



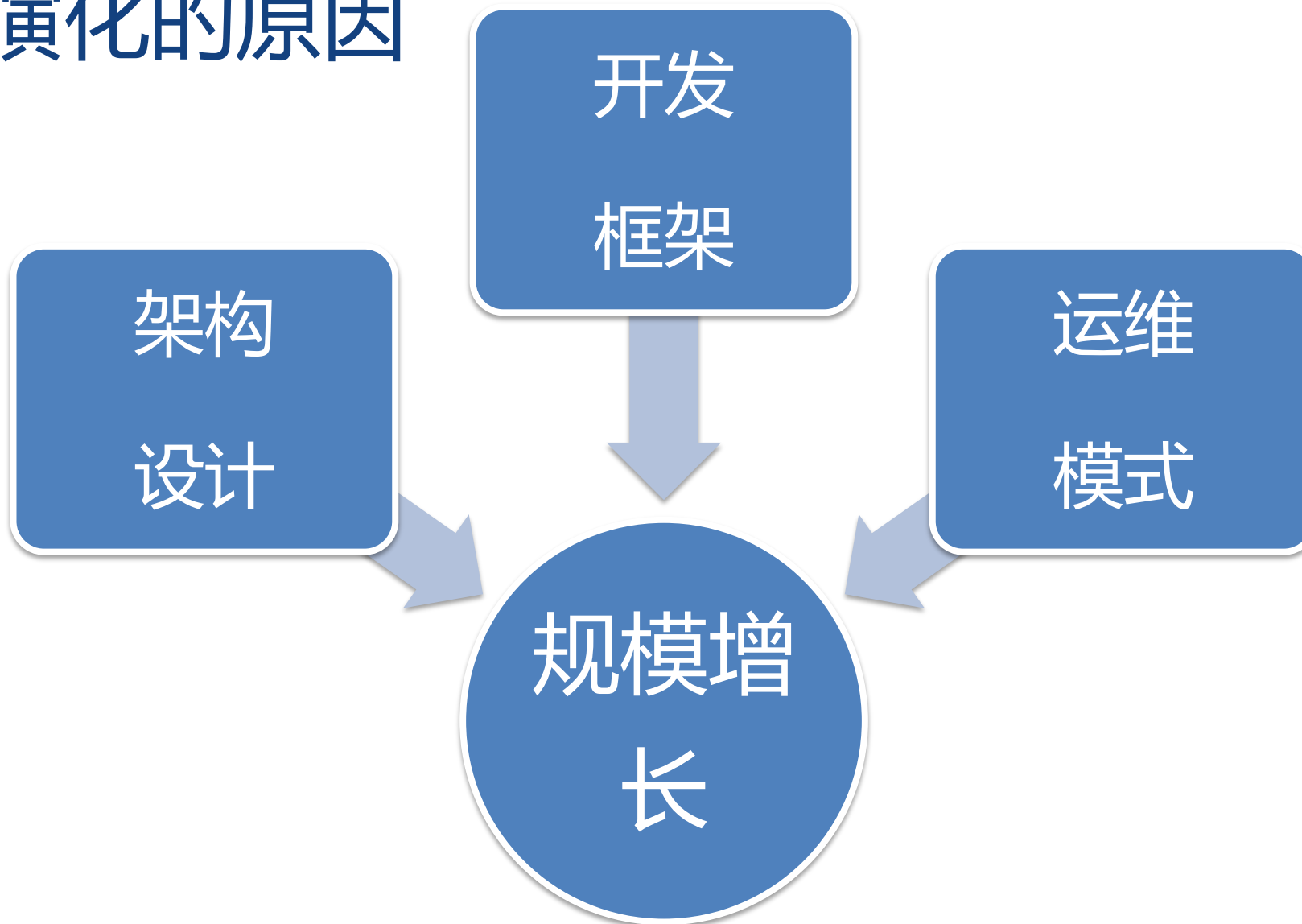
架构迎接未来变化
IAS2017 • NANJING



技术架构演变全景图 从单体式到云原生

曹祖鹏&张亮

技术演化的原因

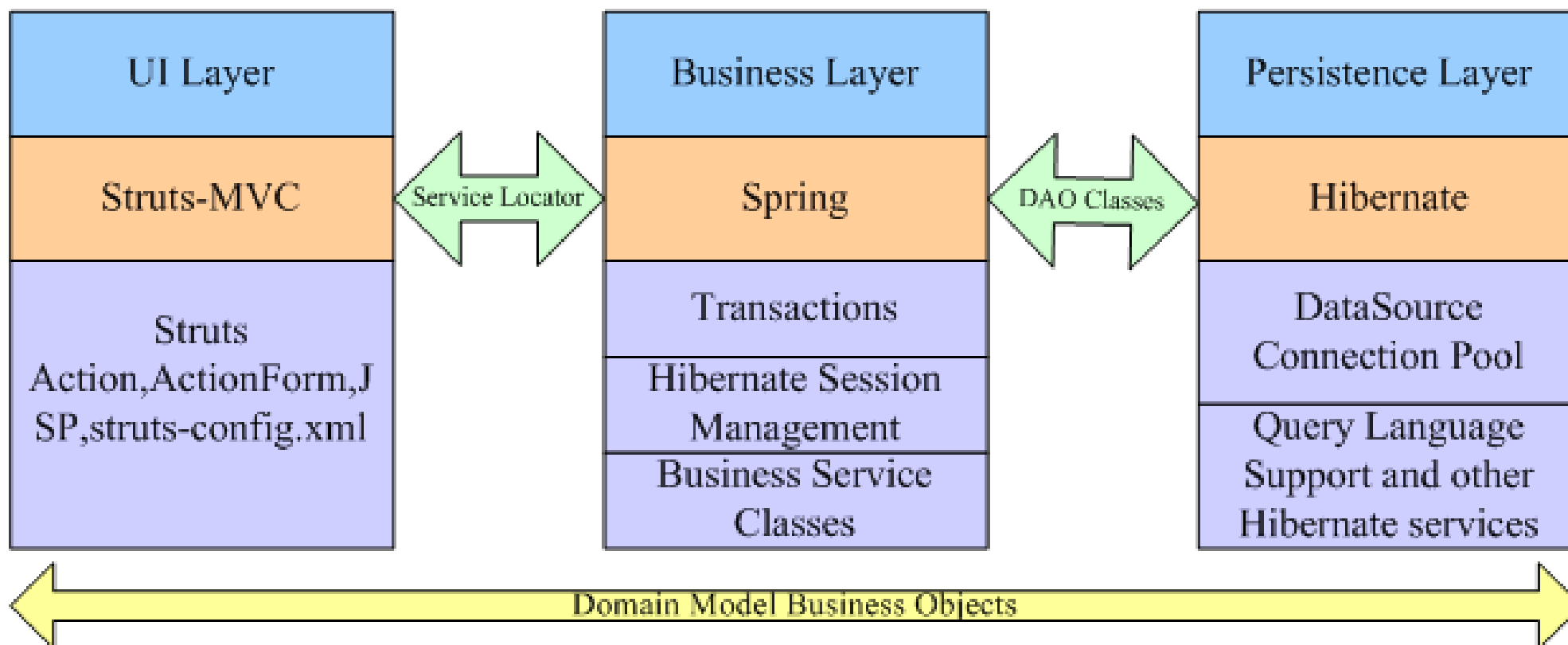


技术演化方向

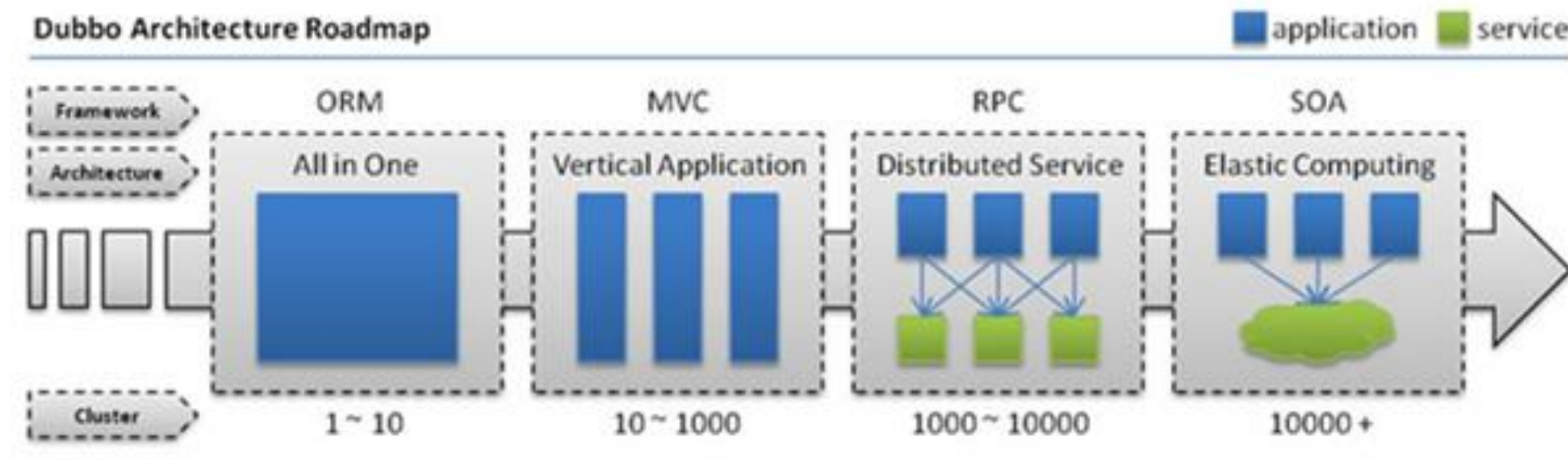
- 架构设计
 - 分层 -> SOA -> 微服务
- 开发框架
 - 单体式 -> 分布式 -> 云原生
- 运维模式
 - 脚本化 -> 工具化 -> 自动化

开发框架 – 从SSH说起

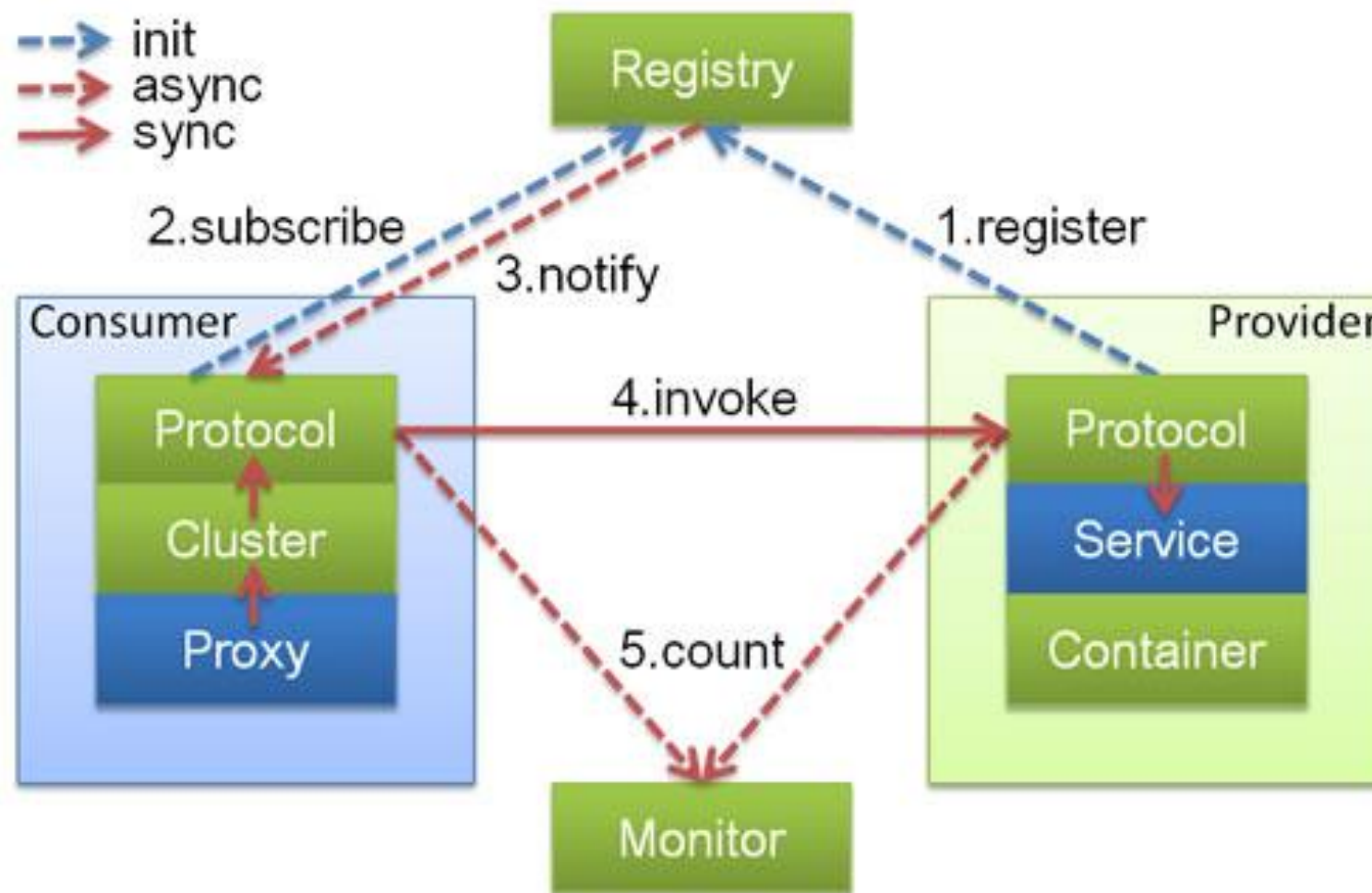
Figure :Overview of framework architecture with Struts, Spring, and Hibernate.



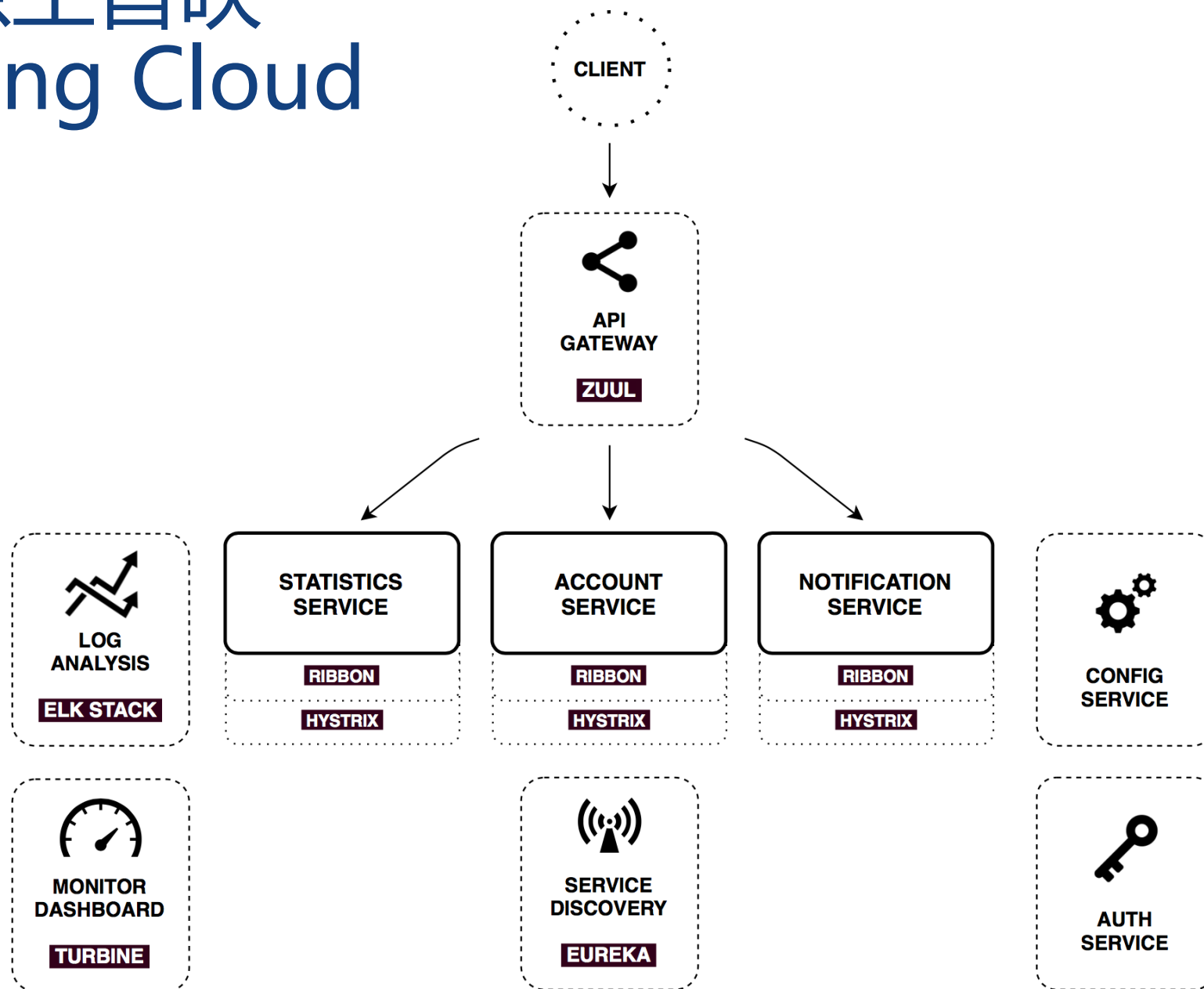
单体式架构 -> 分布式架构



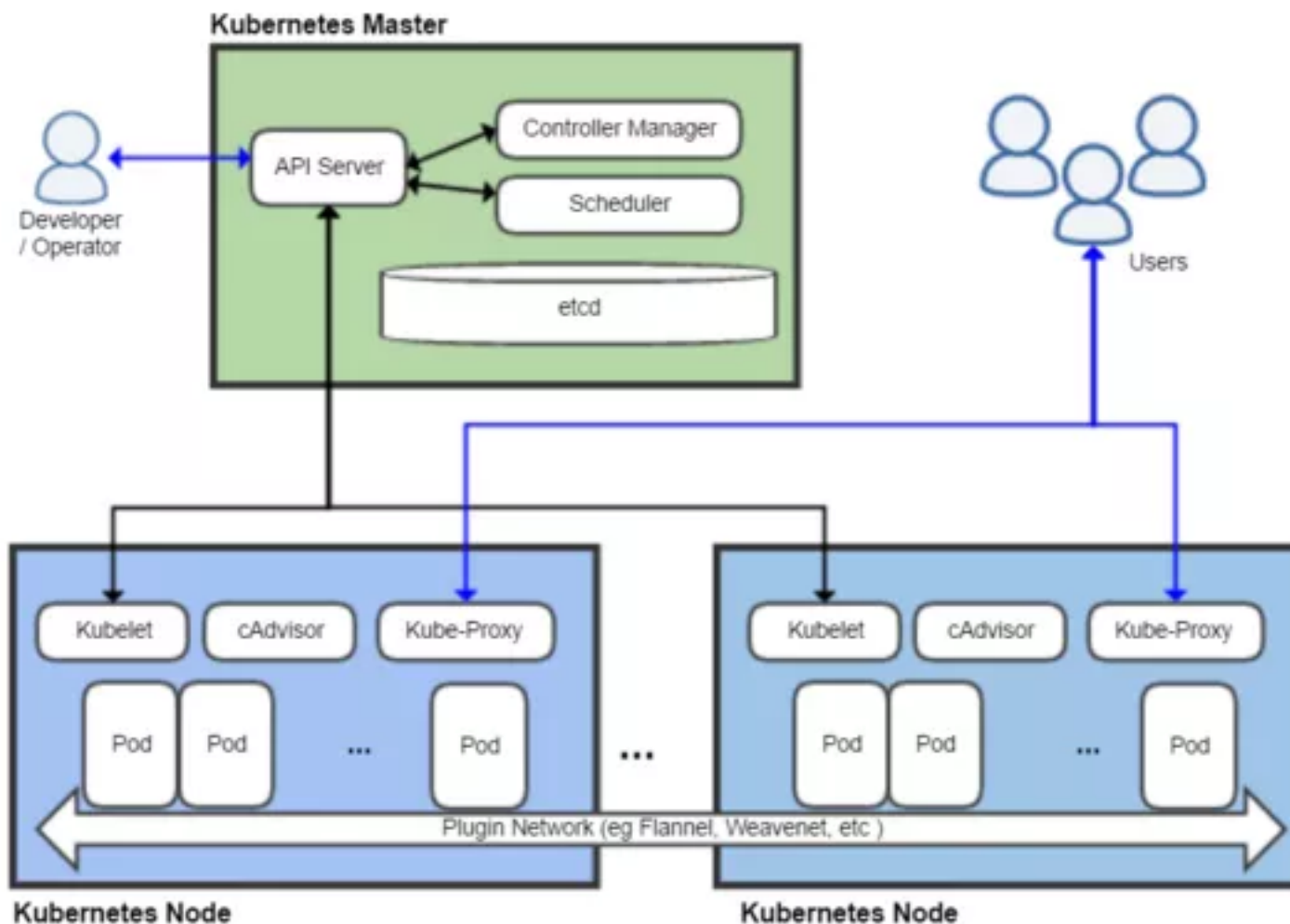
分布式框架 - Dubbo的出现



云原生首映 Spring Cloud



运维模式改变 - 容器 + 编排 (K8S)



DevOps Experience
Auto Scaling & Self Healing
Resilience & Fault Tolerance
Distributed Tracing
Centralized Metrics
Centralized Logging
API Gateway
Job Management
Singleton Application
Load Balancing
Service Discovery
Configuration Management
Application Packaging
Deployment & Scheduling
Process Isolation
Environment Management
Resource Management
Operating System
Virtualization
Hardware, Storage, Networking

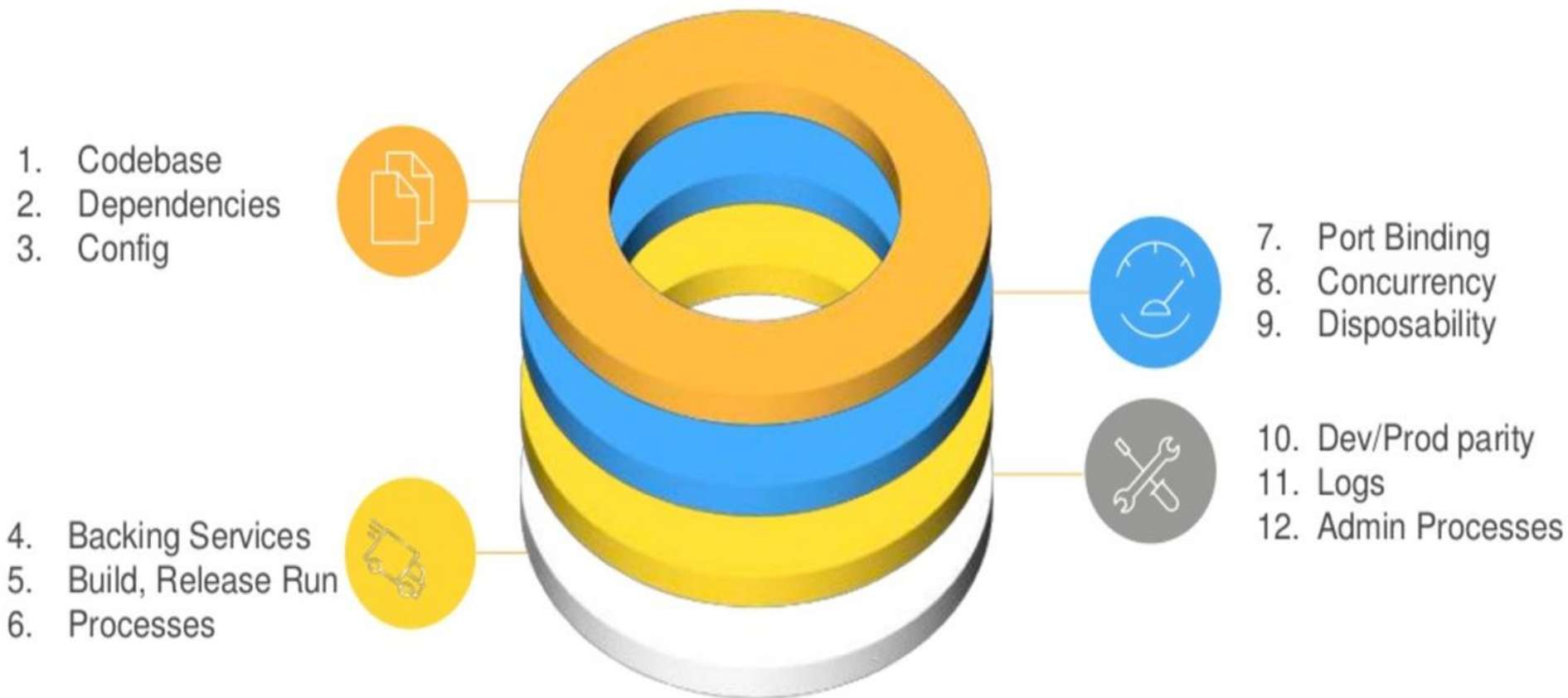


Capability	Spring Cloud with Kubernetes
DevOps Experience	Self service, multi-environment capabilities
Auto Scaling & Self Healing	Pod/Cluster Autoscaler, HealthIndicator, Scheduler
Resilience & Fault Tolerance	HealthIndicator, Hystrix, HealthCheck, Process Check
Distributed Tracing	Zipkin
Centralized Metrics	Heapster, Prometheus, Grafana
Centralized Logging	EFK
Job Management	Spring Batch, Scheduled Job
Load Balancing	Ribbon, Service
Service Discovery	Service
Configuration Management	Externalized Configurations, ConfigMap, Secret
Service Logic	Apache Camel, Spring Framework
Application Packaging	Spring Boot maven plugin
Deployment & Scheduling	Deployment strategy, A/B, Canary, Scheduler strategy
Process Isolation	Docker, Pods
Environment Management	Namespaces, Authorizations
Resource Management	CPU and memory limits, Namespace resource quotas
IaaS	GCE, Azure, CenturyLink, VMware, Openstack

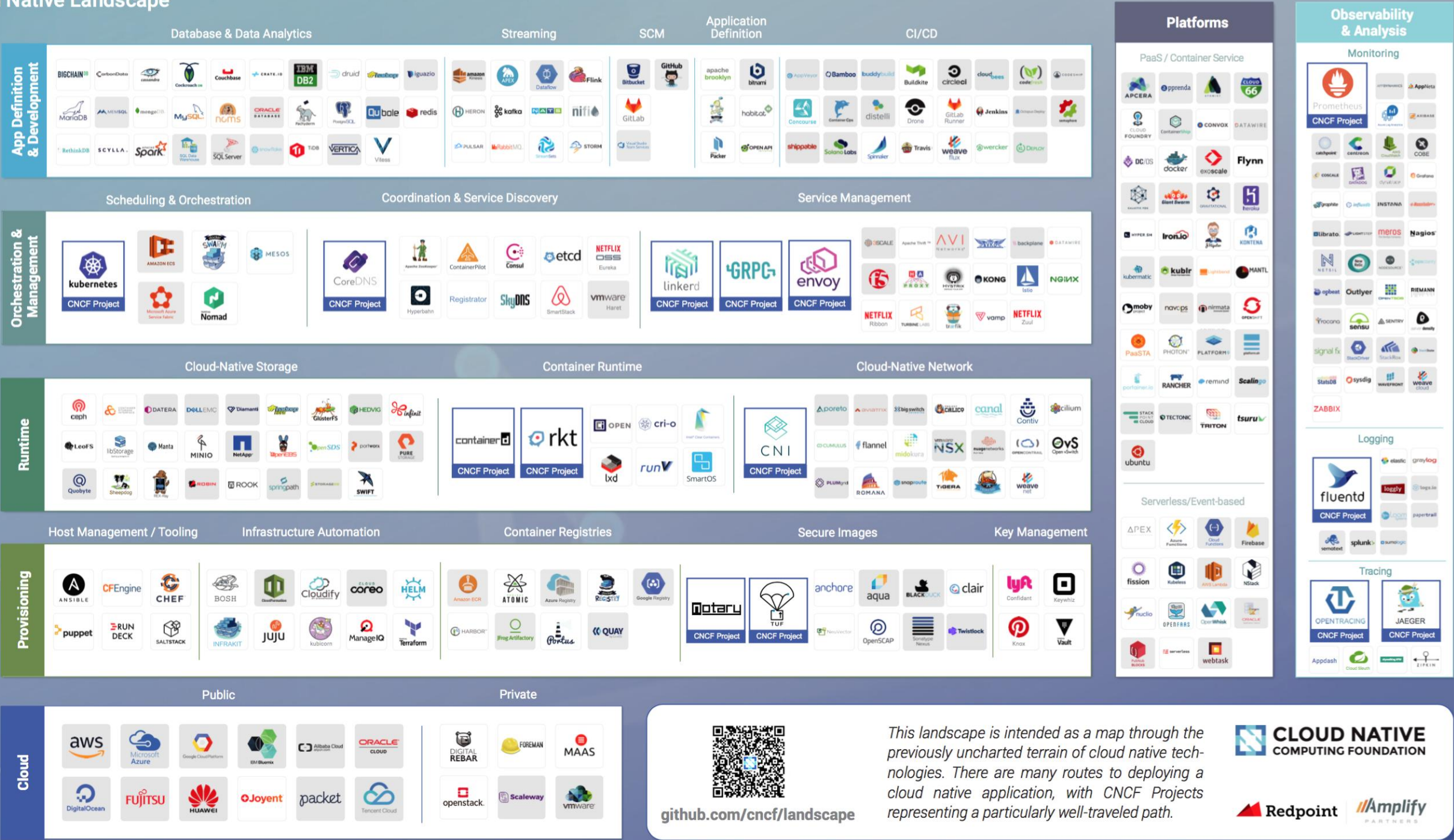
云原生是一种模式

- 可用性 & 伸缩性
- 自动化部署 & 管理
- 效率提升
- 随处运行

云原生十二要素



Cloud Native Landscape
v0.9.9



github.com/cncf/landscape

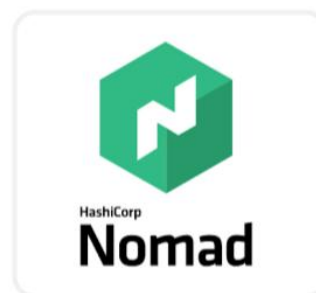
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.



编排领域

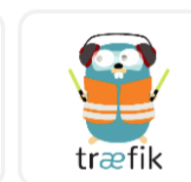
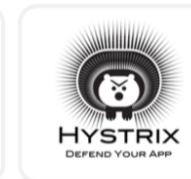
Scheduling & Orchestration

Orchestration & Management



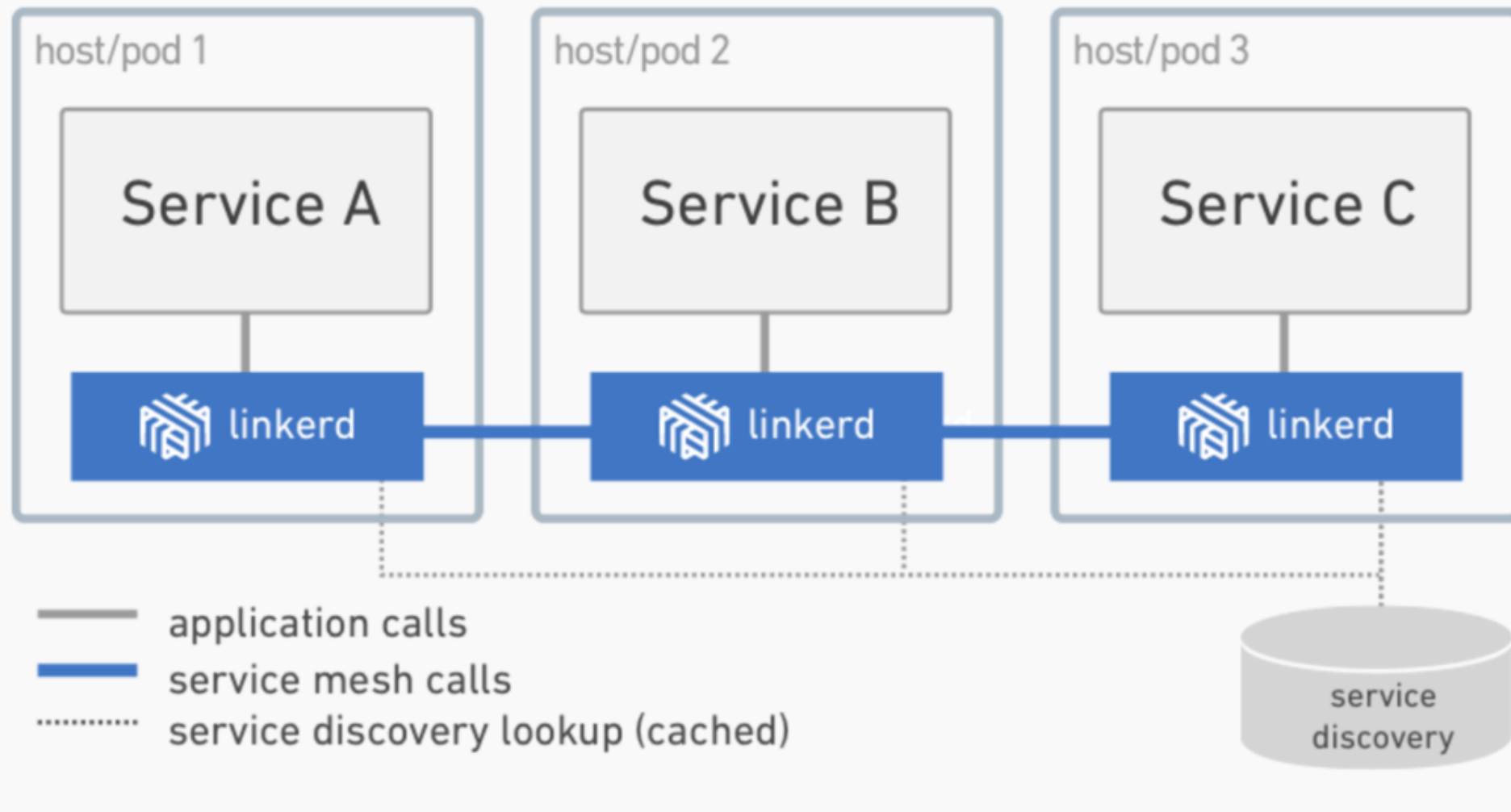
服务治理领域

Service Management

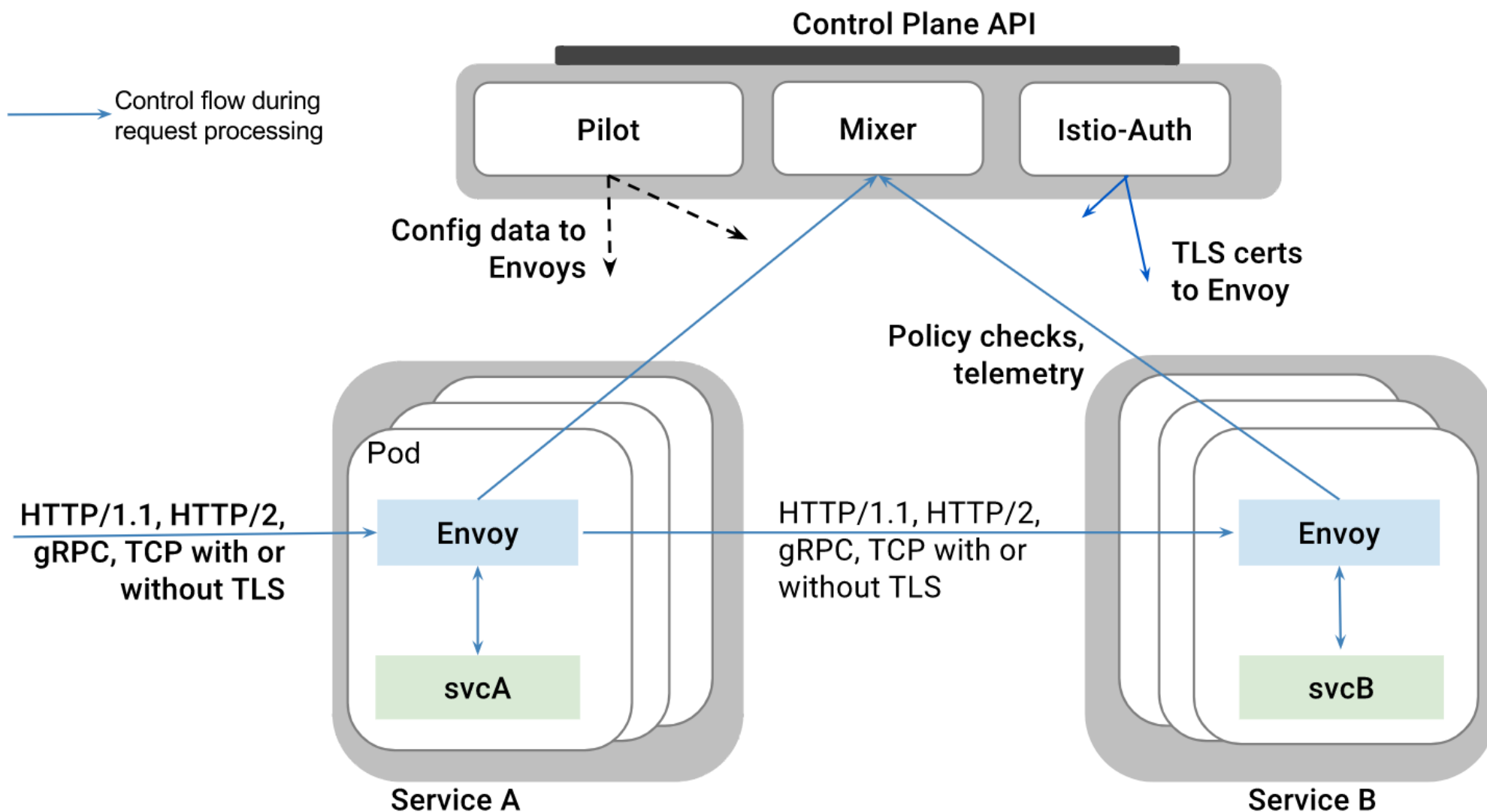


Service Mesh - Linkerd

Linkerd instances form a service mesh, allowing application code to communicate reliably.



Service Mesh - Istio



对开发和运维模式的改变

- 对运维模式的改变
 - 自动化
 - 可视化
- 对开发模式的改变
 - 关注业务本身胜于非功能需求
 - 调试模式改变

2015年的微服务技术栈



NETFLIX Ribbon



2016年的微服务技术栈



OPENSIFT



HYSTRIX
DEFEND YOUR APP

2017年的微服务技术栈



OPENSIFT

2018年的微服务技术栈？



未来趋势 – 像使用物理机一样使用云

- 操作系统
 - 容器调度+编排
- 运行时
 - 容器
- 通信
 - Service Mesh

信息参考

- **Cloud Native Landscape Project**
(<https://github.com/cncf/landscape>)
- **Spring Cloud for Microservices Compared to Kubernetes**
(<https://developers.redhat.com/blog/2016/12/09/spring-cloud-for-microservices-compared-to-kubernetes/>)
- **8 Steps to Becoming Awesome with Kubernetes**
(<http://bit.ly/8stepsawesome>)

THANKS



架构迎接未来变化

IAS2017 • NANJING