

爱奇艺基于 Docker 的容器云之 日志实践

杨成伟

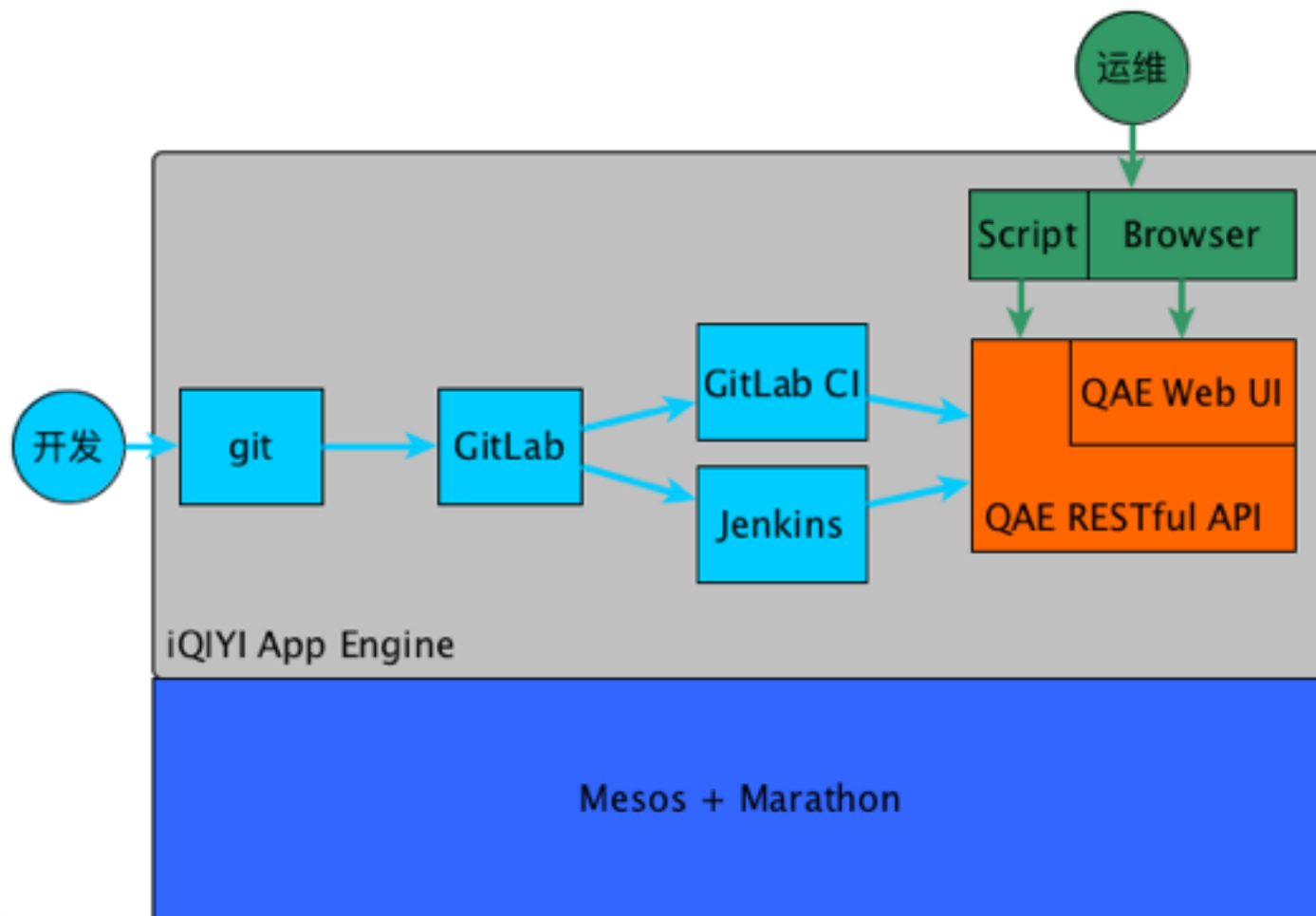
内容提要

- 爱奇艺容器云介绍
- iQIYI App Engine
- 日志
 - Mesos
 - Marathon
 - Docker
- QAE 早期方案
- 重新审视日志需求
- QAE 现行方案
- 总结

爱奇艺容器云介绍

- 基于 Mesos + Docker 生态
- 批处理任务平台 (Mesos + Sisyphus/Chronos + Docker)
 - 只提供 RESTful API 接口
 - 支撑公司大部分类型转码业务, 每周启动容器超过 200 万
 - 稳定运行近 2 年
- iQIYI App Engine (Mesos + Marathon + Docker)
 - WEB UI + RESTful API
 - 支撑公司其余转码业务及其它业务, 超过 200 个 app
 - QPS 近百万
 - 活跃开发中

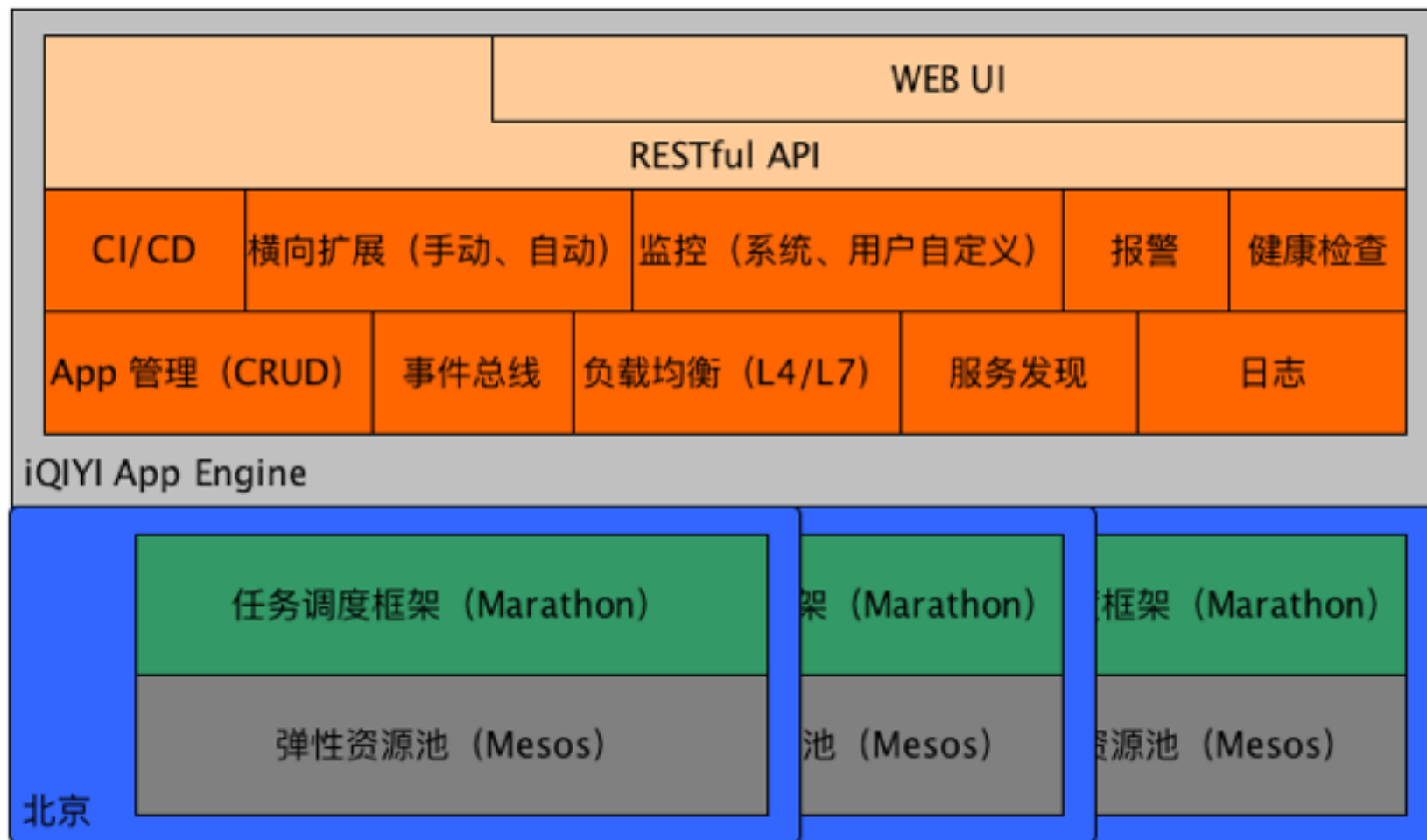
iQIYI App Engine——Scope



iQIYI App Engine——Features

- 基于 Mesos + Marathon + Docker
 - 高可用性、故障转移、健康检查、轻量、隔离、秒级启动.....
- 基于角色的组鉴权
- WEB UI + RESTful API
- 全面监控（计算节点、App、容器、进程、自定义）
- 订阅式报警
- 扩展（手动，自动）
- 服务发现、L4/L7 负载均衡
- 日志（实时，离线）等

iQIYI App Engine——Components



日志

- 写日志
 - print
 - log lib
 - 二次采集、存储：fluentd、flume、logstash 等
- 读日志
 - 原始日志
 - 脚本处理
 - 搜索 tag、全文搜索
 - 统计、分析、呈图报表
 - ELK、Kafka、Hadoop、Spark、Storm 等

Mesos、Marathon 和 Docker 怎么看？

日志——Mesos

- 提供一个 sandbox
- 捕获 stdout & stderr
- 以 WebSocket 的方式查看文件
 - 自动滚动 (tail -f)
 - 实时加载
- 对 Docker 容器内无感知
- 提供 HTTP API 浏览、查看、下载文件

mode	nlink	uid	gid	size	mtime		
-rW-r--r--	1	root	root	170 B	Aug 30 14:30	stderr	Download
-rW-r--r--	1	root	root	2 KB	Aug 30 14:30	stdout	Download

日志——Marathon

日志? 你找 Mesos?

日志——Docker

- 捕获 stdout & stderr
- docker logs
- 支持多种 log driver
 - none
 - json-file
 - syslog
 - journald
 - gelf
 - fluentd
 - awslogs
 - 等

日志

有什么问题？

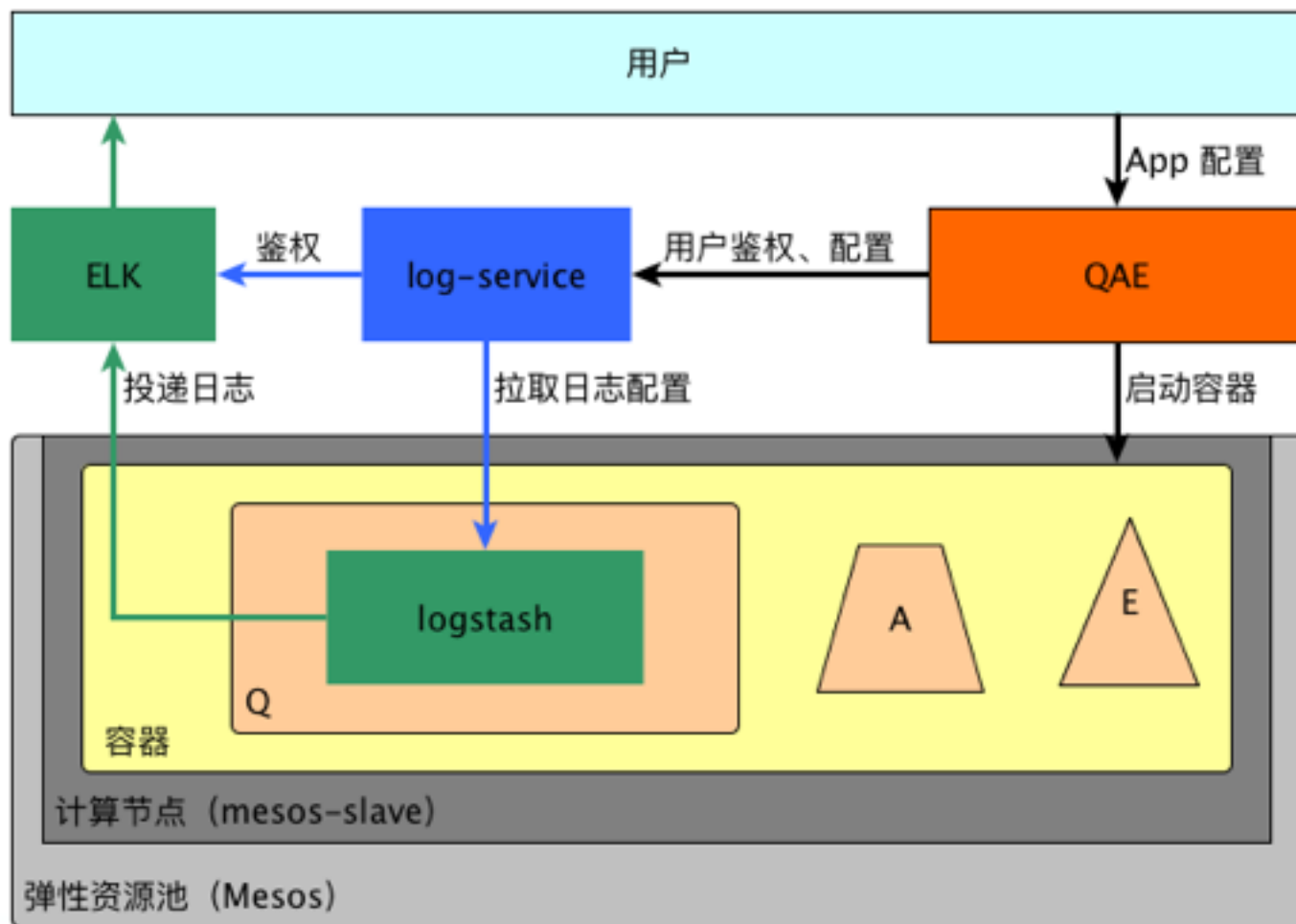
日志

只有 stdout & stderr

日志——stdout & stderr 为什么不能感动用户

- 应用日志并没有输出到 stdout & stderr, 输出到日志文件了
 - PaaS 平台自动重定向到 stdout & stderr?
- 日志不仅在 1、2 个文件中
 - access
 - error
 - api.log
 - moduleA.log、moduleB.log.....
 - processA.log、processB.log.....
 - 重定向再分拆就不那么优雅了
- 不想改代码

QAE 早期方案



QAE 早期方案——用户反响

- 延时好长，有时候得几分钟
- 没有原始日志?

QAE 早期方案——总结

- 容器内采集占用资源多，Logstash 笨重得超乎想象
- Logstash 采集规则学习成本高
- 只能提供离线日志方案，延时长
- 不能解决原始日志需求

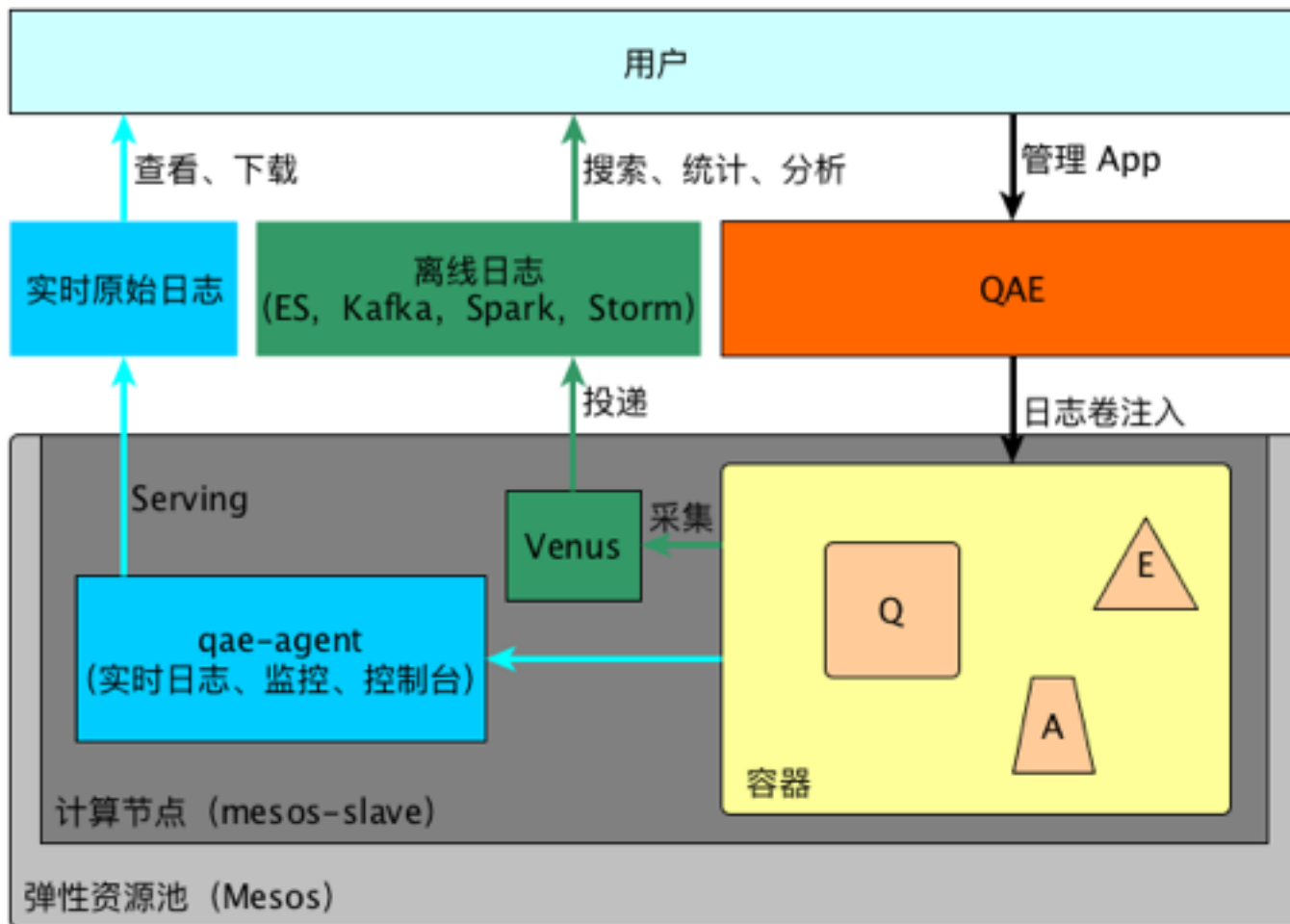
重新审视日志需求——客观需求

- 强需求
 - 实时日志，业务故障了，火烧眉毛
 - 原始日志，怀疑日志搬运工，是不是搬错了
 - 可查看，下载，方便快速定位，一眼看不出来再用脚本查找
- 弱需求
 - 聚合
 - 索引、搜索：标签、全文
 - 统计、分析
 - 呈图、报表

重新审视日志需求——用户主观需求

- 不改代码，没工夫理你
 - 新需求
 - 旧 bug
 - 产品经理
- 多个日志文件，而不是 1、2 个
 - 日志文件分类，易于处理
 - 我代码拆开太难，但是又想用 docker

QAE 现行方案



QAE 现行方案——用户反响

这还差不多

总结

- 较好的解决了用户对日志的需求
- 用户很难忽悠、很难伺候
- 用户并不是很愿意为新技术买单
- 应用由自上而来的业务驱动，而非自下而来的技术驱动
- 公有云的方案并不一定适合私有云
- 在技术和用户需求之间妥协
- Docker：为什么你说不能在里面跑多个应用？

THANKS!

