

开放数字化架构平台（ODA Canvas）研究与设计

任翔¹ 冯学军¹ 李滨¹ 孙学志¹ 杨强¹
REN Xiang FENG Xuejun LI Bin SUN Xuezhi YANG Qiang

摘要

随着信息技术的迅猛发展和数字化转型的深入推进，电信行业面临业务复杂性增加、数据量激增及安全挑战，对于灵活、高效、安全的数字化架构平台的需求日益迫切。深入研究了开放数字化架构平台（ODA Canvas）的构建与设计，详细阐述了 ODA Canvas 平台的设计目标，进一步分析了 ODA Canvas 平台的关键技术，包括基于事件的 ODA 组件自动化开通技术、分布式全系统追踪与性能分析技术、面向 API 的异构系统混合集成技术。平台的构建让模块化、可重用的云原生组件触手可及，不仅极大地促进了不同供应商和运营商之间的协作，更推动了电信业务模式的创新和运营效率的提升。

关键词

数字化转型；开放数字架构（ODA）；云原生技术；分布式全系统追踪；能力开放

doi: 10.3969/j.issn.1672-9528.2024.10.007

0 引言

随着数字化进程的不断推进，云计算、大数据、AI、边缘计算和算力网络等新兴数字技术不断涌现，在数字化转型的驱动下，电信领域业务复杂度急剧上升，应用与数据量激增，云和工作负载日益复杂，安全问题也愈发严峻。这些变化为电信运营商带来了一系列挑战：新业务上线周期长、应用跨域集成难度大、自动化智能化水平不足等。因此，建设统一数字化底座，为用户提供开放共享平台化能力、敏捷化的业务支撑、智能化的运维支撑的需求愈加迫切。

电信管理论坛（TM forum，简称 TMF）是电信领域重要的国际标准组织，它在 2020 年提出了开放数字化架构（open digital architecture，简称 ODA），旨在为电信运营商和供应商构建一个多元化和充满活力的合作伙伴生态^[1]。ODA 项目自推出以来，获得了全球多家通信服务提供商（CSPs）和技术合作伙伴的广泛支持。至今 TMF 已发布了 88 个 OpenAPI 和 59 个业务组件规范，覆盖了运营商 B 域和 O 域软件系统，ODA 已成为电信行业数字化转型的重要推动力。

1 平台设计

为加速 ODA 的落地实现，实现组件的可插拔运行和统一的运维管理，设计并实现了一套敏捷、云原生的数字化底座——开放数字化架构平台（ODA Canvas），以确保不同厂商之间的产品能够在统一的平台中实现无缝集成和协作。

开放数字化架构平台（ODA Canvas）作为一个执行环境，依托 Kubernetes 平台，基于事件和 ODA 组件数据模型实现

零配置自动化运维能力，同时平台为 ODA 组件提供了一系列云原生服务^[2]，方便更高效的业务运营和故障管理。

（1）API 网关 / 服务网格：以标准化方式暴露 API，确保服务的可访问性和安全性。

（2）可观测性服务：监控和管理组件，保障系统的透明度和可控性。

（3）CI/CD 流水线：提供灵活的持续集成和持续部署流程，提高发布效率。

（4）EFK 日志服务：建立统一的日志收集和查询体系，加强了日志管理的集中化。

（5）身份管理系统：注册组件并授予用户相应的权限，增强了系统的安全性。

1.1 总体架构

开放数字化架构平台（ODA Canvas）是基于 IT 思路建立的一个可编排、云原生和零接触运维的数字化系统。该平台通过标准化的组件和对接的 Open API，实现组件可插拔，系统可组装。作为研发运维一体化、标准化平台，它具备组件集中管理、快速部署、服务集中监控、在线管控、DevOps 集成及多项目多租户管理等能力^[3]，平台功能架构图如图 1 所示。

（1）组件部署与生命周期管理：通过自动化的方式，简化组件的部署、升级和删除流程。通过容器化部署和微服务架构，实现应用的快速上线和全生命周期管理。

（2）资源优化与自动化：平台提供云资源的智能调度和管理功能，可根据应用的负载情况调整资源，提高资源利用率。同时，通过自动化管理，减少人工干预，降低运维成本。

1. 浪潮通信信息系统有限公司 山东济南 250101

(3) 安全性和隐私保护：平台提供身份验证、访问控制、数据加密等功能，确保应用和数据的安全，保护隐私。

(4) 组件集成与协同：云原生管理平台能整合各种服务和组件，实现组件的快速集成和协同工作。

(5) 统一管理与监控：平台提供统一的管理界面和监控工具，方便用户对整个云环境进行管理和监控。

(6) 快速故障定界定位：通过分布式日志、告警监控等功能，实时汇总和在线分析，实现故障的自动发现、自动分析和在线条件检索，方便运维、研发人员排查故障。

(7) 服务微管控：通过一系列细粒度的治理策略，提供方法级治理和动态生效功能，在故障发生时可以多管齐下，在线调整，快速恢复业务。



图1 功能架构图

1.2 技术架构

开放数字化架构平台（ODA Canvas）以自研自主可控为基础，使用云原生架构，以容器、微服务、DevOps、服务网络、智能网关、可观测性等技术，实现服务开发、部署、管控、运维一体化。

云原生架构通过一组架构原则和设计模式，将应用中的非业务代码部分进行最大化的剥离，从而让云设施接管应用中原有的大量非功能特性^[4]，具备轻量、敏捷、高度自动化的特点。平台技术架构图如图2所示。

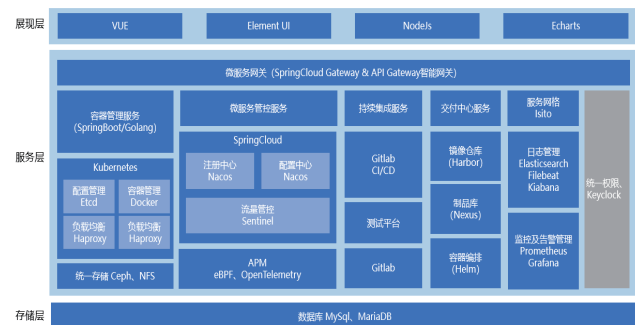


图2 技术架构图

1.3 部署架构

开放数字化架构平台（ODA Canvas）作为运营商核心的

基础设施平台，承载着企业的关键业务，任何服务中断都可能对企业运营产生重大影响。因此，平台部署架构的高可用设计尤为关键。开放数字化架构平台（ODA Canvas）部署通过冗余设计、负载均衡、故障转移和自动恢复等多重保障机制，确保了系统在面对单点故障时仍能稳定运行，保证了业务的连续性和服务的可靠性，平台部署架构如图3所示。

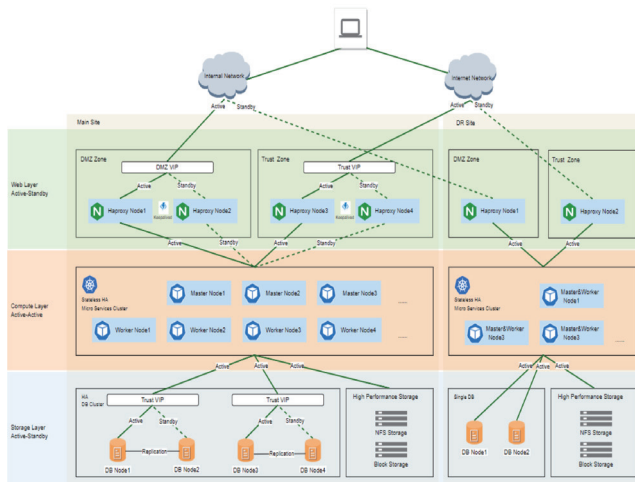


图3 部署架构图

(1) Web层：采用Keepalived和HAProxy的高可用性方案，确保Web服务的连续性和稳定性。Keepalived负责健康检查和故障转移，而HAProxy作为负载均衡器，能够在服务器出现故障时自动将流量切换至健康的服务器，从而实现无缝的服务接管。

(2) 计算层：使用至少3台服务器作为Master节点，负责集群管理、调度和微服务的协调；使用至少4台服务器作为Worker节点，每个无状态服务启动多个副本，通过反亲和和设置使其分布不同的工作节点。所有应用层设备上的服务均是启动状态，且同时提供服务。

(3) 存储层：为业务系统提供高可用数据环境和高性能存储。

① 数据库存储：配置4台服务器作为数据库节点，分为2组进行分库分表设计，每组内的2台服务器互为主备，确保数据的高可用性和一致性。后端使用高性能的磁阵，提供稳定且高效的存储解决方案。

② 文件存储：采用MinIO作为S3兼容的分布式对象存储，用于存储业务系统文件和附件。MinIO的高可用性与后端磁阵的强大存储能力相结合，满足业务对存储性能的严格要求。

③ 共享存储：选用Ceph作为Kubernetes环境中的分布式存储解决方案，为容器日志和其他管理网元数据提供持久化存储。Ceph不仅具备高性能和分布式存储的特性，还具备强大的自我修复和数据复制能力，确保数据的可靠性和可用性。

2 关键技术

开放数字化架构平台（ODA Canvas）是一款可靠、高效、可扩展的云原生管理平台，云原生的代表技术包括容器、微服务、服务网格、不可变基础设施和声明式 API，具备组件可插拔、系统可组装、服务可管控能力。平台主要解决的问题是：（1）组件的自动化开通：自动完成组件的安装、OpenAPI 的发布与订阅、功能与权限的配置，以及组件的初始化设置。（2）服务的管控和治理：在组件化场景下，平台专注于“组件异构集成”和“服务异常定位”，即组件间如何高效互联，组件与异构系统如何集成，当发生故障时如何快速定位。

2.1 基于事件的 ODA 组件自动化开通技术

ODA 组件是经过标准化的应用，支持通过 Open API 相互连接。ODA 提供了一种标准的六边形元数据描述格式，借助声明式 YAML 描述文件，可以全面编排组件的核心功能，包括需要发布的 Open API、所依赖的 Open API，以及安全设置、事件通知、环境依赖和管理操作^[5]。这种设计使得组件在 Canvas 平台上的部署过程能够实现全自动化，无需人工干预。ODA 组件定义流程如图 4 所示。

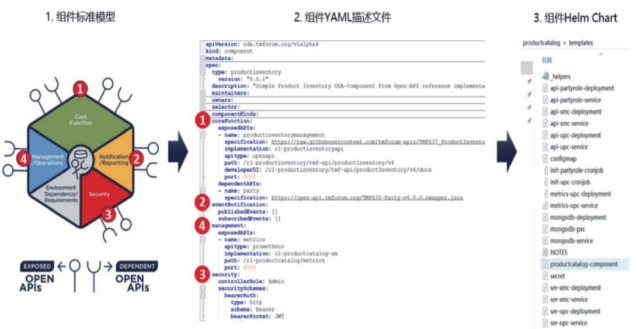


图 4 ODA 组件定义流程

平台基于组件数据模型和 Kubernetes CRD（custom resource definition）扩展机制设计了 Component 和 OpenAPI 两种自定义资源，当平台创建自定义资源的实例时，Kubernetes 会依据预设的数据模型标准和验证规则进行严格的校验，并将其存储在 etcd 数据库中。

在自定义资源被存入 etcd 之后，资源的生效需要 Operator 实现^[6]。Operator 是一种用于管理自定义资源的控制器，通过监听自定义资源的状态变化，自动执行操作，实现对自定义资源的生命周期管理和业务逻辑的自动化。为基于事件的 ODA 组件零接触自动化开通，平台设计研发了多种 Operator，包括：（1）Component Operator：管理组件生命周期，从 ODA 组件中获取 API 发布、安全设置、事件通知、环境依赖和管理操作等元数据，并根据这些信息自动触发对应的开通动作；（2）OpenAPI Operator：管理 OpenAPI 的生

命周期，为组件中每个 OpenAPI 创建独立的 API 资源。负责收集 API 的名称、描述、端点地址、协议等关键信息，并将这些信息发送给 API 网关组件，完成 API 的注册和对外开通；

（3）Security Operator：负责从 ODA 组件中提取元数据，并据此在统一鉴权组件中自动配置身份服务。它涵盖了组件的功能菜单、安全控制策略、预定义角色和权限，以及默认用户设置等多个安全维度。

此外，平台充分利用 Kubernetes 社区提供的 Operator，以实现组件服务的深度治理。

2.2 分布式全系统追踪与性能分析技术

平台基于分布式全系统追踪与性能分析技术，解决组件化场景下分布式服务异常和性能瓶颈定位及日常状态监控困难的问题。平台创新性的融合分布式调用链和内核数据，通过故障模拟和模型训练，实现了故障的自动发现、自动分析和在线条件检索。通过这种设计，平台能迅速推送疑似故障位置、潜在原因以及相应的概率分析，将分布式架构异常定位时间大幅缩短至分钟级。分布式全系统追踪与性能分析实现架构如图 5 所示^[7]。

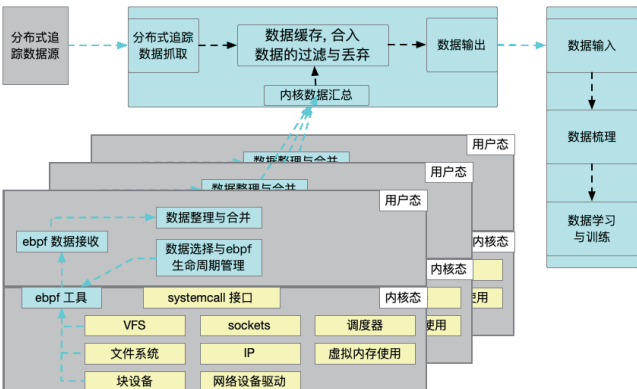


图 5 分布式全系统追踪与性能分析架构

数据源收集：平台基于 eBPF（Extended Berkeley Packet Filter）技术抽取适当的内核函数的参数、返回值和相应的引用数据，以无侵入方式保证程序的安全性、可靠性。

数据整合：数据整合的目的是实现内核数据与应用服务提供的分布式追踪数据集的无缝对接。平台引入限时缓存机制，通过多样化的关联要素，对数据进行精确关联与融合，提升数据访问效率。采用基于特征的数据过滤算法识别最具代表性的样本数据。

数据训练：通过实施混沌测试来模拟各种潜在的异常情况以获取海量数据样本，利用聚类算法逐步构建出一个精准而高效的异常定位和识别模型。这一模型的建立，能够帮助用户迅速而准确地定位问题。

2.3 面向 API 的异构系统混合集成技术

面向 API 的异构系统混合集成技术，以异构 API 资产为

驱动，融合低代码、云原生、AIGC 思维链、服务网关等前沿技术，实现了异构系统服务的低代码开发、服务发现、智能组合、自动纠错和实时监控。

该技术作为平台的底层 API 调度引擎，负责原始 API 的接入管理、原始 API 的编排组合、服务的调度代理、认证、限流、熔断、负载、监控^[8]。整体的架构如图 6 所示。



图 6 面向 API 的异构系统混合集成技术整体架构

API 智能网关作为业务系统的出入口，通过接口封装、协议转换和安全认证等功能来实现数据的高效传输和无缝交互，包含两大核心部分：管理端和调度端。管理端负责 API 的接入管理、鉴权配置以及限流熔断等策略的设定，调度端负责进行 API 的调度工作。

服务编排调度旨在将多个独立的 API 服务组合成一个完整的业务 API 服务，以满足复杂的业务需求。它提供了可视化的编排界面、灵活的流程控制、事件驱动的执行机制以及实时的监控和管理功能。通过服务编排调度，企业可以快速构建、部署和调整业务服务，实现复杂业务流程的快速构建和灵活调整。

针对电信行业的特定需求，该技术采用基于机器学习的资源调度算法，根据历史数据和实时信息预测网络资源的需求变化，实现资源的动态分配和均衡负载。同时，利用加密算法和身份验证机制增强业务接口的安全性。

3 总结

ODA 作为一种新型的数字化架构，正在逐步成为通信行业数字化转型的关键驱动力。随着技术的不断进步和市场需求的不断变化，开放数字化架构平台（ODA Canvas）将继续向标准化、智能化、自动化和安全化的方向发展。同时，ODA 的应用领域也将不断拓展，构建一个更加完善的生态系统来推动整个行业的发展。

参考文献:

- [1] TM FORUM. A future vision for the software market that the telecom industry needs to survive and thrive [EB/OL].(2020-06-17)[2024-05-13].<https://www.tmforum.org/resources/whitepapers/a-future-vision-for-the-software-market-that-the-telecom-industry-needs-to-survive-and-thrive/>
- [2] ODA Canvas Design. TM Forum[EB/OL].(2024-04-18)[2024-05-13].<https://tmforum-oda.github.io/oda-ca-docs/canvas/Canvas-design.html>.
- [3] CNCF. CNCF Cloud Native Definition v1.0[EB/OL].(2021-10-01)[2024-05-13].<https://github.com/cncf/toc/blob/master/DEFINITION.md>.
- [4] Cloud Native Computing Foundation. Cloud native landscape[EB/OL]. [2024-06-25].<https://landscape.cncf.io/>.
- [5] TM Forum. ODA Component Inventory v13.0.0[EB/OL]. [2024-05-06].<https://www.tmforum.org/resources/introductory-guide/oda-component-inventory-v13-0-0-ig1242/>.
- [6] Kubernetes Operator[EB/OL]. [2024-03-15].<https://kubernetes.io/docs/concepts/extend-kubernetes/operator/>.
- [7] EMRE ATES, LILY STURMANN, MERT TOSLALI, et al. An automated, cross-layer instrumentation framework for diagnosing performance problems in Distributed Applications[C]//Proceedings of the ACM Symposium on Cloud Computing (SoCC'19). New York: Association for Computing Machinery, 2019:165-170.
- [8] 王瑜, 丁偕. 面向服务管控的 API 网关设计与实现 [C]// 中国计算机用户协会网络应用分会. 中国计算机用户协会网络应用分会 2017 年第二十一届网络新技术与应用年会议论文集. 上海: 万达信息股份有限公司, 2017:222-233.

【作者简介】

任翔(1992—),男,山东济宁人,本科,初级工程师,研究方向:云原生、虚拟化、分布式计算与系统等。

冯学军(1979—),男,河南安阳人,本科,研究方向:云原生、5G、算力网络、通信大模型等。

李滨(1984—),男,山东济南人,硕士,研究方向:5G、AI大模型、机器学习、智能控制、软件工程等。

孙学志(1985—),男,山东潍坊人,本科,副高级工程师,研究方向:软件工程、5G、SDN、NFV、物联网、AI等。

杨强(1992—),男,内蒙古牙克石人,硕士,中级工程师,研究方向:大数据、软件工程、智能网关等。

(收稿日期: 2024-07-16)