法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，讲师及小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

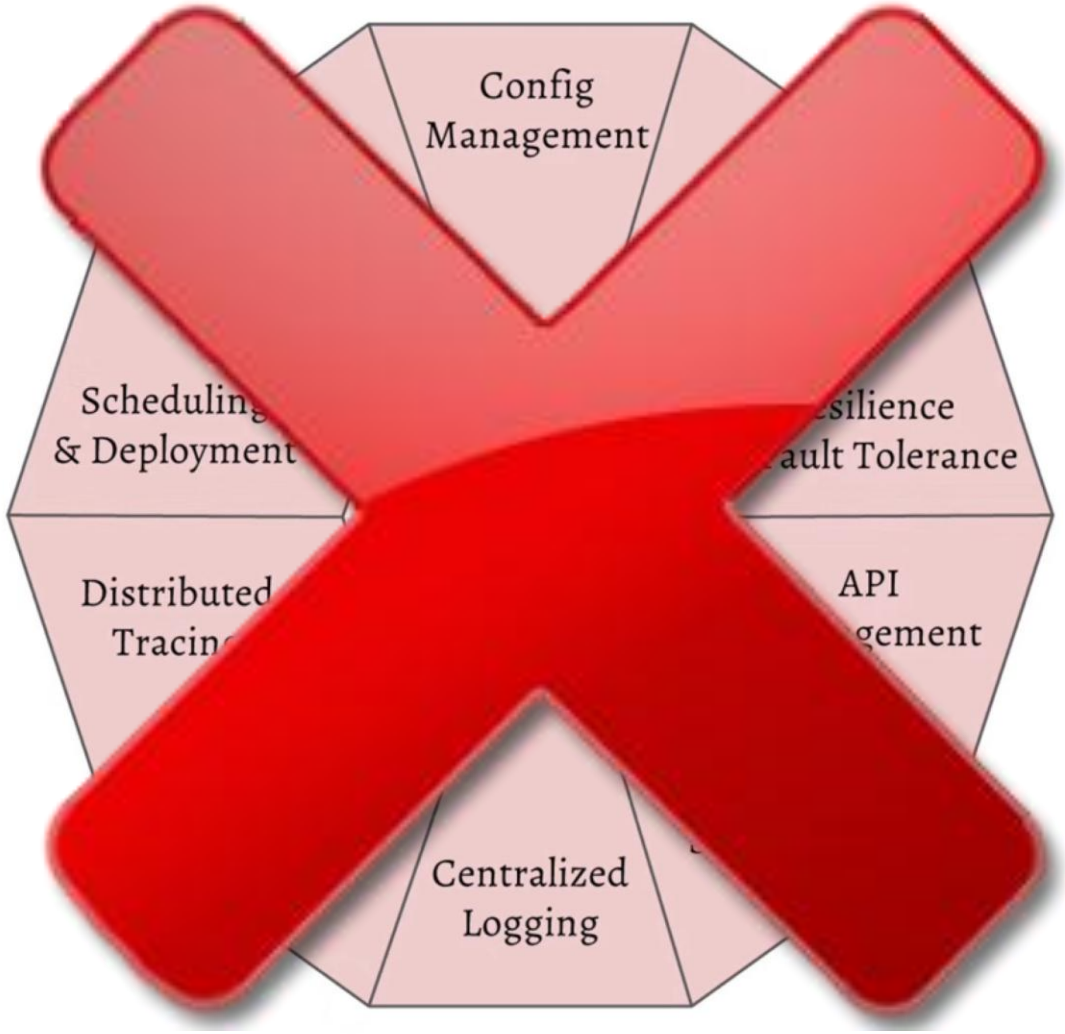
- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

微服务与Kubernetes

目录

1. 什么是微服务
2. Spring Cloud
3. Dubbo
4. ServiceComb
5. SeriveMesh… A new Micro Service?

1. 什么是微服务—— From Redhat



- 配置管理
- 自动伸缩
- 服务发现
- 调度/部署
- 弹性
- 分布式跟踪
- API管理
- 中心化指标
- 中心化日志
- 服务安全

1. Steve Y 对Amazon和Google的吐槽

- 微服务是否是个伪问题？
- https://coolshell.cn/articles/5701.html
- 上面这篇文档能帮助我们清晰的认知到，什么情况下需要微服务，什么情况下不需要

1. Steve Y 对Amazon和Google的吐槽

1) 所有团队的程序模块都要以通过Service Interface 方式将其数据与功能开放出来。（陈皓注：Service Interface也就是Web Service）

2) 团队间的程序模块的信息通信，都要通过这些接口。

3) 除此之外没有其它的通信方式。其他形式一概不允许：不能使用直接链结程序、不能直接读取其他团队的数据库、不能使用共享内存模式、不能使用别人模块的后门、等等，等等，唯一允许的通信方式只能是能过调用 Service Interface。

4) 任何技术都可以使用。比如：HTTP、Corba、Pubsub、自定义的网络协议、等等，都可以，Bezos不管这些。（陈皓注：Bezos不是微控经理吗？呵呵。）

5) 所有的Service Interface，毫无例外，都必须从骨子里到表面上设计成能对外界开放的。也就是说，团队必须做好规划与设计，以便未来把接口开放给全世界的程序员，没有任何例外。

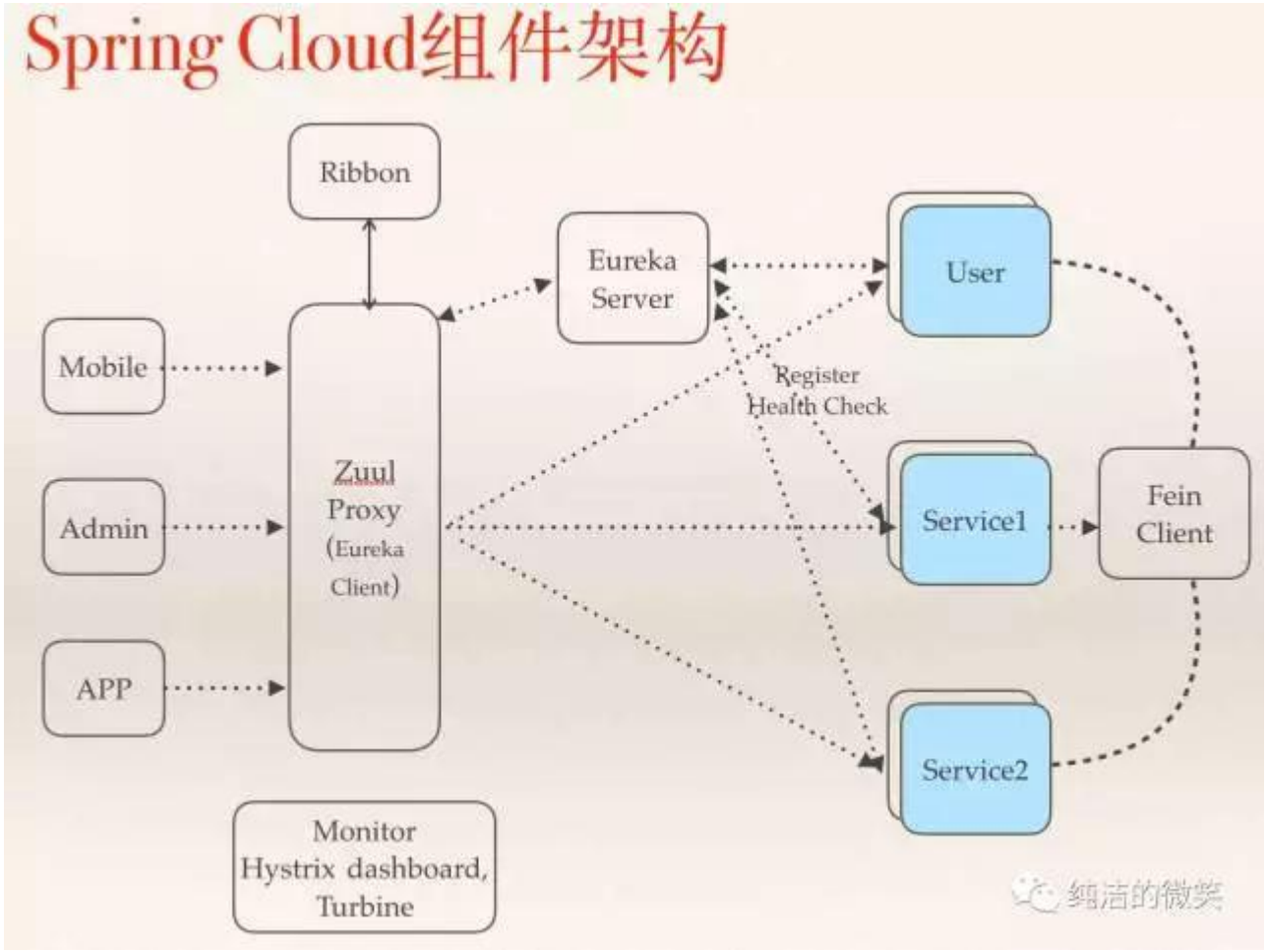
6) 不这样的做的人会被炒鱿鱼。

7) 谢谢，祝你有个愉快的一天！

1. Steve Y 对Amazon和Google的吐槽

Service = 平台？成功

2. Spring Cloud



2. Spring Cloud

1、请求统一通过API网关（Zuul）来访问内部服务.

2、网关接收到请求后，从注册中心（Eureka）获取可用服务

3、由Ribbon进行均衡负载后，分发到后端具体实例

4、微服务之间通过Feign进行通信处理业务

5、Hystrix负责处理服务超时熔断

6、Turbine监控服务间的调用和熔断相关指标

2. Spring Cloud

《Spring Cloud构建微服务架构（一）服务注册与发现》

《Spring Cloud构建微服务架构（二）服务消费者》

《Spring Cloud构建微服务架构（三）断路器》

《Spring Cloud构建微服务架构（四）分布式配置中心》

《Spring Cloud构建微服务架构（五）服务网关》

《Spring Cloud构建微服务架构（六）高可用服务注册中心》

《Spring Cloud构建微服务架构（七）消息总线》

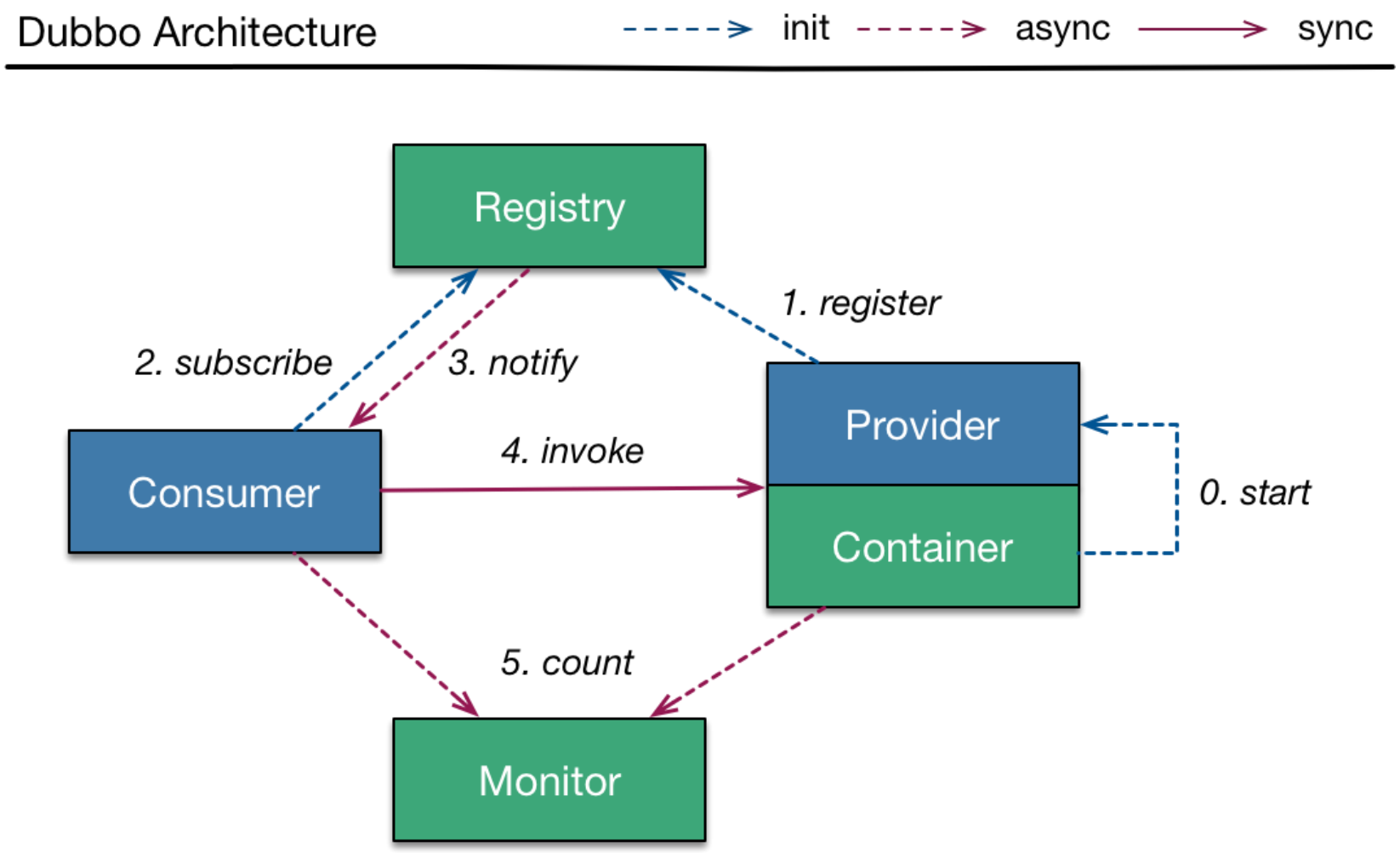
2. Spring Cloud vs K8s

Microservices Concern	Spring Cloud & Netflix OSS	Kubernetes
Configuration Management	Config Server, Consul, Netflix Archaius	Kubernetes ConfigMap & Secrets
Service Discovery	Netflix Eureka, Hashicorp Consul	Kubernetes Service & Ingress Resources
Load Balancing	Netflix Ribbon	Kubernetes Service
API Gateway	Netflix Zuul	Kubernetes Service & Ingress Resources
Service Security	Spring Cloud Security	-
Centralized Logging	ELK Stack (LogStash)	EFK Stack (Fluentd)
Centralized Metrics	Netflix Spectator & Atlas	Heapster, Prometheus, Grafana
Distributed Tracing	Spring Cloud Sleuth, Zipkin	OpenTracing, Zipkin
Resilience & Fault Tolerance	Netflix Hystrix, Turbine & Ribbon	Kubernetes Health Check & resource isolation
Auto Scaling & Self Healing	-	Kubernetes Health Check, Self Healing, Autoscaling
Packaging, Deployment & Scheduling	Spring Boot	Docker/Rkt, Kubernetes Scheduler & Deployment
Job Management	Spring Batch	Kubernetes Jobs & Schedul[illegible]bs
Singleton Application	Spring Cloud Cluster	Kubernetes Pods

Java

Docker

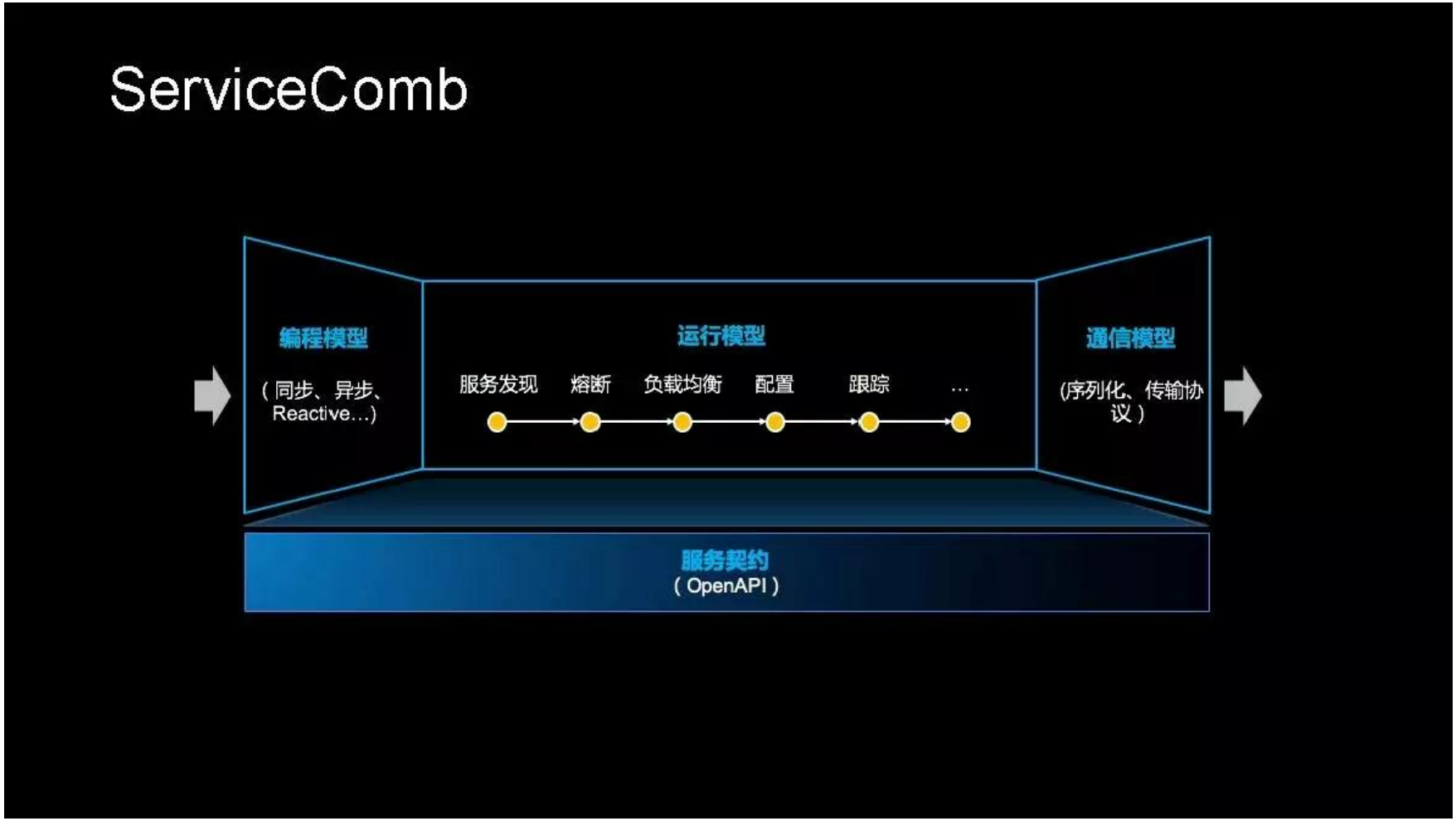
3. Dubbo


Dubbo Architecture
init
async
sync
Registry
1. register
2. subscribe
3. notify
Provider
Consumer
4. invoke
0. start
Container
5. count
Monitor

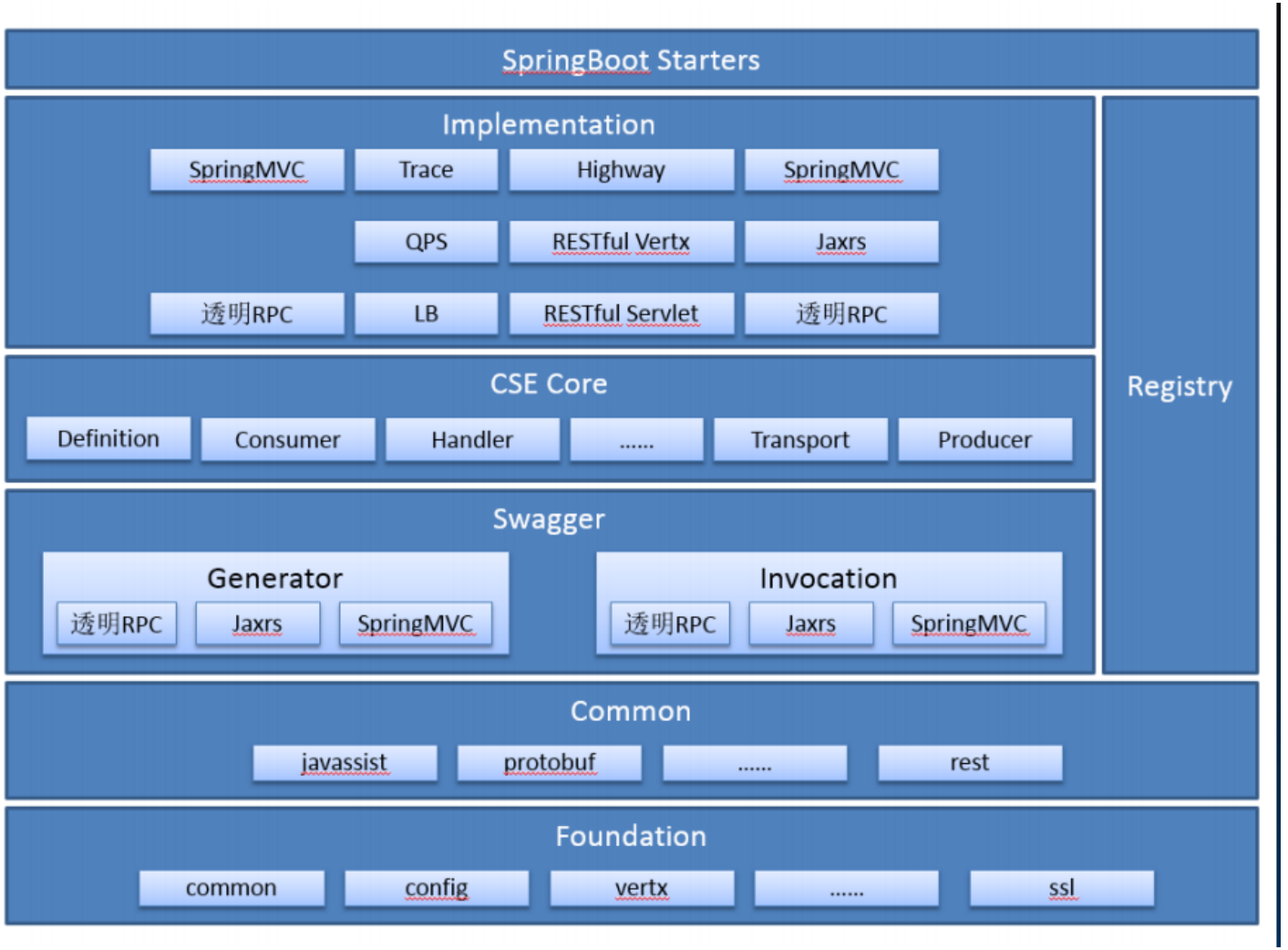
3. Dubbo vs SpringCloud

	Dubbo	Spring Cloud
服务注册中心	Zookeeper	Spring Cloud Netflix Eureka
服务调用方式	RPC	REST API
服务监控	Dubbo-monitor	Spring Boot Admin
断路器	不完善	Spring Cloud Netflix Hystrix
服务网关	无	Spring Cloud Netflix Zuul
分布式配置	无	Spring Cloud Config
服务跟踪	无	Spring Cloud Sleuth
消息总线	无	Spring Cloud Bus
数据流	无	Spring Cloud Stream
批量任务	无	Spring Cloud Task
……	……	……

4. ServiceComb



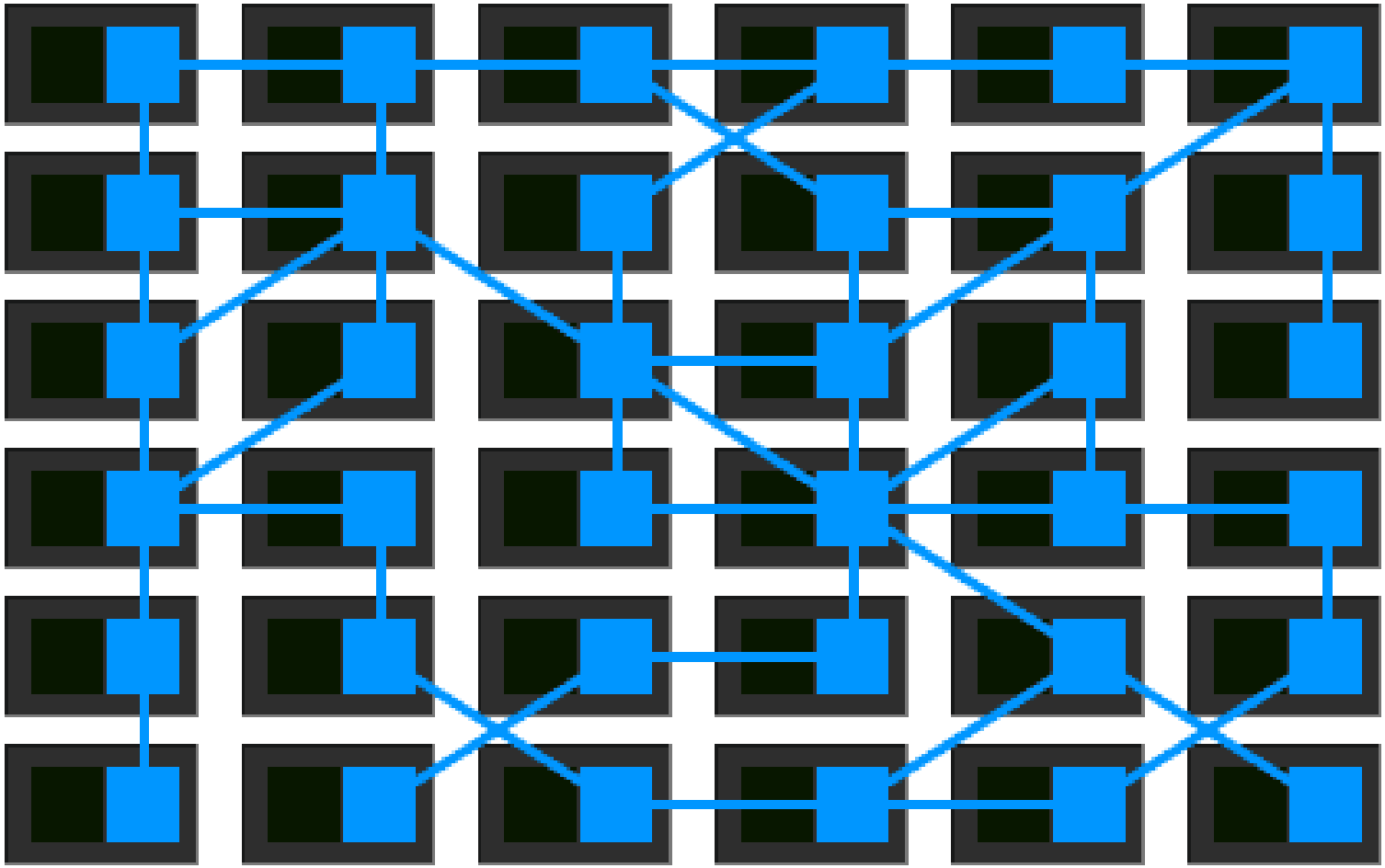
4. ServiceComb



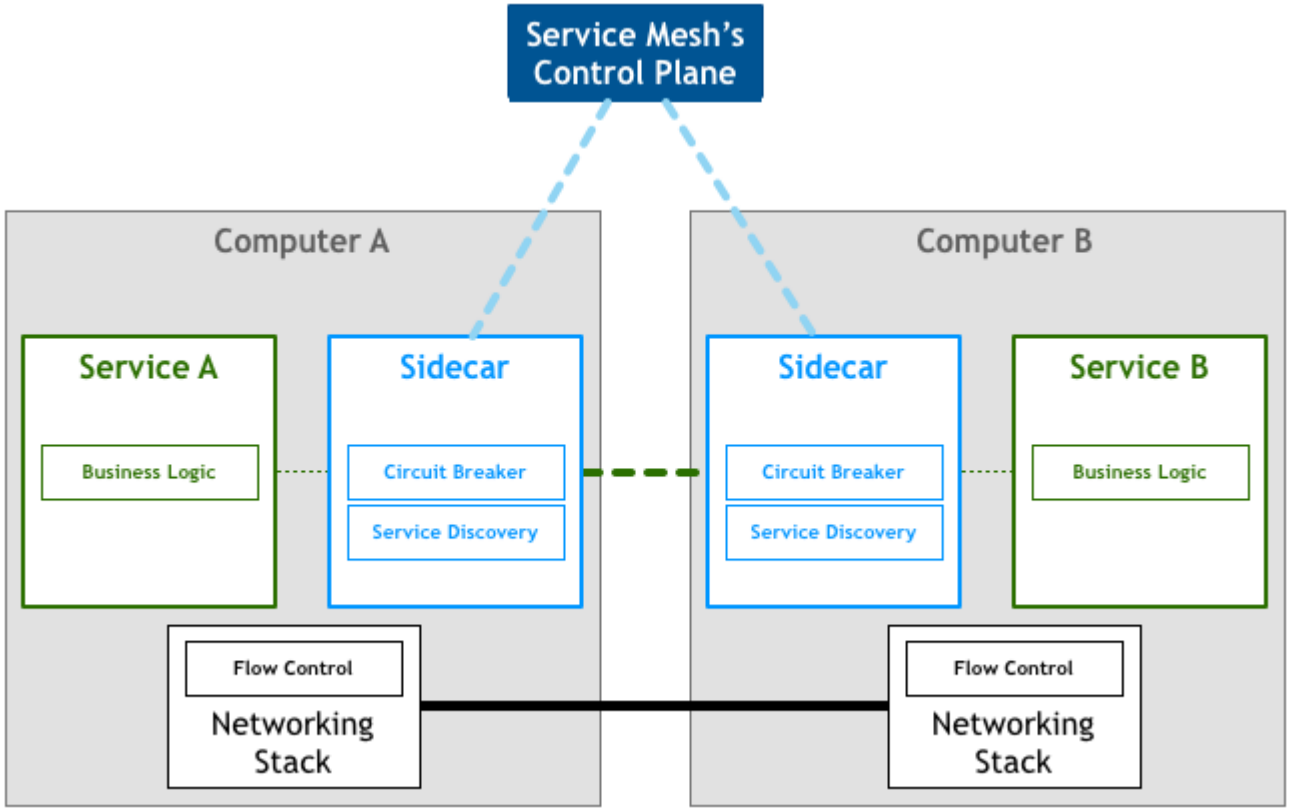
4. ServiceComb

- 《如何评价华为新开源的ServiceComb微服务框架？》
- 封装度 > SpringCloud
- 语言支撑丰富度(Java/Go) > Dubbo

5. ServiceMesh



5. ServiceMesh

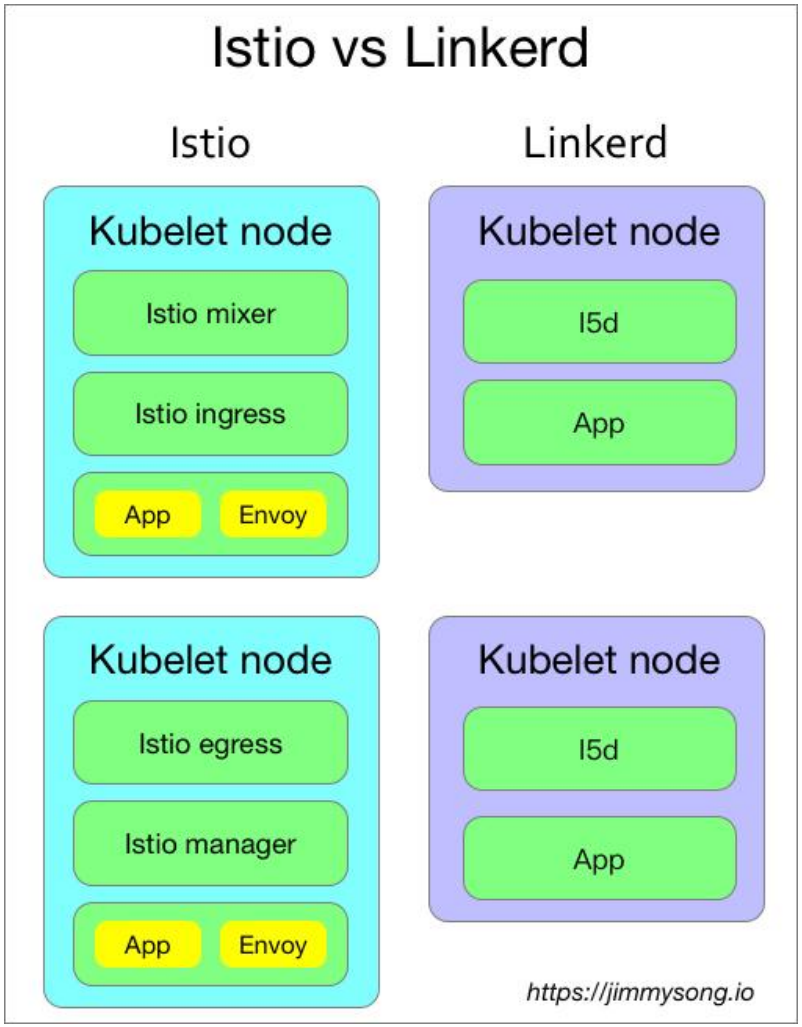


语言绑定=痛苦的开发态 -> 无绑定、无侵入

5. ServiceMesh

Feature	Istio	Linkerd
部署架构	Envoy/Sidecar	DaemonSets
易用性	复杂	简單
支持平台	kuberentes	kubernetes/mesos/Istio/local
当前版本	0.3.0	1.3.3
是否已有生产部署	否	是

5. ServiceMesh



- Istio依然非常复杂，符合IBM的一贯风格
- Linkerd生产可用
- 新的ServiceMesh：

https://github.com/runconduit/conduit

On k8s only

Rust

《Conduit官方文档》

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop