

ArchData

技术峰会成都站

主办方：



2018年1月27日成都菁蓉国际广场3W COFFICE 蓉漂茶馆

打造微服务下的智能应用性能管理平

演讲人：王东



个人简介

- 创业公司CTO，负责公司大数据平台、微服务框架以及DevOps、AIOps平台的研发工作；
- 毕业于天津大学，毕业后一直从事软件相关研发和架构设计工作，曾经在普元软件任资深架构师、IBM GBS任咨询经理、亚马逊任架构师等，后加入创业公司，从事研发和管理工作；
- 热爱编程，喜欢钻研新技术，对于微服务、企业架构、大数据以及DevOps有浓厚的兴趣



Agenda

- 谈谈应用性能管理 (Application Performance Monitoring)
- 为什么需要AIOps
- 构建智能应用性能管理平台
- 智能告警算法介绍
- 卓越运营(Operation Excellent)

Agenda

- 谈谈应用性能管理 (Application Performance Monitoring)
- 为什么需要AIOps
- 构建智能应用性能管理平台
- 智能告警算法介绍
- 卓越运营(Operation Excellent)

什么是APM

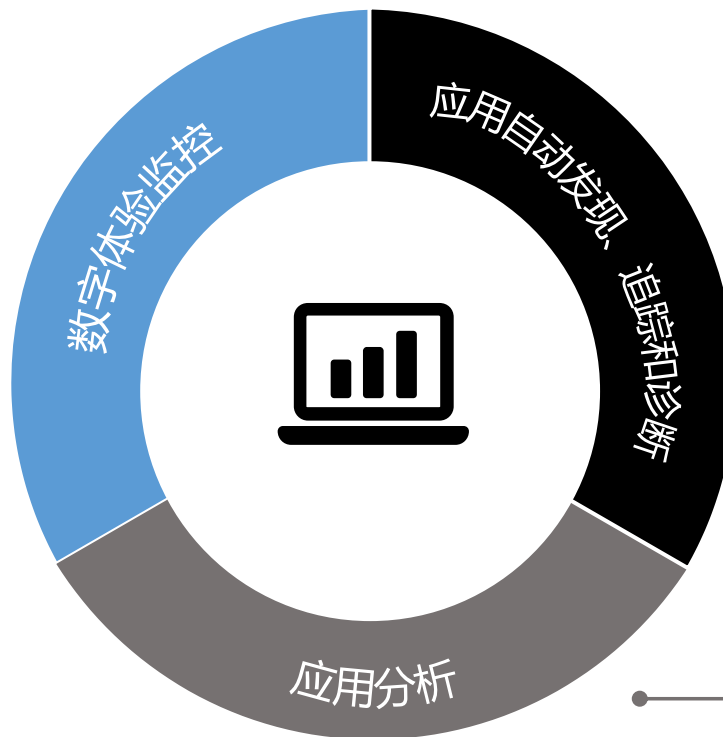
APM (Application Performance Management) 即应用性能管理，属于IT运维管理（ITOM)范畴。主要是针对企业关键业务的IT应用性能和用户体验的监测、优化，提高企业IT应用的可靠性和质量，保证用户得到良好的服务，降低IT总拥有成本(TCO)。

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

Gartner对APM核心能力的定义

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



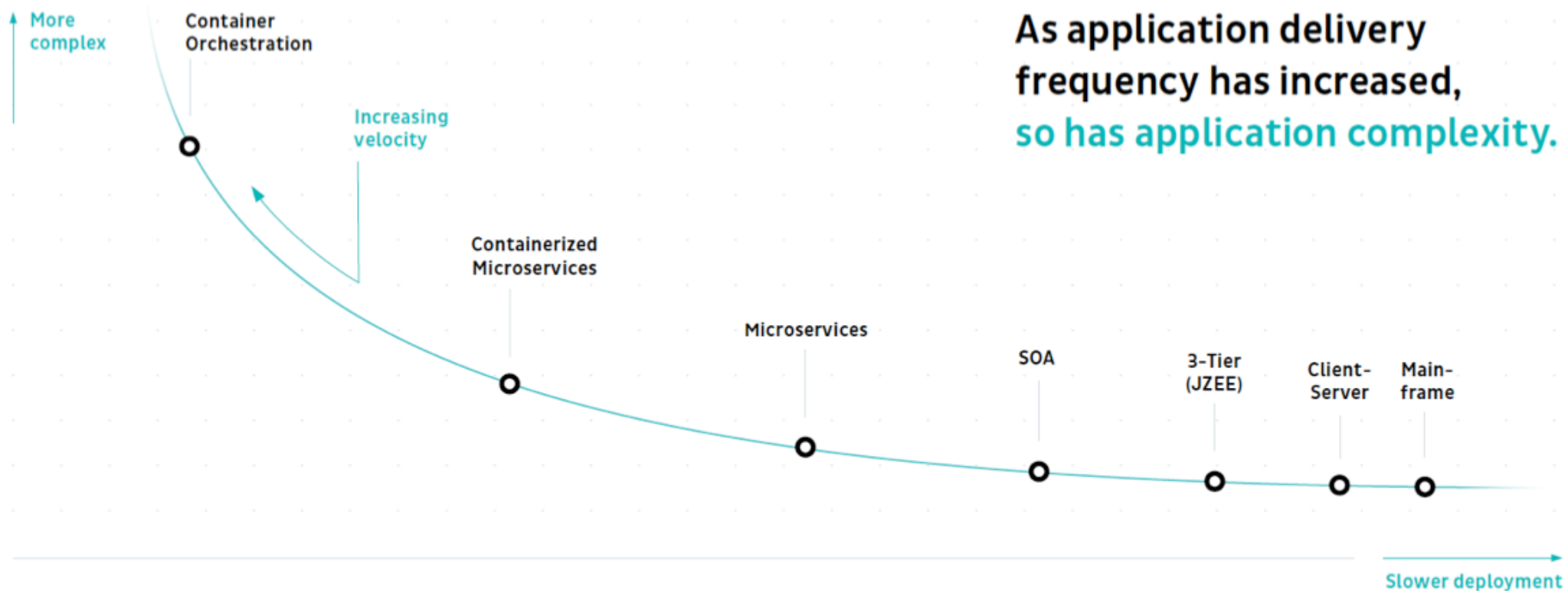
自动发现应用之间的逻辑关系，自动建模、应用组件的
深入监控及性能关联分析

通过机器学习，进行针对
JAVA及.NET应用的根源分析

经过几十年的发展，APM大约经历了三个阶段



软件架构的发展经历



云计算、容器化和微服务给监控带来的挑战...

依赖关系复杂

微服务环境下，不仅仅是服务之间的依赖关系变的复杂，而且，由于性能方面的要求，服务往往跟基础设施有比较紧密的关系

持续交付

敏捷的持续交付往往需要运维团队快速了解新的服务代理的扩展、性能、错误率等方面的影响

容器化环境

容器带来的不确定性（比如ip、端口），微服务架构不仅仅需要识别服务所在，还需要及时评估服务质量防止某个微服务的QoS降低带来的连锁反应

服务注册发现和可靠性

通过服务的注册、发现来管理服务，并通过熔断、重试等机制保障服务的可靠性

一切皆服务

云计算环境下的一切皆服务的模式使得很多组织不再关心承载微服务的基础设施，但是需要对云环境和容器保持监控的可见性



... 基础设施、环境和架构的复杂性导致 ...

01 运维成本不断增加

云计算和微服务的采用使得IT系统和环境越来越复杂，传统IT运维监控工具只能专注于IT系统的某个方面，需要大量的工具和资源来应对复杂的监控运维任务，导致成本不断攀升

02 运维效率越来越低

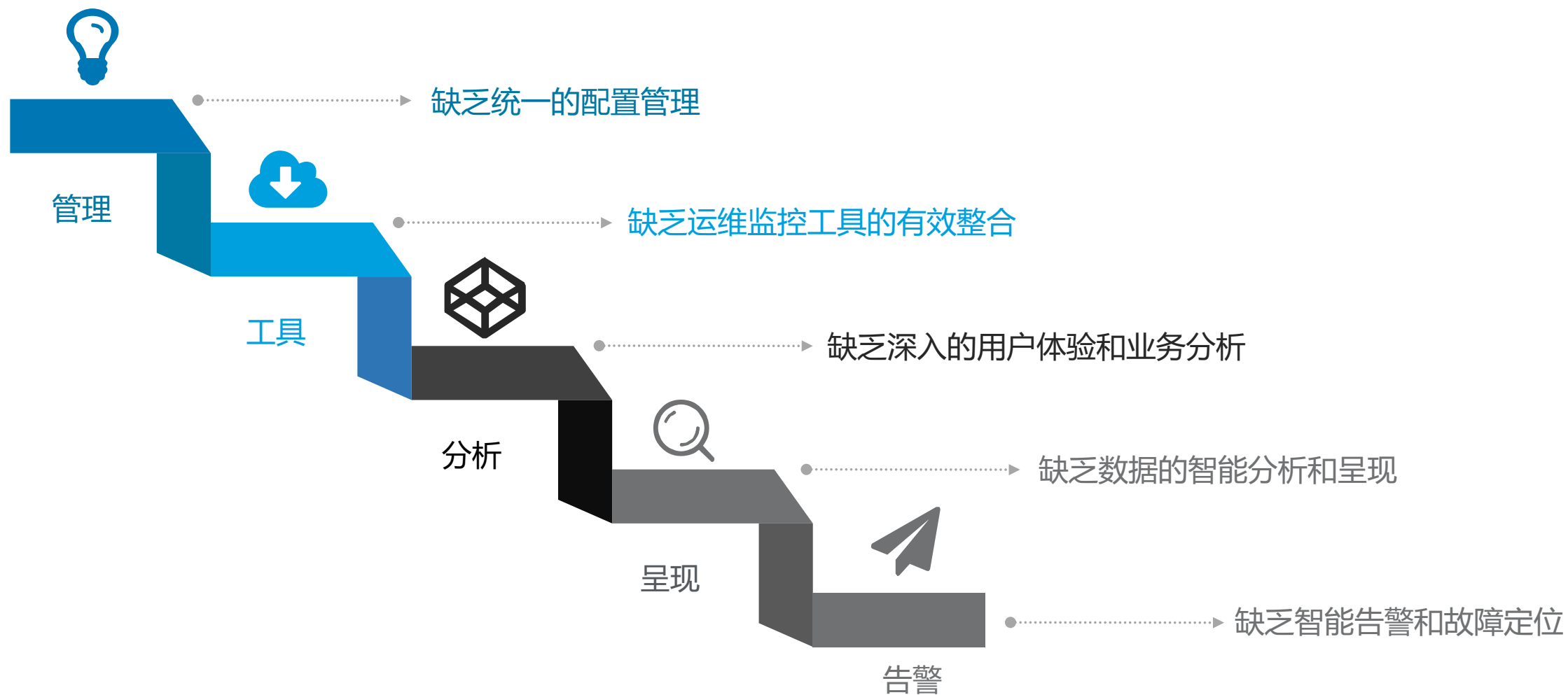
传统运维监控工具无法有效连通IT基础资源、应用程序、用户体验以及业务事务等监控单元，出现问题需要逐个排查，解决问题效率低下



03 运维质量难以提高

线上问题无法复现，无法迅速发现问题并定位问题根源，疲于处理大量告警和宕机等问题，运维长时间处于被动地位

... 传统的应用性能监控面临前所未有的压力，主要原因是

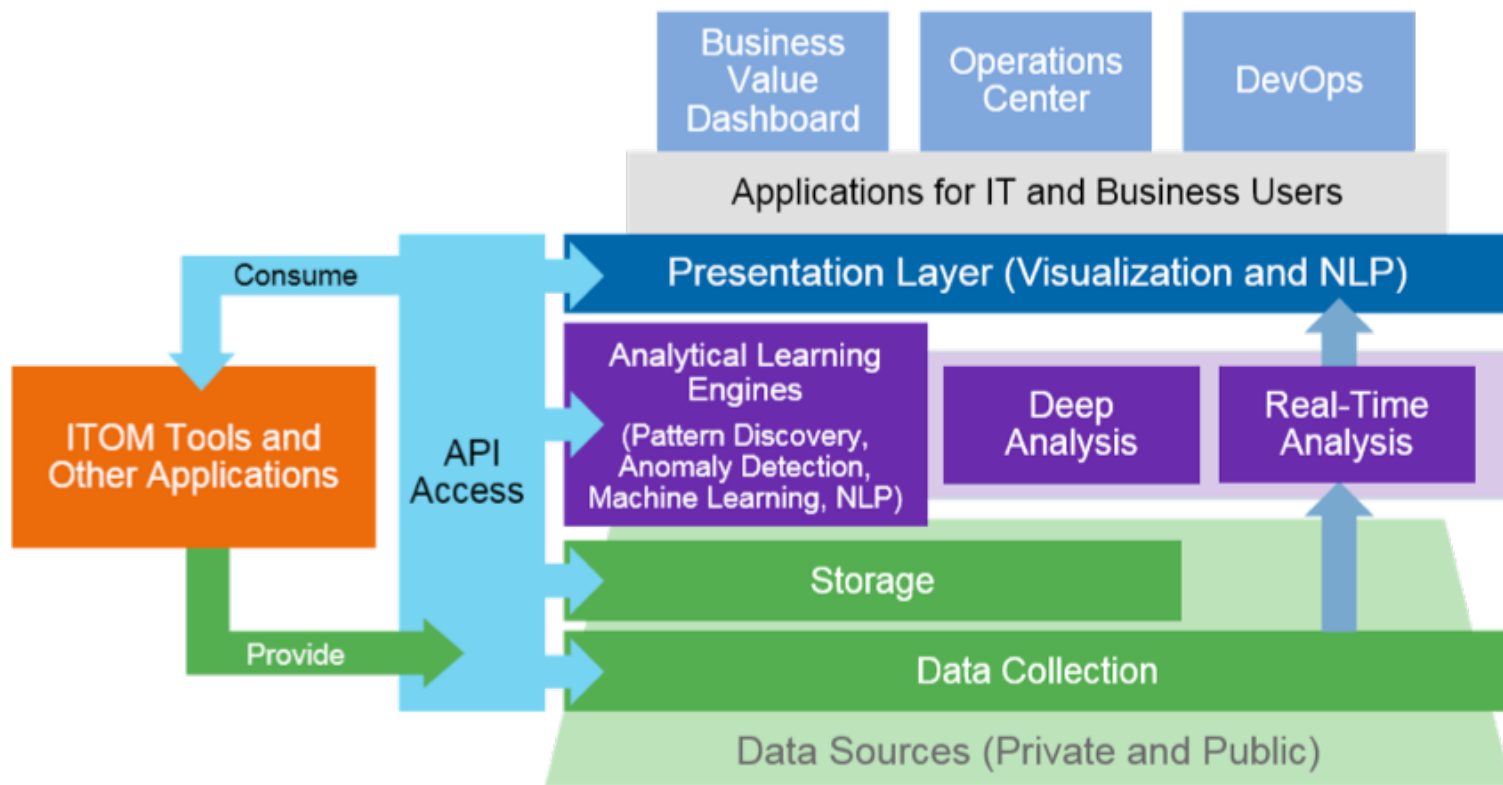
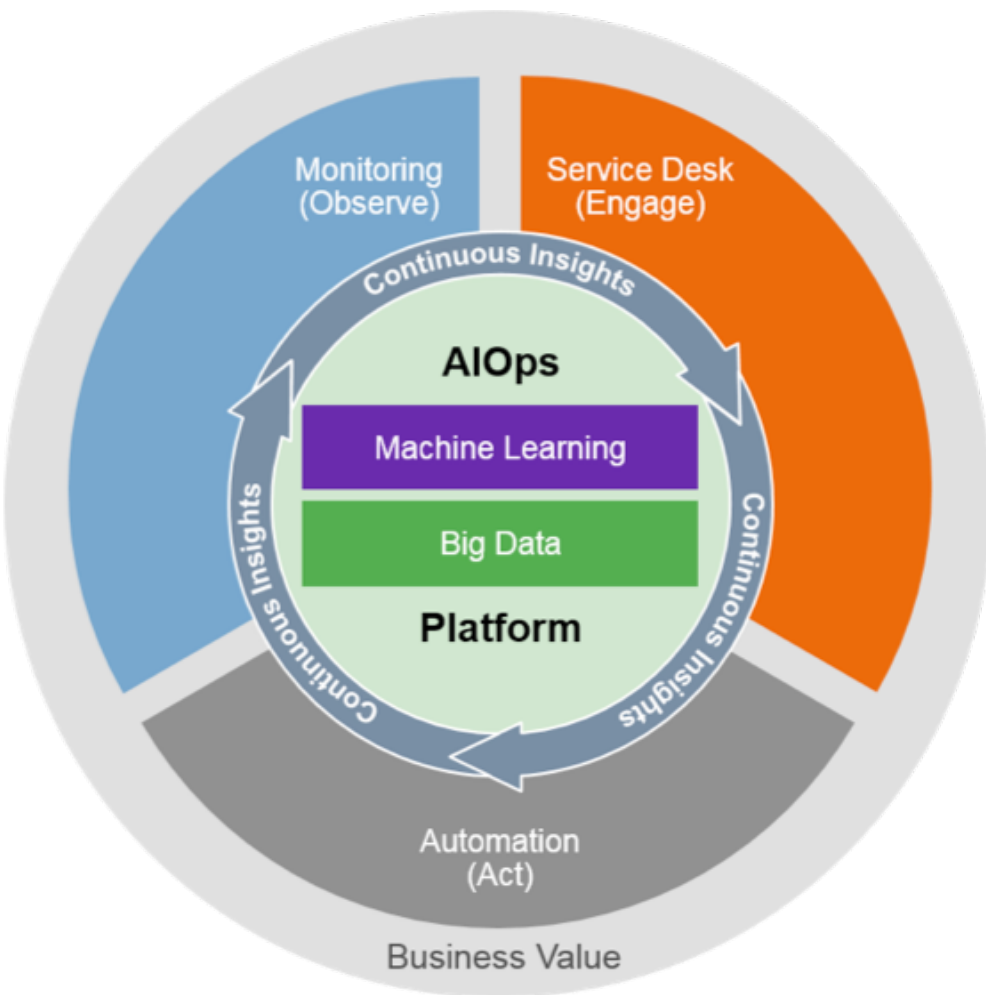


Agenda

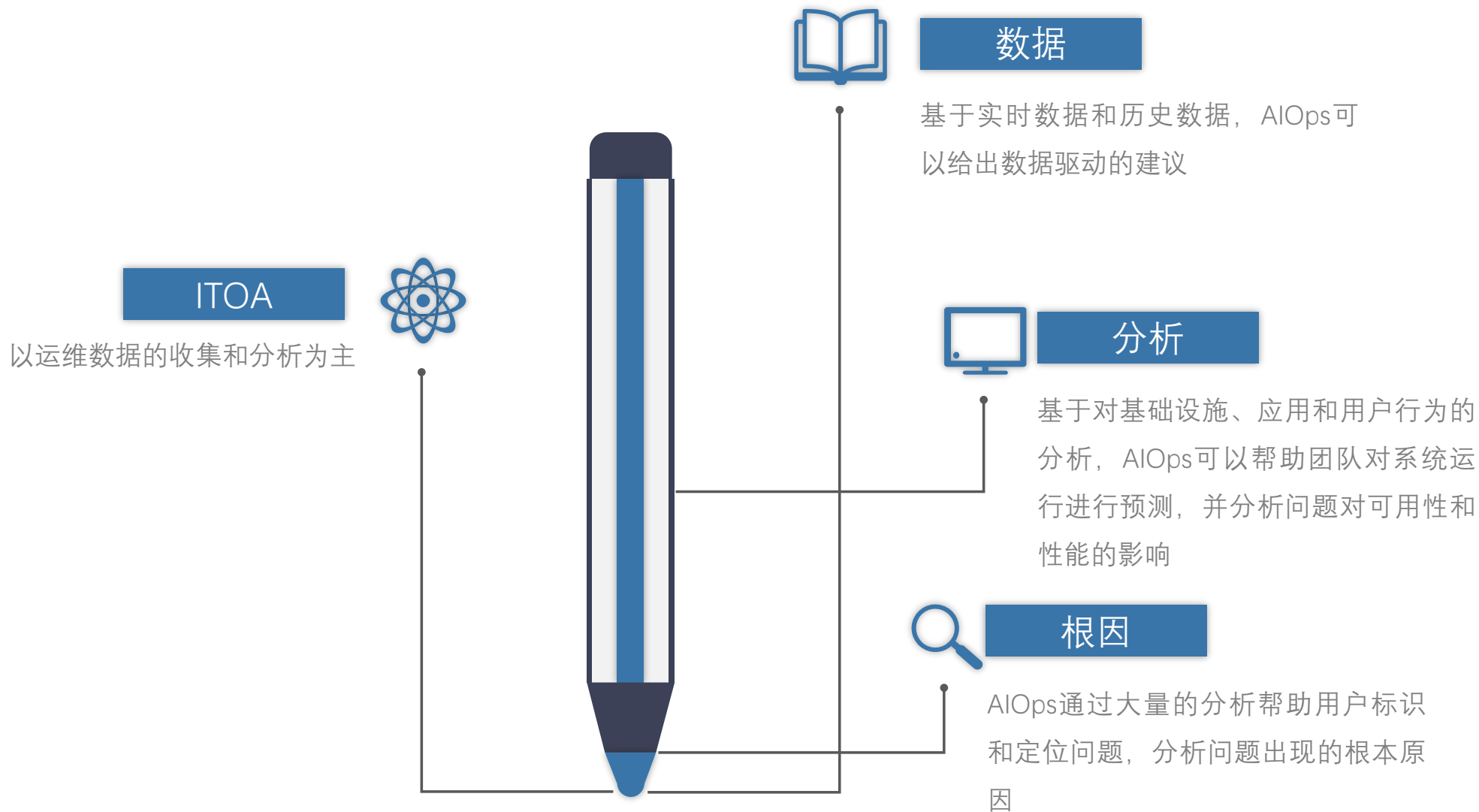
- 谈谈应用性能管理 (Application Performance Monitoring)
- 为什么需要AIOps
- 构建智能应用性能管理平台
- 智能告警算法介绍
- 卓越运营(Operation Excellent)

什么是AIOps ?

Algorithmic IT Operations , 也就是通过智能化算法进一步提升运维效率 , 包括问题分析、故障预测和运维决策



AIOps与传统ITOA有什么不同？



... 传统APM无法解决问题快速发现和根因诊断，AIOps才是正解

根据Gartner的统计，到2020年，至少**50%**的企业会采用AIOps来提高IT运维能力和业务洞察力，而这个比例现在不足**10%**

采用AIOps的优势

- 企业认为AIOps使得他们在竞争中获得优势（61%）
- 具有战略意义（57%）
- 新技术无缝部署(38%)
- 降低企业运营和技术维护成本(53%)

采用AIOps的优势

- 企业认为AIOps使得他们在竞争中获得优势（61%）
- 具有战略意义（57%）
- 新技术无缝部署(38%)
- 降低企业运营和技术维护成本(53%)

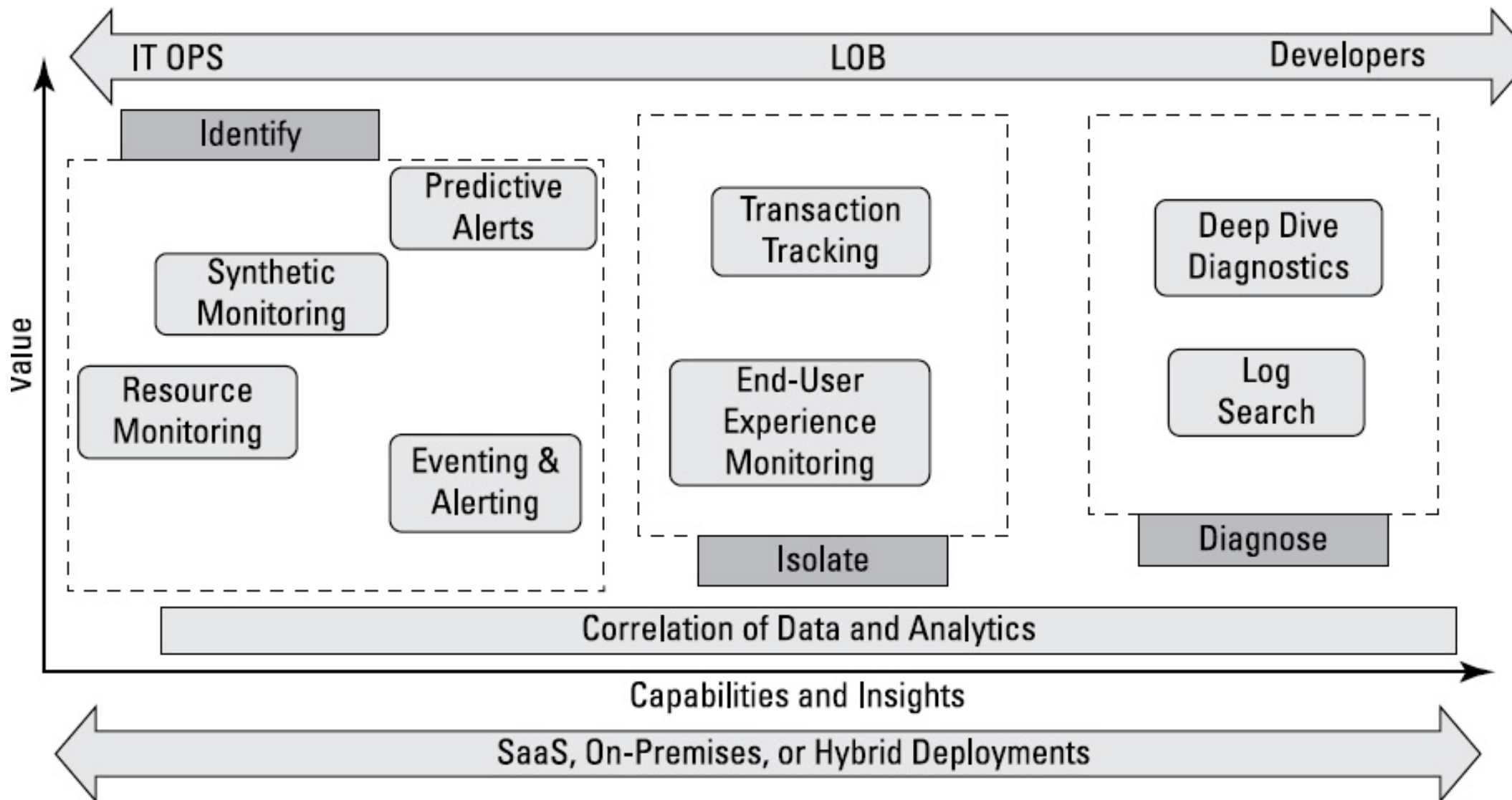
AIOps解决的问题

- 性能分析优化
- 故障检测
- 业务影响分析
- 监控异常检测
- 业务关联分析
- 容量规划

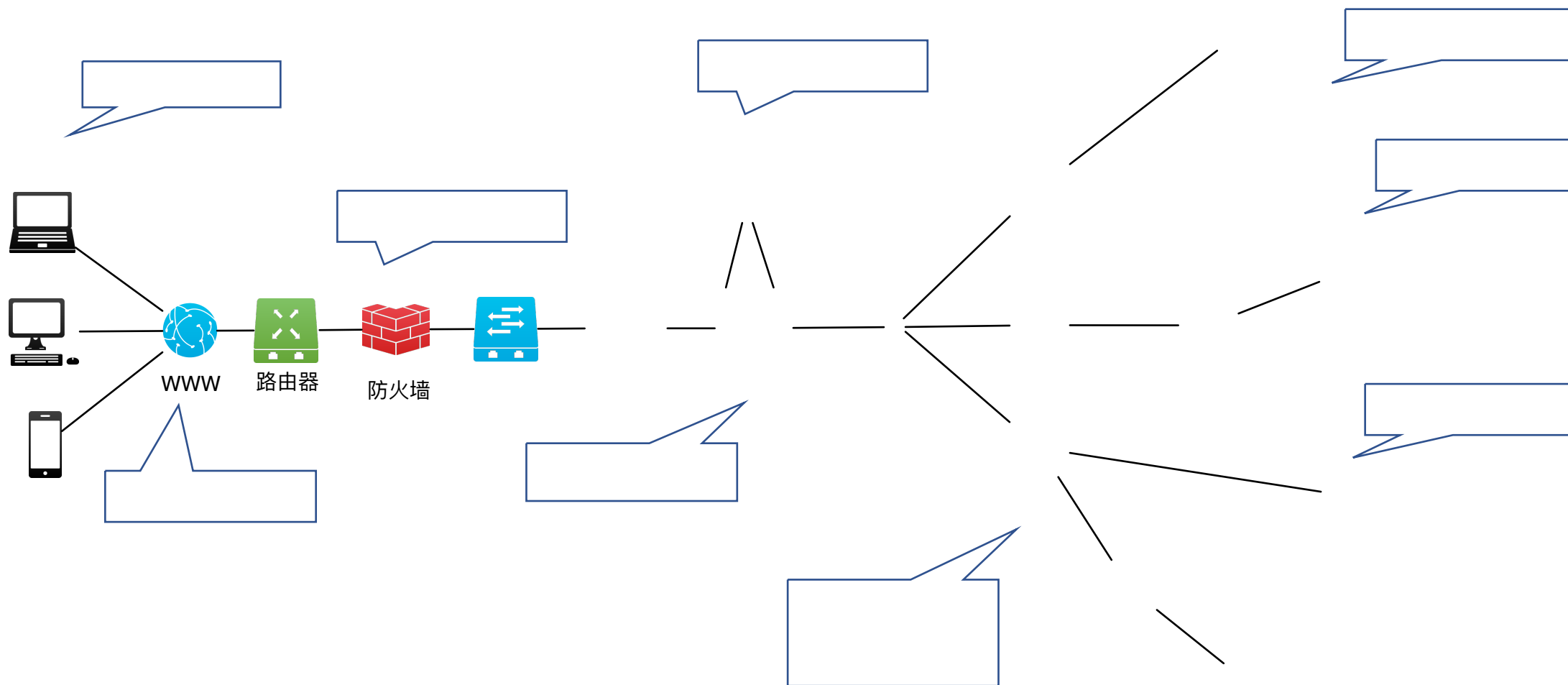
Agenda

- 谈谈应用性能管理 (Application Performance Monitoring)
- 为什么需要AIOps
- **构建智能应用性能管理平台**
- 智能告警算法介绍
- 卓越运营(Operation Excellent)

应用性能管理的核心能力



微服务监控需要实现端到端的监控





从数据层面打通APM、日志和基础资源监控，发挥AIOps

应用性能管理

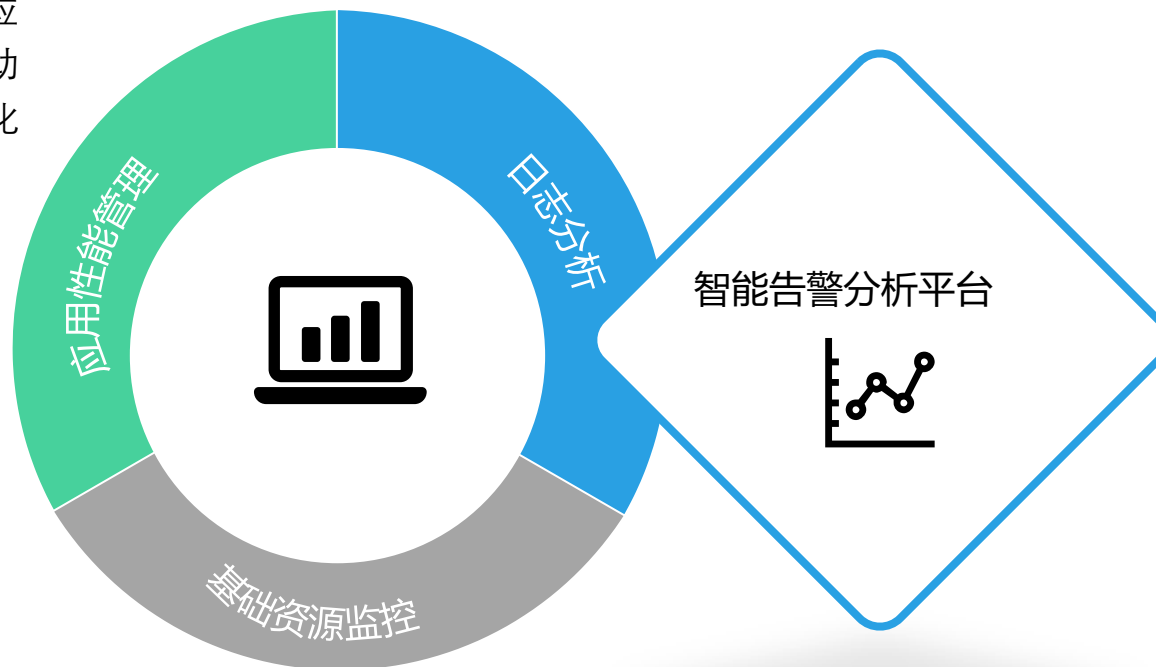
对企业业务系统进行即时监控以实现对应应用程序性能、故障进行系统化管理，帮助企业实现应用性能的最优化、改进与优化终端用户体验

日志分析

对日志进行统一收集、提取和存储，提供可视化的日志检索方式；对日志进行分析，剖析业务事务、系统异常等

基础资源监控

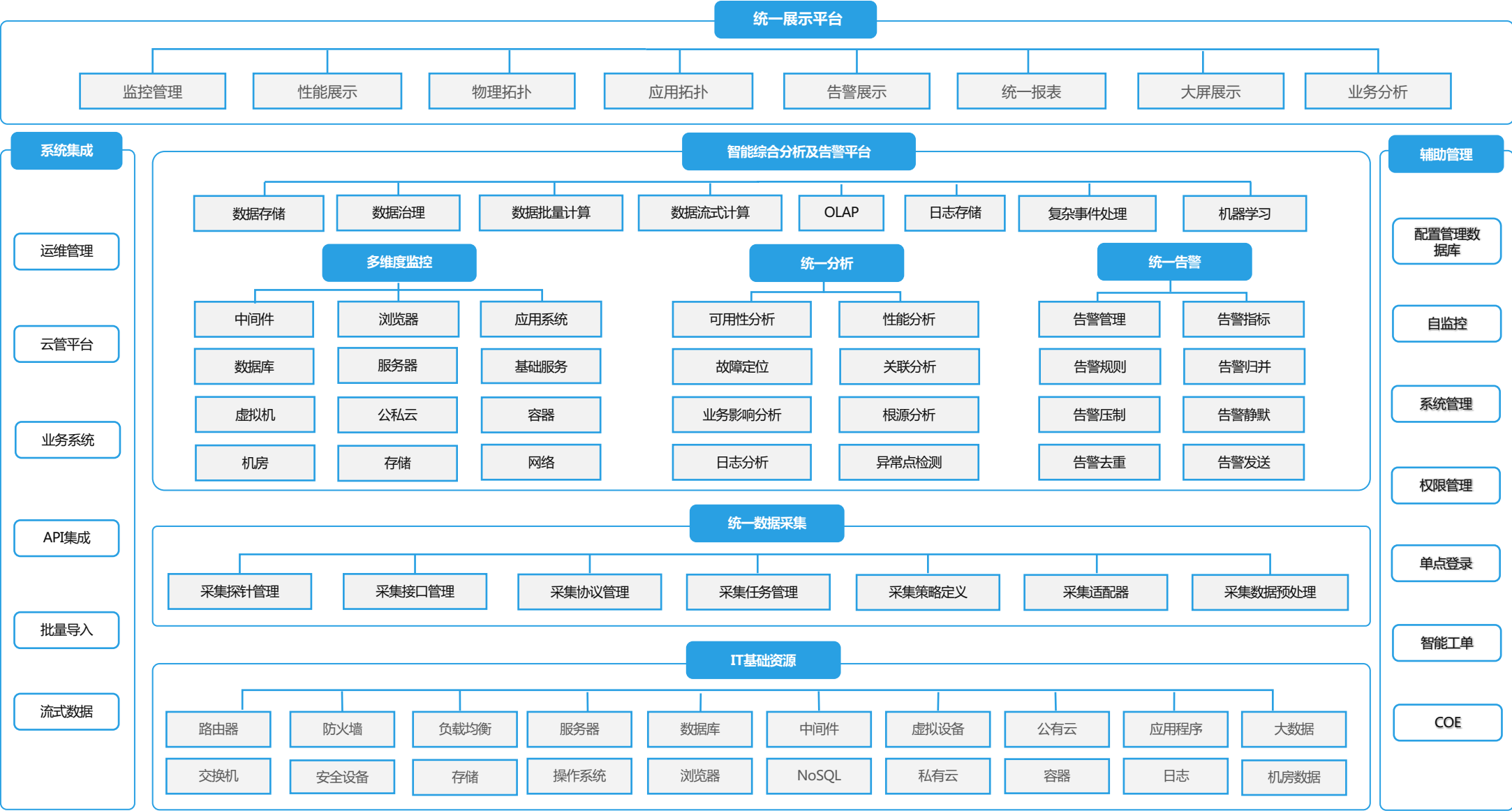
统一监控管理主机、网络、网络以及平台虚拟化等平台基础环境及应用的全面监控



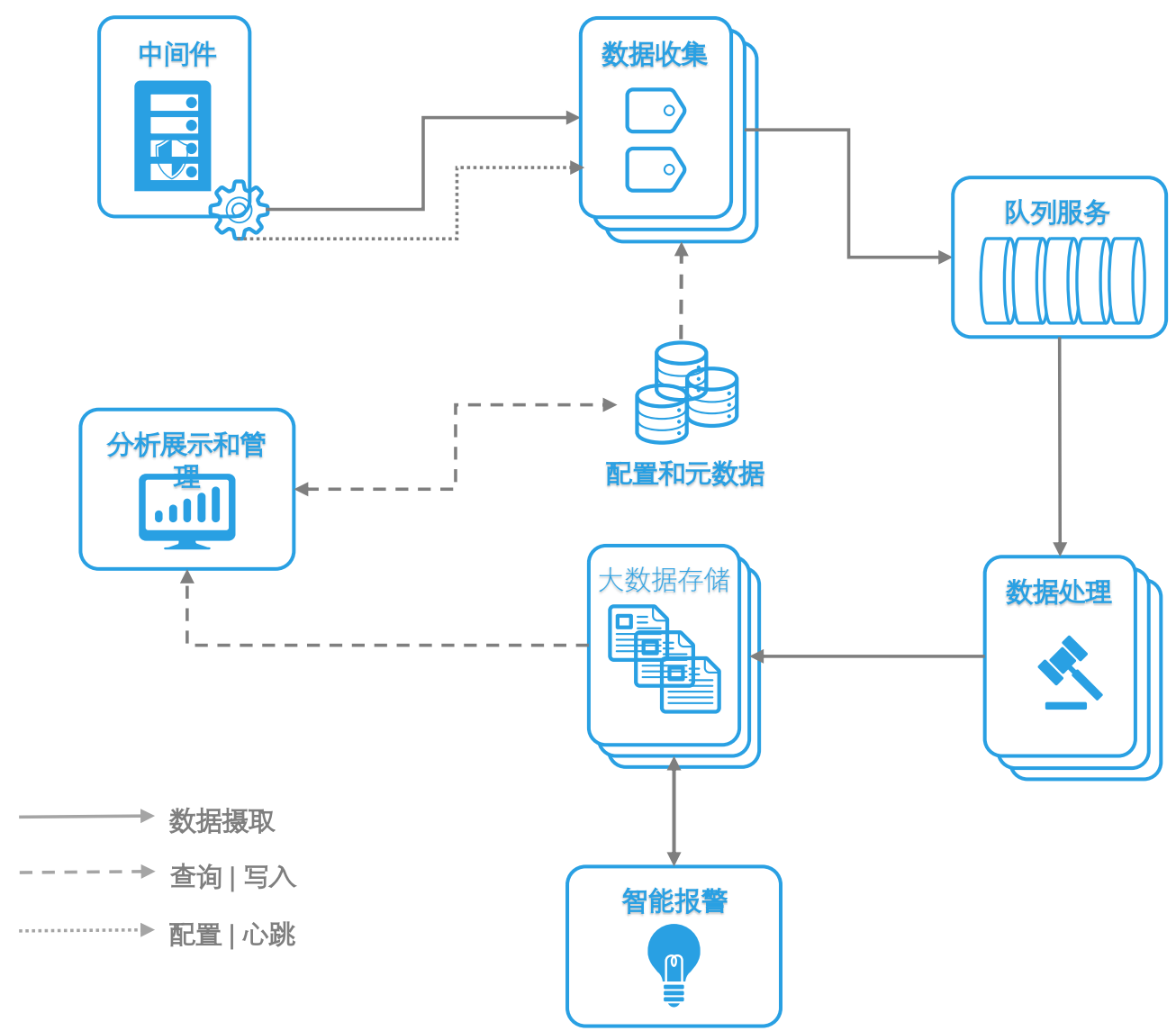
智能告警分析平台

多渠道接入告警数据，通过灵活的配置方式，创建告警指标、告警规则并建立多重告警发送通道；通过机器学习和AI算法，建立灵活的告警归并、压制和静默能力，并提供异常点检测、关联分析、根因分析等能力，辅助用户运维决策

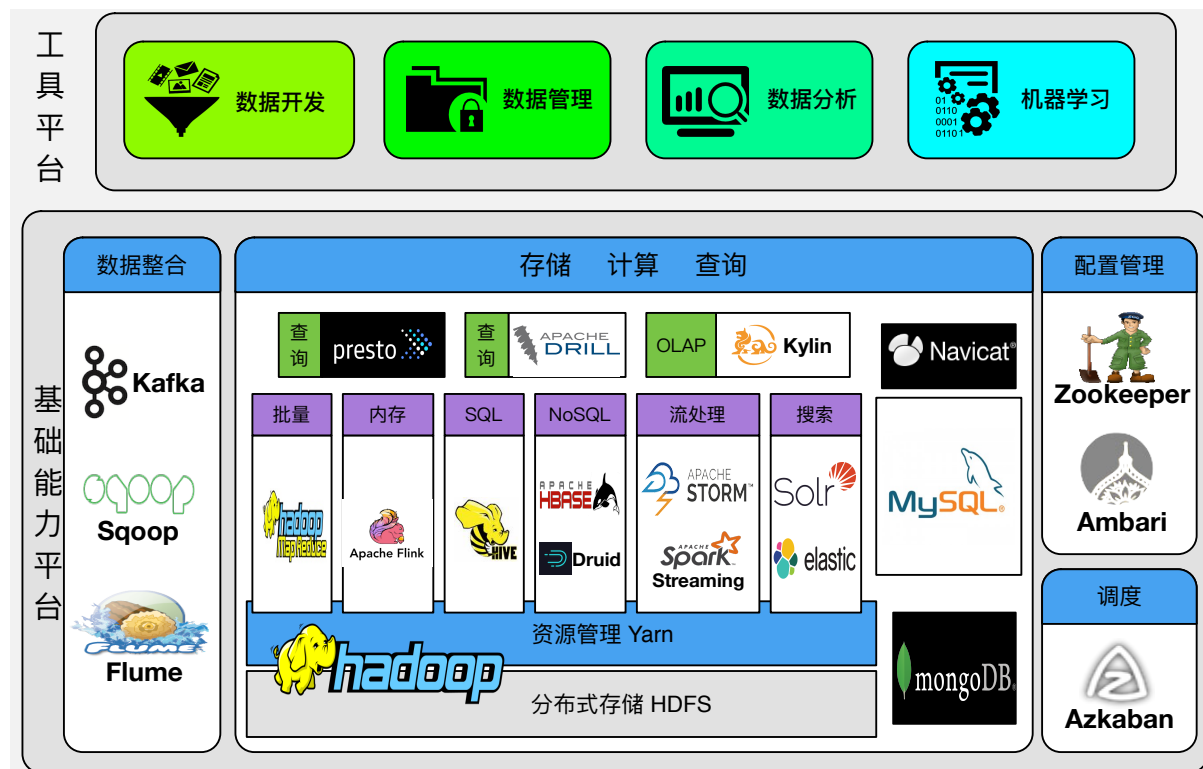
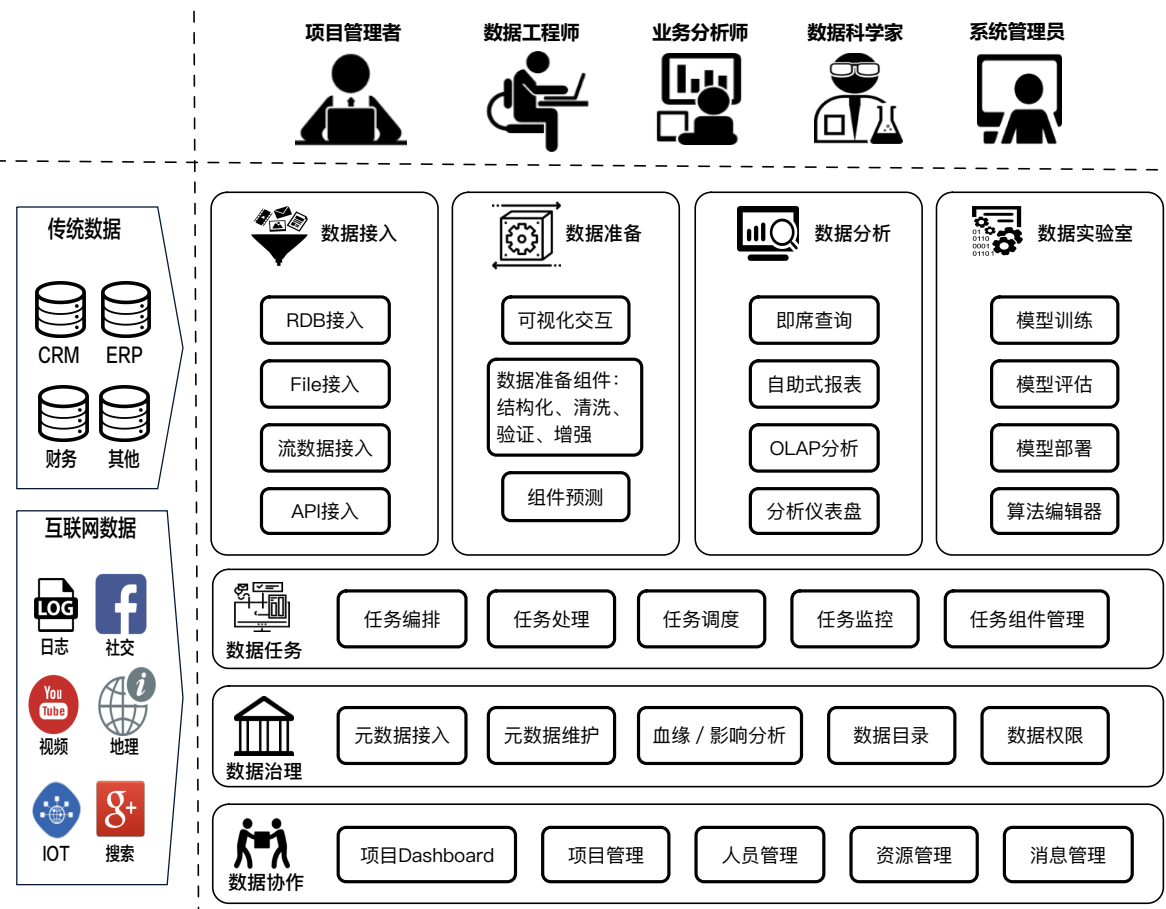
智能监控平台功能架构



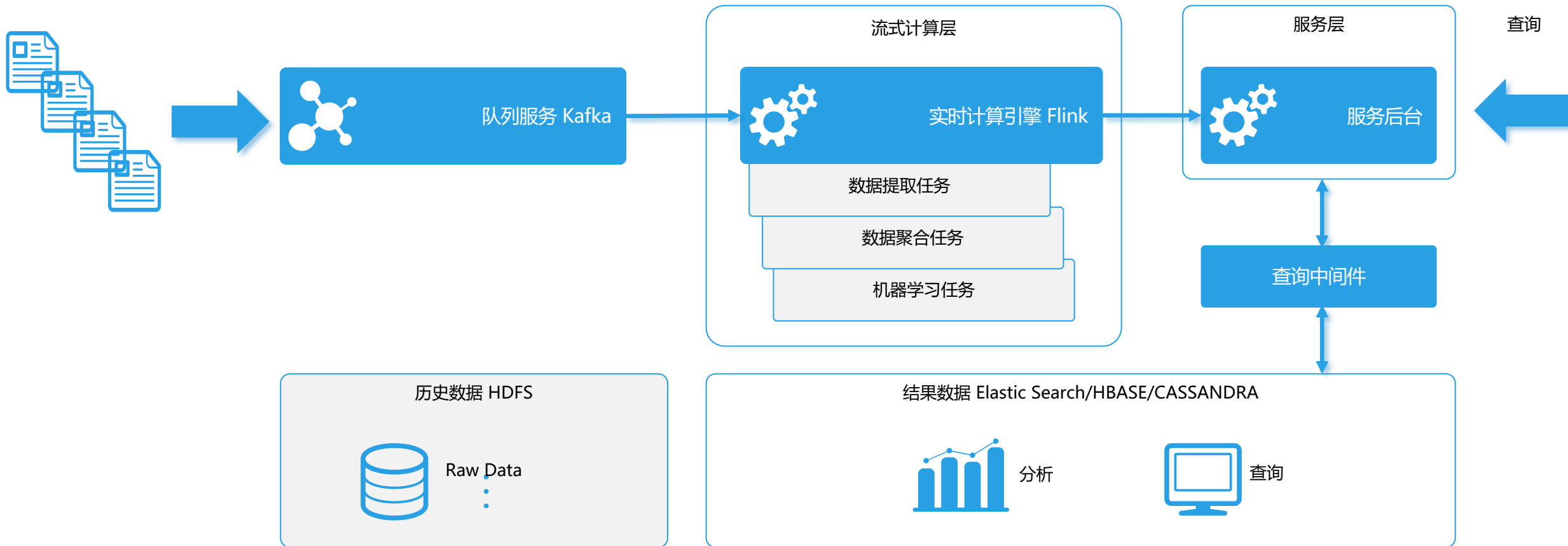
应用性能管理(APM)架构



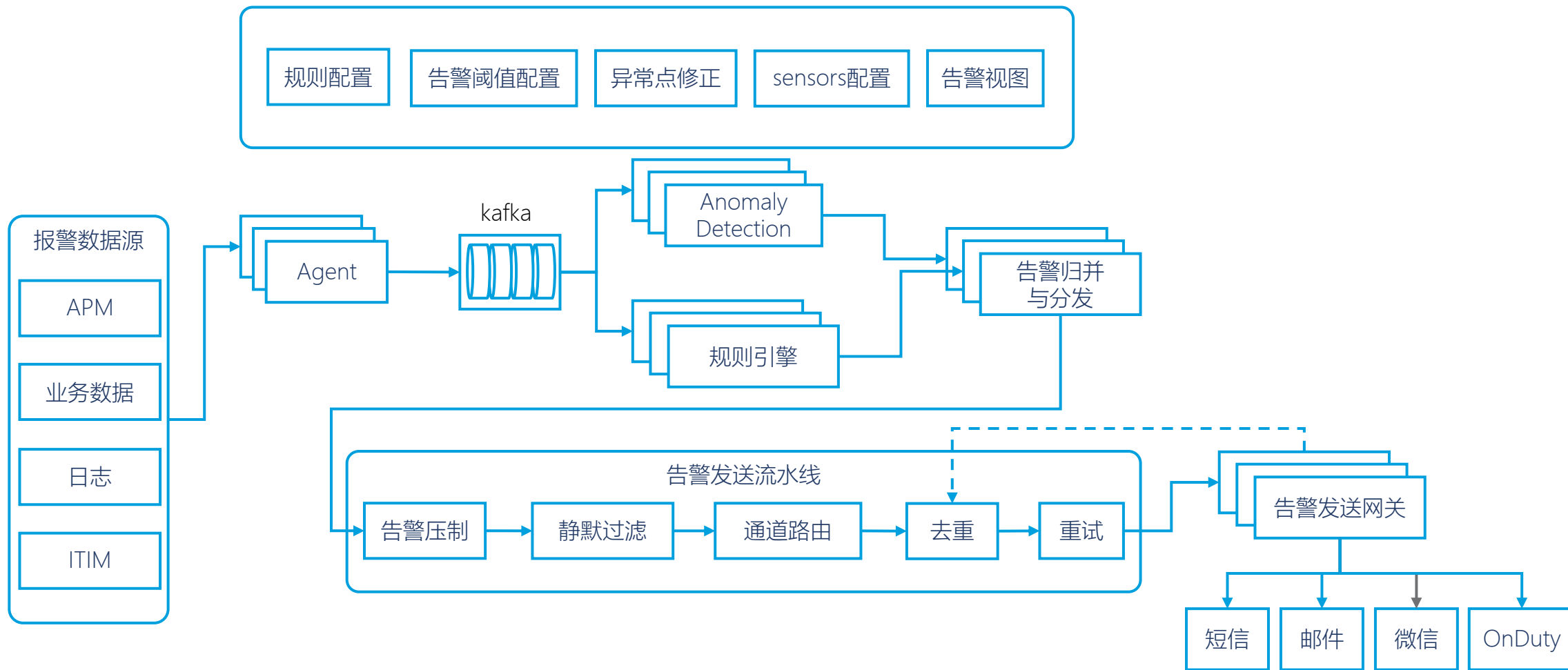
为了应对海量监控数据的存储和管理，基于Hadoop体系



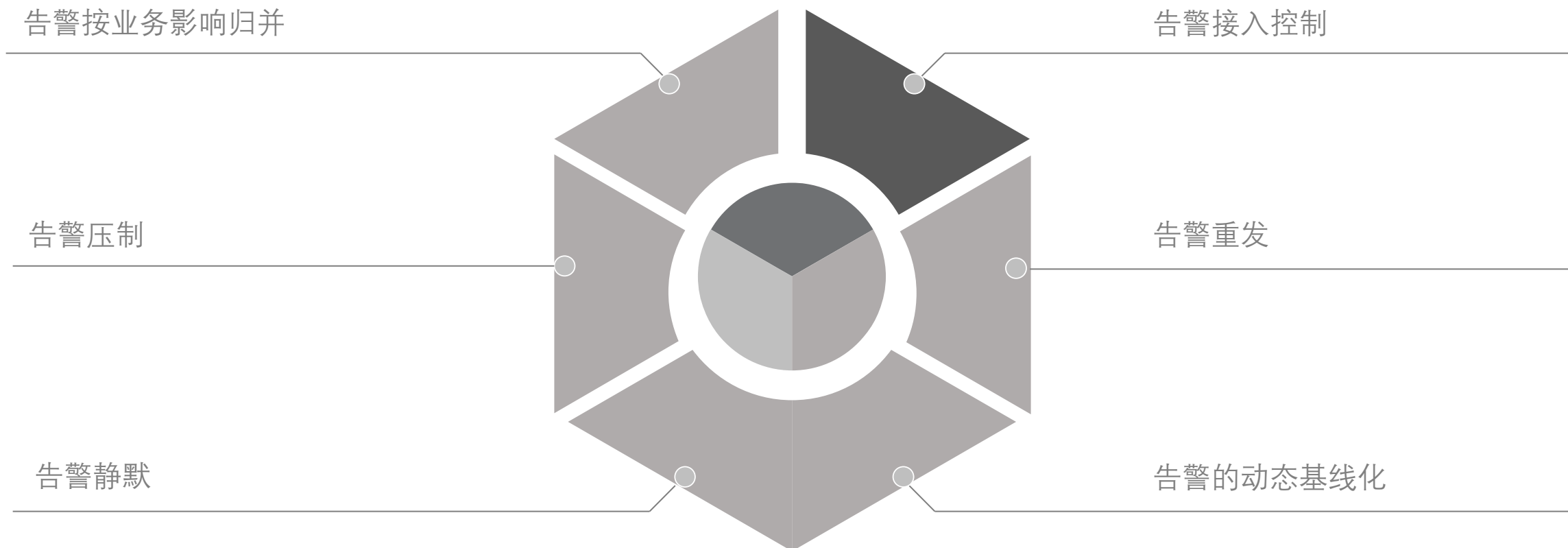
并在大数据管理平台之上打造了流式实时计算引擎...



...基于流式计算引擎构建智能告警平台



... 并提供智能告警分析能力



异常点检测算法概述



窗口高斯

适用于满足高斯分布的
指标数据

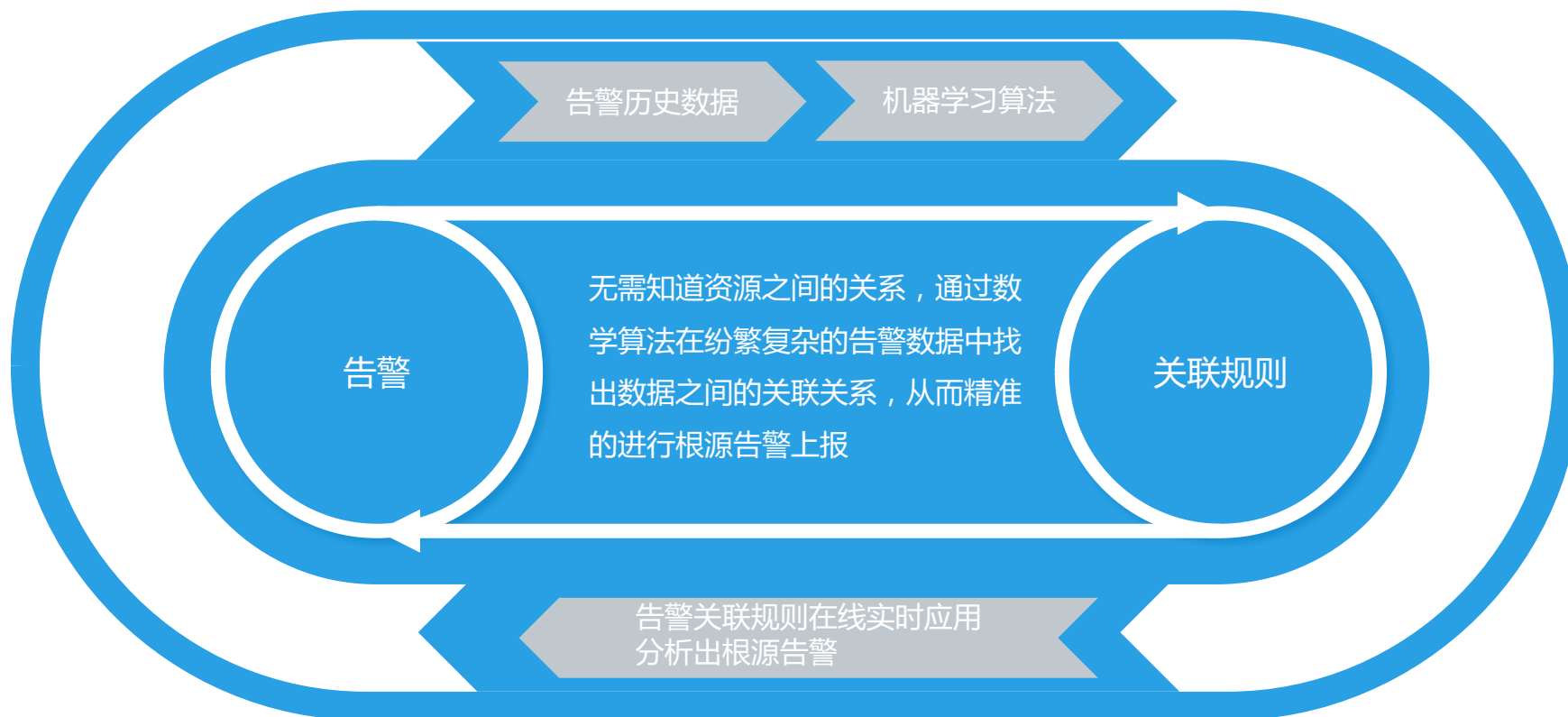
CAD KNN & Isolation forest

- 使用上下文信息
- 基于距离和随机森林
- 适用于周期序列数据

神经网络

- 基于神经网络算法
- 适用于周期&趋势时间序列数据

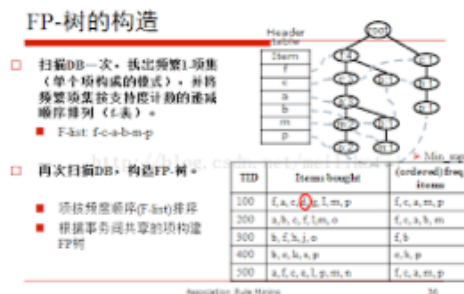
... 并根据告警的不同情形进行关联分析，自动发现根本问题



通过候选集生成和情节的向下封闭检测两个阶段来挖掘频繁项集。



不断地迭代FP-tree[4]的构造和投影过程



ECLAT: Another Method for Frequent Itemset Generation

- Determine support of any k-itemset by intersecting tid-lists of two of its (k-1) subsets.

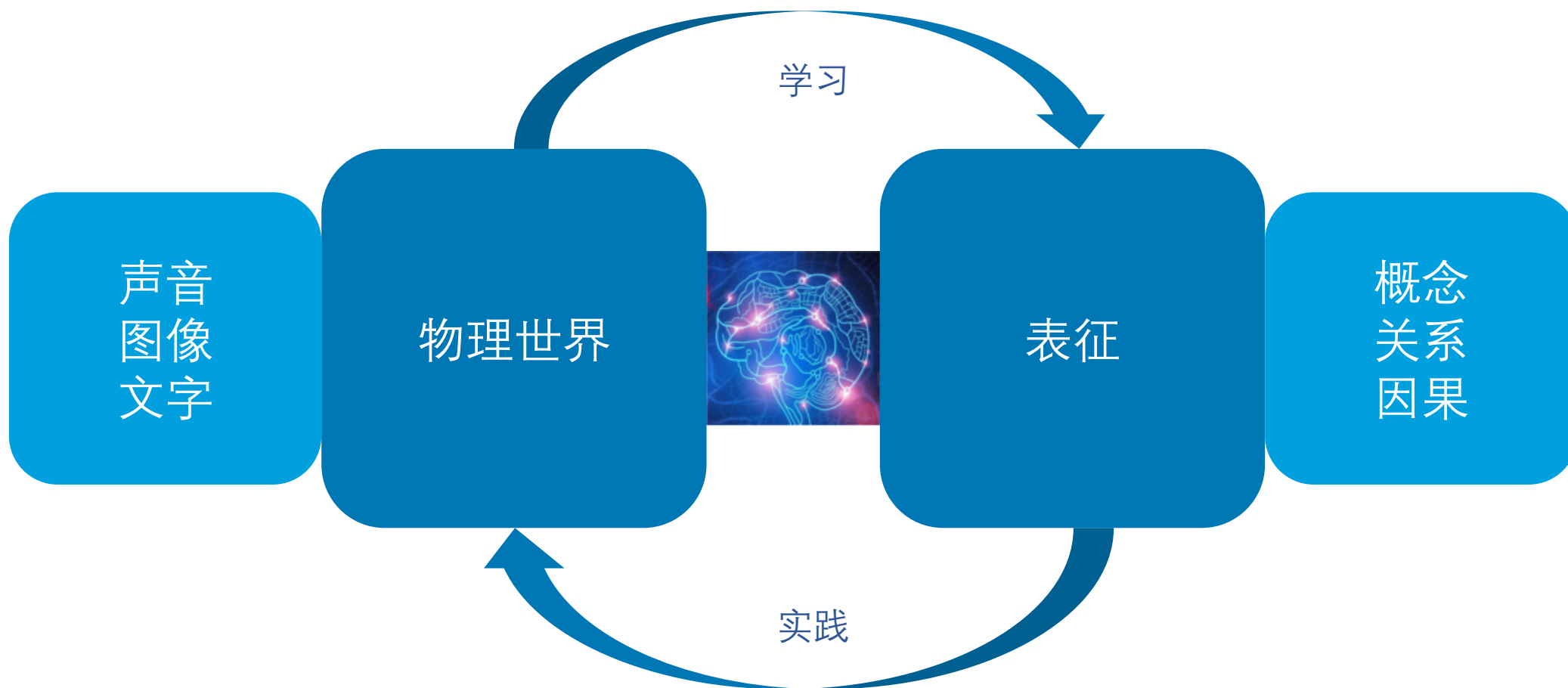


- 3 traversal approaches:
 - top-down, bottom-up and hybrid
- Advantage: very fast support counting
- Disadvantage: intermediate tid-lists may become too large for memory

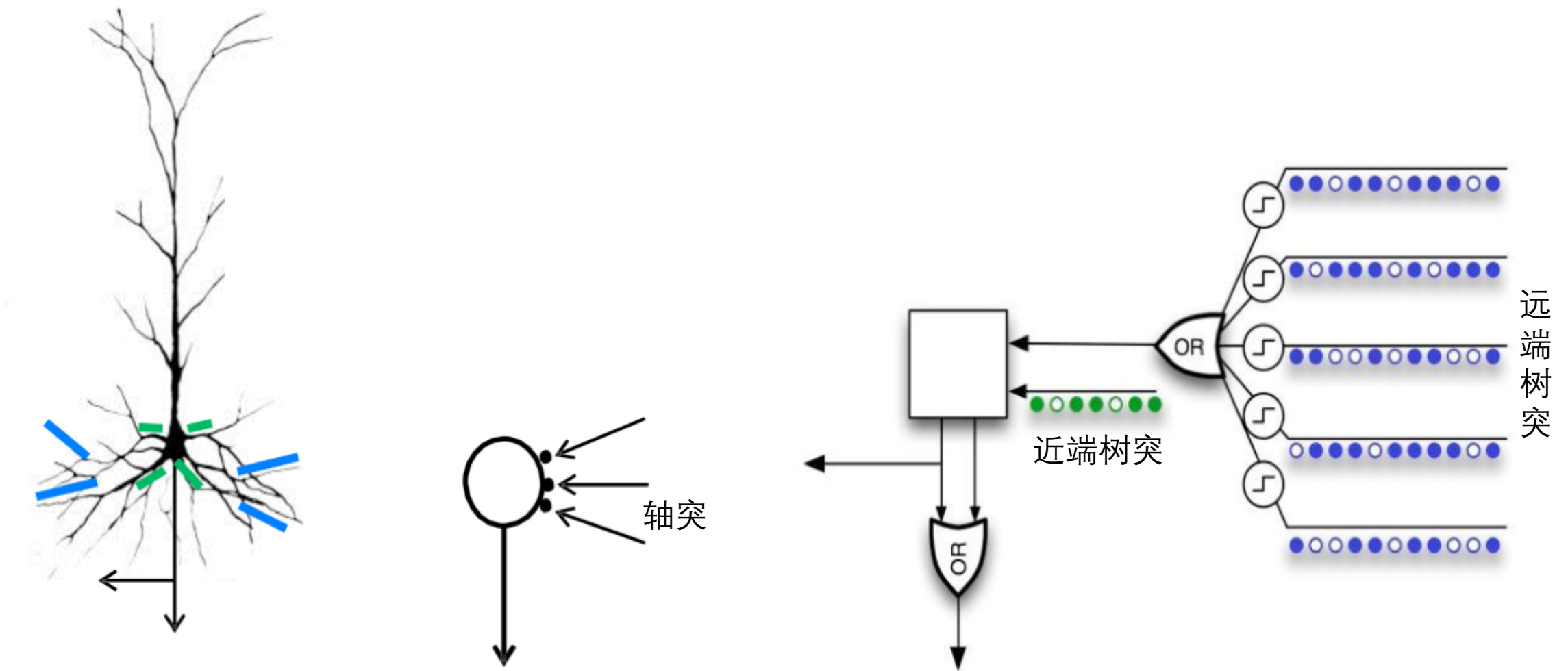
Agenda

- 谈谈应用性能管理 (Application Performance Monitoring)
- 为什么需要AIOps
- 构建智能应用性能管理平台
- 智能告警算法介绍
- 卓越运营(Operation Excellent)

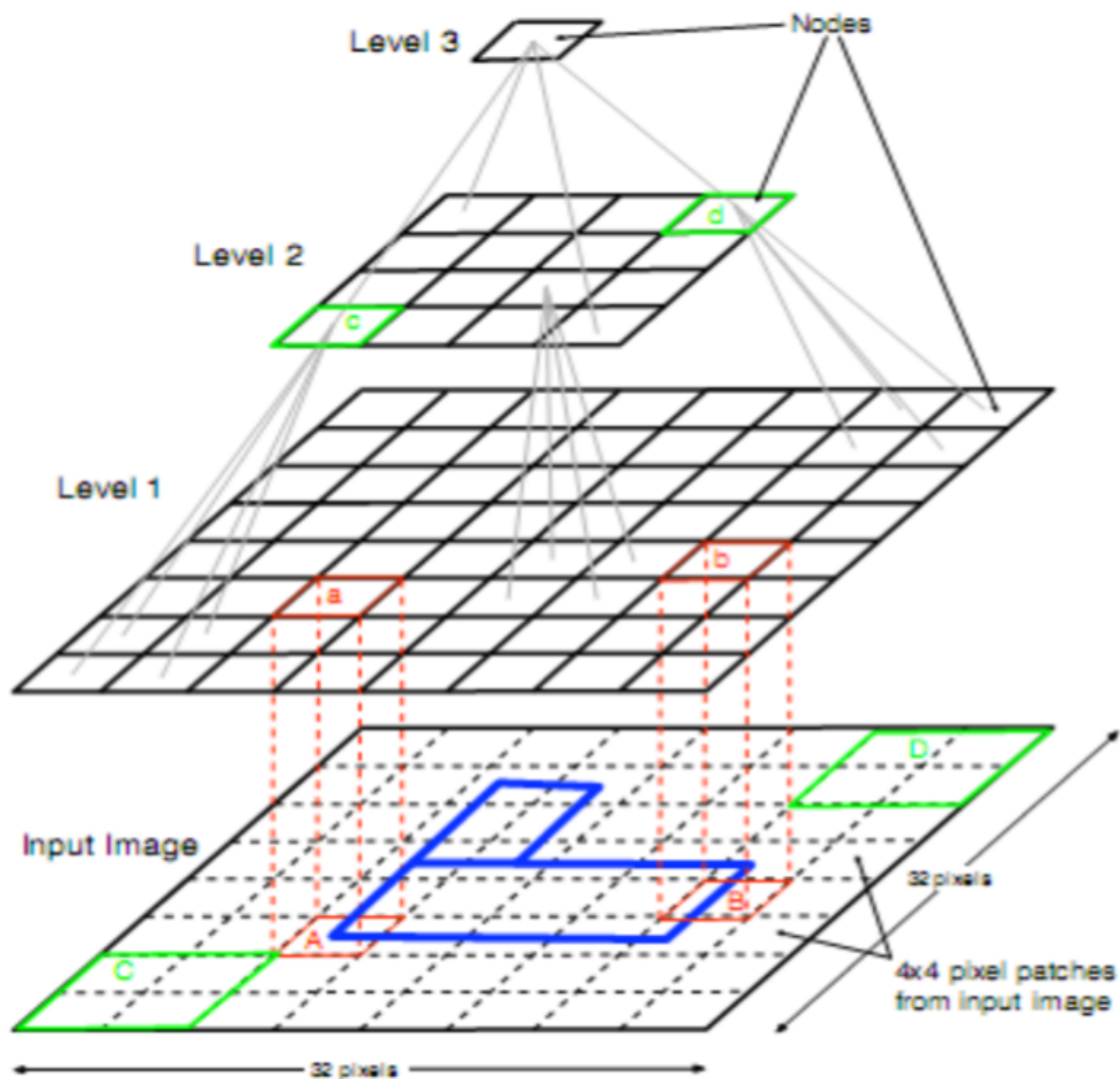
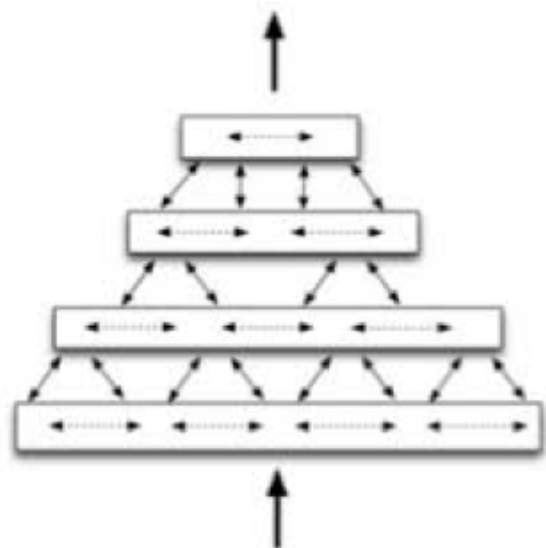
人类大脑如何认知



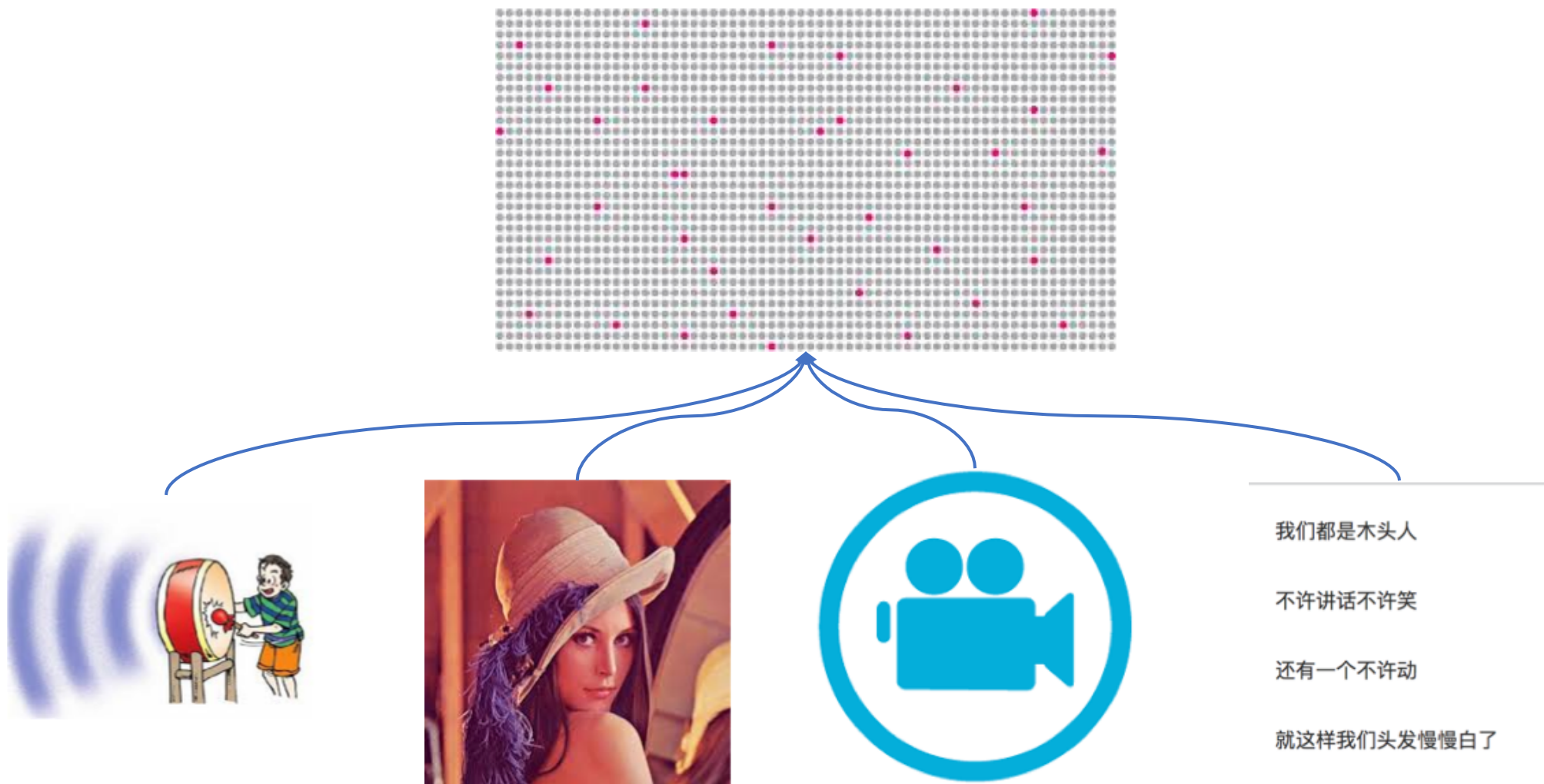
神经元



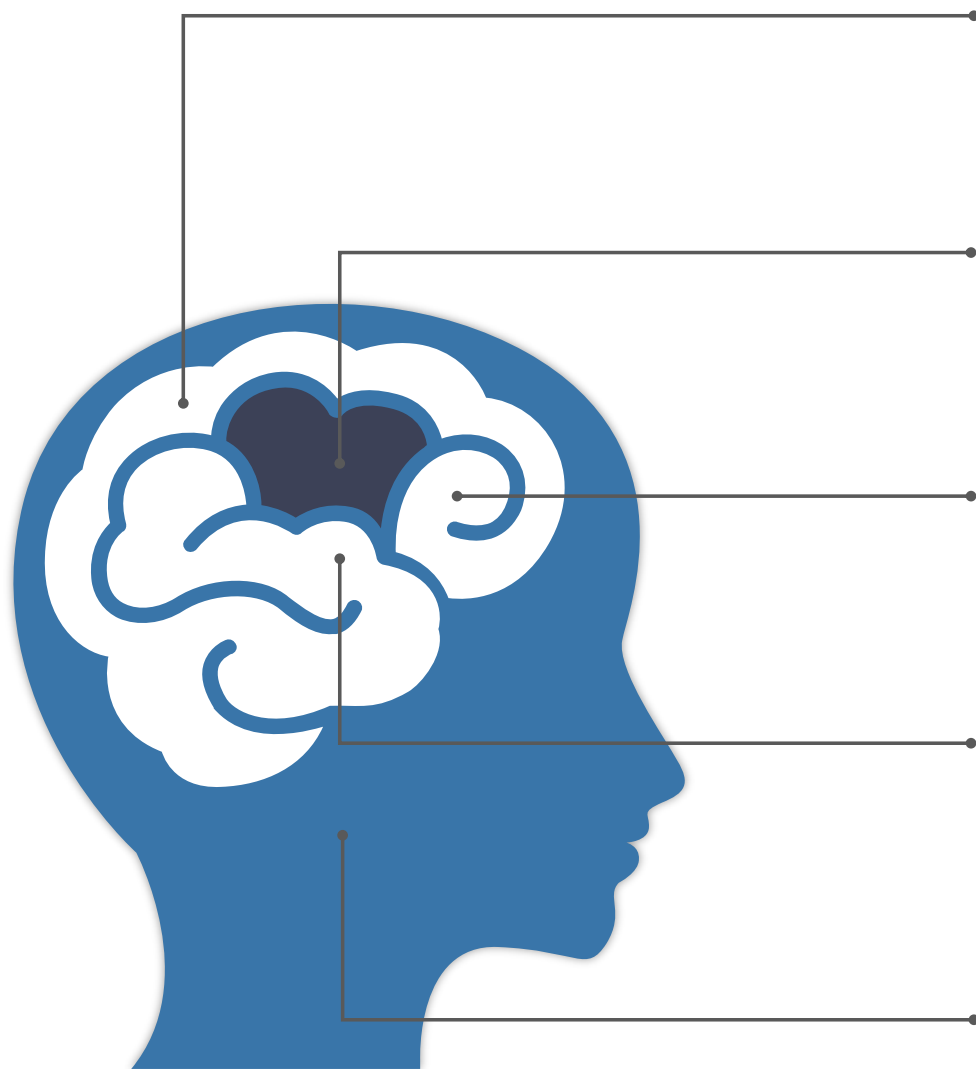
脑皮层



万物表征 – SDR (稀疏分布式表征)



脑皮质算法 (Hierarchical Temporal Memory)



持续在线学习

根据快速变化的数据流进行**实时调整**

时间和空间分析

可以同时模拟**时间和空间**的变化

通用性

•通过**通用性**的大脑皮层算法，进行预测和建模

强大的异常检测能力

实时检测数据流的扰动，不依靠**僵化的阈值**设置和过时的算法

存储

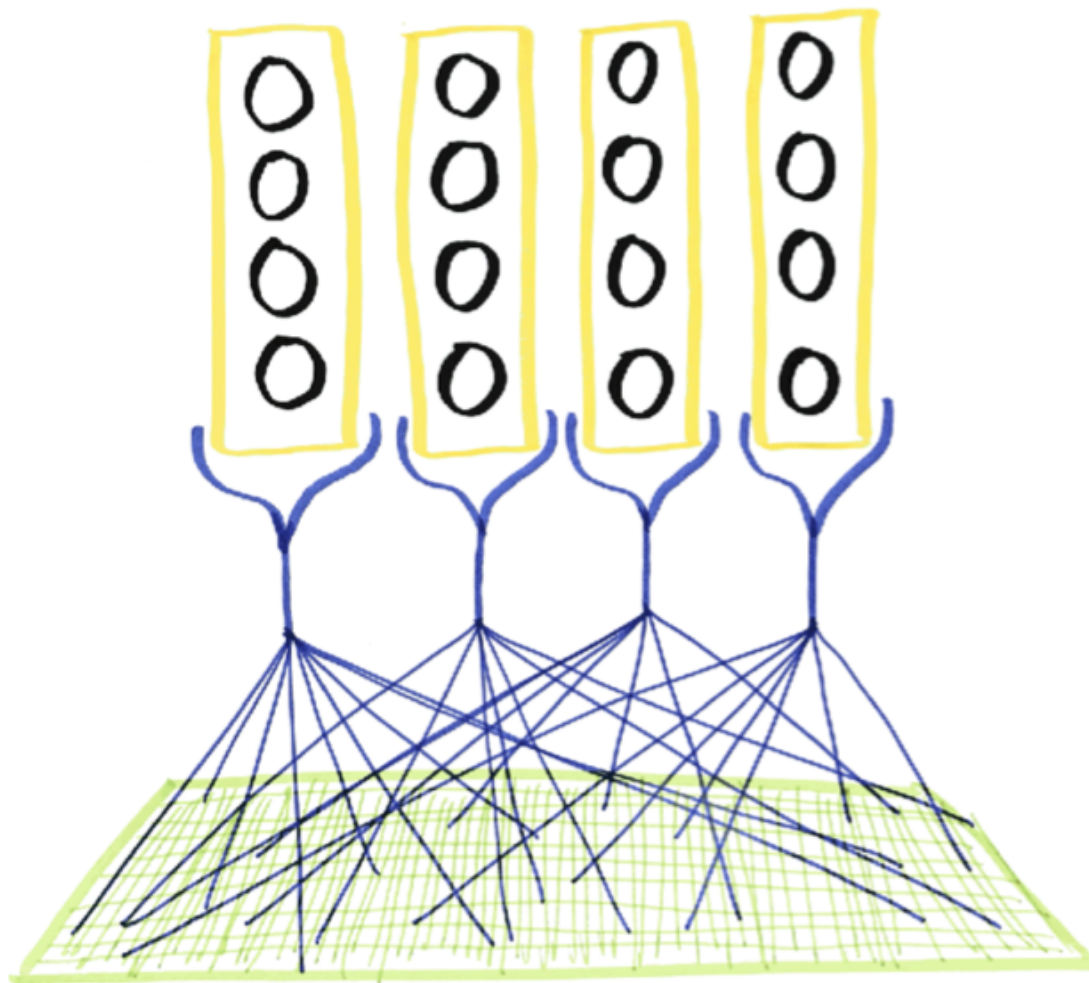
•层级实时存储算法

空间池（Spatial Pooling）

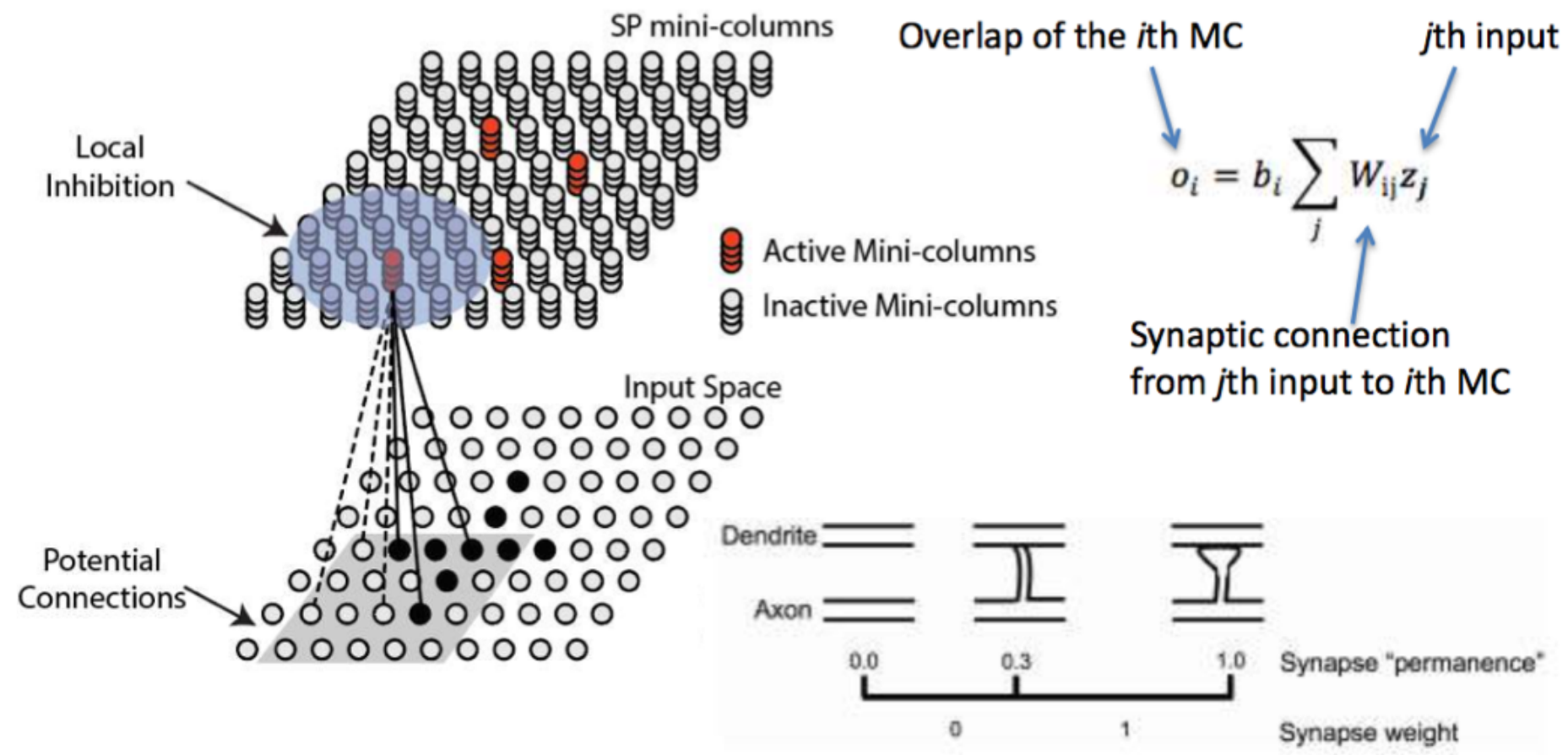


中生代技术
FRESHMAN TECHNOLOGY

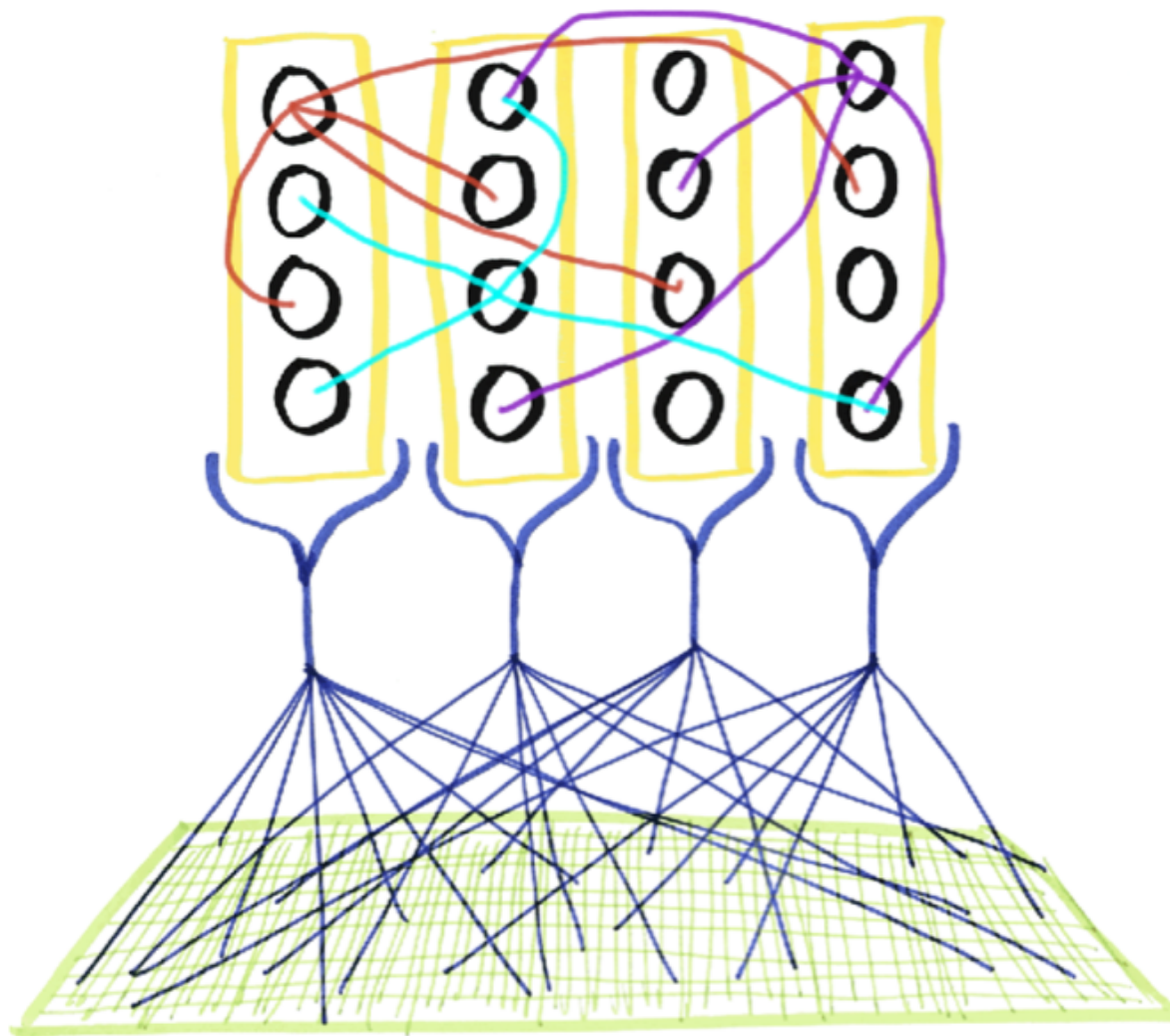
IT大咖说
知识共享平台



空间池（Spatial Pooling）



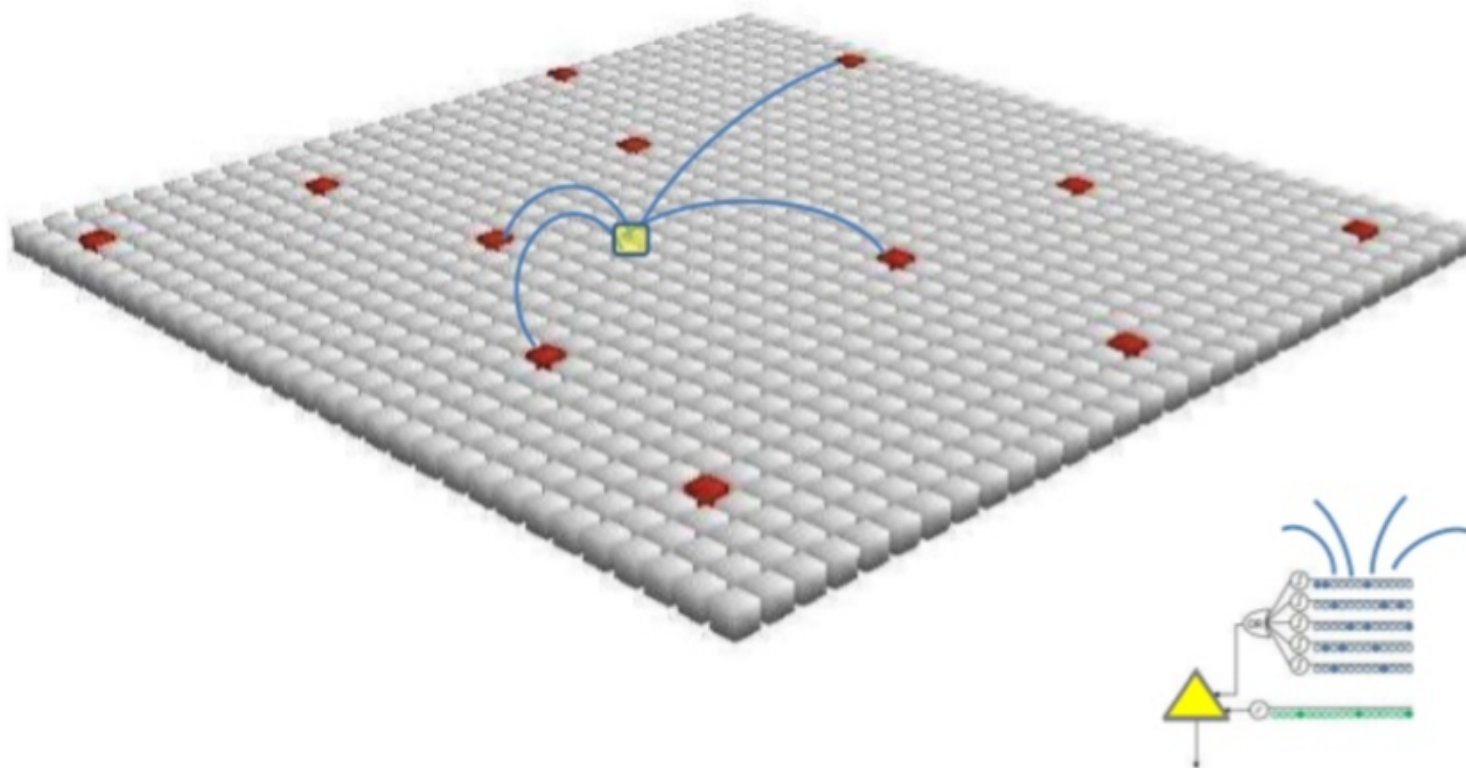
暂时性记忆 (Temporal Memory)



暂时性记忆 (Temporal Memory)

Learning Transitions

Form connections to previously active cells.
Predict future activity.



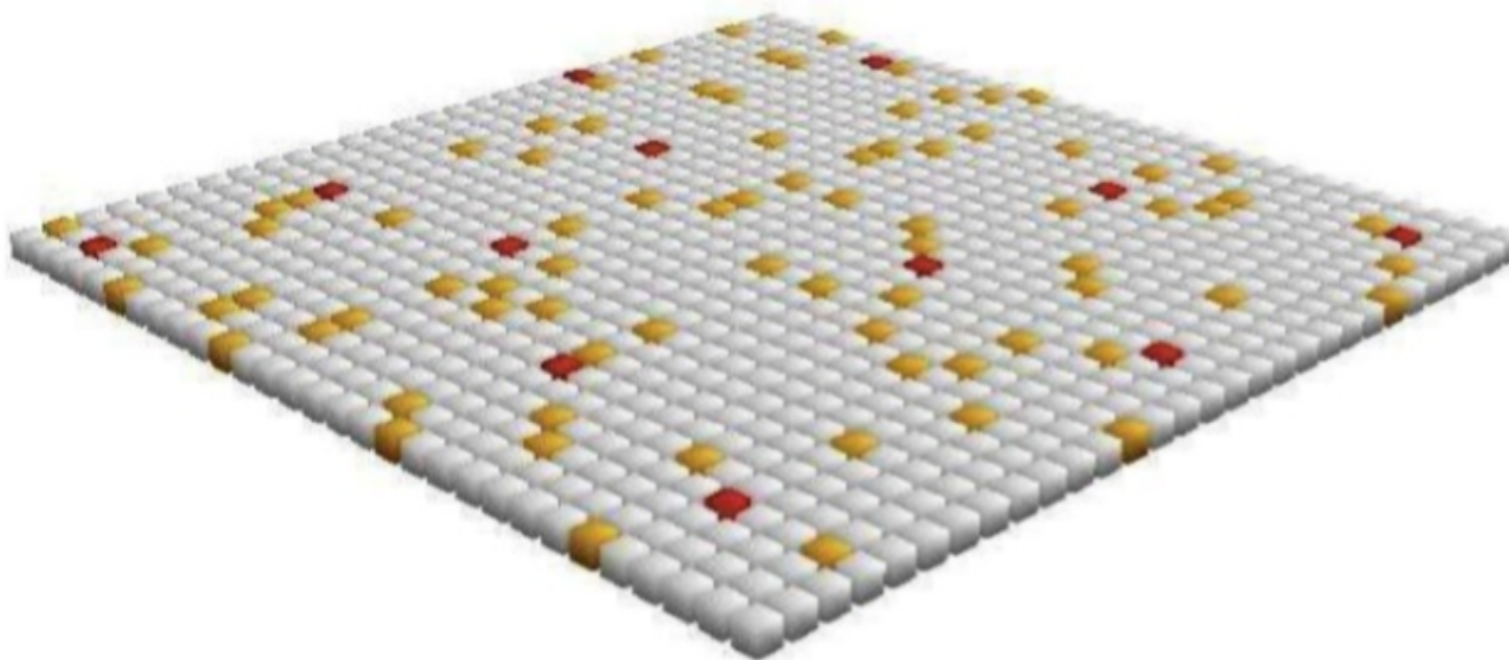
暂时性记忆 (Temporal Memory)

Learning Transitions

Multiple predictions can occur at once.

A-B A-C A-D

Clip slide



- This is a first order sequence memory.
- It cannot learn A-B-C-D vs. X-B-C-Y.
- Mini-columns turn this into a high-order sequence memory.

使用HTM算法的异常点检测



使用HTM算法的异常点检测



```
/**
 * Create HTM Model to be used by NAB
 * @param modelParams OPF Model parameters to parameters from
 */
public HTMModel(JsonNode modelParams) {
    LOGGER.trace("HTMModel({})", modelParams);

    // Create Sensor publisher to push NAB input data to network
    supplier = PublisherSupplier.builder()
        .addHeader("timestamp,value")
        .addHeader("datetime,float")
        .addHeader("T,B")
        .build();

    // Get updated model parameters
    Parameters parameters = getModelParameters(modelParams);

    LOGGER.info("RUNNING WITH NO EXPLICIT P_RADIUS SET");

    // Create NAB Network
    network = Network.create( name: "NAB Network", parameters)
        .add(Network.createRegion( name: "NAB Region")
            .add(Network.createLayer( name: "NAB Layer", parameters)
                .add(Anomaly.create())
                .add(new TemporalMemory())
                .add(new SpatialPooler())
                .add(Sensor.create(ObservableSensor::create,
                    SensorParams.create(SensorParams.Keys::obs, ...values: "Manual Input", supplier))))));
}
```



使用HTM算法的异常点检测

告警

首页告警集成设置APM

test退出

系统CPU负载

EditStatus

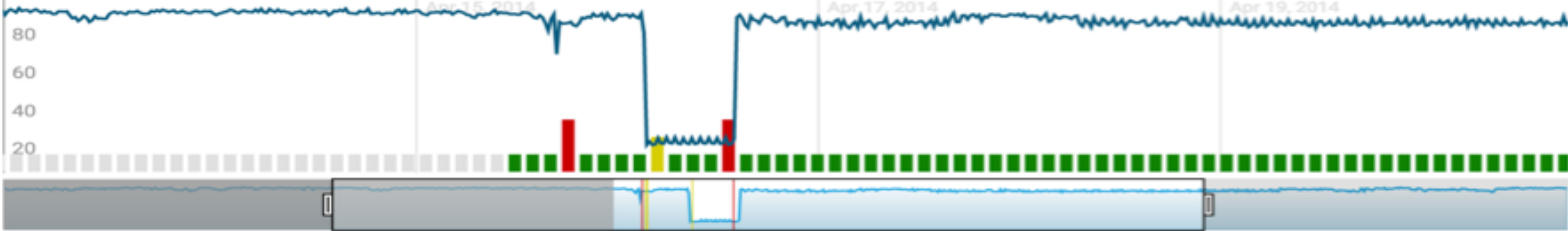
状态:RUNNING 创建时间:2018-01-23 13:03:13

'SiData'.metric(node_cpu.load).over(instance=bj-rc-dptd-web-uat-v-test-1).by(instance).timewindow(1MIN).aggregate(AVG)

异常历史

host=bj-rc-dptd-web-uat-v-test-1

1小时



通知内容详情

系统 {{host}} 的 cpu.load={{value}}, 异常信度 {{confidence}}

告警 {{level}}

告警时间 {{event_time}}

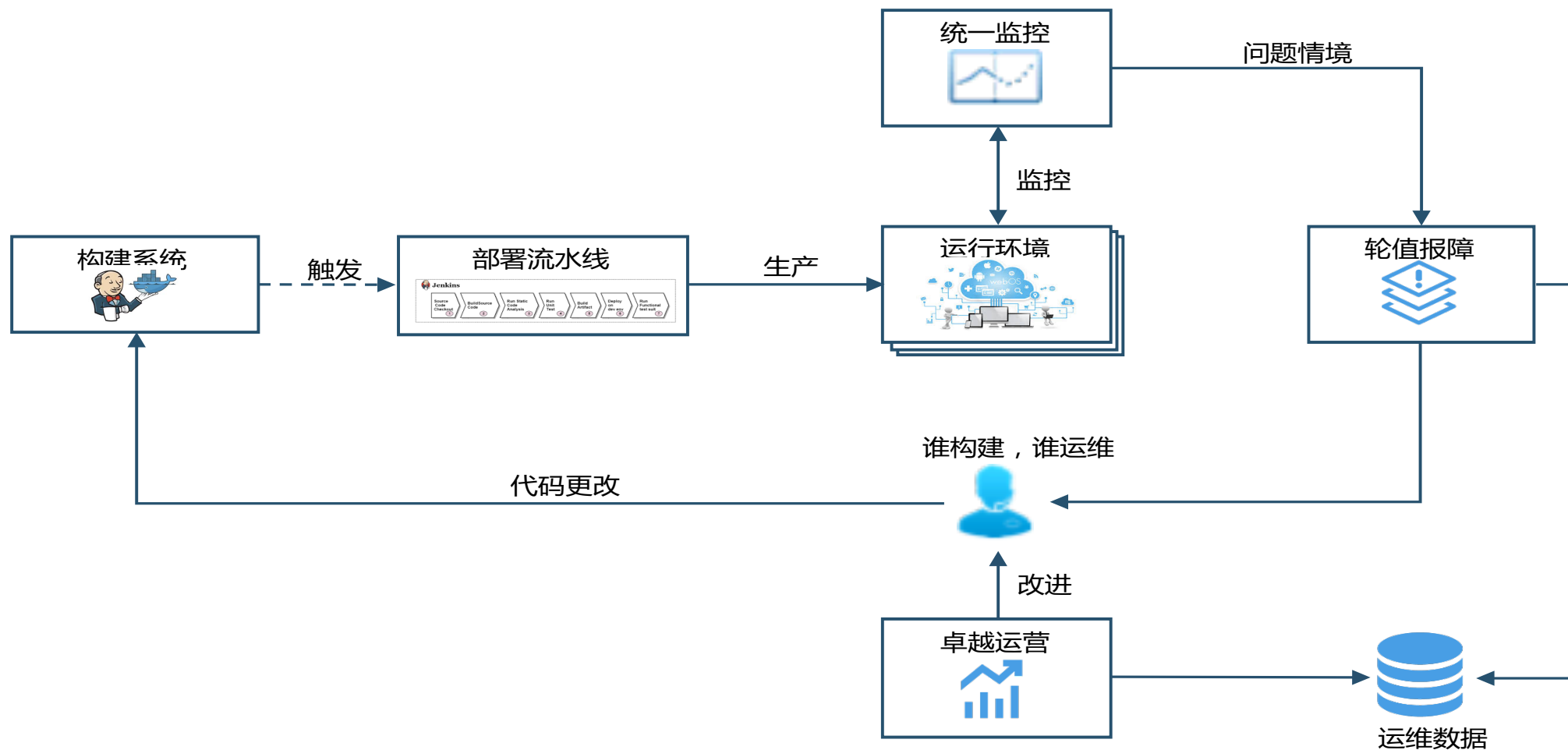
告警列表

告警等级	告警名称	相关指标	告警内容	状态
2级	SI-IDLE	node_cpu.idle	系统 bj-rc-dptd-web-uat-v-test-1 的 cpu.load=12.6, 异常信度 0.997 告警 1级 告警时间 2018-01-24 18:23:17	未处理

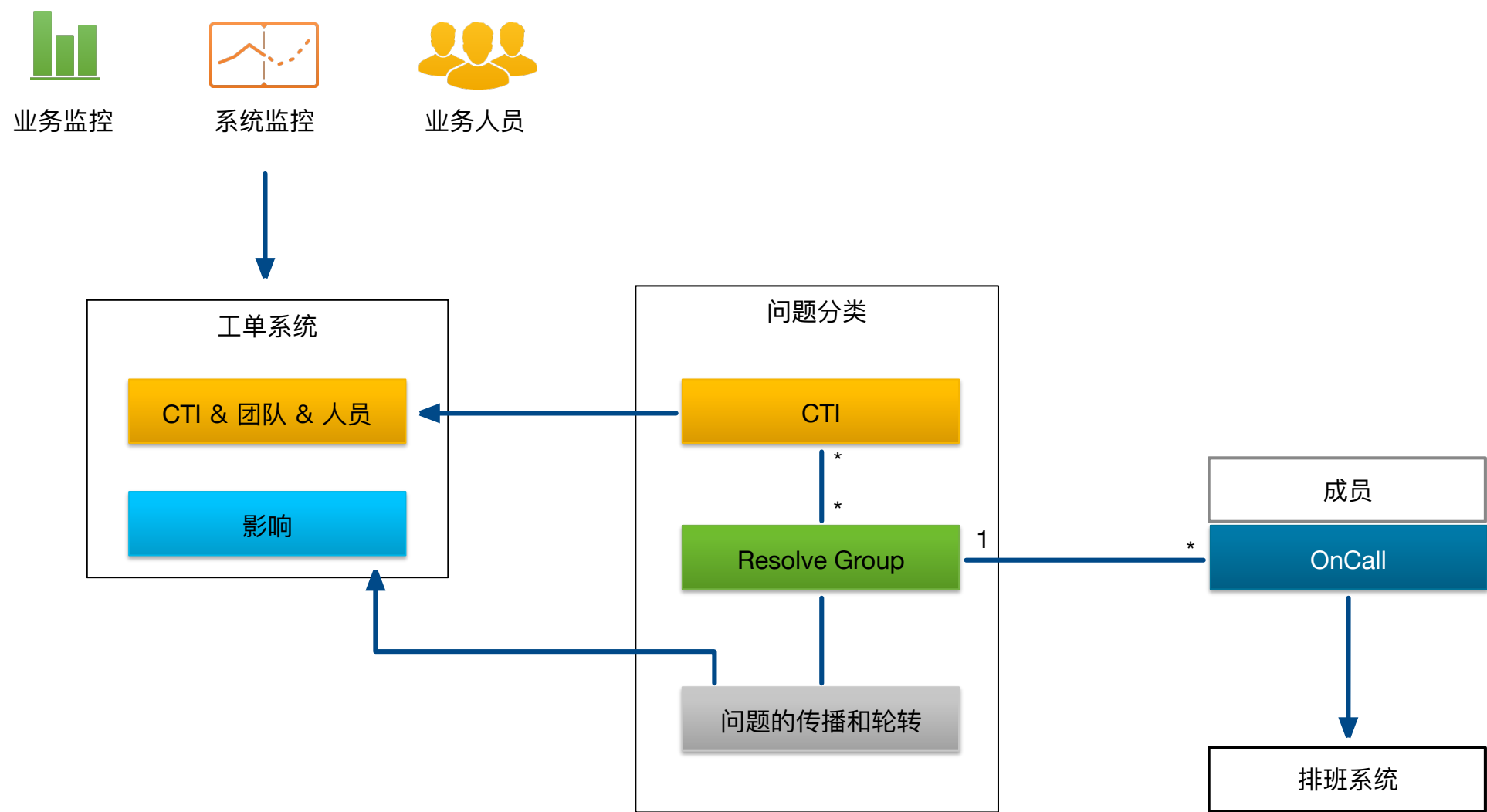
Agenda

- 谈谈应用性能管理 (Application Performance Monitoring)
- 为什么需要AIOps
- 构建智能应用性能管理平台
- 智能告警算法介绍
- **卓越运营(Operation Excellent)**

结合DevOps平台，构建“开发+部署+监控+告警+”



智能报障平台(OnDuty)



智能报障平台(OnDuty)

- 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

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	解除关联

添加关联

故障根因管理

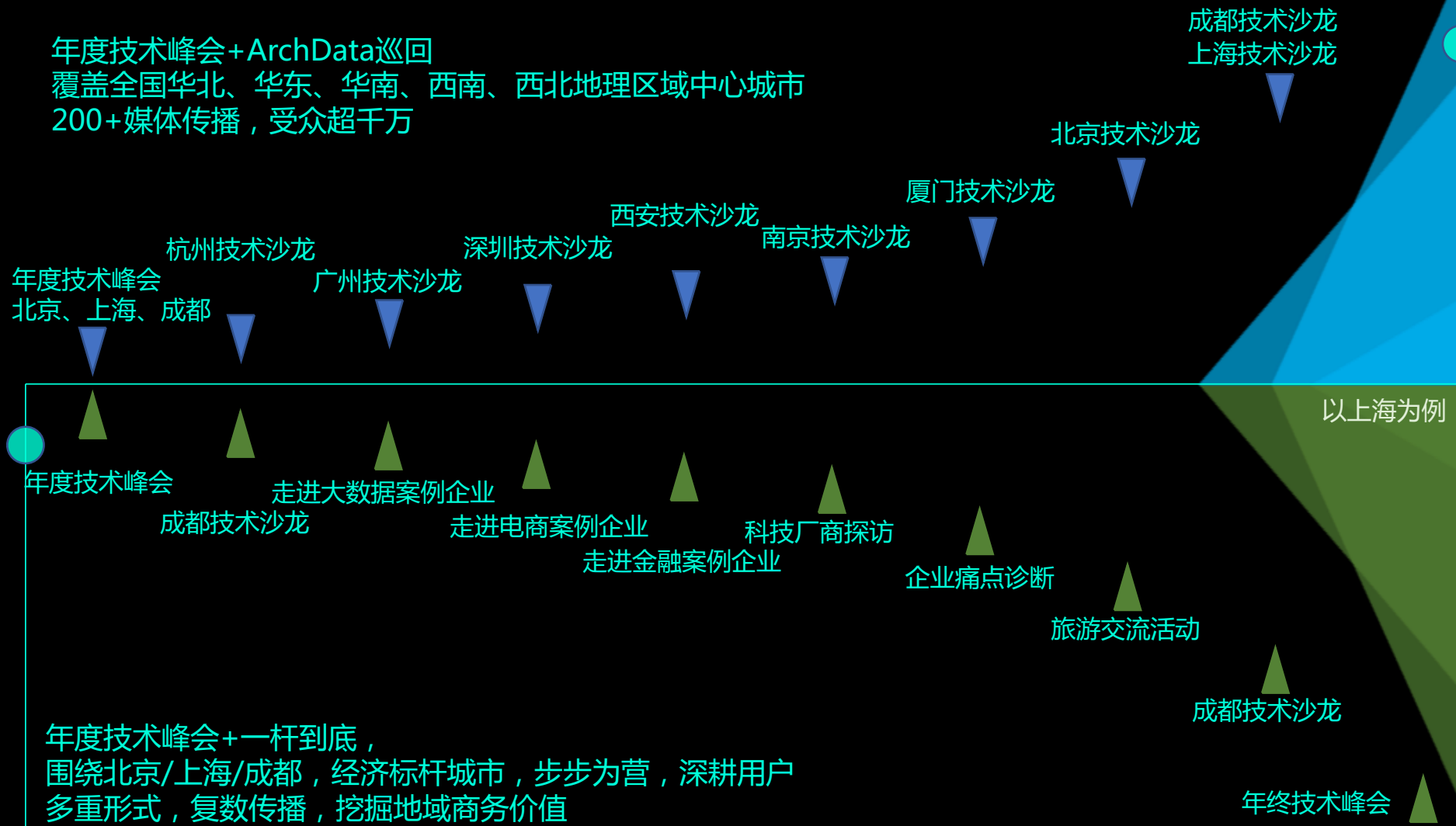
名称	描述	操作
系统维护	计划内的人为事件，如计划内关机、重启、网站升级、网站维护等。	编辑 删除
网络故障	网络原因导致的故障，如主干网络发生的故障。	编辑 删除
硬件故障	服务器硬件导致的故障，如磁盘故障、内存故障、网卡故障等。	编辑 删除
软件故障	软件原因导致的故障，如操作系统故障、Web服务器运行故障、应用程序BUG、软件服务容量超载等。	编辑 删除

取消

保存

横跨全年的品牌传播与实效营销盛宴

同城合纵



全域连横

合纵连横，在中国开发者群体中缔造品牌营销奇迹



中生代技术
FRESHMAN TECHNOLOGY

ArchData技术峰会
聚焦人工智能、大数据、基础架构、区块链等
前沿课题

中生代技术提供咨询内训服务
技术架构，研发管理，敏捷开发，大数据
微服务，AI，机器学习等

中生代技术提供人才服务
对接研发主管，内推精准人才