

之家云能力建设与管理实践

演讲嘉宾：黄振

CONTENTS



起源

之家云的成立



历程

发展历程与现状



演进

聚焦降本增效



未来

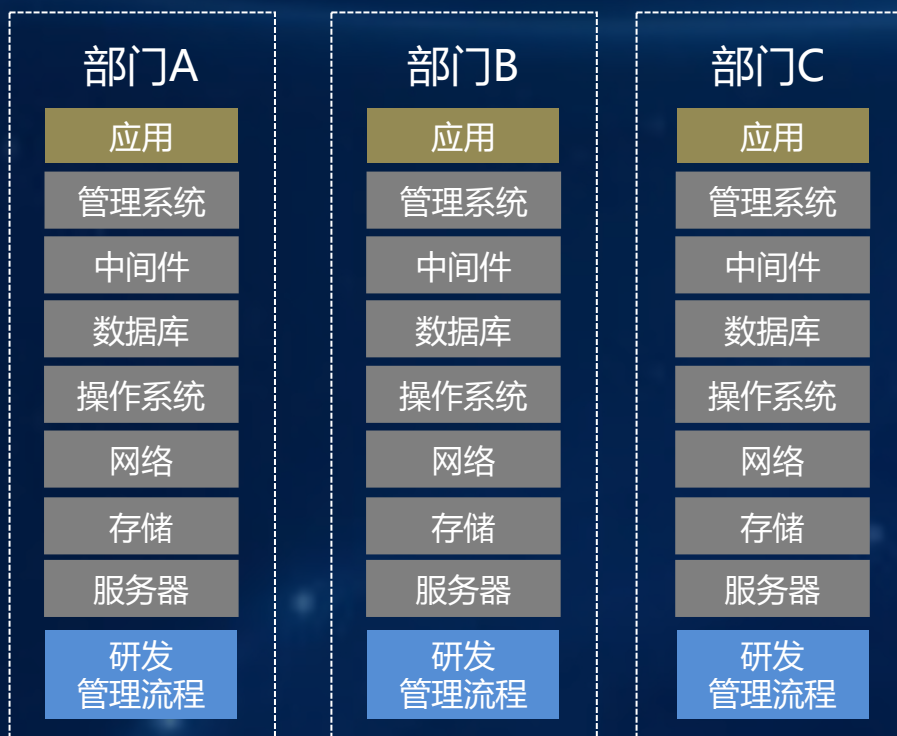
挑战与展望

起源

之家云开端

+ P 搭建稳定好用的云计算，实现IT基础设施统一，共识公司级研发度量指标，提升整体研发效能

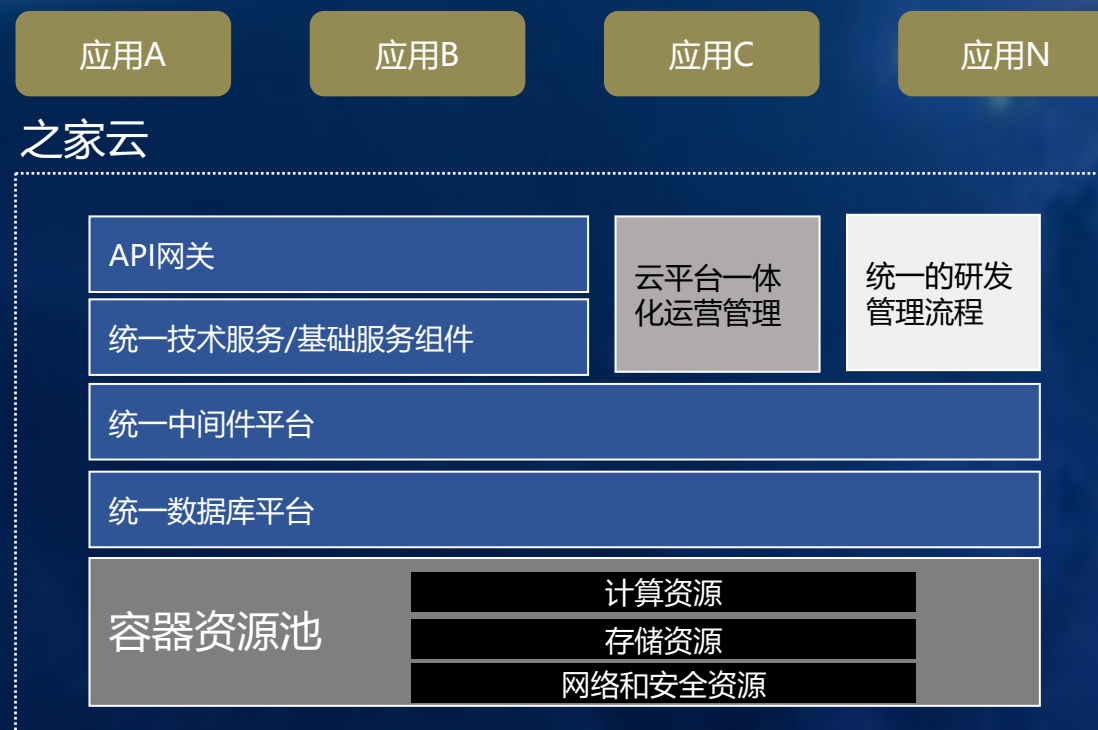
传统建设模式



烟囱式结构大幅降低了新业务启动速度和资源复用率

- 产品烟囱：不同的产品间，定位和功能雷同
- 系统烟囱：总是重复开发，浪费研发资源
- 组织烟囱：隐性的部门墙，缺乏协同与信息共享

云计算建设模式



使用统一的IT基础设施与研发管理流程，让各BU专注业务研发

- 通过公司级的IT基础设施，减少系统重复开发，提升资源使用效率
- 通过公司级研发效能度量指标，提升管理、技术、协作等多方面的系统性改进

历程

发展历程与现状

+ 之家云平台经历了1.0搭建、2.0融合、3.0进化三个阶段，现已全面进入4.0平台突破阶段

1.0 平台搭建

(2018-2019)

规划、建设

- ✓ 云战略正式启动，云计算项目升级为部门级PMO项目并命名为“汽车之家云平台”，云战略全速前进
- ✓ 建立云平台团队，专职负责之家云的研发与运营
- ✓ 应用中心A-One上线并整合各BU发布系统9套，以应用配置管理、容器化等业务具体问题为切入点，制定并达成分步搭建目标

2.0 平台融合

(2019-2020)

统一、规范

- ✓ 资源与技术统一，打破烟囱，整合成统一的资源池，实现资源的统一管理、实现技术中间件的统一运维和管理，**原地标准化中间件资源池**
- ✓ 推出研发类、数据库类、中间件类**研发规范**，建立基于之家云的公司级研发规范
- ✓ 完善之家云产品功能、保证之家云可靠性，应用容器化比例超过**90%**、**100%**核心API接入云网关

3.0 平台进化

(2020-2022)

进化、赋能

- ✓ 从局部云化，到实现研发全链路技术栈的全面云化，新发布分布式服务平台、企业效能平台、资产云、图数据库等**13个**云服务
- ✓ 之家云产品从可用，到移动化的随处可用，推出包含A-One、云日志、云监控、AutoBI、项目管理等功能的掌上之家云1.0
- ✓ 业务范围从对内提效，扩展到对外赋能，协助人工智能等业务**接入公有云服务市场**

4.0 平台突破

(2022至今)

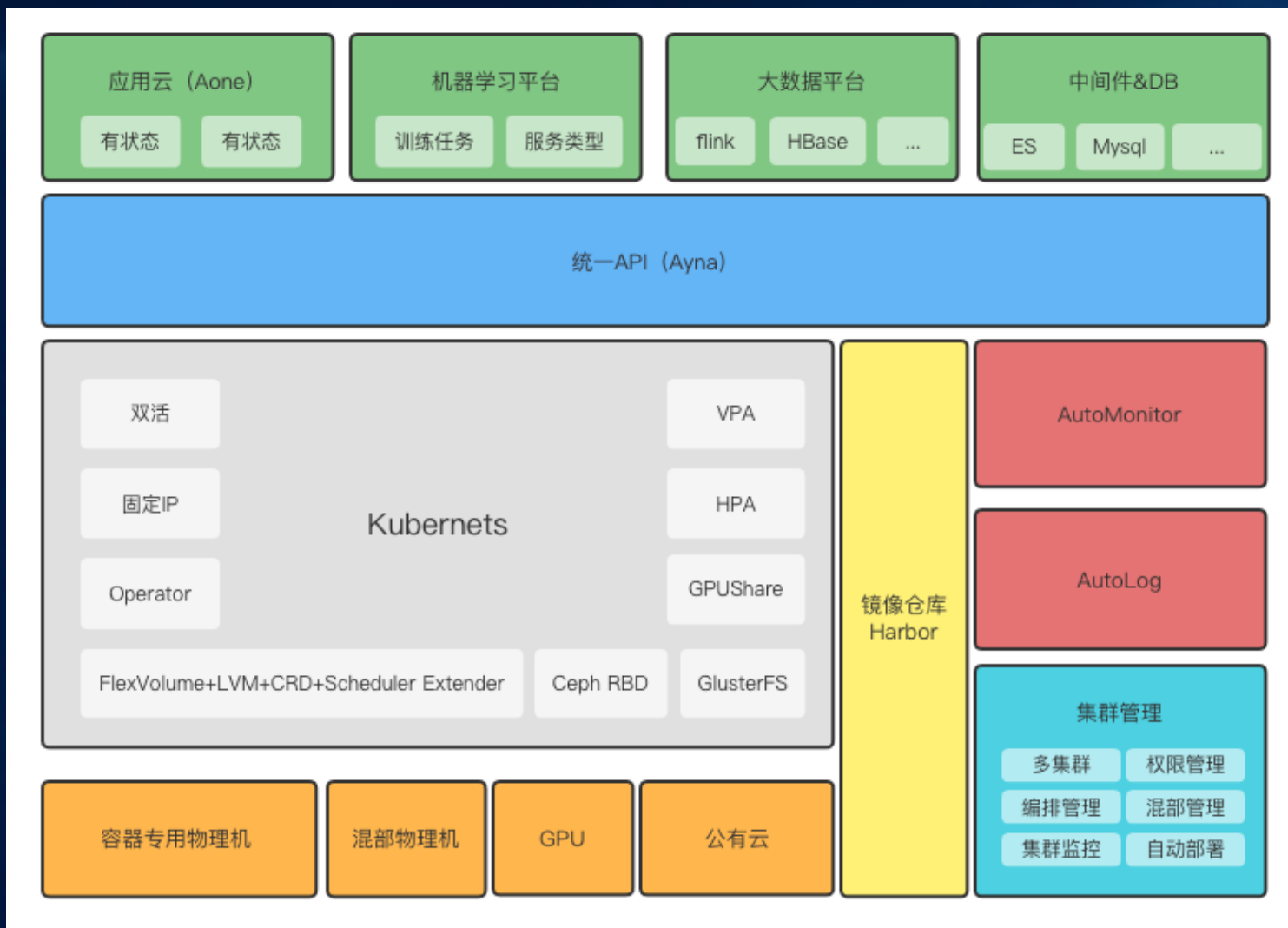
突破、革新

- ✓ 数据驱动运维，依托覆盖全面的用户及运维系统数据，以数据来驱动运维决策与执行
- ✓ 降本增效，落地之家FinOPS体系，通过盘活存量IT资源，以存量资源支撑近期业务增长，并为远期业务增长扩充储备，首度实现主机0采购
- ✓ 技术创新，通过持续在云原生、大数据领域创新，落地大规模混部、中间件容器化，在实践中继续突破，保持运维体系的优越性

+ 之家云以云资源及技术服务为导向，以技术序列员工为客户，最终服务于C端用户与B端商户



+ 容器平台：建设公司级容器平台，实现从技术探索到规模落地，从单一功能到全生态业务支撑



发展初期，技术探索

- 集群规模100+

底层稳定，规模落地

- 集群规模500+
- 稳定性建设：集群组件优化、部署标准化、资源合理化调控、自愈、准入

功能全面，用户价值

- 集群规模1100+
- 功能完善：固定IP、有状态存储、原地升级、一键上下线负载

能力增强，生态完善

- 集群规模1900+
- 能力增强：GPUShare、GlusterFS、VPA • 生态完善：DB、中间件容器化

混部创新，降本增效

- 300+ 离在线混部、250+ 在离线混部
- 400+ 储备资源池

+ 中间件平台化：实现平台化中间件服务，从无序到有序，实现资源统一化管理，高效资源调度把控

平台化前

各业务分别申请，**独立实例部署**

开发人员**自行安装部署**

无统一化利用率标注

开发**独立维护**

平台化后

资源统一调配，**多实例部署**

平台申请，**自动化部署交付**

有统一化利用率标准

运维**统一维护**

VS

业务摸底 运维接管 规范制定 迁移升级 **资源治理**

云PAAS平台建设

技术创新

平台服务规模

Codis: x+实例，缓存总量xT+

Redis: x+实例，缓存总量xT+

Es x+索引，存储总量xT+

Zk x+集群 节点总数xW+

Mq: x+队列，日消息量x亿+

kafka : x+队列，日消息量x亿+



ZooKeeper



+ 微服务：发布微服务平台ASF，集成服务注册和发现、服务治理、可观测、动态配置等能力



核心功能建设

- 支持 **spring cloud** 服务治理
- 支持**跨语言** (java、go)
- 支持**多协议** (dubbo、grpc、restful)
- 高可用**微服务网关**

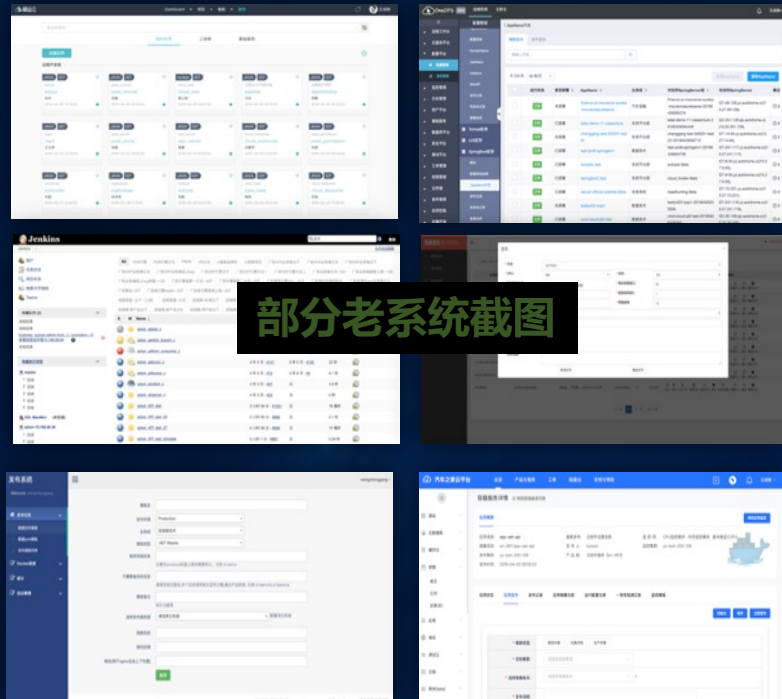
服务稳定性建设

- **无损上/下线**
- **go 语言服务黄金指标监控**
- **自适应负载均衡**
- **服务诊断**
- **多注册中心支持**
- 注册中心/配置中心**异地双活**

+ DEVOPS: 整合各BU自建平台系统, 推出一站式应用管理、CI/CD、监控服务、日志管理平台

系统各自为政、发布、监控、日志需要频繁跳转、体验差, 系统维护成本高

实现多云、多环境、多类型混合部署, 从注册、部署、运维到注销的应用全生命周期管理, 结合AutoCMP、AutoLog等产品, 实现DEVOPS



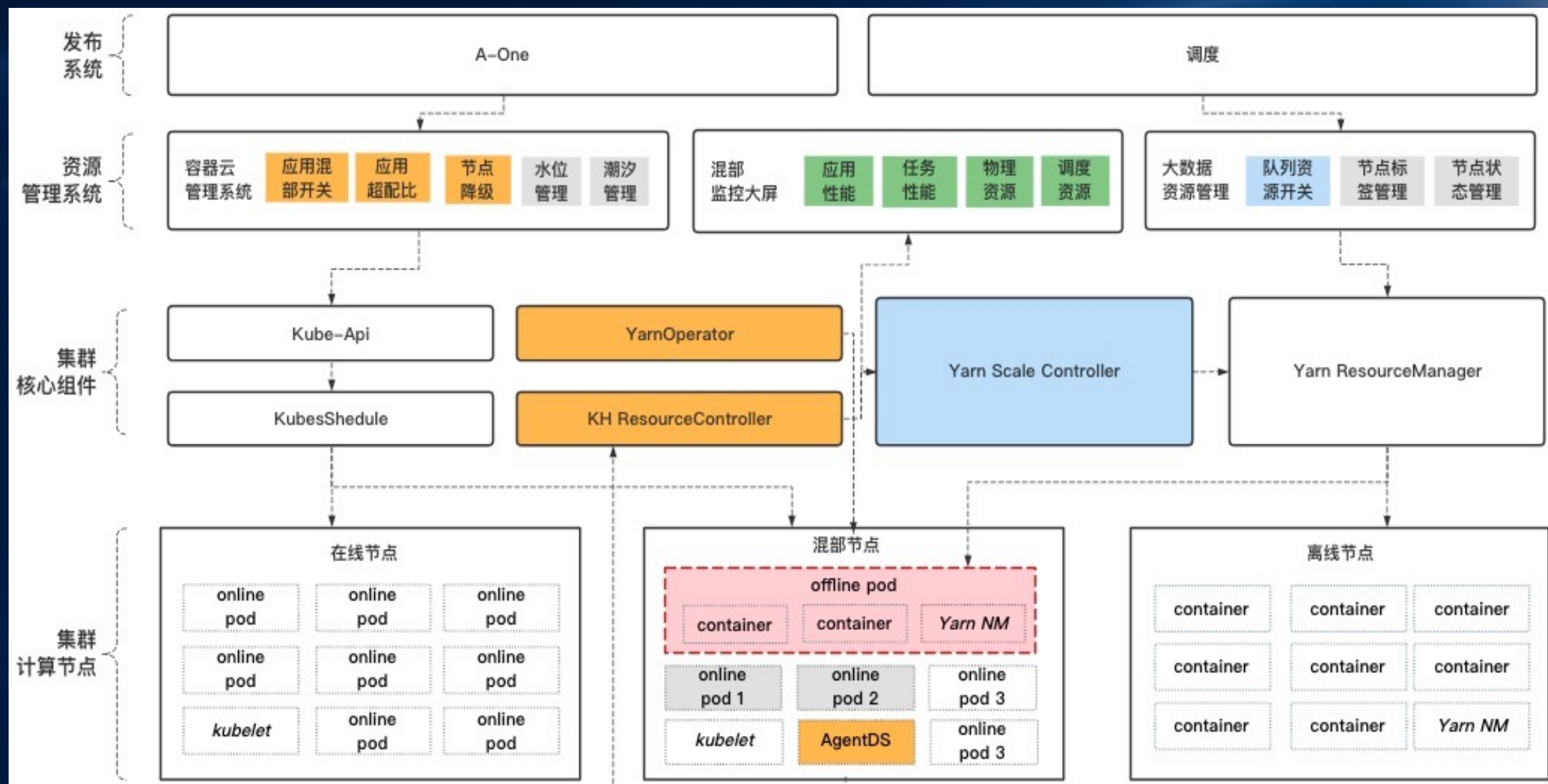
演进

聚焦降本增效

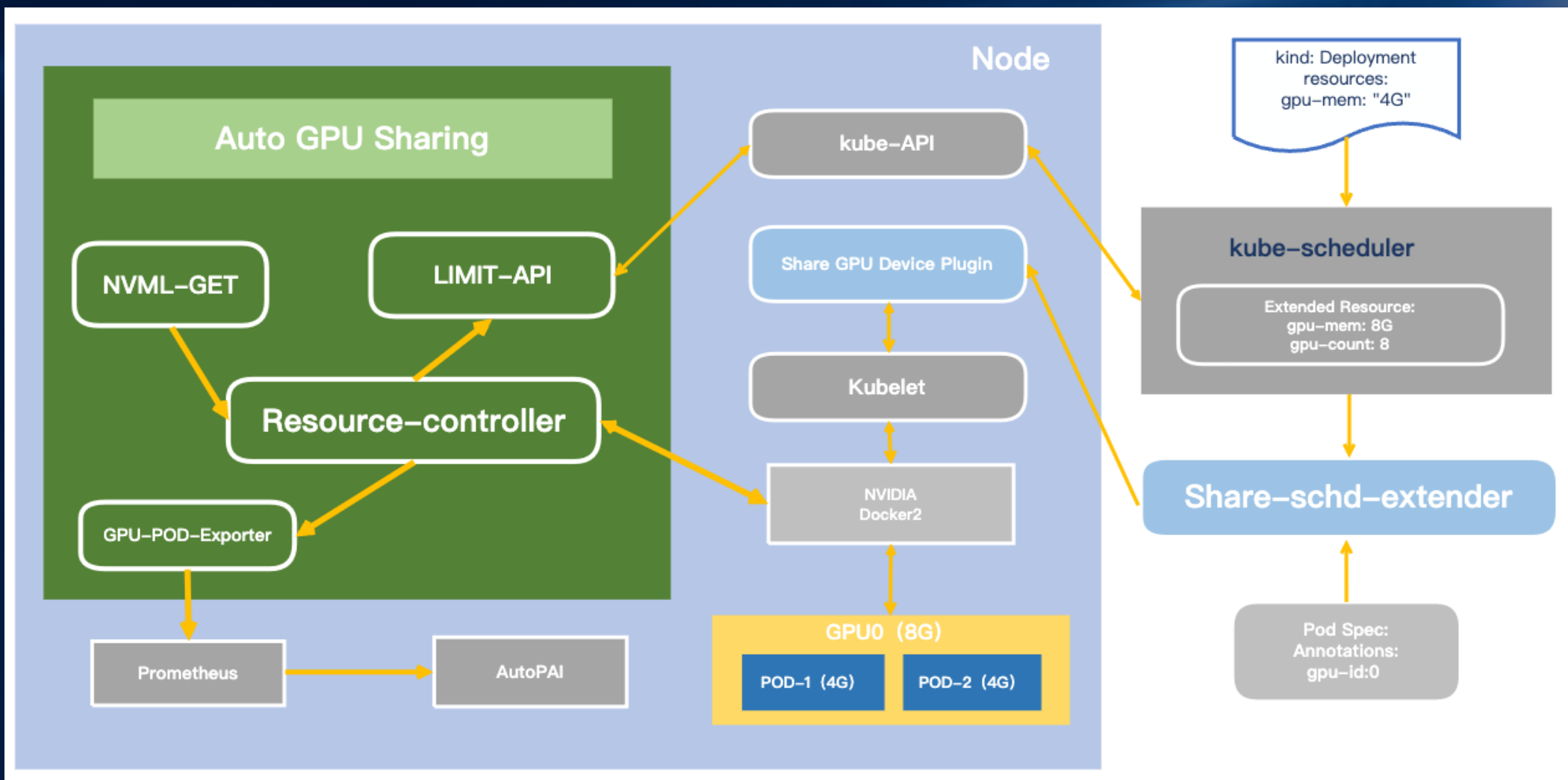
+ 通过管理方向、技术方向双重入手，盘活存量IT资源，以存量IT资源支撑汽车之家业务增长



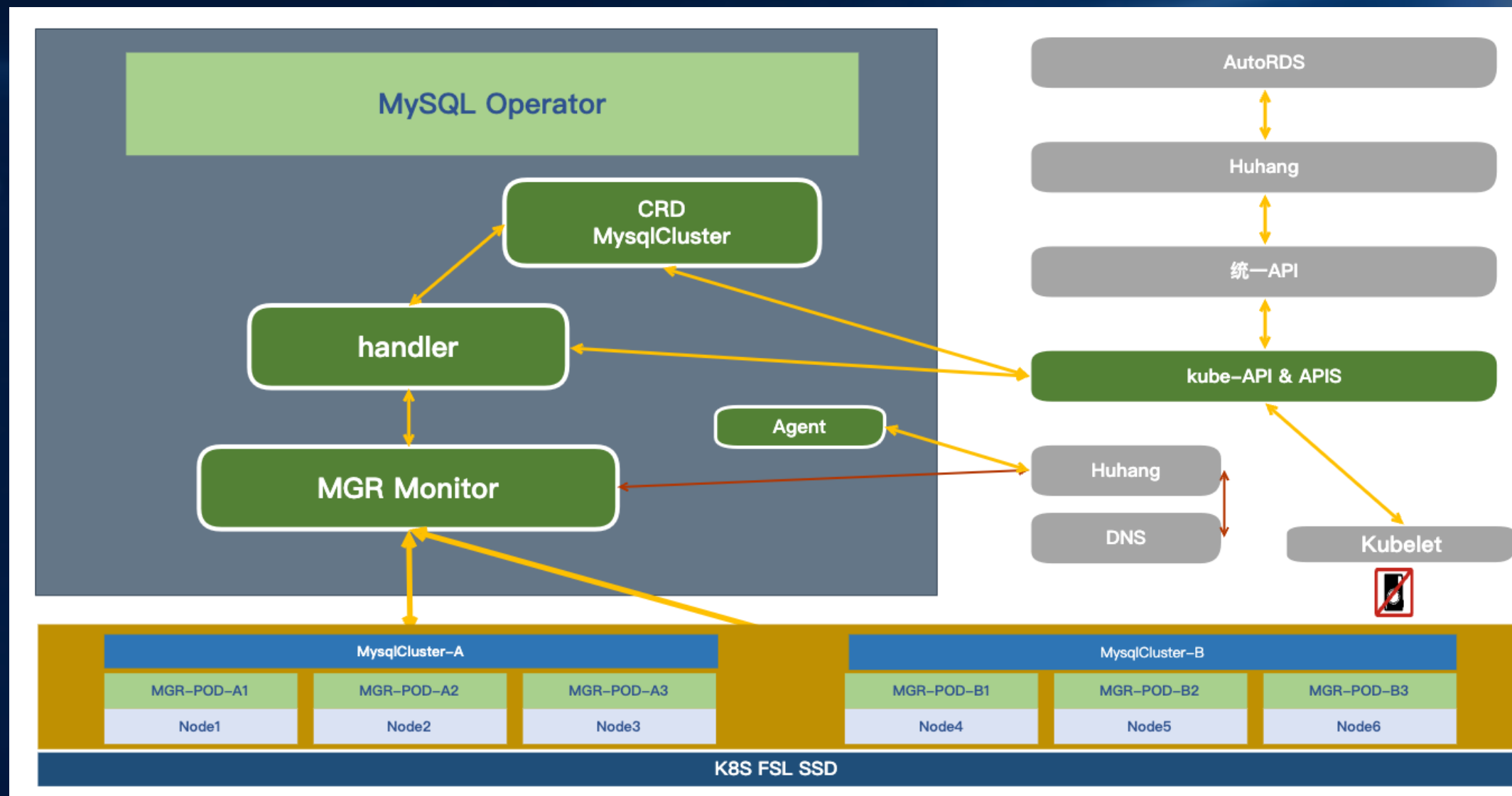
+ 大数据+容器化混部，落地大规模混部集群，利用在离线业务错峰冗余，实现资源复用



+ GPU资源提效，基于开源GPU Share技术，实现GPU卡以显存为单位分配、调度



+ 中间件容器化, 以DB Operator模式为先河, 实现一键编排、资源定义、故障自愈



+ 大数据冷数据治理，落地EC存储技术，应用精细化的数据治理策略，实现冷数据治理

冷数据转换服务

转换

校验

备份

记录

静默

通知

冷数据分析服务

根据白名单生成冷数据文件列表

冷数据文件列表写到消息队列

运维集市，运数据分析

转存储黑白名单
Include
exclude

数据存储想详情
访问时间、分类次数、策略..

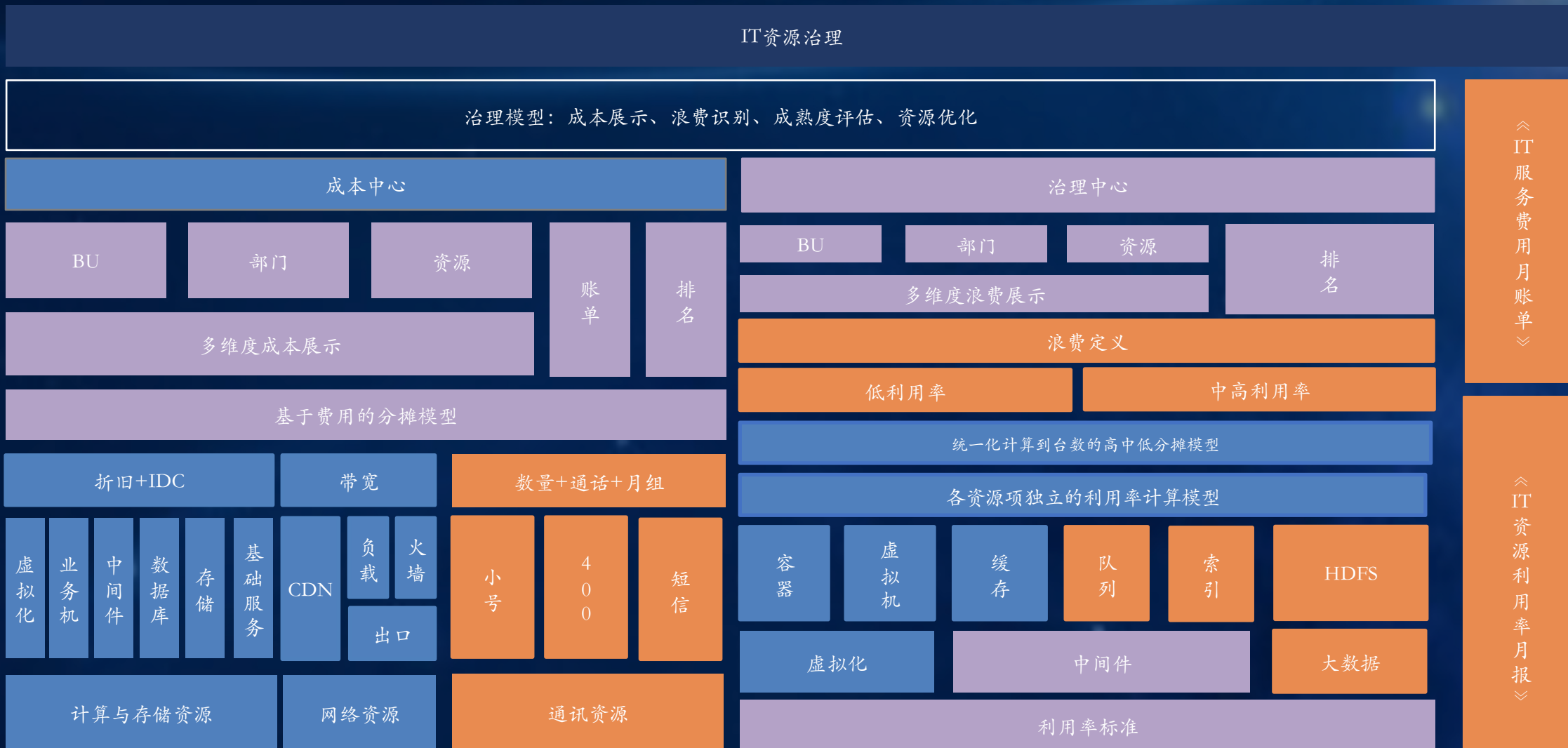
统计分析日报
历史压缩量
压缩列表

元数据解析

fsimage

autitlog

+ 践行FinOPS，落地多维度成本、利用率数据洞察，结合技术规范标准，形成资源优化闭环



+ 实施利用率判定，以主机低利用率为基础，结合部分细化服务的独立标准，实施利用率分摊



低利用率判定标准

| 主机-低利用率标准 | |
|--------------|------------|
| CPU使用率 < x% | 内存使用率 < x% |
| 磁盘空间使用率 < x% | 磁盘IO < x% |
| 连接数 < xk | 网卡带宽 < xGb |

| 虚拟机服务-低利用率标准 | |
|--------------|------------|
| CPU使用率 < x% | 内存使用率 < x% |
| 磁盘空间使用率 < x% | 磁盘IO < x% |
| 连接数 < xk | 网卡带宽 < xGb |

| 容器服务-低利用率标准 |
|-------------|
| CPU使用率 < x% |
| 内存使用率 < x% |
| 网卡流量 < xkb |

| 缓存服务-低利用率标准 |
|-------------|
| 容量使用率 ≤ x% |

- 独享资源，依据主机利用率判定标准、结合资源所属负责人，以台数分摊至BU、部门
- 共享资源，已制定服务利用率标准的资源，依据主机利用率计算该资源项各利用率池数量，结合独立标准计算利用率占比，分摊数量，未制定通用利用率标准的资源，利用率统一归属至云平台

+ 推进低利用率治理，针对不同业务场景，基于通用的治理手段基础、制定专项操作方案

| 分类 | 资源类别 | 独享资源 | 共享资源 | 手段 |
|-----|---------------|------|------|-----------|
| 物理机 | 业务机 | √ | - | 1、2、3、4、5 |
| 虚拟化 | 虚拟机 | √ | - | 1、2、3、4、5 |
| | 容器 | - | √ | 1 |
| 中间件 | Redis | √ | √ | 1、2、3、4 |
| | Elasticsearch | √ | √ | 1、2、3、4 |
| | Kafka | √ | √ | 1、3、4 |
| | RabbitMQ | √ | √ | 1、2、3、4 |
| | Zookeeper | √ | √ | 1、3、4 |
| 大数据 | hadoop | - | √ | 1、4 |
| | Hbase | - | √ | 1、2、3 |
| | Flink | - | √ | 1、2 |
| 数据库 | MySQL | √ | √ | 1、2、4 |
| | SqlServer | √ | √ | 1 |
| | MongoDB | √ | √ | 1、3 |
| | TiDB | √ | √ | 1 |
| | StartRocks | √ | √ | 1 |
| 存储 | FastDFS | - | √ | 1 |
| | Ceph | - | √ | 1 |

手段1：无效应用下线

✓ 梳理无效的业务，及时对无效实例进行腾退

手段2：服务容器化部署

✓ 容器化部署可以有效节省资源，提高资源利用率

手段3：独享资源合并部署

✓ 合并部署，提高单击部署密度，及时下线冗余的物理机

手段4：资源规格缩容量

✓ 治理低利用率实例，缩容腾挪冗余资源

手段5：离在线混部

✓ 支持潮汐调度，提高离线闲时资源的利用率

+ 执行浪费治理，借鉴低利用率治理成功经验，确定浪费标准，将利用率治理全面升级为浪费治理

目标：浪费资源 = 0

| | |
|-------------------|-------------------------------------------------------------|
| 低利用率资源浪费 | 中高利用率资源浪费 |
| 低利用率占比 $\leq 5\%$ | 冷数据占比：ES $\leq 55\%$ 、Kafka $\leq 30\%$ 、Hadoop $\leq 50\%$ |



中高利用率浪费的治理手段

Hadoop

- ✓ 表生命周期标记和过期数据删除
- ✓ 废弃表确认删除
- ✓ 冷数据转EC存储
- ✓ 无效任务下线
- ✓ 资源申请不合理的任务整改

ES

- ✓ 删除无流量的闲置索引
- ✓ 减少索引保留时间
- ✓ 减少分片数，提升单分片大小；合并同类索引；调大索引分割周期

Kafka

- ✓ 删除消费流量为 0 的 topic



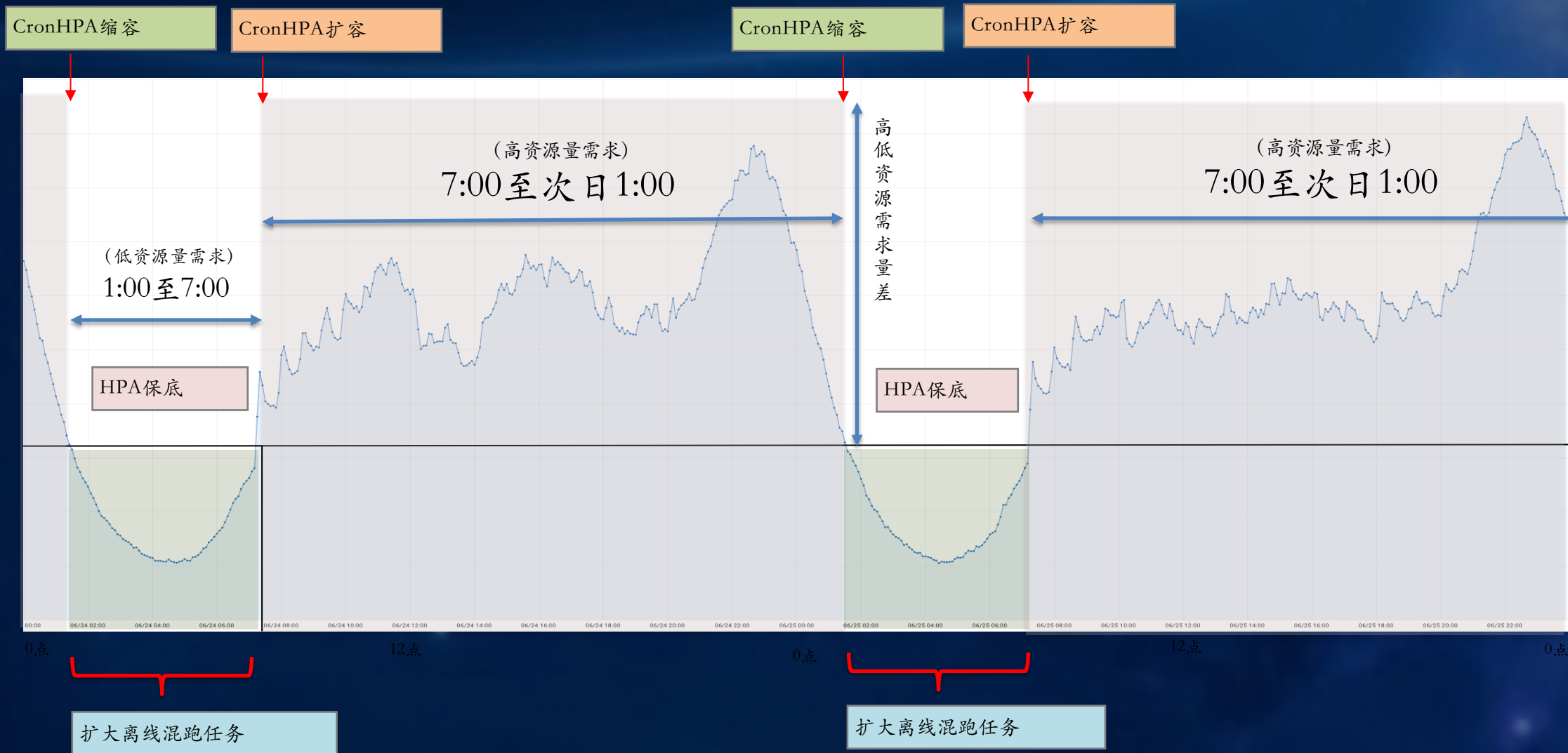
未来

挑 战 与 展 望

+ 全量在离线混部、秒级的混部伸缩感知，同时实现100%储备主机计算资源利用



+ HPA与在离线混部架构深度融合，落地适合于之家存量资源现状下的降本增效模式

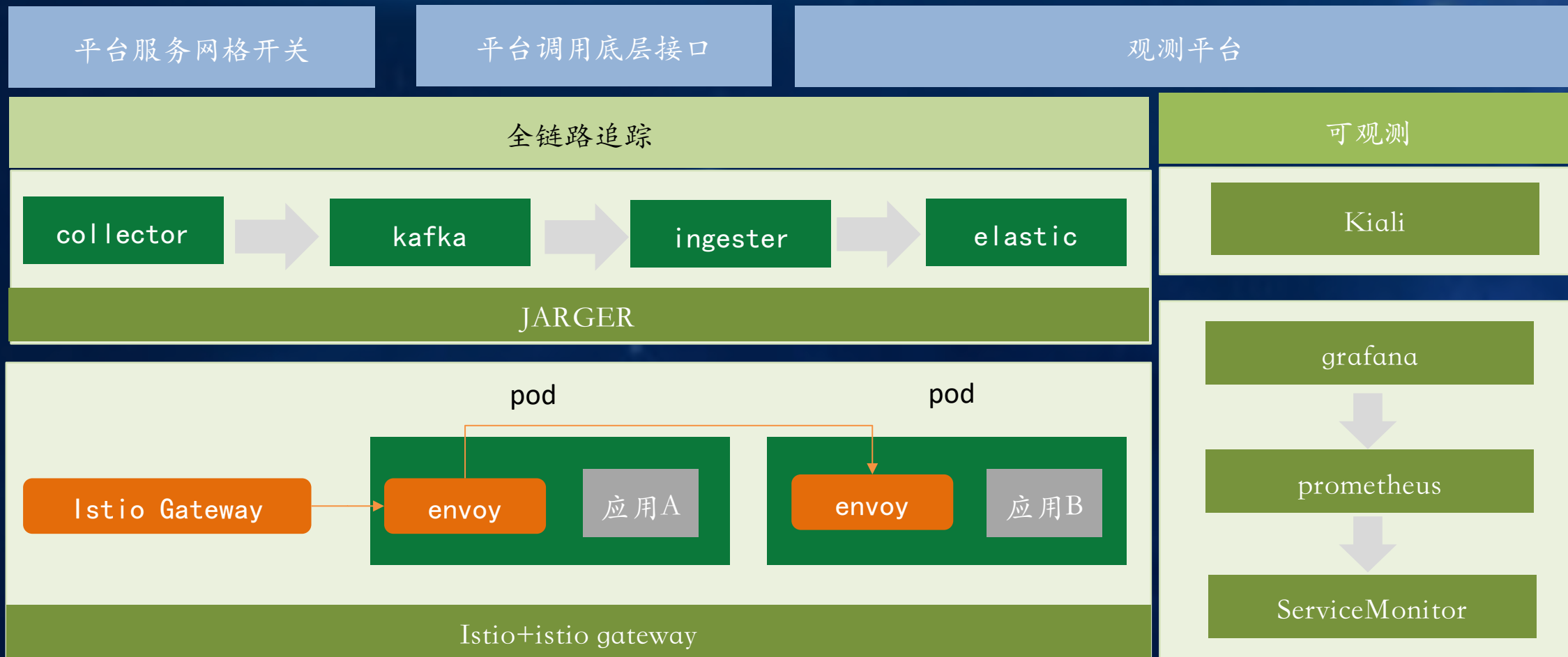


+ 提供资源层统一、开放各技术工具服务，各类技术公共能力集中聚合，实现系统复用



+ 建设可承载日30亿请求的超大规模服务网格基础建设，支撑C端业务大规模上量

AONE 平台勾选服务网格选项



THANKS

共创 共建 架构 未来