ArchSummit
全球架构师峰会 2017

主办方 Geekbang 极客邦科技 InfoQ

Service Mesh在华为公有云的实践

田晓亮
华为架构师

SPEAKER
INTRODUCE

田晓亮

华为 架构师

8年软件行业经验，曾就职于三星，2012年进入云计算领域，对PaaS，DevOps，APM有深入的研究和实践经验。方案支撑近千台VM中应用部署管理监控

2016年加入华为担任架构师，负责微服务的Go语言开发框架及Service Mesh设计和落地，Go语言微服务框架被华为5G核心网络采用，Service Mesh服务商用上线公有云

TABLE OF

CONTENTS 大纲

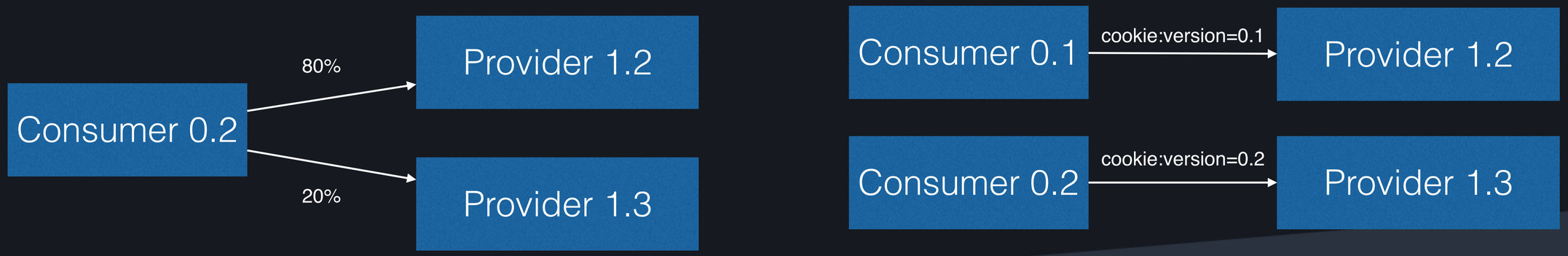
- 简介
- Service Mesh在华为内部的技术演进
- 实现一个Service Mesh
- 使用Service Mesh快速构建微服务
- 生态与扩展

我们在构建微服务，而构建微服务是困难的

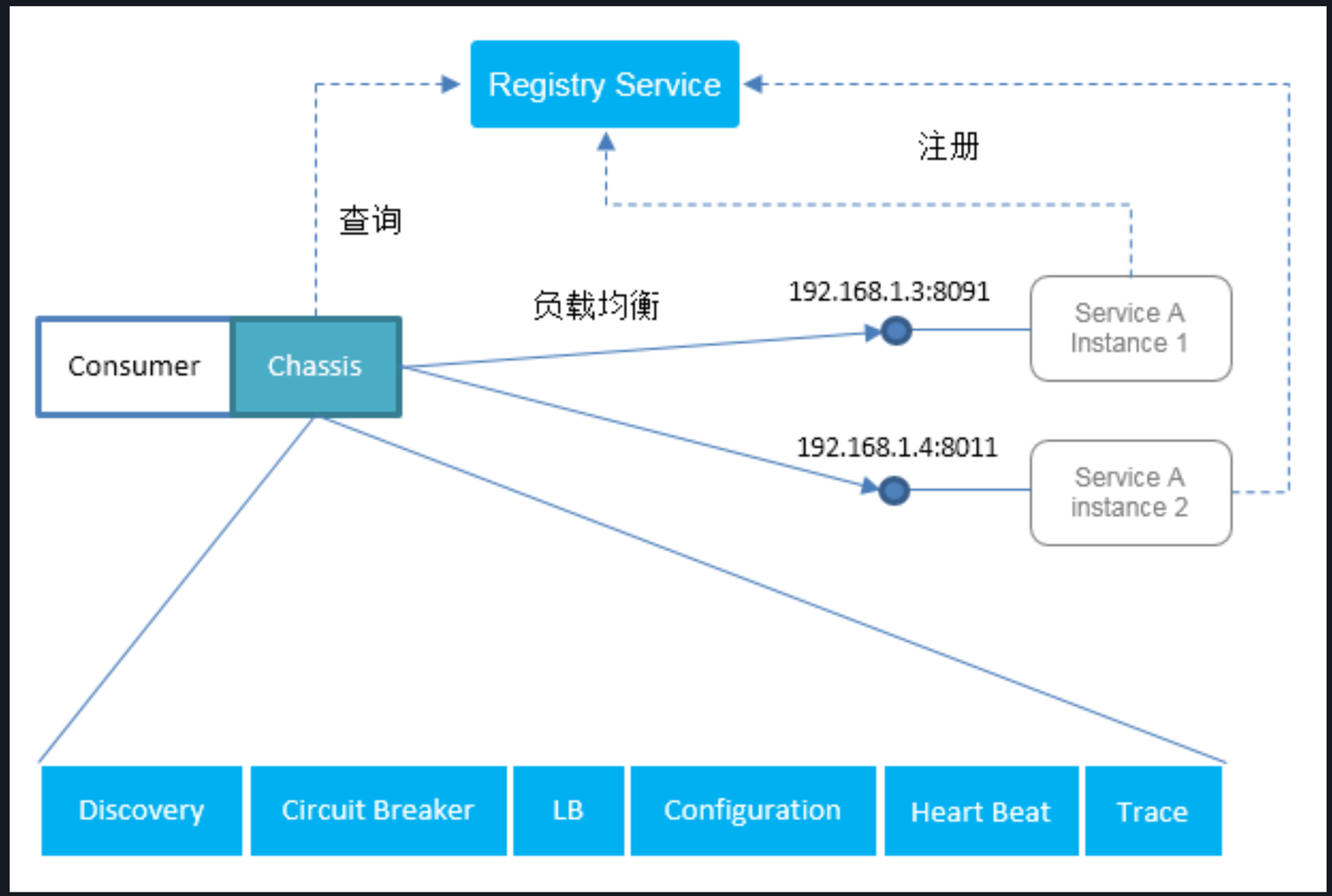
微服务架构模式引入的问题

路由规则

- 引流
- 支持权重
- 根据消费者信息或请求特征进行引流



Chassis


Registry Service
注册
查询
负载均衡
192.168.1.3:8091
Service A
Instance 1
Consumer
Chassis
192.168.1.4:8011
Service A
instance 2
Discovery
Circuit Breaker
LB
Configuration
Heart Beat
Trace

语言开发框架

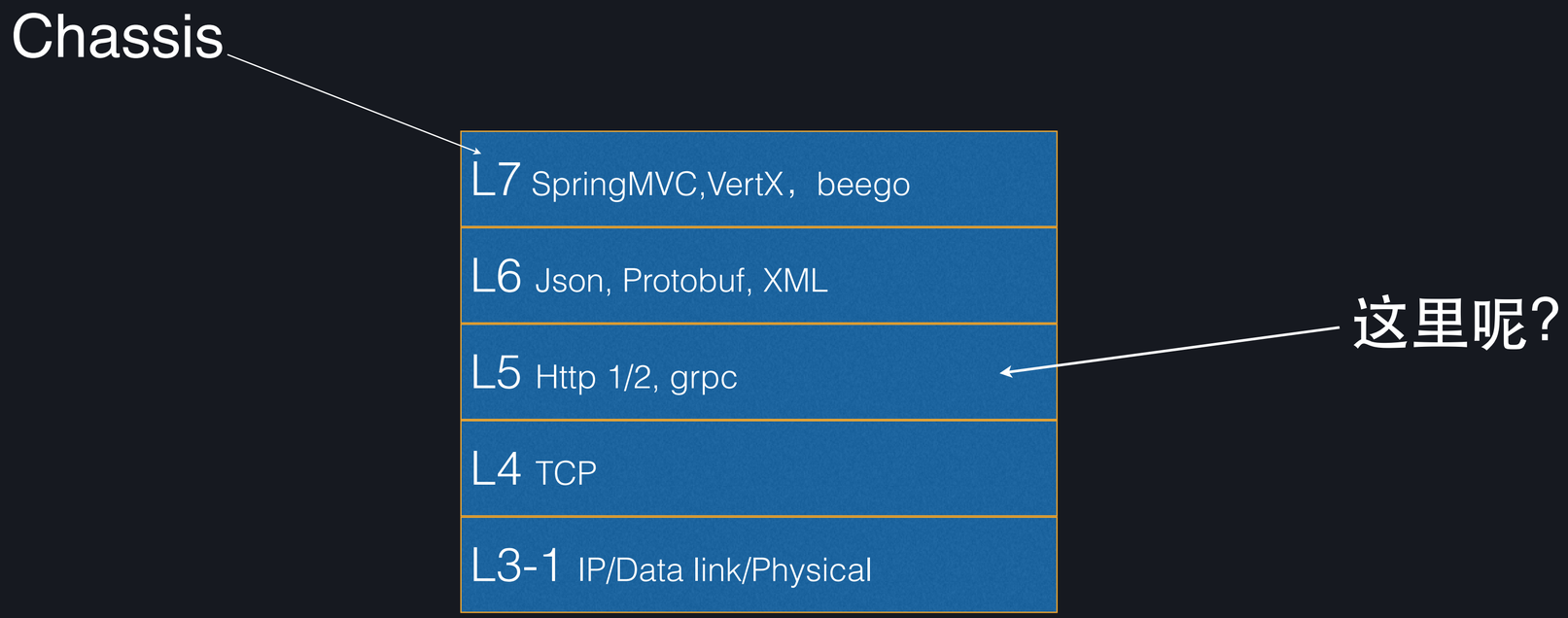
Go-kit

Go-micro

还有其他解题思路么?

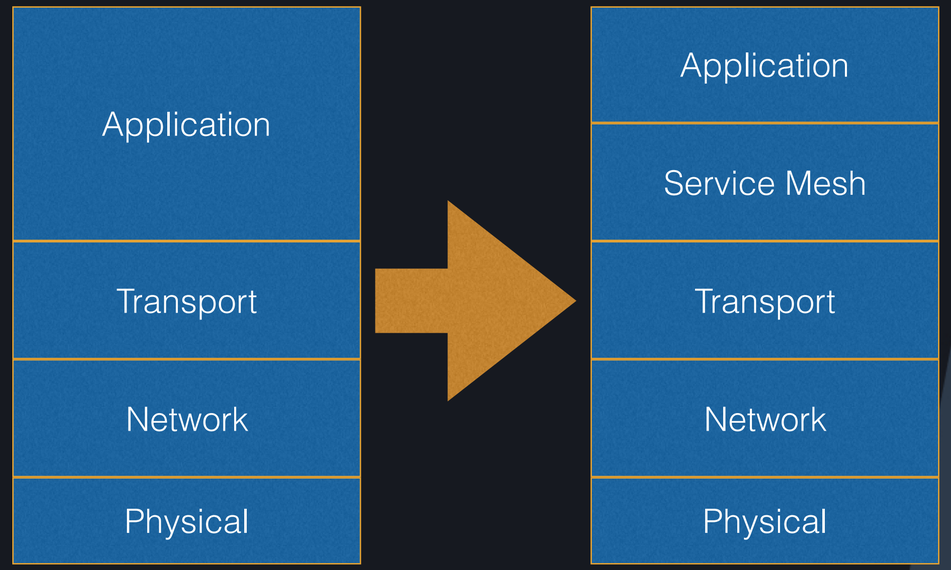
- 多语言SDK支持
- 学习曲线
- 绑定特定技术栈，不容易切换方案
- 老旧单体应用

抽象


Chassis
L7 SpringMVC,VertX，beego
L6 Json, Protobuf, XML
L5 Http 1/2, grpc
L4 TCP
L3-1 IP/Data link/Physical
这里呢?

Service Mesh

- 一种基础设施层，服务间的通信通过Service Mesh进行
- 可靠地传输复杂网络拓扑中服务的请求，将服务变成现代的云原生服务
- 一种网络代理的实现，通常与业务服务部署在一起，业务服务不感知
- 一种TCP/IP之上的网络模型



为什么使用Service Mesh

- 无需多种语言的微服务框架开发
- 对业务代码0侵入
- 不适合改造的单体应用
- 开发出开的应用既是云原生的又具有独立性

没有银弹

- 新的故障点
- 一定程度的性能降低
- 侵入式框架有更强的定制和扩展能力
- 部署复杂性

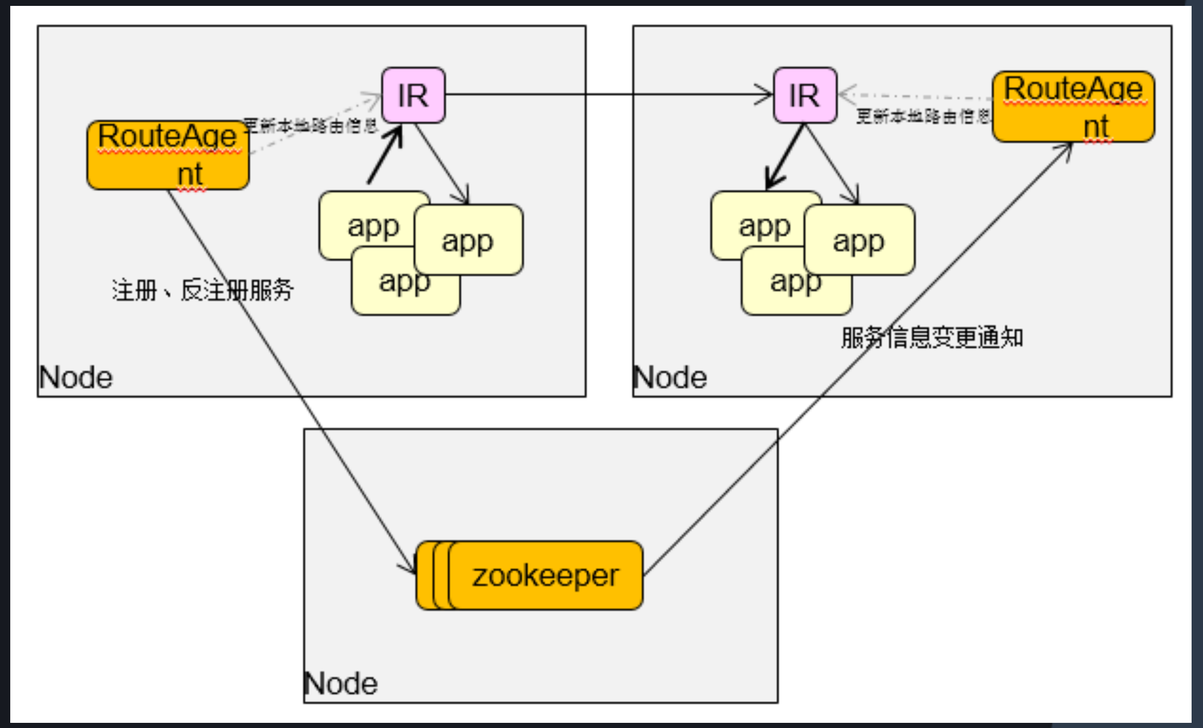
TABLE OF

CONTENTS 大纲

- 简介
- Service Mesh在华为内部的技术演进
- 实现一个Service Mesh
- 使用Service Mesh快速构建微服务
- 生态与扩展

第一代：基于Nginx的微服务代理

- 2013年,微服务开发平台中的组件，公司内部某电信业务
- 400多个微服务，800左右实例，200多个数据中心



Sidecar模式

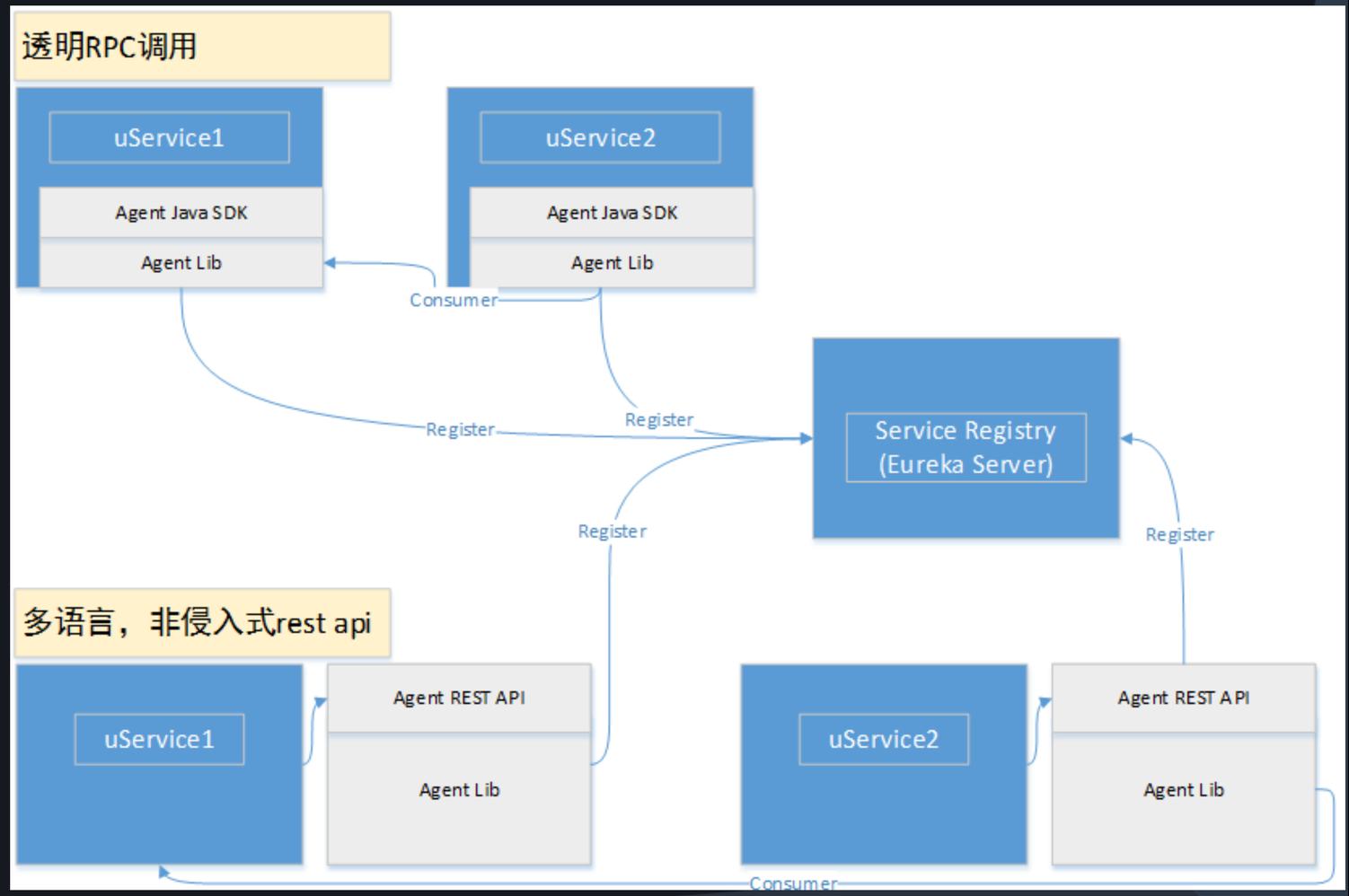
- 基于容器的分布式系统设计模式
- 容器可以共享存储与网络的能力

Sidecar优势

- 以容器作为基础打包单元，可以分给不同的团队进行开发测试
- 可重用
- 以容器作为错误边界，使服务能够正确推出
- 独立回滚与更新

第二代：HSA SideCar

- 2016年，基于Java 微服务框架开发
- 提供SOAP协议转换
- 与微服务部署在一个Pod
- 占用资源很高



Mesher

- Service Mesh的一种实现
- 基于自研的Go语言微服务框架（即将开源）开发
- 使用CSE注册中心与配置中心
- 以Sidecar的方式部署在微服务所运行的环境中，也可以PerHost模式运行
- 支持多种部署方式
- 占用资源小（闲置10多M，并发运行时30多M）

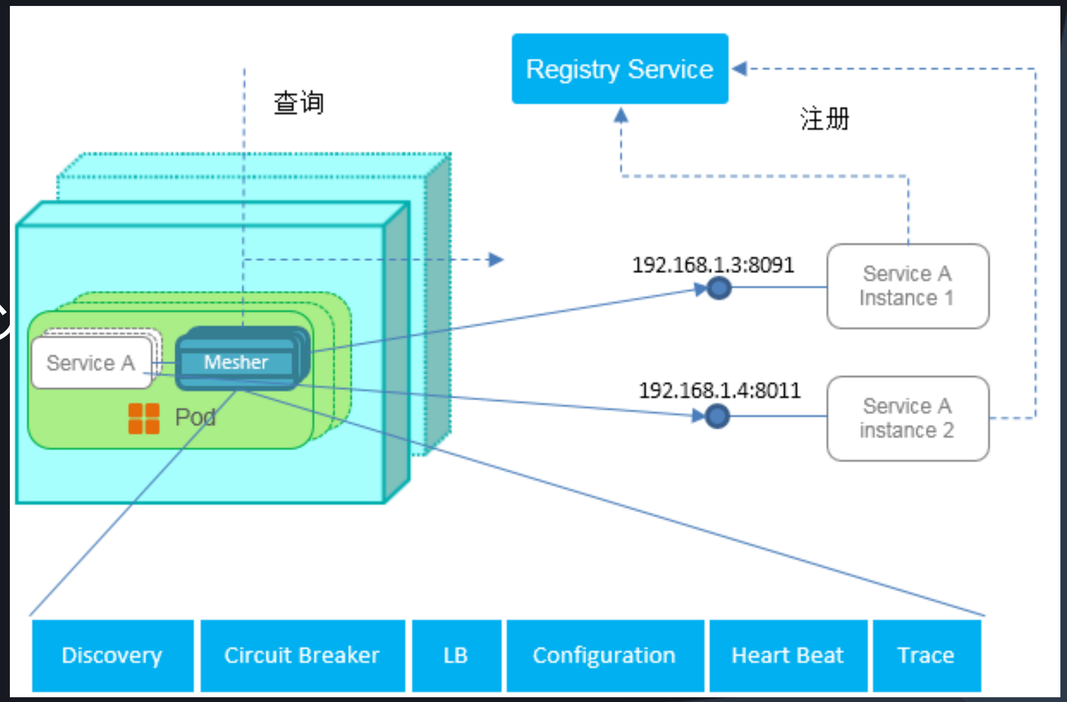
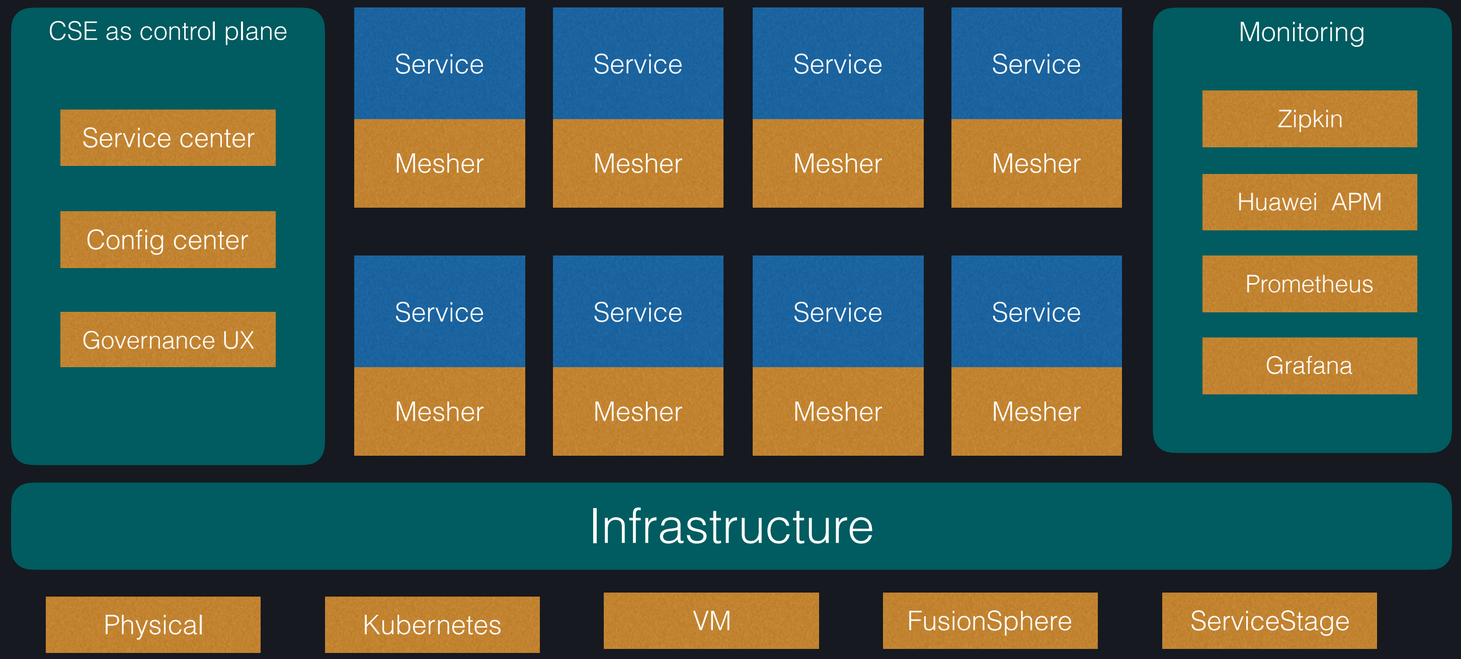


TABLE OF
CONTENTS 大纲

- 简介
- Service Mesh在华为内部的技术演进
- 实现一个Service Mesh
- 使用Service Mesh快速构建微服务
- 生态与扩展
- 与PaaS平台结合

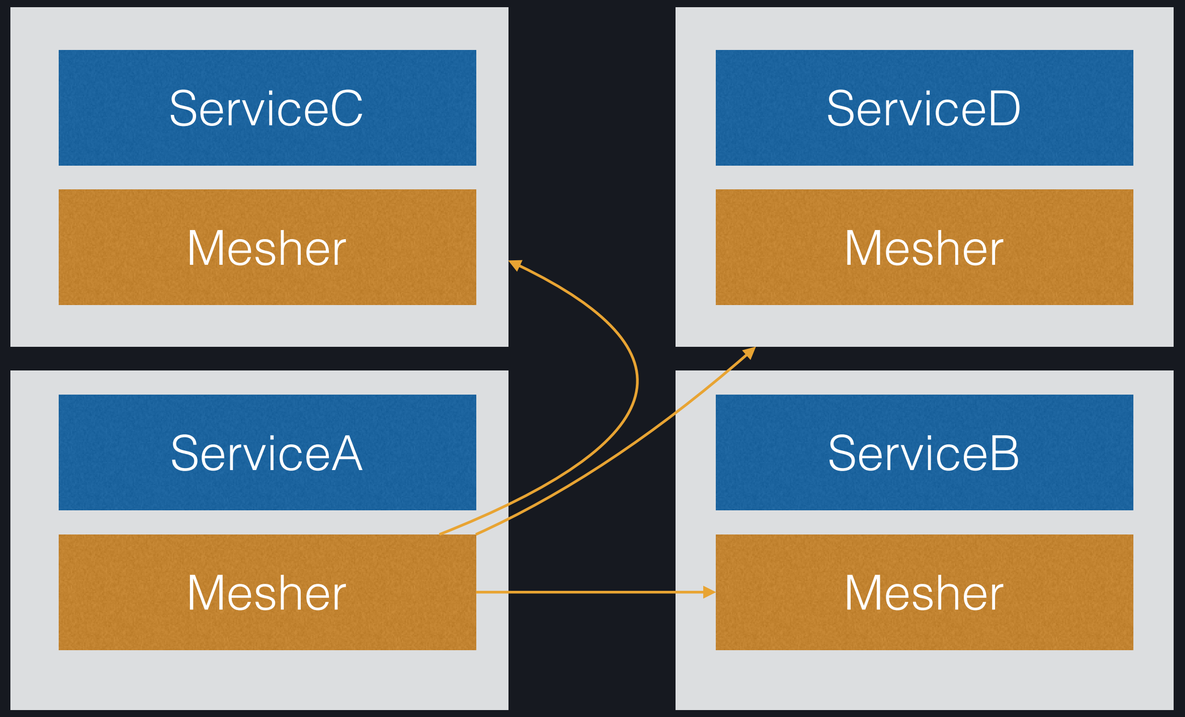
整体架构


CSE as control plane
Service center
Config center
Governance UX
Service
Mesher
Service
Mesher
Service
Mesher
Service
Mesher
Service
Mesher
Service
Mesher
Service
Mesher
Service
Mesher
Monitoring
Zipkin
Huawei APM
Prometheus
Grafana
Infrastructure
Physical
Kubernetes
VM
FusionSphere
ServiceStage

Design Goal

- 尽可能提供定制化
- 减少对业务影响
- 高性能
- 使服务可视化
- 使服务更具弹性

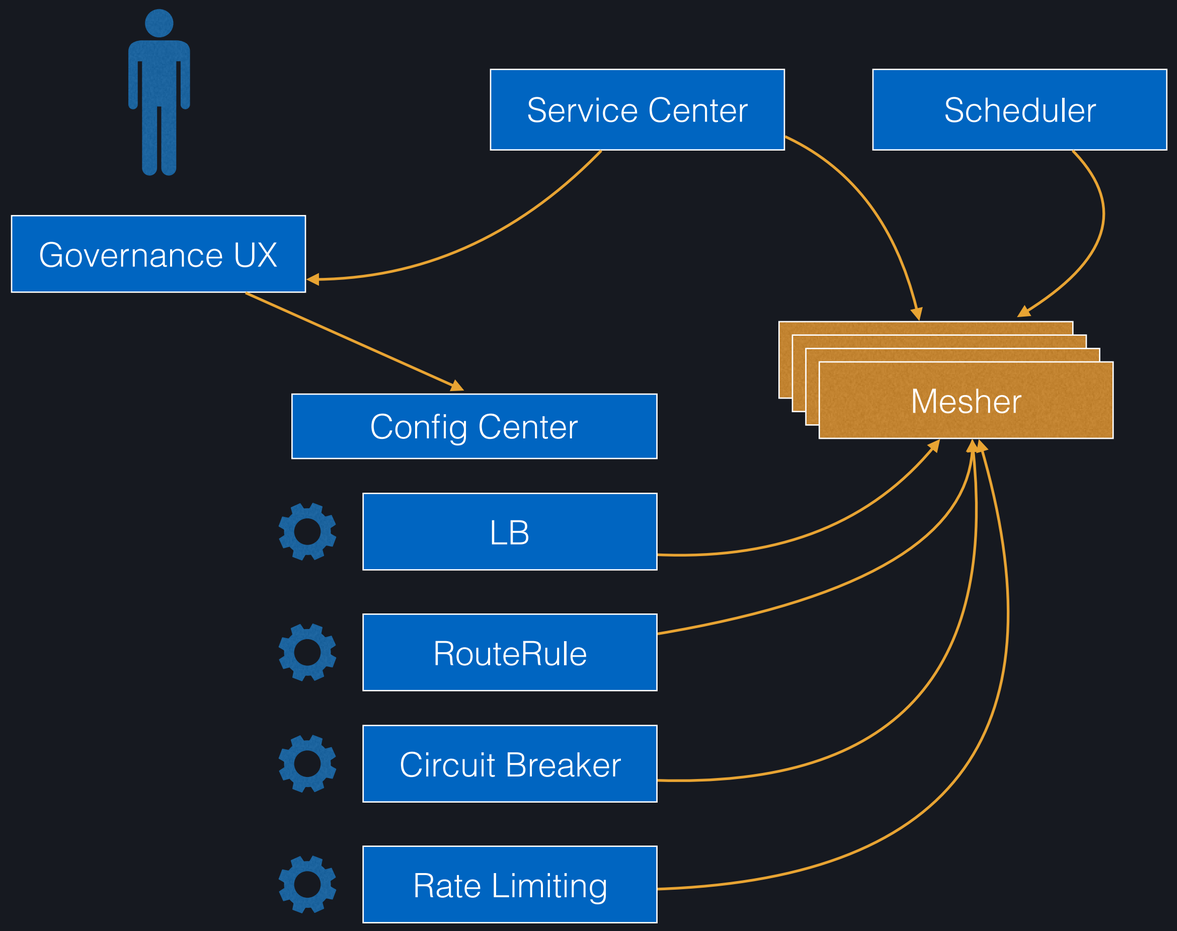
Data Plane



即Mesher组件本身，对所有请求进行处理

- 注册发现服务
- 执行路由策略
- 负载均衡
- 透明TLS传输
- 生成监控数据

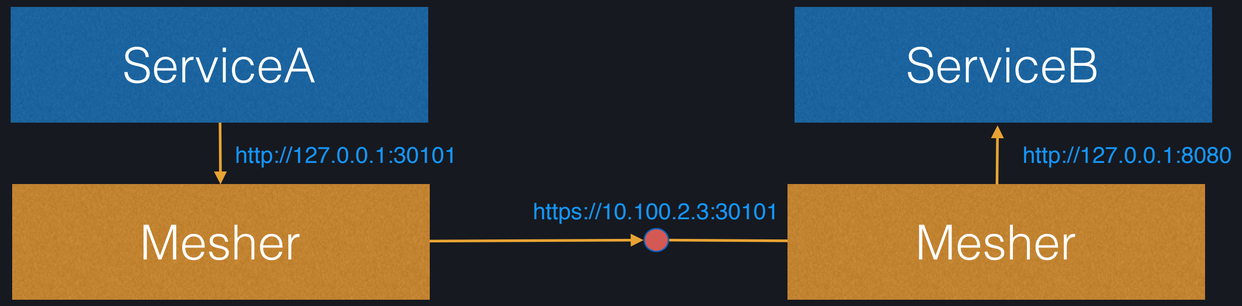
Control Plane



可人工介入，为所有运行的Mesher提供配置下发，不会碰服务请求

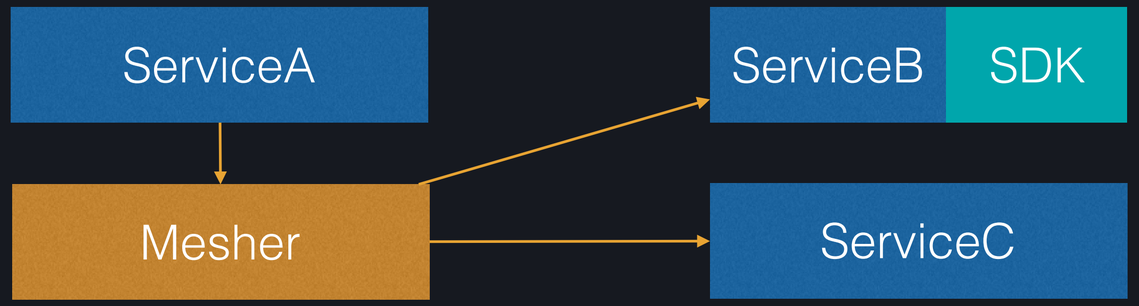
- 注册中心
- 下发配置
- 对接监控服务
- 调度引擎

调用场景1



- 透明TLS传输能力，用户可托管自己的证书
- 提供者可获得服务端限流能力和处理链功能

调用场景2

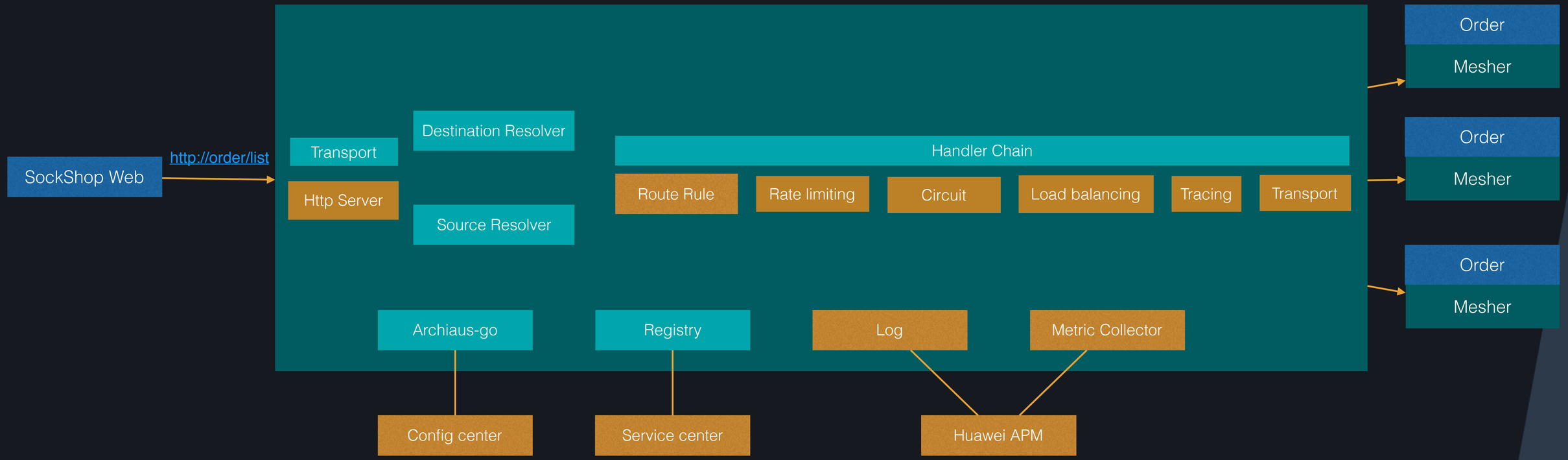


- 消费者与Mesher一起部署
- 提供者为Java或者Go语言微服务框架开发
- 提供者为普通的业务应用

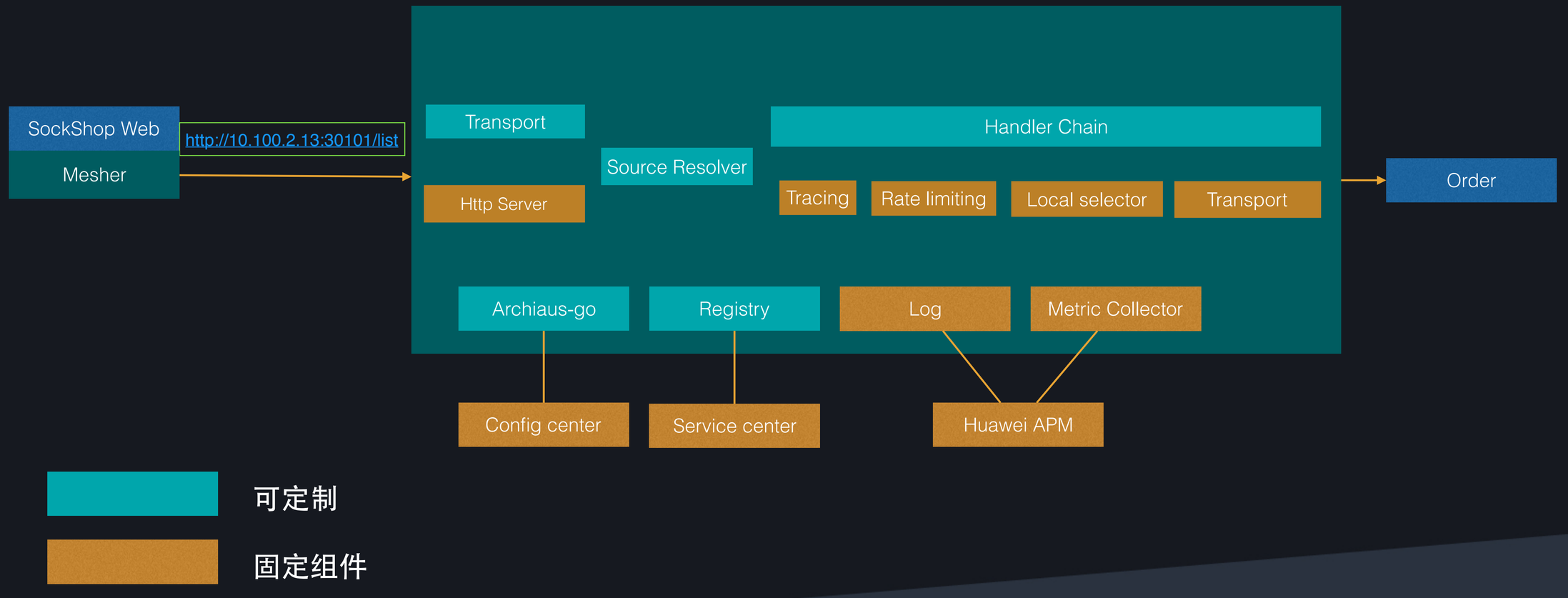
调用场景3

- 提供者都与Mesher部署
- 消费者为Java或者Go语言微服务框架开发
- 消费者不使用框架

接受本地请求

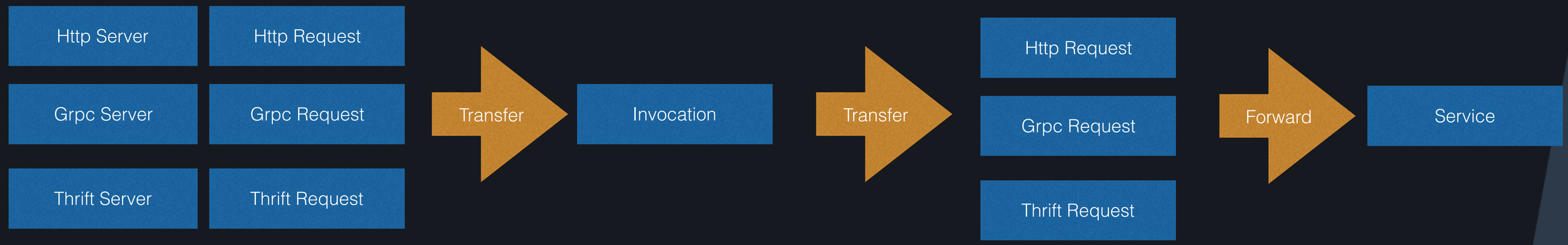

SockShop Web
http://order/list
Transport
Http Server
Destination Resolver
Source Resolver
Handler Chain
Route Rule
Rate limiting
Circuit
Load balancing
Tracing
Transport
Archiaus-go
Registry
Log
Metric Collector
Config center
Service center
Huawei APM
Order
Mesher
Order
Mesher
Order
Mesher
可定制
固定组件

接受远程请求


SockShop Web
Mesher
http://10.100.2.13:30101/list
Transport
Http Server
Source Resolver
Handler Chain
Tracing
Rate limiting
Local selector
Transport
Order
Archiaus-go
Registry
Log
Metric Collector
Config center
Service center
Huawei APM
可定制
固定组件

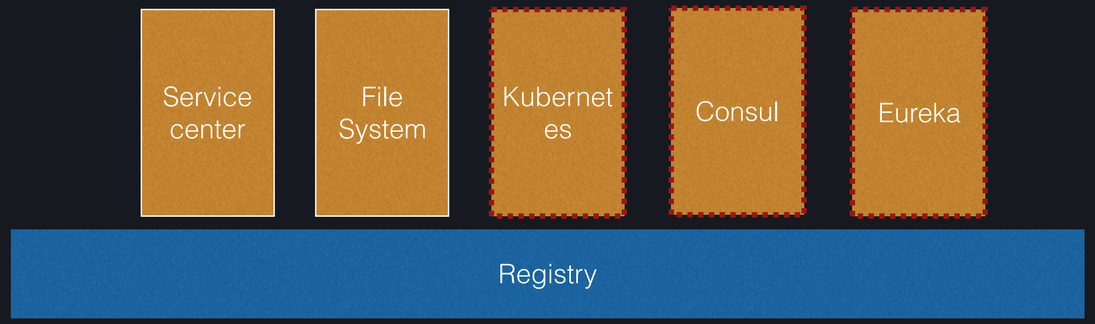
多协议支持

- 任意协议请求都被抽象为Invocation模型进行处理，最终再转换为协议请求转发



Registry

- 负责发现注册
- 插件化注册中心客户端

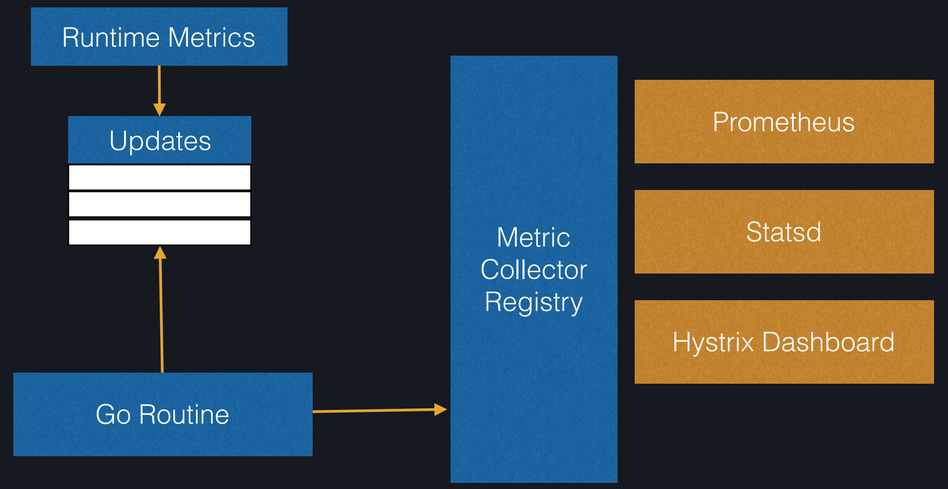


动态治理

- 对接华为公有云微服务引擎的Service center 与 Config center

监控

- 处理链中提供分布式调用追踪
- Admin API 提供了路由查询，健康检查，Metric等信息
- 使用Collector Registry模型，并使用Channel主动上报Metric数据
- Promethues，Grafana，Zipkin对接



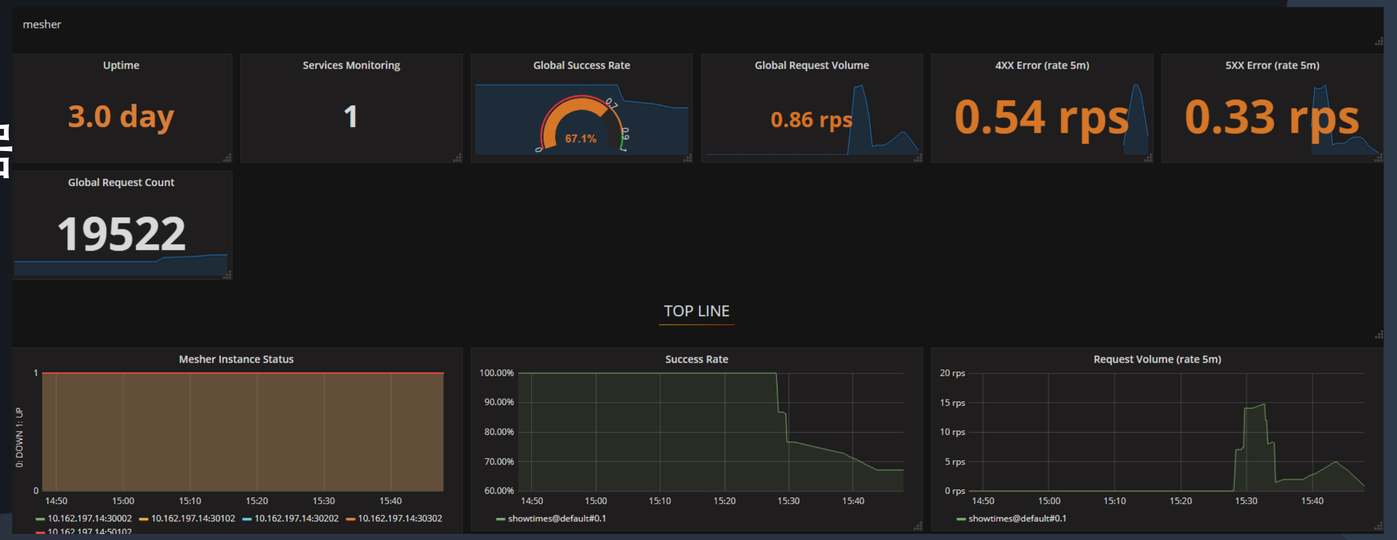


TABLE OF

CONTENTS 大纲

- 简介
- Service Mesh在华为内部的技术演进
- 实现一个Service Mesh
- 使用Service Mesh快速构建微服务
- 生态与扩展

部署

- 本地或虚机部署: 样例
- kubernetes部署
- ServiceStage部署

最小化配置项

- HTTP_PROXY
- SERVICE_NAME
- PROVIDER_ADDR
- CSE_REGISTRY_ADDR

部署复杂性提升，如何消弭?

- Infrastructure as Code—开发者自行选型
- 命令行注入工具—Mesher开发团队定制
- 编排服务自动注入—最完整的产品体验

应用与Mesher集成最佳实践

- 使用容器,容器网络与编排调度平台
- 业务代码监听在本地端口，并且不要暴露业务服务端口。使用Mesher端口对外提供服务
- 使用SideCar部署模式
- 将服务提供者地址写在配置文件中，在本地里信息是http://x.x.x.x:port,当需要与Mesher使用时，改为http://ServiceName

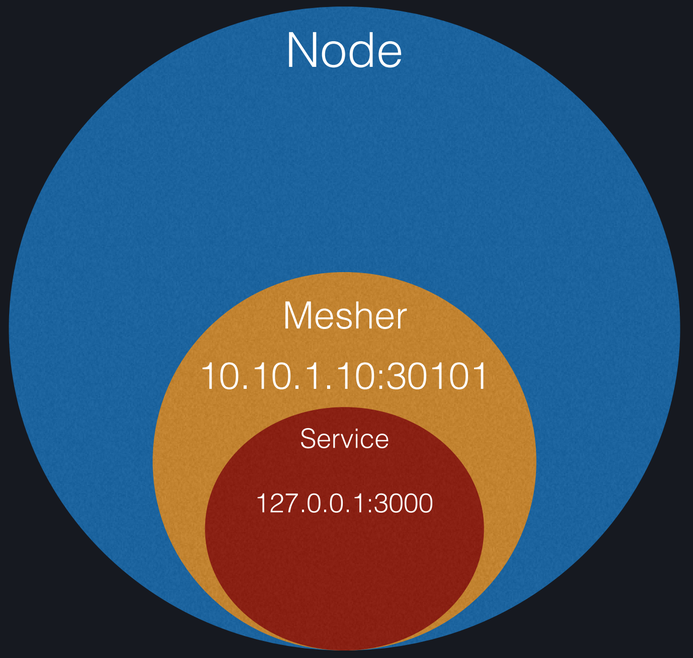


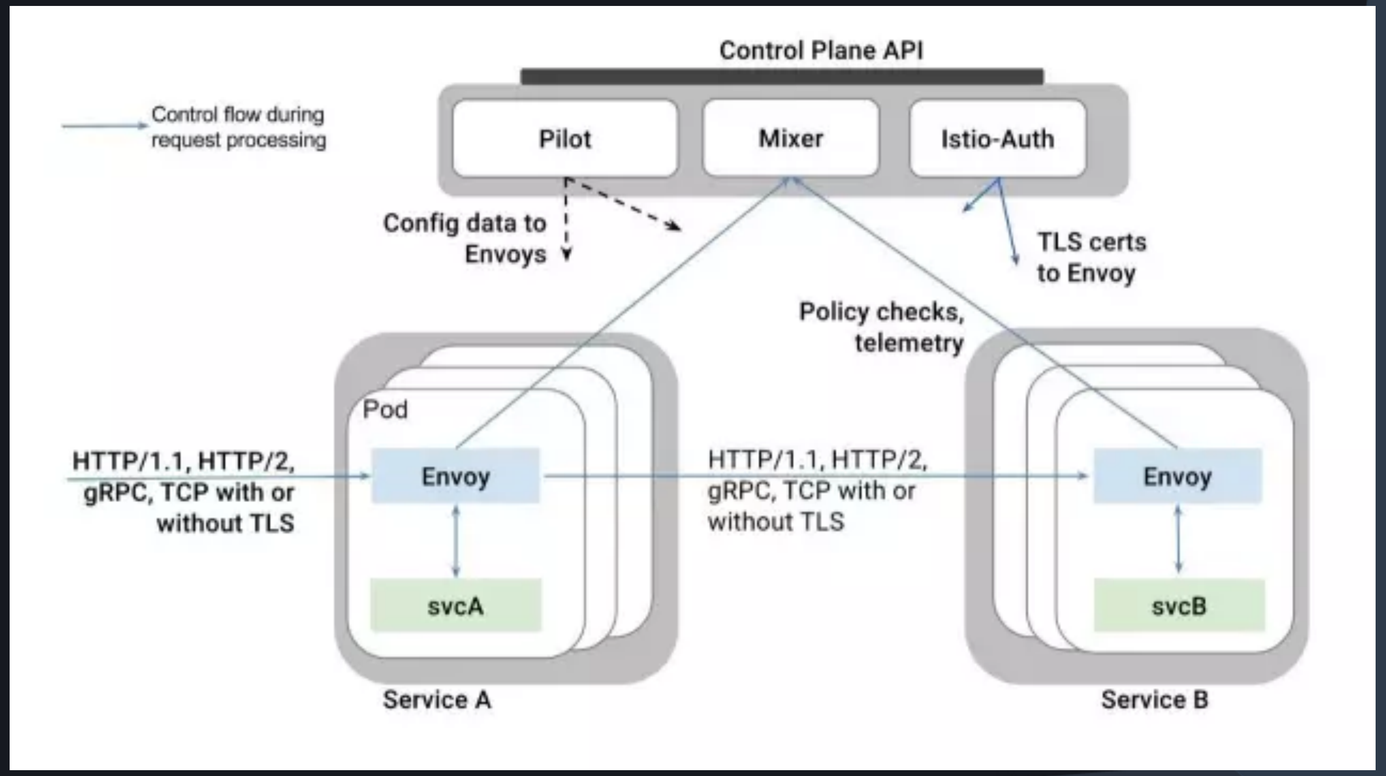
TABLE OF

CONTENTS 大纲

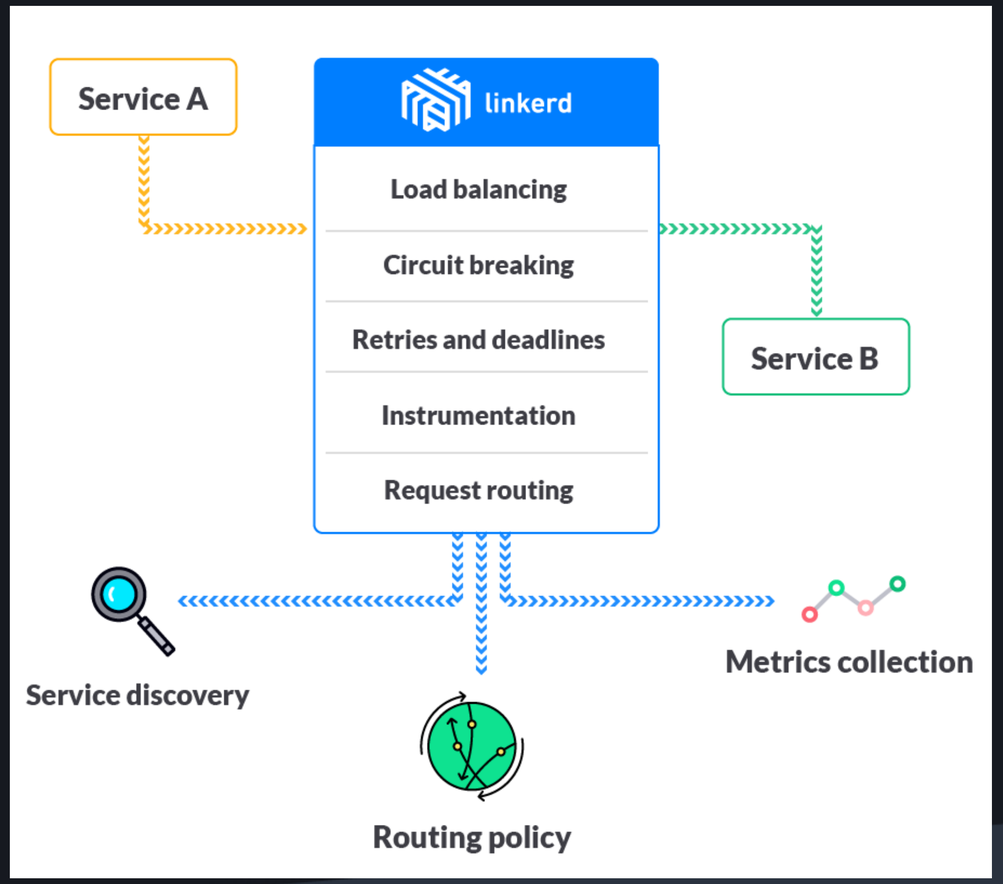
- 简介
- Service Mesh在华为内部的技术演进
- 实现一个Service Mesh
- 使用Service Mesh快速构建微服务
- 生态与扩展

Istio与Envoy

- 作为一个平台，专注在控制面的功能，将Envoy整合到平台中，作为数据面Service Mesh
- Mesher可以作为Envoy的一种替代方案



Linkerd


Service A
linkerd
Load balancing
Circuit breaking
Retries and deadlines
Instrumentation
Request routing
Service B
Service discovery
Metrics collection
Routing policy

Service Mesh Landscape

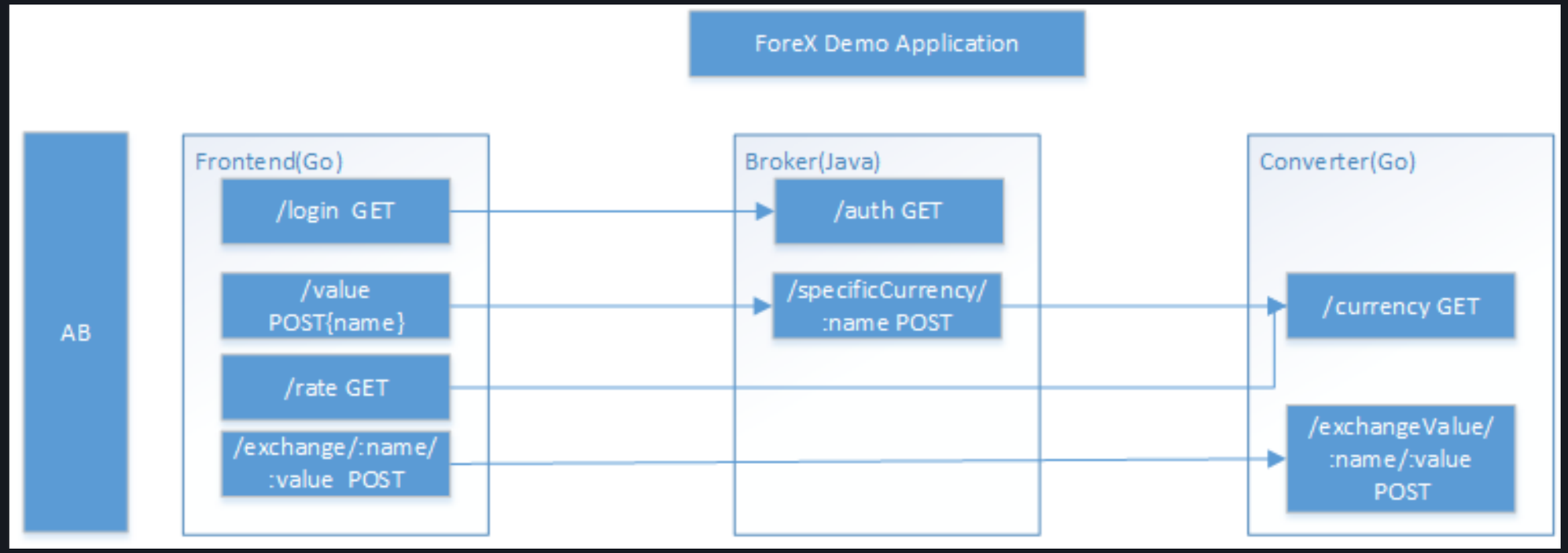
- Data Plane：Linkerd，Nginx，Envoy, Mesher
- Control Plane：Istio, CSE

性能对比

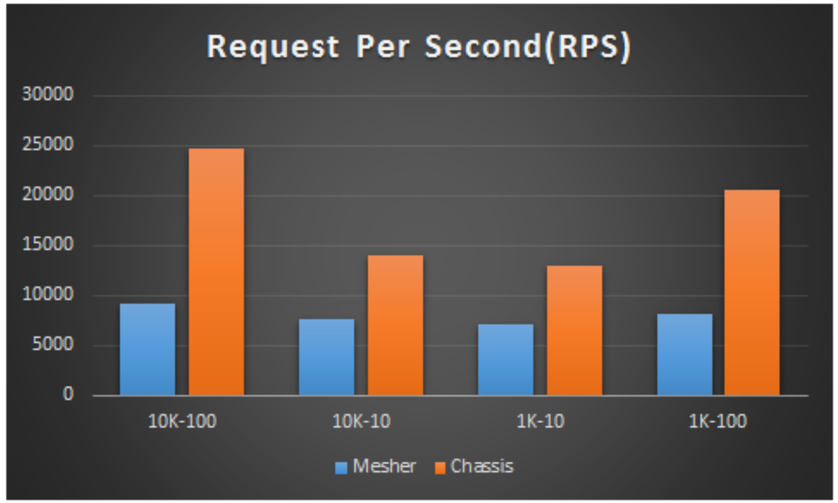
4U8G

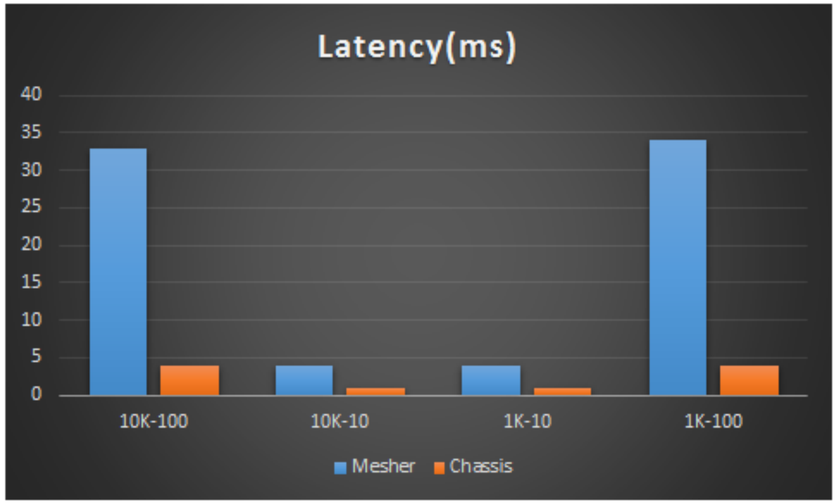
	Mesher1.0	Istio 0.1.6（Envoy）	Linkerd1.1.3
Latency(ms)	17.2	465	34.7
TPS	8264	388	4235
CPU	300%	170%	310%
Memory	35M	20M	196M
并发	20	20	20

与ServiceComb侵入式框架对比

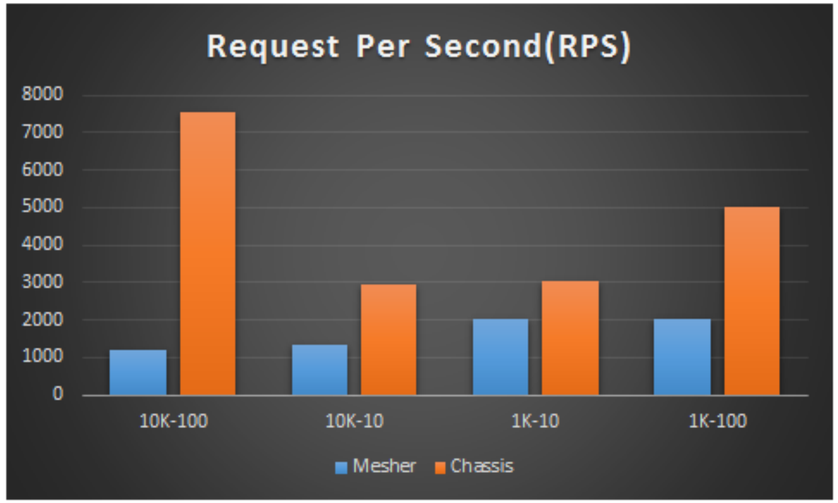


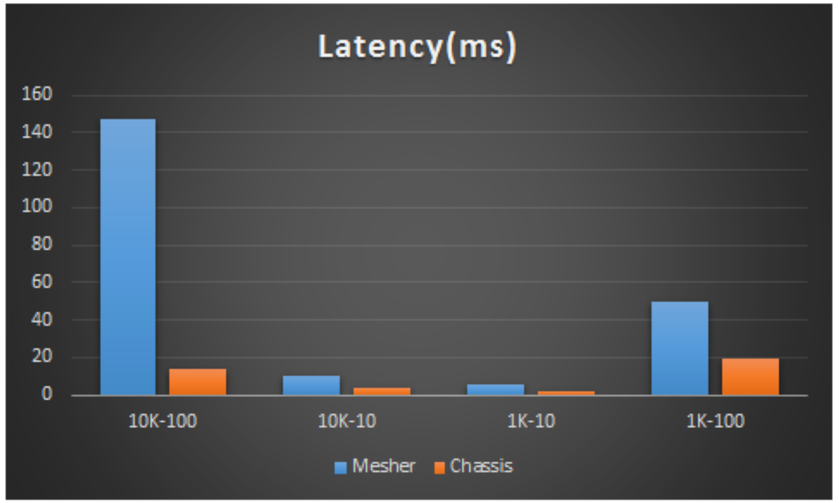
Running workload on High end(24C,128G) machines



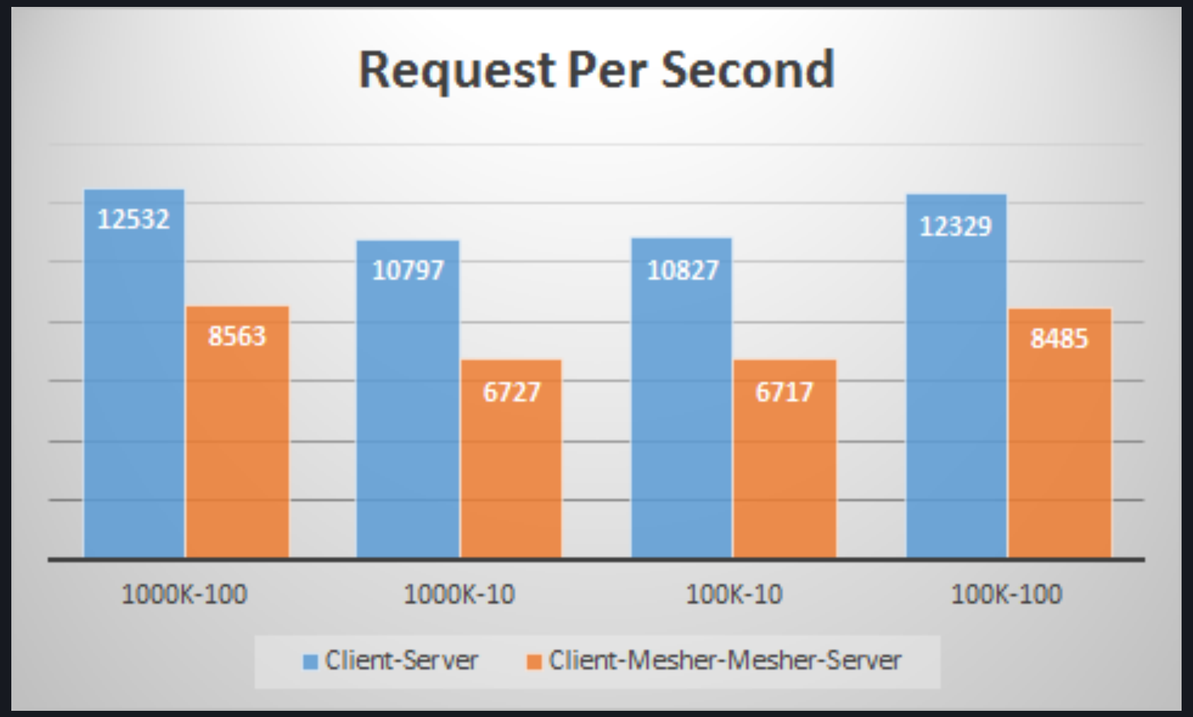


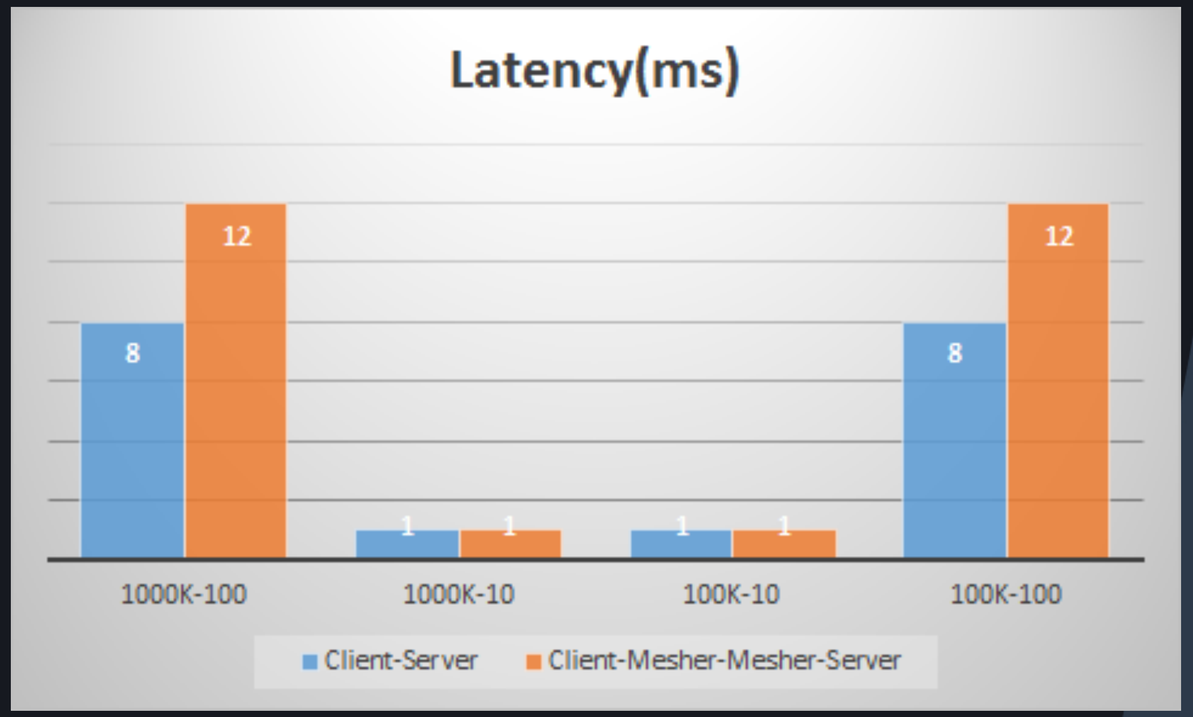
Running workload on Normal(4C,8G) machines





Mesher与普通应用性能对比





为何性能下降依然要使用ServiceMesh?

为什么要开发Mesher

- Istio的性能问题
- Istio强绑定Kubernetes平台（1.7.4+）
- 从成本角度讲Linkerd并不适合做SideCar部署，JVM资源占 用较多
- 在ServiceComb和CSE中的积累
- 最大优势，侵入式与非侵入式的无缝结合，混编，统一的产品体验。

开源

- Service Center

https://github.com/ServiceComb/service-center

- ServiceComb-Java

https://github.com/ServiceComb/ServiceComb-Java-Chassis

- Mesher examples

https://github.com/huawei-microservice-demo/mesher-example

未来

数据面竞争点：特性，可配置，扩展性，性能

管理面竞争点：特性，可配置，扩展性，易用性

最重要的：编排调度平台与数据面的整合，为开发者带来最完整的产品体验

Take away

- The Hardest Part of Microservices: Calling Your Services
- Envoy如何在Lyft使用
- Kubernetes与云原生应用概览
- 模式之服务网格
- Service Mesh：下一代微服务?
- 数人云|万字解读:Service Mesh服务网格新生代--Istio

THANK YOU

如有需求，欢迎至 [讲师交流会议室] 与我们的讲师进一步交流

xiaoliang.tian@gmail.com

ArchSummit
全 球 架 构 师 峰 会 2017