派拉软件 PARAVIEW SOFTWARE | Red Hat

上海派拉 API 网关平台红帽联合

解决方案技术白皮书

2021 容器云 职业技能大赛

聚力 开放从容不凡 变现容器价值

目录

1. 执行摘要

1.1 汽车行业数字化转型是必由之路

在全球汽车市场呈现缓慢增长之际，中国的市场地位正在逐年上升。多家外资汽车厂商已经增资了其在国内的合资公司。在新形势下催生的产业组成结构、新市场下诞生的崭新技术趋势，无一例外将成为汽车行业发展的重要事件。汽车制造业数字化转型升级是未来发展的必经之路。

如今，消费者希望能够实现全天候无间断的网络连接，这就是为什么在接下来的时间里，汽车行业将会更多地关注于联网汽车，以及许多其它与汽车相关的数字化转型的业务。红帽及其全球合作伙伴生态系统提供了综合的开放式数字化转型平台，旨在帮助汽车企业更快地进行创新，并通过卓越的扩展性、安全性和高效率将新服务推向市场，实现数字化转型。

1.1.1 数字化转型为车企开辟“第二跑道”

中国汽车行业处于一个挑战与机遇并存的阶段，一方面，传统车企面对核心业务利润下滑，品牌价值弱化的压力，但同时，新能源汽车，自动驾驶，网联汽车，移动出行等也带来了新的业务增长点。

“电动化，网联化，智能化，共享化”，围绕着新四化，越来越多的车企将战略重心转向价值链延伸以及用户的全生命周期管理上，通过云计算，5G 网络，物联网（IoT），人工智能，机器学习，构建以“开放平台，开放协作，开放创新”为宗旨的数字化转型高速公路，打造汽车移动生活圈，成为了车企在数字化时代脱颖而出的“第二跑道”。

1.1.2 让用户享受期望的数字化体验

利用 DevOps、容器和微服务来加速现有应用程序并构建新的云原生应用程序，为客户提供期望的数字化体验。

什么是数字化体验？“数字化体验”是通过数字技术实现的用户与组织之间的互动，其目标是提供“零接触”体验：即通过预测和满足客户的需求来提升对用户的服务，需要使用来自早期交互的数据来提供个性化关注以快速解决问题。

在汽车行业，尤其是目前的车后市场，服务敏捷性对于满足用户的期望至关重要。

为实现这一目标，许多组织正在全面改进 IT 基础架构以实现 DevOps。利用微服务和容器平台，可实现更快，更高效的应用程序开发和服务交付。

1.1.3 开放协作

开放式车企取决于合作伙伴之间的紧密合作，所有合作伙伴共同致力于实现开放式框架的总体目标。红帽及其合作伙伴正在努力提供解决实际问题的方案。红帽的使命是成为客户、贡献者和合作伙伴社区的催化剂。没有一家供应商能够提供完整的端到端解决方案，因此合作至关重要。红帽生态系统汇集了行业领先的合作伙伴和值得信赖的开源社区，提供创新并经过验证的集成解决方案，如多核部署以及将单片应用程序转换为云原生微服务的环境。这些产品和服务经过测试、支持和认证，可在基于 Red Hat 的基础架构上执行，详细的认证政策可确保兼容性和稳定运行。

红帽和派拉紧密协作，将红帽的 PaaS 容器云敏捷集成解决方案和派拉的 API 网关平台产品深度整合，帮助中国车企在数字化转型中的 API 经济战略和规划路线的重要领域获得先机。

1.2 汽车行业如何在 API 经济中获益

API 的定义是一种应用编程接口（API），它是非常古老的术语，被用来描述软件程序的技术接口，一个软件程序通过它调用另一个软件程序。通常情况下，这些 API 非常复杂，并不意味着真正的广泛使用。企业内部的一些其他软件程序可能会使用 API 来调用该程序；公司外部的合作伙伴也可能会使用它，但难度很大。

这个长期存在的定义并不是让企业对 API 经济感到兴奋的原因。兴奋点反而是围绕着所谓的业务 API 或网络 API。这些业务或网络 API 是可识别的业务资产的易于理解的接口 -- 例如，客户记录、账户、产品目录、价格、订单等等。业务 API 是企业的一个公共角色，它暴露了所定义的资产、数据或服务，供组织内部或外部的选定的开发人员消费。使用业务 API 对应用系统来说很简单，开发人员可以使用、访问、理解和调用。而且，由于商业 API 扩展了企业并打开了新的市场，应用开发者可以很容易地利用、宣传和聚集公司的资产，以便广泛地消费。

业务 API 的适当运用可以带来很大的好处，包括：

- 整合和标准化组织内的通用 API 或简单的业务服务。

- 通过拥有企业业务服务的中央储存库和索引，如 " 检索信用分数 "，降低运营成本。
- 通过内部和外部各方安全、快速地获取商业服务，加速数字项目并改善上市时间。
- 确定合作生态系统，特别是在你的行业之外制定新的增值产品和服务，以提高竞争力。
- 为货币化目的定义新的商业模式，如移动市场；也就是说，将贵公司的业务能力与合作伙伴的业务能力聚合在一起，以提供多样化的相关或互补服务。

1.2.1 API 战略的常见业务驱动因素

那些成功执行 API 战略的公司专注于四个关键驱动因素中的一个或多个：速度、范围、物联网（IoT）和领域。

1.2.1.1 速度（也被称为双速 IT、双模 IT 或多速 IT）

这一驱动力的重点是允许企业和 IT 组织以不同的速度运行。传统的核心记录系统的 IT 管理必须是以一定的速度推进。试图将快速变化强加给企业的核心系统可能会导致中断或安全暴露。然而，企业需要对新的机会和竞争威胁作出非常迅速的反应。它需要更高的变化率，而不是通过对记录系统的控制变化来实现。使用 API，你可以预先打包核心系统资产供企业使用，以创建新的和创新的参与系统。这个驱动因素往往是推动 API 在企业中使用的第一个因素。

1.2.1.2 影响力

为了进入新的市场和获得新的客户，你可以向其他企业提供 API，如合作伙伴，他们可以为你的企业创造额外的收入和新客户。

例如，一个汽车经销商可以通过独立的三方购车应用程序，来接触新客户。经销商可以接触到一组可能没有直接联系过他们的潜在汽车买家。

1.2.1.3 物联网或设备

在许多行业，设备与 API 结合使用，以提供新的和创新的解决方案。这往往会以三种方式之一发生：

- 一个设备通过 API 调用发送数据，例如一辆联网的汽车向保险公司发送关于司机行为的数据。
- 一个设备通过 API 调用被发送一个命令，比如说一个车载援助服务发送全球定位系统（GPS）技术和地图的更新。
- 一个设备通过一个非 API 调用发送数据，使用其他技术，如 MQTT（一种用于遥测设备的高容量信息传递协议和传输）。因为并非所有的数据调用都需要一个动作。然而，API 可以访问企业内部的数据，并寻找或应对特定的情况或事件。例如，车辆监测传感器不断发送数据，并分析数据以确定潜在的问题。如果发现问题，监测公司会在故障发生前使用 API 提醒司机和服务部门。

1.2.1.4 业务领域

通常情况下，业务领域指的是多个业务线之间的互动。它们在很大程度上可以独立工作，但从共享数据中受益。API 允许数据以受控的、安全的方式被共享。业务领域也可以被看作是物理位置。拥有多个地点的公司，其中可能包括云和企业内部的数据中心，有时会使用 API 作为一种方法来保护和控制不同地点之间的数据流。对基于地理和国家规范的监管和合规约束的考虑变得很明显。

企业往往一开始就把重点放在对速度的要求上。在这一领域取得初步成功后，他们会处理其他驱动因素。企业从所有四个驱动因素的 API 中受益的情况并不少见。

1.2.2 汽车行业的 API 运用前景

通过科学的 API 识别方法可以找到应用于汽车行业的 API 用例。

1.2.2.1 内部开发（移动应用开发）API

- 一般信息访问

一般信息是不针对使用该应用程序的特定客户的信息。它可能包括关于汽车制造商及其产品的一般信息，如产品目录和产品描述、可用的配件、激励措施、车主手册信息、服务地点、产品价格和可用性、以及评级和评论。对于经销商来说，API 提供的信息可以包括位置、开放时间、可用库存和价格。

- 自定义信息和交易信息访问

提供为使用该应用程序的客户量身定做的信息和交易。显然，这些 API 需要额外

的安全性，以帮助确保适当的访问。符合这一类别的 API 可能包括将特定的车辆与所需的服务进行映射，管理召回，评估以旧换新的价值和安排服务预约。

1.2.2.2 移动优势与安全

在移动设备上使用该应用程序的客户可以从使用手机或平板电脑的功能与汽车制造商或经销商提供的 API 结合起来中获益。设备功能样本包括相机、GPS 服务、近场通信（NFC）和数字钱包。移动应用程序可以为一系列的功能访问 API；典型的例子包括解锁车门或启动汽车。然而，这些 API 需要被保护，以便只有车主可以访问和执行这些功能。

从移动应用到汽车 API 的直接调用可以被黑客利用，导致盗窃或其他危险。因此，验证用户的身份是至关重要的。

1.2.2.3 伙伴合作 API

合作伙伴提供许多汽车部件，如收音机、导航系统、电话集成和娱乐系统，以及安全功能，如车道警告系统。接近车辆警报等。这些合作伙伴需要访问汽车 API 以与汽车的活动整合，合作伙伴提供的免提控制的 API 有助于提高安全性。所有这些组件都为汽车增加了价值，并为汽车制造商和合作伙伴创造收入。

同样，需要建立适当的安全机制来保护 API 调用和信息。车载信息娱乐系统提供的功能已经远远超过了简单的立体声接收器。很快，它们可以被用来提高驾驶体验的各个方面。为了最大限度地提高创新和减少上市时间，汽车制造商应该为外部合作伙伴和开发人员提供他们所需的工具，以创建直接在汽车信息娱乐系统中运行的应用程序。这些应用程序应向司机提供音频和语音内容和信息，并向乘客提供视频和深度参与。API 方法包括访问车辆信息、I/O 命令、通信能力、音频和视频播放、导航、电话系统、用户界面和信息娱乐系统命令和实用程序。

随着客户习惯于车内人机交互，他们可能会要求能够直接从车辆上访问和购买产品和服务。制造商与希望直接向汽车提供优惠和电子商务界面的保险供应商和广告商合作，可以获得很多好处。开发商可以创建车载电子商务应用程序，使消费者能够根据驾驶行为和模式分析来购买个性化的保险政策。零售商可以让购物者从车上购买产品，并提供路边取货，以加快服务速度。这些应用程序必须在所有各方之间提供安全的数据整合，保护敏感的消费者数据并满足行业标准，如支付

卡行业数据安全标准和监管要求。

这些应用程序还需要保护用户隐私，同时向保险公司和零售合作伙伴提供所有这些宝贵的消费者数据。一个 API 解决方案能够创建一个数据透镜，安全地将重要的驾驶数据，如支付、驾驶资料和汽车位置，投射为制造商的数据。一个完整的 API 解决方案使汽车、企业和合作伙伴之间的交易整合和协调成为可能，以提供安全、可靠的服务。

API 有助于使作为汽车制造商或经销商的你做生意变得容易。它们可以从一些途径引入新的价值，如零部件或汽车交付的供应链整合、独立汽车应用的转售协议以及与银行和保险公司的整合，以简化销售过程。

1.2.2.4 公共 API

汽车公司可以将内部使用的和与合作伙伴使用的许多相同的 API 部署为公共 API，以推动更多的业务并帮助获得新客户。汽车购买的竞争非常激烈。来自三方公司的应用程序使买家能够比较多个制造商的汽车，以提高购买体验。所有的人都可以通过访问公共 API 来获得汽车产品和配件、细节、价格、评论等方面的好处。

许多汽车公司提供了公共 API。将企业的触角延伸到可以向你发送业务的其他行业，是转向 API 经济的一大诱因。例如，为特殊场合推荐礼物的应用程序或旨在帮助家庭计划送孩子上大学的应用程序可能会建议购买新车或二手车。

1.2.2.5 社交网络 API

您可能已经使用腾讯或阿里、抖音等公司的社交 API，将这些信息与你自己的 API 混合起来。在社交媒体中对提到您的公司和趋势的具体内容采取行动，可以利用它来获得机会或避免问题。您可以将提及你的公司的微博与你自己的分析相结合，以确定你是否必须采取行动解决客户满意度问题，促进积极的评论或提供产品折扣。对客户的投诉迅速采取行动，并提供优惠，可以将负面评论转化为与你的公司做生意的建议。

当我们进入汽车时，社交活动并没有停止，所以提供一个安全的社交环境是汽车行业内的一个关键焦点。制造商和功能供应商正在研究 API，允许语音命令访问或控制汽车的各个方面，以实现短信、电话访问和其他功能。社交媒体本身是汽

车制造商的宝贵资源。社交媒体上的评论，如 " 去上大学 " 或 "16 岁生日即将到来 "，可以引发营销机会。自动驾驶汽车腾出的时间可以进一步增加有针对性的车内广告、客户关系管理（CRM）数据的销售和新的 " 边开边送 " 商业模式。

汽车公司需要开发动态的移动应用程序，以一种吸引人的、非侵入性的方式直接向潜在客户和顾客推销他们的产品。为了使这些应用程序尽可能地无冲突，开发人员需要与社交媒体 API 集成，特别是利用社交身份进行无缝登录。一个允许与流行网络的社交身份整合的解决方案可以在两个渠道上保持便捷的访问。公司可以将他们的 CRM 系统或移动应用与移动营销 API 整合，创造一个真正优化的移动营销平台。

一个 API 解决方案可以实现网络和移动的统一访问，以确保消费者通过他们选择的渠道获得个性化的体验。此外，移动营销自动化让你为你的移动营销方案配置和建立业务规则。例如，假设你想根据客户的购买日期和驾驶里程数，向移动数据库中的所有客户发送汽车保养提醒短信。只需在企业中配置业务规则，并将最佳维护计划与汽车诊断仪的 API 相关联，你就能在指尖上获得一份客户名单。

1.2.2.6 设备集成和可穿戴设备

Gartner 公司估计，到 2020 年，全球约有五分之一的车辆将拥有某种形式的无线连接，总计超过 2.5 亿辆联网车辆，使在远程信息处理的主要功能领域产生重大影响。

自动驾驶、信息娱乐和移动服务。这些趋势正在改变着汽车和交通之间的关系。车主和制造商超越了驾驶体验，成为真正的互联体验。

随着联网汽车的使用案例变得越来越普遍，它们可以被归纳为若干类别，包括远程服务、远程车辆功能、位置服务、礼宾服务和增值服务 -- 所有这些都是通过 API 实现的。

1.2.2.7 数据和分析

汽车制造商和经销商可以收集客户的行为数据并进行分析，以帮助识别营销机会。这种策略可以通过传统的信息收集或通过车联网技术。

通常情况下，数据和分析是针对特定的内部受众的。然而，通过 API，数据和分

析可以提供给更多的内部受众，他们可以从相同的数据中获得额外的价值。

汽车可以为营销提供有价值的数据，或为其他行业提供机会。例如，汽车数据可以确定旅行路线，这对沿途或附近做广告的企业可能很有价值。企业可以通过 API 向寻找附近餐馆或活动的司机推广产品，而保险公司可以将他们认为非常有价值的驾驶行为信息货币化。

1.2.2.8 行业标准

汽车行业正在与 W3C 标准组织合作，为行业定义 API 标准 (https://www.w3.org/2013/02/autobg.html.en) 互联汽车需要标准来实现与所提供信息的消费者的互操作性。所有的供应商 - 汽车、娱乐系统、移动电话等 -- 都需要在整个制造链中合作，以避免接口的不兼容实现。供应商有望通过与他们的合作伙伴生态系统整合的增值服务进行竞争，提高他们的数字参与度，并为他们的客户提供个性化的产品 -- 但不是通过不同的 API 接口。

1.3 网关平台是最重要的 API 基础设施

1.3.1 解决哪些问题

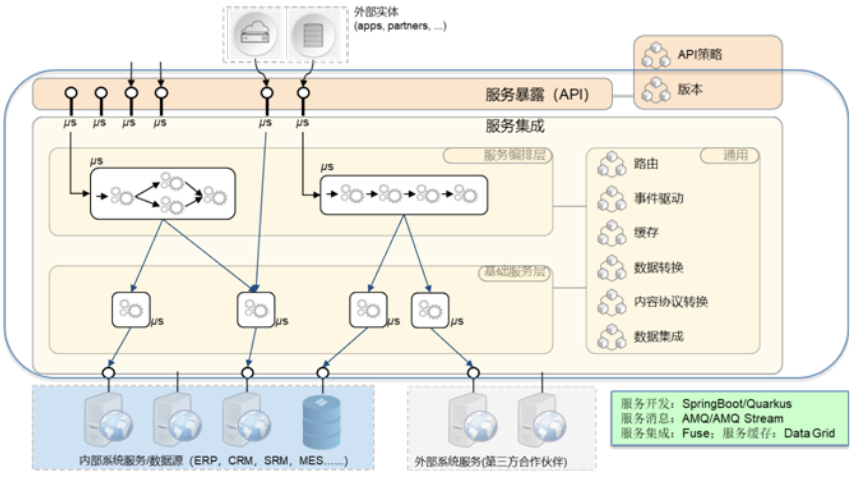
以红帽企业级开源产品 OpenShift 容器平台为基础，以红帽 Integration 中间件（Fuse）为集成层打造的派拉 API 网关解决方案能够完美满足企业的 API 业务要求，它主要解决以下四方面问题：

- 提升企业管理、运营能力

- 加强电子商务信息化能力
- 达成信息安全首要建设目标
- 汽车行业的数字化转型助力

1.3.2 网关的核心功能

下图是企业总体集成功能架构，API 网关的功能边界在蓝色圆角方框，目的是保障内部与外部和内部到内部的 API 连接可靠性、稳定性和安全性。



我们用网关平台的四类功能来支撑汽车行业 API 应用：

- API 智能安全网关
  - 安全认证、身份认证、流量控制、路由转发、AI 智能识别恶意访问、异常处理、负载均衡、WAF、服务降级、服务熔断、蓝绿测试；
  - 基于微服务化容器化、智能安全认证的 API 集成网关，提供稳定、高效、可扩展的自研平台。
- API 监控管理
  - API 网关管理、分布式网关集群管理、插件扩展、日志分析、日志告警。
  - 提供完善的 API 分析，实现可视化、智能化、一体化决策。

- API 管理平台
  - 项目管理、API 部署、API 发布、API 设计、API 运维、审批管理　API 测试、服务管理、运营管理。
  - 提供完整云端 API 生命周期管理 , 提供灵活、无代码的 API 编排平台
- API 开放平台
  - API 中心　API 文档　API 审批　商户注册　API 申请　沙箱申请　热门 API　API 详情
  - 通过免费增值、购买、订阅或者消费模式的任意组合，实现 API 盈利和业务创新。

1.4 业务上的特色是什么

1.4.1 业务价值

本方案的业务价值体现在以下几个方面：

- 打通 API 网关通道
  - 建立 API 集成标准规范
  - 实现统一接入，便于 API 管理，提供 API 智能分析
  - 实现文档自动生成、为消费者提供指引，提高服务质量
- 业务流程风险控制
  - 简化和控制开发者对服务和数据的调用
  - 建立更广阔的合作伙伴和能力开放生态系统，借力工具，降低移动应用交付周期
- 提高企业生产力
  - 为 H5 页面 /API 提供端到端的身份安全、数据安全、安全防护方案
  - 根据风险程度作出不同的处理：限流、限速、断流、引流到蜜罐

- 降低运营成本
- 变现数据价值，创造新利润增长点
- 建立数字生态系统，增强业务价值
- 通过分析和优化，提高效率

1.4.2 功能场景

派拉 API 平台是一款安全高效准确传输的应用集成和数据交换平台，基于云原生、多租户、自主研发的新一代 API 平台商业产品。

API 网关平台包含了 API 智能安全网关、API 管理平台、日志监控分析、API 门户、Fuse 集成平台，可以为业务中台和技术中台提供底层支撑的一体化平台。其主要特点是高可靠、高性能、强大的安全防护和扩展能力，是敏捷响应业务需求和企业数字化转型的保障。

- API 智能安全网关是最核心的模块，它是整个产品的主要底层组件，功能包括：路由转发、安全、认证、格式转换、流控、微服务等。
- API 管理平台主要包括：实现了以项目为视角的 API 设计、API 定义、API 开发、API 测试、API 发布、API 上线、API 下线的全生命周期管理，以及服务注册中心集成管理、部署环境管理、应用管理、API 网关授权、API 授权、运营管理、审批管理、日志查询、异常告警、自定义 BI 报表以及集群扩展等功能。
- API 门户主要包括了欢迎门户主界面、API 中心、文档中心、开发者中心、开发者自助注册、自动对接的集成功能。
- 红帽 Fuse 集成平台是基于开源社区（如 Apache Camel 和 Apache ActiveMQ），是企业敏捷集成解决方案的一部分。其采用的分布式方案能让团队按需部署集成式服务。以 API 为中心且基于容器的架构能将服务分离出来，以便单独进行创建、扩展和部署。

1.5 技术上的特色是什么

1.5.1 开源容器化

- OpenShift 对云原生的支持：容器化、微服务化、DevOps 化、中间件等。
- 支持大型企业的 API 生命周期管理结合 K8S 和 Docker 环境，实现 API 定义、开发、测试、部署、上线、文档的生命周期管理，实现服务提供者、服务消费者、运营管理的发布消费一体化流程。

1.5.2 强大的 API 安全防护能力

1.5.2.1 身份认证

- 派拉有自己的统一身份认证管理中台，后期的安全认证中心是 API 智能安全网关的核心组成部分，派拉的 API 智能安全网关产品和统一身份认证产品是无缝集成。
- 派拉 API 智能安全网关支持 APIKEY、JWT、BasicAuth、Oauth2、SAML、OIDC 等安全插件，功能丰富，开箱即用。
- API 鉴权插件，通过 ABAC 或者应用的授权可以实现第三方合作伙伴、开发者等访问 API 的细粒度授权。

1.5.2.2 安全防护

- 自带 WAF 产品（可以实现防御 SQL 注入、黑客扫描、XSS/CSRF 等常用攻击、IP/ 用户的主动拦截和防护）。
- 流量调度中心（可以根据后端服务的状态和服务器状态，进行动态的流量权重变更。根据实际情况进行服务降级和服务熔断操作）。
- 流量控制（可以根据用户 /IP、API、服务、全局进行限流操作）。
- 自带大数据日志分析产品，可以进行用户的异常行为调查分析。

1.5.2.3 数据安全

- 自带加解密、加签验签等安全插件，支持国密算法和国际算法。
- 传输层支持 SSL/TLS 加密传输模式。

1.5.2.4 超大规模企业的性能优势

- 性能卓越，单集群节点支持 10w 并发。
- API 智能安全网关基于 Nginx 研发，2 台 8 核 8G 的集群节点可以支撑 10w 并发。
- 日志基于自研的日志分析产品，在 5kw 条数据下，查询时间不超过 3S。
- 可以结合 K8S 和 docker 实现流量自动缩扩容，满足大流量并发自动扩展。
- 使用 Redis 的缓存，减少 IO 操作，支持流量高并发。

1.5.2.5 低代码全面集成

- 通过与红帽 Fuse 整合，采用其包含基于浏览器的拖放式界面 Fuse Online，可以快速创建集成服务。
- 借助红帽 Fuse 的 200 多个连接器，用户几乎可以连接所有对象——从传统系统到软件即服务（SaaS）应用，从应用编程接口（API）到物联网（IoT）设备。它还支持编写自定义组件，以满足特定的终端连接需求。

2 方案背景

2.1 业务发展趋势

2.1.1 企业级数据互联互通

API 智能安全网关是企业业务数据传输的关键枢纽，主要适用的传输场景如下：

- 集团内部系统的业务数据交互。
- 系统间的快速集成及标准 API 规范。
- 私有云、公有云、传统办公的不同网络区域的数据交互。
- 第三方合作伙伴、公有云、移动端和内部业务系统的数据交互。
- 内部到外部的公有云或者银企直联的数据交互。

2.1.2 企业级的 API 安全保护

企业沉淀了很多的业务能力需要发布到外网，将 API 开放给合作伙伴或者第三方，

通过 API 生命周期安全、身份认证、安全防护、日志审计、SDK 动态感知等手段对 API 进行安全防护，防止造成 API 被非法调用和数据泄露。

2.1.3 企业级的 API 开放能力

企业会将很多业务能力开放到互联网，通过企业内部业务能力、合作伙伴能力、第三方云端服务能力进行快速整合，敏捷响应客户需求，API 门户帮助第三方合作伙伴实现自助注册以及自动化对接，可以在很大程度上降低沟通成本。第三方合作伙伴快速集成的主要步骤如下：

- API 提供者进行服务的发布上线。
- API 消费者在 API 门户上进行账号注册，并进行 API 申请，申请授权。
- 授权通过后，下载 SDK，并根据 API 文档进行业务开发测试联调。

2.1.4 跨网络的分布式网关集群代理

企业需要多个区域网络（混合云）的部署分布式网关，派拉 API 安全网关实现分布式网关集群代理能力，实现 API 安全网关的幂等性，也就是在任何一个网络区域客户端访问本地的安全网关都可以访问其他网络区域的 API, 并可以提高网络区域交互的安全性。

2.1.5 技术发展趋势

- 双速 IT、稳敏双态、云原生容器化、K8s 化。
- 容器化避免开发和运维的不一致性，大幅度提高效率；开发运维领域的自动化。
- 微服务化。
- DevOps 化。
- 开放式混合云：速度、稳定、规模。
- 开放平台、开放协作、开放创新、开源化。

2.2 业务挑战

2.2.1 异构问题

- 存量业务系统非常多。
- 技术标准不统一。
- 套件和软件定制化产品多。
- 耦合性强，开发维护成本高，服务模型多样化，业务协议不统一。

2.2.2 体验和交付

- 漫长而复杂的开发流程。
- 重复而费时的测试工作。
- 缓慢而低效的发布流程。

2.2.3 基础架构问题

- 营销端云化，如何利用云端基础设施，以及设计制造如何集成的便利问题。
- 系统扩展，架构演进难度大，牵一发而动全身。
- 应用扩容成本高，效率低。

2.2.4 监控问题

- 运营产品数据丢失。
- 监控维度不立体，端到云动监控体系如何建立。
- 故障分析查找困难。

3. 方案概述

3.1 目标概述

3.1.1 制定管理和规范体系

制定企业管理集成规范，降低集成成本和运维成本；实现业务对接自助化，对接自动化，降低沟通成本。

3.1.2 敏捷响应客户需求

建设 API 低代码开发平台，快速响应业务需求；打通企业上下游业务，实现企业生态建设；借力服务编排平台，敏捷响应业务变化。

3.1.3 数字化安全价值

减少 API 数据泄露和攻击风险，提升企业信息化安全，数字化下云化、移动化、微服务化场景下提升 API 安全；保护 API 资产暴露风险和数据安全。

3.1.4 及时的智能监控告警

实现端到端、端到云、云到端的全链路监控追踪体系，实现预测问题，定位问题，解决问题的一体化流程；建设应急响应体系，增强业务可用性。

3.2 设计原则

3.2.1 统一安全防护

通过 API 网关统一管理 API 授权和流量控制，让业务系统可以专注于业务的集成。

3.2.2API 的全生命周期管理

通过 API 管理平台，统一管控 API 从定义、发布、上线、下线的全生命周期管理工作，配合相应的审批流程，实现 API 的规范发布。

3.2.3 API 能力外放

通过 API 门户，将 API 能力外放，做到服务资产化，提高 API 的复用率，实现企业对外提供服务的能力。

3.2.4 快速集成

通过与红帽 Fuse 整合，赋予网关强大的协议对接和服务编排的能力，作为数据的提供层，能够低代码高效地对服务进行集成。

3.3 方案特点概述

3.3.1 容器化

容器化的好处是什么?

- 开源化：基于纯开源容器平台底座：OpenShift。
- 平台化：可以在平台上扩展和复用其他厂商的方案。
- 标准化：通过容器化实现应用开发标准化。
- 多集群；混合环境多集群部署。

3.3.2 混合多云化

- 支持混合云多云部署。
- 可以在不同的公有云上部署。
- 可以在私有云包括裸金属上部署。
- 国际化：支持全球部署。
- 支持混合云多云部署：支持国内国双循环、出海业务、一套代码库、“一次编写，到处运行”。

4. 方案架构

4.1 方案架构概述

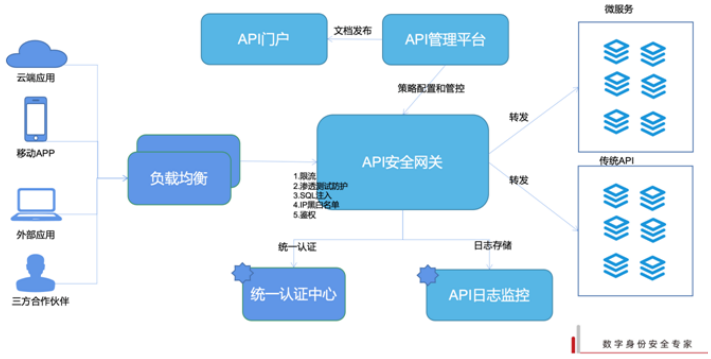
4.1.1 平台逻辑架构

- 外部业务系统通过 F5 或 SLB 负载分发到 API 智能安全网关，通过 API 智能安全网关代理到后端业务服务（传统的系统服务、遗留的企业服务总线，以及微服务）。
- API 安全网关是最核心的模块，它是整个产品的主要底层组件，功能包括：路由转发、安全、认证、格式转换、流控、微服务等。
- API 日志监控实现了多维度的日志监控、告警及数据报表等功能模块。
- API 管理平台主要包括：实现了以项目为视角的 API 设计、API 定义、API 开发、API 测试、API 发布、API 上线、API 下线的全生命周期管理，以及服务注册中心集成管理、部署环境管理、应用管理、API 网关授权、API 授权、运营管理、

审批管理、日志查询、异常告警、自定义 BI 报表以及集群扩展等功能。并且在 API 管理平台对 API 的定义会同步生成 API 门户的文档和测试样例。

- API 门户主要包括了欢迎门户主界面、API 中心、文档中心、开发者中心、开发者自助注册、自动对接的集成功能。

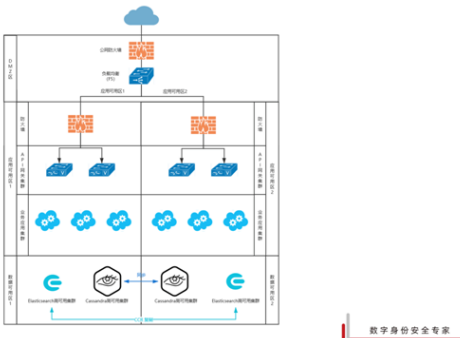
平台逻辑架构



4.1.2 单数据中心的双区高可用架构

在单一数据中心部署两套网关及数据存储服务，外部业务系统通过 F5 或 SLB 负载分发到两组 API 智能安全网关，两组数据中心相互同步，组成的双区高可用架构。

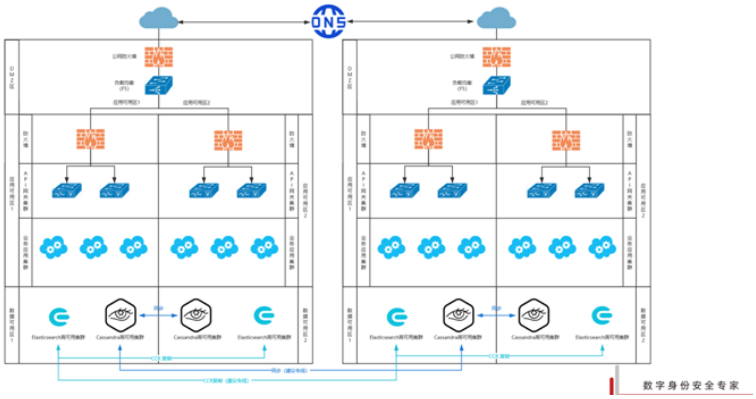
单数据中心的双区高可用架构



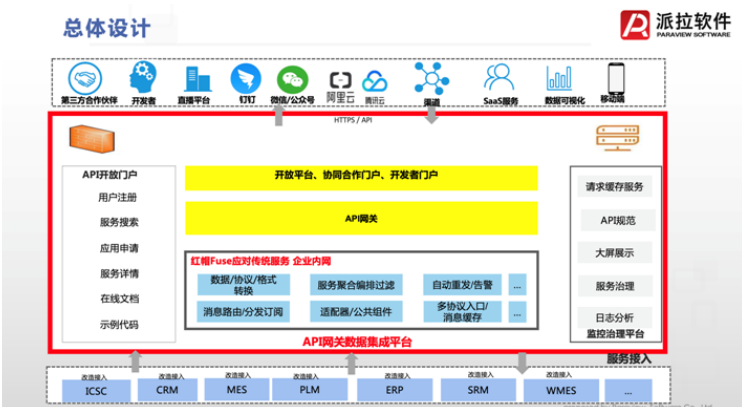
4.1.3 异地双活的高可用架构

与单数据中心同步类似，两组异地数据中心通过专线进行同步，外部业务系统流量通过 DNS 解析在两套环境间进行切换，常用于异地灾备。

异地双活的高可用架构

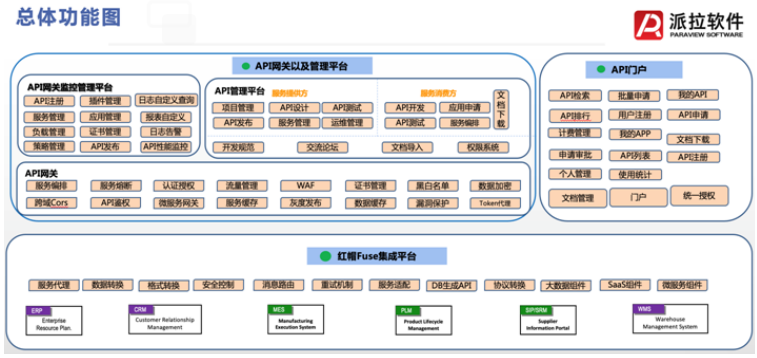


4.1.4 系统架构图



在企业系统服务至上，采用红帽 Fuse 来应对企业系统中协议和数据格式不一致的问题，并对服务进行编排聚合和过滤。API 网关对服务安全，流量，转发策略进行统一的管控，并由网关管理平台进行 API 的全生命周期管理和审批流程的管理。之后将相关的开放信息同步到门户，实现数据资源的外放。

4.1.5 功能架构图



4.1.5.1 API 管理

主要是一次性实现 API 的检索，排行，文档的管理。

4.1.5.2 服务管理

实现服务的新增、添加、删除、更新操作，主要包括服务名称、服务地址、服务方法、重试次数、请求超时时间、连接超时时间、响应超时时间。

4.1.5.3 应用管理

实现应用的增删改查操作，主要包括名称标示 APP 内部标示、组、标签信息。

4.1.5.4 负载管理

实现负载的增删改查操作，主要包括名称、代理名称、代理描述等实现。实例的增删改查操作，可以实现轮训、Http Header、Client IP、请求 Cookie、应用的请求方式，实例主要是 IP/ 端口、代理名称组成。

4.1.5.5 策略管理

实现API网关插件的所有增删改查操作，可以针对安全、认证、监控分析、格式转换、log、部署插件进行操作。

4.1.5.6 证书管理

实现证书的管理和配置。

4.1.5.7 集群管理

实现分布式集群的管理，并可以进行自动切换操作，可以进行授权、管理、集群状态展示、备注等信息。

4.1.5.8 插件自定义扩展

实现自定义插件的管理，主要是可以界面化管理开发的插件，自动发布到 API 网关上，实现自动化管理。

4.1.5.9 插件发布

实现插件发布的管理，根据集群可以选择集群的使用版本号。并且可以进行上线和下线操作。

4.1.5.10 日志分析平台

主要是对 ES 地址、ES 模板、Grafana 地址国际化模式等后端进行配置

根据请求 json 报文自定义菜单列表，主要是提供上传模板 json 报文、导出配置、新增等操作，可以自定义增删字段报文。

可以实现根据 JSON 自定义查询日志字段，以及大数据日志查询以及自定义 BI 报表。

集成 grafana 图表的维护，主要可以查看和删除操作。

4.1.5.11 用户管理

实现用户的增删改查操作，修改密码等

4.1.5.12 角色管理

实现角色的添加、删除、更改、查询操作、授权管理，可以展示出资源树供选择，不同的角色看到的资源菜单是不一致的。

4.1.5.13 资源管理

实现资源的增加、删除、查询、更新等操作，可以实现菜单的自动配置和添加，主要是三级页面的操作

4.1.5.14 机构管理

实现机构的添加、删除、更改、查询操作

5. 业务功能

5.1API 智能网关

5.1.1 运行引擎功能

API 安全网关功能主要包括负载均衡、反向代理、负载健康检查、缓存支持的功能，安全网关支持的传输协议包括：HTTP/HTTPS、TCP/IP、Websocket、GRPC、Dubbo。

5.1.2 基础功能插件

基础功能插件主要分为基础插件、安全、认证、微服务、流量控制、格式转换共 6 类插件

- 基础插件：包括：Session 保持、全局交易流水号、日志记录。
- 安全插件包括：IP 黑白名单、跨域支持、MD5 验签等安全方式。
- 认证插件：包括：APIKey、Basic 验证、ACL 认证、Ouath2 插件、Ldap 等认证方式。
- 流量控制：包括：请求限流、交易限流、请求报文大小限制、请求终止、响应报文限制、响应报文大小限制。
- 格式转换：包括：JSON 格式转换、XML2JSON 格式转换、模板格式转换。

5.1.3 集成扩展插件

集成扩展插件功能包括：ElasticSearch 日志插件、应用授权插件、跨域插件、防重放、加验签（国际算法）、加解密（国际算法）、Hmac、灰度发布、全局流量控制、Mock 功能。

5.1.4 增值开发插件

- Kafka 日志集成很多的客户需要将流量日志打到 Kafka，然后从 Kafka 在抽取到

大数据日志存储中心，日志中心在统一进行展现。

- 服务熔断降级当服务器压力剧增的时候，根据当前业务情况以及流量，对一些服务有策略的降级，以此缓解服务器资源的压力以保障核心任务的正常运行，同时也保证了大部分客户能得到正常的响应。当某个目标服务响应超时或大量异常，熔断该服务的调用。
- 智能安全防护通过集成 WAF 插件，实现防御 SQL 注入、XSS、CSRF 等 WEB 攻击，并可以根据 AI 算法识别大流量中的安全威胁。
- 服务注册中心集成（Nacos/Eureka/Zookeeper）在和微服务网关进行集成的时候，通常需要同步 Nacos 或者 Eureka 中心的服务，实现 API 网关和服务注册中心的联动。
- 全链路监控日志当客户端需要通过 API 安全网关访问后端服务的时候，派拉 API 网关产品通过埋点可以实现端到端、端到云、云到端、以及微服务内部的调用的全链路监控。
- 高级服务路由实现统一服务请求入口，提供基于 HTTP 请求头的高级服务路由功能。
- 国密支持支持国密算法 SM2/SM3/SM4 的加验签、加解密，也支持 360 浏览器端的国密证书。
- 高级格式转换主要支持 Restful 到 Dubbo，以及 Restful 到 Grpc 的格式转换功能。
- 灰度发布通过请求头、请求体实现流量灰度切换、在服务路由中设置灰度标识，实现接口的灰度发布。

5.2 API 管理平台

5.2.1 API 管理

- 项目管理项目管理具有多租户功能，一个项目经理可以创建项目，并可以管理项目内的人员。
- API 生命周期管理通过 API 设计、API 定义、API 测试、API 发布、API 运维、API 消费实现 API 的生命周期管理。

- API 文档管理可以通过 Swagger 导入也可以单独进行 API 的字段定义，在 API 发布后会同步到 API 门户上。
- API 部署环境管理实现 API 多个部署环境的管理配置。
- 服务管理配置 Eureka 或者 Nacos 的注册中心，以及对服务中心的服务进行定时同步。
- 应用管理管理员可以在后台添加应用，也可以在门户上进行自助申请，管理员添加应用包括应用 ID 应用名称、认证方式、用户、状态等参数。添加应用后还可以实现配置应用的访问认证方式，目前主要包括 APIKEY、BasicAuth、Hmac、Ouath2 这四种方式。还需要针对网关授权和 API 授权进行申请，有运维人员进行审批，审批通过后，才可以访问 API. 实现消费端的应用配置管理，并对应用实现 API 网关授权和 API 授权。
- API 权限管理针对 API 网关授权给应用的权限主要包括：APIkey、BasicAuth、NoAuth、Hmac、Oauth2。
- API 授权实现某个应用的 API 授权。
- 审批管理实现对 API 发布、API 上线、API 下线、应用网关授权、应用 API 授权流程审批。
- 运营管理通过后台实现门户的 API 产品、API 文档、个人申请的相关管理，并可以发布到门户上。
- 在线扩展和部署可以自定义开发网关插件，通过管理平台上传并发布到网关。
- 插件策略配置当定制化一些插件，需要在插件策略里面进行统一配置。
- 分布式集群管理实现多个 API 网关集群节点的管理，可以快速切换集群节点，并进行权限控制，以及可以进行多个集群节点的状态监控。

5.2.2 API 监控

主要包括: 自定义日志查询、日志审计、自定义 BI 报表、异常告警、用户权限管理。

5.2.3 API 门户

5.2.3.1 门户主页

API 门户是展示企业所有的业务能力，实现应用集成的一体化自动流程，门户主页包括热门 API、消息通知、轮播图、自助集成流程等功能模块。

5.2.3.2 API 中心

集中展示企业服务能力，根据项目为视角展示 API，API 可以进行在线申请，单个 API 可以展示具体详情，包括请求报文、响应报文、响应码、集成接入文档等。

5.2.3.3 API 文档

可以实现 API 的文档管理，主要文档分为：帮助文档、集成文档、集成规范、SDK 文档下载。

5.2.3.4 开发者中心

服务消费方实现需要进行自动注册，通过实名认证或者法人实名后才可以进行应用申请、API 申请调用、API 的查询、用户的基本信息等功能。

- 我的应用：用户可以通过企业开放出来的 API 进行应用申请、API 申请和调用，先通过沙箱环境的应用创建和 API 申请，测试通过之后可以申请正式环境的 API，运维管理员可以对 API 申请进行审批，保证开发者的应用访问 API。
- 我的 API：主要展示了用户申请的 API 详情以及 API 的申请状态，同时也可进行 API 解绑等操作。个人中心主要展示是用户个人信息和账号密码的修改等操作。

5.2.3.5 多租户自助申请

实现多租户进行自助注册账号，用户信息需要通过才可以进行注册应用，通过应用申请 API 的权限，通过后就可以进入联调阶段。

5.2.3.6 应用服务集成层

应用服务集成层用于简化集成开发以跨混合环境连接应用程序和数据。

以红帽 Fuse 为核心的应用服务集成层是一套综合的集成和消息传递技术，用于跨混合基础架构连接应用程序和数据。它是一种敏捷、分布式、容器化且以 API

为中心的解决方案。它提供服务组合和编排、应用程序连接和数据转换、实时消息流、变更数据捕获。所有这些都与云原生平台和工具链相结合，以支持全方位的现代应用程序开发。

红帽 Fuse 的定位是分布式云原生集成平台，它基于 Apache Camel 和 Apache ActiveMQ 等开源社区项目构建。红帽 Fuse 是敏捷集成解决方案的一部分。其分布式方法允许团队在需要时部署集成服务。以 API 为中心、基于容器的架构将服务解耦，因此可以独立创建、扩展和部署它们。

分布式云原生集成平台的优势在于：

- **混合部署**

红帽 ® Fuse 支持在多种环境中部署，无论是在本地、公共 / 私有云中，还是作为支持所有用户的托管服务，为不同的用户提供他们需要的集成解决方案。所有部署选项都可以互操作，因此集成专家、开发人员和业务用户可以相互协作。选项包括：

Red Hat Fuse - 独立的，部署在私有服务器上

Red Hat Fuse on Red Hat OpenShift® - 适用于公共或私有云

Red Hat Fuse Online - 为集成平台即服务 (iPaaS) 解决方案托管

- **具有低代码 UI/UX 的内置 iPaaS**

应用程序开发人员和业务用户可以使用拖放界面作为完整集成平台即服务 (iPaaS) 解决方案的一部分。红帽 Fuse 包括 Fuse ignite，这是一个基于浏览器的拖放界面，用于快速创建集成服务。Ignite 包含在 Fuse Online 中，为企业用户提供完整的 iPaaS 解决方案。Ignite 还可以与 Red Hat OpenShift 上的 Fuse 一起使用，为开发人员提供低代码工具。集成专家和开发人员还可以使用 Red Hat Developer Studio 创建需要更复杂功能的集成，并将这些集成添加到 Fuse ignite 服务中。

- **基于容器的集成**

作为云原生解决方案，Fuse 支持在容器中开发和连接微服务。这使应用程序开发人员能够不断创新。开发人员可以独立部署服务、API 和集成来支持他们的应用程序。

- **处处集成**

借助红帽 Fuse 的 200 多个连接器，用户几乎可以连接到所有对象，从旧系统到软件即服务 (SaaS) 应用程序，从应用程序编程接口 (API) 到物联网 (IoT) 设备。可以针对特定的端点连接要求编写自定义组件。使用企业集成模式 (EIP)（企业连接的标准）从端点轻松创建集成。

本方案选用的是在 OpenShift 上的红帽 Fuse 组件。对于想要简化 IT 基础设施的用户，Fuse 可以部署在 OpenShift 上，在其中自动配置和管理容器。这样 Fuse 就可以利用到 Openshift 容器平台的优势，您可以：

- 轻松部署容器原生集成并快速扩展。
- 部署在公共或私有云中。
- 使用与 Fuse 本地部署相同的技术基础。
- 使用基于 Fuse Online 浏览器的拖放式 UI/UX 快速创建和部署集成。

6. 方案特点

以红帽开源技术打造的应用服务集成层具有以下特点：

- 跨混合云的应用程序和数据集成。
- 使用 200 多个可插拔连接器部署基于企业集成模式 (EIP) 的集成，以跨混合云连接新数据和现有数据。
- API 对接和适配。
- 在 API 的整个生命周期中创建、部署、监控和控制 API。通过 API 优先的方法，跨混合和多云环境扩展您的集成。

- 容器原生基础设施。
- 以流行的容器标准开发和管理服务，以及在分布式环境中打包和部署轻量级容器，以轻松适应和快速扩展。
- 实时消息传递、变更数据捕获和数据流。
- 以高吞吐量和低延迟在应用程序和系统之间实时共享数据。
- 企业用户自助服务。
- 使非技术用户能够使用自助式低代码开发平台配置和部署集成和消息传递。
- 缩短上市时间。
- 通过自助服务平台和自动化软件开发生命周期工具为微服务开发人员和 DevOps 团队提供支持。

7. 关于 OpenShift 容器平台

7.1 OpenShift 容器平台概述

红帽 ® OpenShift 容器平台将开发人员和 IT 运维团队统一到了一个平台上，从而可以跨混合云和多云基础架构方便地构建、部署和管理应用。红帽 OpenShift 能够使企业在交付现代和传统应用的过程中，缩短开发周期，降低运营成本，从而取得更大收益。红帽 OpenShift 构建于开源创新和行业标准的基础上，包括 Kubernetes 和红帽企业 Linux®（世界领先的企业级 Linux 发行版）。

OpenShift 为 Web 应用程序的部署实施了多语言平台和服务。 它将容器与安全增强型 Linux（SELinux）环境结合使用，以实现适合企业的安全多租户环境。您可以在自己的基础设施或公共云中部署 OpenShift，也可以使用 Red Hat 的基于云的托管服务 OpenShift Online。

最新版本的 OpenShift 使用来自云原生计算基金会（CNCF）行业标准的 Kubernetes 平台，用于管理和运行容器内的应用程序。通过遵守开放容器倡议

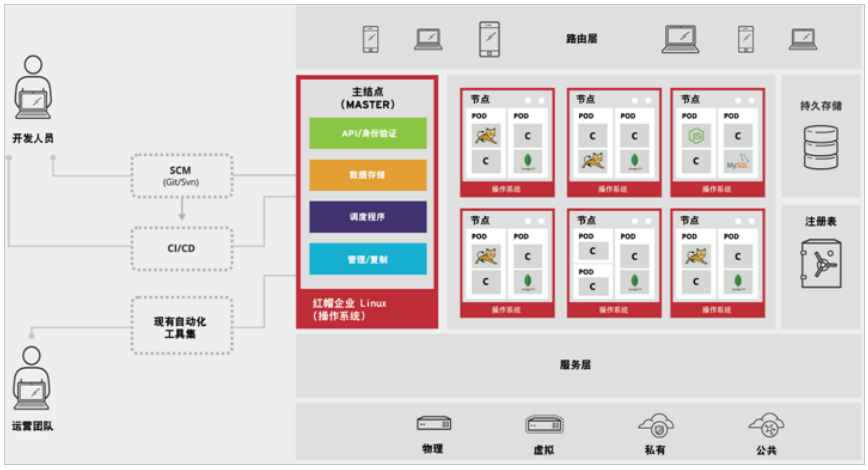
（OCI）中的镜像和运行时规范来确保运行任何应用程序镜像的能力。

OpenShift 为您提供了直接轻松部署 Web 应用程序代码的能力，使用预定义的镜像构建器库，或者您可以携带自己的容器镜像。通过在 OpenShift 中支持持久性卷等功能，您可以不仅限于运行无状态的满足 12 因素的微服务或云本原生用程序。使用 OpenShift，您还可以部署数据库和许多传统应用程序，否则这些应用程序无法在传统的平台即服务（PaaS）产品上运行。

OpenShift 是一个完整的容器应用程序平台。这是对现有应用程序可以使用的传统 PaaS 的现代化应用，但也提供了满足未来需求的功能和灵活性。

7.2 OpenShift 容器平台整体架构

红帽 ® OpenShift 是一个应用容器平台，能够帮助开发和 IT 运营团队实现现有企业应用的现代化改造，并通过加速开发和交付流程来交付各种新服务。它基于经过验证的开源技术构建而成，包括 Linux® 容器和 Kubernetes。Linux 容器可以封装应用，使应用与整个运行时环境隔离开。由于容器有利于明确划分职责范围，从而减少开发和运营团队间的冲突，所以是 DevOps 必不可少的要件之一。开源项目 Kubernetes 已入选云原生计算基金会 (CNCF)，并且成为了 Linux 容器编排领域的行业标准。



7.2.1 对于应用开发团队

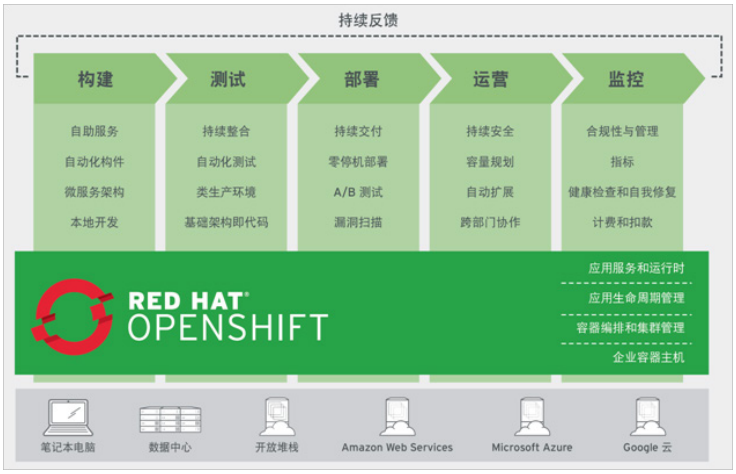
OpenShift 容器平台可为开发人员提供一个自助服务平台，便于置备、构建和部

署应用及其组件。开发人员可利用多种自动化工作流程（如我们的源至镜像 (S2I) 流程）轻松地将版本控制系统中的源代码植入可直接运行的 Docker 格式容器镜像中。OpenShift 容器平台集成了各种持续整合和持续交付（CI/CD）工具，是适用于所有企业机构的理想解决方案。

7.2.2 对于 IT 运维团队

OpenShift 容器平台为 IT 运维团队提供了安全的企业级 Kubernetes，可帮助实现基于策略的控制及应用管理自动化。通过集群服务、调度和编排，用户还可实现负载均衡和自动扩展。内置安全功能，可防止租户入侵其它应用或底层主机。同时，由于 OpenShift 容器平台可将持久存储直接与 Linux 容器连接，因此 IT 部门可在同一平台上运行有状态和无状态应用。

红帽 OpenShift 可为开发和运营团队提供一个通用平台和一组工具，以便在任意基础架构上（在企业内部或在公共云、私有云或混合云中）构建、部署和管理容器化应用。



所有这些功能都能为 DevOps 的多个不同领域和实践提供支持，从而加速创新。另外，由于红帽 OpenShift 可以实现自动化构建和部署功能、实现持续整合 / 持续交付 (CI/CD) 并提供相应的构件和容器指标，所以它能将构建和部署流程中的相关信息迅速提供给开发团队并持续提供各种反馈。这样，开发人员就能快速检测出异常并加以修正。这种方式的成效要远远优于在生产环节中进行修复，因为后者会增加成本并严重影响服务的交付。

7.2.3 OpenShift 容器平台的运维功能

7.2.3.1 运营

- 持续交付：借助内置的管道（可与现有工具相整合）支持功能，团队可以自动实施应用交付过程中的各个步骤，并充分利用现有流程。
- 零停机部署：团队可以借助滚动更新、蓝绿部署和 canary 发布，实现零停机部署，进而消除部署环节的停机时间，并在常规工作时段内频繁地在生产环境中进行部署。
- A/B 测试：对应用流量的全面控制能让团队同时为用户提供多个版本的服务。
- 漏洞扫描：红帽 OpenShift 容器平台包含红帽高级集群管理，后者能持续对容器镜像进行漏洞扫描，从而防止存在恶意安全问题的容器在您的基础架构上运行。

7.2.3.2 部署

- 持续安全：为经过红帽认证的容器提供主动型安全补丁，以便自动触发相关应用容器的重构和部署。
- 容量规划：红帽高级集群管理会跟踪资源的利用趋势，制定相应的容量和假设情景计划。
- 自动扩展：借助基于应用负载的自动扩展容器，自动启动在红帽 OpenShift 上运行的扩展应用。
- 跨部门协作：细粒度访问控制功能可以实现生产环境的可视化，并让运营团队保留对于所执行操作的控制权，从而促成开发、质量保证、安全和运营团队间的协同合作。

7.2.3.3 监控

- 合规性与管理：可在所有容器和环境内自动执行各项策略，支持全面的业务分析功能和详细日志记录。
- 指标：借助容器指标，可以全面了解应用资源使用的长期变化情况。

- 健康检查和自我修复：健康探测器可以自动识别应用存在的问题，便于进行快速修复。
- 计费和扣款：管理组件可以收集容器容量和利用率数据，生成财务报告，以展示各个团队的容器使用情况。

7.2.4 OpenShift 容器平台的开发功能

7.2.4.1 构建

- 自助服务：开发人员可以根据需要，使用他们所偏好的工具轻松快速地创建应用，而无需等待 IT 运营团队设置所需的部署环境。与此同时，运营团队仍能全面控制整个环境。
- 自动化构件：借助经过简化的自动化应用构件，开发人员能以可重复的安全方式，利用应用源代码和二进制文件自动构建容器。
- 微服务：红帽 OpenShift 应用运行时可以提供多种经过认证且受支持的微服务运行时，包括 OpenShift 中可用于构建云原生应用的 Spring Boot、WildFly Swarm、Vert.x 和 Node.js，以及内置的服务发现、负载平衡、单点登录等支持功能。
- 本地开发：使用测试和生产环节中所用的工具在本地开发和部署各种应用。

7.2.4.2 测试

- 持续整合 (CI)：内置的 Jenkins CI 服务器支持功能可以帮助开发人员针对每一次变化自动编写、测试和整合相应的代码。
- 自动化测试：按需部署功能可以根据需要置备和测试具有完整依赖项的应用，以实施复杂的自动化测试方案。
- 类生产环境：不管是本地开发环境还是生产环境，红帽 OpenShift 都可以为它们提供相同的技术堆栈，确保基于完全相同的中间件、语言运行时和操作系统版本来测试和验证应用。
- 基础架构即代码：与应用和环境相关的各个方面均会以声明性的方式来加以描述，以便以代码的形式在版本控制系统中进行版本管理和控制。

7.2.5 红帽 OpenShift 容器平台的优势

7.2.5.1 开源标准

采用开放容器计划（OCI）/Docker 格式的容器以及 Kubernetes 进行容器编排，外加其它开源技术。用户将免受特定供应商的技术或业务路线图的限制。

7.2.5.2 自助服务置备

开发人员可使用最顺手的工具，轻松、快速地按需创建各种应用，同时运维团队也能全面掌控整个环境。

7.2.5.3 持久存储

OpenShift 容器平台支持持久存储，允许用户同时运行有状态的应用和无状态的云原生应用。

7.2.5.4 多语言支持

开发人员可轻松地在同一平台使用多种语言、框架和数据库。

7.2.5.5 自动化

OpenShift 容器平台自带多种功能，包括精简且自动化的容器和应用构建、部署、扩展、运行状况管理等。

7.2.5.6 用户界面

开发人员可直接访问多种命令行工具、多设备 Web 控制台和基于 Eclipse 的整合开发环境（IDE），如红帽 JBoss® 开发人员工作室。

7.2.5.7 运维管理

OpenShift 容器平台附带了红帽高级集群管理和高级安全管理组件，可让用户实时了解自身的容器化应用和基础架构。

7.2.5.8 可扩展性

在 OpenShift 容器平台上运行的应用可在数秒内轻松地扩展到数百个节点上的数千个实例中。

7.2.5.9 强大的生态系统

不断发展壮大的红帽合作伙伴生态系统，旨在为用户提供广泛多样集成。系统中的第三方合作伙伴可提供额外的存储和网络提供商、 IDE 和 CI 整合、独立软件供应商（ISV）解决方案等，与 OpenShift 容器平台搭配使用，让您更加得心应手。

7.2.5.10 容器可移植性

基于红帽应用编程接口（API）支持的标准化 Linux 容器模型构建，让创建于 OpenShift 容器平台的应用能在支持 Docker 格式容器的任意环境中轻松运行。

8. 应用场景

8.1 API 安全防护

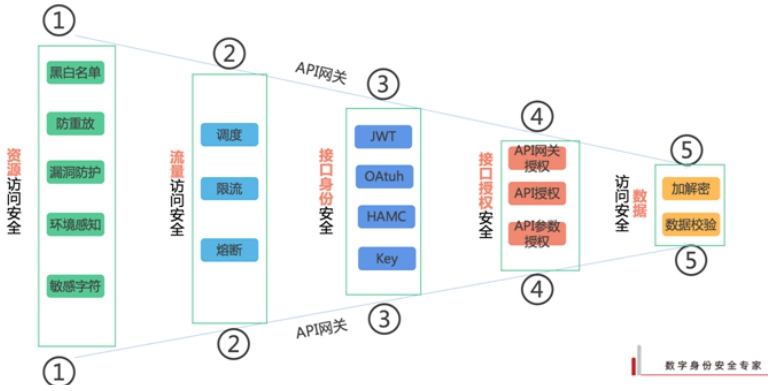
API 身份认证、安全防护、日志审计、数据加解密、SDK 全方位保护 API 安全，防止 API 数据泄露、API 非法调用的风险。

在资源访问安全层面，可以通过黑白名单、防重放、漏洞防护等策略来保护接口的访问安全。

对于流量层面，可以使用调度、限流、熔断等方式对后端服务进行保护。

接口身份安全可以采用 OAuth，JWT,HMAC 等方式对消费者进行基于身份的权限管控。

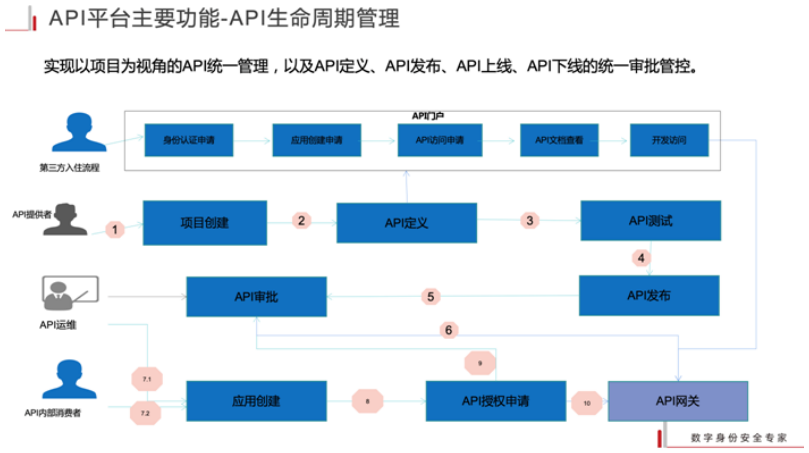
对接口可以进行细粒度的授权，另外可以通过数据加解密和数据加密来保证数据的访问安全。



8.2 API 全生命周期管理

对 API 接口进行生命周期管理，基于关键字搜索功能快速分组，并且对分组后的 API 资产指定责任人，实现 API 资产的导入和导出、资产上下线。

在项目维度对 API 进行管理，API 的全生命流程都可以引入对应的审批流程，来保证项目接口上下线的规范和有序化。



8.3 协议转换和数据能力外放

服务注册接入本身分为两个层面，一个是已有服务的注册接入，一个是需要适配后的服务发布。在设计的时候需要考虑到两个方面的需求。

对于已有服务的存代理接入最简单，即只需要提供业务系统的 Rest 接口服务地址即可，在接入的时候，对相关的日志，安全，流控，负载均衡等策略进行配置，配置完成后即完成服务接入和注册。同时对于路由服务接入需要单独考虑，对于路由服务在接入的时候可以适配到多个原始业务系统的接口服务地址。

服务发布是对原来我们服务适配功能的一个改进，即直接从底向上的进行服务发布，而不需要实现定义服务元数据或模型，制定服务契约格式等，在服务发布完成后再生成相关的基础数据到服务元数据库即可。对于服务发布参考服务适配的能力，我们可以考虑如下场景下的需求。

- 将一个已有的 SOAP WS 服务发布和注册为一个 Http Rest 接口服务。
- 将一个数据库表，或存储过程发布为一个 Http Rest 接口服务。

- 将一个 JMS 消息接口发布为一个 Http Rest 接口服务。
- 将一个 JAR 包中的 API 接口方法或函数发布为一个 Http Rest 接口服务。

相对于其他网关产品来说，派拉 API 网关平台结合红帽 Fuse 之后，进一步加强了服务适配的能力。

以将数据库或者存储过程的 API 转换为例，派拉 API 网关平台可以快速轻松连接所有常见的数据库类型，只需要配置相对应的数据源即可以将生成对应的 API，且可以轻松对数据字段进行调整，完成数据沉淀和数据能力外放的转换。

8.4 服务组合编排

服务组合编排是服务组合，服务组装等，希望通过服务编排能够完成这些事情，而不是简单的完成单一服务的设计和开发。即将多个原子服务组合或组装在一起，最终形成一个新的服务并提供的能力。我们举例来说明下。

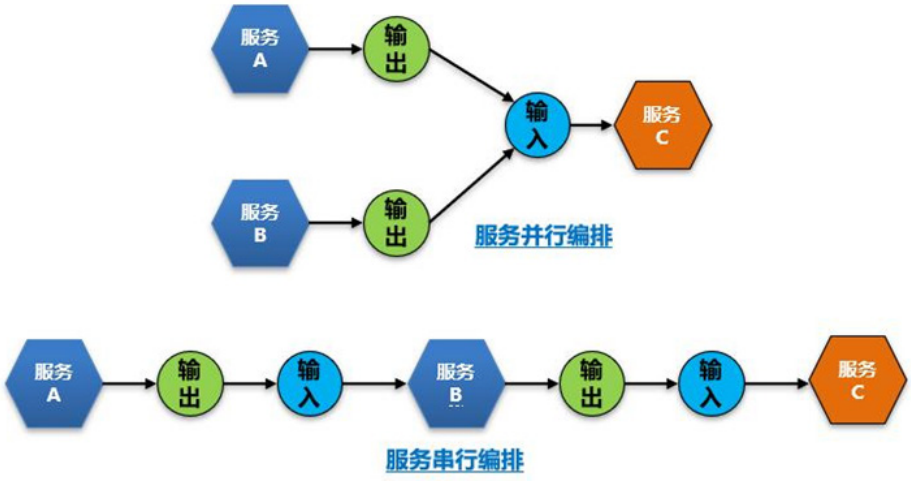
比如存在 A，B，C 三个原子服务，我们通过服务编排形成一个新的 D 服务。

三个原子服务全部是查询服务，希望组装一个新服务，一次返回 A，B，C 三个服务查询结果

这个即我们说的服务组合能力，比如我们可以对合同基本信息查询，合同条款信息查询，合同执行信息查询三个基本原子服务进行组合，最终返回一个服务综合信息查询的服务，一次返回三个查询结果。

对多个原子服务进行流程式的前后串接并形成服务编排。

这个是我们经常看到的一种服务编排场景，即 A，B，C 三个服务直接进行编排，即 A 服务的输出直接变为 B 服务的输入，B 服务的输出又变为 C 服务的输出。



9. 案例分享

9.1 背景

某汽车有限公司，历经多年的发展，不仅缔造了新的行业纪录，更成为中国成长最快的乘用车领军企业。随着计算机网络的不断发展，企业网站作为面向客户以及合作伙伴的重要窗口，对企业形象的营造起着重要作用，对企业发展提供重要助力，是企业价值展现的重要舞台。

目前，该公司旗下有 2 个主要品牌，涉及经销商网站共 30 个，其中 13 个网站服务器位置位于企业外部。此外，有 2 家员工网站也位于企业外部，有待整合。

9.2 痛点

- 随着业务快速发展，业务系统建设更加敏捷，系统越来越多（300+）、系统间集成的数量越来越多（4700+）、而复杂度也越来越高。
- 随着营销从面相经销商到下探到直接面相用户，2C 应用更多、场景更丰富，前

端 Web、App 需要更加快速迭代，而后端应用服务则需要更多适配和兼容丰富前端渠道。

- 后端服务暴露出 API 的安全管控级别不同，没有企业级标准化。
- Aliyun、Openstack 和 OA 网络间应用调用防火墙复杂，开通过程沟通成本高。
- API 众多，没有统一管理平台，造成管理复杂度高，成本浪费。
- 面向 API 的开发编程过程，没有开放门户这样的开发者共享平台，信息交流不畅。
- 内部接口在对外暴露后无法根据服务能力进行动态调整服务等级。

9.3 目标

- 为公司混合云架构提供统一的企业级 API 网关，从而极大减小数据中心之间应用通讯的复杂度，加快 API 提供方和消费方与企业 IT 网络管理方的沟通速度。
- 为微服务 API、集成（ESB/PSB/PI）API 以及点对点 API 提供统一策略的 API 安全访问、认证鉴权服务，快速将后台业务推向前台。
- 通过 API 开放门户，保持 API 开发者各方在同一上下文沟通，简化开发者交流流程，提高面向 API 开发的效率和质量。
- 通过内部 API 外部化（开放），外部 API 内部化（引进）的双线 API 策略，快速构建 API 生态系统，加速数字化转型。

9.4 方案

9.4.1 整体要求

方案需至少满足如下要求：

- 对混合云架构下的多云 API 能进行统一管理。
- 能对开发这赋能，简化 API 开发流程，提高 API 沟通效率。
- 整合内部和外部 API，完善 API 生态，为业务创新赋能。
- 操作简单易用，后期规则、内容与模块调整和维护灵活。

- 项目实施服务要保证质量，能在规定时间内解决相关业务部门在日常使用过程中发现的问题。
- 满足存档需求。

9.4.2 统一 API 平台

9.4.2.1 API 网关

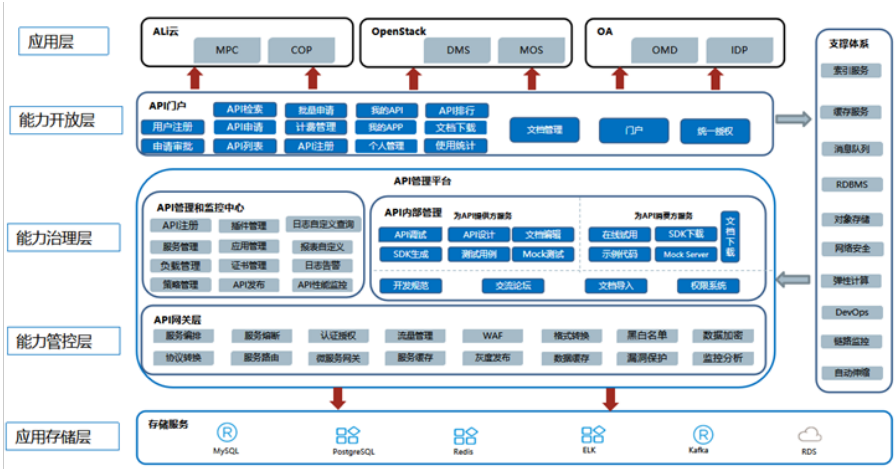
在混合架构下的三个网络区域（Aliyun、OpenStack 和 OA）建设企业级 API 网关，覆盖微服务和传统应用服务 API，为 API 提供统一的调用策略、安全策略和流量控制策略。

9.4.2.2 API 开放门户

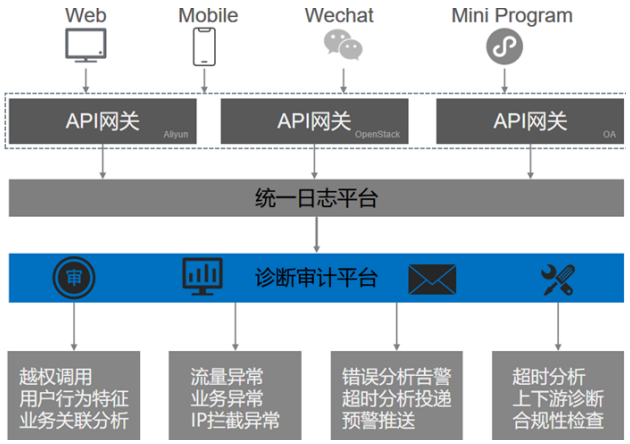
基于场景化的业务分组，对 CIP、CIM、CID 以及 SX 和 SC 的业务进行不同程度的能力开发门户的 API、对通用型的 API 如：会员信息、车牌信息、积分等一站式服务，避免不同由于信息无法共享导致的重复上架。开发人员可以提前检索 API、试用 API，申请 API 等。

9.4.2.3 API 管理平台

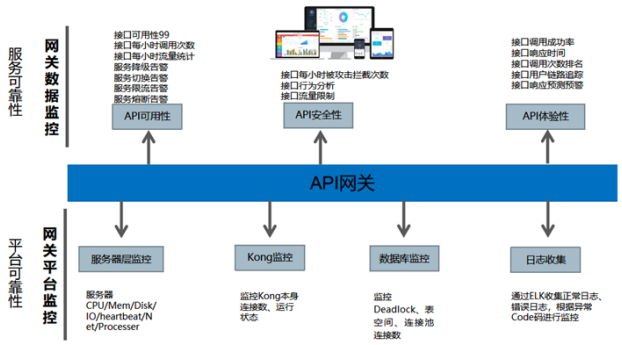
对 API 网关和 API 开放门户进行后台管理，网关 API 的上下架、开放门户的上下架、网关状态的监控等



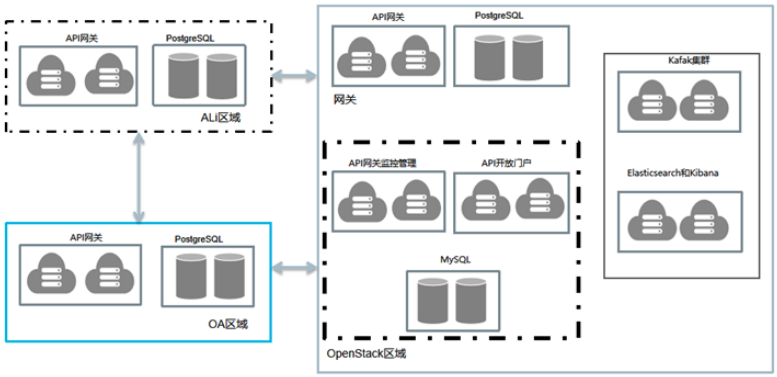
9.4.3 API 诊断审计



9.4.4 监控告警



9.4.5 网络拓扑



10. 关于上海派拉软件

10.1 公司简介

上海派拉软件股份有限公司（简称：派拉软件）成立于 2008 年，致力于为企业单位和政府机构提供专业的一体化零信任身份安全产品和服务，在上海、北京、广州、武汉、成都、长春、深圳设有研发中心和服务机构。

派拉软件将软件定义边界、持续自适应、微隔离等信息安全前沿技术导入身份管理产品的研发与实践中，为各个行业客户提供专业的一体化零信任身份安全解决方案，覆盖内部员工身份治理 (2E)、外部合作伙伴身份治理 (2P)、C 端客户身份治理 (2C)、API 身份治理 (2API)、IoT 身份治理 (2IoT)、云身份治理、特权身份管理。

10.2 发展历程

2008 年 派拉软件股份有限公司成立

2009 年 发布 ParaSecure OSC 安全运维管理软件

2010 年 派拉布局华北区域，成立北京分公司

2011 年 发布 ParaSecure ESC 统一身份管理与安全认证软件

2015 年 派拉布局华南区域， 成立广州分公司

2017 年 布局华中及东北区域， 成立武汉、长春分公司

2018 年 获得上海市科技小巨人企业称号

2019 年 成立深圳、成都分公司

2020 年 引入 C 轮 3 亿元融资，持续发力零信任安全

10.3 公司产品

零信任：软件定义边界、零信任身份管理、用户行为分析、企业安全网关、动态授权平台

身份安全 ： 派拉统一身份管理软件、派拉单点登录软件、派拉 IDaaS 平台、多因素认证平台 MFA、API 身份管理 物联网身份管理、消费者身份管理、细粒度权

限管理

运维安全：特权账号管理平台、日志审计平台、弱密码扫描、堡垒机

数据安全：API 安全网关平台、可信数据交换平台 ESB、大数据安全管理、数据湖

10.4 公司主页

https://www.paraview.cn/

11. 关于红帽公司

11.1 公司简介

红帽将协助为您的 IT 未来奠定更好的基础。我们使用 Red Hat® Enterprise Linux® 彻底改变了操作系统。现在，我们拥有广泛的产品组合，包括混合云基础架构、中间件、敏捷集成、云原生应用程序开发以及管理和自动化解决方案。

红帽提供强化的开源解决方案，使企业能够更轻松地跨平台和跨环境工作，从核心数据中心到网络边缘。通过透明和负责任的运营，我们将继续成为开源社区的催化剂，帮助您构建灵活、强大的 IT 基础架构解决方案。开放源码在过去、现在和未来将持续推动创新。这是世界需要的创新。这种力量超越了数据中心和新兴技术，并将创新掌握在每个人的手中。

红帽成立于 1993 年，在过去 25 年中，我们不断帮助客户应对业务挑战。超过 90% 的财富 500 强公司信赖我们，我们在 40 个国家的 100 多个地区为您服务。2012 年，红帽成为第一家收入超过 10 亿美元的开源技术公司。2019 年，IBM 以约 340 亿美元收购 Red Hat，这是历史上最大的软件收购。与 IBM 联手使红帽能够加强其现有的合作伙伴关系，为客户提供自由、选择和灵活性。

红帽是来自开源领域的领导者，现已成为 IT 领域的领导者。我们的开源解决方案适用于世界上要求最严苛的数据中心和云堆栈。如今红帽不断在持续构建混合云、开发云原生应用和 IT 自动化方面助您一臂之力。

我们相信开放混合云的力量。基于专有技术的独立云部署阻止了云之间的交互。

开放式混合云战略为混合企业环境带来开源软件的互操作性、工作负载可移植性和灵活性。

Red Hat 是 Linux 内核等开源社区项目的主要贡献者之一。红帽工程师帮助改进功能、可靠性和安全性，以确保您的基础架构运行并保持稳定——无论您的场景和工作负载如何。

精英管理、社区建设和透明度等开源价值观正在改变世界处理商业和生活的方式。红帽提供的工具、原则和标准为灵活性和创新奠定了基础。

11.2 发展历程

https://www.redhat.com/en/about/brand/standards/history