

# 微服务下的APM全链路监控

# Agenda

- 谈谈应用性能管理 ( APM )
- **微**服务对APM的**大**影响
- 如何构建适于微服务的APM平台
- 打造监控、告警和报障的闭环
- 对未来的一些思考

# Agenda

- 谈谈应用性能管理 ( APM )
- **微**服务对APM的**大**影响
- 如何构建适于微服务的APM平台
- 打造监控、告警和报障的闭环
- 对未来的一些思考

# 什么是APM

APM (Application Performance Management) 即应用性能管理，属于IT运维管理 (ITOM)范畴。主要是针对企业关键业务的IT应用性能和用户体验的监测、优化，提高企业IT应用的可靠性和质量，保证用户得到良好的服务，降低IT总拥有成本(TCO)。

APM主要特征列表

模块	主要内容
多级应用性能监控	覆盖通讯协议1-7层，通过事务处理过程监控、模拟等手段实现端到端应用监测
应用性能故障快速定位	对应用系统各个组件进行监测，迅速定位系统故障，并进行修复或提出修复建议
应用性能全面优化	精确分析各组件占用系统资源的情况，并根据应用系统性能要求给出专家建议

# APM的发展历程



# Gartner对APM的定义(2014)

维度	描述
终端用户体验	图形化或者数据化展示最终用户在使用应用的可用性、时延、交易执行等用户视角体验。
实时应用架构发现、建模、展示	实时发现应用相关的硬件、软件组件，并对这些组件之间的交互进行展现，以及对实时交易的的路径进行清晰的图形化展现。
用户交易追踪	记录用户所有的访问行为，并根据相应的事件追踪用户的访问在架构中的路径，以及交易在路径中每个节点的性能表现
代码级深度性能诊断	深度代码诊断和分析，跟踪记录每个交易的代码级上下文关联，以及对于资源的消耗
贯穿生命周期的性能管理	性能管理必须贯穿开发、测试和运维各个环节，上述环节的相关人员能够使用统一的标准对应用性能进行监测、分析和诊断。



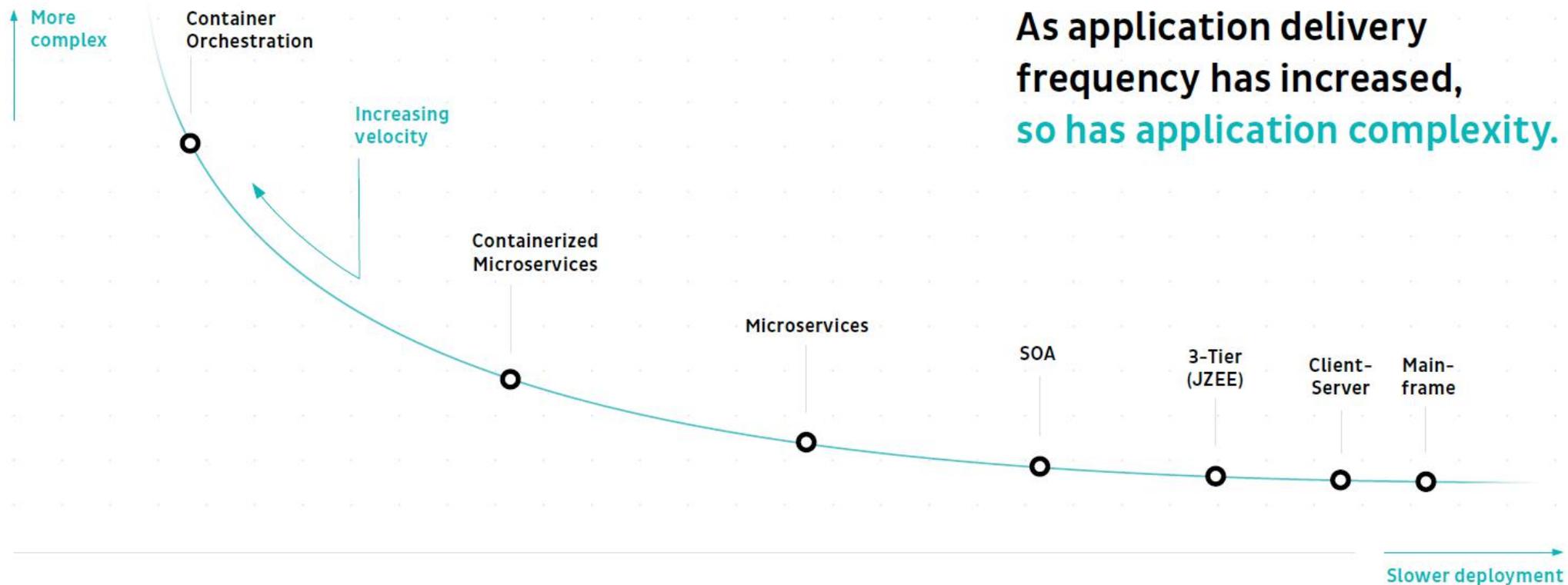
# Gartner对APM的定义(2016)

维度	描述
<b>DEM</b> -Digital experience monitoring	数字体验监控，浏览器及移动设备用户体验监控及利用主动拨测的实现的业务可用性及性能监控
<b>ADTD</b> -Application discovery, tracing and diagnostics	应用自动发现、追踪和故障诊断，自动发现应用之间的逻辑关系，自动建模、应用组件的深入监控及性能关联分析
<b>AA</b> -Application analytics	应用分析，通过机器学习，进行针对JAVA及.NET应用的根源分析

# Agenda

- 谈谈应用性能管理 ( APM )
- 微服务对APM的大影响
- 如何构建适于微服务的APM平台
- 打造监控、告警和报障的闭环
- 对未来的一些思考

# 服务开发架构的发展历程



As application delivery frequency has increased, so has application complexity.

# 微服务带来的挑战

- 依赖关系复杂
- 持续交付
- 容器化环境
- 服务注册、发现和可靠性
- 一切皆服务 ( Everything-as-a-Service)
- DevOps

# 微服务对APM的大影响



微服务的规模和动态性使得数据收集的成本大幅度提高，例如（cpu、内存和网络传输的开销）



大量的监控数据对后台数据处理分析的产生影响



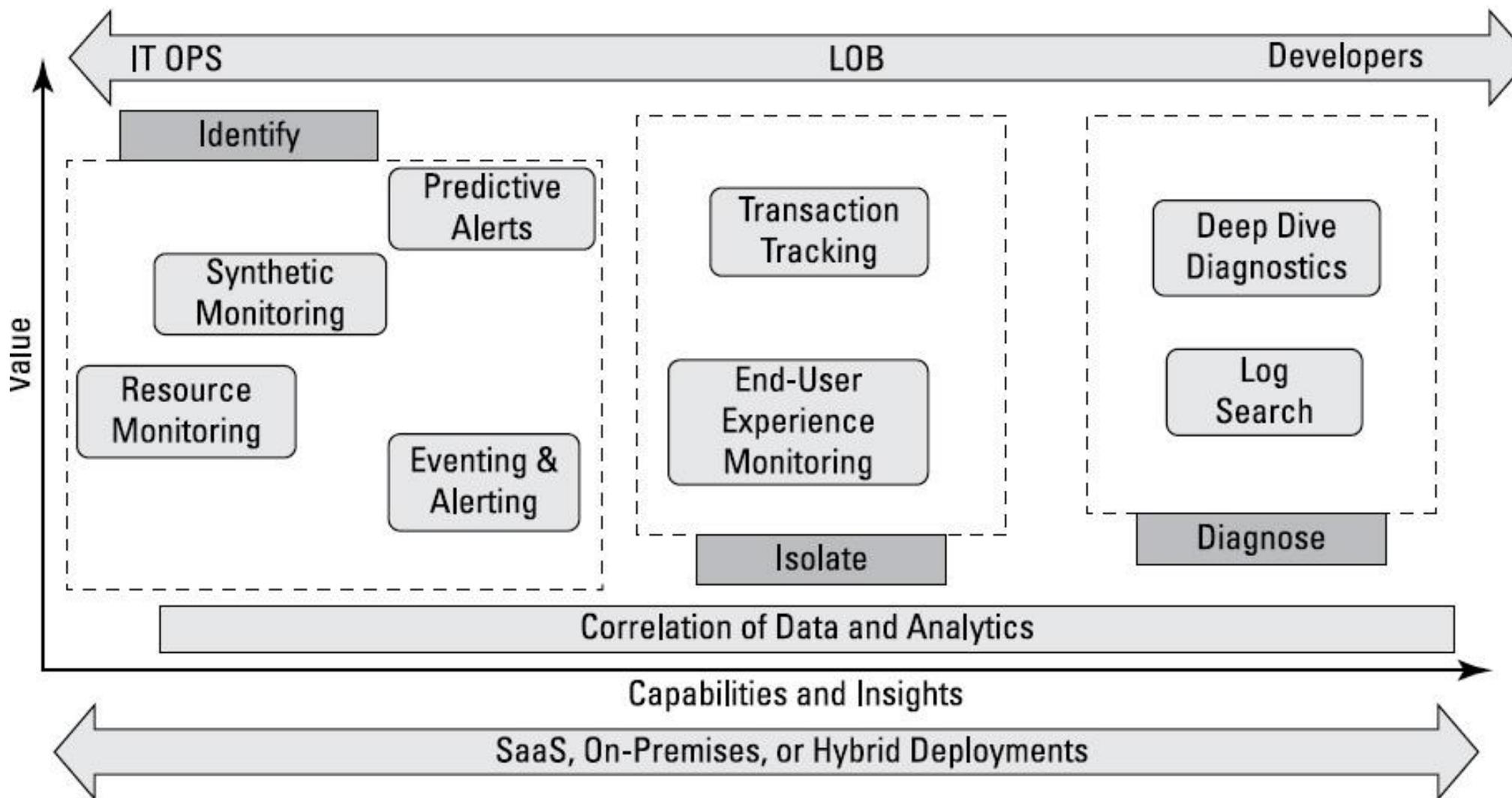
对于可视化和关联分析的要求方面，传统APM缺少好的手段



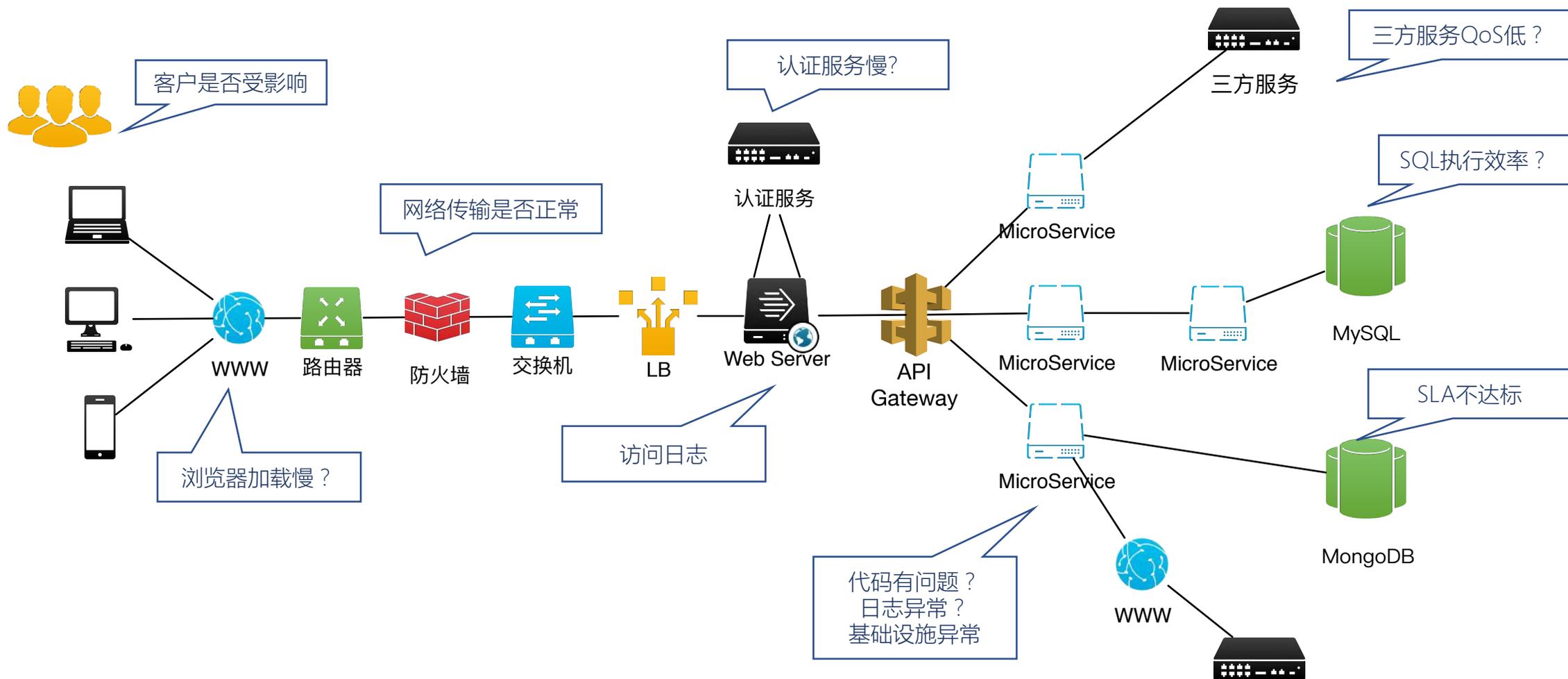
# Agenda

- 谈谈应用性能管理 ( APM )
- **微服务对APM的大影响**
- 如何构建适于微服务的APM平台
- 打造监控、告警和报障的闭环
- 对未来的一些思考

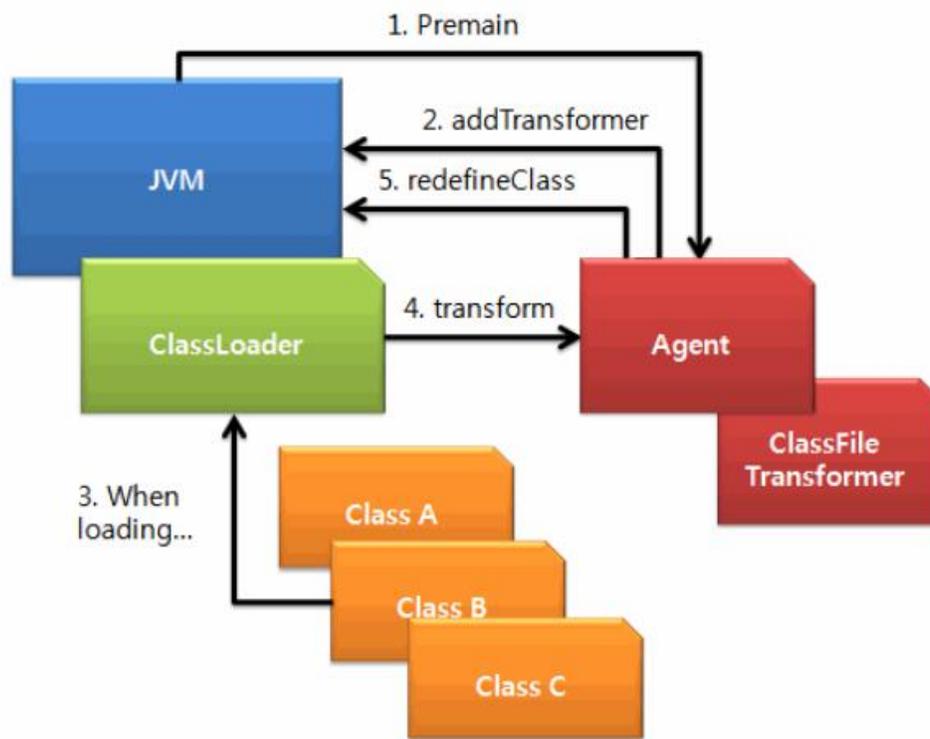
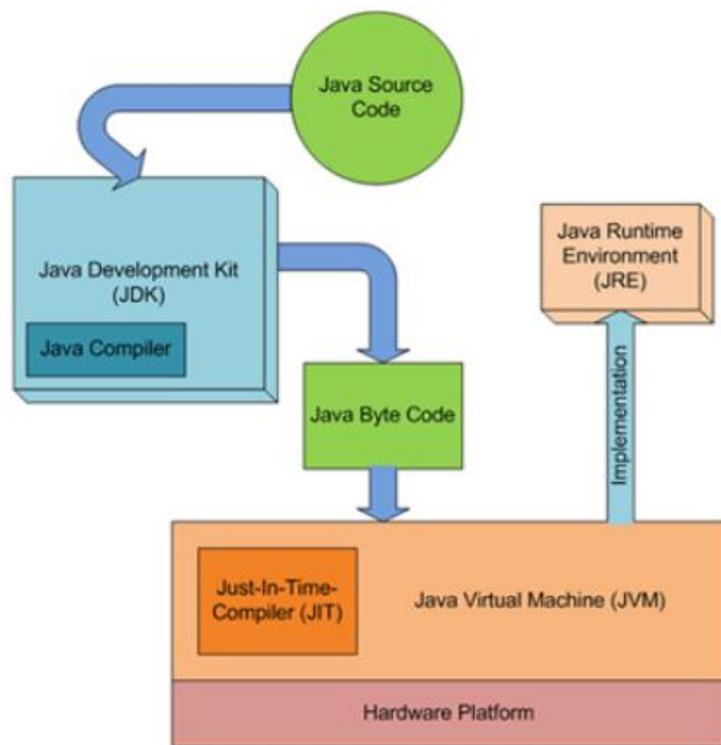
# APM的核心能力



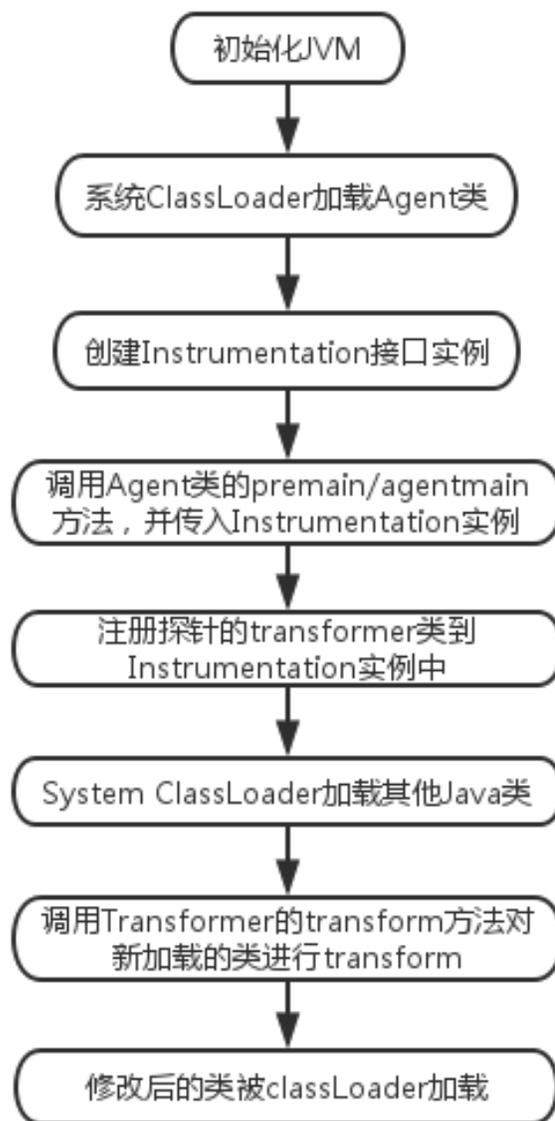
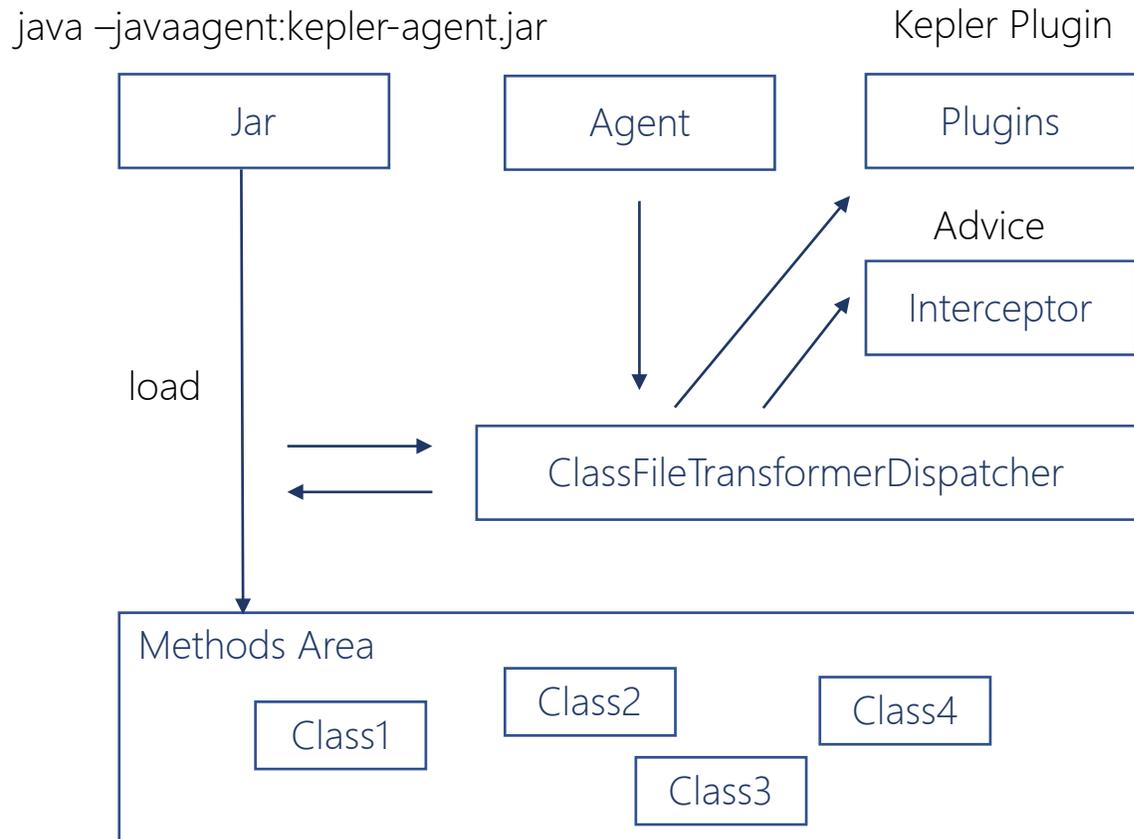
# 基于微服务的应用程序端到端监控



# APM探针的基本原理 (Java Instrument)

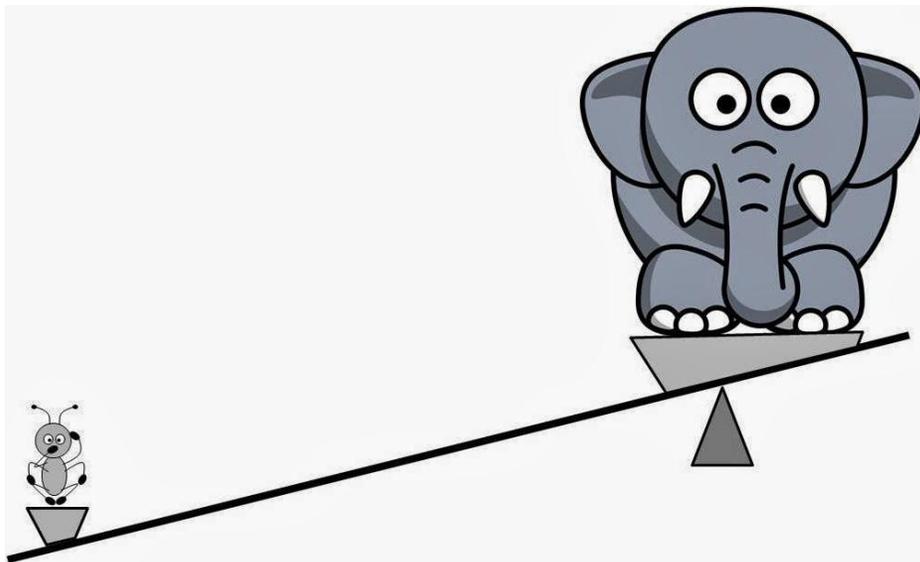


# APM探针的基本原理 (Java探针结构)



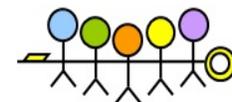
# APM探针的基本原理 (Java Instrument)

成本  
3%~5%的性能开销

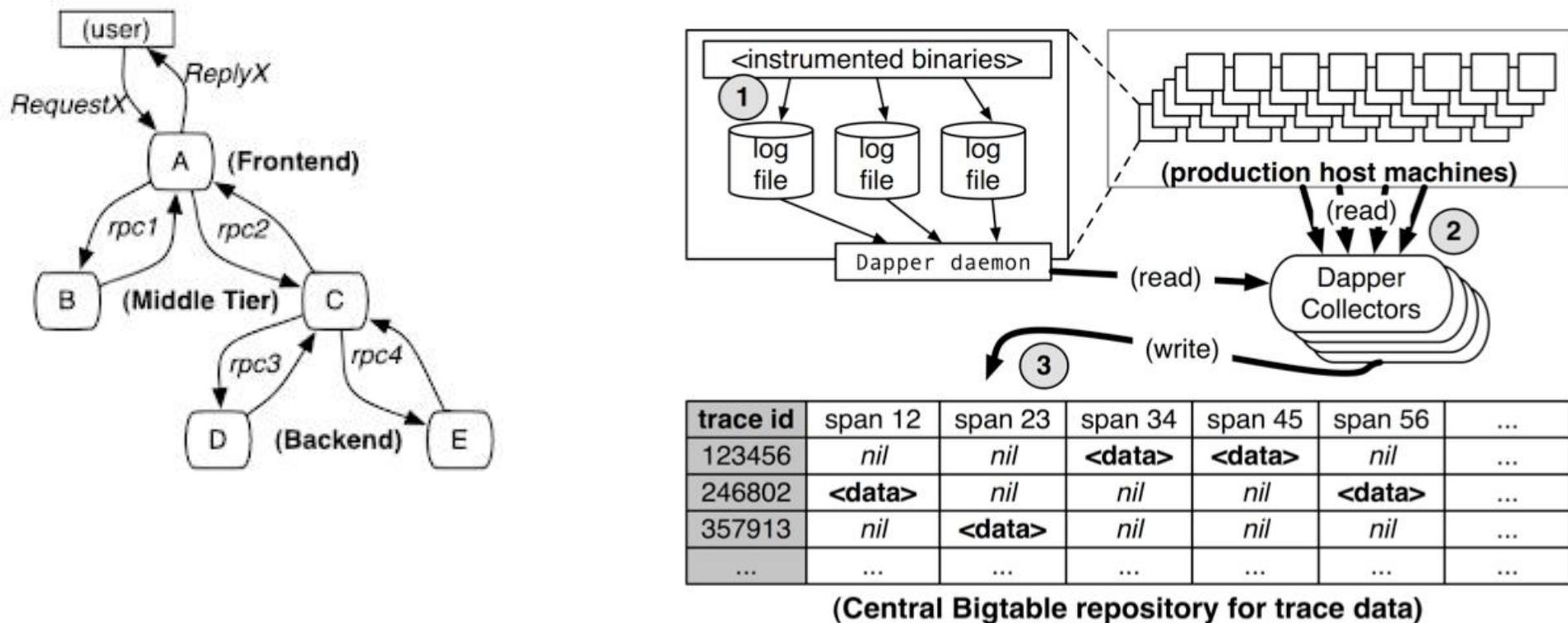


收益

1. 持续改善终端用户体验
2. 端到端无盲点监控
3. 业务拓扑实时发现
4. 性能问题代码级定位
5. 自定义事务深度剖析
6. 平均响应时间, Apdex
7. 吞吐量, 错误率多维度监控指标
8. SQL注入, 跨站攻击等安全事件监测
9. 应用漏洞代码级定位



# 分布式追踪 – Google Dapper

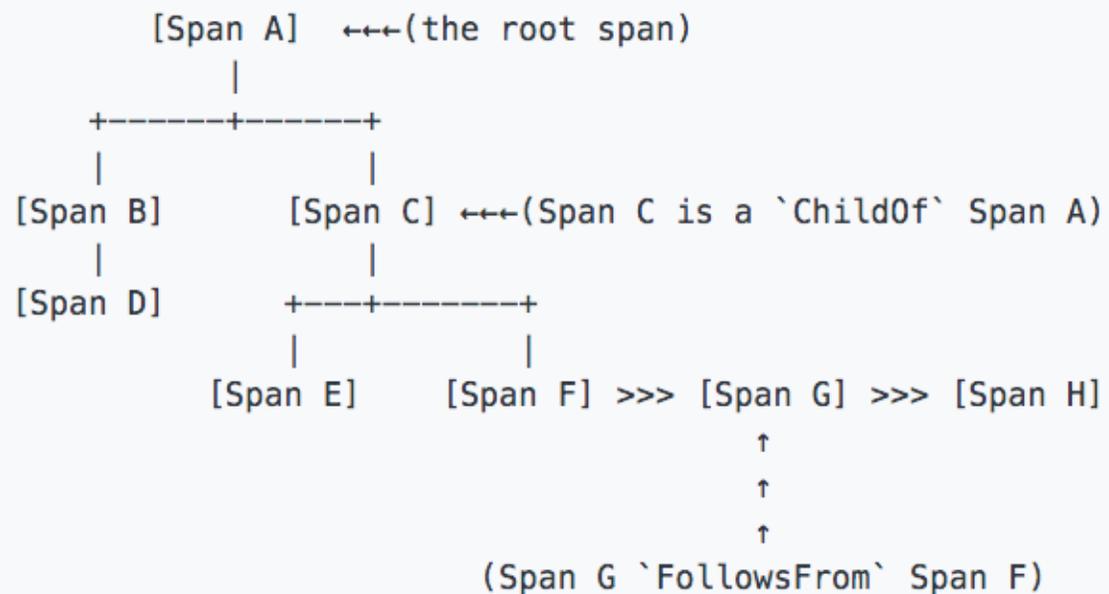


**Figure 5: An overview of the Dapper collection pipeline.**

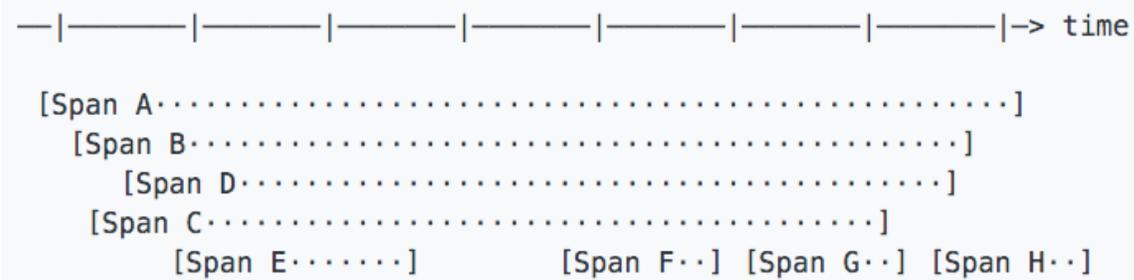
# 分布式追踪 – OpenTracing

<https://github.com/opentracing/specification/blob/master/specification.md>

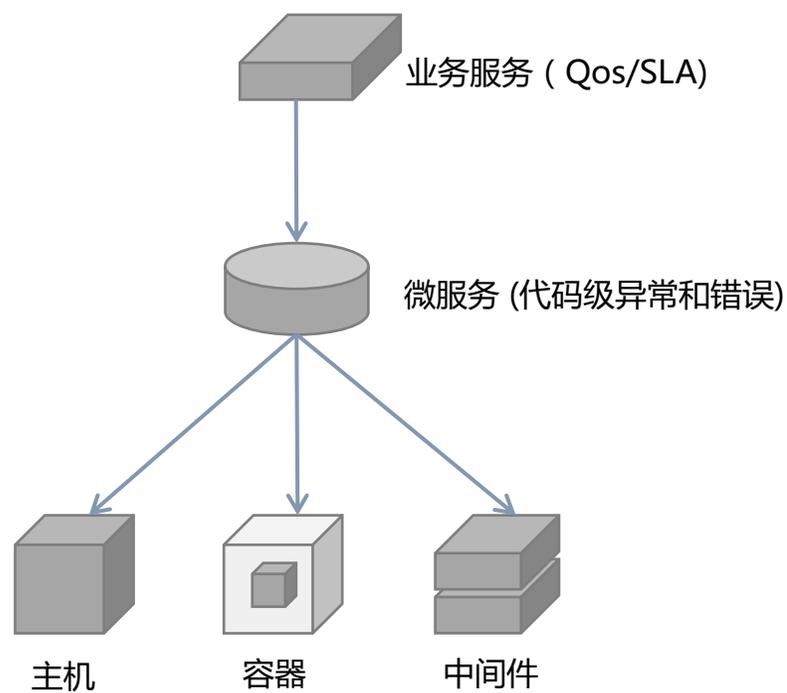
Causal relationships between Spans in a single Trace



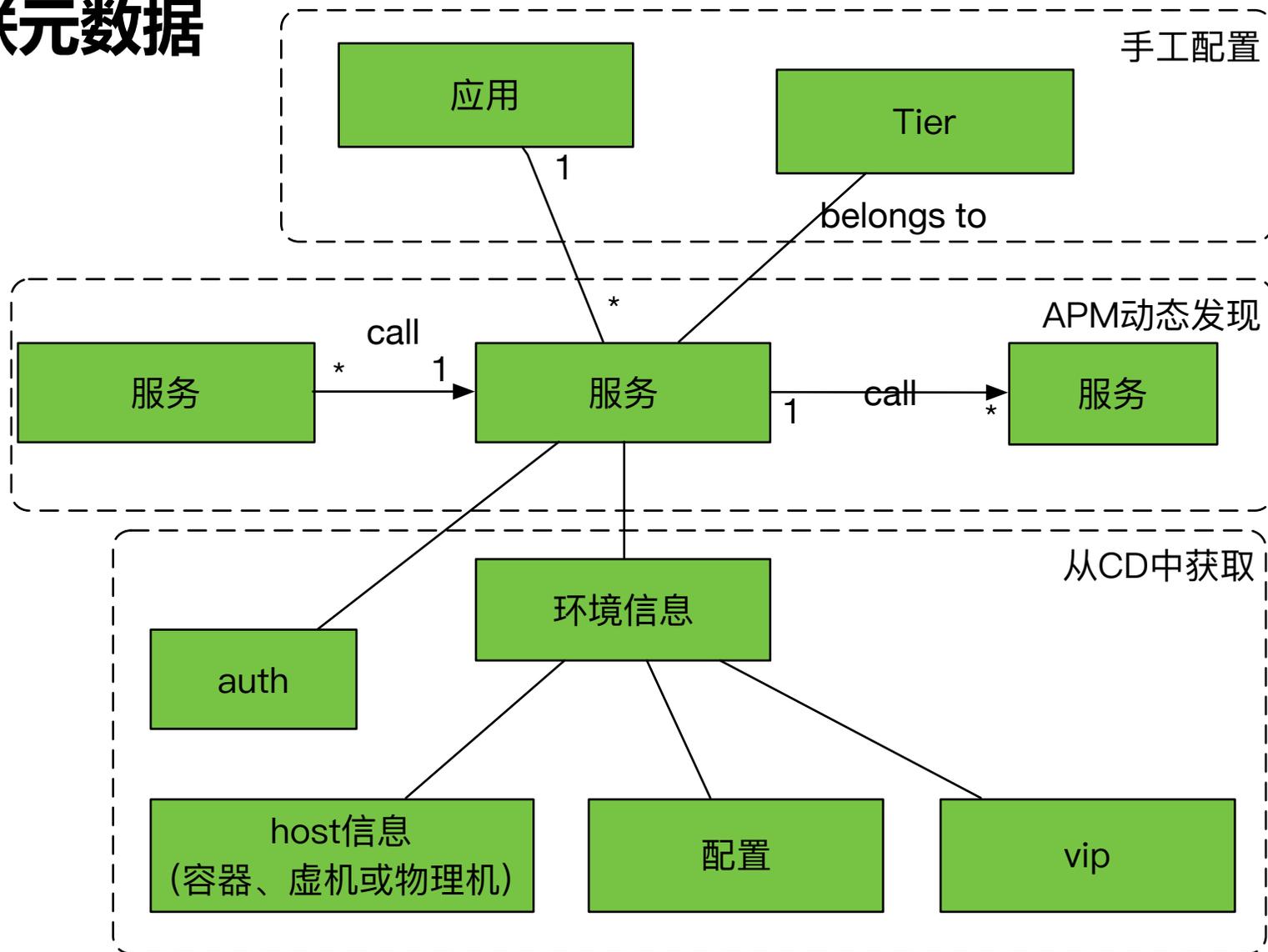
Temporal relationships between Spans in a single Trace



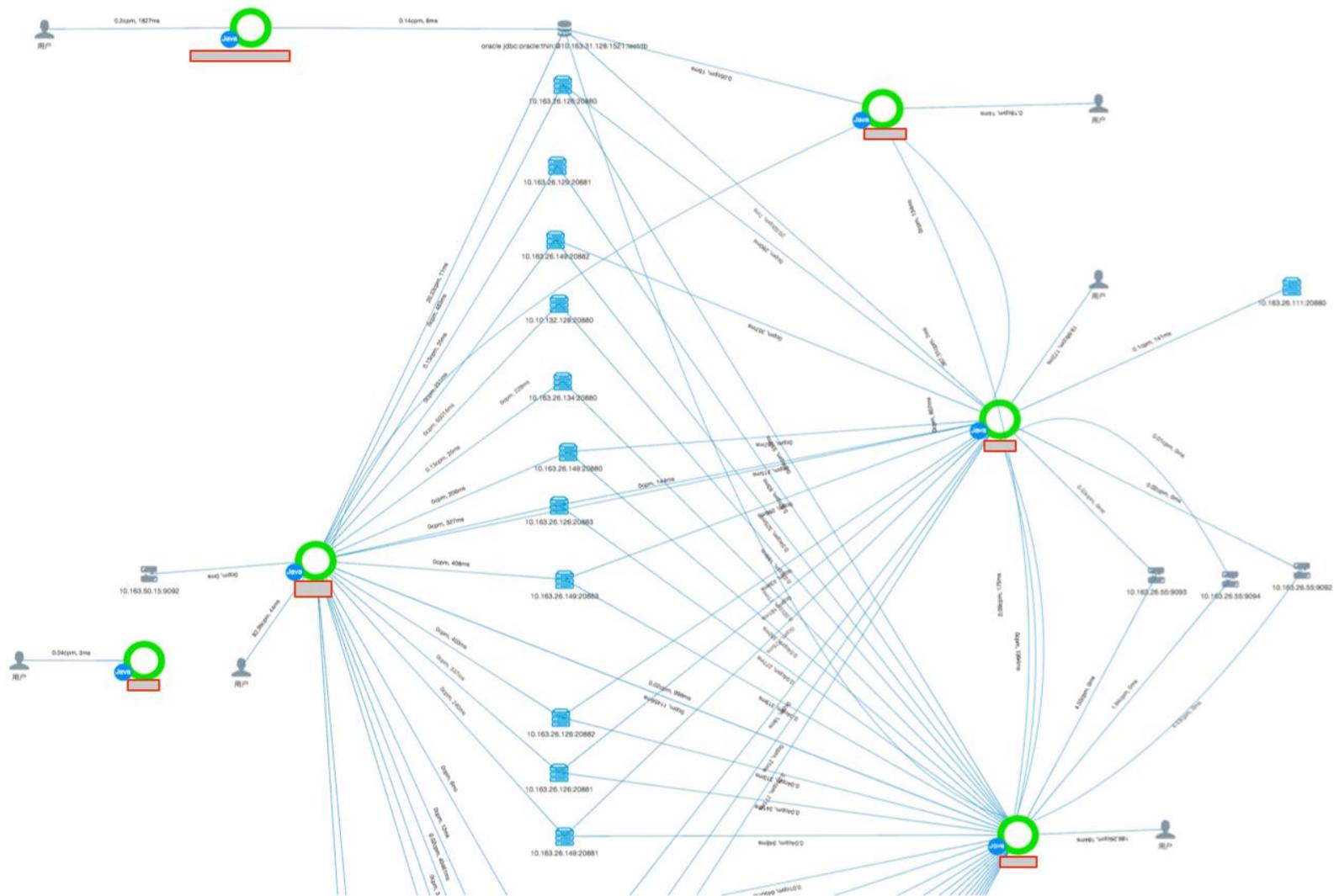
# 追踪一切



# 服务关联元数据



# 服务动态拓扑



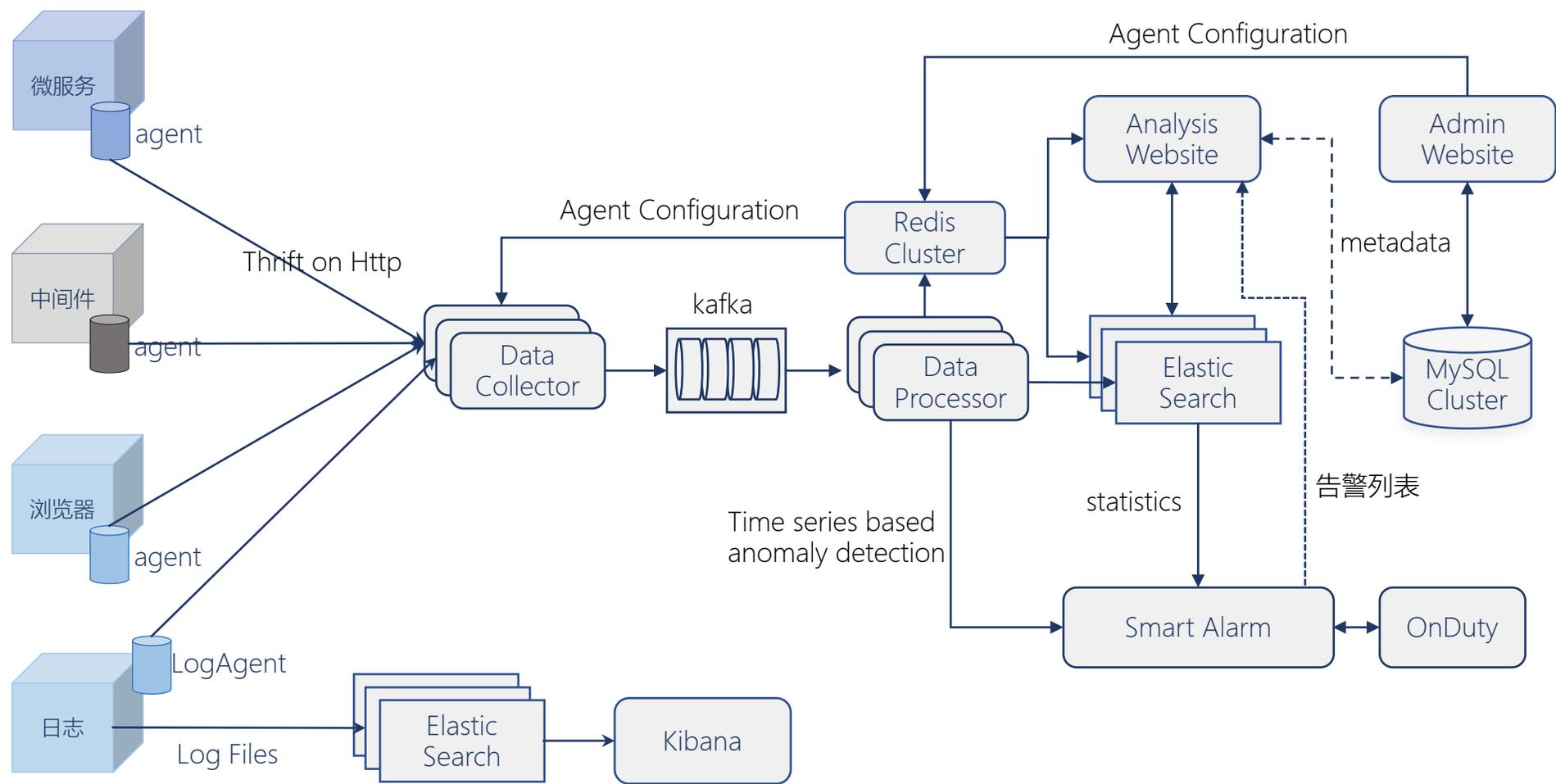
站

# 服务调用链分析

方法名	执行时间(ms)	实例名
▼ org.apache.http.protocol.HttpRequestExecutor.execute	106253	uat-t1-a1
▼ weblogic.servlet.internal.WebAppServletContext.execute	106251	uat-t1-a3
▼ org.apache.struts.action.ActionServlet.process	106239	uat-t1-a3
▼ cn.rongcapital.apm.demo.app.controller.struts1.action.OrderAction.innerQueryList	106238	uat-t1-a3
▼ org.hibernate.impl.SessionImpl.list	78005	uat-t1-a3
▼ oracle.jdbc.driver.OraclePreparedStatement.executeQuery	4002	uat-t1-a3



# APM总体架构



# 探针配置

应用配置



通用 Java Browser

收集器ip 10.200.10.43

收集器端口 8099

采样率 100

心跳周期 60000

ApdexT 100

取消

确定

应用配置



通用 Java Browser

jvm采样周期 60000

线程采样周期 400

取消

确定

应用配置



通用 Java Browser

是否启用浏览器端日志输出  OFF

是否启用OpenTracing  OFF

探针数据发送频率 1000

取消

确定



# APM核心能力

浏览器	应用	数据库	JVM	主机	其它
<ul style="list-style-type: none"><li>• 首屏时间</li><li>• 白屏时间</li><li>• 可操作时间</li><li>• 资源加载完成时间</li><li>• 页面性能指数</li><li>• 浏览器类型和版本</li><li>• DOM解析时间</li><li>• 用户IP</li><li>• 请求重定向时间</li><li>• 页面缓存加载时间</li><li>• ...</li></ul>	<ul style="list-style-type: none"><li>• 应用服务器类型</li><li>• 服务调用相应时间</li><li>• RPC调用详情</li><li>• 方法执行时间</li><li>• 方法执行参数</li><li>• 堆栈信息</li><li>• 吞吐量</li><li>• 服务器性能指数</li><li>• IO及网络异常</li><li>• 服务拓扑动态发现</li><li>• ...</li></ul>	<ul style="list-style-type: none"><li>• 数据库连接时间</li><li>• 数据库类型和版本</li><li>• SQL执行时间</li><li>• SQL调用语句</li><li>• SQL语句参数</li><li>• SQL执行计划</li><li>• SQL调用者堆栈信息</li><li>• ...</li></ul>	<ul style="list-style-type: none"><li>• Java版本</li><li>• 堆使用情</li><li>• 持久带使用情况</li><li>• 青年代使用情况</li><li>• 老年代使用情况</li><li>• GC次数和耗时</li><li>• 线程数</li><li>• 类加载数量</li><li>• ...</li></ul>	<ul style="list-style-type: none"><li>• 操作系统类型和版本</li><li>• CPU使用情况</li><li>• 内存使用情况</li><li>• 磁盘使用情况</li><li>• 网络使用情况</li><li>• 容器间关系</li><li>• ...</li></ul>	<ul style="list-style-type: none"><li>• 外部服务响应时</li><li>• 外部服务错误</li><li>• Cache响应时</li><li>• Cache命中率</li><li>• Cache大小</li><li>• 消息队列吞吐量</li><li>• 消息队列使用率</li></ul>

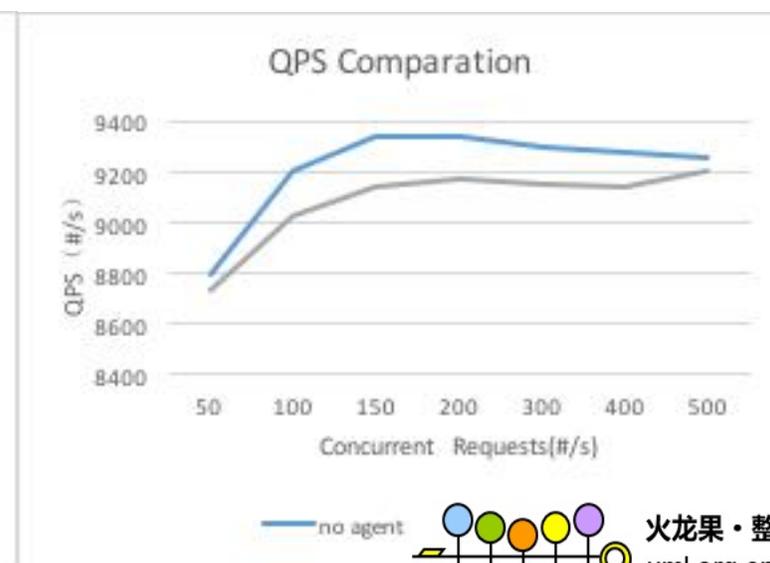
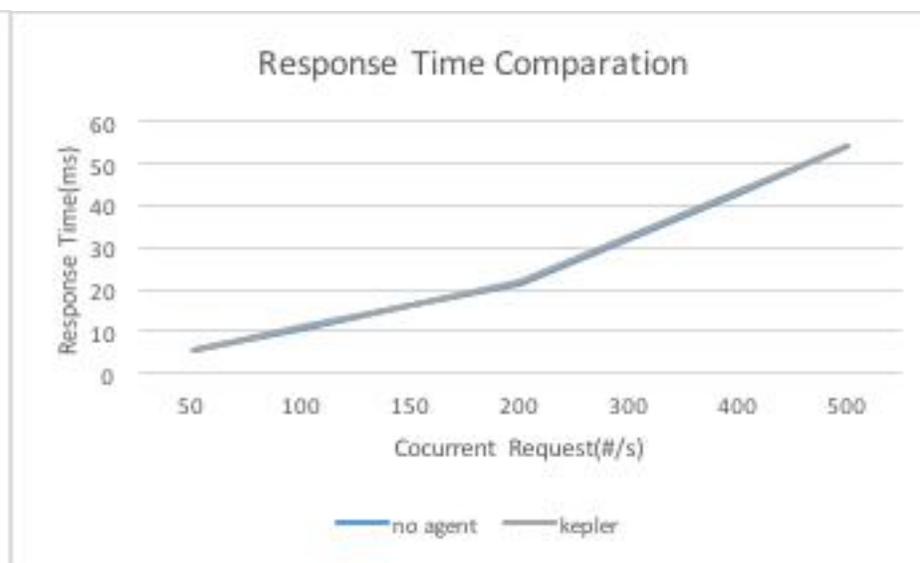
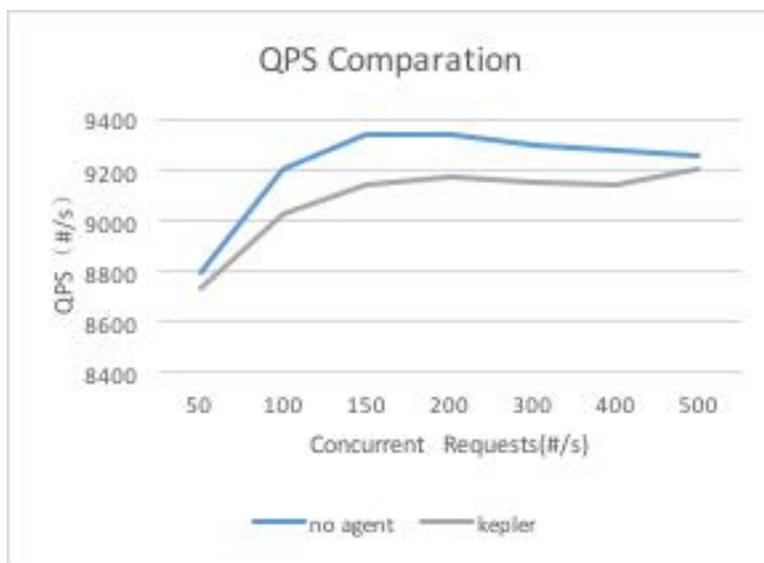
# 性能测试

项目	详情
JDK	java version "1.7.0_101" OpenJDK Runtime Environment (rhel-2.6.6.1.el7_2-x86_64 u101-b00) OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)
jvm参数	-Xms512m -Xmx512m
OS	3.10.0-327.el7.x86_64
Web容器	apache-tomcat-7.0.81
机器配置	虚拟机8C32G200G
测试应用	pinpoint官方的testapp，版本1.6.2

org.apache.catalina.core.StandardHostValve.invoke  
org.springframework.web.servlet.FrameworkServlet.doGet  
com.navercorp.pinpoint.testapp.controller.SimpleController.getCurrentTimestamp

分类	yong gc time (s)	full gc time (s)	total gc time (s)
blank	179.873	1.406	181.279
kepler	206.892	1.505	208.397

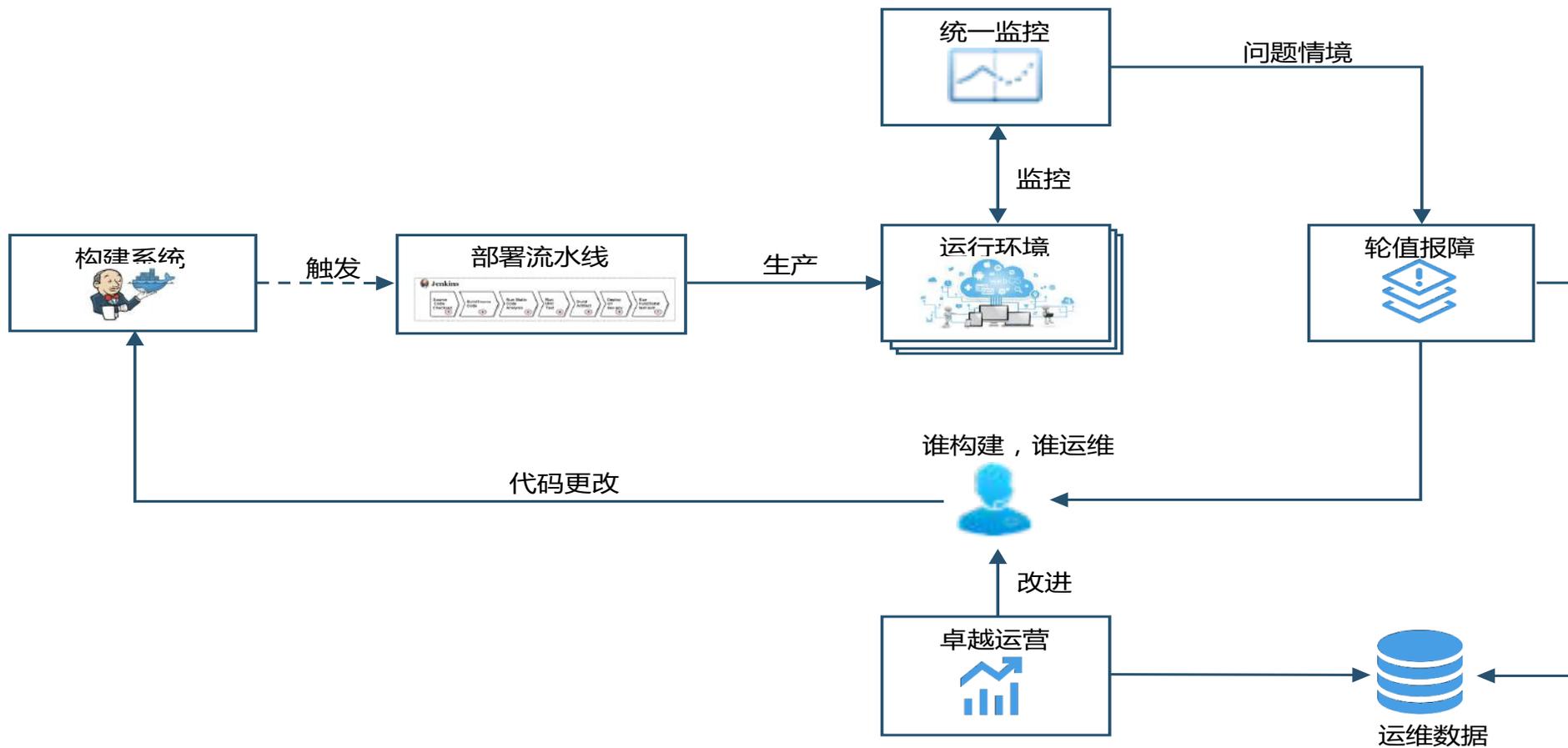
并请求数	qps-blank	qps-kepler	time-blank	time-kepler	cpu-blank	cpu-kepler
50	8800	8728	5.6	5.7	217	228
100	9210	9031	10.9	11.1	231	238
150	9339	9140	16.1	16.4	240	243
200	9344	9171	21.4	21.8	240	246
300	9303	9157	32.3	32.8	241	247
400	9280	9145	43	43.7	246	248
500	9254	9203	54	54.3	240	247



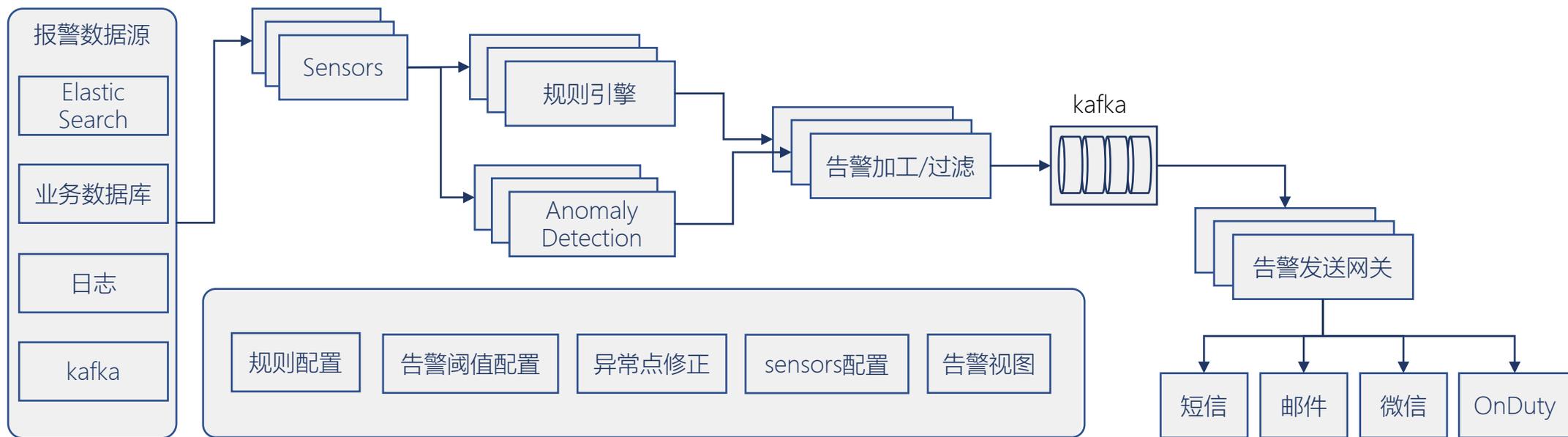
# Agenda

- 谈谈应用性能管理 ( APM )
- **微服务对APM的大影响**
- 如何构建适于微服务的APM平台
- 打造监控、告警和报障的闭环
- 对未来的一些思考

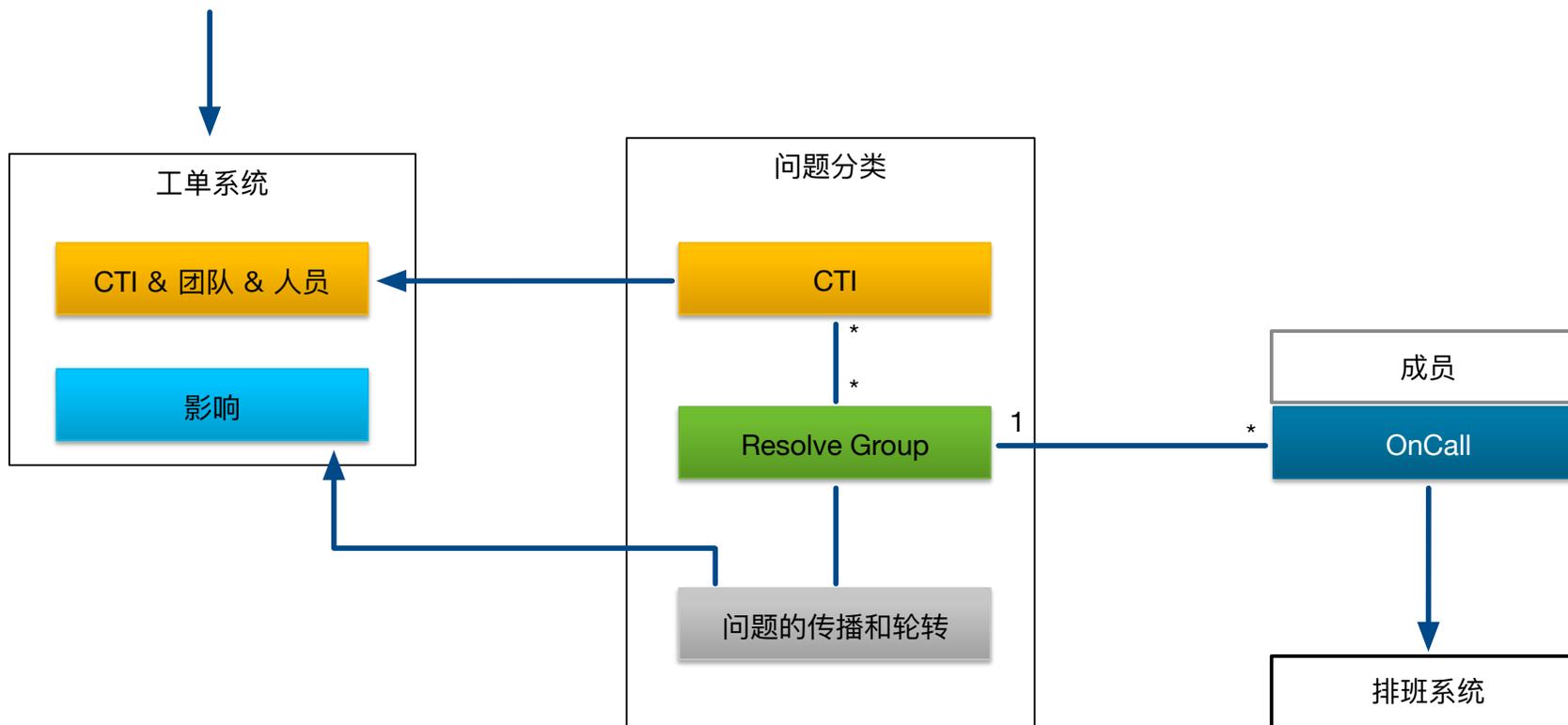
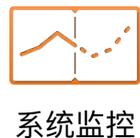
# 构建“部署 + 监控 + 告警 + 报障”闭环



# 告警平台



# 报障平台



# 报障平台

通过故障分类系统、支持组，快速将接入的各监控系统报障通知给相应维护人员，并通过配置的SLA及组织架构，对未及时响应的报障进行上告处理，以达到卓越运维的目的。

Onduty 报障 排班 支持组 admin admin | 退出

编辑支持组 [← 返回列表](#)

支持组名称  
test

支持组描述  
test

添加管理员  
1 adminadmin  
admin@devops.cn

成员列表  
1 adminadmin  
admin@devops.cn

SLA设置

故障等级	1	2	3	4	5
响应时间	30分	3时	12时	1.5天	3天
解决时间	9时	18时	1.5天	3天	6天

[编辑SLA](#)

自定义上报对象  是  否

关联到CTI

Category	Type	Item	操作
test	test	test	<a href="#">解除关联</a>

[+ 添加关联](#)

故障根因管理

名称	描述	操作
系统维护	计划内的人为事件，如计划内关机、重启、网站升级、网站维护等。	<a href="#">编辑</a> <a href="#">删除</a>
网络故障	网络原因导致的故障，如主干网络发生的故障。	<a href="#">编辑</a> <a href="#">删除</a>
硬件故障	服务器硬件导致的故障，如磁盘故障、内存故障、网卡故障等。	<a href="#">编辑</a> <a href="#">删除</a>
软件故障	软件原因导致的故障，如操作系统故障、Web服务器运行故障、应用程序BUG、软件服务容量超载等。	<a href="#">编辑</a> <a href="#">删除</a>

[取消](#) [保存](#)



# Agenda

- 谈谈应用性能管理 ( APM )
- **微**服务对APM的**大**影响
- 如何构建适于微服务的APM平台
- 打造监控、告警和报障的闭环
- 未来的思考和展望

# 大数据能力的充分释放-自动异常点检测

不同服务有不同的特点，  
避免修改每个服务的阈值

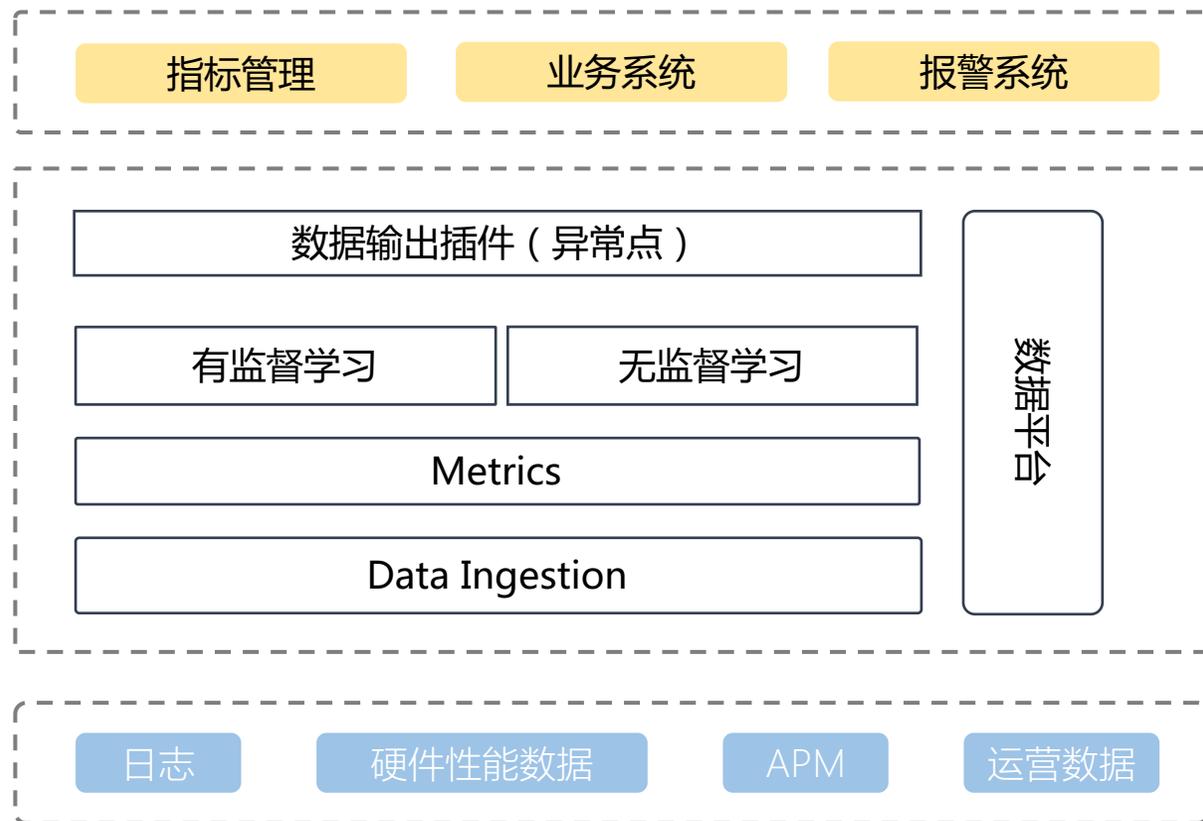


在服务负载发生变化时  
避免手工调整每个服务  
的阈值

避免静态阈值造成的误报

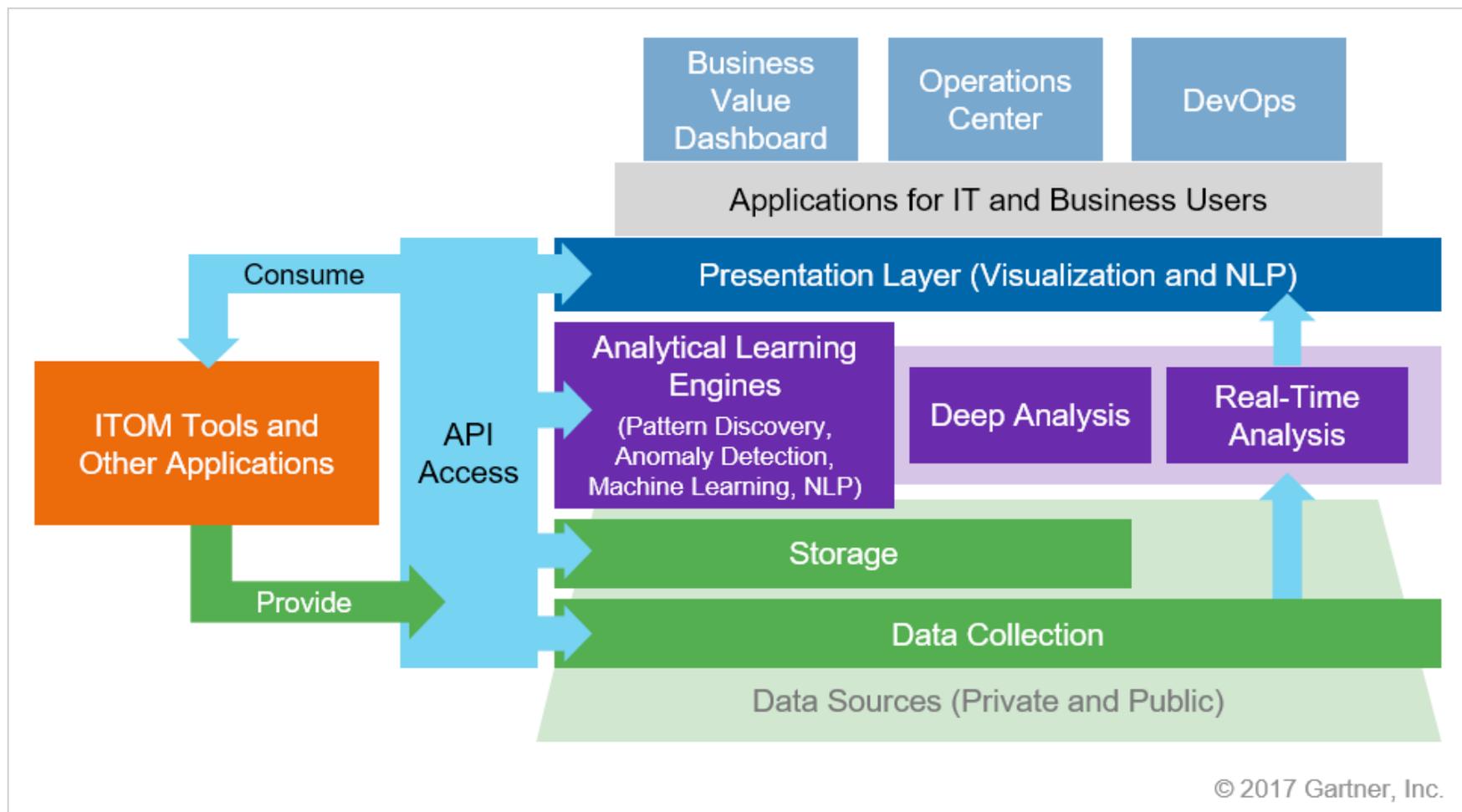
在不生成报警的情况下预测  
不正常的指标，用于诊断问题

# 大数据能力的充分释放-自动异常点检测



# AIOps

- 告警归并
- 根因分析
- 故障自动恢复
- 容量预测



**谢谢！**