腾讯云 | 连接智能未来

腾讯TSF微服务平台

ServiceMesh最佳实践

演讲者　卢政　腾讯云微服务平台产品负责人

new trend
new technology
new application

Cloud + community
Developer conference

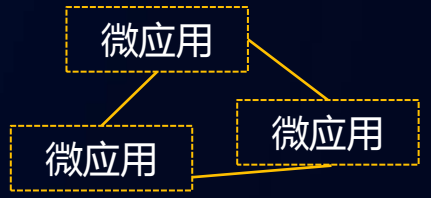
第一阶段 单体架构

- 紧耦合，系统复杂、错综交互，动一发而牵全身
- 重复制造各种轮子：OS、DB、Middleware
- 完全封闭的架构

第二阶段 SOA架构

- 通常通过ESB进行系统集成，松耦合
- 需要集中式、计划内停机扩容或更新
- 团队庞大交互负载

第三阶段 微服务架构



- DevOps: CI, CD, 全自动化
- 可扩展性：自动弹性伸缩
- 高可用：升级、扩容不中断业务

路漫漫其修远兮

“后台研发工程师：Dubbo/Spring cloud 才刚学会皮毛，ServiceMesh又横空出世，这么多概念，模块，怎么办？！”

“运维工程师：开发大哥，帮运维兄弟们加个需求，帮忙搞个白名单工具，来自Chrome浏览器的用户流量，导20%到v1.21版本模块里，回切接口也帮忙整一个，感谢啊！“

“CTO：都突发半小时了！版本回滚的方案还拿不出来。你们干什么吃的？！”

腾讯云TSF －企业技术中台最佳实践

核心能力：

- 服务治理
- 应用生命周期管理
- 配置中心
- 分布式事务
- 数据化运营：日志、监控、告警、调用链

中间件平台：

- 分布式计算调度、配置、事务能力
- 微服务API网关
- Java Spring层面打通消息队列CMQ/CKafka

容器化应用发布管理

- 集成TSF服务注册发现、调用链、基础监控，RPC服务监控
- 支持应用的灰度发布，http流量灰度发布
- 支持有状态应用

Spring Cloud商业化版

- 原生Spring boot、Spring cloud开发体验
- 迁移无改造量，自动接入TSF注册中心
- 支持JAR包、WAR包应用
- 提供Dubbo应用快速迁移方案，无入侵性
- Spring 框架层面整合腾讯消息队列、API网关、云企业总线、云kafka大数据套件等传统中间件

Tencent Service Mesh

- 原生istio、envoy进行深度构建，商业化版本于2018年8月正式发布
- 满足公司内部不同开发语言、框架的无缝互通
- Service Mesh通过sidecar的方式组织服务间的流量，开发者更聚焦业务开发

SpringCloud & TSF

腾讯云SpringCloud，是基于开源Netflix OSS、Hashicorp服务的商业化封装。无额外学习成本，稳定可用，适用于大型生产环境。TSF Spring Cloud还支持API级别的服务路由、鉴权、流量控制。

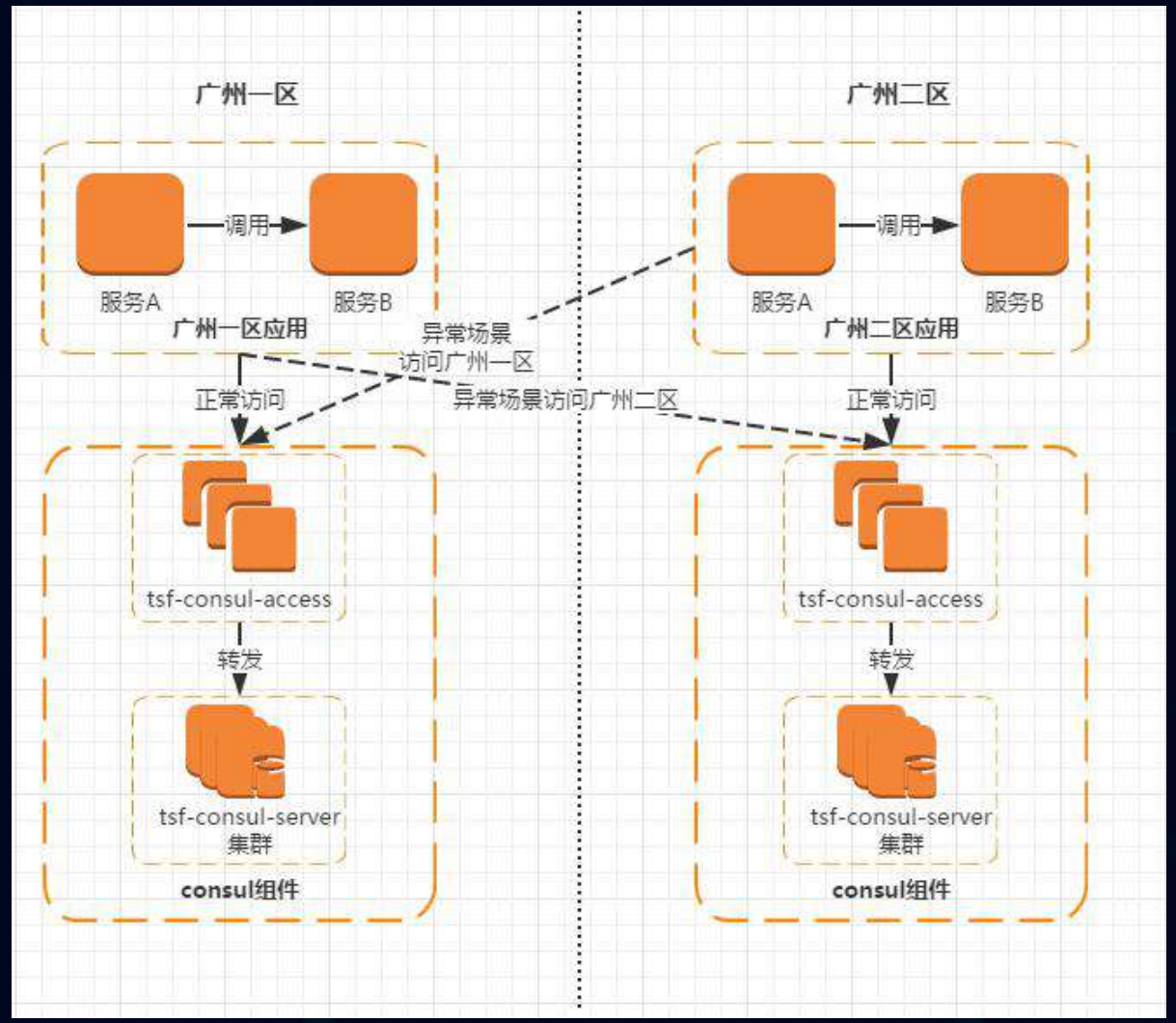
组件名称	组件功能	功能描述
spring-cloud-tsf-core	TSF Context 基础SDK依赖	基础依赖，如提供请求TAG设置等基础能力；
spring-cloud-tsf-encrypt	分布式配置加解密SDK依赖	提供分布式配置内容的加解密能力；
spring-cloud-tsf-consul	服务注册中心、分布式配置SDK依赖	提供服务注册和服务发现能力、分布式配置能力；
spring-cloud-tsf-auth	服务鉴权SDK依赖	根据服务鉴权规则，进行服务间调用的权限控制；
spring-cloud-tsf-sleuth	调用链SDK依赖	TSF扩展了spring-cloud-sleuth的能力生成服务调用链日志信息，进行服务调用的统计和运营数据；
spring-cloud-tsf-route	服务路由SDK依赖	根据服务路由规则，生效负载均衡策略时，选择满足路由规则的服务实例进行调用；
spring-cloud-tsf-ratelimit	服务限流SDK依赖	根据限流规则， 生效相应微服务限流能力；定期上报限流统计数据；
tsf-服务生命周期管理	Springcloud原生不具备	一键式服务扩缩，版本变更，灰度流量能力

SpringCloud & TSF 高可用注册中心

强调服务可用性，数据面一致性由存储层兜底

容灾场景1：同区域的服务全部异常的场景。

容灾场景2：单机房注册中心Consul不可用



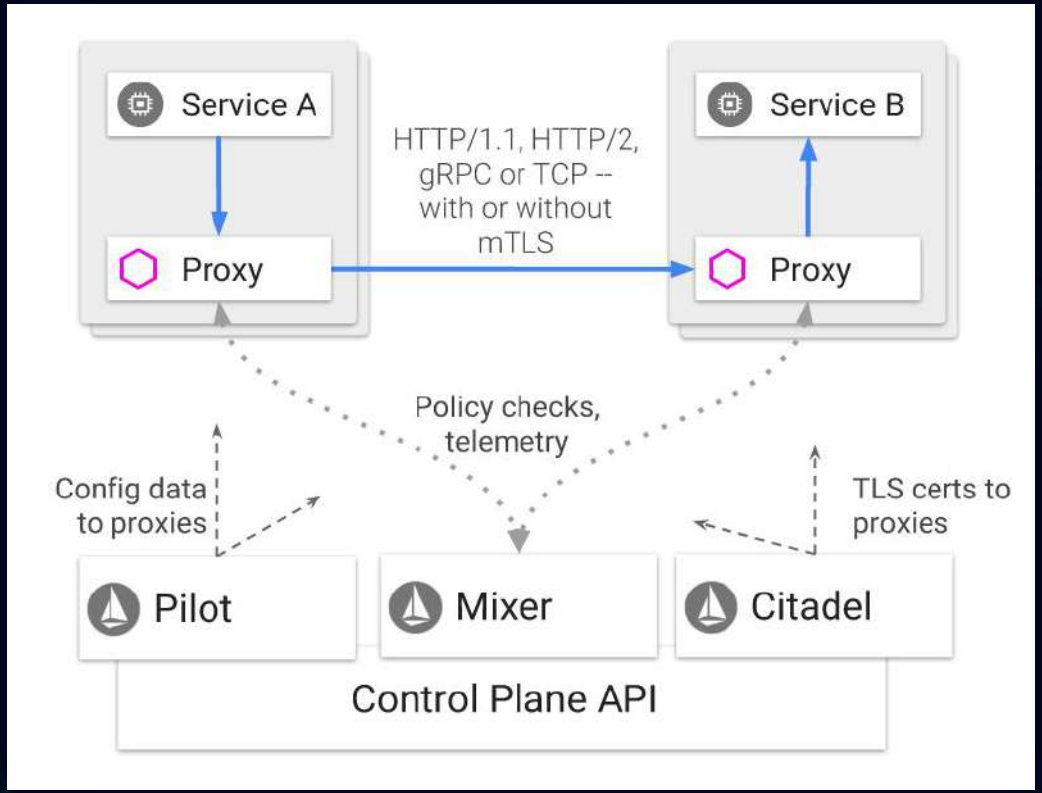
Service Mesh被定义为“下一代微服务平台”。核心组件有4个：

- Envoy：数据面代理，用来协调服务网格中所有流量的出入站流量，并提供服务发现，负载均衡、限流熔断等能力，还可以收集大量与流量相关的性能指标

- Mixer：前提条件检查（包括认证，黑白名单，ACL检查）、配额管理（如限速）、服务上报日志监控

- Pilot：控制器，用来将流量协调、流量灰度的策略和规则发送到Envoy代理，可谓服务网格里的”流量管理员“。

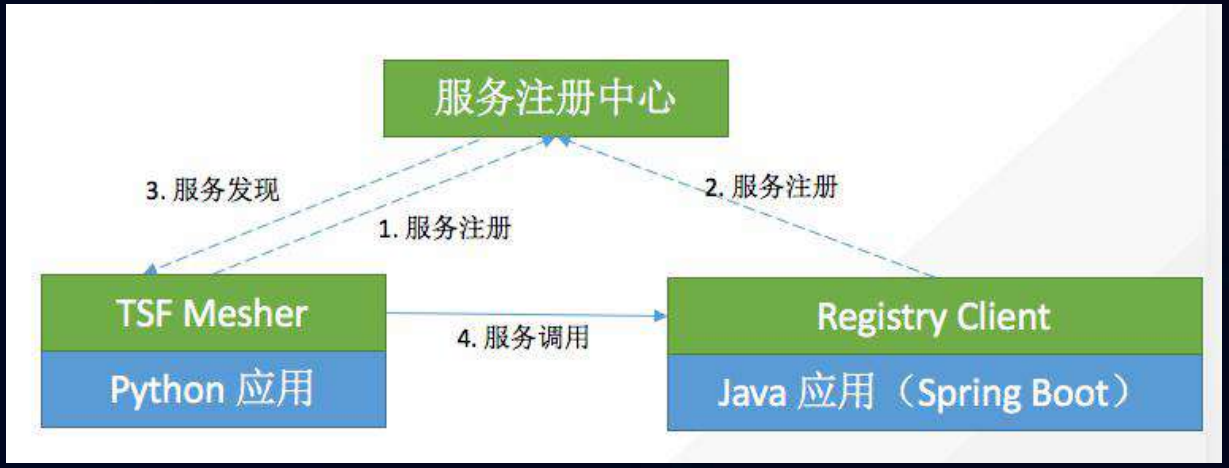
- Auth：是鉴权和安全（如TLS）控制组件。



ServiceMesh & TSF

提供Service Mesh方式，让旧应用不改一行业务代码接入TSF 微服务平台。且研发团队无需抽精力去迭代运维工具、灰度发布工具等。

- Mesher 和旧应用同机部署
- Mesher 替代旧应用注册服务
- Mesher 负责转发请求和响应



优势

- 旧应用不需要改造，可以被微服务应用访问
- 跨语言服务支持
- HTTP/1.1，HTTP/2，gRPC等流量的自动区域感知负载平衡和故障切换。
- 通过丰富的路由规则，容错和故障注入，对流行为的细粒度控制。
- 支持访问控制，速率限制和配额的可插拔策略层和配置API。
- 集群内所有流量的自动量度，日志和跟踪，包括集群入口和出口。
- 安全的服务到服务身份验证，在集群中的服务之间具有强大的身份标识。

ServiceMesh & TSF 优化

在开发选型上，控制面基于开源框架Istio 1.0 来进行构建，数据面基于开源框架envoy 1.7.0来进行构建。

TSF团队对ServiceMesh的能力所做的增强：

1、平台解耦：支持K8S/VM/裸金属服务器环境

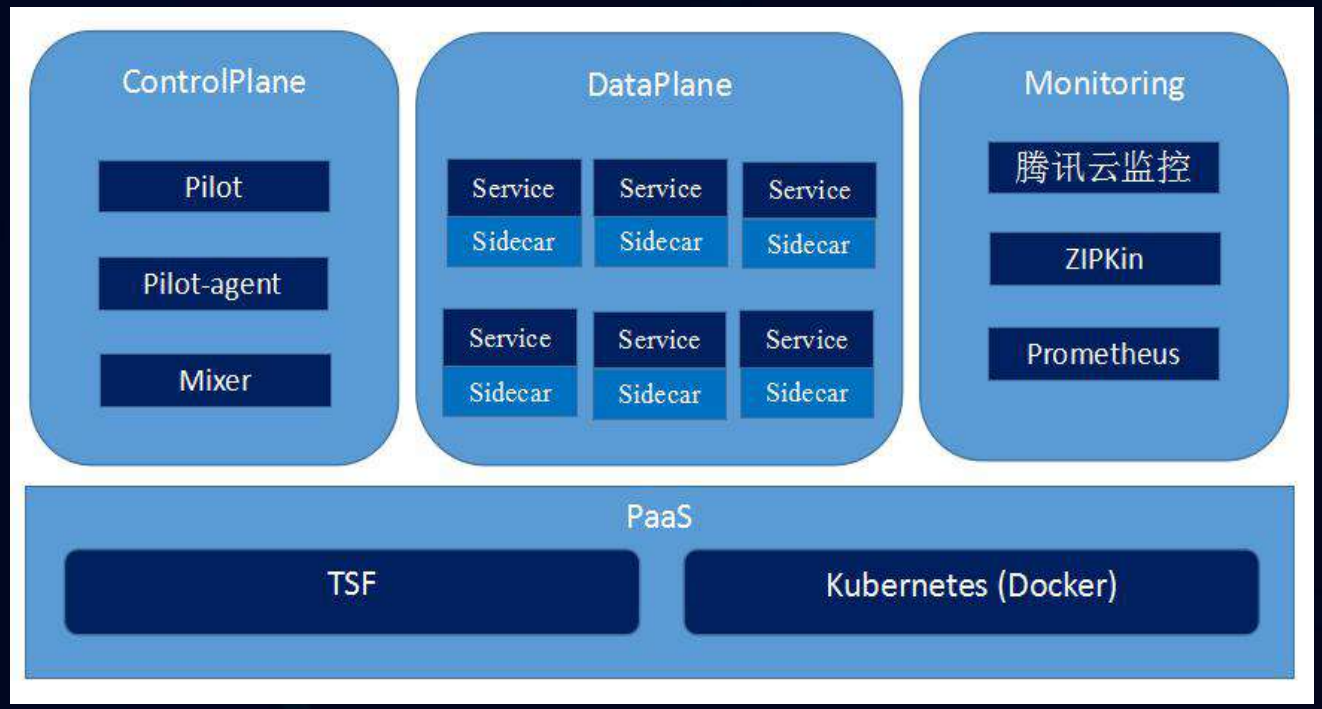
2、新旧兼容：支持Spring Cloud应用、ServiceMesh应用互通，统一治理

3、多租户隔离、管理支持

4、调用链日志落盘，监控大盘

5、 Mixer缓存穿透的问题 ，Envoy流量优化接管,Pilot性能优化等

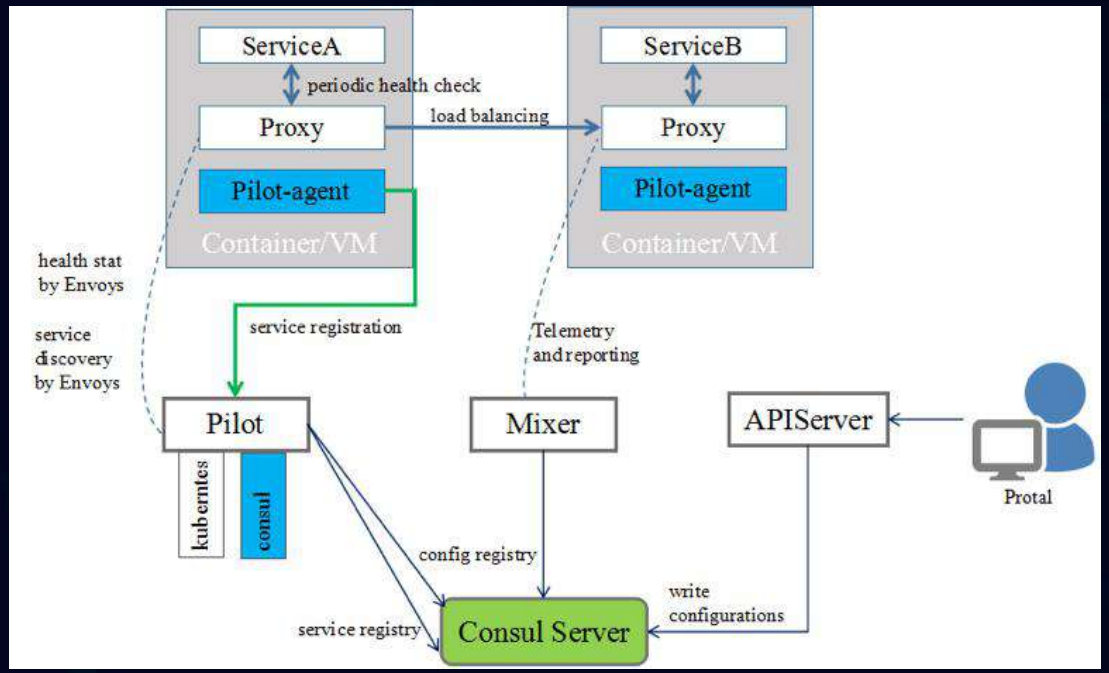
6、支持私有RPC协议（预研）



ServiceMesh & TSF 去K8S依赖

经过改造后，TSF Mesh成功与kubernetes平台解耦，支持VM/容器混合组网，可从容适配任何的底层部署环境 。

问题	增强点	目的
Pilot/Mixer的远程动态配置能力失效	组网中增加Consul Server集群作为服务注册及配置中心 Pilot/Mixer扩展ConfigController接口，增加基于consul的配置管理能力	彻底使用consul作为配置管理中心，脱离kubernetes的crd能力
Pilot无法获取服务节点健康信息	Pilot实现HDS接口，envoy通过HDS接口上报健康信息给Pilot，Pilot将健康信息上报给服务注册中心	使用envoy原生的HDS健康上报能力进行节点健康上报，脱离kubernetes的ReadinessProbe能力
无法通过istioctl进行服务注册	增强Pilot-agent能力，支持启动时根据描述文件自动注册服务 新增APIServer组件，对接consul，负责配置的CRUD	服务通过描述文件自动注册，方便用户进行一站式的微服务开发 通过APIServer屏蔽底层配置中心，对外提供简单的REST接口，方便管理Protal和二次开发SDK的对接

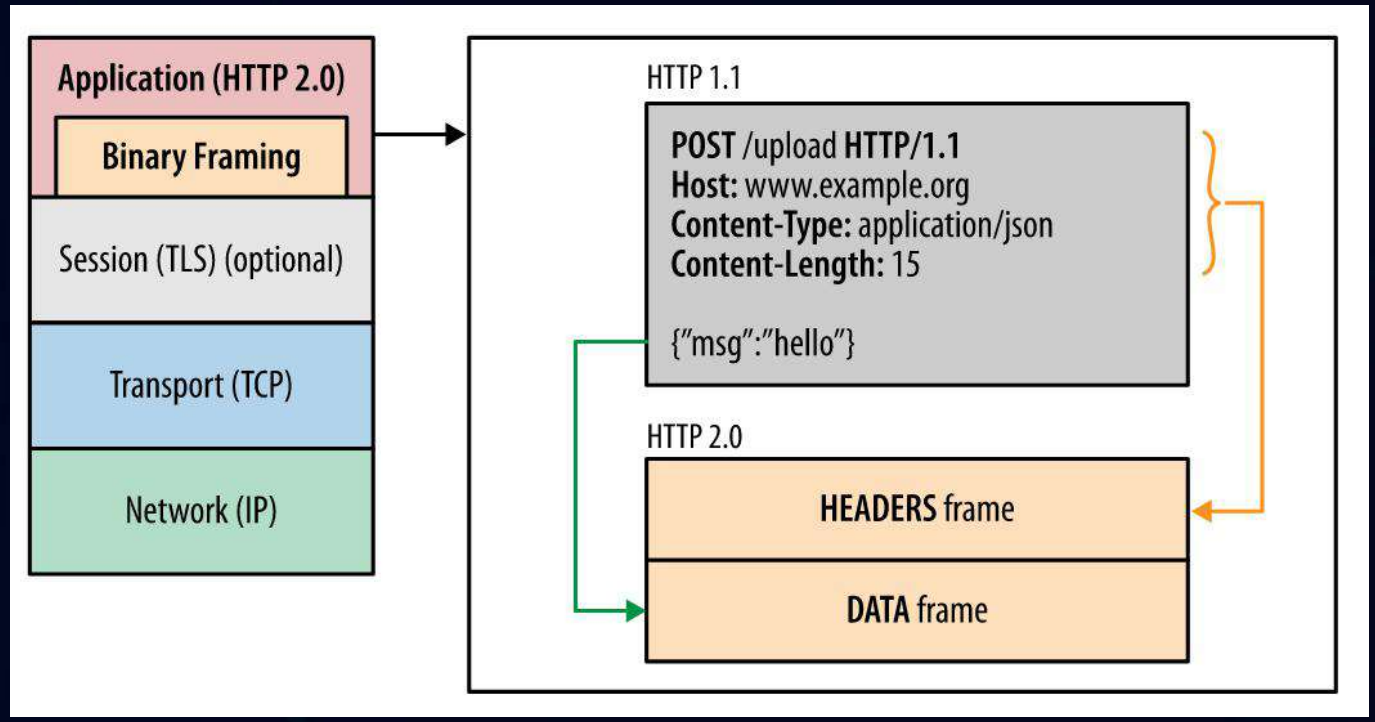


ServiceMesh & TSF 私有协议改造

HTTP/2 是一个经过实践检验的标准。

TSF团队建议，私有协议改造可参考gRPC over HTTP/2的方式，把元数据、请求的服务名放到HTTP/2 Headers里；请求参数、传输body 序列化之后放到DATA frame里。

私有基于HTTP/2，那么它的性能肯定不会是最顶尖的。但是对于大部分分布式系统来说，中庸的QPS可以接受，通用和兼容性才是最重要的事情。



ServiceMesh & TSF 接入

以 Python 应用为例说明如何改造代码来接入 TSF。Python 服务代码本身不需要修改，只需要修改服务间调用的host

- 将原来的 IP:Port 替换为服务名。
- 端口使用 80 或者业务实际端口。
- 其他代码不做修改。

管理面、数据面的监听Agent，可以通过TSF控制台，部署容器镜像 or 发布代码包（JAR/WAR等）时，一并部署，包括pilot-agent, envoy, mesh-dns等。

改造前：

```
sidecarPort = 80
if common.sendAndVerify("127.0.0.1", sidecarPort, "/api/v6/shop/items", headers):
            self.send_response(200)
            self.send_header('Content-type', 'application/json')
            self.end_headers()
            msg = {"result":{"userId":"1234", "userName":"vincent"}}
            self.wfile.write(json.dumps(msg))
        else:
            self.send_response(500)
            self.send_header('Content-type', 'application/json')
            self.end_headers()
            msg = {"exception":"Error invoke %s" % "/api/v6/shop/items"}
            self.wfile.write(json.dumps(msg))
```

改造后：

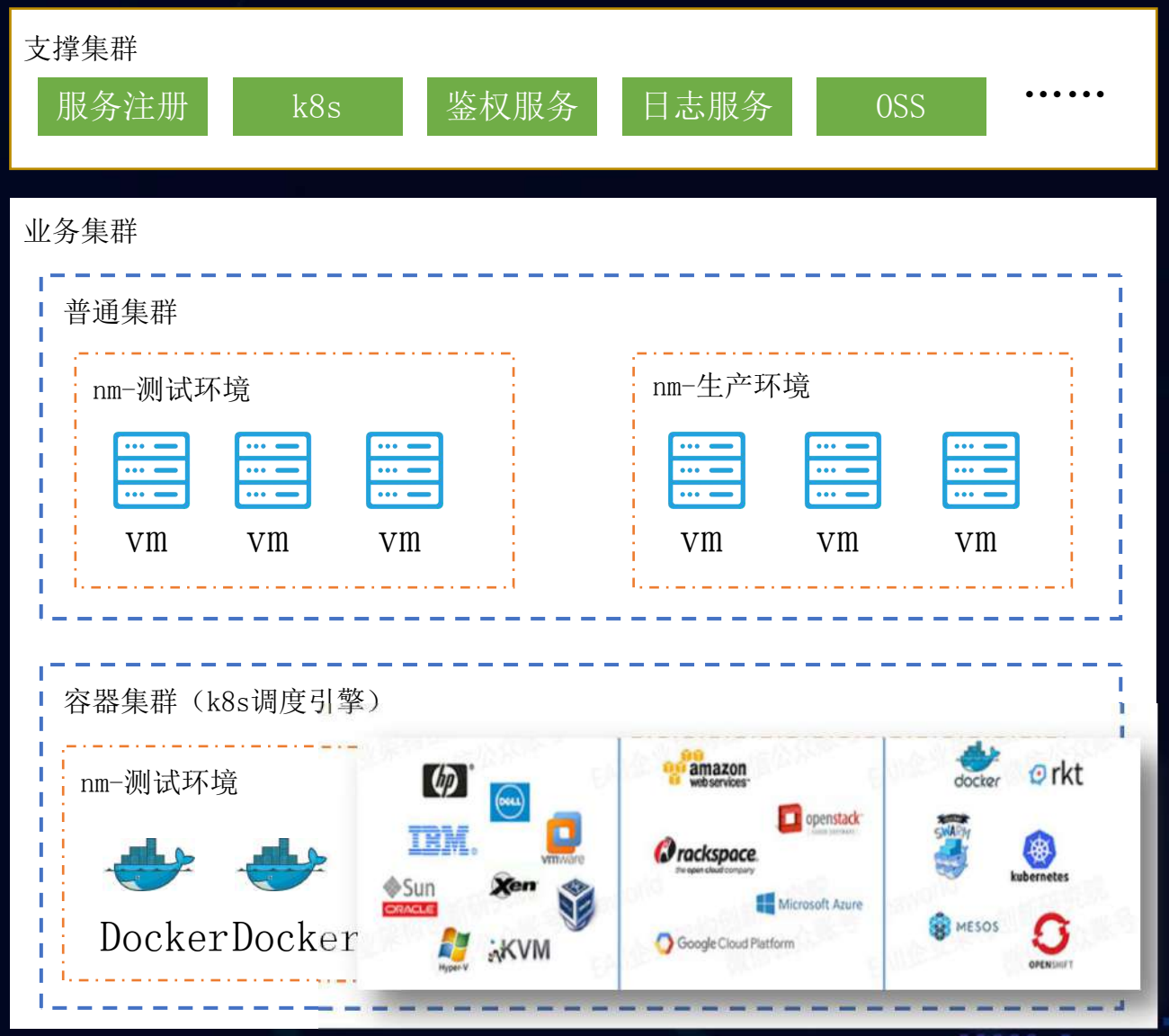
```
sidecarPort = 80
if common.sendAndVerify("shop", sidecarPort, "/api/v6/shop/items", headers):
            self.send_response(200)
            self.send_header('Content-type', 'application/json')
            self.end_headers()
            msg = {"result":{"userId":"1234", "userName":"vincent"}}
            self.wfile.write(json.dumps(msg))
        else:
            self.send_response(500)
            self.send_header('Content-type', 'application/json')
            self.end_headers()
            msg = {"exception":"Error invoke %s" % "/api/v6/shop/items"}
            self.wfile.write(json.dumps(msg))
```


new trend
new technology
new application
Cloud + community
Developer conference

VPC/命名空间，灵活组网

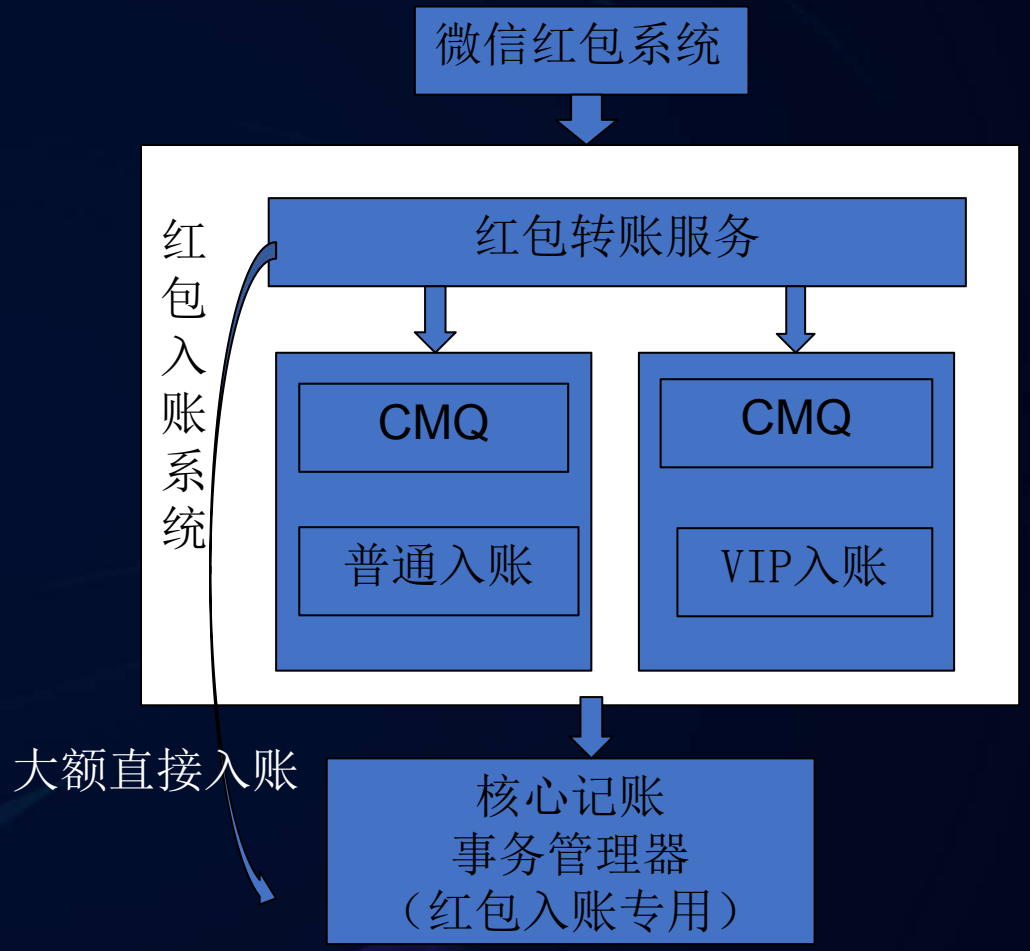
北京金融机房一区 > 开发环境、测试环境、线上环境

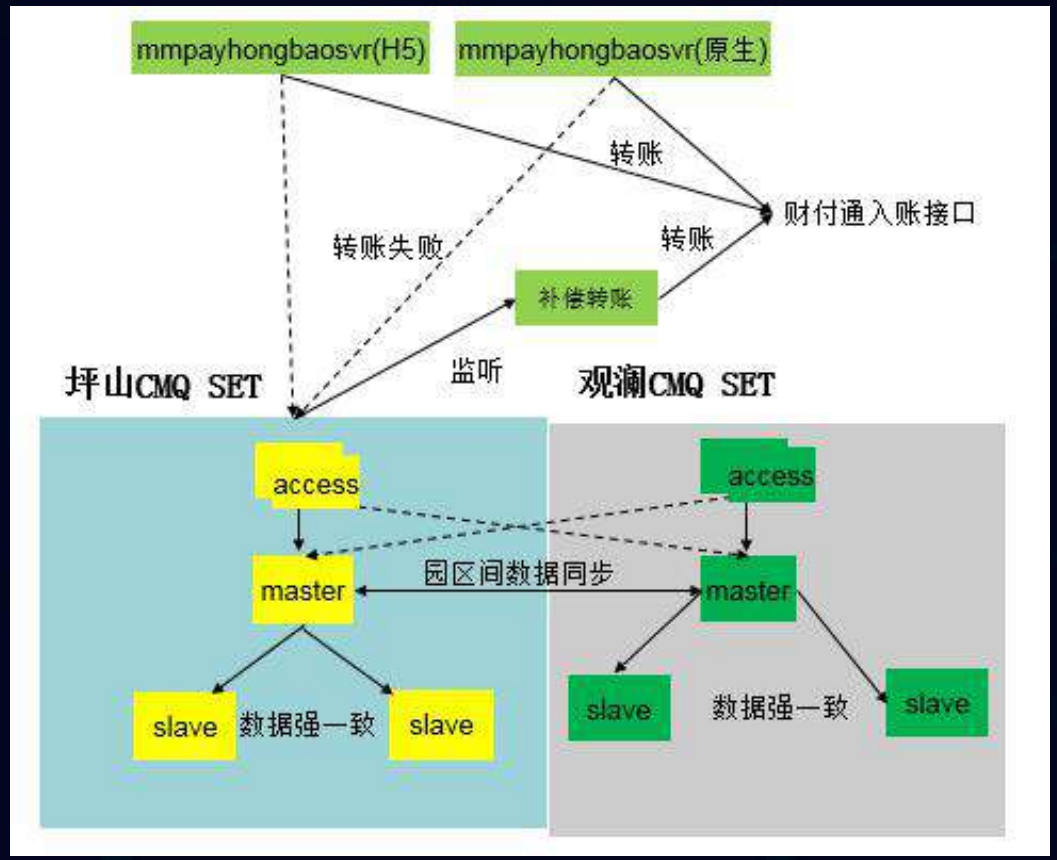
- 多集群管理+腾讯VPC私有网络，满足复杂组网，服务隔离诉求
- 命名空间，环境隔离
- 虚机、容器托管应用， 灵活选择
- 容器托管应用提高资源使用效率



可靠异步调用

消息队列适用场景：异步解耦，削峰填谷，可靠存储，海量堆积





用法一：用户抢到红包后通过CMQ异步入账

用法二：转账失败时先入CMQ，确保最终入账成功

微服务网关、API市场

对多种后端统一管理、鉴权、限流、映射、API市场发布、后端能力以API形式统一输出给多种前端进行调用等能力，满足企业发展API经济的诉求

分布式事务框架

常见的在中石化加油站付款，油费先通过中石化折扣券减免，并联动第三方银行网关、微信支付网关进行结算，如何处理事务一致性问题？

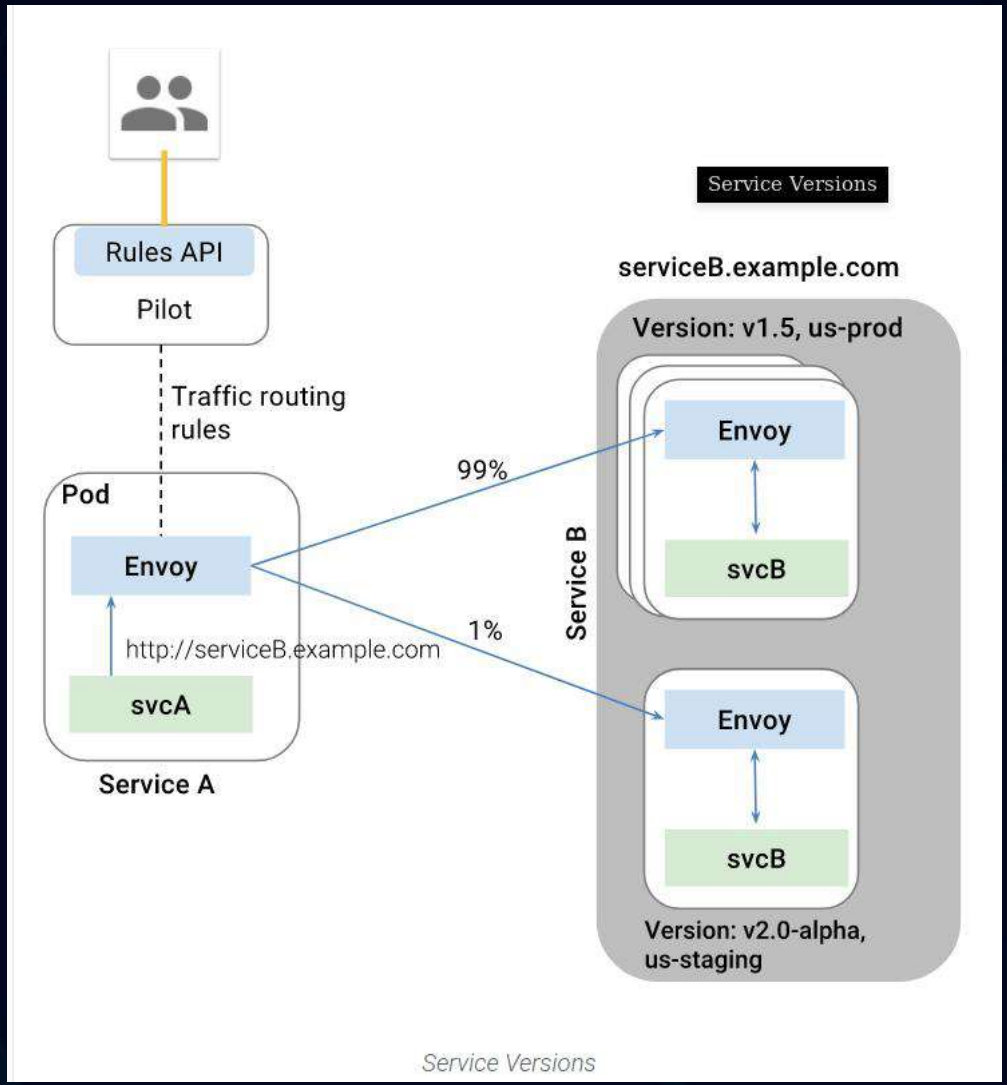

new trend
new technology
new application
Cloud + community
Developer conference

服务治理：路由、鉴权、限流、降级

服务路由使用场景：

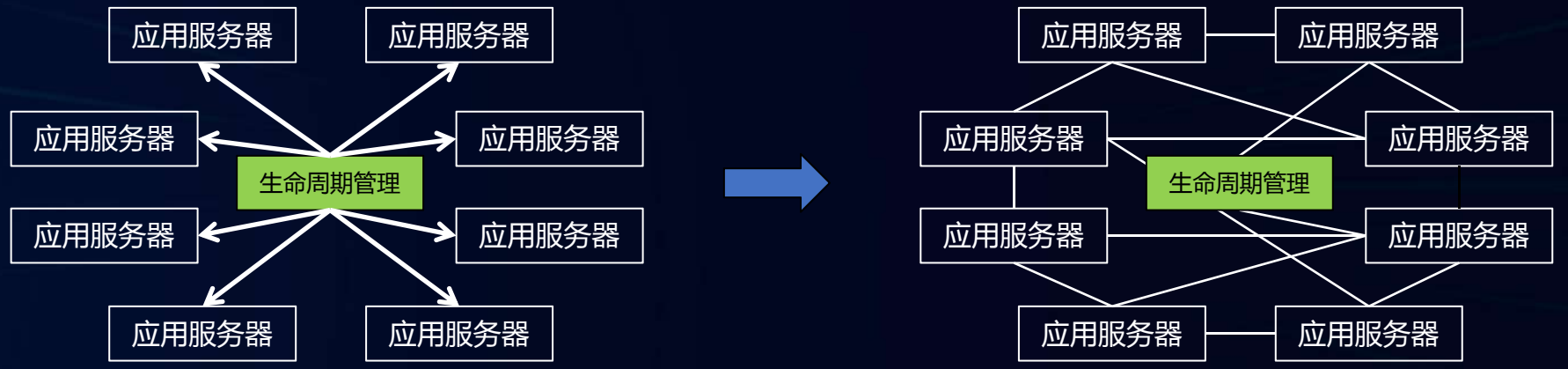
- 灰度发布功能
- 多机房路由优先调用同地机房；
- 部分用户账号内测功能；
- 保障重要服务的运行质量、实现前后端分离、读写分离等功能；

通过TSF控制台，可全局配置服务的鉴权、限流，服务降级策略。统一变更日志平台，支持快速版本/配置回滚。

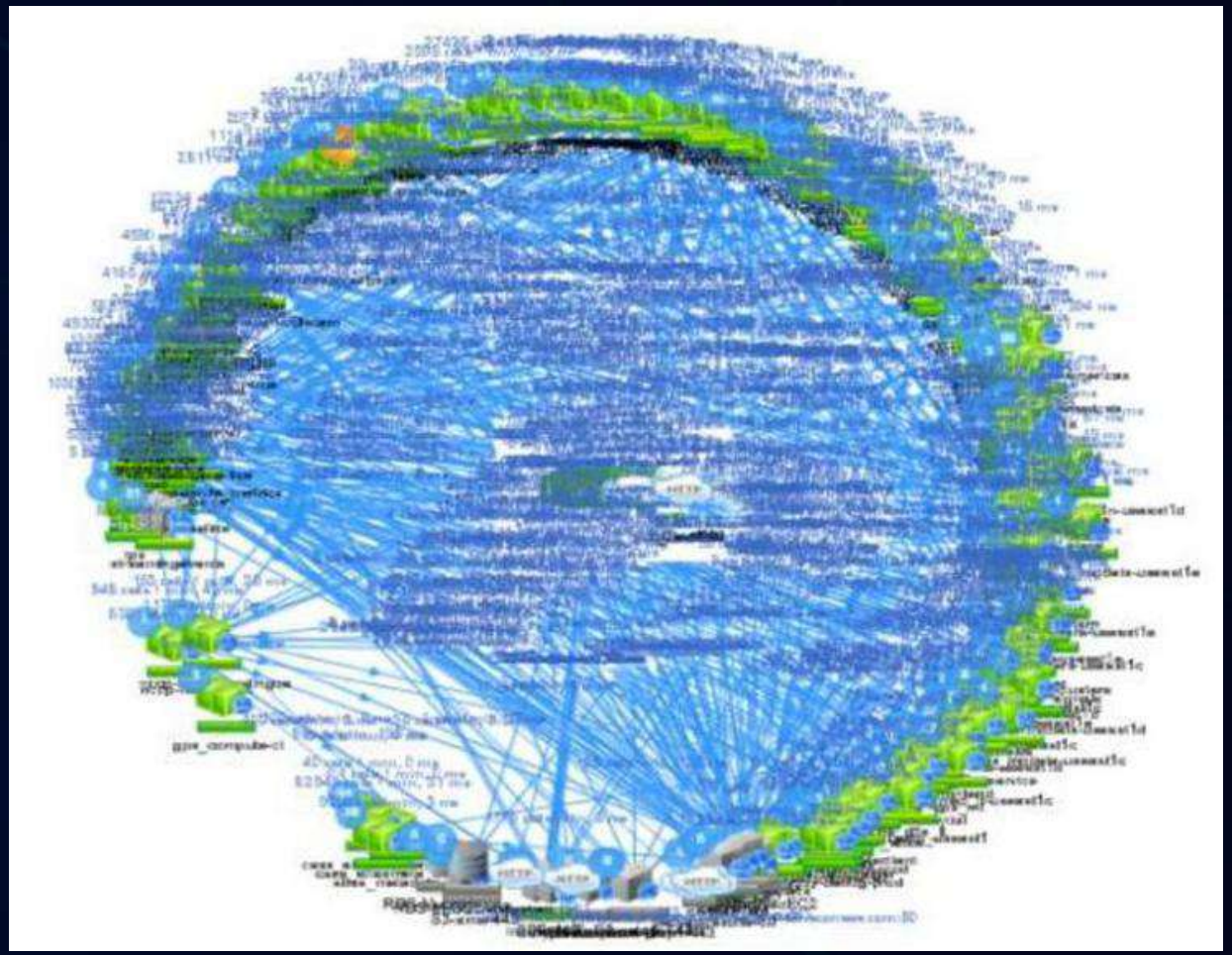


服务发布平台：千级别并发能力

应用是TSF管理的基本单位，支持Java jar包，docker镜像等应用。一个应用下面通常包含了多台机器。TSF提供了分布式应用生命周期管理机制，包括海量应用创建、部署、启动、回滚，扩容缩容和停止下线等。



分布式架构 故障定位



(Image: Bruce Wong, Netflix)

腾讯全链路Tracing

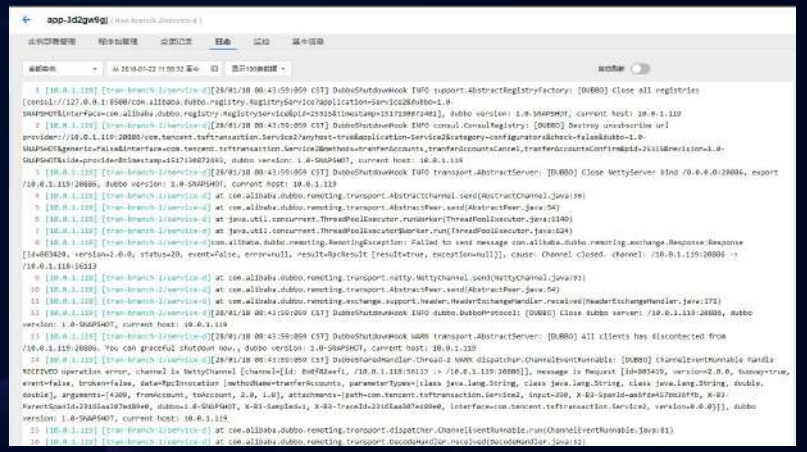
失败率高

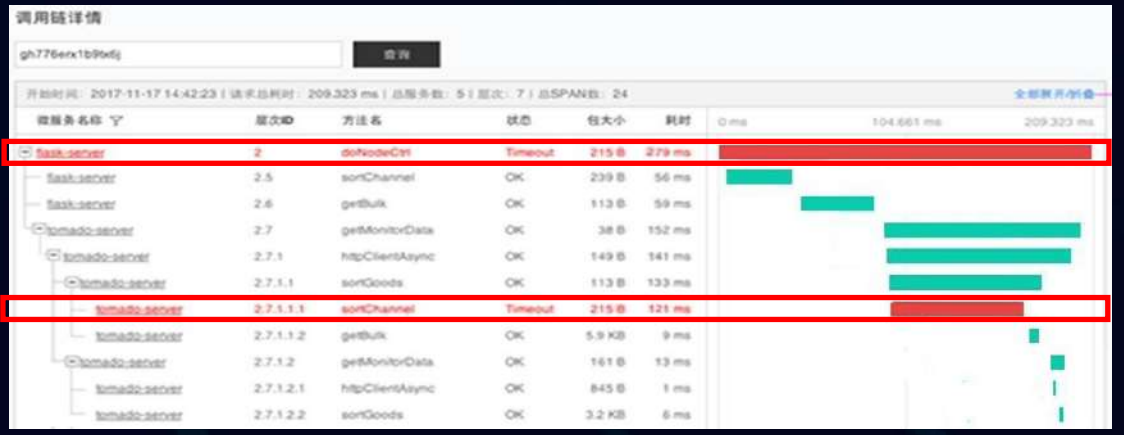
请求耗时长

JVM内存高

服务依赖拓扑图：展示服务之间复杂的调用关系

调用链：发现性能瓶颈、进行异常定位






new trend
new technology
new application
Cloud + community
Developer conference

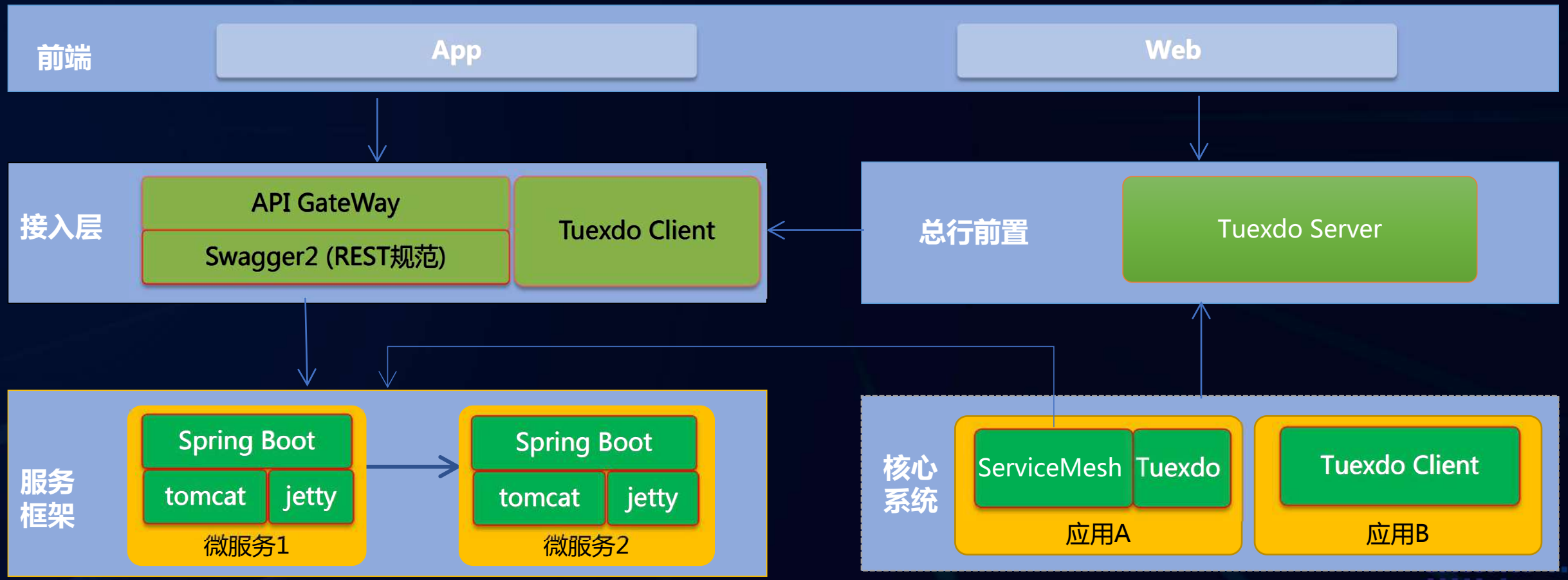
腾讯 网贷解决方案

腾讯云 | 连接智能未来

某银行网贷系统 ServiceMesh商业化落地

网贷业务：基于Service Mesh 的下一代微服务架构

Service Mesh代替旧应用注册服务，负责转发请求和响应。旧应用不需要改造，可以被微服务应用访问。仅在本地agent做流量的转发，提供细颗粒度的服务治理，每个服务的状况都能在云管理平台上感知到。



腾讯新零售打法


产品
货
=
定价
选货
商品推荐
优化商品结构
提升售卖率
智能定价
智能选品
线下导购推荐
线上千人千面
零售
场
=
Location
Experience
Management
降低管理成本
提升购物体验
选址、商圈分析
智能收银
智能巡店
店铺管理优化
用户
人
=
流量
X
转化率
X
客单价
实现精准营销
扩大钱包份额
精准引流
消费者生命周期管理
向上销售
交叉销售

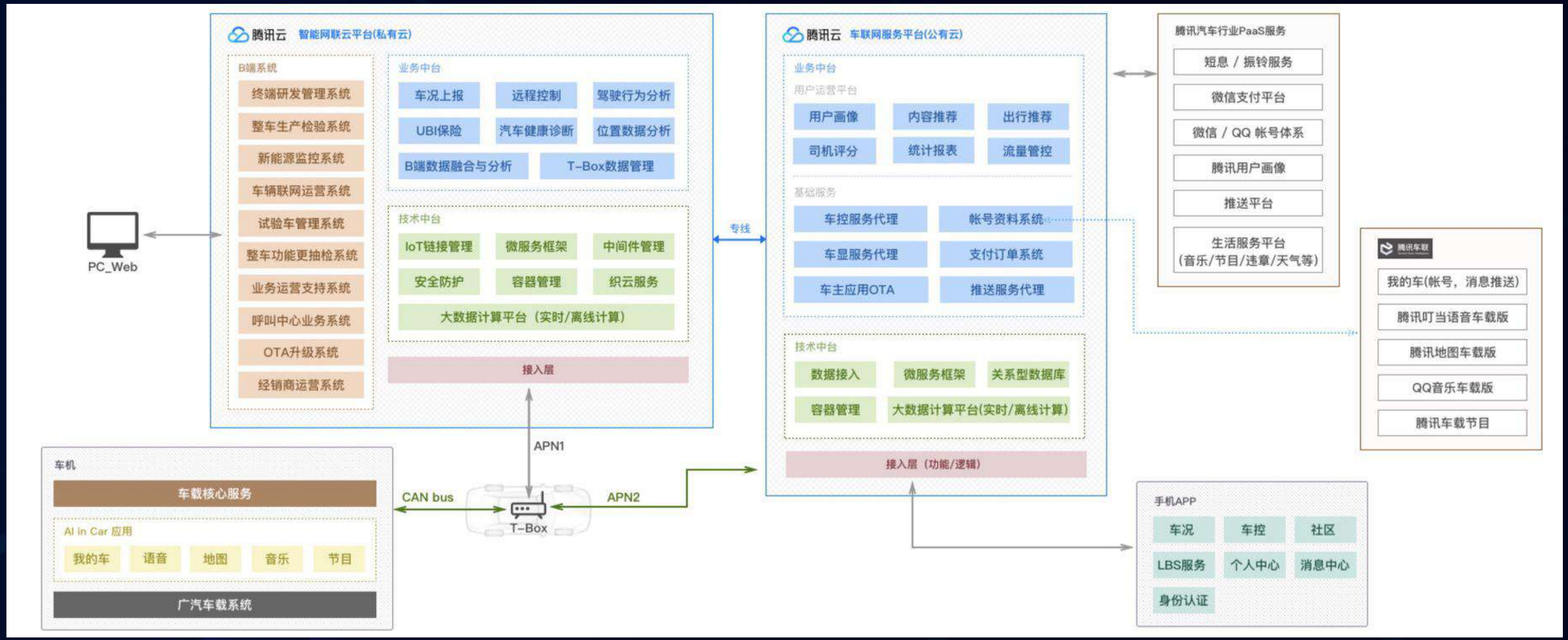
腾讯零售中台方案

一站式微服务平台，为零售业务中台提供无限扩展能力。15天快速构建 互联网+新零售应用

腾讯汽车物联网解决方案

腾讯汽车物联网解决方案

腾讯汽车物联网 技术中台




new trend
new technology
new application
Cloud + community
Developer conference