# ONAP Architecture Overview

# ONAP 架构概述

## 1. Introduction 概述

The ONAP project addresses a rising need for a common platform for telecommunication, cable, and cloud operators--and their solution providers--to deliver differentiated network services on demand, profitably and competitively, while leveraging existing investments.

随着电信运营商、有线运营商、云业务运营商以及他们的方案提供商对通用平台需求的增加，ONAP 项目应时而生，并致力于在充分利用现有投资的前提下，提供按需定制、有竞争力的、差异化的网络服务。

Prior to ONAP, operators of large networks have been challenged to keep up with the scale and cost of manual changes required to implement new service offerings, from installing new data center equipment to, in some cases, upgrading on-premises customer equipment. Many are seeking to exploit SDN and NFV to improve service velocity, simplify equipment interoperability and integration, and reduce overall CapEx and OpEx costs. In addition, the current, highly fragmented management landscape makes it difficult to monitor and guarantee service-level agreements (SLAs).

在 ONAP 诞生之前，大型网络运营商为了提供新的业务，在从安装新的数据中心设备到（某些情况下）升级客户现场设备等一系列工作中，需要执行大量的人工调整工作。这种人工模式的规模和成本均对运营商提出了重大的挑战。许多运营商都在寻求利用 SDN 和 NFV 技术，提高业务创新速度，简化设备的互操作性和集成难度，降低整体的资产投入和运营成本。另外，目前高

度分散的管理场景也使得端到端级别的业务质量难以得到监控和保障。

ONAP is addressing these problems by developing global and massive scale (multi-site and multi-VIM) orchestration capabilities for both physical and virtual network elements. It facilitates service agility by providing a common set of Northbound REST APIs that are open and interoperable, and by supporting YANG and TOSCA data models. ONAP's modular and layered nature improves interoperability and simplifies integration, allowing it to support multiple VNF environments by integrating with multiple VIMs, VNFMs, SDN Controllers, and even legacy equipment. ONAP's consolidated VNF requirements publication will enable commercial development of ONAP-compliant VNFs. This approach allows network and cloud operators to optimize their physical and virtual infrastructure for cost and performance; at the same time, ONAP's use of standard models reduces integration and deployment costs of heterogeneous equipment, while minimizing management fragmentation.

ONAP 通过为物理和虚拟网络设备提供全局的和大规模（多站点和多 VIM）的编排功能来解决这些问题。它通过提供一套通用的、开放的、可互操作的北向 REST 接口，以及支持 YANG 和 TOSCA 数据模型来提高业务敏捷性。ONAP 的模块化和分层特性有助于提高互操作性并简化集成过程，它可以能够与多个 VIM、VNFM、SDN 控制器甚至传统网络设备的集成来支持多个 VNF 的环境。ONAP 对 VNF 的整体要求发布将助力符合 ONAP 标准的 VNF 的商业部署。这样既可以帮助网络和云业务运营商优化他们的物理和虚拟基础设施，以降低成本、提高性能；同时，ONAP 采用标准模型，降低了异构设备的集成和部署成本，同时最大限度地减少了管理的碎片化。

The ONAP platform allows end user organizations and their network/cloud providers to collaboratively instantiate network elements and services in a dynamic, closed-loop process, with real-time response to actionable events. In order to design, engineer, plan, bill and assure these dynamic services, there are three major requirements:

在 ONAP 平台上，终端用户组织和他们的网络/云业务提供商可以在一个动态、闭环过程中进行协作，实例化网络设备和业务，并对操作类事件进行实时响应。为了设计、实施、规划、计费和保障这些动态业务，主要有三方面的要求：

- A robust design framework that allows specification of the service in all aspects – modeling the resources and relationships that make up the service, specifying the policy rules that guide the service behavior, specifying the applications, analytics and closed-loop events needed for the elastic management of the service
- 一个健壮的设计框架，可以在各个方面对业务进行规范，包括：对组成业务的各类资源和

关系进行建模，制定指导业务行为的策略规则，制定业务弹性管理所需的应用、分析和闭环事件。

- An orchestration and control framework (Service Orchestrator and Controllers) that is recipe/policy-driven to provide automated instantiation of the service when needed and managing service demands in an elastic manner
- 一个流程/策略驱动的编排和控制框架（业务编排器和控制器），在必要时提供自动的业务实例化，并能够弹性管理业务需求。
- An analytic framework that closely monitors the service behavior during the service lifecycle based on the specified design, analytics and policies to enable response as required from the control framework, to deal with situations ranging from those that require healing to those that require scaling of the resources to elastically adjust to demand variations.
- 一个分析框架，可以根据指定的设计、分析和策略，密切监控整个业务生命周期中的行为，实现控制框架所要求的响应，从而可以对从设备自愈到根据需求变化对资源进行扩缩容调整等各种情况进行处理。

To achieve this, ONAP decouples the details of specific services and technologies from the common information models, core orchestration platform and generic management engines (for discovery, provisioning, assurance etc). Furthermore, it marries the speed and style of a DevOps/NetOps approach with the formal models and processes operators require to introduce new services and technologies. It leverages cloud-native technologies including Kubernetes to manage and rapidly deploy the ONAP platform and related components. This is in stark contrast to traditional OSS/Management software platform architectures, which hardcoded services and technologies, and required lengthy software development and integration cycles to incorporate changes.

为此，ONAP 将特定业务和技术细节从通信信息模型、核心编排平台和通用管理引擎（用于发现、配置和保障等）中分离出来。此外，它将 DevOps/NetOps 方法的效率和模式与运营商引入新业务和技术所需求的正式模型和过程相结合。它利用包括 Kubernetes 在内的云原生技术来管理和快速部署 ONAP 平台及相关组件。传统的 OSS/管理软件平台架构会对业务和技术进行硬编码，在整合变化时需要很长的软件开发和集成周期，ONAP 与之形成鲜明的对比。

The ONAP Platform enables product/service independent capabilities for design, creation and lifecycle management, in accordance with the following foundational principles:

ONAP 平台根据以下基本原则，支持产品/业务的独立的设计、创建和生命周期管理：

- Ability to dynamically introduce full service lifecycle orchestration (design, provisioning and operation) and service API for new services & technologies without the need for new platform software releases or without affecting operations for the existing services
- 在不需要平台软件新版本发布和影响现有业务操作的情况下，可以为新业务动态进行全生命周期编排（包括设计、配置和运营）和业务 API 部署；
- Carrier-grade scalability including horizontal scaling (linear scale-out) and distribution to support large number of services and large networks
- 电信级的可扩展性，包括横向扩展（线性扩容）和分发，以支持大量的业务和大规模网络；
- Metadata-driven and policy-driven architecture to ensure flexible and automated ways in which capabilities are used and delivered
- 元数据驱动和策略驱动的架构，以确保使用和发布功能的灵活性和自动化；
- The architecture shall enable sourcing best-in-class components
- 架构应该支持采用最好的组件；
- Common capabilities are 'developed' once and 'used' many times
- 常用功能只需要一次开发，多次复用；
- Core capabilities shall support many diverse services and infrastructures
- 核心功能应支持多种不同的业务和基础设施；
- The architecture shall support elastic scaling as needs grow or shrink
- 随着需求增长或收缩，架构应支持弹性扩展。

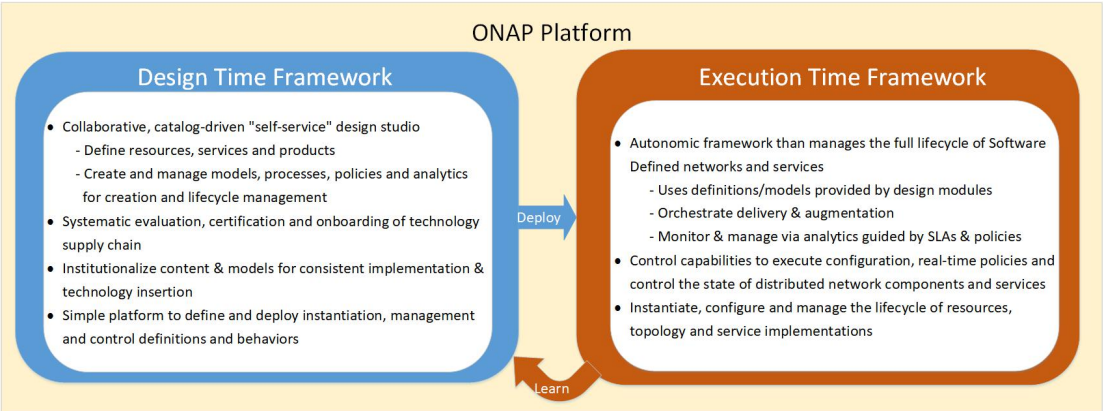


Figure 1: ONAP Platform

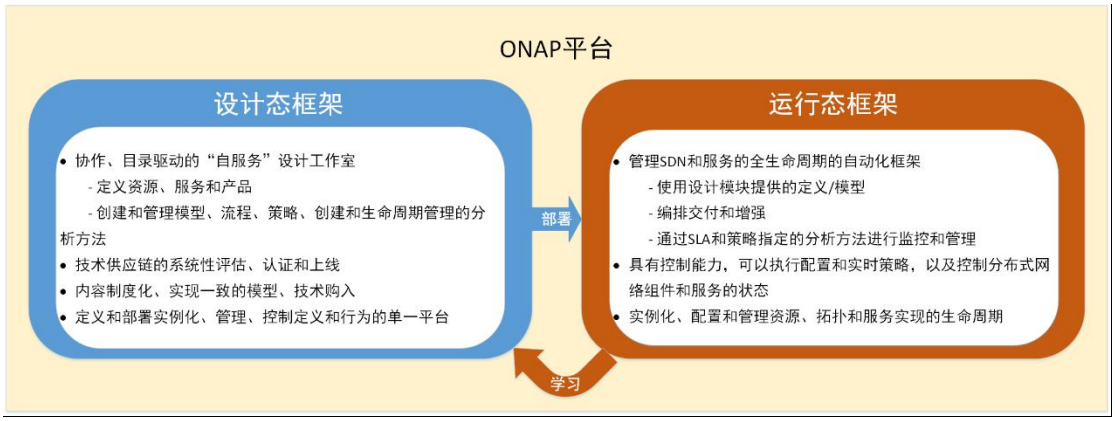


图 1: ONAP 平台

# 2. ONAP Architecture ONAP 架构

The platform provides the common functions (e.g., data collection, control loops, meta-data recipe creation, policy/recipe distribution, etc.) necessary to construct specific behaviors.

ONAP 平台提供了构建特定行为所需的通用功能(例如：数据采集、闭环控制、元数据流程创建、策略/流程分发等)。

To create a service or operational capability, it is necessary to develop service/operations-specific service definitions, data collection, analytics, and policies (including recipes for corrective/remedial action) using the ONAP Design Framework Portal.

当创建一项业务或运营能力时，需要利用 ONAP 设计框架的 Portal 模块来开发业务/运营特定的业务定义、数据采集、分析方法和策略（包括纠正/补救措施的流程)。

Figure 2 provides a high-level view of the ONAP architecture and microservices-based platform components.

图 2 展示的是 ONAP 架构和基于微服务的平台组件的高层次视图。

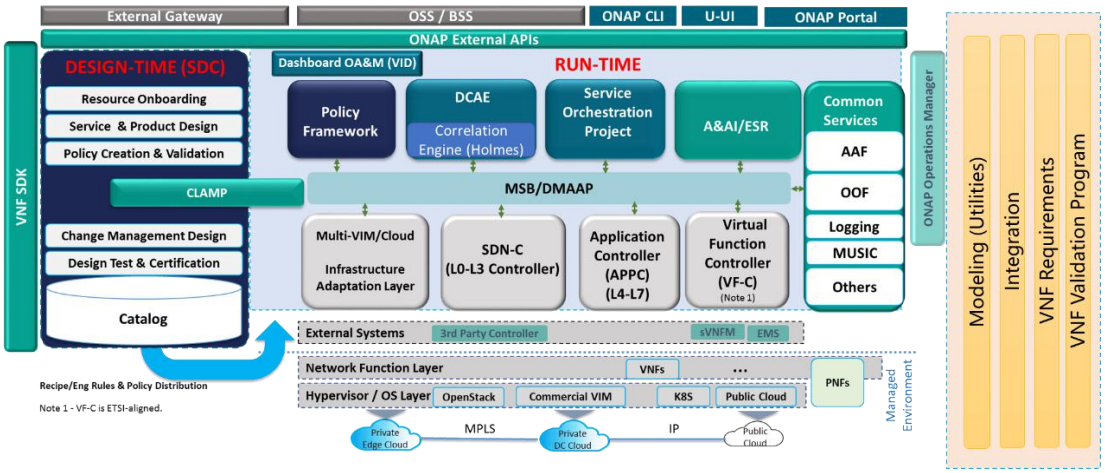


图 2: ONAP 平台架构（北京版本）

In Figure 3 below, we provide a functional view of the architecture, which highlights the role of key new components:

在下面图 3 中，我们提供了 ONAP 架构的功能视图，突出新的关键组件的作用：

1. The Beijing release standardizes and improves northbound interoperability for the ONAP Platform using the External API component.

1.利用外部 API 组件，北京版本对 ONAP 平台的北向接口进行了标准化并改进其互操作性。

2. OOM provides the ability to manage cloud-native installation and deployments to Kubernetes managed cloud environments.

2. OOM 提供了 Kubernetes 托管云环境下管理云原生安装和部署的能力

3. ONAP Common Services now manage more complex and optimized topologies. MUSIC allows ONAP to scale to multi-site environments to support global scale infrastructure requirements. The ONAP Optimization Framework (OOF) provides a declarative, policy-driven approach for creating and running optimization applications like Homing/Placement, and Change Management Scheduling Optimization.

3. 现在 ONAP 的通用服务可以管理更加复杂和优化的拓扑。MUSIC 允许 ONAP 扩展到多站点环境，以支持全局规模的基础架构需求。OOF 提供一种声明性的、策略驱动的方法用于创建和运行优化应用，如归属/位置、变更管理调度优化等。

4. Information Model and framework utilities have evolved to harmonize the topology, workflow, and policy models from a number of SDOs including ETSI NFV MANO, TM Forum SID, ONF Core, OASIS TOSCA, IETF and MEF.

4. 信息模型和框架实用程序已经可以协同众多标准化组织的拓扑、工作流和策略模型，包括 ETSI NFV MANO、TM Forum SID、ONF Core、OASIS TOSCA、IETF 和 MEF 等。

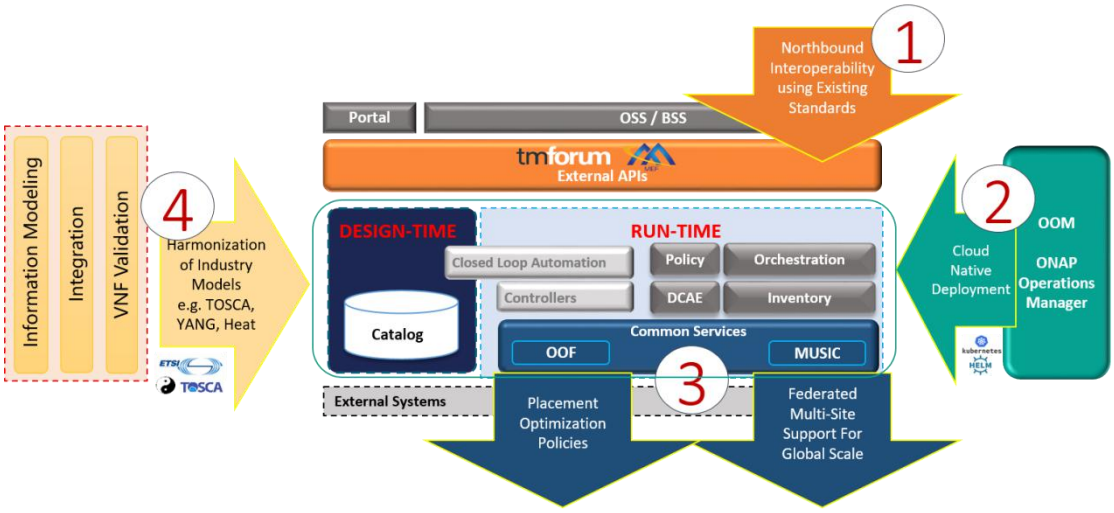


图 3: ONAP 架构的功能视图

# 3. Microservices Support 微服务支持

As a cloud-native application that consists of numerous services, ONAP requires sophisticated initial deployment as well as post-deployment management.

作为由大量业务组成的云原生应用，ONAP 的初始部署和运营管理十分复杂

The ONAP deployment methodology needs to be flexible enough to suit the different scenarios and purposes for various operator environments. Users may also want to select a portion of the ONAP components to integrate into their own systems. And the platform needs to be highly reliable, scalable, secure and easy to manage. To achieve all these goals, ONAP is designed as a microservices-based system, with all components released as Docker containers.

ONAP 的部署方法应足够灵活，以适应各种运营环境的不同场景和目标。用户可能希望只选择部分 ONAP 组件集成到他们自己的系统中。同时，ONAP 平台应该是高可靠、可扩展、安全且易于管理的。为了实现这些目标，ONAP 被设计为基于微服务的系统，其所有组件都通过 Docker 容器发布。

The ONAP Operations Manager (OOM) is responsible for orchestrating the end-to-end lifecycle management and monitoring of ONAP components. OOM uses Kubernetes to provide CPU efficiency and platform deployment. In addition, OOM helps enhance ONAP platform maturity by providing scalability and resiliency enhancements to the components it manages.

OOM 负责协调端到端的生命周期管理和 ONAP 组件的监控。OOM 通过 Kubernetes 来提高 CPU 利用率并提供平台部署方法。另外，通过增强其管理组件的可扩展性和弹性，OOM 有助于提升 ONAP 平台的成熟度。


OOM is the lifecycle manager of the ONAP platform and uses the Kubernetes container management system and Consul to provide the following functionality:

OOM 是 ONAP 平台的生命周期管理器，它利用 Kubernetes 容器管理系统和 Consul 提供以下功能：

**1. Deployment** - with built-in component dependency management (including multiple clusters, federated deployments across sites, and anti-affinity rules)

**1. 部署** - 内置组件的依赖管理（包括多集群，跨站点的联合部署和反亲和性规则）

**2. Configuration** - unified configuration across all ONAP components

**2. 配置** - 所有 ONAP 组件的统一配置

**3. Monitoring** - real-time health monitoring feeding to a Consul GUI and Kubernetes

**3. 监控** - 实时的健康监控，将监控的数据提供给 Consul GUI 和 Kubernetes

**4. Restart** - failed ONAP components are restarted automatically

**4. 重启** - 启动失败的 ONAP 组件将自动重启

**5. Clustering and Scaling** - cluster ONAP services to enable seamless scaling

**5. 集群和扩展** - 集群化的 ONAP 业务可实现无缝扩展

**6. Upgrade** - change out containers or configuration with little or no service impact

**6. 升级** - 在轻微或不影响业务的情况下更换容器或配置

**7. Deletion** - clean up individual containers or entire deployments

**7. 删除** - 清除单个容器或整套部署

OOM supports a wide variety of cloud infrastructures to suit your individual requirements.

OOM 支持大量云基础架构，以满足您的个性化需求。

OOM is integrated with the Microservices Bus (MSB) component project, which provides fundamental microservices support such as service registration/discovery, external API gateway, internal API gateway, client software development kit (SDK), and Swagger SDK. MSB also supports OpenStack (Heat) and bare metal deployment.

OOM 与 MSB 组件项目集成。MSB 提供基本的微服务支持，例如服务注册/发现、外部 API 网关、内部 API 网关、客户端 SDK 和 Swagger SDK。MSB 还支持 OpenStack（Heat）和裸机部署。

# 4. Portal

ONAP delivers a single, consistent user experience to both design-time and run-time environments, based on the user's role. Role changes are configured within a single ONAP instance.

基于用户角色，ONAP 可以在设计态和运行态两种环境下提供统一一致的用户体验。在单个 ONAP 实例中可以对角色进行修改定制。

This user experience is managed by the ONAP Portal, which provides access to design, analytics and operational control/administration functions via a shared, role-based menu or dashboard. The portal architecture provides web-based capabilities such as application onboarding and management, centralized access management, and dashboards, as well as hosted application widgets.

ONAP 的 Portal 对用户体验进行管理，它通过共享的、基于角色的菜单或仪表盘，提了设计、分析和运营控制/管理功能的操作入口。Portal 架构提供了基于 Web 的各种能力，包括应用的上线和管理、集中访问控制、仪表盘和托管应用程序小部件等。

The portal provides an SDK to enable multiple development teams to adhere to consistent UI development requirements by taking advantage of built-in capabilities (Services/ API/ UI controls),

tools and technologies. ONAP also provides a Command Line Interface (CLI) for operators who require it (e.g., to integrate with their scripting environment). ONAP SDKs enable operations/security, third parties (e.g., vendors and consultants), and other experts to continually define/redefine new collection, analytics, and policies (including recipes for corrective/remedial action) using the ONAP Design Framework Portal.

Portal 提供了 SDK，可以使多个开发团队利用内置模块（服务/API/UI 控件）、工具和技术满足其一致的 UI 开发需求。ONAP 也为部分操作人员提供所需（例如：当他们需要与他们的脚本环境集成时）的命令行界面（CLI）。ONAP SDKs 可以使运营/安全、第三方（例如：供应商和顾问）和其它领域的专家不断地定义/优化新的采集、分析方法和策略（包括纠正和补救行为的策略），他们通过使用 ONAP 设计框架的 Portal 就可以完成这些工作。

# 5. Design-Time Framework 设计态框架

The design-time framework is a comprehensive development environment with tools, techniques, and repositories for defining/ describing resources, services, and products.

设计态框架是一个全面的开发环境，它包括了各种工具、技术以及定义/描述资源、服务和产品的资源库。

The design time framework facilitates reuse of models, further improving efficiency as more and more models become available. Resources, services, products, and their management and control functions can all be modeled using a common set of specifications and policies (e.g., rule sets) for controlling behavior and process execution. Process specifications automatically sequence instantiation, delivery and lifecycle management for resources, services, products and the ONAP platform components themselves. Certain process specifications (i.e., ‘recipes’) and policies are geographically distributed to optimize performance and maximize autonomous behavior in federated cloud environments.

设计态框架有助于模型的复用，随着可用模型的增大，可以更进一步地提高效率。资源、业务和产品，以及它们的管理和控制功能都可以使用一套用于控制行为和流程执行的规范和策略（例如规则集）进行建模。流程规范可以自动顺序完成资源、业务、产品和 ONAP 平台组件的实例化、发布和生命周期管理。某些特定的流程规范和策略可以根据地理位置进行分发部署，从而提升联合云环境下的性能优化和自动化程度。

Service Design and Creation (SDC) provides tools, techniques, and repositories to define/simulate/certify system assets as well as their associated processes and policies. Each asset is categorized into one of four asset groups: Resource, Services, Products, or Offers.

SDC 提供定义/模拟/验证系统资产及其相关过程和策略所需的工具、技术和存储库。每一项资产都可以归到资源、业务、产品、要约等四类的一种。

The SDC environment supports diverse users via common services and utilities. Using the design studio, product and service designers onboard/extend/retire resources, services and products. Operations, Engineers, Customer Experience Managers, and Security Experts create workflows, policies and methods to implement Closed Loop Automation/Control and manage elastic scalability.

SDC 环境通过通用服务和实用程序支持不同的用户。通过使用设计工作室，产品和业务的设计人员可以上线/扩展/下线资源、业务和产品。操作人员、工程师、客户体验经理和安全专家创建工作流、策略和方法，来实现闭环自动化/控制和管理弹性扩展能力。

To support and encourage a healthy VNF ecosystem, ONAP provides a set of VNF packaging and validation tools in the VNF Supplier API and Software Development Kit (VNF SDK) and VNF Validation Program (VVP) components. Vendors can integrate these tools in their CI/CD environments to package VNFs and upload them to the validation engine. Once tested, the VNFs can be onboarded through SDC.

为了支持和鼓励一个健康的 VNF 生态，ONAP 在 VNF 供应商 API、VNF SDK 和 VVP 组件中提供一套 VNF 打包和验证工具。厂商可以在他们的 CI（持续集成）/CD（持续交付）环境中集成这些工具，从而打包 VNF，并将其上传到验证引擎。一旦经过测试，这些 VNF 就可以通过 SDC 上线。

The Policy Creation component deals with polices; these are rules, conditions, requirements, constraints, attributes, or needs that must be provided, maintained, and/or enforced. At a lower level, Policy involves machine-readable rules enabling actions to be taken based on triggers or requests. Policies often consider specific conditions in effect (both in terms of triggering specific policies when

conditions are met, and in selecting specific outcomes of the evaluated policies appropriate to the conditions).

Policy Creation 组件用于处理策略；这些是必需提供、维护和/或强制执行的规则、条件、要求、约束、属性或需求。在较低的层面上，策略包括机器可读的规则，使得机器可以基于触发器或请求采取行动。策略通常考虑实际应用中的特定条件（无论是在符合条件时触发特定的策略，还是采取特定的策略以接近特定的条件）。

Policy allows rapid modification through easily updating rules, thus updating technical behaviors of components in which those policies are used, without requiring rewrites of their software code. Policy permits simpler management / control of complex mechanisms via abstraction.

策略允许通过更新规则进行快速修改，从而在不需要重写软件代码的情况下，更新使用这些策略的组件的技术行为。策略允许通过抽象简化对复杂机制的管理/控制。

The Closed Loop Automation Management Platform (CLAMP) provides a platform for designing and managing control loops. CLAMP is used to design a closed loop, configure it with specific parameters for a particular network service, then deploy and decommission it. Once deployed, a user can also update the loop with new parameters during runtime, as well as suspend and restart it.

CLAMP 为闭环控制的设计和管理提供了一个平台。CLAMP 用于设计一个闭环，针对某一项网络业务配置其特定的参数，然后部署和停止使用。一旦部署，用户可以在运行期间内更新控制环的参数，也可以暂停和重新启动它。

# 6. Runtime Framework 运行态框架

The runtime execution framework executes the rules and policies distributed by the design and creation environment.

运行态执行框架执行 SDC 分发的规则和策略。

This allows for the distribution of policy enforcement and templates among various ONAP modules such as the Service Orchestrator (SO), Controllers, Data Collection, Analytics and Events (DCAE),

Active and Available Inventory (A&AI), and a Security Framework. These components use common services that support logging, access control, and data management. A new component, Multi-Site State Coordination (MUSIC), allows the platform to register and manage state across multi-site deployments. The External API provides access for third-party frameworks such as MEF, TM Forum and potentially others, to facilitate interactions between operator BSS and relevant ONAP components.

运行态框架允许在不同的 ONAP 模块（例如 SO，控制器，DCAE，A&AI）中分发策略实施、模板，以及一个安全的框架。这些组件都使用了支持日志、控制访问和数据管理的通用服务。新组件 MUSIC 允许平台注册和管理多个站点的部署状态。外部 API 为第三方框架（例如 MEF、TM Forum 等）提供访问接口，从而支持运营商 BSS 与 ONAP 相关组件间的交互。

## Orchestration 编排

The Service Orchestrator (SO) component executes the specified processes by automating sequences of activities, tasks, rules and policies needed for on-demand creation, modification or removal of network, application or infrastructure services and resources. The SO provides orchestration at a very high level, with an end-to-end view of the infrastructure, network, and applications.

SO 组件通过自动顺序执行按需创建、修改或移除网络、应用或基础架构业务和资源所需的活动、任务、规则和策略，从而执行指定的流程。SO 在一个非常高的层次上进行编排，并提供基础设施、网络和应用的端到端视图。

The External API Northbound Interface component provides a standards-based interface between the BSS and various ONAP components, including Service Orchestrator, A&AI and SDC. This provides an abstracted view of the platform within the existing BSS/OSS environment without lengthy, high-cost infrastructure integration. The Beijing release is the first of a series of enhancements in support of SDO collaborations, which are expected to support inter-operator exchanges and other use cases defined by associated standards bodies such as MEF, TM Forum and others.

外部 API 北向接口组件在 BSS 和各个 ONAP 组件间提供标准化的接口，包括 Service Orchestrator，A＆AI 和 SDC。这样不需要通过长时间和高成本的基础架构集成，就可以在现有 BSS / OSS 环境中提供平台的抽象视图。北京版本及后续 ONAP 版本将持续增强和标准化组织间的合作，最终实现支持 MEF、TM Forum 等相关标准机构定义的运营商之间的交互用例以及其它用例。

The Virtual Infrastructure Deployment (VID) application enables users to instantiate infrastructure

services from SDC, along with their associated components, and to execute change management operations such as scaling and software upgrades to existing VNF instances.

VID 应用可以使用户从 SDC 实例化基础架构服务及其相关的组件，并执行变更管理操作，例如对现有 VNF 实例进行扩展和更新软件。

## Policy-Driven Workload Optimization 策略驱动的工作负载优化

The ONAP Optimization Framework (OOF) provides a policy-driven and model-driven framework for creating optimization applications for a broad range of use cases. OOF Homing and Allocation Service (HAS) is a policy-driven workload optimization service that enables optimized placement of services across multiple sites and multiple clouds, based on a wide variety of policy constraints including capacity, location, platform capabilities, and other service specific constraints.

OOF 提供一个策略驱动和模型驱动的框架，用于为各种用例创建优化应用。 OOF HAS 是一个策略驱动的工作负载优化服务，可基于各种策略约束（包括容量，位置，平台能力和其它特定的业务约束）实现跨多站点和多云的业务优化部署。

ONAP Multi-VIM/Cloud (MC) and several other ONAP components such as Policy, SO, A&AI etc. play an important role in enabling “Policy-driven Performance/Security-Aware Adaptive Workload Placement/Scheduling” across cloud sites through OOF-HAS. OOF-HAS uses Hardware Platform Awareness (HPA) and real-time capacity checks provided by ONAP MC to determine the optimal VIM/Cloud instances, which can deliver the required performance SLAs, for workload (VNF etc.) placement and scheduling (Homing). Operators now realize the true value of virtualization through fine grained optimization of cloud resources while delivering performance and security SLAs. For the Beijing release, this feature is available for the vCPE use case.

ONAP MC 和其它一些 ONAP 组件（如 Policy、SO、A&AI 等）在通过 OOF-HAS 实现跨云站点的“策略驱动的性能/安全感知的自适应工作负载部署/调度”中发挥了重要的作用。OOF-HAS 利用 HPA 和 ONAP MC 提供的实时容量检查，来确定最优的 VIM /云实例，这些实例可以为工作负载（VNF 等）的部署和调度（归属）提供所需的性能 SLA。现在运营商在保障性能和安全 SLA 的同时，可以通过云资源的细粒度优化实现虚拟化的真正价值。这一特性已经在北京版本的 vCPE 用例中得到体现。

## Controllers 控制器

Controllers are applications which are coupled with cloud and network services and execute the

configuration, real-time policies, and control the state of distributed components and services. Rather than using a single monolithic control layer, operators may choose to use multiple distinct Controller types that manage resources in the execution environment corresponding to their assigned controlled domain such as cloud computing resources (network configuration (SDN-C) and application (App-C). Also, the Virtual Function Controller (VF-C) provides an ETSI NFV compliant NFV-O function that is responsible for lifecycle management of virtual services and the associated physical COTS server infrastructure. VF-C provides a generic VNFM capability but also integrates with external VNFMs and VIMs as part of a NFV MANO stack.

控制器是一些应用程序，这些应用程序将云和网络业务耦合，并执行配置和实时策略，以及控制分布式组件和业务的状态。运营商可以选择使用多种不同的控制器类型，来管理执行环境中与其分配的控制域中相应的资源，例如云计算资源（网络配置 SDN-C 和应用 App-C），而不是仅仅使用一个单一整体控制层。此外，VF-C 提供符合 ETST NFV 标签的 NFVO 功能，并负责虚拟业务和相关物理的商用成品（CTOS）服务器基础设施的生命周期管理。VF-C 提供通过的 VNFM 功能，同时也可以与外部的 VNFMS 和 VIM 集成，作为 NFO MANO 堆栈的一部分。

In the Beijing release, the new Multisite State Coordination (MUSIC) project records and manages state of the Portal and ONAP Optimization Framework to ensure consistency, redundancy and high availability across geographically distributed ONAP deployments.

在北京版本中，新项目 MUSIC 将记录和管理 Portal 和 ONAP 优化框架的状态，以确保跨地理分布的 ONAP 部署的一致性，冗余和高可用性。

## Inventory 清单

Active and Available Inventory (A&AI) provides real-time views of a system's resources, services, products and their relationships with each other. The views provided by A&AI relate data managed by multiple ONAP instances, Business Support Systems (BSS), Operation Support Systems (OSS), and network applications to form a "top to bottom" view ranging from the products end-users buy, to the resources that form the raw material for creating the products. A&AI not only forms a registry of products, services, and resources, it also maintains up-to-date views of the relationships between these inventory items.

A&AI 提供系统资源、业务、产品及其相互关系的实时视图。A&AI 提供的视图将多个 ONAP 实例管理的数据、业务支持系统（BSS）、运营支持系统（OSS）和网络应用进行关联，从而形

成一个从终端用户购买的产品到形成产品原材的资源的自上而下的视图。A&AI 不仅形成产品、业务和资源的注册表，还保持这些清单项目的相互关系的最新视图。

To deliver the promised dynamism of SDN/NFV, A&AI is updated in real time by the controllers as they make changes in the network environment. A&AI is metadata-driven, allowing new inventory types to be added dynamically and quickly via SDC catalog definitions, eliminating the need for lengthy development cycles.

为了保证 SDN/NFV 的承诺的动态特性，当控制器在网络环境进行更改时，会对 A&AI 进行实时更新。A&AI 是元数据驱动的，允许通过 SDC 产品目录定义快速地动态添加新的清单类型，从而避免冗长的开发周期。

# 7. Closed-Loop Automation 闭环自动化

The following sections describe the ONAP frameworks designed to address major operator requirements. The key pattern that these frameworks help automate is:

**Design -> Create -> Collect -> Analyze -> Detect -> Publish -> Respond.**

以下各部分描述的是旨在解决主要运营商需求的 ONAP 框架。这些框架帮助实现自动化的关键模式是：

**设计 -> 创建 -> 采集 -> 分析 -> 检测 -> 发布 -> 响应**

We refer to this automation pattern as “closed-loop automation” in that it provides the necessary automation to proactively respond to network and service conditions without human intervention. A high-level schematic of the “closed-loop automation” and the various phases within the service lifecycle using the automation is depicted in Figure 4.

我们将这种自动化模式称为“闭环自动化”，是因为它提供了必要的自动化功能，可以在没有人工干预的情况下对网络和业务条件进行响应。图 4 是“闭环自动化”的高级示意图，显示了

使用自动化功能后业务生命周期内的不同阶段。

Closed-loop control is provided by Data Collection, Analytics and Events (DCAE) and one or more of the other ONAP runtime components. Collectively, they provide FCAPS (Fault Configuration Accounting Performance Security) functionality. DCAE collects performance, usage, and configuration data; provides computation of analytics; aids in troubleshooting; and publishes events, data and analytics (e.g., to policy, orchestration, and the data lake). Another component, “Holmes”, connects to DCAE and provides alarm correlation for ONAP.

闭环控制是通过 DCAE 和一个或多个其它 ONAP 运行态组件提供的。它们共同提供 FCAPS 功能。DCAE 收集性能、使用情况和配置数据，提供分析计算，帮助排除故障和发布事件、数据和分析方法（例如向策略、编排器和数据湖发布）。另一个组件 Holmes 与 DCAE 连接，并为 ONAP 提供告警关联功能。

Working with the Policy Framework and CLAMP, these components detect problems in the network and identify the appropriate remediation. In some cases, the action will be automatic, and they will notify Service Orchestrator or one of the controllers to take action. In other cases, as configured by the operator, they will raise an alarm but require human intervention before executing the change.

通过与 Policy Framework 和 CLAMP 协作，这些组件可以检测网络中存在的问题，并确定适当的补救措施。在某些情况下，这个工作是自动的，并且它们会通知 SO 或其中一个控制器采取行动。在其它情况下，根据操作人员的配置，它们会发出一个警报，在人工干预后再执行操作。

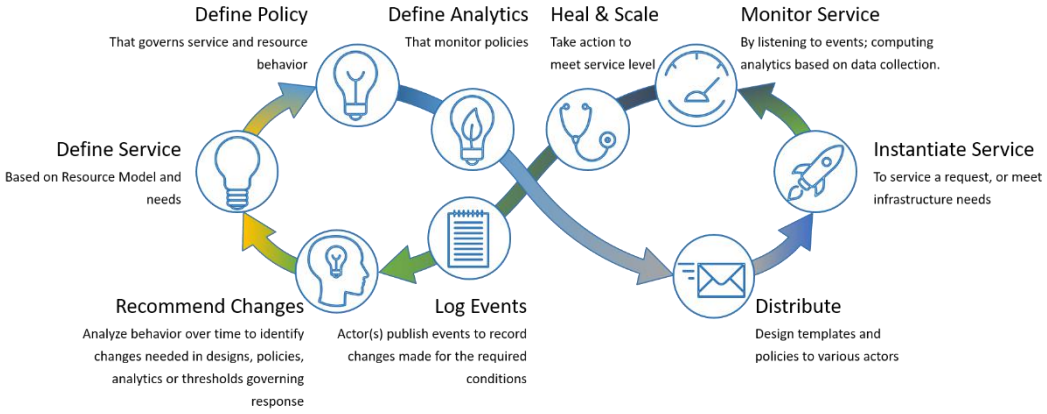


Figure 4: ONAP Closed Loop Automation

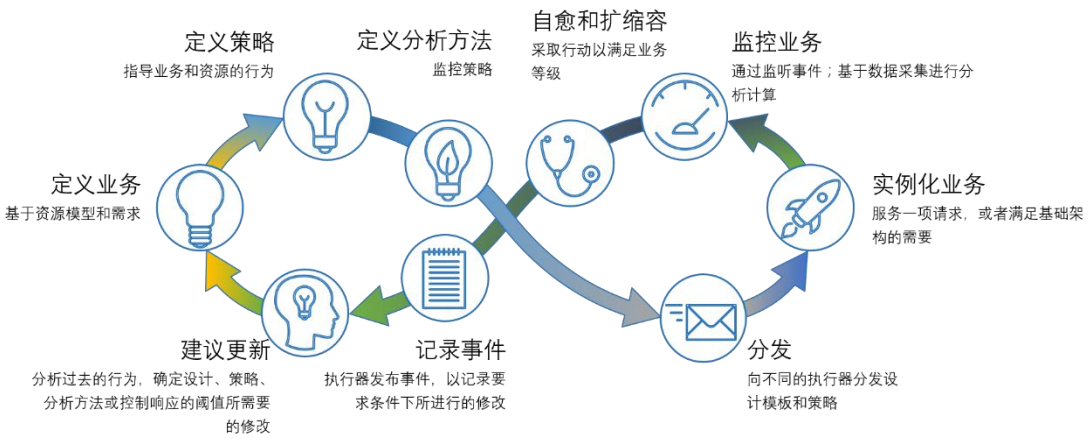


图 4 ONAP 闭环自动化

# 8. Common Services 通用服务

ONAP provides common operational services for all ONAP components including activity logging, reporting, common data layer, access control, secret and credential management, resiliency, and software lifecycle management.

ONAP 为所有 ONAP 组件提供通用的运行服务，包括活动记录、报告、通用数据层、访问控制、弹性和软件生命周期管理。

These services provide access management and security enforcement, data backup, restoration and recovery. They support standardized VNF interfaces and guidelines.

这些服务提供访问管理和安全执行、数据备件恢复和修复。它们支持标准化的 VNF 接口和指南。

Operating in a virtualized environment introduces new security challenges and opportunities. ONAP provides increased security by embedding access controls in each ONAP platform component, augmented by analytics and policy components specifically designed for the detection and mitigation of security violations.

在虚拟化的环境中运行会引入新的安全挑战和机遇。ONAP 通过在每个 ONAP 平台组件中嵌入访问控制来提供更高的安全性，同时专门设计用于检测和缓解违规操作的分析和策略组件来进一步增强安全性。

# 9. ONAP Modeling ONAP 建模

ONAP provides models to assist with service design, the development of ONAP service components, and with the improvement of standards interoperability.

ONAP 提供的模型有助于业务设计，ONAP 服务组件的开发以及标准互操作性的改进。

Models are essential part for the design-time and run-time framework development. The ONAP modeling project leverages the experience of member companies, standard organizations and other open source projects to produce models which are simple, extensible, and reusable. The goal is to fulfill the requirements of various use cases, guide the development and bring consistency among ONAP components and explore a common model to improve the interoperability of ONAP.

模型是设计态和运行态框架开发的重要组成部分。ONAP 建模项目利用成员公司、标准化组织和其他开源项目的经验，生成简单、可扩展和可重用的模型。目的是满足不同用例的需求，指导开发，保证 ONAP 组件间的一致性，并探索一个通用模型来提高 ONAP 的互操作性。

In the Bejing Release, ONAP supports the following Models:

在北京版本中，ONAP 支持下列模型：

- A VNF Information Model based on ETSI NFV IFA011 v.2.4.1 with appropriate modifications aligned with ONAP requirements;
- 基于 ETSI NFV IFA011 v.2.4.1 的 VNF 信息模型，根据 ONAP 的需求进行了适当的修改；
- A VNF Descriptor Model based on TOSCA implementation based on the IM and follow the same model definitions in ETSI NFV SOL001 v 0.6.0.
- 基于 TOSCA 实现（基于 IM）的 VNF 描述符模型，模型遵循 ETSI NFV SOL001 v 0.6.0 中

相同模型的定义。

- VNF Package format based on ETSI NFV SOL004 specification.
- 基于 ETSI NFV SOL004 规范的 VNF 包格式。

# 10. ONAP Use Cases ONAP 用例

The ONAP project tests blueprints for real-world use cases to enable rapid adoption of the platform. With the first release of ONAP (“Amsterdam”), we introduced two blueprints: vCPE and VoLTE. Subsequent releases test additional functionality and/or new blueprints.

ONAP 项目为真实应用场景进行蓝图测试，以实现平台的快速推广采用。在 ONAP 的第一个发布版本（阿姆斯特丹版本）中，我们介绍两种蓝图：vCPE 和 VoLTE。后续的版本将测试更多的功能和/或新的蓝图。

### Virtual CPE Use Case vCPE 用例

In this blueprint, many traditional network functions such as NAT, firewall, and parental controls are implemented as virtual network functions. These VNFs can either be deployed in the data center or at the customer edge (or both). Also, some network traffic will be tunneled (using MPLS VPN, VxLAN, etc.) to the data center, while other traffic can flow directly to the Internet. A vCPE infrastructure allows service providers to offer new value-added services to their customers with less dependency on the underlying hardware.

在这个用例中，许多传统的网络功能（例如 NAT、防火墙和家长控制）都是作为虚拟网络功能实现的。这些 VNF 可以部署在数据中心或者客户边缘（或两者相结合的方式）。另外，一些

网络流量也会通过隧道（利用 MPLS VPN、VxLAN 等）传输到数据中心，而其它的流量则直接流向互联网。一个 vCPE 基础架构允许业务提供商向客户提供新的增值业务，同时降低对底层硬件的依赖性。

In this blueprint, the customer has a physical CPE (pCPE) attached to a traditional broadband network such as DSL (Figure 1). On top of this service, a tunnel is established to a data center hosting various VNFs. In addition, depending on the capabilities of the pCPE, some functions can be deployed on the customer site.

在这个蓝图中，客户拥有一个连接到传统带宽网络（如 DSL）的物理 CPE（pCPE），如图 5 所示。在这业务之上，建立一条到掌控多个 VNF 的数据中心的隧道。另外，根据 pCPE 的能力，可以在客户站点部署一些功能。

This blueprint traditionally requires fairly complicated orchestration and management, managing both the virtual environment and underlay connectivity between the customer and the service provider. ONAP supports such a use case with two key components – SDN-C, which manages connectivity services, and APP-C, which manages virtualization services. In this case, ONAP provides a common service orchestration layer for the end-to-end service. It uses the SDN-C component to establish network connectivity. Similarly, ONAP uses the APP-C component to manage the VNF lifecycle. Deploying ONAP in this fashion simplifies and greatly accelerates the task of trialing and launching new value-added services.

这个蓝图传统上需要相当复杂的编排和管理，需要同时管理虚拟环境以及客户和业务提供商的底层连接。ONAP 通过两个关键组件来支持这种用例，管理连接业务的 SDN-C 和管理虚拟业务的 APP-C。在这种情况下，ONAP 为端到端业务提供一个通用的业务编排层。它利用 SDN-C 组件建立网络连接。同样，ONAP 利用 APP-C 组件来管理 VNF 的生命周期。以这种方式部署 ONAP 简化并极大加快了业务试用和新增业务的推出。

In the Beijing Release, the vCPE blueprint incorporates Policy-Driven Workload Optimization, which is supported by OOF, Multi-VIM/Cloud, Policy, SO, A&AI and other ONAP components.

在北京版本中，vCPE 蓝图结合了由 OOF、Multi-VIM/Cloud、Policy、SO、A&AI 和其它 ONAP 组件所支持的策略驱动的工作负载优化。

This enables ONAP to place VNFs in the right cloud/region based on constraints such as capacity, location and hardware platform awareness (HPA).

这使得 ONAP 可以根据容量、地理位置和 HPA 等约束将 VNF 部署到正确的云/区域中。

NFV will bring with it an era of continuous, incremental changes instead of periodic step-function software upgrades. The new change management feature executes an in-place upgrade workflow for the virtual gateway (vG) VNF in a vCPE environment.

与原来周期性的、阶梯式软件更新升级不同，NFV 将开启一个小步快跑、迭代式软件更新的时代。更新管理功能可以在 vCPE 环境中为虚拟网关（vG）VNF 远程更新工作流。

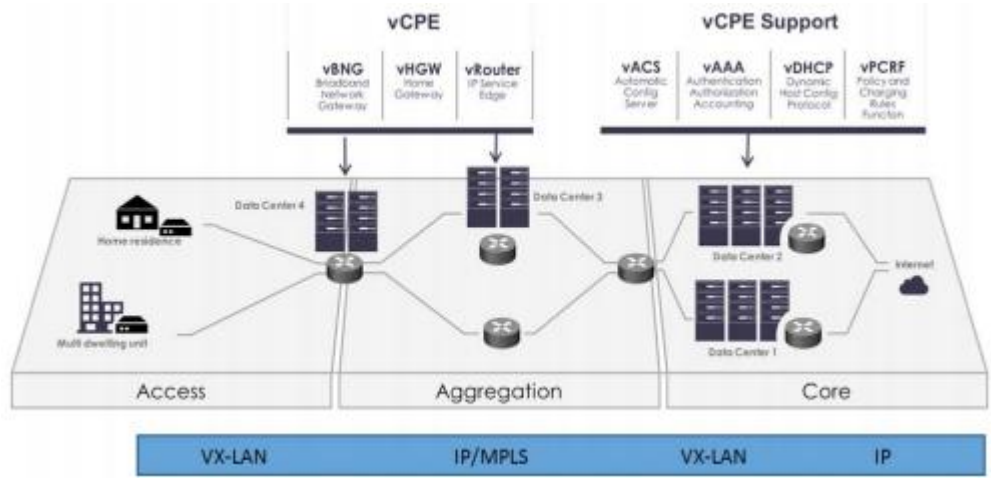


图 5 ONAP vCPE 架构

Read the Residential vCPE Use Case with ONAP whitepaper to learn more.

了解更多信息请阅读 Residential vCPE Use Case with ONAP whitepaper。

## Voice over LTE (VoLTE) Use Case VoLTE 用例

The second blueprint developed for ONAP is Voice over LTE. This blueprint demonstrates how a Mobile Service Provider (SP) could deploy VoLTE services based on SDN/NFV. This blueprint incorporates commercial VNFs to create and manage the underlying vEPC and vIMS services by interworking with vendor-specific components, including VNFMs, EMSs, VIMs and SDN controllers, across Edge Data Centers and a Core Date Center.

为 ONAP 设计的第二个蓝图是 VoLTE。这个蓝图演示了移动业务提供商（SP）如何基于 SND/NFV 部署 VoLTE 业务。这个蓝图结合商业 VNF，通过与特定厂商的组件（包括 VNFM、EMS、VIM 和 SDN 控制器）的相互配合，跨越边缘数据中心和核心数据中心创建和管理底层的 vEPC 和 vIMS 业务。

ONAP supports the VoLTE use case with several key components: SO, VF-C, SDN-C, and Multi-VIM/Cloud. In this blueprint, SO is responsible for VoLTE end-to-end service orchestration. It collaborates with VF-C and SDN-C to deploy the VoLTE service. ONAP uses the SDN-C component

to establish network connectivity, then the VF-C component completes the Network Services and VNF lifecycle management (including service initiation, termination and manual scaling) and FCAPS (fault, configuration, accounting, performance, security) management. VF-C can also integrate with commercial VIMs in the Edge and Core datacenters via abstract interfaces provided by Multi-VIM/Cloud.

ONAP 通过几个关键组件来支持 VoLTE 用例，包括 SO、VF-C、SDN-C 和 Multi-VIM/Cloud。在这个蓝图中，SO 负责 VoLTE 业务端到端的编排。它与 VF-C 和 SDN-C 合作来部署 VoLTE 业务。ONAP 利用 SDN-C 组件建立网络连接，利用 VF-C 组件完成网络业务和 VNF 生命周期的管理，以及 FCAPS 的管理。VF-C 还可以通过 Multi-VIM/Cloud 提供的抽象接口，与边缘和核心数据中心的商用 VIM 集成。

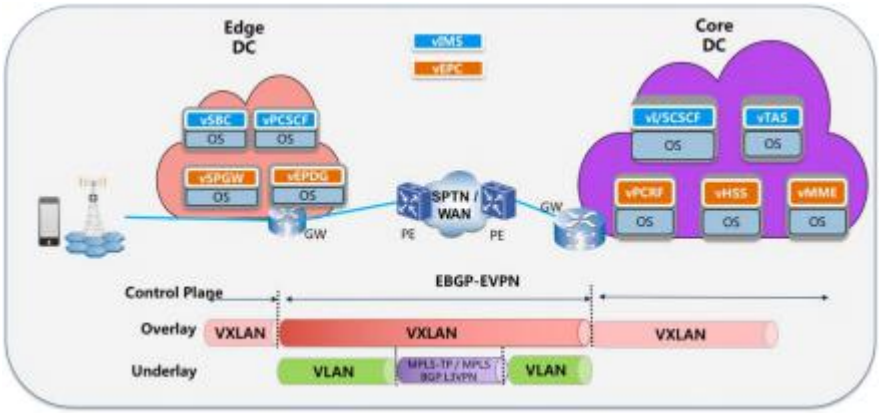


图 6 ONAP VoLTE 架构

Using ONAP to manage the complete lifecycle of the VoLTE use case brings increased agility, CAPEX and OPEX reductions, and increased infrastructure efficiency to Communication Service Providers (CSPs). In addition, the usage of commercial software in this blueprint offers CSPs an efficient path to rapid production.

使用 ONAP 管理 VoLTE 用例的完整生命周期，可以提高敏捷性，减少设备资产投入和运营成本，并提高通信业务提供商（CSP）的基础设施的效率。另外，在本蓝图中使用商业软件也为 CSP 提供了快速生产的有效途径。

The Beijing release has enriched the VoLTE blueprint with manually triggered scale-in and scale-out, demonstrating the dramatic flexibility of NFV.

北京版本在 VoLTE 蓝图中增加了人工触发扩缩容场景，展示了 NFV 的巨大灵活性。

Read the VoLTE Blueprint with ONAP whitepaper to learn more.

了解更多信息请阅读 VoLTE Use Case with ONAP whitepaper。

# 11. Conclusion 结论

The ONAP platform provides a comprehensive platform for real-time, policy-driven orchestration and automation of physical and virtual network functions that will enable software, network, IT and cloud providers and developers to rapidly automate new services and support complete lifecycle management.

ONAP 平台为物理/虚拟的网络功能提供了一个实时的、策略驱动的编排自动化综合平台，这将使软件、网络、IT、云业务提供商和开发人员可以快速自动化的部署新的业务，并支持全生命周期管理。

By unifying member resources, ONAP will accelerate the development of a vibrant ecosystem around a globally shared architecture and implementation for network automation–with an open standards focus–faster than any one product could on its own.

联合 ONAP 成员的资源，聚焦开放的标准体系，借助全球共享的平台架构和网络自动化的实现，ONAP 将比任何一个单独的产品更快速地发展一个充满活力的生态系统。

# 12. Resources 资源

Watch videos about the major platform components on YouTube and Youku

在 Youtube 和 Youku 上观看主要平台组件的相关视频。

Read about how ONAP can be deployed using containers

阅读了解如何利用容器部署 ONAP。

# 附：中英文对照表

| 序号 | 英文 | 中文 |
|---|---|---|
| 1 | operator | 运营商 |
| 2 | infrastructure | 基础架构 |
| 3 | performance | 性能 |

| | | |
|---|---|---|
| 4 | deploy | 部署 |
| 5 | response | 响应 |
| 6 | recipe | 流程 |
| 7 | policy | 策略 |
| 8 | instantiation | 实例化 |
| 9 | provisioning | 供应 |
| 10 | view | 视图 |
| 11 | analytics | 分析方法 |
| 12 | dashboard | 仪表盘 |
| 13 | onboard | 上线 |
| 14 | corrective | 纠正 |
| 15 | remedial | 补救 |
| 16 | design time | 设计态 |
| 17 | execution time / runtime | 运行态 |
| 18 | repositories | 资源库 |
| 19 | retire | 下线 |
| 20 | process | 流程 |
| 21 | inventory | 清单 |
| 22 | create | 创建 |
| 23 | publish | 发布 |
| 24 | SDO | 标准化组织 |

注：以下专用名词在译文中保持原文，以下翻译仅供参考。

1) SLA（service-level agreements）：业务质量

2) VIM（Virtualised Infrastructure Manager）：虚拟化基础设施管理器

3) VNFM（Virtualised Network Functions Manager）：虚拟化网络功能管理器

4) VNF（Virtualised Network Function）：虚拟化的网络功能

5) OSS（Office of Strategic Services）：运营支撑系统

6) SDC（Service Design and Creation）：业务设计和创建模块

7) SDK（Software Development Kit）：软件开发工具包

8) Policy Creation：策略制定

9) CLAMP（Closed Loop Automation Management Platform）：闭环自动化管理平台

10) SO（Service Orchestrator）：业务编排器

11) DCAE（Data Collection, Analytics and Events）：数据采集、分析和事件模块

12) A&AI（Active and Available Inventory）：活跃和可用清单

13) VF-C（Virtual Function Controller）：虚拟功能控制器

14) FCAPS（Fault Configuration Accounting Performance Security）：故障、配置、计费、性能、安全

15) Policy Framework：策略框架

16) OOF（ONAP Optimization Framework）：ONAP 优化框架

17) OOM（ONAP Operations Manager）：ONAP 运行管理模块

18) MSB（Microservices Bus）：微服务总线

19) VVP（VNF Validation Program）：VNF 验证程序

20) MUSIC（Multi-Site State Coordination）：多站点状态协调

21) VID（Virtual Infrastructure Deployment）：虚拟基础架构部署

22) HAS（Homing and Allocation Service）：归属分配服务

23) MC（Multi-VIM / Cloud）：多 VIM/云

24) HPA（Hardware Platform Awareness）：硬件平台感知