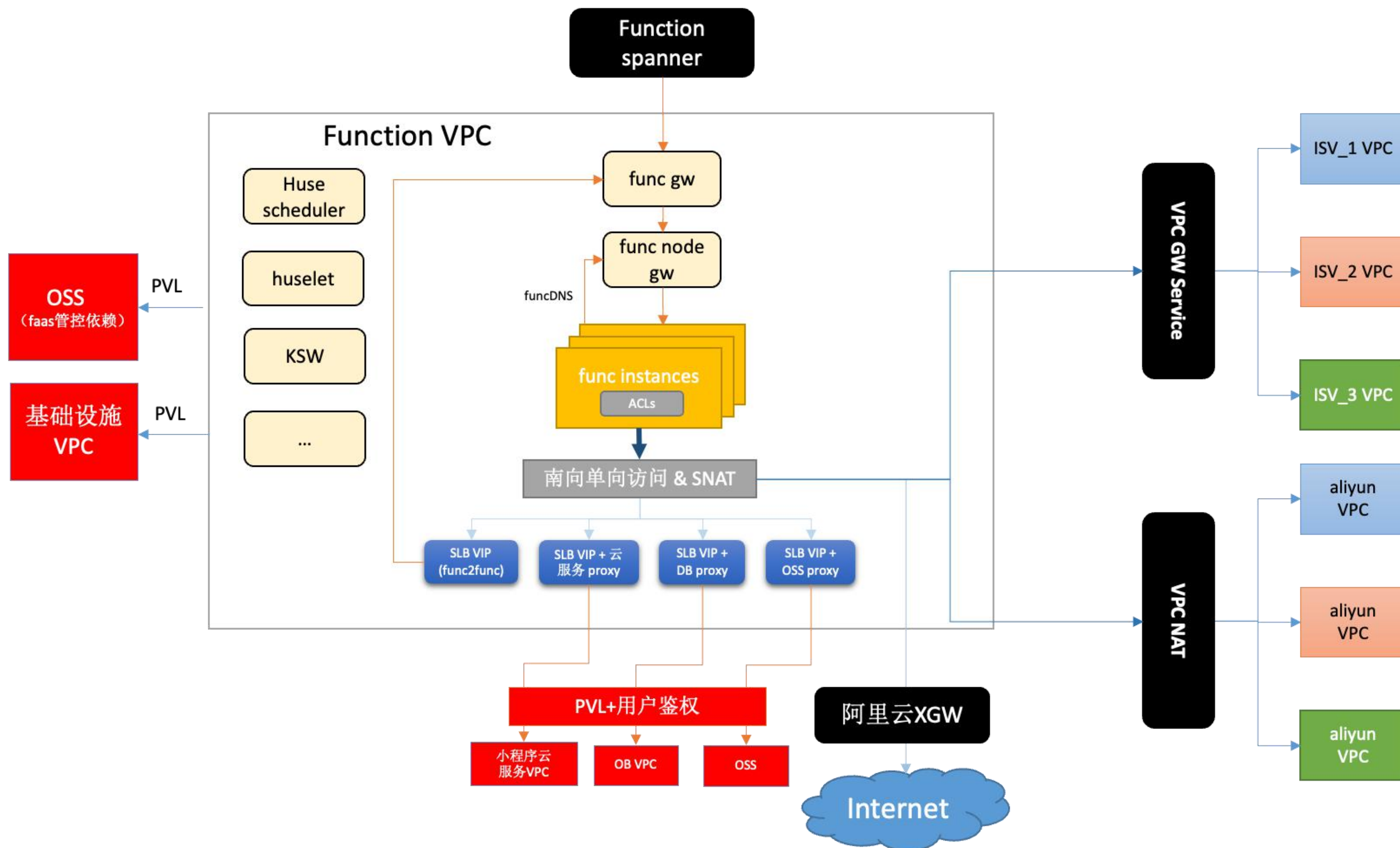
轻量级的 vmm 管理，它是 host 上的一个内核模块。

蚂蚁 FaaS 安全建设 容器横向安全



- 网络行为审计（网络相关的操作，如 connect, accept, recvmsg 等）
- DNS 请求审计、拦截
- 端口监听管控
- 基于五元组的信息的四层网络管控

蚂蚁 FaaS 安全建设 免鉴权访问



- 代理服务访问
(func2func、DB、OSS、云服务等)
- 公网访问
- 用户租户VPC访问
- 其它出口网络

蚂蚁 FaaS 的体验优化

01

研发体验

02

运维体验

蚂蚁 FaaS 的体验优化 研发效能

创建
函数

发布
函数

测试
函数

新建函数

* 函数名称:

支持大小写字母、数字、-和_，但必须以字母开头、以字母和数字结尾，最多 50 个字符

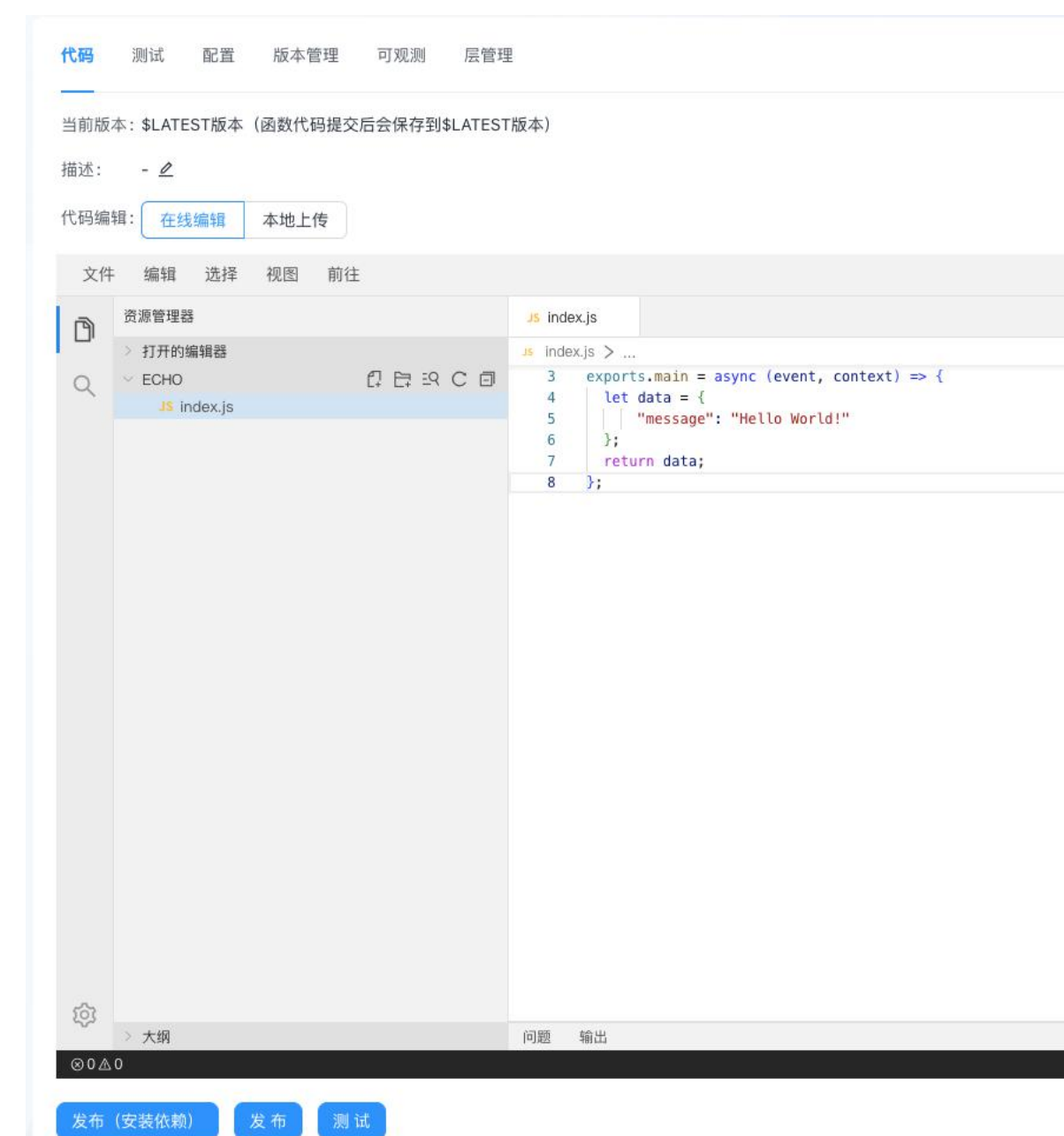
* 运行环境:

* 函数内存:

描述:

模板类型:

- 普通函数
- 访问 DB
- 访问 OSS
- 函数调用函数
- 云调用
- 访问 Redis
- 访问 MySQL



手动测试

测试结果: 成功

返回结果:

```
{
  "root": {
    "item": {
      "message": "Hello World!"
    }
  }
}
```

FaaS 研发体验：代码快写快发，10S内完成一个函数上线

蚂蚁 FaaS 的体验优化 研发效能

开发者营地



从 0 到 1 轻松创建支付宝小程序

开发者营地互动体验环节，支付宝开放平台带来了《从 0 到 1 轻松创建支付宝小程序》实战活动，快速吸引了开发者聚集围观，排队报名体验支付宝云开发云函数开发、云函数秒级部署、以及云函数调用，通过支付宝小程序云开发的产品降低了参与门槛，现场有不少家长带着小学生现场体验，其中一个 6 年级的小朋友在 5 分钟内快速部署完成，也有开发者现场带女朋友体验，用代码表白 `let data = { "message": "Dear , have nice day" };`

6年级小朋友5分钟现场完成支付宝小程序+FaaS云函数研发

蚂蚁 FaaS 的体验优化 运维效能

traceID	状态	执行时间	耗时	操作
0a4b4bb016883916841803921346	成功	2023-07-03 21:41:24	141ms	执行日志
0a4b4baf16883916698596921346	成功	2023-07-03 21:41:09	7ms	执行日志
0a4b4bae16883916574713901382	成功	2023-07-03 21:40:57	144ms	执行日志
暂无更多数据				

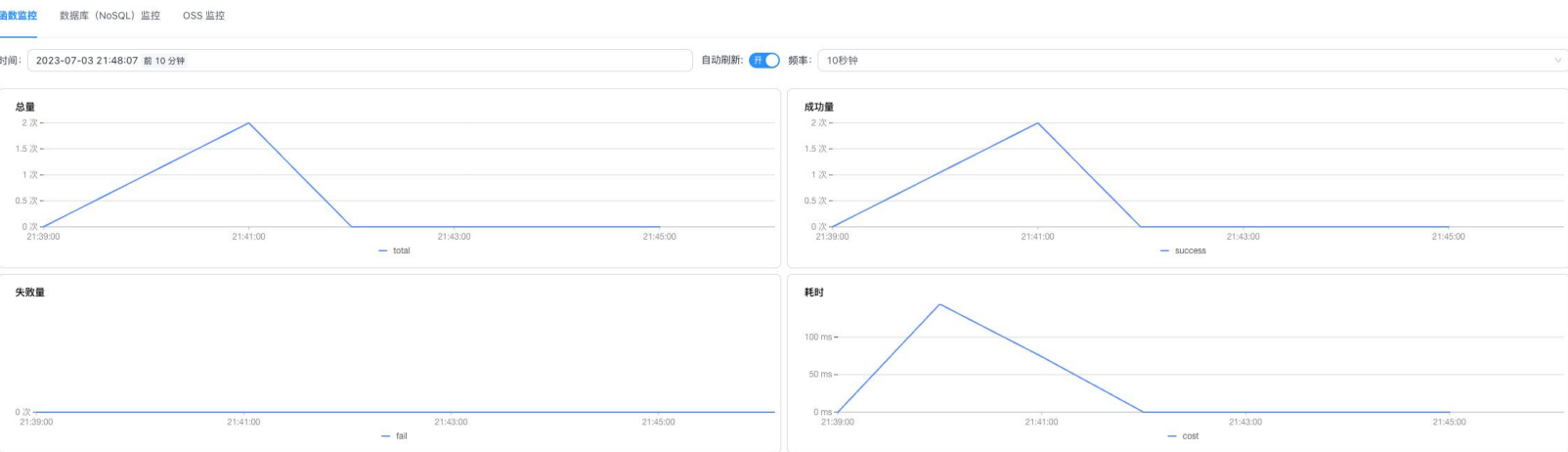
执行记录

日志查询

指标监控

请输入关键字查询

2023-07-03 21:41:24	[echo] [USER] [INFO] [0a4b4bb016883916841803921346] execute: 8efcd268-029f-4bad-80fb-ae1d8fe2d972-lead-vtg-0-2-10-0,function result: {"message":"Hello World!"}
2023-07-03 21:41:09	[echo] [USER] [INFO] [0a4b4baf16883916698596921346] execute: 75c757e5-1e5c-4a3d-b5f5-bf43f36c0518-lead-vtg-0-2-9-0,function result: {"message":"Hello World!"}
2023-07-03 21:40:57	[echo] [USER] [INFO] [0a4b4bae16883916574713901382] execute: 75c757e5-1e5c-4a3d-b5f5-bf43f36c0518-lead-vtg-0-2-9-0,function result: {"message":"Hello World!"}



报警规则

请输入规则名称

新建规则

规则名称	状态	类型	触发条件	等级	创建人	创建时间	操作
abc	启用	自定义	函数监控 总量 最近1个周期平均值等于0 周期为一分钟	低	cloudmonitor	2023-07-03 12:44:35	编辑 删除

链路

日志

指标

告警

FaaS研发效能：Serverless一站式平台、免运维、自动集成监控告警

蚂蚁 FaaS 的改变和优势

低成本

- 内存开销 < 1MB
- 启动速度 < 90ms
- 用户只为流量付费

安全保障

- 下一代安全容器
- 免鉴权调用

研发效能

- 纯业务、快写开发
- 无应用申请流程
- 无碎片代码和版本

运维效率

- Serverless一站式平台
- 免运维
- 自动集成监控告警等

FaaS 的实践和收益

01

典型 FaaS 实践案例

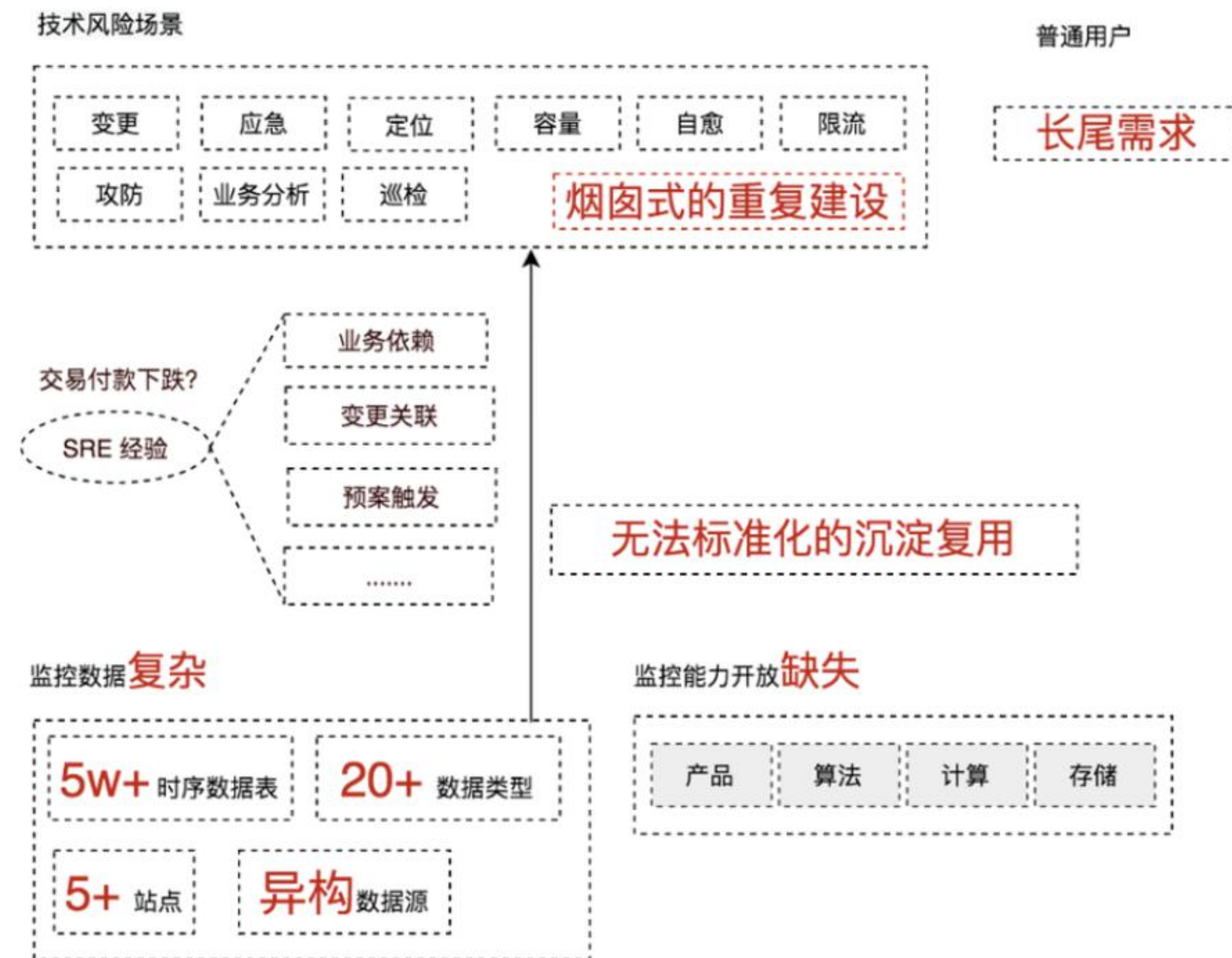
02

FaaS 带来的收益和改变

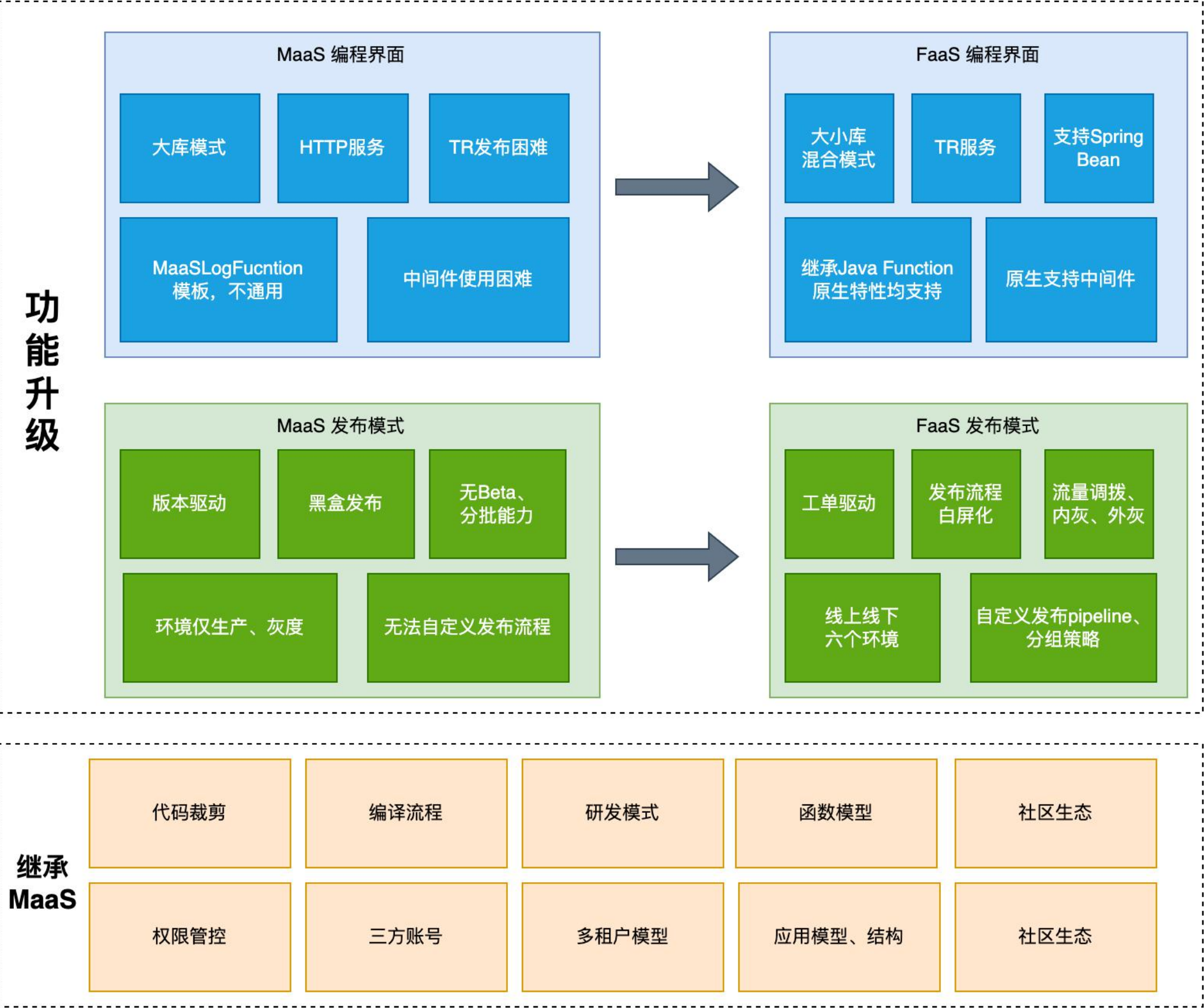
典型 FaaS 实践案例 MaaS On FaaS

MaaS 面临的问题

- 定制化监控分析及使用场景多，对于差异化诉求，难以平台化支持及快速响应。
- 监控数据抽象不足，用户难以标准化、便捷的使用监控实体模型 及 时序数据。
- 监控及风险领域的 AIOps 平台架构缺少重要一环，即 单场景 及 多个单场景 运维模块串联的流程化 AI 运维能力



典型 FaaS 实践案例 MaaS On FaaS



体系无缝整合

与现有应用框架及中间件打通

函数式运维能力升级

支持流量调拨、金丝雀发布等

风险平台能力整合

函数监控，灰度、流量、变更等平台能力整合

MaaS 通用模块下沉，业务部分重构为垂直样板间，快速构建标准化 FaaS 研发平台

典型 FaaS 实践案例 洛克公园

研发团队情况

- 4位前端同学，同时维护前后端 及 函数代码；相对编程能力较弱。
- 一位后端同学，维护阿里云应用

业务痛点

- 业务需求多：客户业务快速发展，业务需求频繁，但现在后端服务发布或者小程序发布流程都比较复杂，难于快速响应业务需求。
- 研发门槛较高：希望尽量降低研发门槛，比如更简单便捷的监控，业务链路的自动串联等等。
- 云成本较高：希望尽量降低研发成本，希望支付宝小程序云能够进一步降低服务器的成本。



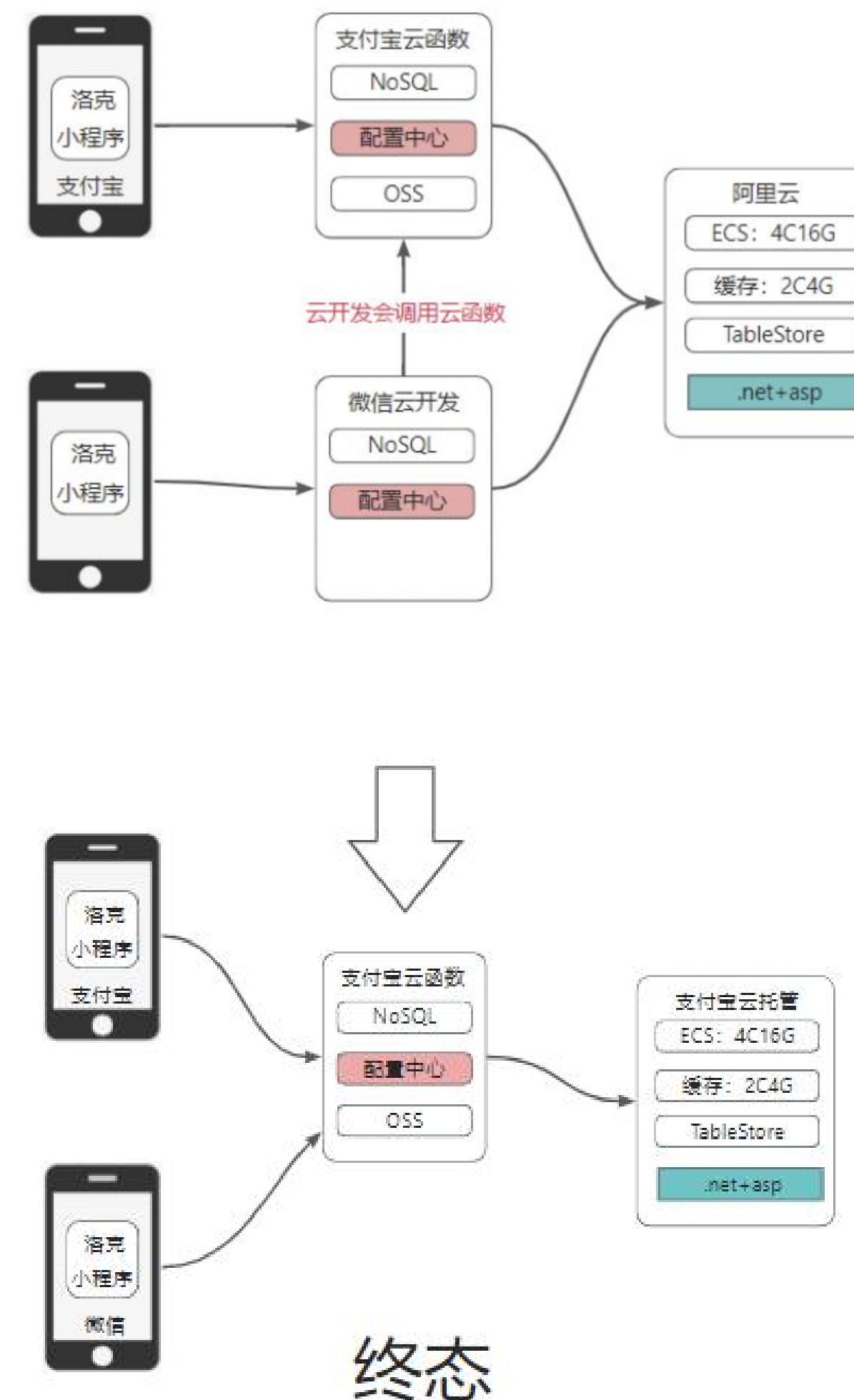
典型 FaaS 实践案例 洛克公园

核心诉求

- 统一技术栈以及较低的研发门槛
- 快速业务研发与发布
- 运维托管和流量波动下的动态扩缩容

成果收益

- 研发迭代速度显著提升
- 新需求的发布时间从原来需要数天时间，成功缩短至仅需数小时即可全部完成
- 计费成本更低



未来展望

01

极致性能

02

极致效能

未来展望

极致性能



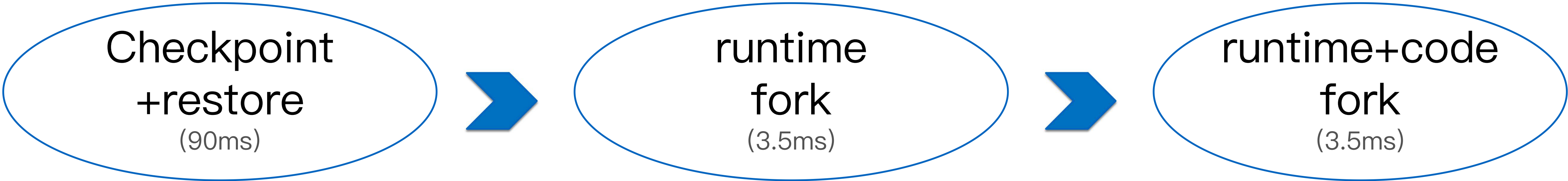
FORK

极致效能



AIGC

极致性能：毫秒级的冷启动



```
1 System Info:
2   Kernel: 5.10.112-005.ali5000.alios7.x86_64
3   CPU type: Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz
4   CPU frequency: 2499.919
5   cpuset.cpus: 0-63
6   cpuset.mems: 0
7
8   Count: 500
9   Total: 1.766555049s
10  Slowest: 5.116015ms
11  Fastest: 2.953933ms
12  Average: 3.529154ms
13  Requests/sec: 283.04
```

极致效能：AIGC + FaaS



AI 函数开发

自然语言对话

业务代码生成

语义式检索

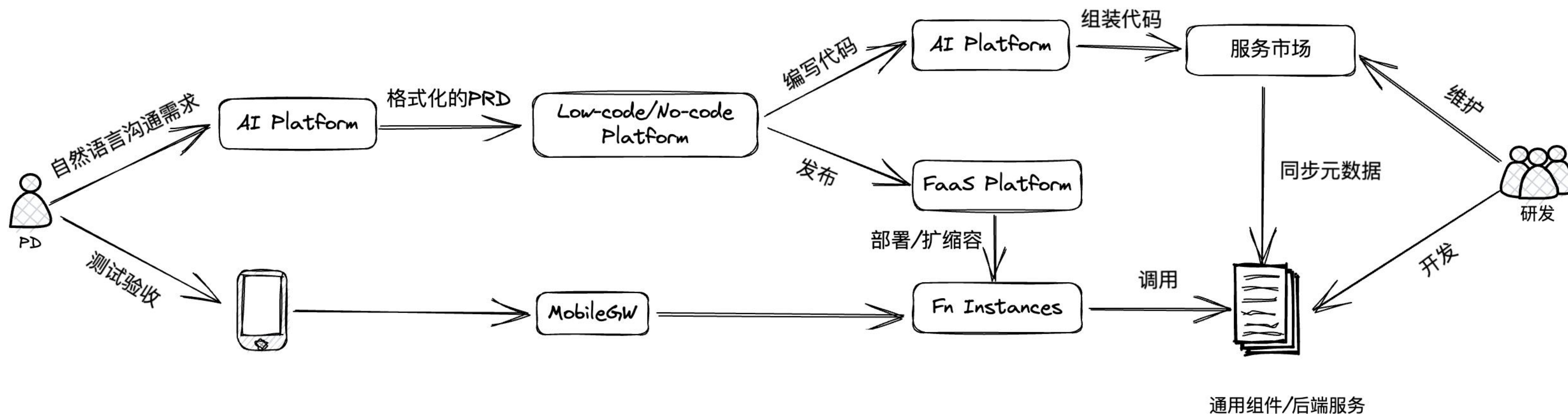
模版及组件导购

风险托管

流程自动化

蚂蚁 NLPGPT, CodeGPT, OpsGPT,.....

极致效能：AIGC + FaaS



想一想，我该如何把这些 技术应用在工作实践中？

THANKS



Rickey17

上海 浦东新区



扫一扫上面的二维码图案，加我为朋友。