



GOPS2018
Shenzhen

GOPS

全球运维大会 2018

2018.4.13-4.14

中国·广东·深圳·南山区 圣淘沙大酒店（翡翠店）





GOPS2018
Shenzhen

从如履薄冰到纵横捭阖

——唯品会运维生态链打通实践

王俊峰



GOPS2018
Shenzhen

目录



1

曾经面临的问题

2

标准化建设之路

3

打通生态，全面赋能

4

几点感悟



GOPS2018
Shenzhen

面临的问题

业务线复杂，技术栈不统一

不同人员面对不同业务线，盲区非常多，难以人力共享

各种运维平台都很完备，但是散兵作战没有形成集团军

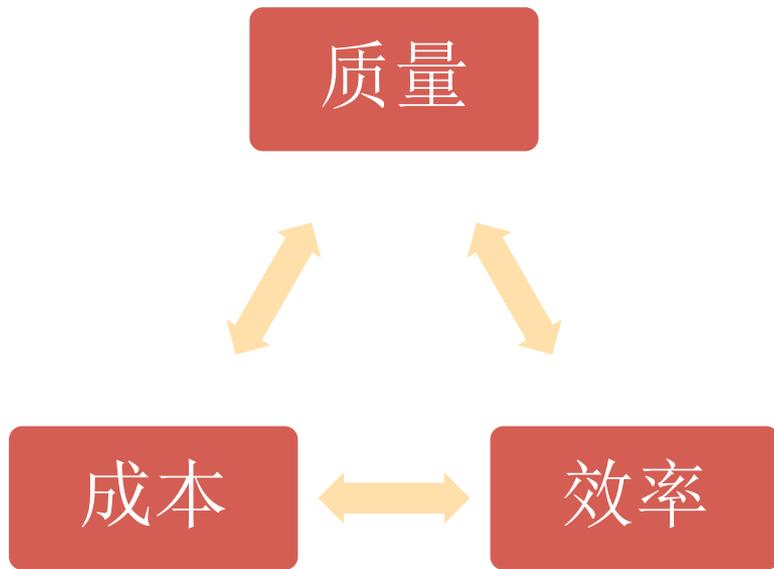
平台建设要考虑各种特殊场景，甚至妥协



GOPS2018
Shenzhen

我们的反思

1. 技术人如何确保质量、成本、效率三者的平衡？
2. 我们建设了那么多的工具平台，运维人员为何依然疲惫不堪？
3. 运维平台建设蒙面狂奔时如何不忘初心？



出路？



GOPS2018
Shenzhen

1. 平台建设者必须深刻理解运维痛点。
2. 平台建设的技术选型不是最重要的。
3. **标准化程度**决定了运维自动化的上限。
4. 自动化的水平决定了运维新的增长点。

标准化迫在眉睫！





GOPS2018
Shenzhen

目录

1 曾经面临的问题

➔ 2 标准化建设之路

3 打通生态，全面赋能

4 几点感悟



GOPS2018
Shenzhen

技术组件思想的提出

纵向是组件

最底层硬件、操作系统、各类应用基础软件（nginx、haproxy、zookeeper等）、应用框架等都可以拆分成一个个组件。有助于组件服务化和组件研究的技术深度。

横向是流程

运维相关的发布流程、变更流程、故障处理流程、问题跟踪流程等。这些流程就像线来串联各项工作和组件。

组件思想打破了**运维绑定业务线**的工作模式

组件思想奠定了标准化基础



GOPS2018
Shenzhen

技术成长

组件专家组负责，专家组明确组件的发展方向，探索最佳实践，有利于技术积累和人员的技能成长。

消除业务壁垒

业务流程之间的差异会逐步缩小，运维人员面对的是每个标准的组件。

组件服务化

运维人员对外转型技术输出，提供服务化的产品，开发仅按照标准api使用，无需关注底层细节。

标准化目标



GOPS2018
Shenzhen



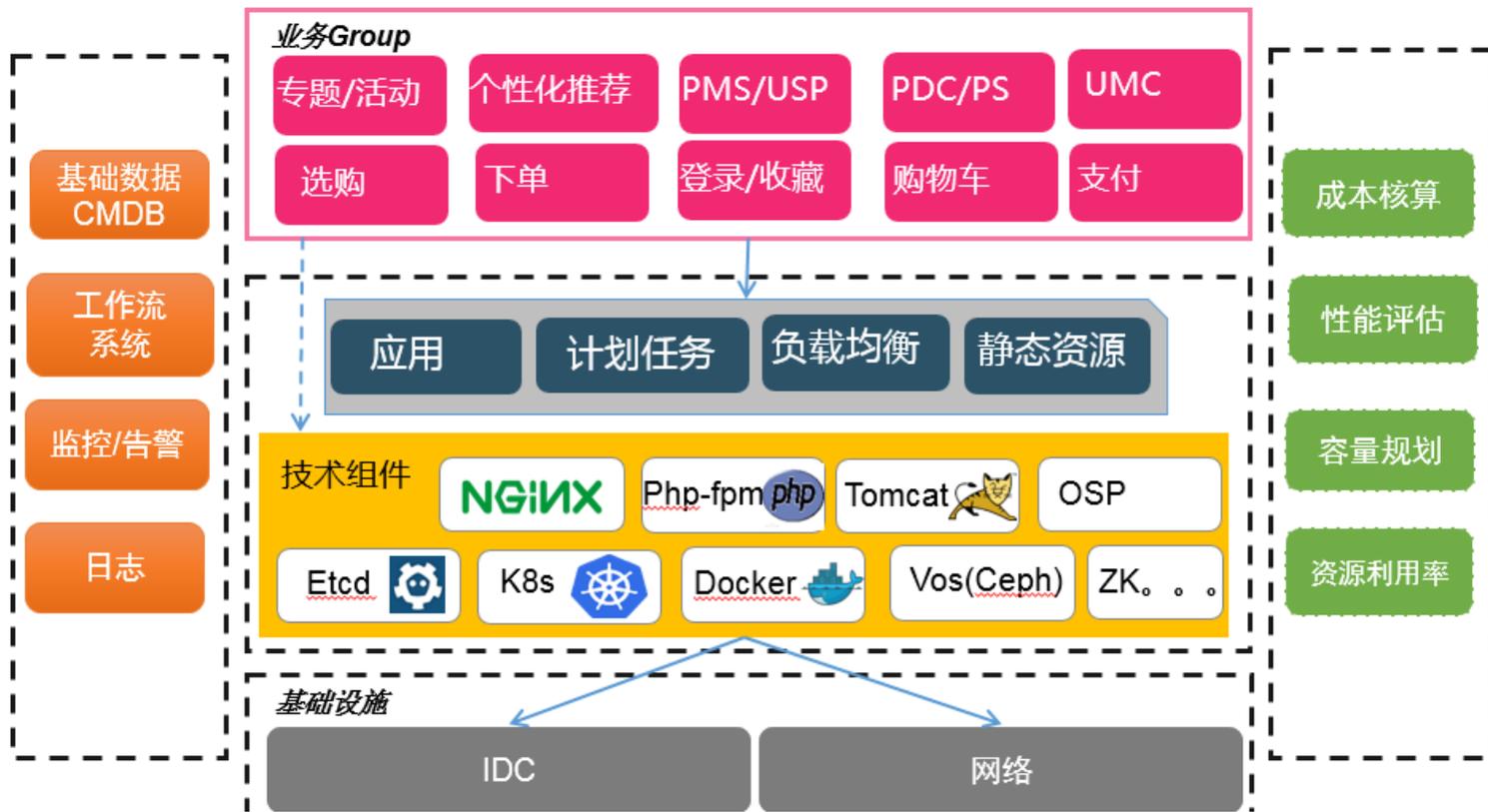
运维标准化





GOPS2018
Shenzhen

标准化体系





GOPS2018
Shenzhen

配置标准化——分层治理

Janitors平台

目标：前置业务开发维护

业务域配置

环境变量web_env.sh

Tomcat/Php/OSP应用
配置

组件配置

基础组件

Agent组件

Puppet

保持配置标准统一

OS层配置

硬件层

C
M
D
B

监控标准化



GOPS2018
Shenzhen

曾经的监控是这样的。。。。

应用运维

- zabbix
- Nagios
-

大数据

- zabbix
- Nagios
-

DBA

- zabbix
- Nagios
-

统一！快！精准！

自动化、监控插件标准化

监控视图定制、充分发挥数据价值

为开发赋能、系统可扩展



GOPS2018
Shenzhen

监控标准化目标拆分

事件来源即CMDB
应用信息标准化
(应用类型、基础
组件等)

监控模板标准化。
专家组负责技术
组件的监控设计、
阈值设置。
监控模板统一的
版本控制。

告警规则标准化。
监控系统与告警
系统分离，各司
其责。告警根据
设备等级、应用
等级、严重等级
区分。告警发送
对象统一来源：
CMDB。

重新设计监控系统——VIPFalcon



GOPS2018
Shenzhen

- 基于openfalcon二次开发
- 2万+监控节点
- 500万+metric
- 重新开发数据聚合
- 落地hive数据分析



GOPS2018
shenzhen

新老监控体系对比

	VipFalcon	Zabbix
架构	分布式架构，核心组件（Transfer，Judge、Graph、hbs和api）可以根据瓶颈水平扩展，解决需要部署多套Zabbix的痛点。	主要功能都集中在server端，server端成为性能瓶颈，不易水平扩展，数据同步导致延迟。
采集扩展	灵活的Plugin方式，面向HostGroup维度，通过Git进行维护	以Host维度支持自定义脚本，没有版本管理，不易维护
存储	支持适于时间序列数据存储的RRD和OpenTSDB，易扩展，读写性能高于关系型数据库。	数据存储在mysql中，数据量大时，读写性能是瓶颈
管理	以HostGroup为维度，支持继承和覆盖，易于维护	模板以Host为维度，不易维护
运维生态	打通变更和发布系统，屏蔽变更和发布过程中的无效告警，打通CMDB	没有打通CMDB，变更和发布系统
编程语言	Go语言，良好的高并发性能，语法简单，有利于二次开发	C语言，不利于二次开发



GOPS2018
Shenzhen

监控标准化的效果

对内提升质量

- 提升效率
- 降低维护成本
- 提升用户体验
- 控制风险



对外价值输出

- 打通运维体系
- 为开发赋能
- 精细运维/运营
- 业务成本核算



GOPS2018
Shenzhen

变更标准化

风险矩阵SDK

- 变更对象和技术风险两个维度评估
- 变更对象以CMDB存储为准

标准变更模板库

- 原子组件变更模板由组件专家主负责
- 标准变更模板最终固化、沉淀专家技术、经验，既是技术文档，又是变更单，在变更流程中平台化



GOPS2018
Shenzhen

目录

1 曾经面临的问题

2 标准化建设之路

➔ 3 打通生态，全面赋能

4 几点感悟

打通运维生态



GOPS2018
Shenzhen

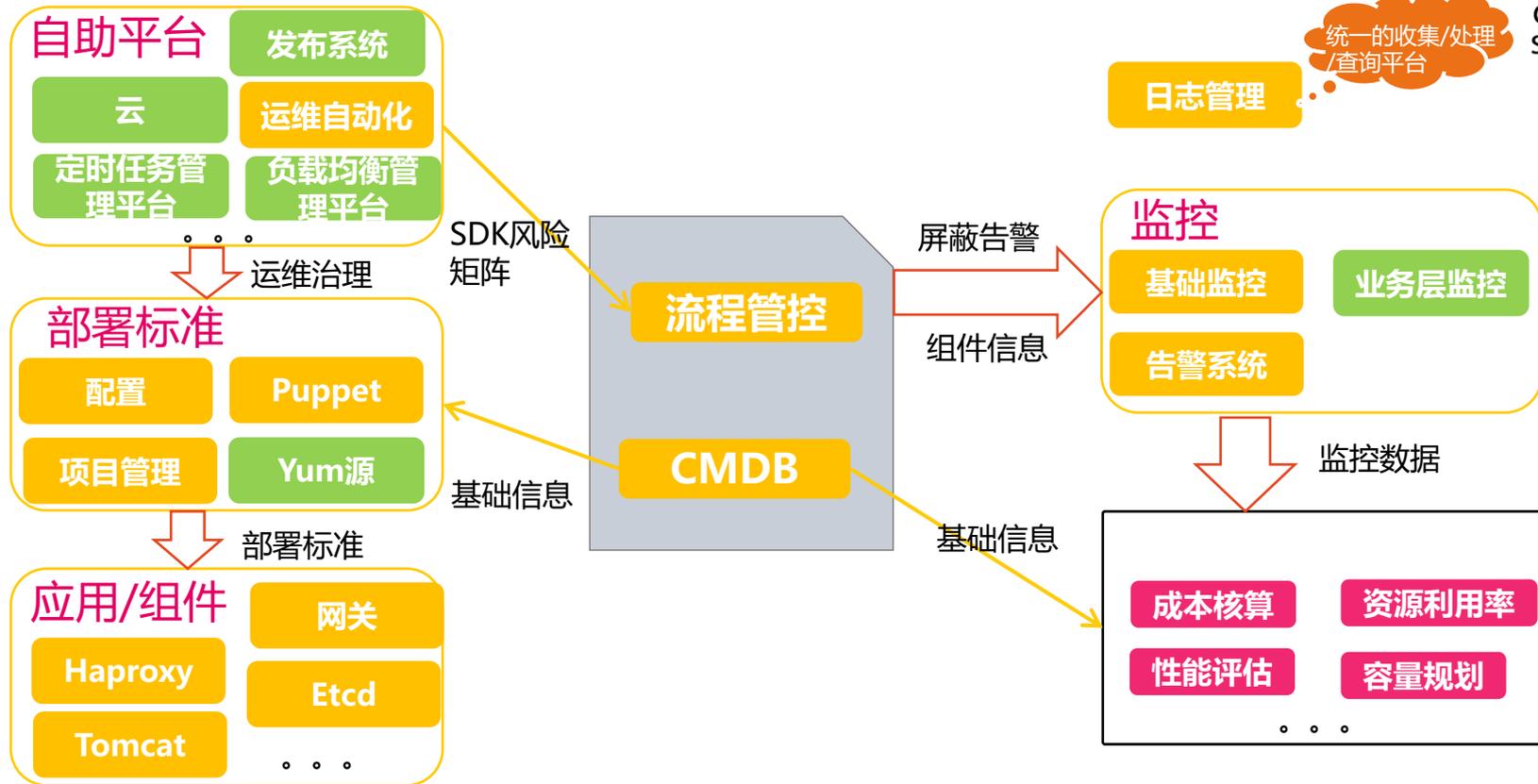
运维流程为核心



运维流程+CMDB
为核心



整体架构





GOPS2018
Shenzhen

各个击破，先从变更开始打通

Devops变更失控
变更流程不遵守

风险评估不准确
效率和流程总是
冲突

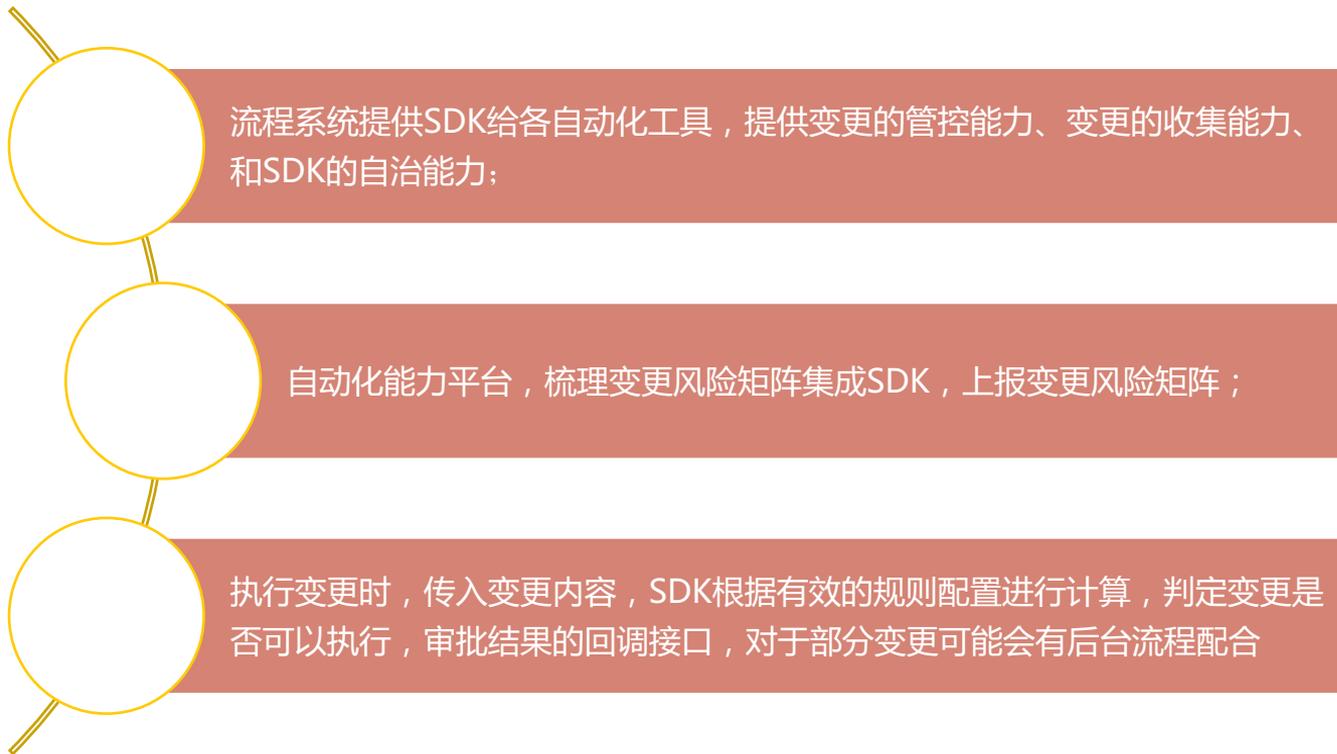
变更质量控制困难





GOPS2018
Shenzhen

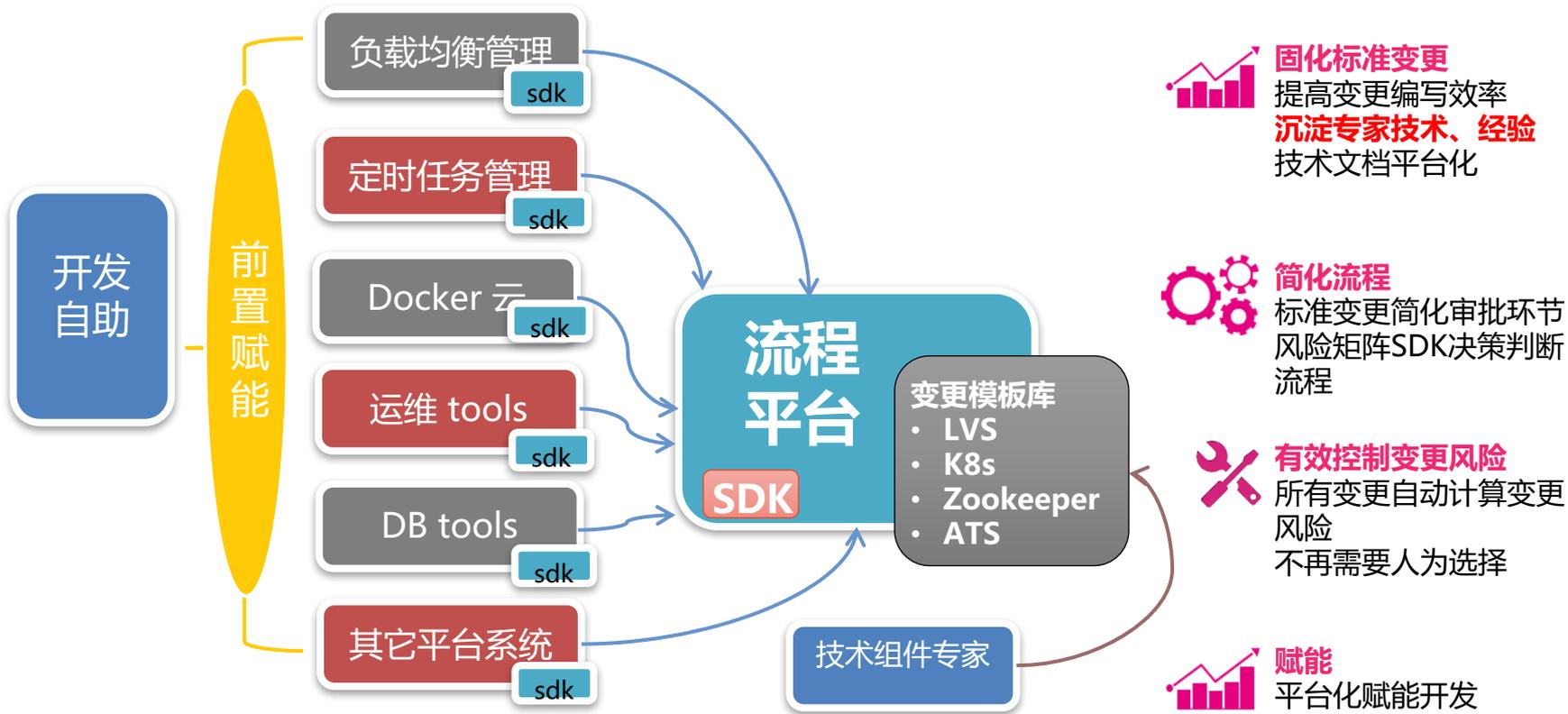
变更打通设计思路





GOPS2018
Shenzhen

变更打通，赋能工具



固化标准变更
提高变更编写效率
沉淀专家技术、经验
技术文档平台化



简化流程
标准变更简化审批环节
风险矩阵SDK决策判断
流程



有效控制变更风险
所有变更自动计算变更
风险
不再需要人为选择

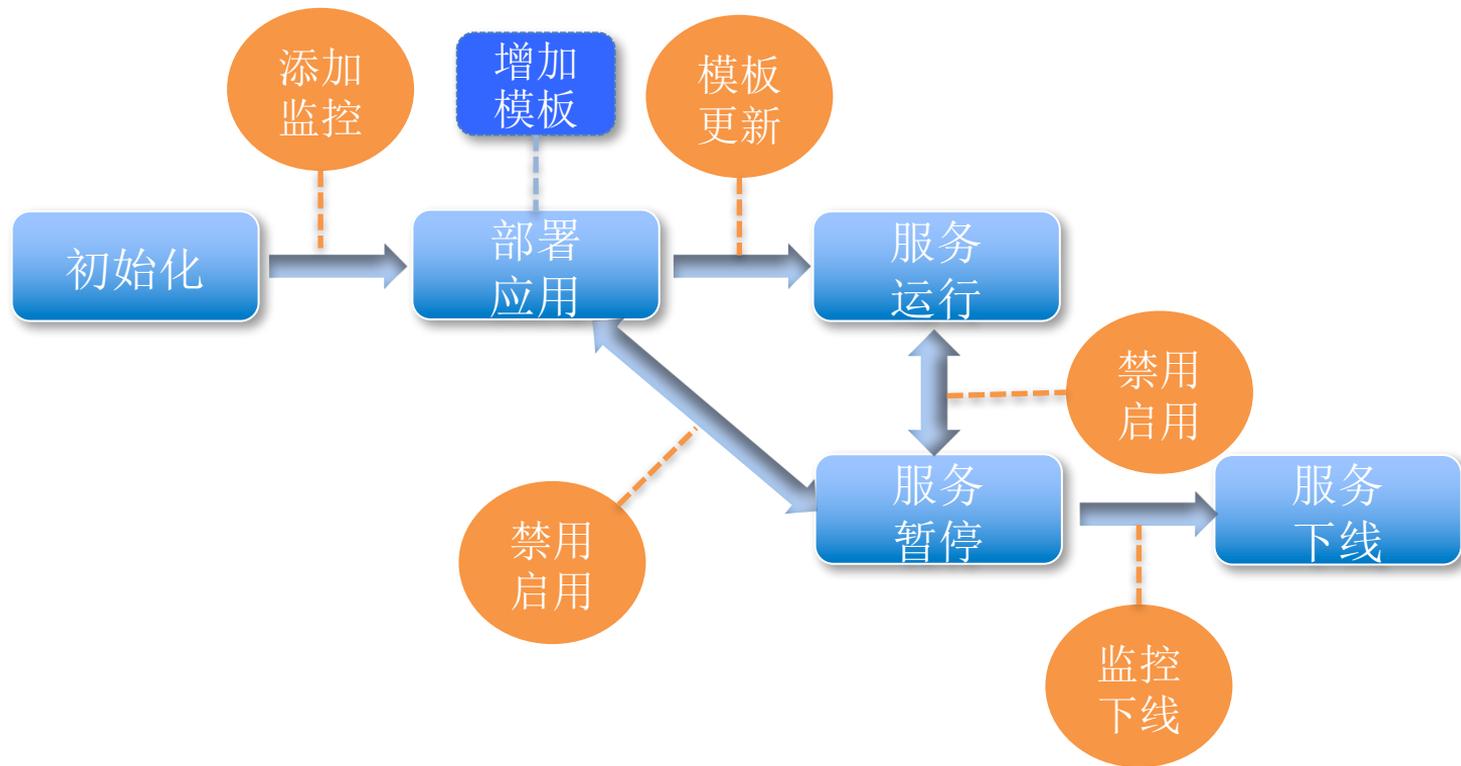


赋能
平台化赋能开发



GOPS2018
Shenzhen

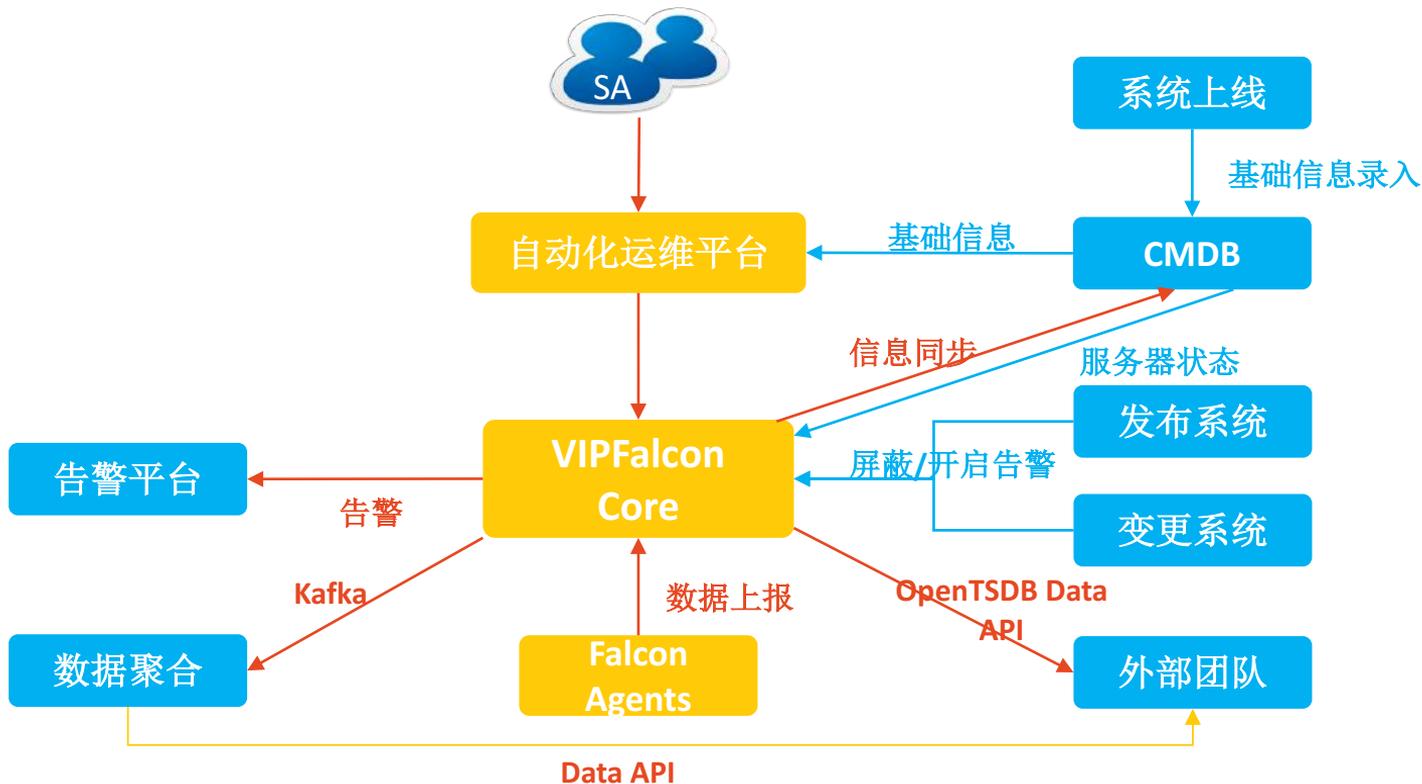
监控打通实现自动化





GOPS2018
Shenzhen

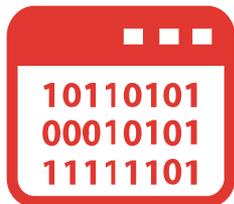
监控与外部系统对接



连接一切后产生了什么



GOPS2018
Shenzhen



全面的集中的数据



效率和流程管控

再回首



GOPS2018
Shenzhen

质量

- 转型平台建设者
- 自动化到智能化

效率

- 降低门槛
- 效率提升
- 流程管控

成本

- 成本核算
- 容量评估
- 服务器选型
- 应用画像



GOPS2018
Shenzhen

目录

1 曾经面临的问题

2 标准化建设之路

3 打通生态，全面赋能

➔ 4 几点感悟



GOPS2018
Shenzhen

几点感悟

- 有了标准化的土壤，自动化智能化才能生根发芽。
- 标准化需要强有力的领导、明确的方法论。

- DEVOPS建设要有业务视角。
- 单一兵种需要整合成集团军才能发挥战斗力。



GOPS2018
Shenzhen



Thanks

高效运维社区
开放运维联盟

荣誉出品



GOPS2018
Shenzhen

想第一时间看到高效运维社区
的新动态吗？

