

源于社区 服务社区

 中国DevOps社区峰会 2023 · 广州



智造产线边缘的云原生规模实践

赵浩廷 红帽软件 架构师





赵浩廷

红帽软件 制造行业 解决方案

- 在IBM 和 Red Hat 工作，参与过售前和交付工作
- 中间件 云原生 DevOps
- 和制造业有不解之缘：整装型为主，也参与过一些炼化类
- 从早年的工厂流程信息化、生产质量管控到数字化产线





目录

- 1 边缘侧云原生的业务驱动力
- 2 资源受限下的架构设计挑战
- 3 开源设计模式参考
- 4 落地业务场景：MES、PLC和管理
- 5 Q&A



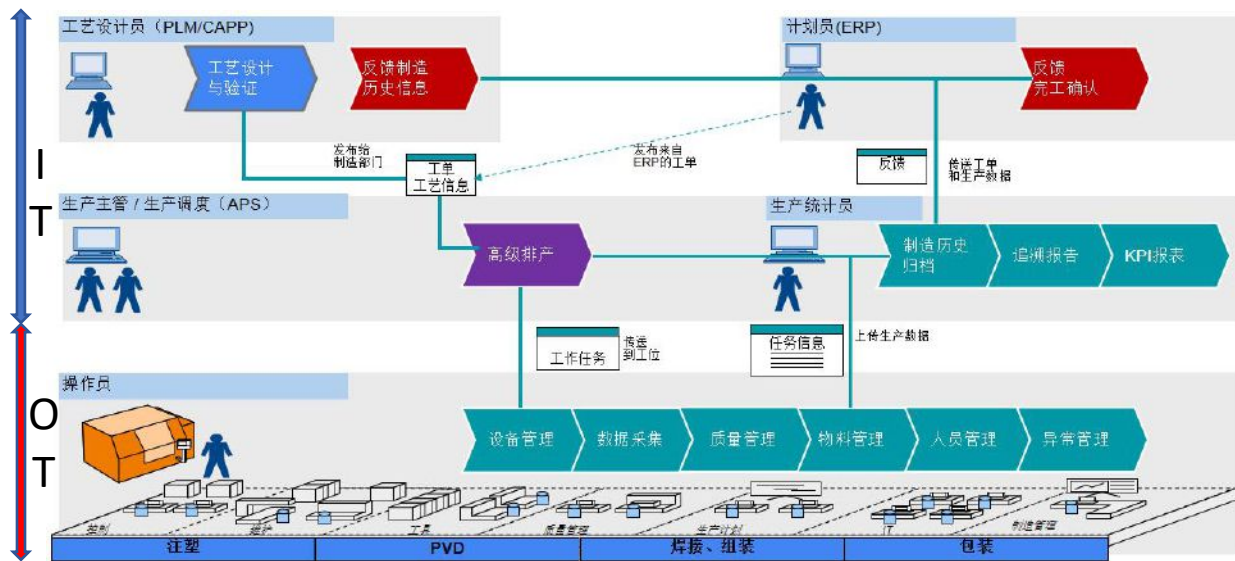


边缘侧云原生的业务驱动力

智能制造关注的趋势引领IT/OT融合

	安全	质量	响应	效率	环保
智能工厂	生产环境安全 <ul style="list-style-type: none"> 环境风险自动识别 人员伤害自动预防 危害时间主动响应 	交付质量保障 <ul style="list-style-type: none"> 系统质量风险消除 确保质量可追溯性 制造缺陷自动探测 	快速订单响应 <ul style="list-style-type: none"> 生产系统直连市场 生产资源快速调配 产品交期精准预测 	生产成本优化 <ul style="list-style-type: none"> 最优的生产计划 生产资源高效利用 通过大数据分析不断改进 	环境友好型制造 <ul style="list-style-type: none"> 清洁能源利用 环保的制造工艺 零危废排放
智能生产		生产工艺稳定 <ul style="list-style-type: none"> 工艺全过程监控 工艺参数偏离预警 生产缺陷自动识别 	灵活定制生产 <ul style="list-style-type: none"> 设备自动识别订单 灵活定制化加工工艺参数灵活调整 		
智能物流	安全供应链 <ul style="list-style-type: none"> 供应节点状态监控 物流实时监控 供应链风险预警 	敏捷供应链 <ul style="list-style-type: none"> 客户需求快速响应 货物送达时间精准 大数据分析实现需求提前预测 		高效绿色供应链 <ul style="list-style-type: none"> 适当库存水平 合理的库存布局 运输成本最优化 	

典型工业制造场景（整装）：



- ✓ 生产了多少产品（那种产品），有多少在制品（在制品状态）
- ✓ 产品的质量如何（多少次品），出现质量的产品在哪里
- ✓ 消耗了多少资源（原材料，包材，人工时）
- ✓ 产品和批次能否追踪追溯（BOM，物料，操作，设备参数）
- ✓ 工厂的生产效率是多少，生产成本（创造价值）多少
- ✓ 如果改变生产计划会有什么样的影响



制造业中的 IT 和 OT 系统必须服务于两个目的：数据和分析以及控制

数据与分析

数据的可用性与兼容性

跨集群, 系统, 和物理机器

集中报告和信息汇总

跨系统查询数据以获取端到端报告

监控、响应和警报

跨生产线的自动化决策和响应

读

控制技术

仿真和虚拟调试

跨越产线和厂区

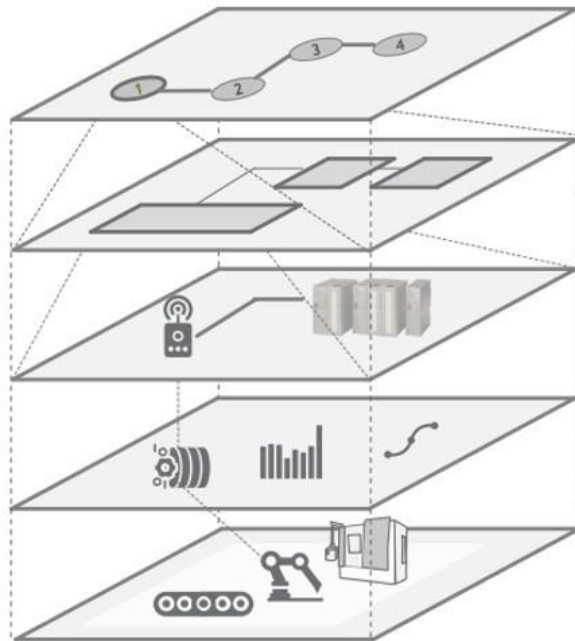
控制和配置

从中央系统控制每个流程或机器

灵活添加新系统/传感器

不改变基本配置

读/写



流程

直接与间接

组织

传送线, 柔性单元, ...

控制

PLC, RC, NC
云/边缘

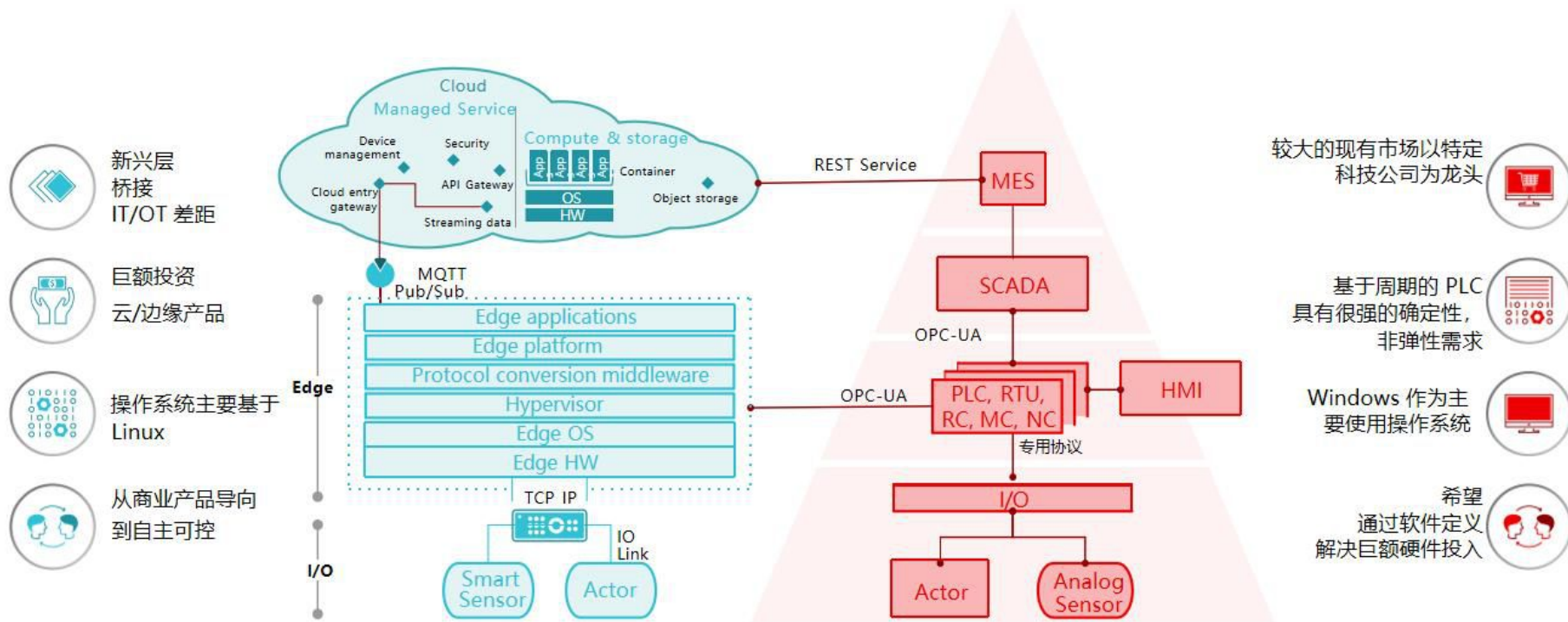
数据

时间序列, 图象, 视频, 地理信息, 二进制

资产

设备, 机器人, AGV, 仓库, 传送带

IT和OT在业务成本压力下渐渐出现整合与融合



制造自动化的架构从硬件定义转向软件定义

数据与分析的趋势

边缘计算

将弥合 IT/OT 差距
并为生产车间释放
云的力量

综合数据产出

用于构建/模拟
非结构化环境训练AI
机器人系统

联邦数据

将取代数据湖：
数据合约，
-本体 和 web3

自动化控制技术的发展趋势

MES

- 将 MES 功能转移并迁移到公共/私有云环境中。WIN 实施将被现代应用技术取代；适用于移动应用程序、平板电脑、PC、VR 眼镜的瘦 MES 客户端
- 重用 DigitalTwin 仿真模型以实现过程可视化目的

SCADA

- SCADA的数据采集功能将被边缘/云数据和分析所取代
- 基于 WIN 的单一监控功能将被 Web 技术和面向服务的架构取代

HMI

- HMI 应用程序（例如 WinCC）将被 Web 技术（例如 Angular、react、vue 框架）取代。因此，HMI-IPC 可以由具有开放性的操作系统（例如 Linux）的经济高效硬件提供支持

PLC,
NC, MC

- 在定制印刷电路板 (PCB) 中设计的 PLC 功能将被使用 RT 操作系统在 COTS（商业通用）硬件上运行的软 PLC 所取代。容器技术将有助于跨 COTS 硬件分配 PLC 功能 + 相邻数据分析功能（例如计算机视觉）

Robot
Control

- 机器人变成了认知自动化。系统解决的工作流程和任务对于传统自动化来说过于复杂和通用。这需要结合 DeepL 技术来理解世界的复杂模型以进行精准预测。将需要高性能硬件和最先进的软件模式

Sensors &
actuators

- 模拟传感器被轻量级嵌入式智能传感器替代；智能现场设备被轻量操作系统取代 - 决策将更加靠近数据源 - 传感器能够使用机器学习推理模型收集、推断数据并采取行动



Why NOW?

技术成熟度满足行业趋势

技术进步与成熟



Linux 实时内核的成熟度

- ✓ 实时核心广泛应用8年多
- ✓ 用于实时核心的芯片组和网络组件
- ✓ 比 Windows 更好的性能、稳健性和稳定性



工业控制的云原生增强

- ✓ 基于k8s容器的优化
- ✓ 用于改造工业控制系统的完整技术堆栈 (DevOps、数据处理、监控、微服务相关)



开放OT通信协议

- ✓ OCP-UA 成为标准 (IEC 62541 实现)
- ✓ 在通用 IT 平台上实施 PROFINET 和 EtherCat

制造行业趋势



- 技术上受自动化巨头的约束
- 巨大的成本导致技术束缚不可避免地被打破
- 打破技术束缚->降低生产成本->更高兼容性





资源受限下的架构设计挑战



边缘云原生架构设计的需求和约束

- 对**成本**高度敏感（“按成本设计”）
- **无人工厂**（“熄灯操作”）
 - 维护意味着安排现场作业由非专业人员更换
 - 节点故障恢复能力（自我修复；继续以降低的容量运行；或者上级备份支撑点）
 - 对集群重启的恢复能力（例如从站点范围的电源故障中正常恢复）
- **边缘规模**
 - 对具有 10 种不同配置的 10 万个站点进行变更管理
 - 高度可并行化：集中式策略/管理，但完全本地执行
- 运维人员无法水平扩展，对**自动化**提出的需求
 - 能够以完全可重复的方式远程操作
 - 升级软件和配置、扩展、备份/恢复、重建整个堆栈，无需 CLI
- 空间、功率、热设计、电尘约束站点





结合实际场景所定义的前置条件

- 将故障域限制为单个站点/设备
- 容忍单个设备故障和断电（→ 单个集群重新启动）
- 容忍不完善的网络上行连接
- 无共享架构（shared-nothing architecture）：平台级别没有分发/拉伸/联合





结合实际场景所定义的前置条件

对**成本**极为敏感（必须保持每个站点的修复成本/开销很小）。
通常也受到严重的（空间、灰尘、静电干扰、电力、冷却.....）**资源限制**。

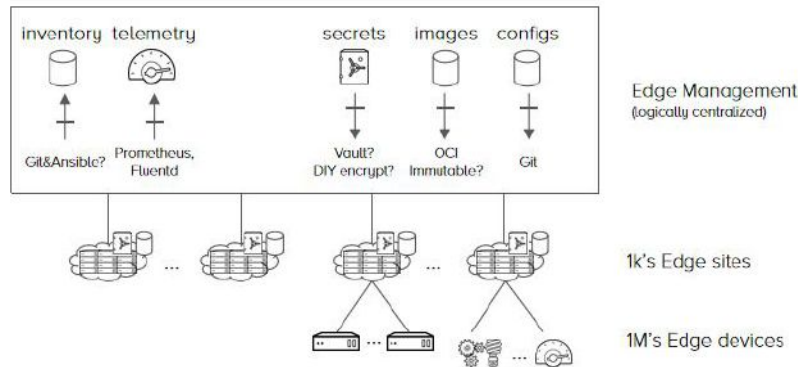
- 每个站点/设备的**系统冗余度** <10%，
- 1 个节点非 HA /3 个节点 HA 的**最小空间占用**
- 根据需求增加容量的能力（节点**纵向扩容**，基于基线配置）
- **空间小** -每个站点没有备用/空服务器或者只有1-2 个工控机容量余量，
- 能够分配多个角色/将**角色重新分配**给节点而无需重新配置，





如何集中管理数百万个站点/设备?

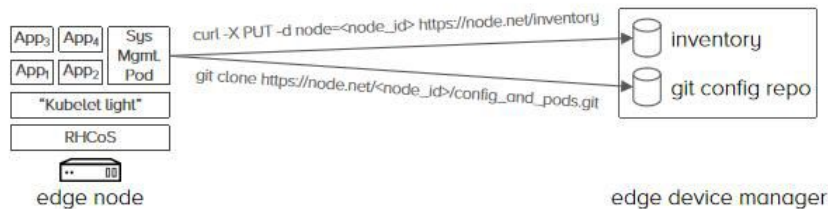
- 站点/设备根据集中管理的策略/配置自主实施部署与配置。
- 无需通过 WAN 进行配置 (PXE、Ansible playbooks、Ignition等)
- 站点/设备提取并缓存必要的工件
- 缓存可能已准备好/预填充
- 单一库存配置 (类CMDB)
- 单层遥测 (日志?)



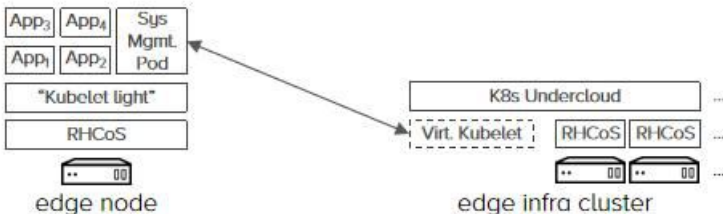
如何让设备加入更“自治”？

架构设计考虑:

- 独立节点，在“Kubelet light”中作为静态 Pod 或者守护集运行工作负载。
- 运行系统管理 Pod，用于注册带有设备注册表的节点、轮询 Pod 和配置映射的配置存储库、触发操作系统和系统级别组件更新等



或者



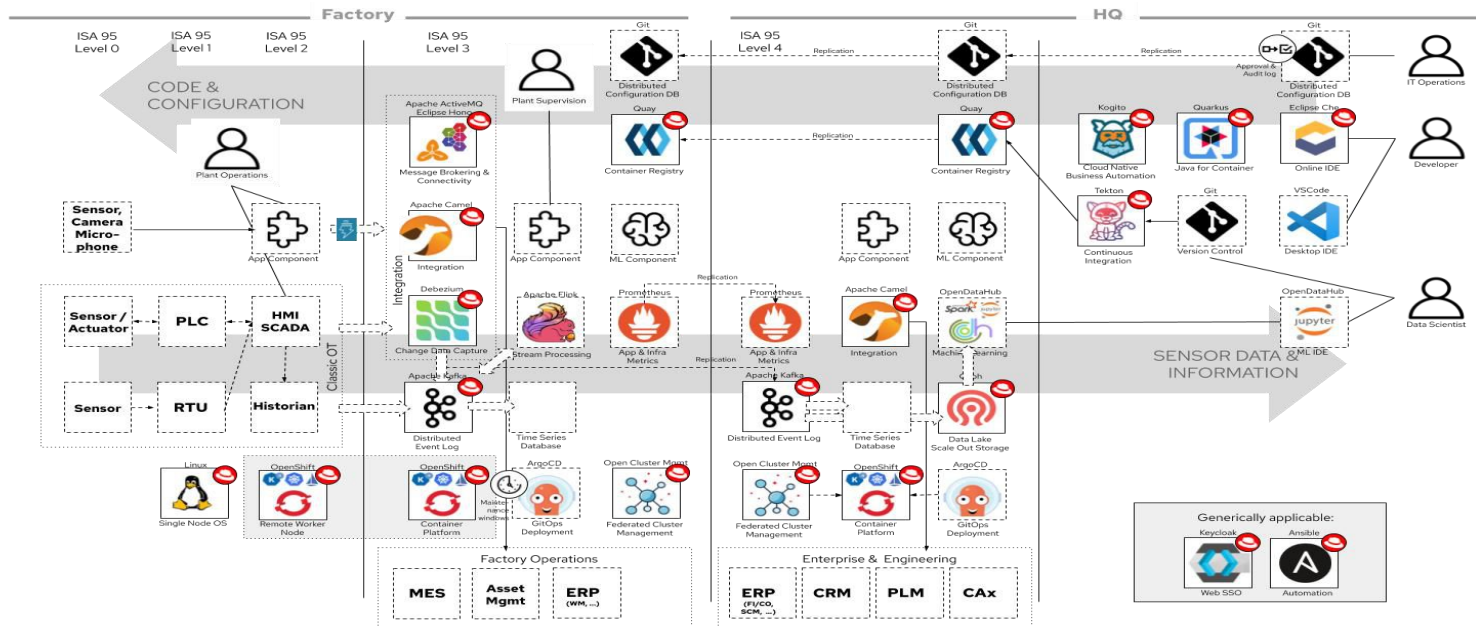
<https://github.com/virtual-kubelet/virtual-kubelet>



开源设计模式参考



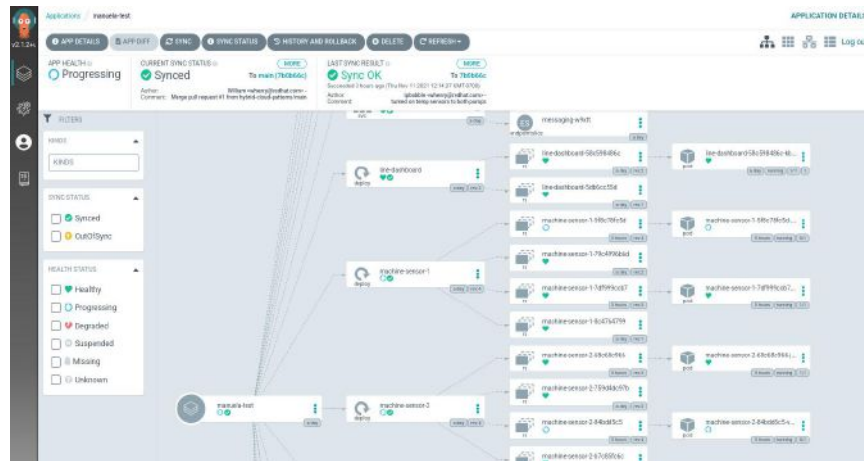
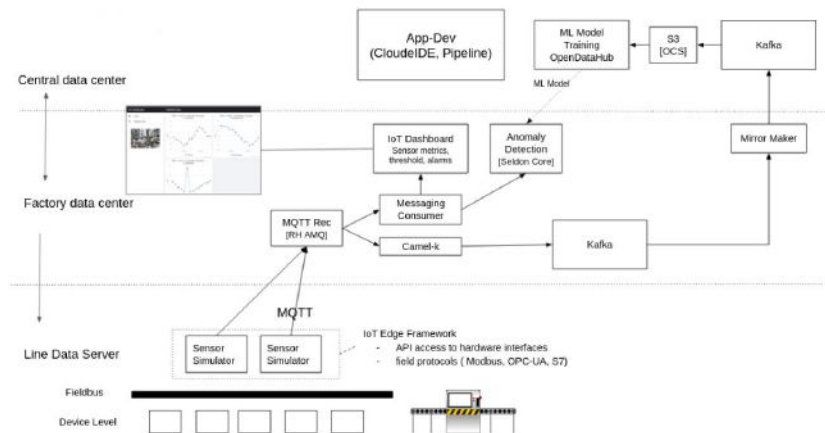
工业边缘开源参考架构: Manuela



<https://github.com/sa-mw-dach/manuela>



使用GitOps 规模化部署的模式



<https://github.com/validatedpatterns/industrial-edge>





GitOps 的落地模式参考

1. 满足多集群的应用发布要求；
2. 满足多环境以及多区域的应用发布要求；
3. 采用Kustomize的满足集中维护公共配置，以及兼顾维护特有配置的要求；
4. 修改公共配置时 **只需要在一个地方修改**；
5. 版本升级修改镜像标签时 **只需要在一个地方修改**；
6. 满足对单个集群粒度的应用生命周期管理；
7. 满足对单个集群粒度的集群特有配置修改；（通常是一次性的初始化）

Example: <https://github.com/xdevops-caj-acm/parksmmap/tree/main/k8s-manifests>

Kustomize: <https://github.com/kubernetes-sigs/kustomize>

Kustomize document: <https://kubernetes.io/docs/tasks/manage-kubernetes-objects/kustomization/>

Kustomize document: <https://kubectl.docs.kubernetes.io/references/kustomize/kustomization/>

Google Cloud | Kustomize: <https://cloud.google.com/anthos-config-management/docs/concepts/kustomize>





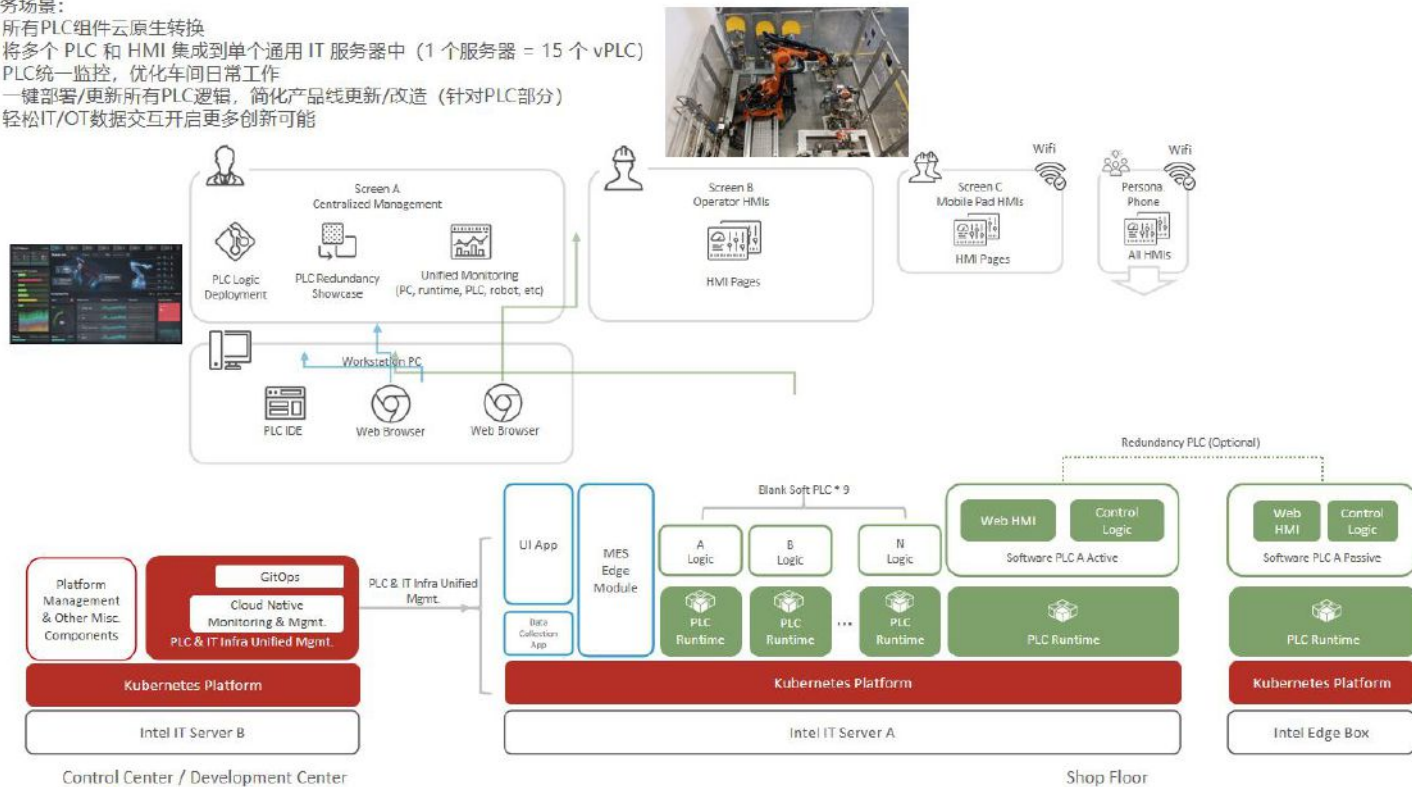
落地业务场景： MES、PLC和管理



某一线欧美车企生产工艺组云原生转型场景

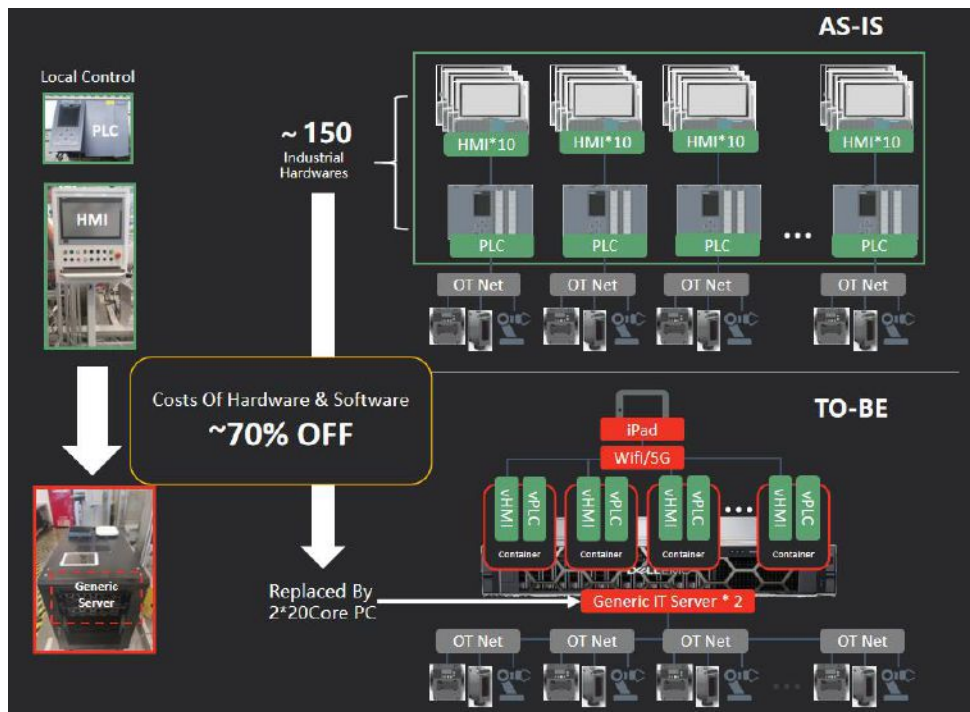
业务场景:

- 所有PLC组件云原生转换
- 将多个PLC和HMI集成到单个通用IT服务器中 (1个服务器 = 15个vPLC)
- PLC统一监控, 优化车间日常工作
- 一键部署/更新所有PLC逻辑, 简化产品线更新/改造 (针对PLC部分)
- 轻松IT/OT数据交互开启更多创新可能

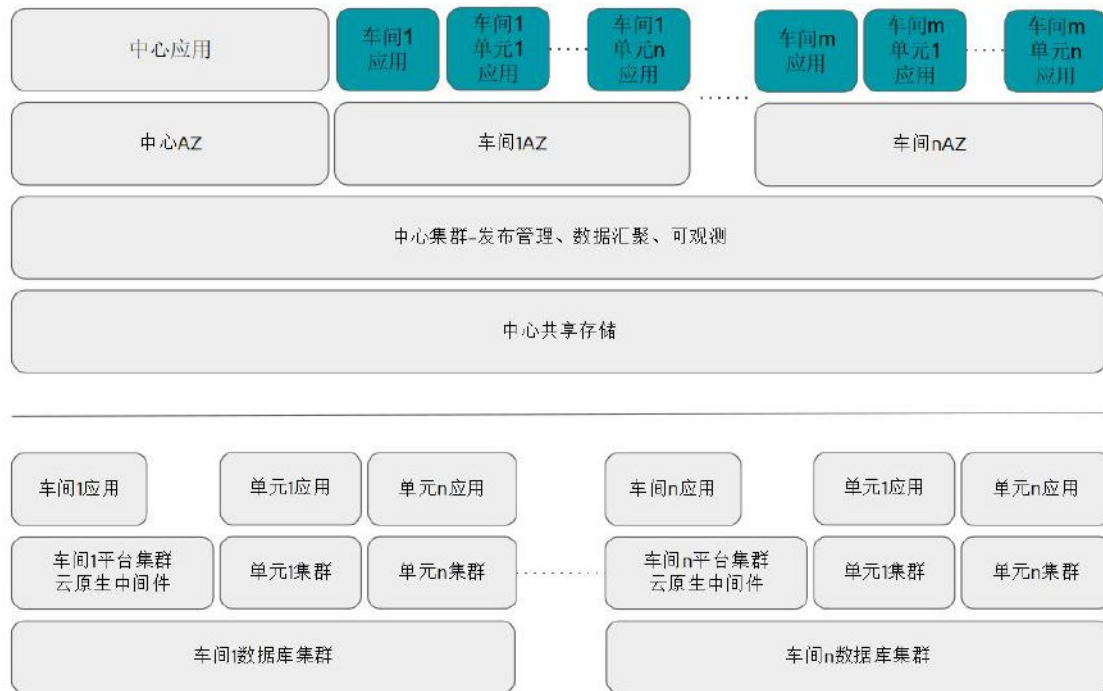


生产自动化控制的成本效益（续前）

- 使用通用的IT 硬件替代
- 备件（千万级甚至过亿）成本降低
- 除部分功能安全要求场景外
- 业务逻辑可按需定制并通过CI/CD实现快速发布
- 实现了IT/OT在边缘工艺的交互



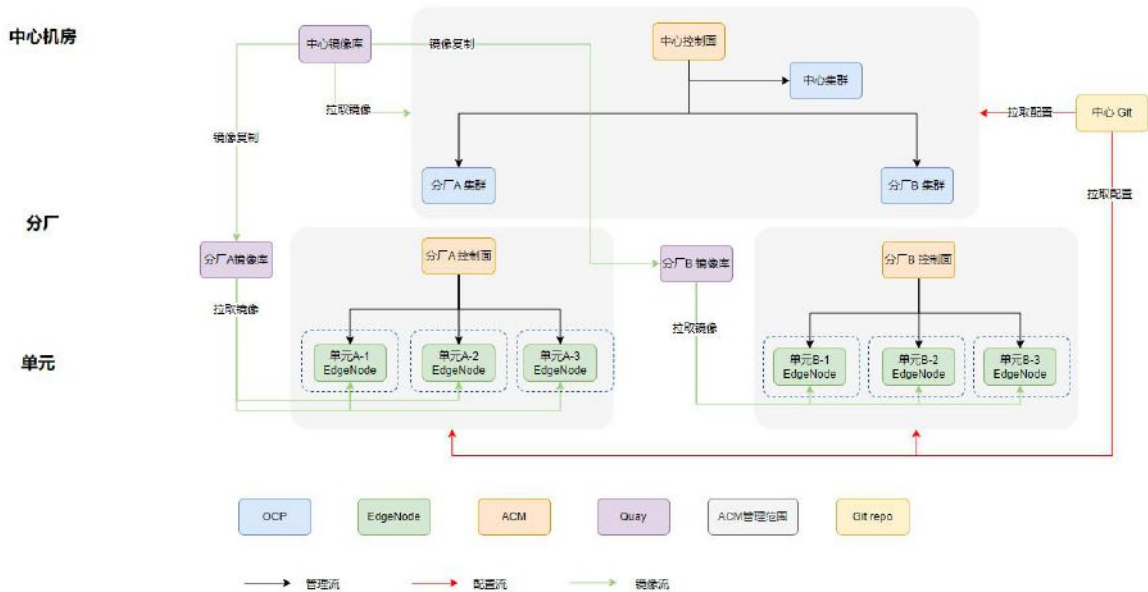
某龙头家电制造企业应用逻辑架构（转型过渡架构）



- 应用是“无状态”的对待
- 状态由业务逻辑保持
- 利用数据中心的资源进行冗余设计
- 单元业务“自治”
- 通过数据库+消息流完成业务
- 管理与业务操作分离
- 微服务设计，按领域单一镜像打包，约12+业务领域
- 车间级的数据可能丢失！

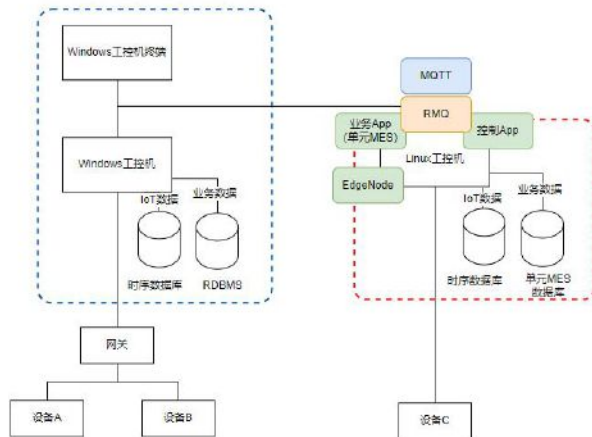
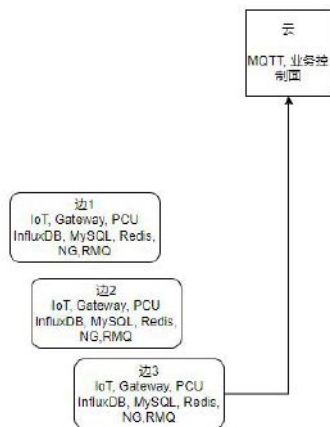
应用发布管理（续前）

- 使用GitOps支撑规模化部署（CD）
- 中心管理原则，厂区基层维护人员
- 应用配置差异使用Kustomize实现导入，如ConfigMap
- 运维团队人员2+1 支持100+的工艺单元



某外资500强日化企业 使用GitOps 规模化自动部署IoT应用

- 采用了标准的Kustomize做法
- 按业务线进行划分
- ArgoCD CR也是GitOps的资源之一
- 云/边端分开管理，本地安全开关
- 中间件依赖 Helm 版本化

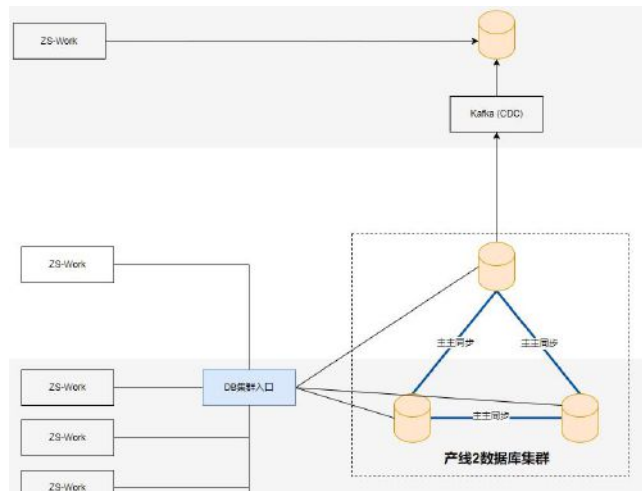
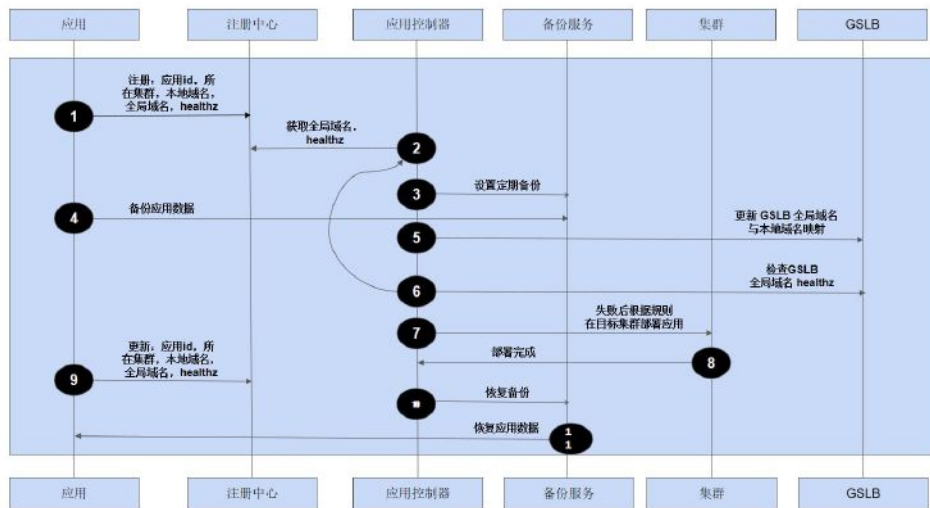


GitOps Repo Template	
base	
-- IoTApp	
-- GateWayApp	
middleware	
-- mysql	
-- redis	
-- emqx	
-- rmq	
overlays	
-- regionA	
-- line1	
-- line2	
-- regionB	
-- line1	
-- line2	

高可用—伪命题?

看上去矛盾的需求:

- 数据可丢弃
- 离线自治生产要求
- 业务数据高可用





工程实践的收获和感想

- IT和OT永远说两套语言
- 应用现代化技术或者说云的技术是IT发起的变革
- 成本和资源在工业自动化是永恒的话题：备件成本、维护成本
- 近边端的MES应用仍然是IT应用，对接设备的协转已经对IT很友好
- 部分老旧产线的改造需要时间和更多架构的妥协
- 制造业在开放、自主控制 and 创新的道路上越走越远



转型是一个只有开始的“旅程”



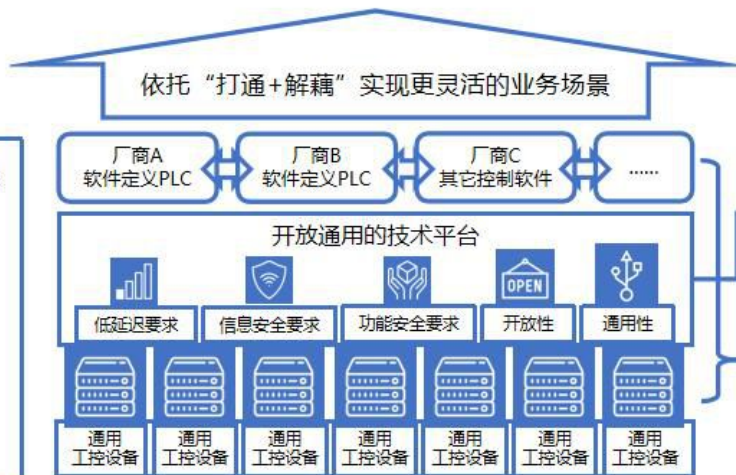
转型价值

✓ 大大增加控制域的设计、部署、使用、维护灵活性

- 简化控制域设计及部署
- 更低成本实现控制域一体化升级等维护动作
- 简化底层通用硬件的维修
- 增加控制设备的使用灵活性

✓ 为更丰富的应用场景提供基础

✓



转型难点



符合工控领域特殊要求且开放通用的平台是技术核心



开放一体化的应用及设备生态是成败关键



源于社区 服务社区

THANKS!

