

Architect

SACC

2022 中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2022

· 激发架构性能 点亮业务活力

云上会议 网络直播 | 2022年10月27-29日

IT168.com

ChinaUnix.net

ITPUB

# 中国银联服务网格技术实践

中国银联 资深架构师 丁春华





## 微服务治理成本

服务发现：服务发现

服务调用：使用RPC/Restful协议进行通信

服务追踪：链路追踪

服务容错：重试/熔断/服务降级

服务安全：服务鉴权

服务部署：顺序部署/滚动部署/蓝绿/金丝雀

数据调用：分布式事务

分组限流：路由/限流



## 问题

### 框架能力分散

服务治理能力、标准不统一、重复建设  
整体治理能力升级相对滞后  
不满足部分业务快速迭代的需求

框架与应用紧耦合，不能独立升级  
应用引入不同版本SDK，引发冲突  
应用存在多种开发语言，需重复实现

### 嵌入框架SDK

### 云上竞争力不足

弹性扩缩容能力相对薄弱  
大规模服务集群治理能力缺失

## 建设目标

- 01

✓

### 标准化服务治理框架能力，避免重复建设

兼容现有服务治理框架能力  
形成内部标准化的服务治理 API  
形成银联内部的服务定义和服务管理流程
- 02

✓

### 解决服务治理和业务开发的耦合问题

支持多种语言(JAVA, C, GO)  
支持边车独立迭代升级(轻薄SDK)  
支持多协议插件独立升级(兼容存量框架)  
直接对接银联基础设施, 应用专注业务开发  
避免80%以上非业务功能重复开发
- 03

✓

### 解决公司级大规模服务治理问题

支持分层分区管控, 单管控单元万级节点数, 按需扩容  
支持管控指令、配置规则秒级生效  
支持灰度路由、中心路由、调用链跟踪、双发等高级功能  
支持问题快速定位, 可视化集群拓扑、实时交易质量、事件热力图

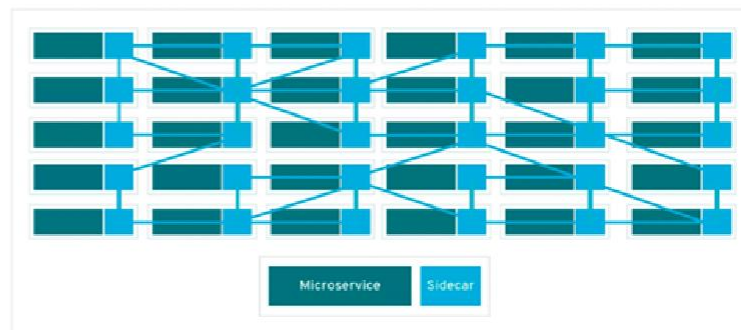
下沉基础服务设施层，为系统开发和运维人员，提供统一易用的应用管理和服务治理能力。

## 服务网格(ServiceMesh)

A *service mesh* is a configurable, low-latency infrastructure layer designed to handle a high volume of network-based interprocess communication among applications on infrastructure services using application programming interfaces (APIs).

服务网格是可配置的、低延迟基础设施层，旨在处理基于API的应用架构服务之间的大量网络通信。

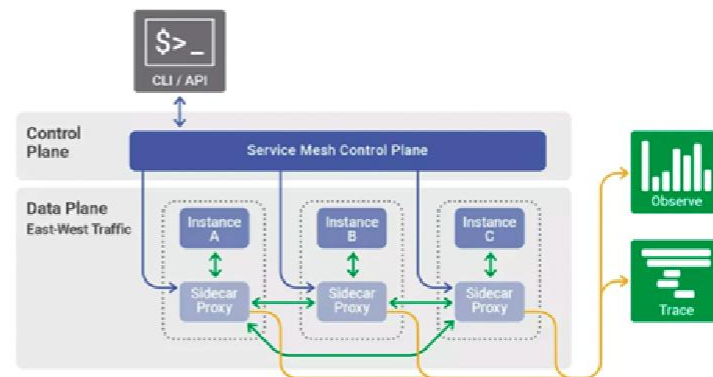
Ngixn

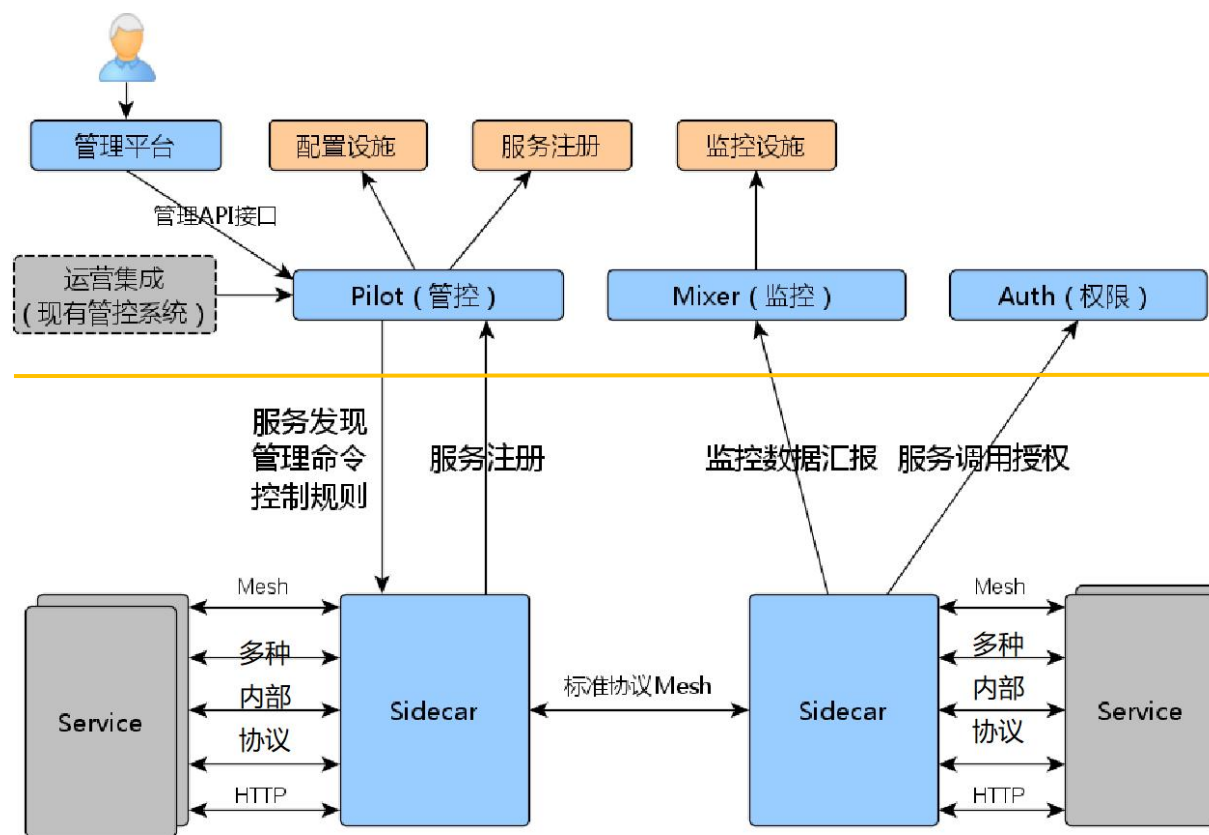


A *service mesh* is a tool for adding observability, security, and reliability features to applications by inserting these features at the platform layer rather than the application layer.

服务网格是一个工具，在平台层向应用程序增加可观察性、安全性、可靠性功能，而不是应用层。

Buoyant







### 1.UPSC

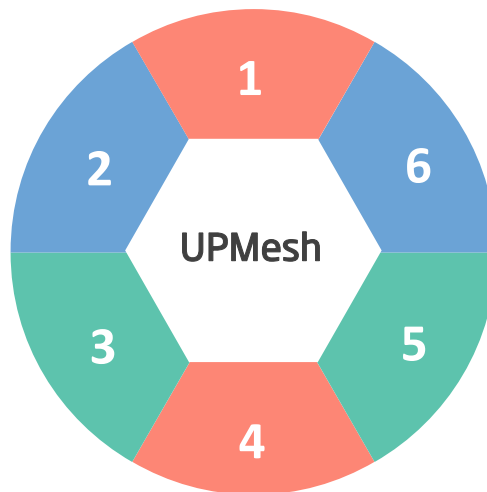
节点管理、进程守护、服务通信、服务治理、协议转换

### 2.Console

配置、管理、监控、处理

### 3.Mixer

收集、汇聚、转送



### 4.Pilot

注册、发现、互联互通、配置下发、遥测、控制指令

### 5.SMI

联机运维、云管整合

### 6.Monitor

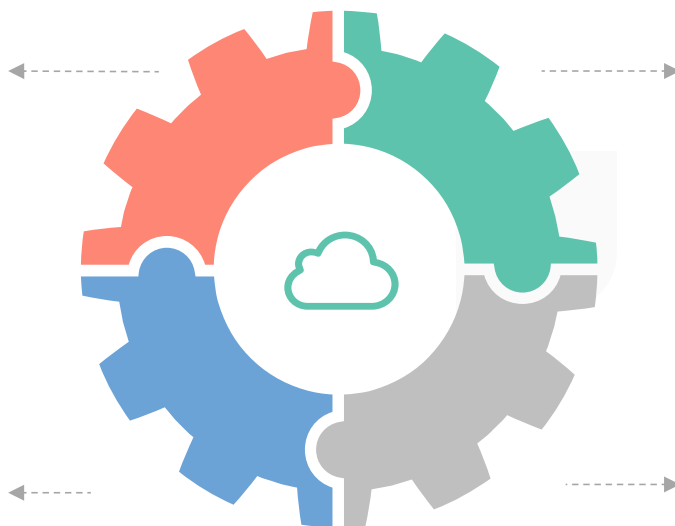
计算、分析、报告

## 服务治理

- ✓ 服务注册发现
- ✓ 服务路由（灰度、异地）
- ✓ 服务限流、熔断
- ✓ 负载高可用（动态、双发等）
- ✓ 服务流量调度
- ✓ 服务鉴权
- ✓ 故障注入

## 服务生命周期管理

- ✓ 服务上线、下线
- ✓ 服务守护、优雅升级
- ✓ 服务配置（流控）的管理、下发



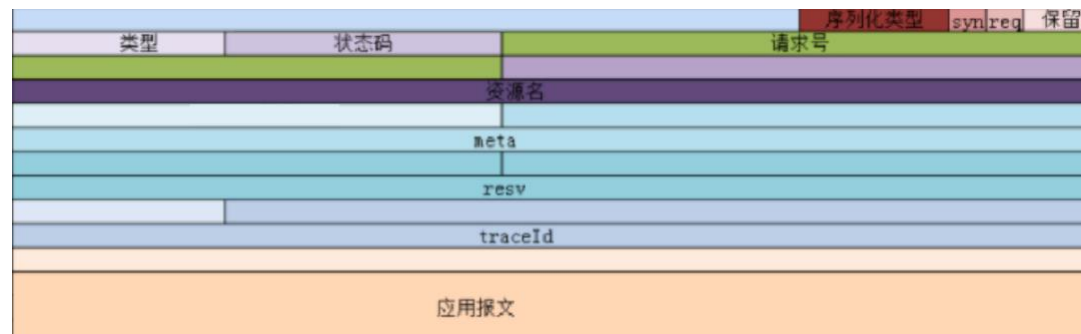
## 数字化运维

- ✓ 服务的管理、状态监控
- ✓ 服务调用链跟踪（对接现有系统）
- ✓ 服务集群和单点的运行质量（次数、qps、耗时、成功率等）
- ✓ 节点资源状态

## 其他

- ✓ 泛服务化能力
- ✓ 内部协议转换
- ✓ 适配标准治理API (XDS)
- ✓ 推进API管理

## 通信协议



## 万物皆服务

m3 :// modulea.sysa.up / svra ? api=xxx & unit=DC01



服务协议



系统名/子系统名



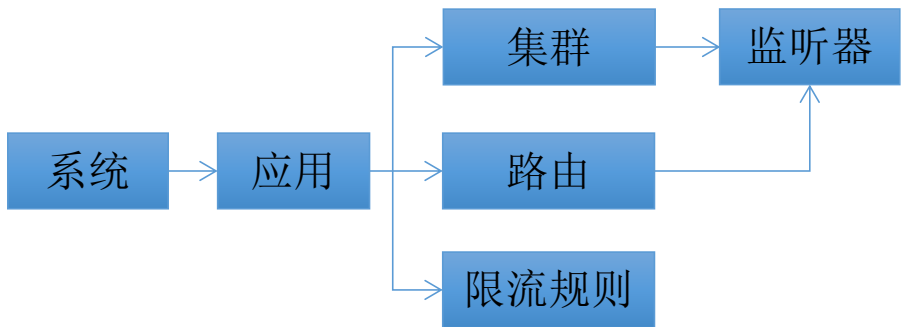
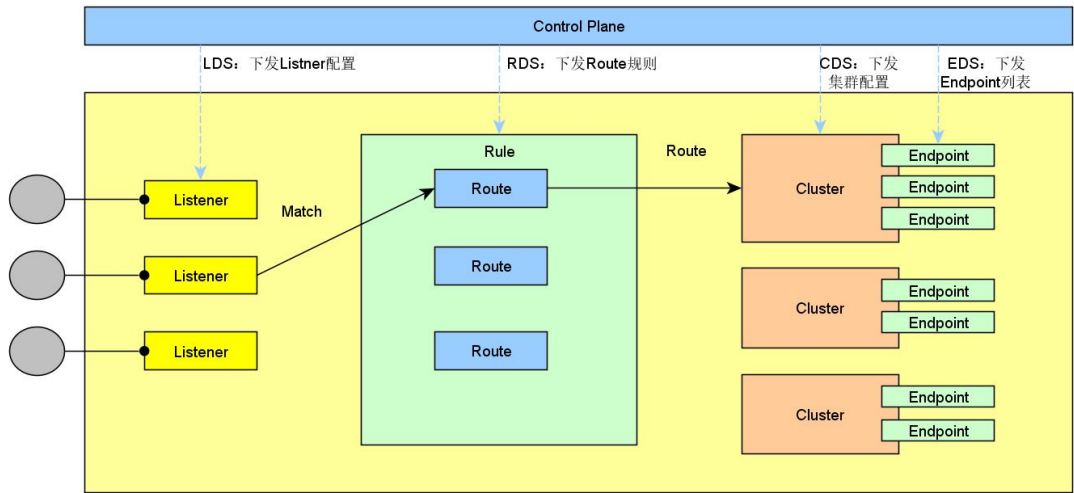
服务名



其他参数



# 概念抽象



名词	概念	描述
服务URL	服务资源统一标识符	由协议，系统，服务和其它参数四部分组成，例如： m3://sysa/svr?api=xxx&unit=DC01
Listener	监听器	应用监听端口和相关路由规则的配置
Route	路由	服务URL与集群之间映射关系的配置
Cluster	集群	应用的多个对等的部署实例
Endpoint	节点	集群(Cluster)中可用的IP和端口

概念	描述
系统	系统指提供一套完整服务的多个应用的组合。
应用	可对外提供特定服务或完成特定业务逻辑处理任务的应用程序，可用于配置路由、限流等规则。
集群	应用的多份部署实例组成的一个逻辑单位，可用于配置监听器。例如：sysa.svr.dc01.cluster，sysa.svr.dc01.cluster。

# 路由规则秒级生效，预置多种灰度发布方式

## □ 金丝雀发布

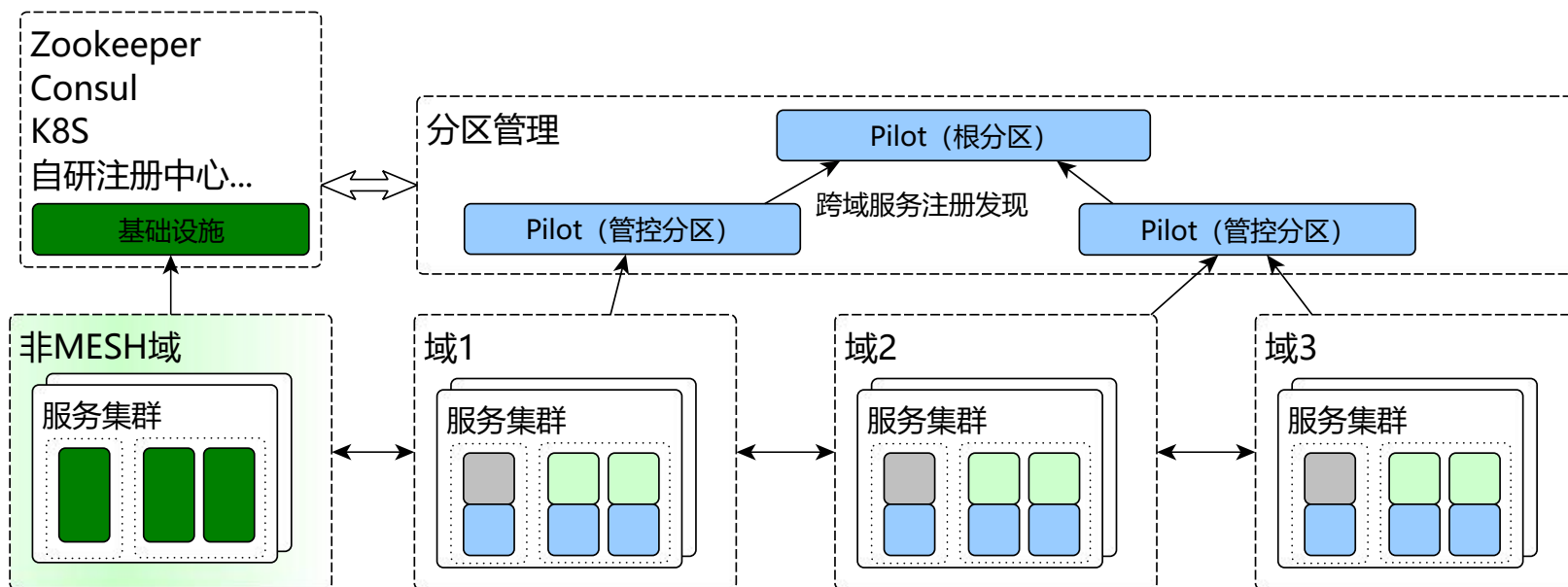
```
id: sysa.svr.dc01.route
url: m3://sysa/svr?unit=DC01
dst:
  - cluster: sysa.svr.dc01.cluster
    weight: 99%
    subset:
      version: 1.0
  - cluster: sysa.svr.dc01.cluster
    weight: 1%
    subset:
      version: 2.0
```

## □ 蓝绿发布

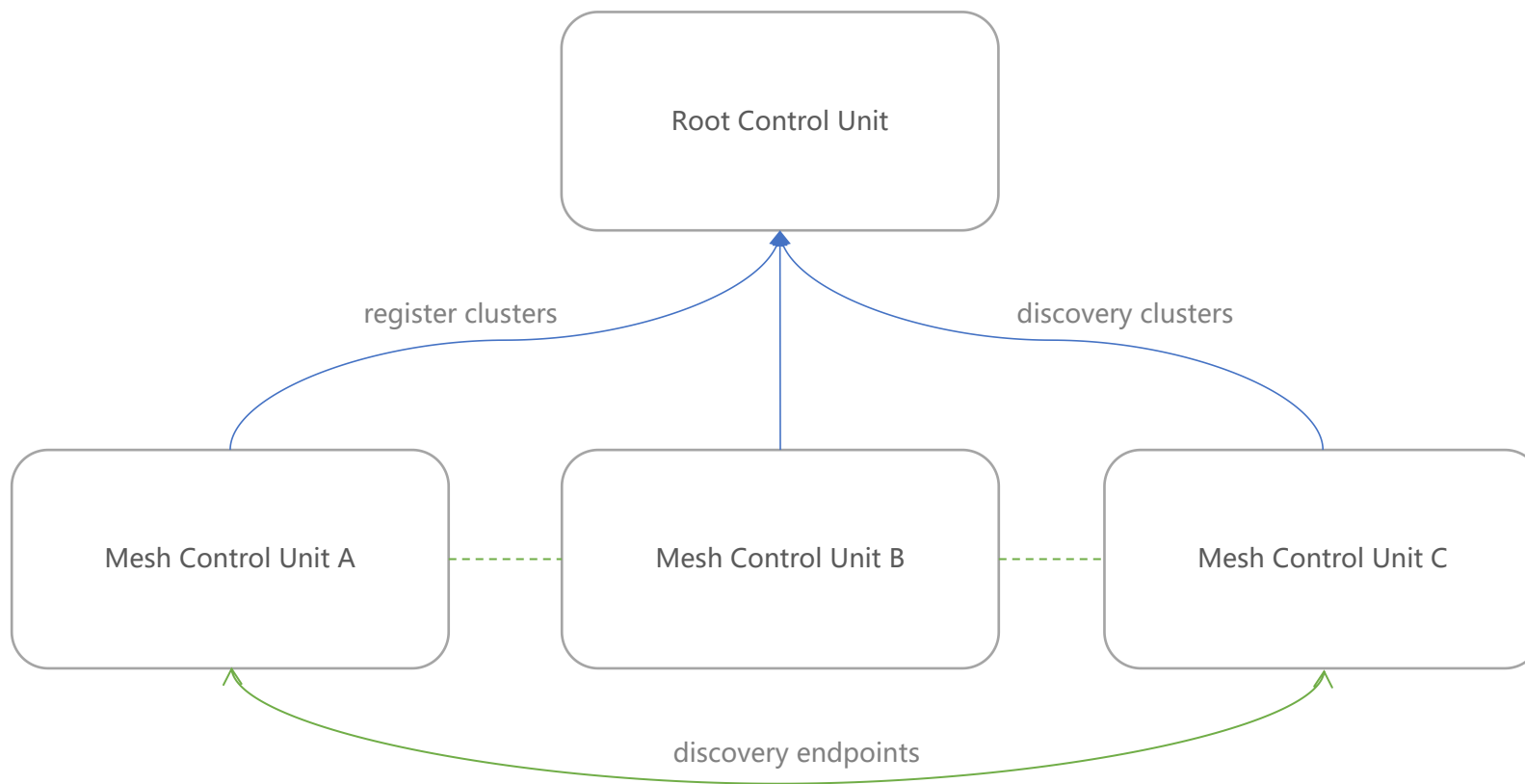
```
id: sysa.svr.route
url: m3://sysa/svr?unit=DC01
dst:
  - cluster: sysa.svr.blue.dc01.cluster
dst:
  - cluster: sysa.svr.green.dc01.cluster
```

## □ ABTest

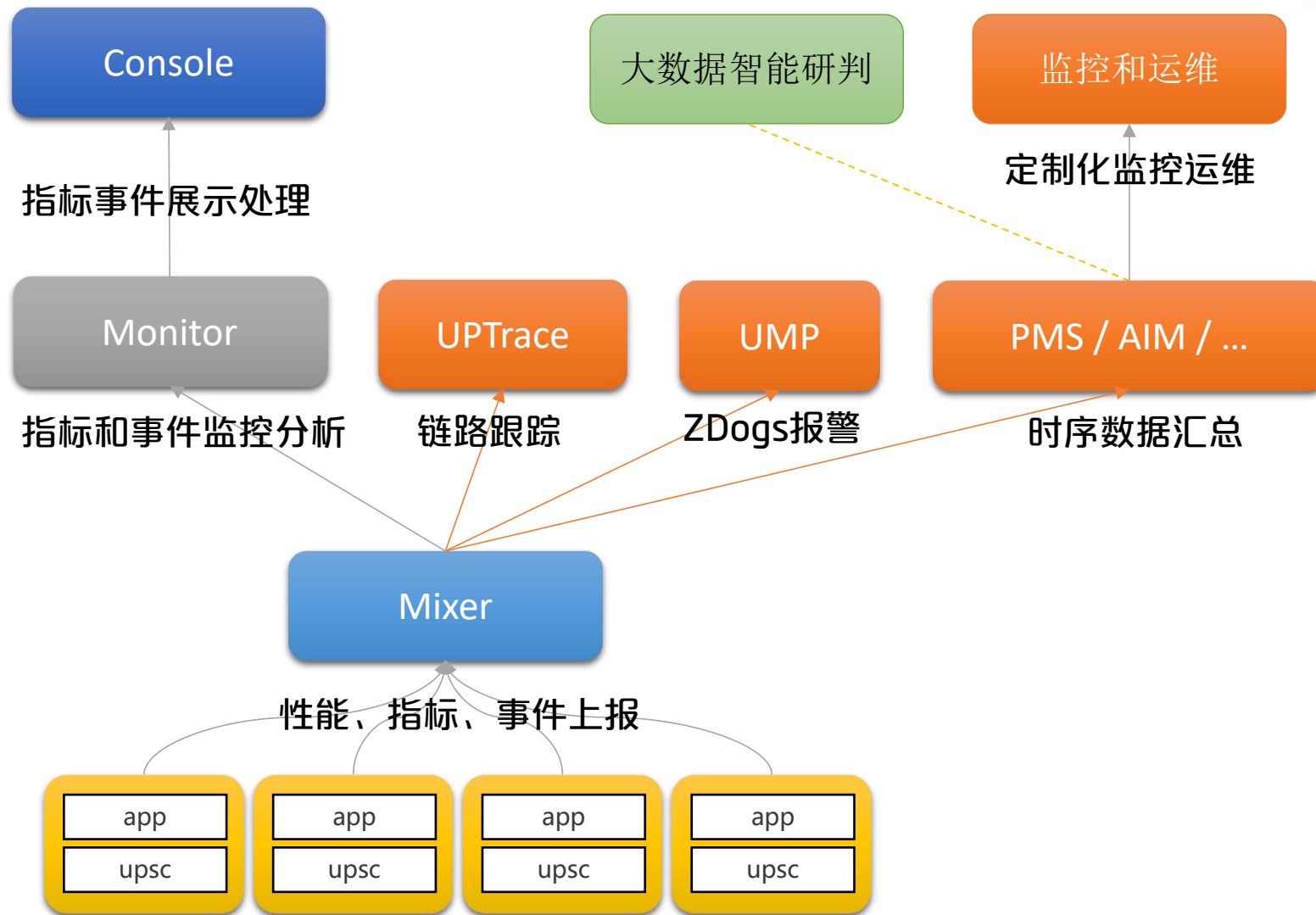
```
id: sysa.svr.route
url: m3://sysa/svr?unit=DC01&reqHash=A
dst:
  - cluster: sysa.svr.A.dc01.cluster
url: m3://sysa/svr?unit=DC01&reqHash=B
dst:
  - cluster: sysa.svr.B.dc01.cluster
```



- ✓ 分区管控，独立存取： 划分不同管控分区，各分区数据独立存取
- ✓ 分级可见，权限管理： 被调方路由指定系统、集群可见，调用方主动引用
- ✓ 互联互通： 多个不同注册中心相互联通
- ✓ 容量性能： 单管控分区万级节点数，秒级增量生效







调用监控



调用链分析

#	Start Time	Path	Res. (ms) ↓	Exception	Agent	Client IP	Transaction
1	05/13 19:54:31.344	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*342
2	05/13 19:53:35.337	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*341
3	05/13 19:52:31.330	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*340
4	05/13 19:51:32.323	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*339
5	05/13 19:50:36.316	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*338
6	05/13 19:49:32.308	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*337
7	05/13 19:48:36.302	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*336
8	05/13 19:47:32.295	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*335
9	05/13 19:46:29.288	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*334
10	05/13 19:45:32.282	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*333
11	05/13 19:44:29.275	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*332
12	05/13 19:43:26.269	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*331
13	05/13 19:42:36.262	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*330
14	05/13 19:41:32.253	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*329
15	05/13 19:40:32.246	m3://mesh/mixer	0		atbasas172.21.52.77	127.0.0.1:47382	atbasas172.21.52.77*1589351257346*328

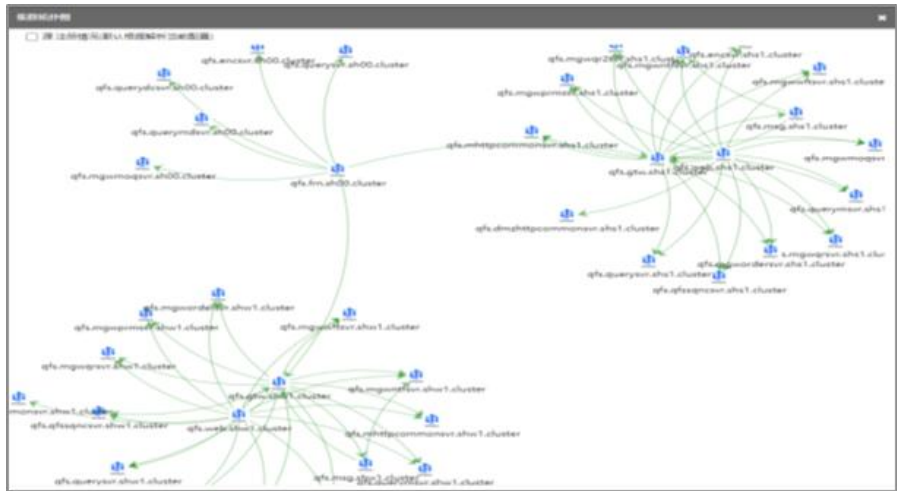
  

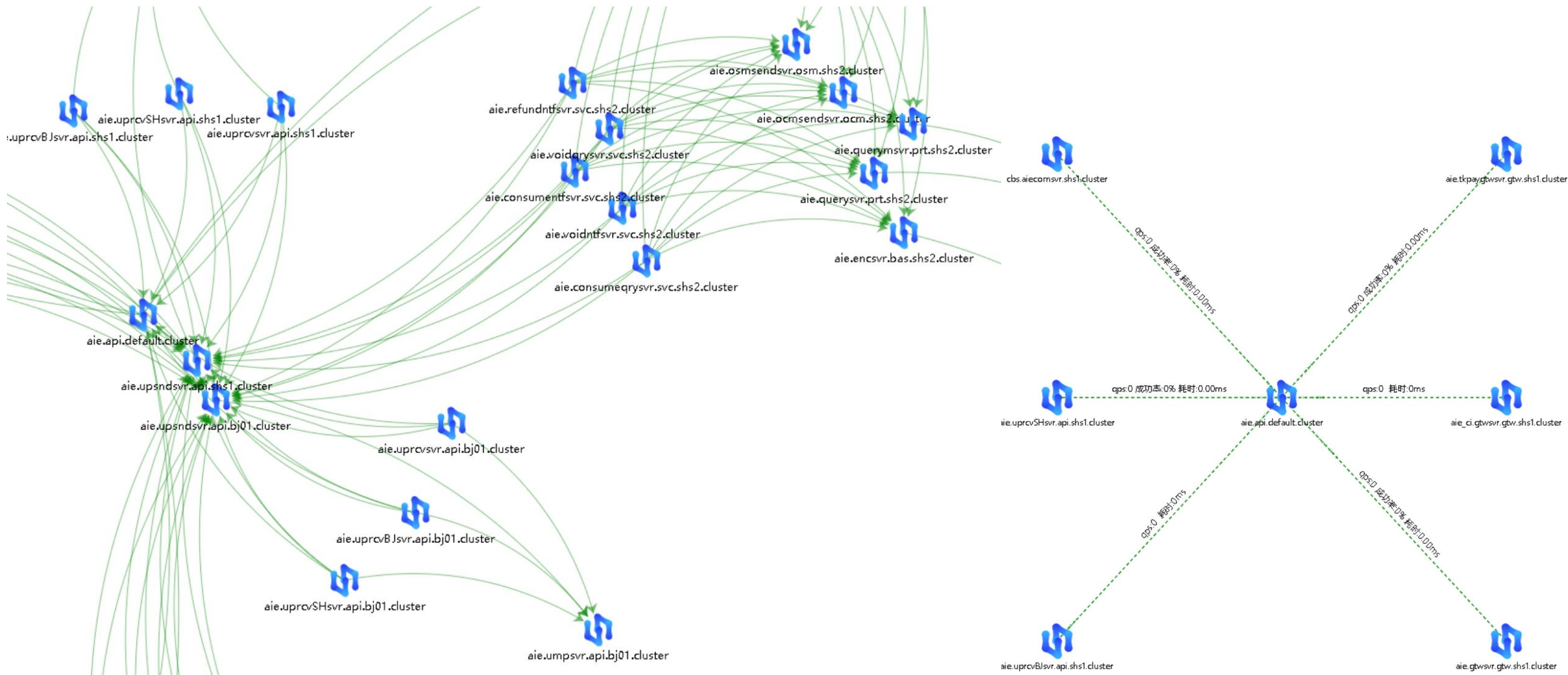
Method	Argument	Start Time	Gap(ms)	Exec(ms)	Exec(%)	Self(ms)	Class	API	Agent	Application
mesh_call1@	m3://mesh/mixer	19:49:32.308	0	0	0	0		WAPIE_PROD...	atbasas172.21.52...	opt_d_fm_q
NOTE_ADDRESS	127.0.0.1:47382									
["value":{"m3://mesh/mixer"},"selfField":{"s"578		19:49:32.308	0	0	0	0		WAPIE_CONS...	atbasas172.21.52...	opt_d_fm_q
mesh_call1@	m3://mesh/mixer	19:50:37.113	64895	0	0	0		WAPIE_PROD...	upmeshes172.21.53...	mesh.mixer
NOTE_ADDRESS	172.21.52.77:41316									

根因分析



服务全景拓扑图









## 初见成效

01



基础能力下沉，技术红利初显，  
并将长期释放

02



存量系统低成本接入改造低

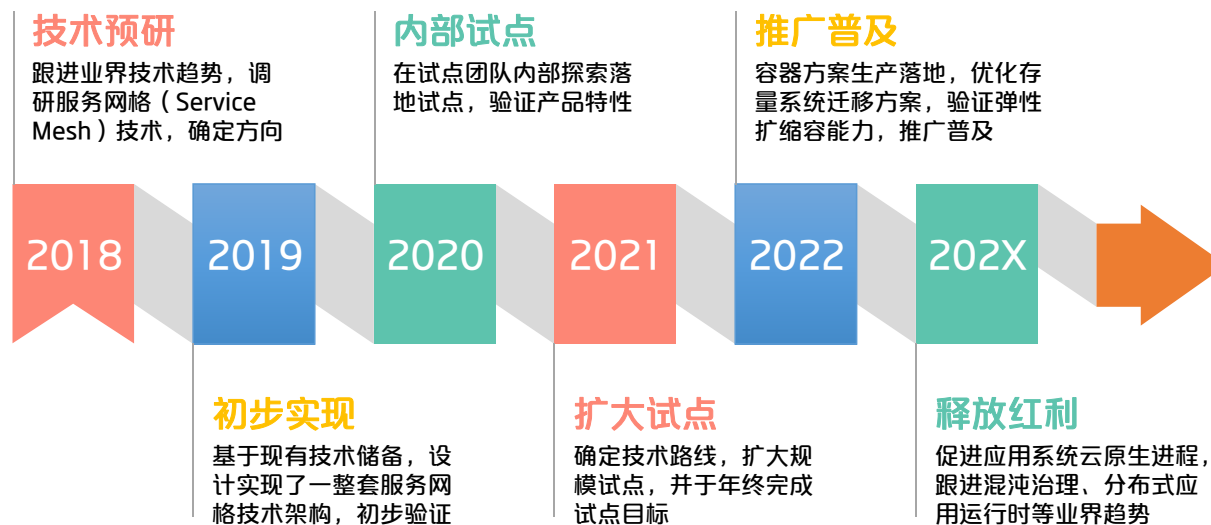
改造简单、无损切换、支持业务维度的灰度和中心切换

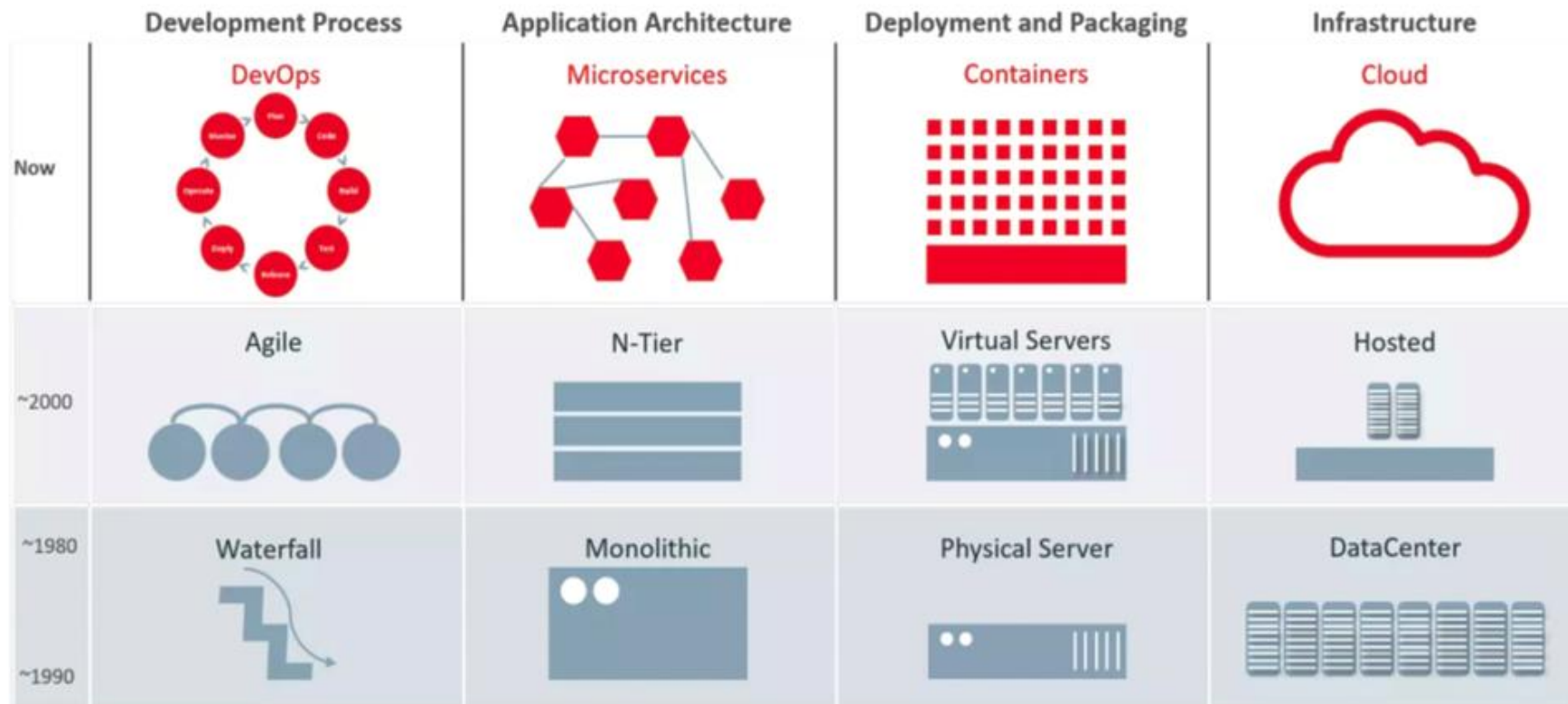
03

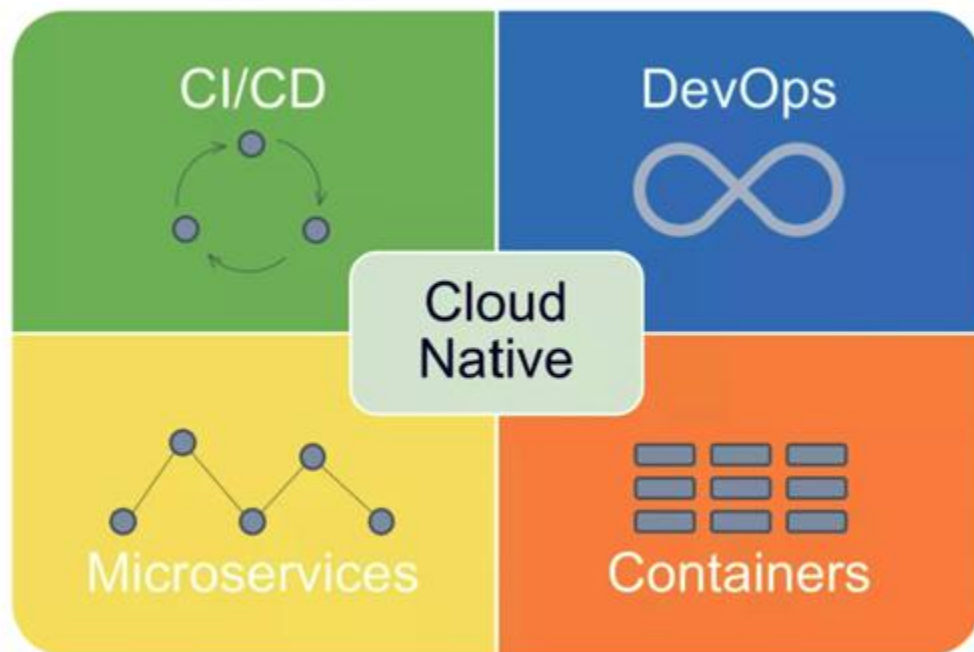


直接获得众多基础能力

运营管理、微服务生命周期、服务治理、数字化监控







## Cloud Native

Cloud native is an approach to building and running applications that exploits the advantages of the cloud computing delivery model. When companies build and operate applications using a cloud native architecture, they bring new ideas to market faster and respond sooner to customer demands.



### 提升稳定性和易用性

优化MESH升级，服务治理，  
管理监控，性能高可用等功能，  
提升稳定性和易用性，  
满足用户使用需求。

## 2021年

产品更加稳定、易用  
至少落地5个重要系统

### 容器化集群支持

引进容器化控制面组件，实现  
虚拟机和容器统一治理，  
具备容器化集群治理能力。

## 2022年

虚拟机和容器统一治理，  
支撑容器化集群  
满足混合部署使用需求

### 集成丰富的云原生组件

满足更为云原生化服务治理需求，  
为用户提供更为丰富的服务治理能力。

## 2023年

完善产品的云原生特性，  
比如基础设施服务化、  
分布式运行时、Chaos  
Mesh等，满足用户更为丰富的  
云上系统建设需求







THANKS