

构建微服务云原生应用

架构师杨波



扫码试看/订阅

《Spring Boot & Kubernetes 云原生微服务实践》

CHAPTER

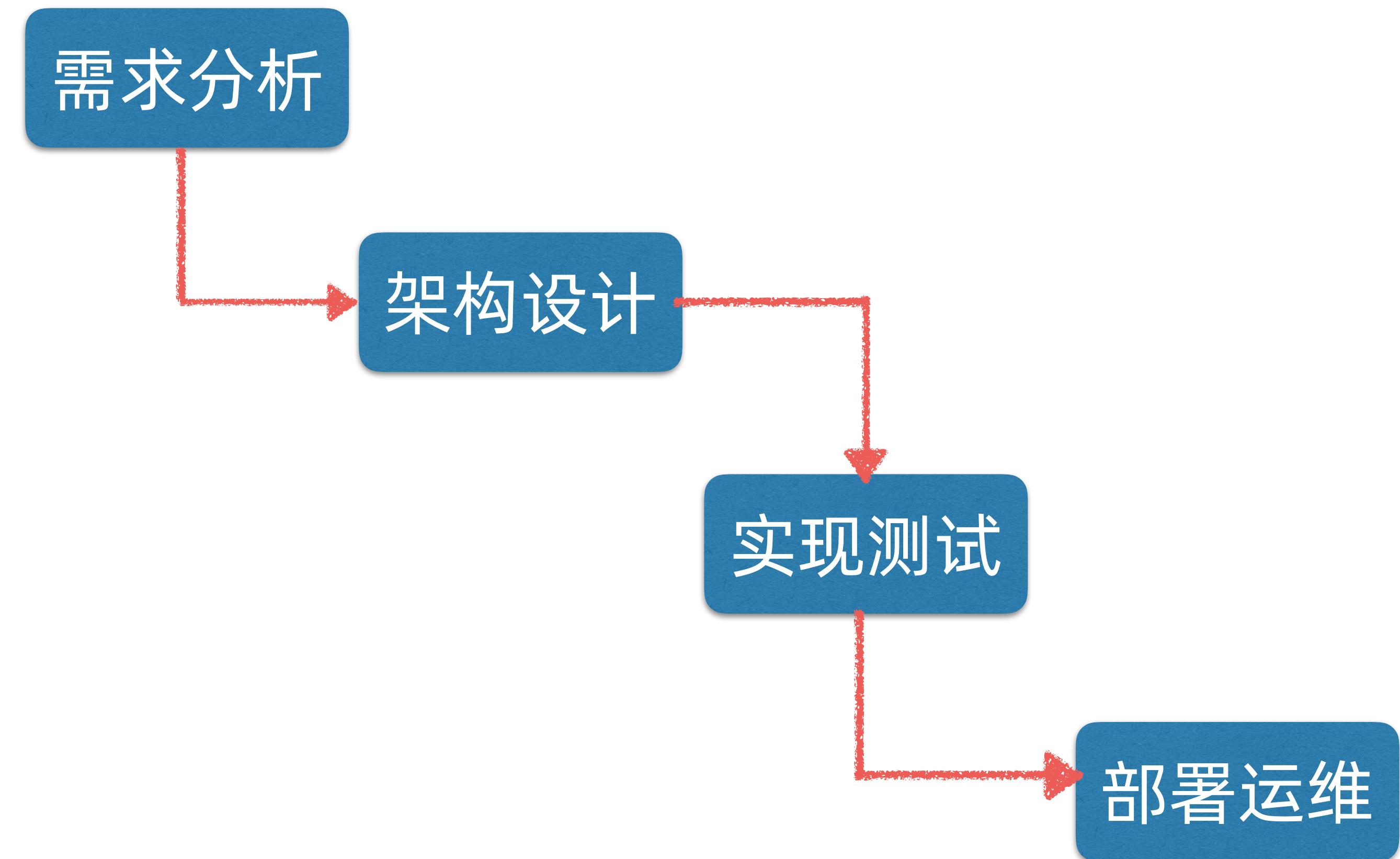
7

可运维架构设计和实践

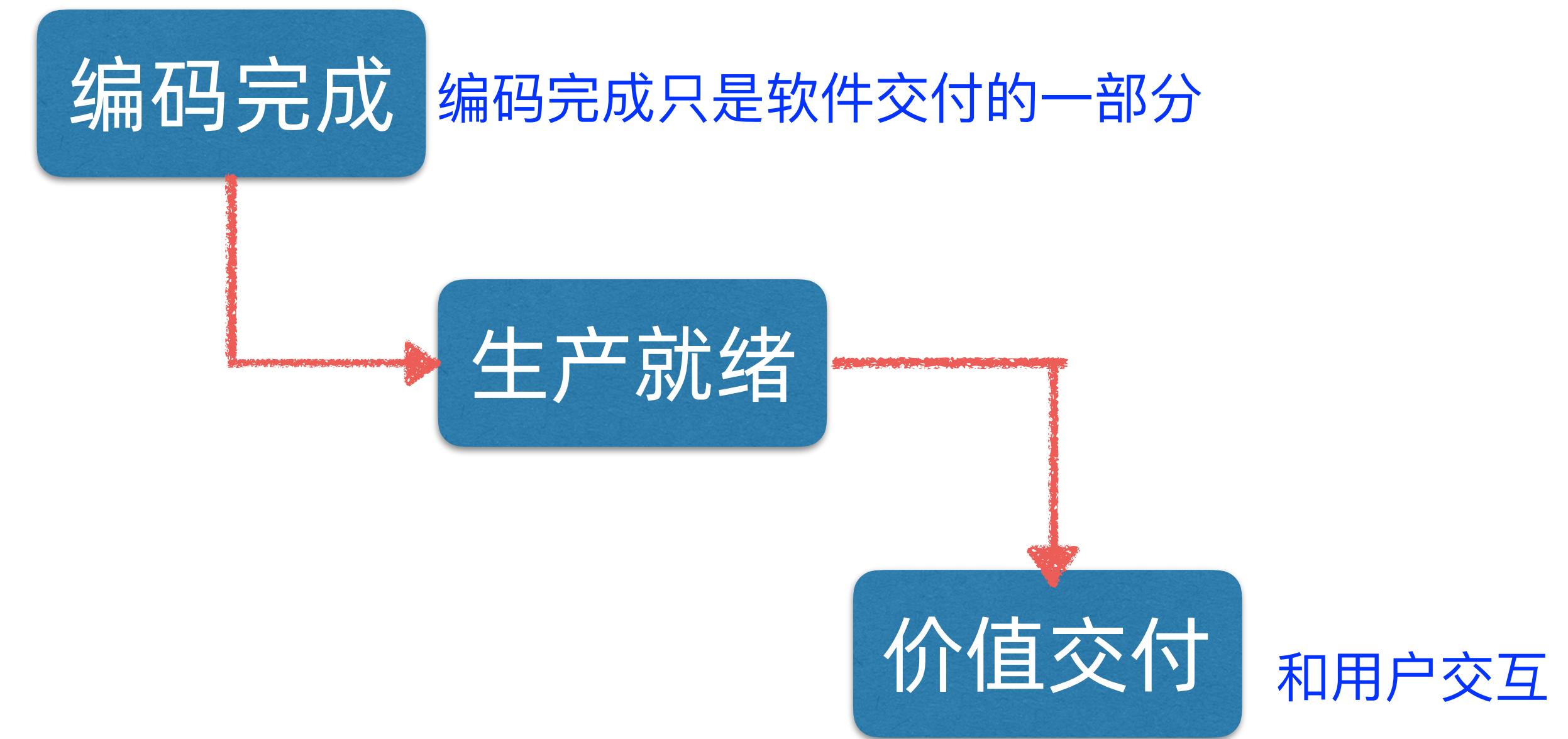
第 1 部分

何谓生产就绪 Production Ready?

经典软件工程阶段



互联网软件交付阶段



何谓生产就绪

功能测试OK

性能测试OK

高可用考量

比如有状态:主备
无状态:水平扩展

配置管理

Production
Ready

扩展升级考量

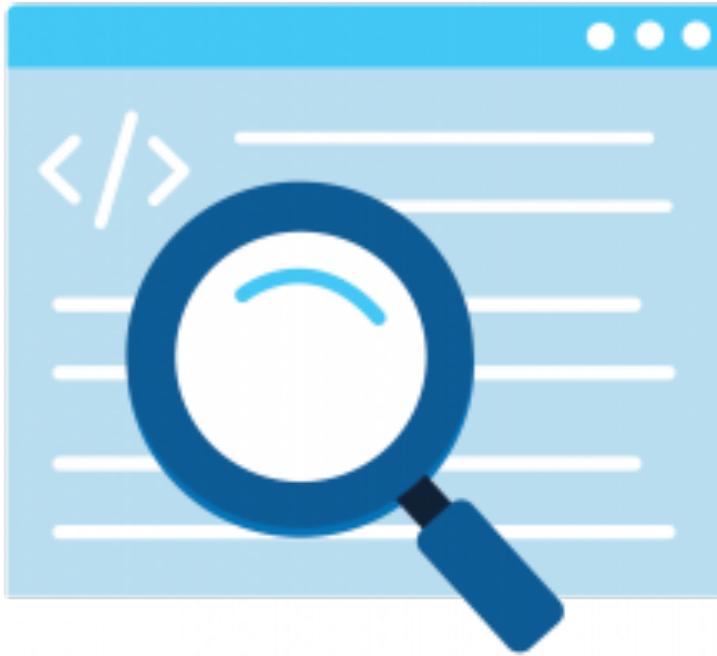
日志监控

安全性考量

Metrics监控

健康检查

调用链监控

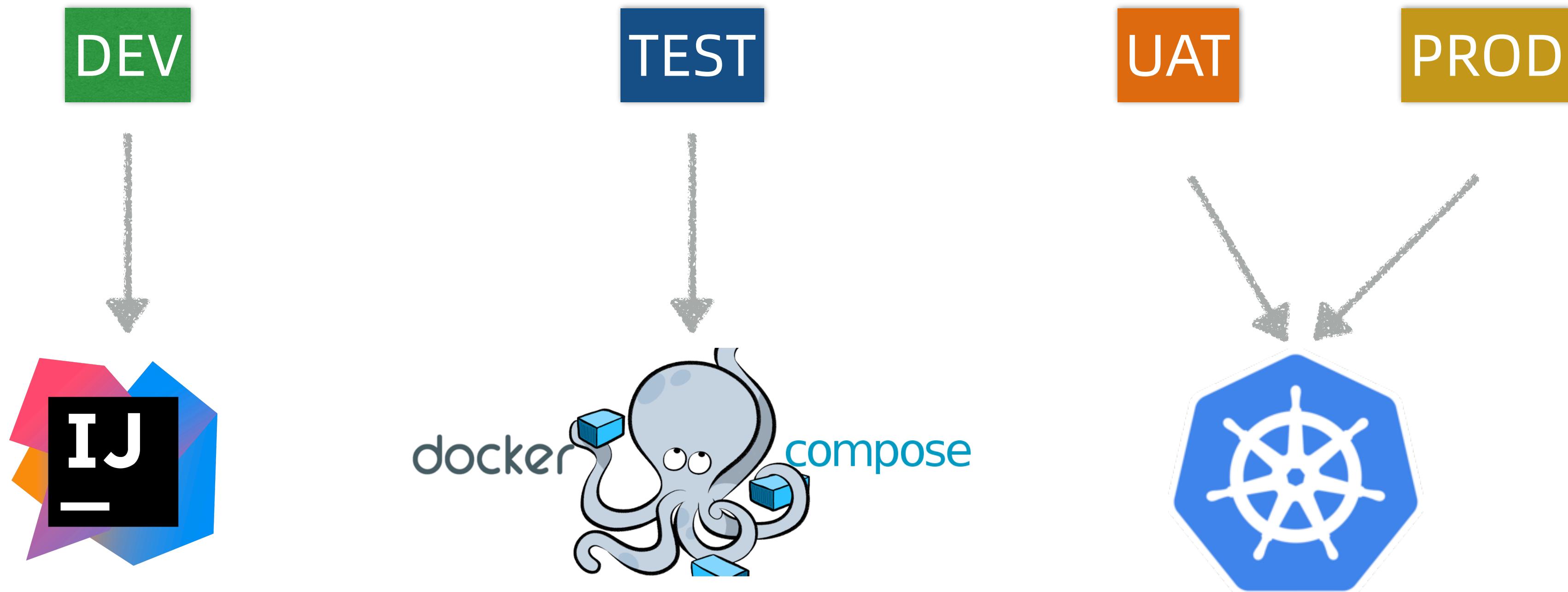


第 2 部分

SpringBoot如何实现分环境配置？

原理:基于 spring 的 profile

Staffjoy环境

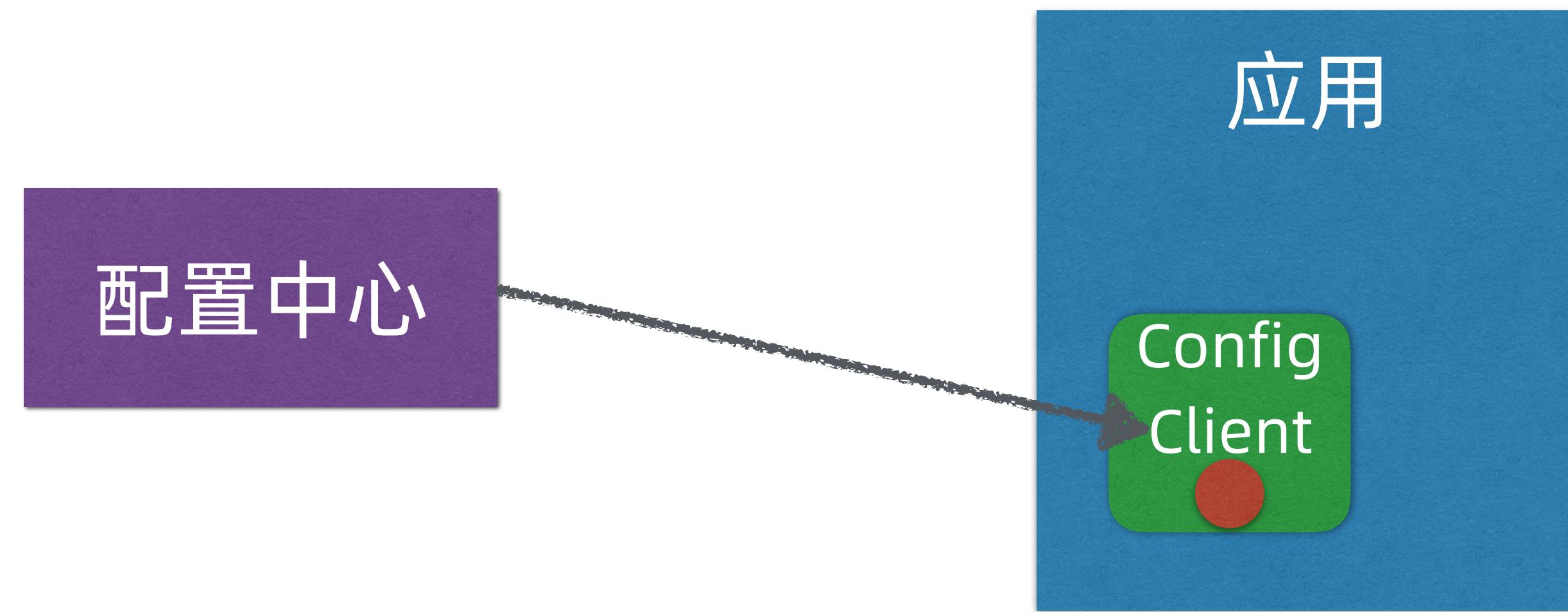


如account-svc的定义多个环境的properties文件，
公共部分放在application.yml/properties

第 3 部分

Apollo vs Spring Cloud Config vs K8s ConfigMap

何谓动态配置更新



配置客户端

动态的概念:将配置给客户端，新的配置的动作需要应用去实现

如 apollo 只负责配置交付，可能需要应用重启，自己实现

Apollo vs Spring Cloud Config vs K8s ConfigMap

	Apollo	Spring Cloud Config	K8s ConfigMaps
配置界面	统一界面管理不同环境和集群配置	无，通过git操作	Cli或Dashboard
配置存储	DB mysql	Git	Etcd 注入到pod中
配置生效时间	实时推送+应用配合	近实时+应用配合	近实时+应用配合 如环境变量或 mount
动态配置	支持，实时推送	复杂+Mq 消息队列	支持发布更新
版本管理	UI支持发布历史和回滚	无，通过git操作	无，需自己管理 可以回退 pod
灰度发布	支持	不支持	支持灰度发布
授权/审计/审核	UI操作，修改和发布权限分离	需通过git仓库设置	K8s平台部分支持
实例配置监控	可见哪些客户端配置生效	不支持	可查询容器环境变量
客户端支持	原生Java/.Net，提供API，支持Spring标注	Spring应用+标注支持	语言无关 环境变量或mount文件系统

参考样例

Apollo动态配置

<https://github.com/ctripcorp/apollo-use-cases>

SpringCloud集中配置

springcloud配置参考项目

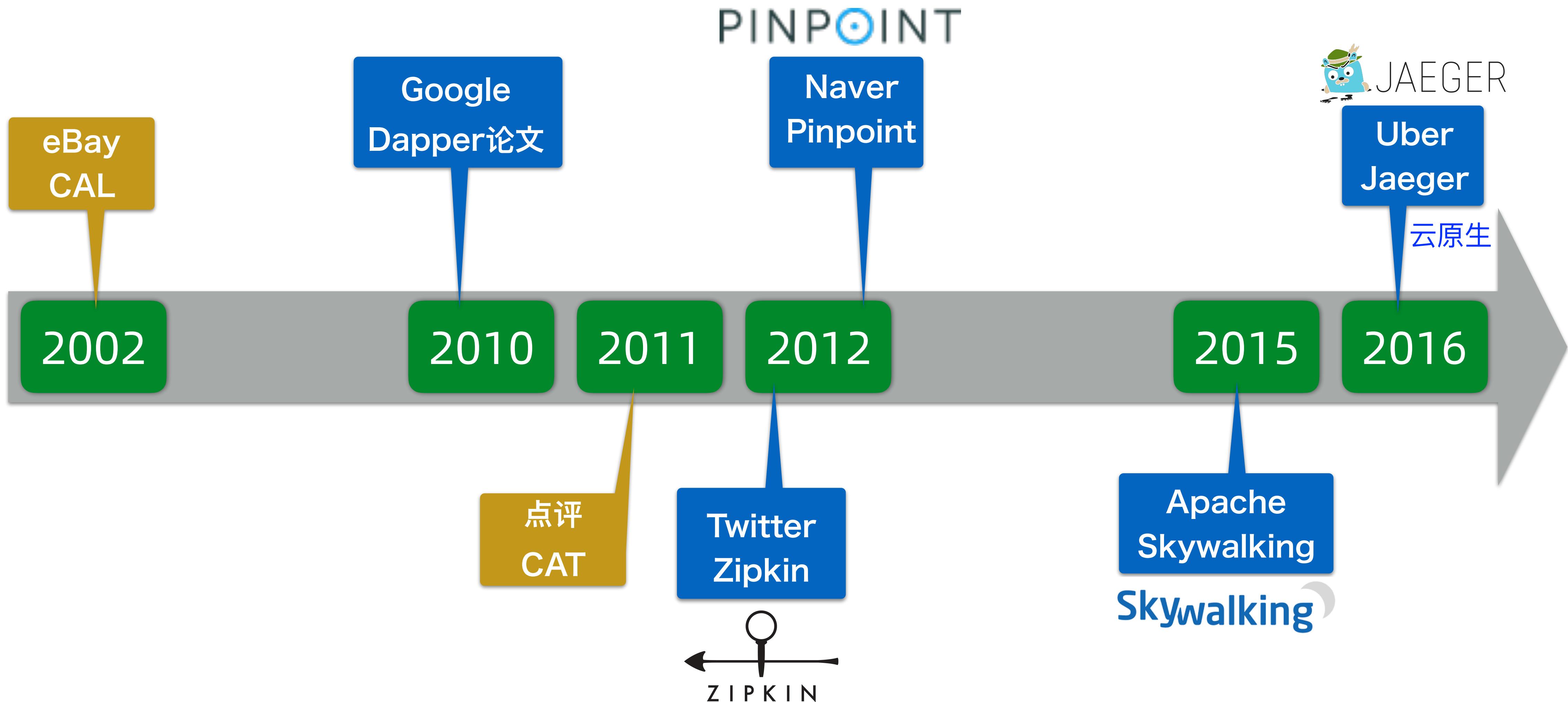
<https://github.com/sqshq/piggymetrics/tree/master/config>

第 4 部分

CAT vs Zipkin vs Skywalking

可以定位问题、排障，也可以监控性能

演进历史



CAT vs Zipkin vs Skywalking(1)

	CAT	Zipkin	Apache Skywalking
调用链可视化	有	有	有
聚合报表	非常丰富	少	较丰富
服务依赖图	简单	简单	好
埋点方式	侵入	侵入	非侵入，运行期字节码增强
VM指标监控	好	无	有
告警支持	有 告警模块	无	有
多语言支持	Java/.Net	丰富	Java/.Net/NodeJS/PHP自动 Go手动 手动埋点
存储机制	Mysql(报表), 本地文件/ HDFS (调用链)	可选in memory, mysql, ES(生 产), Cassandra(生产)	H2, ES(生产)

CAT vs Zipkin vs Skywalking(2)

一般需要定制开发一些插件

	CAT	Zipkin	Apache Skywalking
社区支持	主要在国内，点评/美团	文档丰富，国外主流	Apache支持，国内社区好
国内案例	点评、携程、陆金所、拍拍贷	京东、阿里定制不开源	华为，小米，当当，微众银行
APM	应用性能管理 Yes	No	Yes
祖先源头	eBay CAL	Google Dapper	Google Dapper
同类产品	暂无	Uber Jaeger, Spring Cloud Sleuth <small>spring对zipkin封装</small>	Naver Pinpoint
Github Stars(2019.6)	9.6k	11.2k	9.2k
亮点	企业生产级，报表丰富	社区生态好	非侵入，Apache背书
不足	用户体验一般，社区一般	APM报表能力弱	时间不长，文档一般，仅限中文社区

第三方包库，提供插件
其他不支持的需要自己定制插件

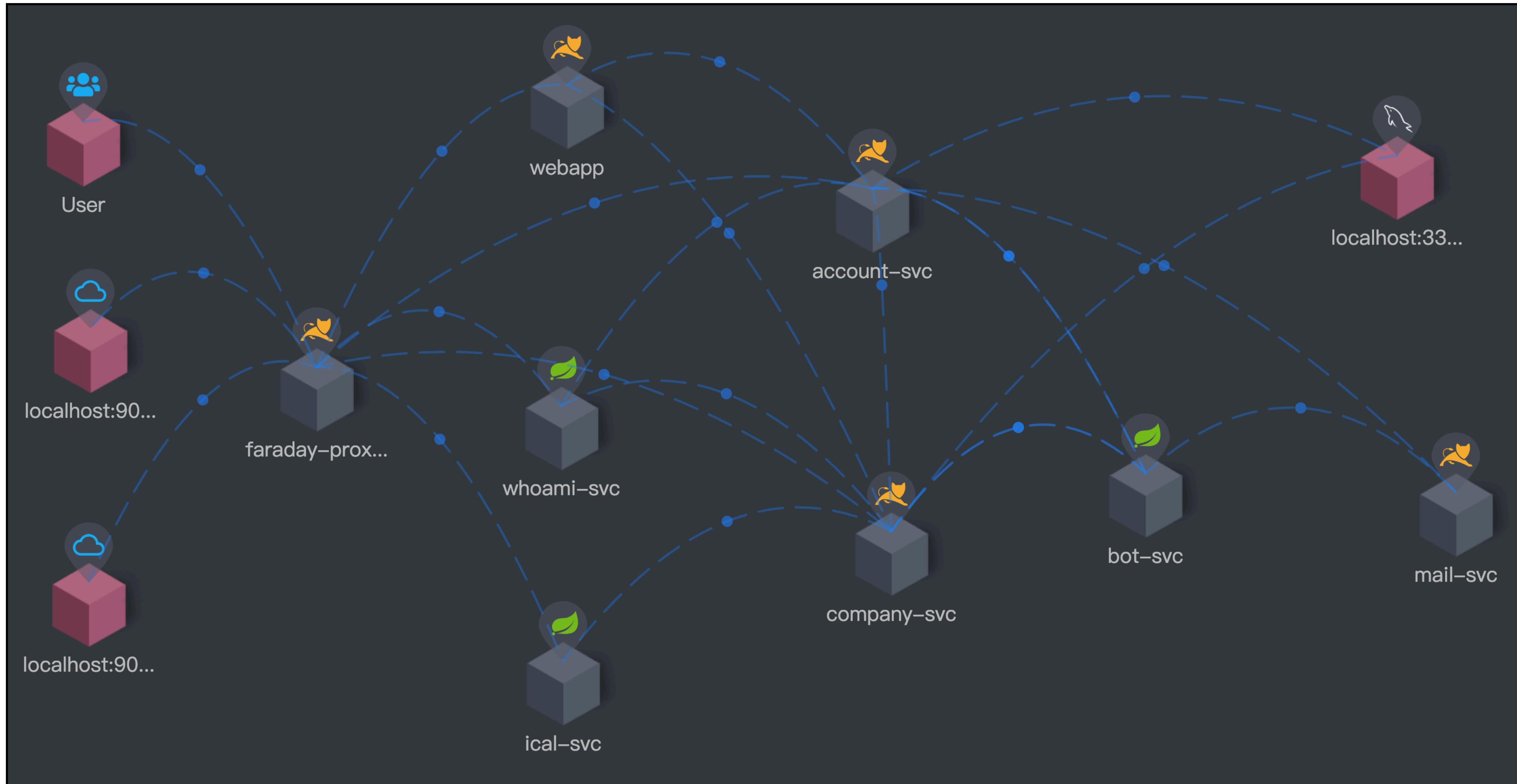
Skywalking Java Agent 支持库

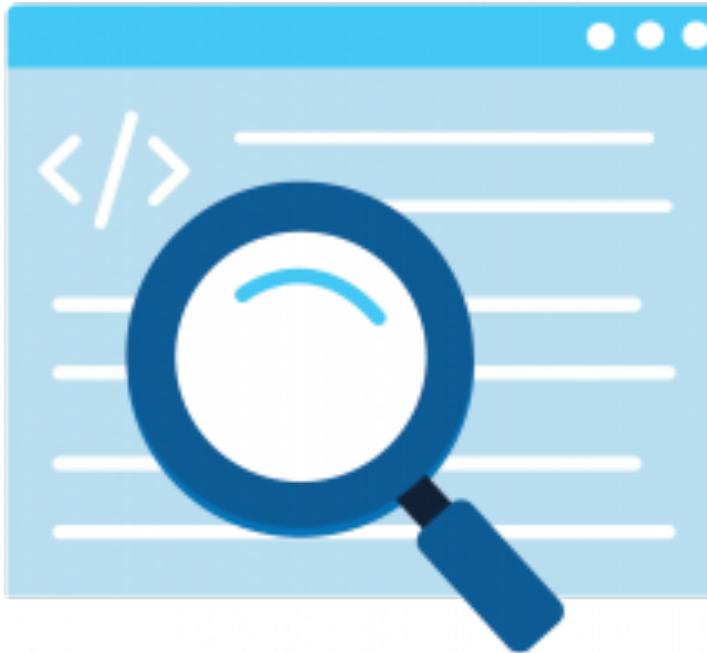
- HTTP Client
 - [Feign](#) 9.x
 - [Netflix Spring Cloud Feign](#) 1.1.x, 1.2.x, 1.3.x
 - [Okhttp](#) 3.x
 - [Apache httpcomponent HttpClient](#) 4.2, 4.3
 - [Spring RestTemplate](#) 4.x
 - [Jetty Client](#) 9
 - [Apache httpcomponent AsyncClient](#) 4.x
- HTTP Gateway
 - [Spring Cloud Gateway](#) 2.1.x.RELEASE (Optional²)
- JDBC
 - [Mysql Driver](#) 5.x, 6.x, 8.x
 - [Oracle Driver](#) (Optional¹)
 - [H2 Driver](#) 1.3.x -> 1.4.x
 - [Sharding-JDBC](#) 1.5.x
 - [ShardingSphere](#) 3.0.0, 4.0.0-RC1
 - [PostgreSQL Driver](#) 8.x, 9.x, 42.x
- RPC Frameworks
 - [Dubbo](#) 2.5.4 -> 2.6.0
 - [Dubbox](#) 2.8.4
 - [Apache Dubbo](#) 2.7.0
 - [Motan](#) 0.2.x -> 1.1.0
 - [gRPC](#) 1.x
 - [Apache ServiceComb Java Chassis](#) 0.1 -> 0.5,1.0.x
 - [SOFARPC](#) 5.4.0

- MQ
 - [RocketMQ](#) 4.x
 - [Kafka](#) 0.11.0.0 -> 1.0
 - [ActiveMQ](#) 5.x
 - [RabbitMQ](#) 5.x
- NoSQL
 - [Redis](#)
 - [Jedis](#) 2.x
 - [Redisson](#) Easy Java Redis client 3.5.2+
 - [Lettuce](#) 5.x (Optional²)
 - [MongoDB Java Driver](#) 2.13-2.14,3.3+
 - [Memcached Client](#)
 - [Spymemcached](#) 2.x
 - [Xmemcached](#) 2.x
 - [Elasticsearch](#)
 - [transport-client](#) 5.2.x-5.6.x
 - [Solr](#)
 - [SolrJ](#) 7.x
- Service Discovery
 - [Netflix Eureka](#)
- Distributed Coordination
 - [Zookeeper](#) 3.4.x (Optional² & Except 3.4.4)
- Spring Ecosystem
 - [Spring Bean annotations\(@Bean, @Service, @Component, @Repository\)](#) 3.x and 4.x (Optional²)
 - [Spring Core Async SuccessCallback/FailureCallback/ListenableFutureCallback](#) 4.x
- [Hystrix: Latency and Fault Tolerance for Distributed Systems](#) 1.4.20 -> 1.5.12

<https://github.com/apache/skywalking/blob/master/docs/en/setup/service-agent/java-agent/Supported-list.md>

Staffjoy依赖监控





第 5 部分

结构化日志和审计日志

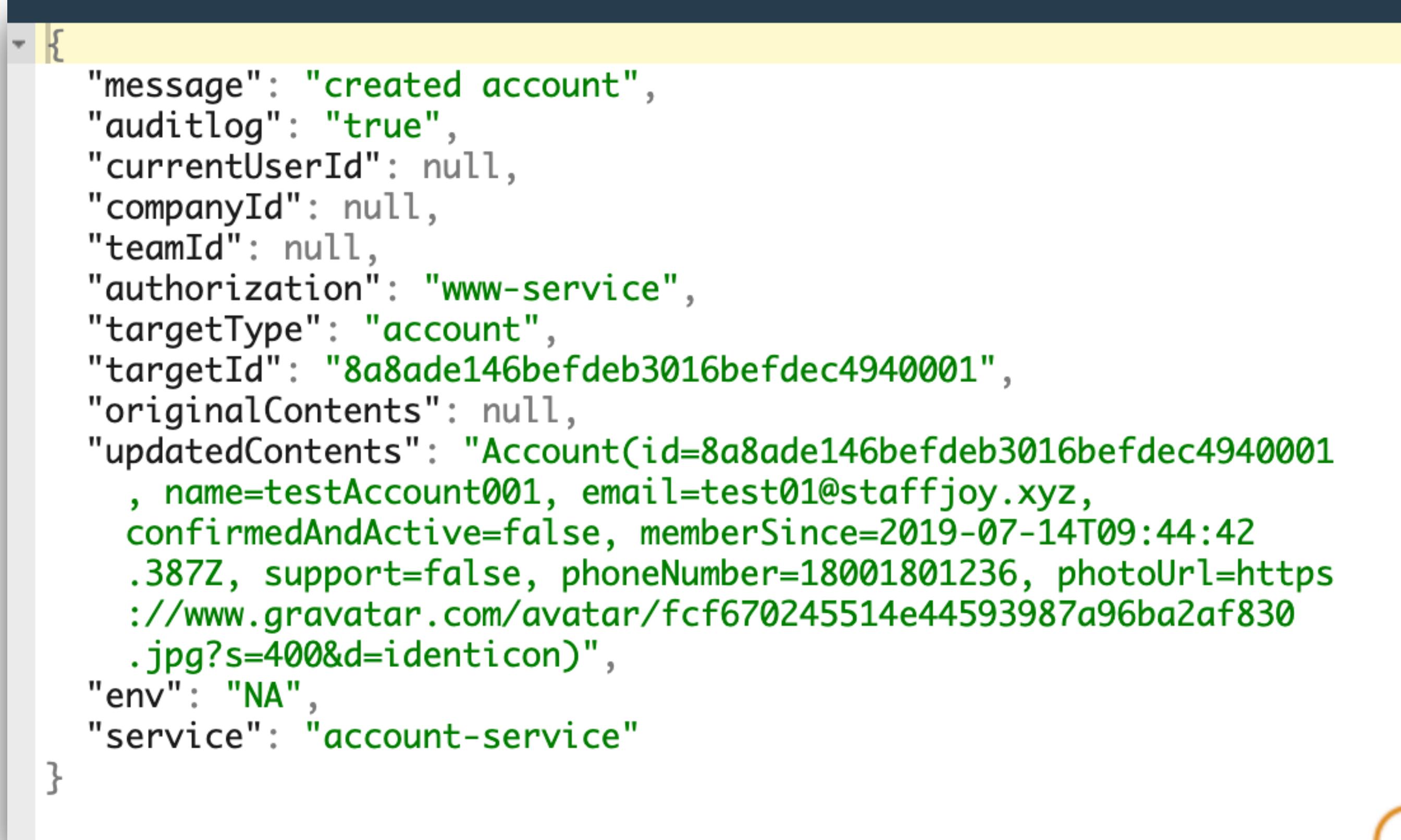
<https://github.com/jacek99/structlog4j>

java结构化日志示例

结构化和业务审计日志样例

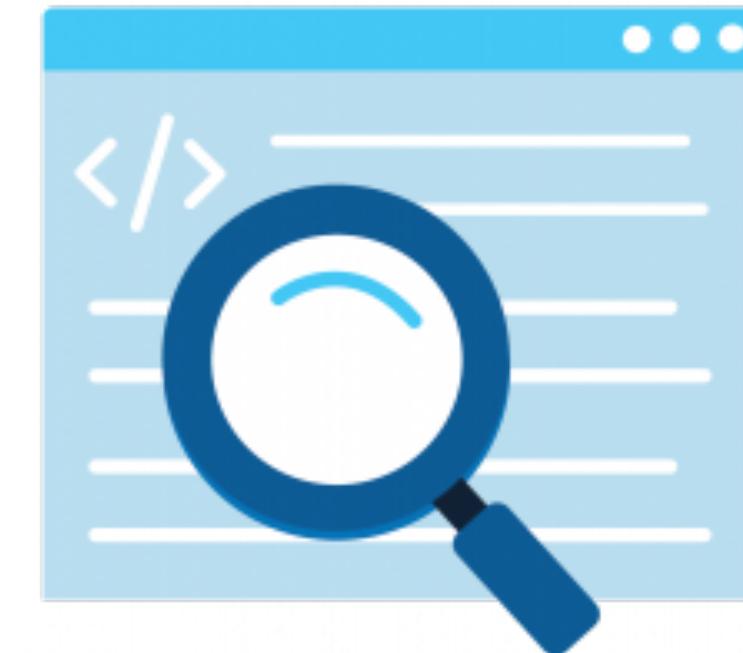
审计日志:一般把重要的操作用log记录下来, 后续去审计

如 account-svc
AccountService



```
{  
    "message": "created account",  
    "auditlog": "true",  
    "currentUser": null,  
    "companyId": null,  
    "teamId": null,  
    "authorization": "www-service",  
    "targetType": "account",  
    "targetId": "8a8ade146befdeb3016befdec4940001",  
    "originalContents": null,  
    "updatedContents": "Account(id=8a8ade146befdeb3016befdec4940001  
        , name=testAccount001, email=test01@staffjoy.xyz,  
        confirmedAndActive=false, memberSince=2019-07-14T09:44:42  
        .387Z, support=false, phoneNumber=18001801236, photoUrl=https  
        ://www.gravatar.com/avatar/fcf670245514e44593987a96ba2af830  
        .jpg?s=400&d=identicon)",  
    "env": "NA",  
    "service": "account-service"  
}
```

如 json 结构



第 6 部分

集中异常监控和Sentry

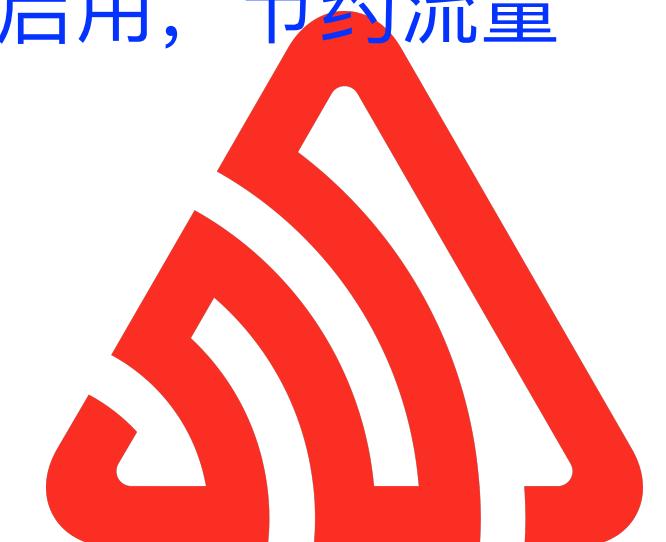
ELK: 常用日志架构

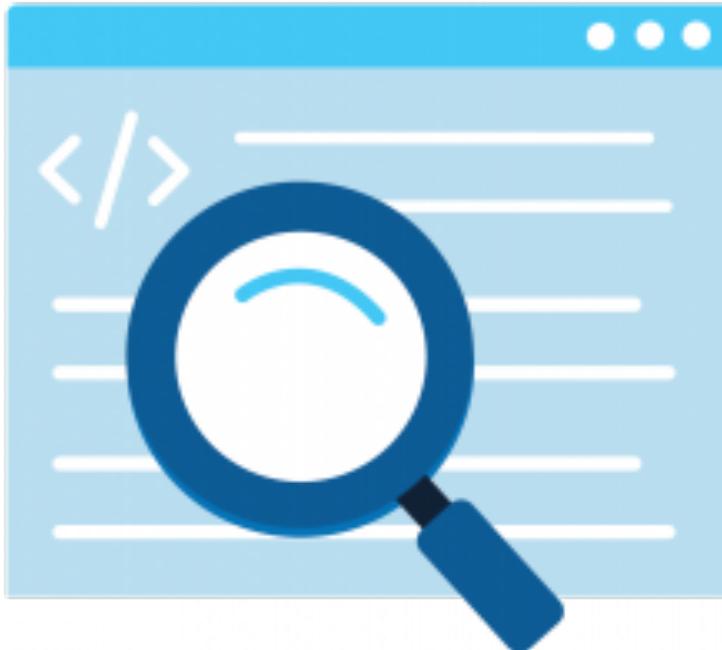
示例:acconut-svc 中 StaffjoyConfig
应用如 ServiceHelper的handleException()
可以监控异常的日志(相比一般的日志)

结合SentryClientAspect 部分环境不启用, 节约流量

<https://sentry.io>

<https://github.com/getsentry/sentry>



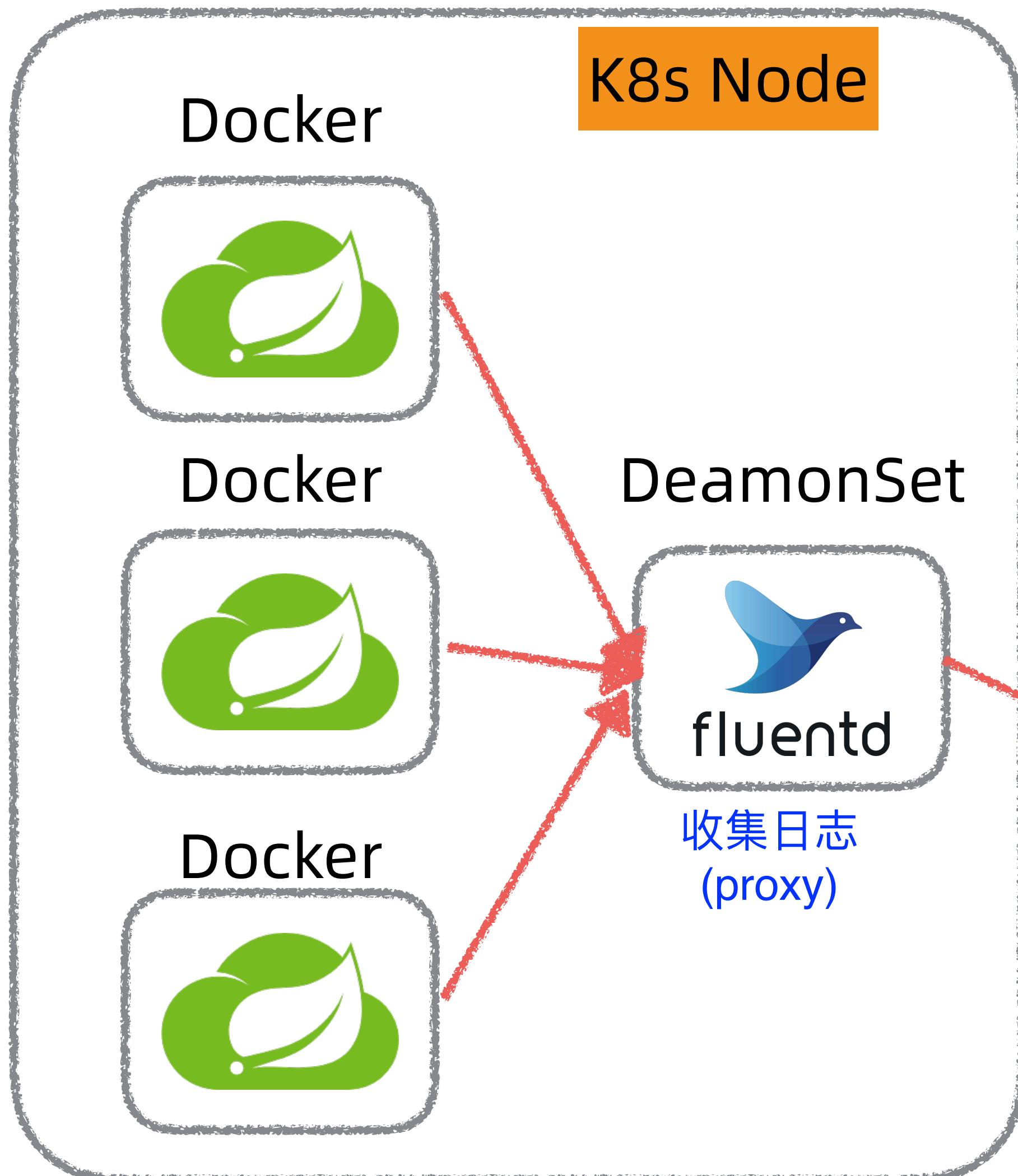


第 7 部分

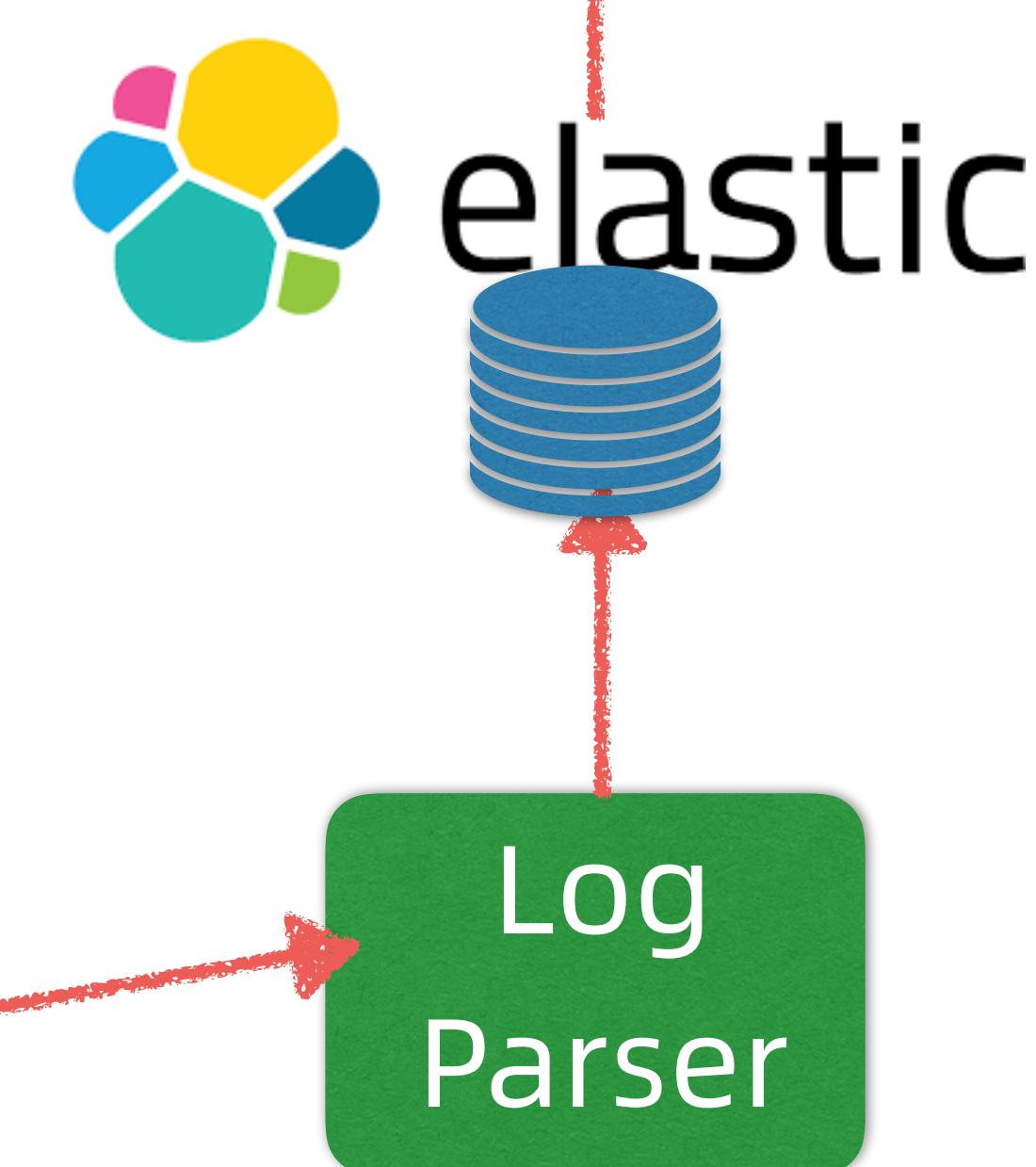
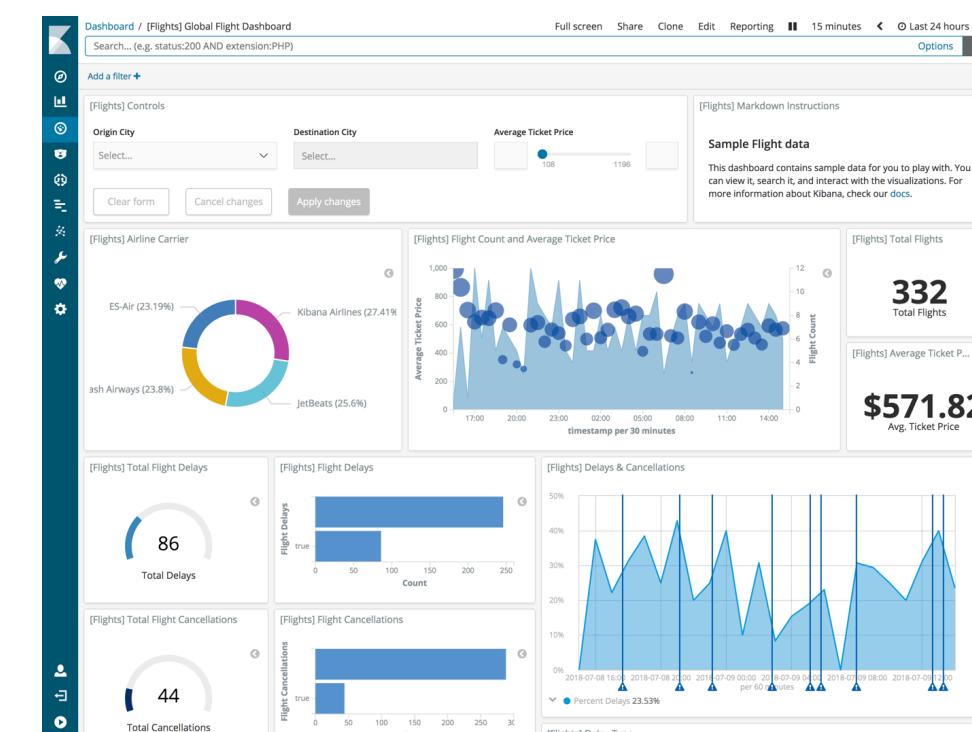
EFK & Prometheus & Skywalking + K8s 集成架构

及客时间

EFK + K8s

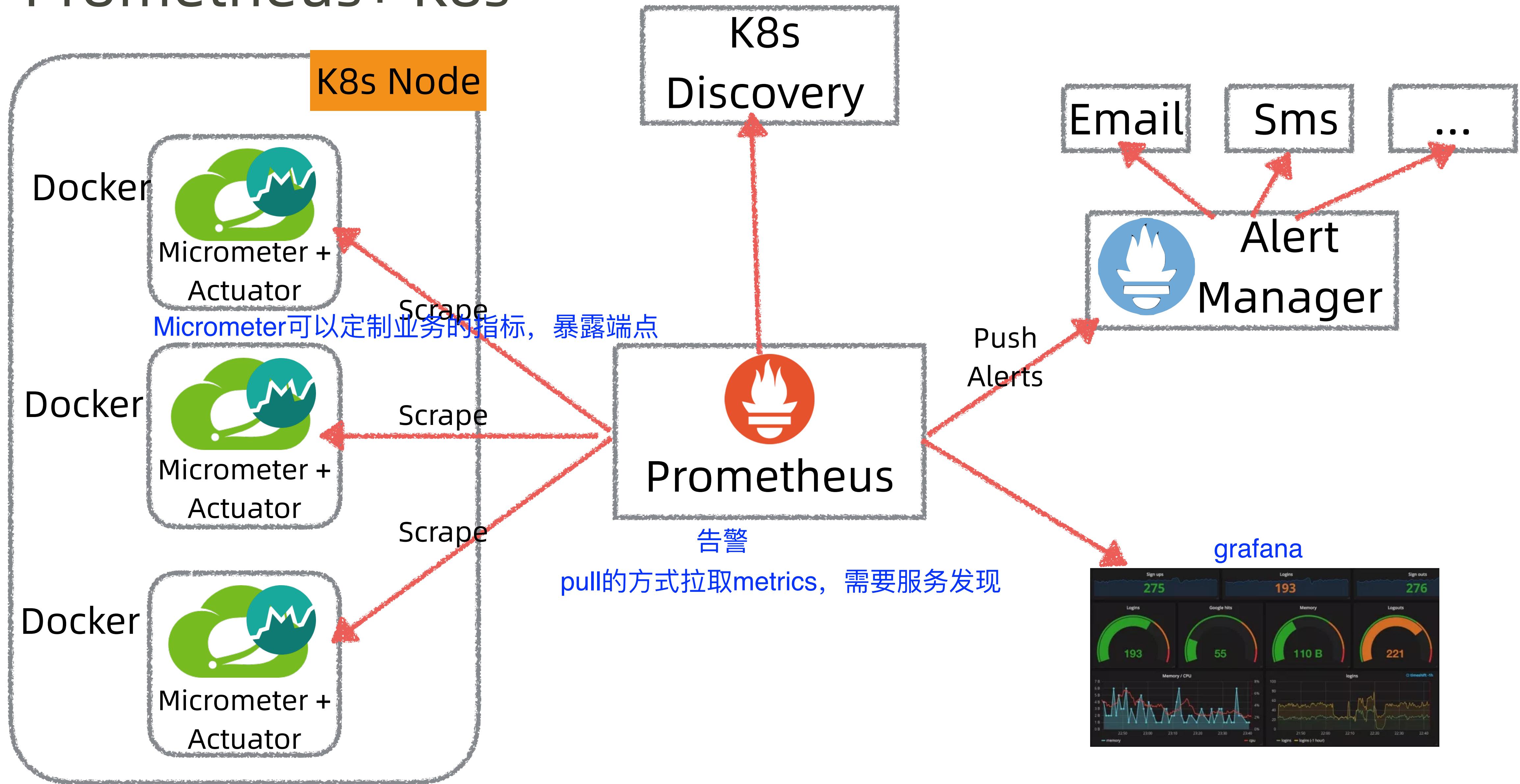


kibana

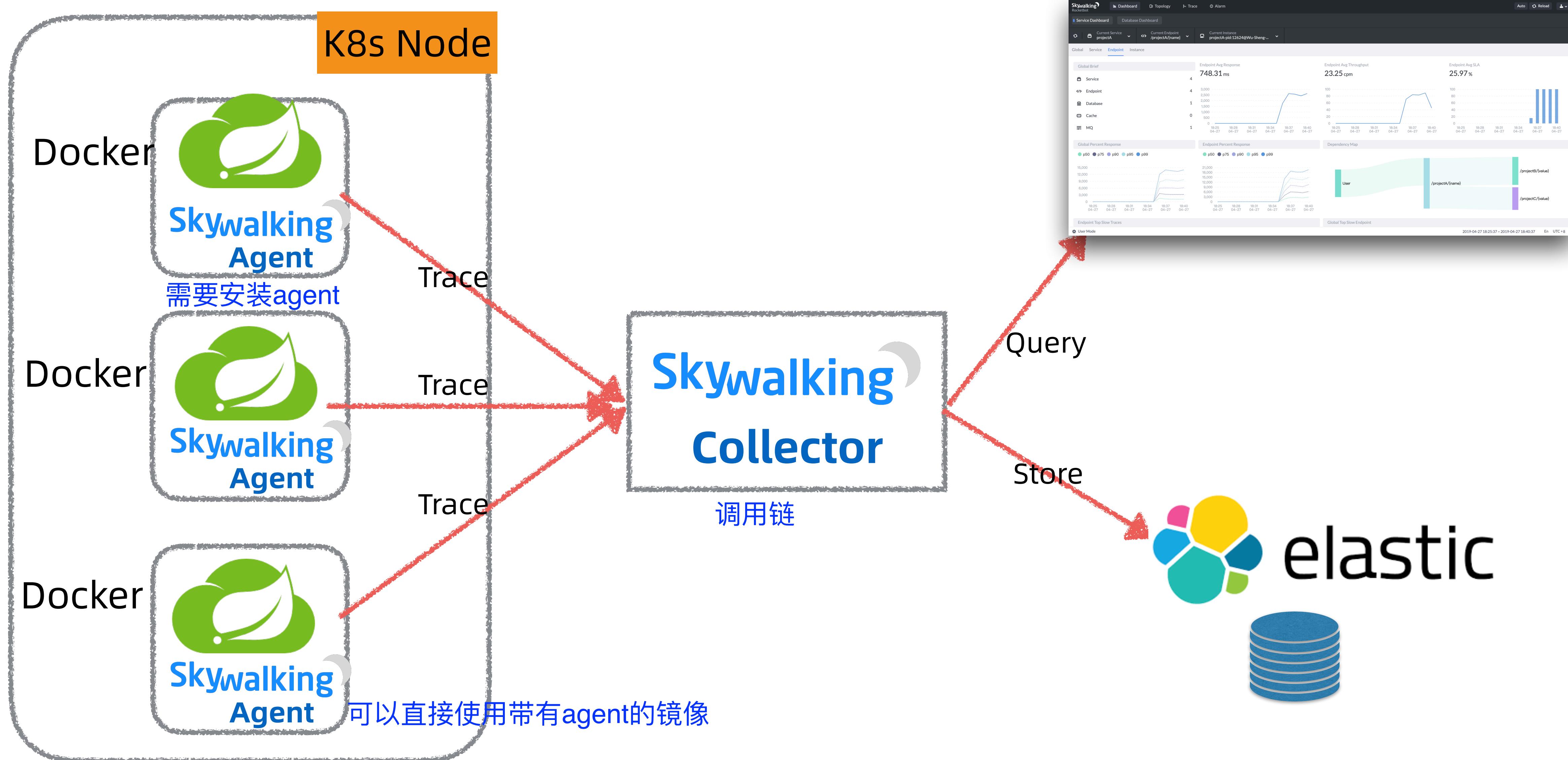


可以自己开发或者logstash
可以解析和过滤

Prometheus+ K8s



SkyWalking + K8s





扫码试看/订阅

《Spring Boot & Kubernetes 云原生微服务实践》