思科以应用为中心的基础架构的基本信息

首次发布：2014 年 8 月 1 日

最后修改日期：2014 年 11 月 10 日

美洲总部
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
电话： 408 526-4000
800 553-NETS (6387)
传真： 408 527-0883

本手册中所述产品的规格和信息可能会发生改变，恕不另行通知。我们认为本手册中的所有声明、信息和建议都准确无误，但不提供任何明示或暗示的保证。用户必须对任何产品的使用负有全部责任。

针对随附产品的软件许可和有限保修在与产品一同交付的信息包中进行了说明并包含在了本参考书中。如果您无法找到软件许可或有限保修，请联系您的思科代表从而获得一份。

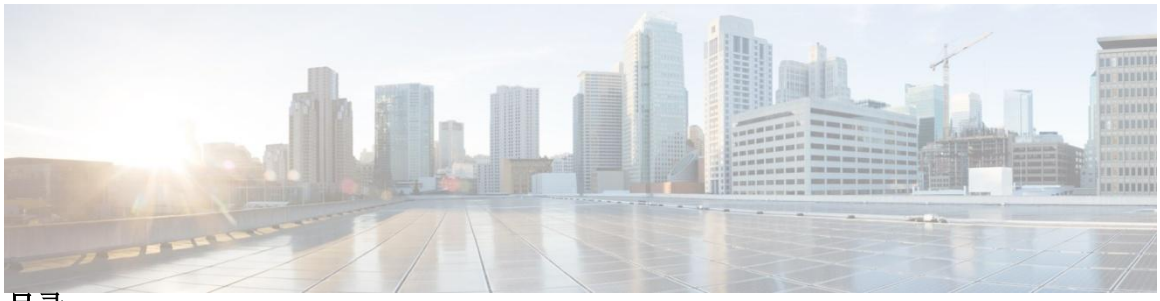
思科执行的 TCP 报头压缩是对加州大学伯克利分校（UCB）开发的某一程序的修改，它是 UNIX 操作系统的 UCB 公用版的一部分。版权所有。© 1981，加利福尼亚大学董事。

尽管本文包含其他担保，本手册中供应商的所有文档和软件均按“原样”提供，可能包含错误信息。思科及其上述供应商不提供任何明示或暗示担保，包括但不限于对适销性、适合特定用途和非侵权的担保，或由交易过 程、使用方式或贸易惯例所产生的担保。

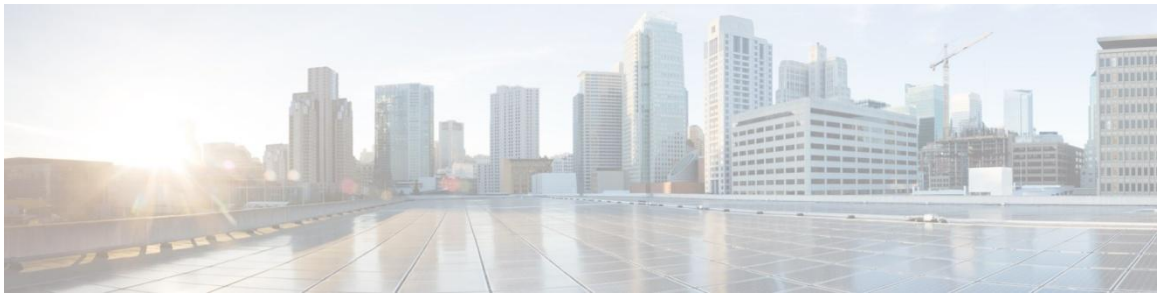
任何情况下，思科或其供应商均不对任何间接性、特殊性、后果性或附带性损害承担责任，包括但不限于由于使用或未能使用这些设计而导致的利润损失或数据丢失或损坏，即使思科或其供应商已被告知存在此类损害的可能性。

本文档中使用的任何互联网协议 (IP) 地址和电话号码并不代表实际地址和电话号码。本文档中包括的任何示例、命令显示输出、网络拓扑图和其他图形仅用于说明目的。在图示内容中使用的 IP 地址或电话号码纯属虚构，如有雷同，纯属巧合。

思科和思科徽标是思科系统公司和/或其在美国以及其他国家/地区的附属公司的商标。如需思科商标的列表，请访问：http:// www.cisco.com/go/trademarks。本文提及的第三方商标均归属其各自所有者。使用“合作伙伴”一词并不意味着思科与任何其他公司之间存在合作伙伴关系。(1110R)



目录



前言

前言包含下列几部分：

目标读者

本指南主要面向数据中心负责并懂得下列一个或多个领域的管理员：

- 虚拟机安装和管理
- 服务器管理
- 交换机和网络管理

文档惯例

命令说明适用下列惯例：

惯例	说明
加粗	加粗文本表示你按照说明逐字输入的命令和关键词。
斜体	斜体文本表示用户提供数值的参数。
[x]	方括号中表示可选元素（关键字或参数）。

文档惯例

惯例	说明
[x | y]	方括号里面有被竖线隔开的关键字或参数表示可选项。
{x | y}	大括号里面有被竖线隔开的关键字或参数表示必选项。
[x {y | z}]	嵌套的方括号或大括号表示可选元素或必选元素中的可选或必选项。方括号内的大括号和竖线表示可选元素中的必选项。
变量	在无法使用斜体的上下文中表示你为之提供数值的变量。
字符串	一组非引用的字符。字符串两端不要加引号，否则字符串将会包含引号。

示例采用下列惯例：

惯例	说明
屏幕字体	终端会话和交换机显示的 i 型你西采用屏幕字体。
屏幕字体黑体	你必须用屏幕字体黑体输入的信息。
屏幕字体斜体	你为之提供数值的参数采用屏幕字体斜体。
< >	非打印字符，如密码，用尖括号表示。
[ ]	对系统提示的默认回复采用方括号。
#----------	在一行代码开头的叹号（！）或井号（#）表示注释行。

本文采用下列惯例：

备注 表示读者应注意的地方。备注包含指南中没有涉及的有用建议或参考。

小心 表示读者应小心的地方。在这种情况下，用户可能会进行可能导致仪器损坏或数据丢失的操作。

警告 重要的安全说明

这种警告符号表示危险。用户所处的情况可能导致人身伤害。在操作任何设备之前，注意与电路有关的危险，熟悉预防事故的常规做法。通过每个警告末尾的语句编号可以找到随机附带的安全警告的译文。

保存说明

相关文档

《以应用为中心的基础架构》文档集包含的下列文档可以在 Cisco.com 中的下列网址查阅：http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html。

网页文档

- *思科 APIC 管理信息模型参考*
- *思科 APIC 在线帮助参考*
- *思科 APIC Python SDK 参考*
- *思科 ACI 兼容性工具*
- *思科 ACI MIB 支持列表*

下载文档

- 知识库文章（KB 文章）可通过下列网址下载：http://www.cisco.com/c/en/ us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/ products-configuration-examples-list.html
- *思科以应用为中心的基础架构发行指南*
- *思科以应用为中心的基础架构的基本信息指南*
- *思科 APIC 启动指南*
- *思科 APIC REST API 用户指南*
- *思科 APIC 命令行界面用户指南*
- *思科 APIC 默认值、时间和系统消息指南*
- *思科 ACI NX-OS 系统日志参考指南*
- *思科 APIC 第 4 层到第 7 层服务开发指南*
- *思科 APIC 第 4 到第 7 层设备套件开发指南*
- *思科 APIC 第 4 层到第 7 层设备套件测试指*
- *思科 ACI 固件管理指南*
- *思科 ACI 故障排除指南*

- *思科 ACI 交换机命令参考，NX-OS 版本 11.0*
- *思科 ACI MIB 快速入门*
- *思科 Nexus CLI 到思科 APIC 映射指南*
- *以应用为中心的基础架构矩阵硬件安装指南*
- *思科 Nexus 9336PQ ACI-Mode 交换机硬件安装指南*
- *思科 Nexus 9396PX ACI-Mode 交换机硬件安装指南*
- *思科 Nexus 9396TX ACI-Mode 交换机硬件安装指南*
- *思科 Nexus 93128TX ACI-Mode 交换机硬件安装指南*
- *思科 Nexus 9504 ACI-Mode 交换机硬件安装指南*
- *思科 Nexus 9508 ACI-Mode 交换机硬件安装指南*

思科以应用为中心的基础架构（ACI）模拟器文档

如欲查看下列思科 ACI 模拟器文档，可登录 http://www.cisco.com/c/en/us/support/ cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html。

- *思科 ACI 模拟器版本注释*
- *思科 ACI 模拟器安装指南*
- *思科 ACI 模拟器启动指南*

思科 Nexus 9000 系列交换机文档

如欲查看思科 Nexus 9000 系列交换机文档，可登录 http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html。

思科应用虚拟交换机文档

如欲查看思科应用虚拟交换机（AVS）文档，可登录 http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html。

文档反馈

如需为本文提供技术反馈或报告错误或遗漏，请将您的意见发送至 apic-docfeedback@cisco.com。十分感谢您的反馈。

获取文档和提交服务请求

关于获得文档、使用思科漏洞搜索工具（BST）、提交服务请求和搜集其他信息的信息，请登录 http:// www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html 参考《思科产品文档最新动态》

订阅《思科产品文档最新动态》，以 RSS 邮件的方式查看所有全新的和经过修改的思科技术文档，通过阅读器应用直接把内容推送到桌面。RSS 邮件是一项免费服务。

第 1 章

全新的和经过修改的信息

本章包括以下部分：

全新的和经过修改的信息

下表概况说明了本指南针对当前版本而进行的一些重要修改。该表并未包含本指南的所有更改和本版本的所有新特性。

表 1：思科 APIC 版本 1.0 中的新特性和行为变化（2h）

特性	说明	在文中的位置
--风暴控制。	执行第 2 层风暴控制。	--关于流量风暴控制，参考第 28 页
--AAA VMM 域标签。	VMM 域可以被标记为安全域，这样它们就可以对包含在安全域中的用户可见。	--用户接入：角色、权限和安全 域，第 70 页
--VMM 组件按需同步。	被激活的组件提供了一个手动激活选项，用于在 VMM 控制器和 APIC 之间拉动和再次同步组件。	--VMM 组件按需刷新，第 93 页
--通向 IP 地址选项的原子计数器端点。	支持选择目标 MAC 地址或 IP 地址。	--原子计数器，第 133 页
--删除 VMM 域指南。	找出建议的工作流程顺序。	删除 VMM 域指南，第 93 页

全新的和经过修改的信息

特性	说明	在文中的位置
--自定义 RBAC 规则。	确定用于制定自定义 RBAC 规则的使用案例场景和指导方针。	--自定义 RBAC 规则，第 70 页 --RBAC 规则样本，第 167 页
--健康得分计算。	确定系统、pod、租户和 MO 层级健康得分的计算方法。	--健康得分，第 128 页
--多节点 SPAN ERSPAN 指导方针和首标类型	确定 ERSPAN 首标类型和使用 ERSPAN 的指导方针	--多节点 SPAN，第 134 页
--EPG 不带标签和带标签的 VLAN 首标。	为使用不带标签的 EPG VLANS 提供指导方针和限制。	--服务器组，第 13 页
--桥接域传统模式。	提供用于配置传统桥接域的指导方针。	--桥接域和子网，第 18 页
--更新到 AAA LDAP 和 TCACS+配置，包含示例。	增加 AAA LDAP 和 TCACS+配置示例。	--LDAP/主动式目录验证，第 77 页 --TACACS+ 验证，第 76 页
--VMM 带内 ESX VMK0 配置指导方针。	本文没有重大改动。	关于向 ACI 带内 VLAN 迁移 Vcenter 管理程序 VMK0 的指南，第 94 页
--针对配置导入/导出最佳努力、原子、合并和更换选项的更新。	说明配置导入/导出策略的增强。	--配置导出/导入，第 112 页
--更新删除（带擦除选项）。	为使用删除边缘交换机（带擦除选项）提供指南。	矩阵组件，第 39 页

第 2 章

思科以应用为中心的基础架构

本章包括以下部分：

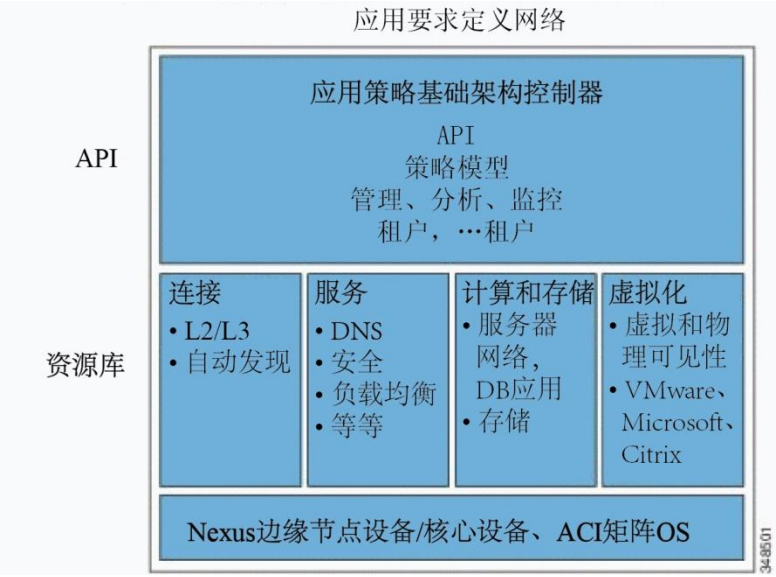
关于思科以应用为中心的基础架构

思科以应用为中心的基础架构满足应用定义网络的要求。这个基础架构简化、优化并加速应用部署的整个生命周期。

关于思科应用策略基础架构控制器

思科应用策略基础架构控制器（APIC）API 支持应用直接连接一个包含网络、计算和存储能力的安全、共享且高性能的资源库。下图概括描述了 APIC。

表 1：APIC 概览

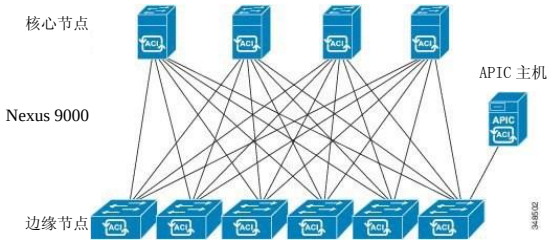


APIC 管理可扩展的 ACI 多租户矩阵。APIC 提供了一个针对矩阵的自动化和管理、策略编程、应用部署和健康监控的统一管理点。作为一个复制、同步、集群化控制器实施的 APIC 优化了性能、在任何地方为任何设备提供支持，提供物理和虚拟基础机构的统一操作。APIC 使得网络管理员可以轻松地为应用定义最佳网络。数据中心操作员可以清楚地看到应用使用网络资源的方式，轻松地隔离和解决应用和基础设施问题，并监控和描述资源使用类型。

思科以应用为中心的基础架构概览

思科以应用为中心的基础架构（ACI）矩阵包含思科 Nexus 9000 系列交换机，APIC 以边缘节点设备/核心设备矩阵模型运行。这些交换机把每个边缘节点同每个核心节点连在一起，形成了一个“胖树”网络；所有其他设备都与边缘节点连接。APIC 管理 ACI 矩阵。为 APIC 推荐的最小配置是三个复制主机的集群。APIC 矩阵管理功能不在矩阵数据路径中运行。下图概括描述了边缘/核心 ACI 矩阵。

表 2：ACI 矩阵概述

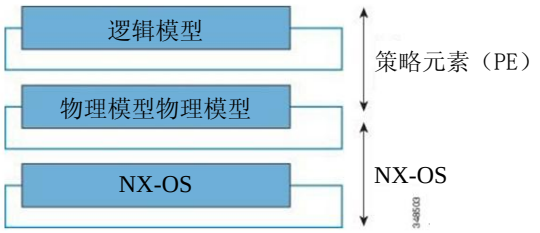


ACI 矩阵在高带宽链路（40 Gbps，未来 100-Gbps）提供了一致的低延迟转发。源地址和目标地址位于同一个边缘交换机上的流量在本地处理，而所有其他流量通过核心交换机从入口边缘节点设备传输到出口边缘节点设备。虽然从物理的角度这种基础架构看起来像两跳，而实际上它是单个的第 3 层跳跃，因为这个矩阵作为一个第 3 层交换机运行。

ACI 矩阵面向对象的操作系统（OS）在思科 Nexus 9000 系列的每个节点上运行。它使得能够为系统的每个可配置元素进行对象编程。

ACI 矩阵操作系统把来自 APIC 的策略渲染为可在物理基础架构中运行的物理模型物理模型。这种物理模型物理模型类似于编译的软件；它是交换机操作系统可以执行的那种模型。下图显示了逻辑模型和物理模型物理模型以及交换机操作系统之间的关系。

图 3：被渲染为物理模型物理模型的逻辑模型



所有的交换机节点都包含一套完整的物理模型物理模型。当一名管理员在代表一种配置的 APIC 中创建一个策略时，APIC 会更新这个逻辑模型。然后 APIC 就会执行创建一个完全详尽的策略，将其推送到更新了物理模型的所有交换机节点。

备注 思科 Nexus 9000 系列交换机仅能执行这个物理模型。每个交换机都有一套物理模型。如果所有 APIC 都离线，那么矩阵会继续运行，但无法修改矩阵策略。

APIC 负责矩阵激活，交换机固件管理、网络策略配置和安装。尽管 APIC 为矩阵充当集中化策略和网络管理引擎，它完全从数据路径中移除，包括转发拓扑。因此，矩阵仍可以在与 APIC 的通讯丢失后转发流量。

思科 Nexus 9000 系列交换机提供模块化且固定的 1G、10G 和 40G 以太网交换机配置。该配置既可以在思科 NX-OS 独立模式下运行，从而与当前的思科 Nexus 交换机兼容和一致；也可以在 ACI 模式下运行，以便完全利用 APIC 的应用策略驱动服务和基础结构的自动化特性。

确定矩阵的行为方式

ACI 矩阵允许客户自动化和安排可扩展、高性能的网络资源、计算资源和存储资源，从而部署云。定义 ACI 矩阵行为方式的重要参与者包括：

- IT 计划员、网络工程师和安全工程师
- 通过 APIC API 访问系统的开发者
- 应用和网络管理员

REST 架构是一种支持云计算的重要开发方法。ACI API 基于 REST。万维网（World Wide Wed）是一种符合 REST 架构风格的最大的系统实施。

云计算在规模和方法方面与传统计算有所不同。传统环境包括软件和维护需求，需要各种相关技能，消耗大量运行成本。大规模基础架构为云应用所用的系统设计提供支持，而前者的部署成本曲线下降得非常迅速。在这种基础架构风格下，系统管理员、开发团队和网络人员通过协作做出价值更高的贡献。

在传统环境下，计算资源和端点的网络接入通过虚拟 LAN（VLAN）或刚性面层（如多协议标签交换（MPLS））管理，通过严格定义的网络服务（如负载均衡器和防火墙）推动流量。APIC 的设计过程中考虑了可编程性和集中化管理。通过抽象网络，ACI 矩阵使得操作员可以在网络中动态而不是静态地配置资源。如此一来，部署时间（上市时间）可以从数月或数周减少到几分钟。通过调用 API，改变虚拟或物理交换机、适配器、策略和其他软硬件部件的配置可以在几分钟内完成。

从传统做法变革到云计算方法要求从数据中心获得更加灵活、可扩展的服务。这些变革需要大量技能高超的人员。APIC 的设计过程中考虑了可编程性和集中化管理。APIC 的一种重要特性是被称为“REST”的网页 API。APIC REST API 接受并返回 HTTP 或 HTTP 消息，后者包含 JavaScript 对象表示法（JSON）或可扩展标记语言（XML）文件。现在很多网页开发者使用 RESTful 方法。在整个网络中采用 API 使得企业可以轻松地与其他内部或外部的服务提供方开创和整合服务。这个过程将网络从一个复杂的静态资源包转化为在售的动态服务交换。

第 3 章

ACI 策略模型

本章包括以下部分：

关于 ACI 策略模型

ACI 策略模型实现了应用程序需求策略的具体要求。APIC 在结构基础架构中自动传递策略。当用户或程序对结构中的一个对象发起管理变更时，APIC 首先会将该变更应用到策略模型中。随后，策略模型变更引发了实际管理端点的变更。这种方法被称作模型驱动架构。

策略模型的关键特征

策略模型的关键特征包括以下内容：

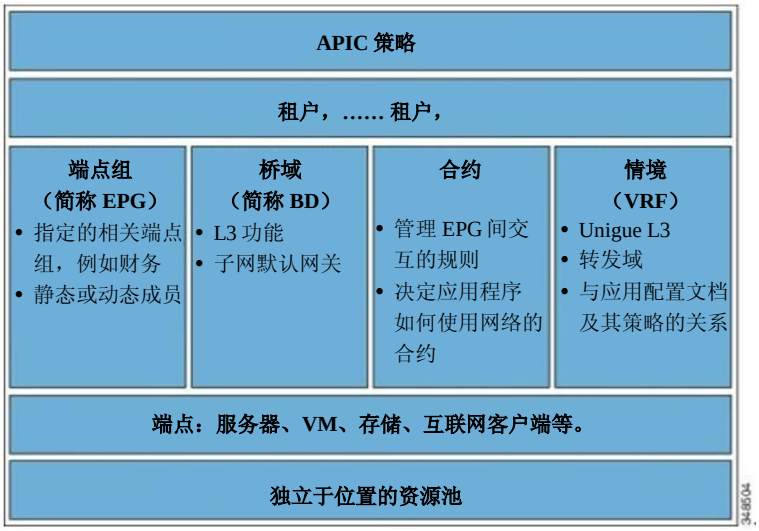
- 作为一种模型驱动的结构，该软件全面展现了系统（模型）的管理状态和运行状态。该模型统一应用到结构、服务、系统行为、连接到网络的虚拟设备和实体设备。
- 逻辑域和具体域是分开的；通过应用与可用的实体资源相关的策略，逻辑配置转换为具体配置。实施的配置没有不符合具体实体的。具体实体是作为 APIC 策略模型变更的副作用而配置的。具体实体可以是实物，但并非一定得是实物（比如虚拟机或 VLAN）。
- 系统禁止与新接入的设备通信，除非策略模型更新，已包括该新设备。
- 网络管理员不直接配置逻辑系统资源和实物系统资源，而是定义逻辑（独立于硬件的）配置和控制系统行为不同方面的 APIC 策略。

模型中的管理对象操控把工程师从管理独立、单个的组件配置任务中解放出来。这些特征使得灵活、自动的工作负荷配置成为可能，能够找到基础构架中任何地方的任何工作负荷。与网络相连的服务可以轻松配置，APIC 提供了一个自动框架，管理那些与网络相连的服务的生命周期。

逻辑构造

策略模型管理着整个结构，包括基础架构、验证授权、安全、服务、应用和诊断。策略模型中的逻辑构造定义了结构如何满足结构中所有功能的需求。下表展示了 ACI 策略模型逻辑构造的概览。

表 4：ACI 策略模型逻辑构造概览



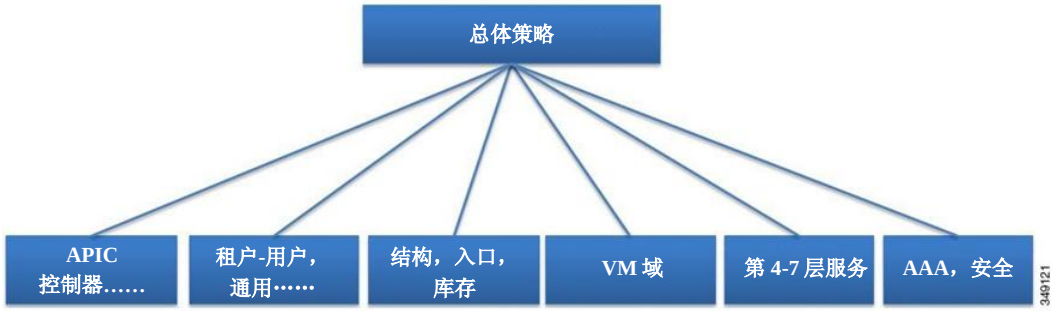
结构范围内的管理员或租户管理员创建了预定义的策略，包括应用程序或共享资源需求。这些策略使得应用程序设置、与网络相关的服务、安全策略、租户子网等实现自动化，将管理员置于根据应用程序处理资源池的位置上，而不是根据基础架构构建模块。应用程序需要驱动网络行为，而不是反过来。

管理信息模型

结构包括实物组件和逻辑组件，都记录在管理信息模型（简称 MIM）中，可以用分层的管理信息树（简称 MIT）展现。信息模型由 APIC 上运行的程序来储存并管理。与 OSI 的公共管理信息协议（简称 CMIP）及其他 X.500 变体类似，APIC 能够控制管理资源，方法是展示它们的管理特征作为对象属性，这些属性可以根据对象在 MIT 分层结构中的位置来获得。

树上的每个节点都代表了一个管理对象（简称 MO）或一组对象。MO 是结构资源的抽象。MO 可以代表具体对象，例交换机、适配器，也可以代表逻辑对象，例如应用配置文档、端点组，或故障。下表展示了 MIT 概览。

表 5：管理信息树概览



分层结构开始于顶层的总体策略（称作 Root），包括父节点和子节点。树上的每个节点都是一个 MO，结构中的每个对象都有一个唯一的专有名称（简称 DN），描述了该对象，并确定其在树上的位置。

下面的管理对象包括了管理系统运行的策略：

- APIC 控制器包括了一个复制的同步群集控制器，提供针对多租户结构的管理、策略编程、应用部署和健康监控。
- 租户是策略的容器，使得管理员能够行使基于域的接入控制。系统提供以下四种租户：
    - 用户租户由管理员根据用户的需要而定义。其包含各项策略，管理各种资源的运行，例如应用程序、数据库、网络服务器、与网络相连的存储、虚拟机等。
    - 通用租户由系统提供，但可以由结构管理员配置。其包含各项策略，管理所有租户可访问的资源的运行，例如防火墙、负载均衡器、第 4 至 7 层服务、入侵检测设备等。
    - 基础架构租户由系统提供，但可以由结构管理员配置。其包含各项策略，管理各种基础架构资源的运行，例如结构 VXLAN 叠加等。它使得结构提供者可以选择性地将资源配置给一个或多个用户租户。基础架构租户策略可由结构管理员配置。
    - 管理租户由系统提供，但可以由结构管理员配置。其包含多项策略，管理结构管理功能的运行，这些功能用于结构节点的带内和带外配置。管理租户包含一个私人的带外地址空间，用于 APIC/结构内部通信，其处于结构数据路径之外，通过交换机的管理端口提供接入。管理租户能够发现与虚拟机管理者的通信，并使其自动化。
- 接入策略管理着转换接入端口的运行，提供与如下资源的连接，例如存储、计算、第 2 层和第 3 层（桥接与路由）连接、虚拟机管理程序、第 4 层至第 7 层设备，等。如果租户需要一些接口配置，且不是默认链接提供的那些接口配置，思科发现协议（简称 CDP），链路层发现协议（简称 LLDP），链路聚合控制协议（LACP），或生成树，管理员必须配置接入策略，以激活这些处于边缘交换机接口的配置。

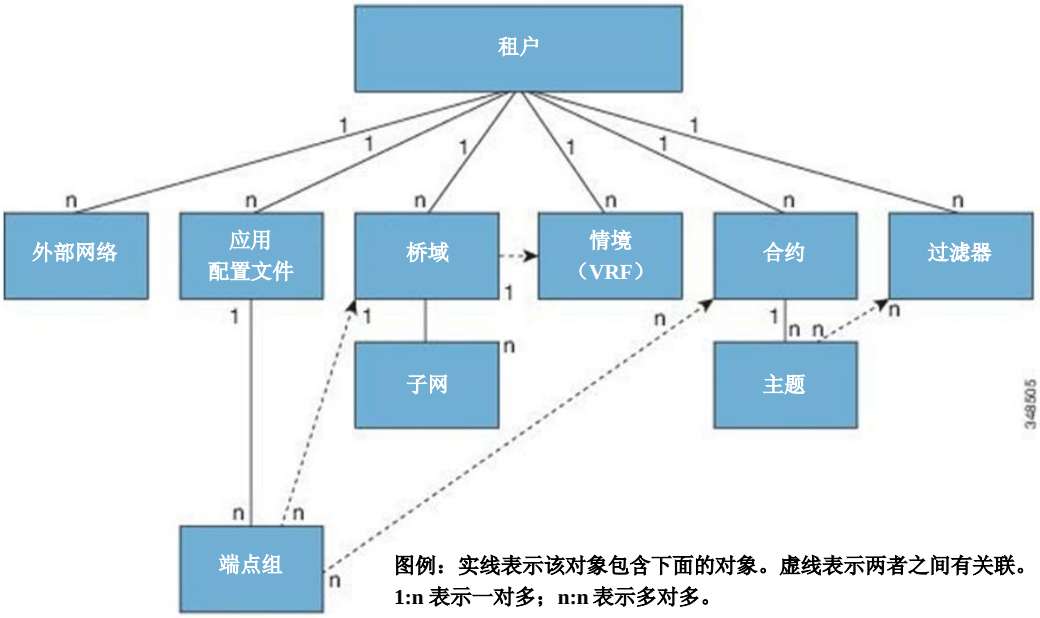
- 结构策略管理着转换结构接口的运行，包括如下功能，例如网络时间协议服务器同步（简称 NTP），中间系统到中间系统协议（简称 IS-IS），边界网关协议（简称 BGP）路由反射器，域名系统（简称 DNS）等。结构 MO 包括如下对象，如电源、风扇、机箱等。
- 虚拟机（简称 VM）域集合了拥有相似网络策略需求的 VM 控制器。VM 控制器可以共享 VLAN 或虚拟可扩展局域网（简称 VXLAN）空间和应用端点组（简称 EPG）。APIC 与 VM 控制器通信，以发布网络配置，例如接口组，随后应用到虚拟工作负载上。
- 第 4 层至第 7 层服务集成生命周期自动化框架，使得系统在服务上线或下线时能够动态响应。这些策略提供了服务设备组合和库存管理功能。
- 访问、认证、计费（简称 AAA）策略管理着思科 ACI 结构的用户权限、角色和安全域。

分层策略模型与 RESTful API 接口非常匹配。当调用时，API 读取或写入到 MIT 上的对象。URL 直接映射到专有名称上，这些名称可以识别出 MIT 上的对象。MIT 上的任何数据都可以被描述成一个设施齐全的结构树文本文档，该文档采用了 XML 或 JSON 编码。

租户

租户（fvTenant）是一个应用策略的逻辑容器，使得管理员能够执行基于域的接入控制。租户代表了一个从策略角度来看的独立单元，但并不代表一个专用网络。租户可以代表服务供应者设置中的客户，企业设置中的组织机构或域，或只是一组方便分组的策略。下表展示了管理信息树（MIT）中租户部分的概览。

表 6：租户



图例：实线表示该对象包含下面的对象。虚线表示两者之间有关联。1:n 表示一对多；n:n 表示多对多。

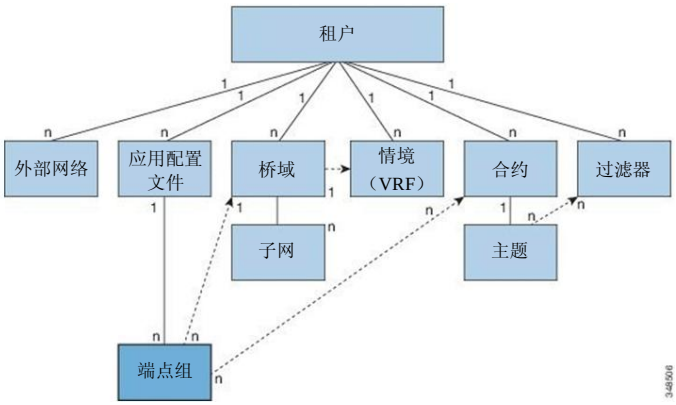
租户可以彼此独立，也可以共享资源。租户包含的主要组成部分有：过滤器、合约、外部网络、桥域、情境、包含端点组（EPG）的应用配置文档。租户内的单元获得了其策略。租户可以包含一个或多个虚拟路由与转发（VRF）实例或情境；每个情境可以与多个桥域相关联。

租户是应用策略的逻辑容器。结构可以包含多个租户。你必须先配置一个租户，然后才能部署第 4 层至第 7 层的服务。

端点组

端点组（EPG）是策略模型中最重要的对象。下表展示了在管理信息树（MIT）中应用程序端点组位于何处，以及他们与租户中其他对象的关系。

表 7：端点组



EPG 是一个管理对象，它是一个包含一系列端点的指定逻辑单元。端点是直接或间接与网络相连的设备。他们有地址（标识）、位置、属性（例如版本或补丁级别），可以是实物，也可以是虚拟的。知道一个端点的地址，就能接入其他所有的身份信息。EPG 在实物和逻辑拓扑中均完全解耦合。端点的示例包括服务器、虚拟机、与网络相连的存储、或是互联网上的客户端。EPG 中的端点成员可以是动态的，也可以是静态的。

EPG 包含拥有共同策略需求的端点，例如安全性、虚拟机移动性（VMM）、QoS、或第 4 层至第 7 层服务。端点并非是独立配置和管理，而是放在一个 EPG 中，作为一组整体管理。ACI 结构可以包括以下类型的 EPG：

- 应用程序端点组（fvAEPg）
- 第 2 层外部的外网实例端点组（l2extInstP）
- 第 3 层外部的外网实例端点组（l3extInstP）
- 带外（mgmtOoB）或带内（mgmtInB）接入管理端点组

策略应用于 EPG，而绝不会应用于单独的端点。EPG 可以由 APIC 中的管理员进行静态配置，或者是由例如 vCenter 或 OpenStack 的自动系统进行动态配置。

备注 当 EPG 使用静态绑定路径，与此 EPG 相关联的封装 VLAN 必须是一个静态 VLAN 池的一部分。

无论 EPG 如何配置，EPG 策略都被应用到其包含的端点上。

WAN 路由器连接至结构是使用静态 EPG 配置的一个示例。为了配置

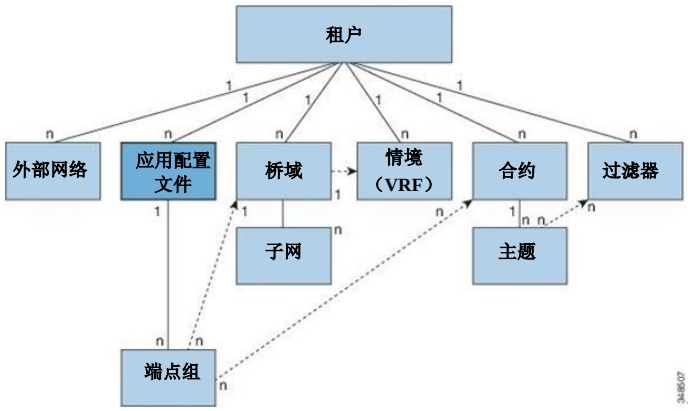
WAN 路由器连接至结构，管理员要配置一个 l3extInstP EPG，包含相关联的 WAN 子网中的所有端点。当端点在其连接生命周期中前进时，结构通过发现程序获知到端点。通过获知端点，结构根据情况应用 l3extInstP EPG 策略。例如，当一个 WAN 连接客户端发起一个与应用程序（fvAEPg）EPG 内服务器的 TCP 会话，l3extInstP EPG 将在与 fvAEPg EPG 网络服务器开始通信前就将其策略应用到那个客户端端点上。当客户端服务器 TCP 会话结束、客户端与服务器之间的通信结束时，该端点将不再存在于结构之中。

虚拟机管理连接至 VMware vCenter 是使用动态 EPG 配置的一个示例。一旦虚拟机管理域被配置在结构中，vCenter 就启动动态配置 EPG，使得虚拟机端点能够根据需要启动、运动、关闭。

应用配置文件

应用配置文档（fvAp）创立了应用程序需求模型。应用配置文档是针对 EPG 分组的便捷的逻辑容器。下表展示了在管理信息树（MIT）中应用配置文档位于何处，以及他们与租户中其他对象的关系。

表 8：应用配置文档



应用配置文档包含一个或多个 EPG。现代应用包含多个组成部分。例如，一个电子商务应用可能需要网络服务器、数据库服务器、位于存储区域网络的数据、能够实现金融交易的与外部资源的接入。应用配置文档包含所需的尽可能多（或者尽可能少）的 EPG，它们与一个应用的能力有着逻辑关联。

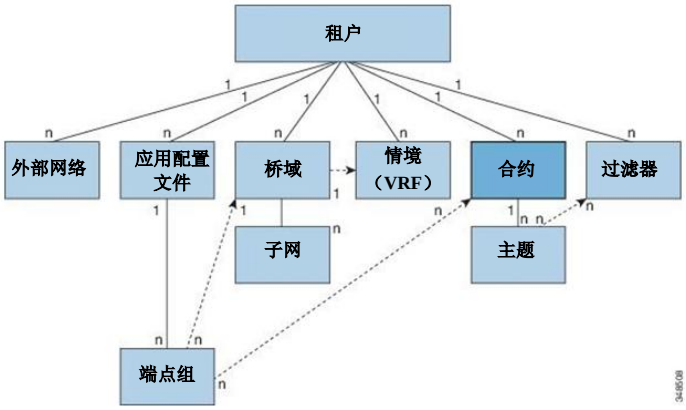
EPG 可以根据以下的某一点进行组织：

- 它们提供的应用（例如附录 A 中示例里的 sap）
- 它们提供的功能（例如基础架构）
- 他们在数据中心机构里位于哪里（例如 DMZ）
- 结构或租户管理员选择使用的无论何种组织原则

合约

除了 EPG 之外，合约（vzBrCP）也是策略模型中的关键对象。EPG 要根据合约规则才能与其他 EPG 通信。下表展示了在管理信息树（MIT）中合约位于何处，以及他们与租户中其他对象的关系。

表 9：合约



管理员使用合约来选择 EPG 之间传递的流量的方式，包括允许的协议和端口。如果没有合约，EPG 间通信是默认禁用的。EPG 内通信不需要合约；EPG 内通信一般是默许的。

合约管理着以下类型的端点组通信：

在 ACI 结构应用 EPG 之间（fvAEPg），包括租户内和租户间。

备注 在共享服务模式的情况下，租户间通信需要合约。合约被用于指定情境中的静态路由，尽管租户情境不强制执行策略。

- 在 ACI 结构应用 EPG 和第 2 层外部的外网实例 EPG（l2extInstP）之间
- 在 ACI 结构应用 EPG 和第 3 层外部的外网实例 EPG（l3extInstP）之间
- 在 ACI 结构带外或带内管理 EPG 之间

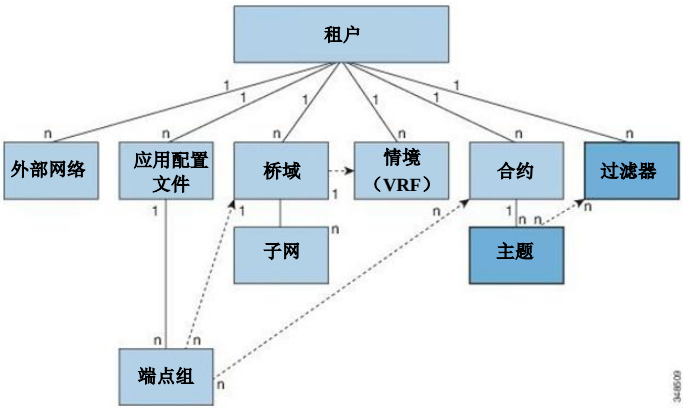
合约管理着被标记为供应者、消费者或两者都有的 EPG 之间的通信。EPG 供应者提供合约，潜在的消费者 EPG 必须要遵守该合约。EPG 与合约之间的关系既可以是供应者，也可以是消费者。当一个 EPG 提供了一个合约，与该 EPG 的通信可以由其他 EPG 发起，只要通信遵守提供的合约。当 EPG 消费了一个合约，那么消费 EPG 内的端点可以与提供该合约的 EPG 内的任何一个端点发起通信。

备注 一个 EPG 可以提供并消费同一个合约。一个 EPG 也可以同时提供并消费多个合约。

标签、过滤器和主题管理 EPG 通信

标签、主题和过滤器管理的对象使得 EPG 和合约之间的混合与匹配能够实现，从而满足各种应用或服务传递需求。下表展示了在管理信息树（MIT）中应用主题和过滤器位于何处，以及他们与租户中其他对象的关系。

图 10：标签、主题和过滤器



合约可以包含多个通讯规则，多个 EPG 可以消费和提供多个合约。标签控制的是一对特定的 EPG 之间通信时应用哪些规则。策略设计者可以简洁地表示复杂的通信策略，并在同一个应用的多个实例中重复使用这些策略。例如，附录 A 中的样例策略展示了同一个合约如何使用标签、主题和过滤器来区分需要 HTTP 或 HTTPS 的不同 EPG 之间怎么发生通信。

标签、主题和过滤器根据以下选项定义 EPG 通信：

- 标签是只有一个属性“名称”的管理对象。标签能够分类，哪些对象可以或是不可以与其他对象通信。标签匹配是首先完成的。如果标签不匹配，就不会处理其他合约或过滤器信息。标签匹配属性可以是以下值中的一个：至少一个（默认）、全部、没有、或只有一个。附录 B 展示了所有这些标签匹配类型及其结果的简单示例。

备注 标签可以被应用到一系列供应者和消费者管理对象上，包括 EPG、合约、桥域、DHCP 中继策略和 DNS 策略。标签不会跨对象类型应用；应用 EPG 上的标签和桥域上的标签没有任何关系。

标签决定着哪个 EPG 消费者和 EPG 供应者可以彼此通信。标签匹配决定着合约的哪个主题被指定的合约的 EPG 供应者或 EPG 消费者所使用。

标签的两种类型如下：

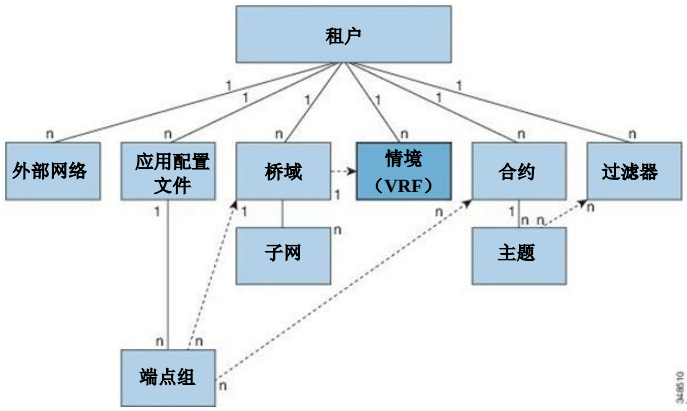
- 应用于 EPG 的主题标签。主题标签匹配使得 EPG 可以选择合约内主题的子集。
- 应用于 EPG 的供应者/消费者标签。供应者/消费者标签匹配使得消费者 EPG 能够选择供应者 EPG，反之亦然。

- 过滤器是第 2 层至第 4 层的字段，TCP / IP 报头字段，例如第 3 层协议类型，第 4 层的端口，等等。根据相关的合约，EPG 供应者决定进出双方向的协议和端口。合约主题包括与过滤器（及其方向）的关联，应用于制造和消费合约的 EPG 之间。
- 主题包含在合约内。合约内的一个或多个主题使用过滤器来指定可以进行通信的流量类型，以及通信如何发生。例如，对于 HTTPS 信息，主题指定方向，过滤器指定 IP 地址类型（例如 IPv4）、HTTP 协议和允许的端口。主题决定过滤器是单向的还是双向的。单向过滤器用于一个方向。单向过滤器定义进入通信或出通信，但两者不一样。双向过滤器对两者而言是一样的；它们既定义进入通信，也定义出通信。

情境

情境（fvCtx）是一种独特的第 3 层转发和应用策略域。下表展示了在管理信息树（MIT）中情境位于何处，以及他们与租户中其他对象的关系。

图 11: 情境



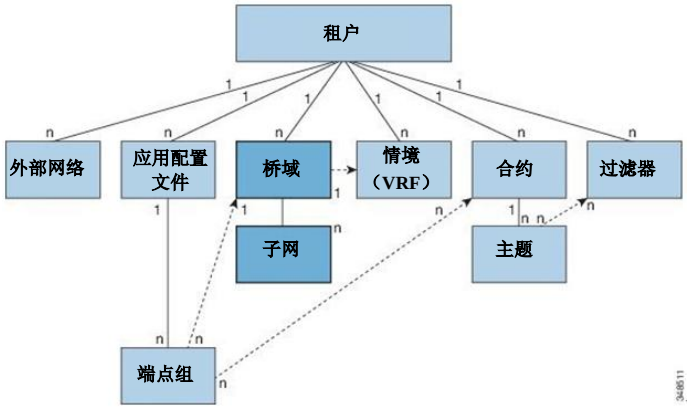
情境定义了第 3 层地址域。一个或多个桥域与情境相关联。第 3 层域中所有的端点都必须拥有唯一的 IP 地址，因为如果策略允许的话，可以直接在这些设备之间转发数据包。一个租户可以包含多个情境。当管理员创建了一个逻辑设备之后，管理员可以创建一个逻辑设备情境，为设备群提供了选择标准策略。逻辑设备可以根据合约名称、图形名称、或是图形内的功能节点名称进行选择。

备注 情境相当于网络世界中的虚拟路由和转发 VRF 实例。

桥域和子网

桥域（fvBD）代表了结构中的第二层（L2）转发构造。下表展示了在管理信息树（MIT）中桥域位于何处，以及他们与租户中其他对象的关系。

图 12: 桥域



一个桥域必须和一个情境相连，并有至少一个与其相关联的子网（fvSubnet）。桥域确定了唯一的第 2 层 MAC 地址空间，以及如果能够激活泛洪时的第 2 层泛洪域。情境确定了唯一的 IP 地址空间，而该地址空间可以包括多个子网。这些子网被定义在一个或多个桥域中，涉及相应的情境。

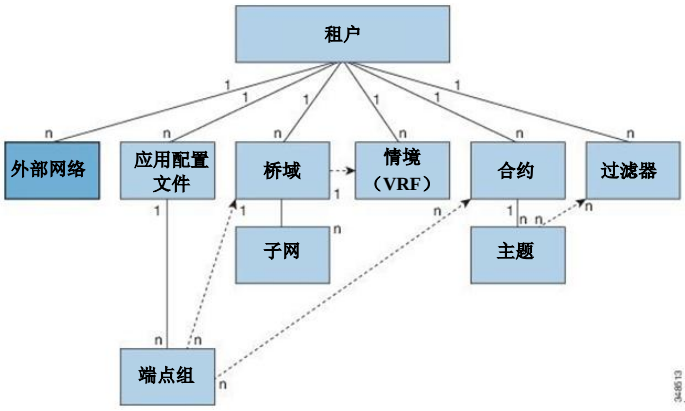
桥域可以跨越多个交换机。一个桥域可以包含多个子网，但一个子网只能被包含在唯一的一个桥域内。子网可以跨越多个 EPG；一个或多个 EPG 可以与一个桥域或子网相关联。

备注 桥域传统模式只允许每个桥域有一个 VLAN。当指定桥域传统模式时，桥域封装被用于所有涉及该桥域的 EPG；如果定义了 EPG 封装，将被忽略。此外，单播路由不适用桥域传统模式。

外部网络

外部网络对象策略控制着与外界的连接情况。一个租户可以包含多个外部网络对象。下表展示了在管理信息树（MIT）中外部网络位于何处，以及他们与租户中其他对象的关系。

图 13：外部网络



外部网络策略规定了相关的第 2 层（l2extOut）或第 3 层（l3extOut）的属性，这控制着外部公共网络或私有网络与 ACI 结构之间的通信。外部设备，例如连接到 WAN 和企业核心的路由器，或是现有的第 2 层交换机，与边缘交换机的前部面板接口相连。提供此类连接功能的边缘交换机被称作边界叶片。连接外部设备的边界边缘交换机接口可以被配置作为桥接接口，也可以被配置作为路由接口。在路由接口的情况下，静态路由或动态路由都可以使用。边界边缘交换机也可以执行正常边缘交换机的功能。

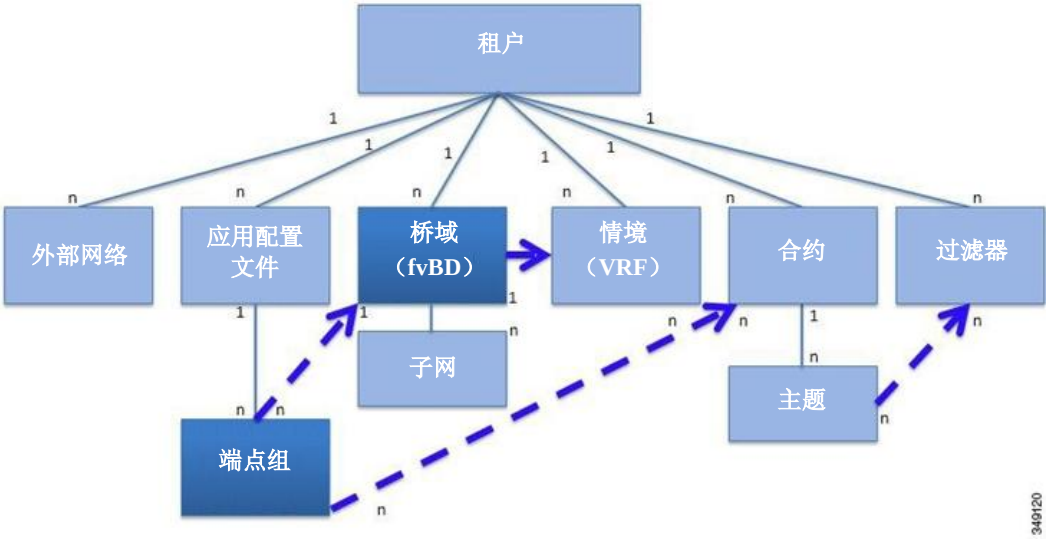
管理对象关系和策略解析

关系管理对象展现了没有共享包容（父子）关系的管理对象实例之间的关系。MO 关系是建立在源 MO 和目标 MO 之间，采用以下两种方式之一：

- 一种明确的关系（fvRsPathAtt），定义的是基于目标 MO 域名（DN）的关系。
- 一种指定的关系，定义的是基于目标 MO 名称的关系。

下表中的虚线展示了几种常见的 MO 关系。

图 14：MO 关系



例如，EPG 和桥域之间虚线表示的是这两个 MO 之间的关系。在该表中，EPG（fvAEPg）包含了一个关系 MO（fvRsBD），采用了目标桥域 MO（fvDB）的名称来命名。例如，如果“生产”是该桥域的名称（tnFvBDName=production），那么关系名称就是“生产”（fvRsBdName=production）。

在策略决议是基于名称关系的情况下，如果在现有租户中没有找到那个拥有匹配名称的目标 MO，那么 ACI 结构将尝试在通用租户内解析。例如，如果用户租户 EPG 包含了一个目标是桥域的关系 MO，而该桥域不存在于租户中，那么系统会尝试在通用租户内解析该关系。如果一个指定的关系既不能在现有租户内解析，也不能在通用租户内解析，ACI 结构会尝试解析到一个默认策略。如果默认策略存在于现有租户内，那么就会使用它。如果不存在，那么 ACI 结构会在通用租户内寻找一个默认策略。桥域、情境和合约（安全策略）指定关系不会解析到默认策略。

跨租户 EPG 通信系统

一个租户内的 EPG 可以与另一个租户内的 EPG 通信，方式是通过共享租户内包含的合约接口。合约接口是一个 MO，可以被不同租户包含的 EPG 作为合约消费接口使用。通过与接口相关联，EPG 消耗了接口代表的主题，成为共享租户内包含的合约。租户可以参与一个单独的合约，该合约也是在同样的第三方被定义。更加严格的安全要求可以通过定义租户、合约、主题和过滤器方向得到满足，这样租户就能保持完全彼此独立。

标记

对象标记简化了 AIP 运行。在 API 运行中，一个对象或一组对象可以通过标记名称来引用，而无需通过专有名称（DN）来引用。标记是它们所标记物体的子对象；除了名称之外，它们没有其他属性。

使用标记给一组对象分配一个描述性的名称。相同的标记名称可以分配给多个对象。多个标记名称可以分配给一个对象。例如，为了能够实现方便、可搜索的接口接入所有的网络服务器 EPG，可以分配一个网络服务器标记给所有这样的 EPG。整个结构的网络服务器 EPG 都可以通过引用网络服务器标记来定位。

标记

第 4 章

ACI 矩阵的基本信息

本章包括以下部分：

关于 ACI 矩阵的基本信息

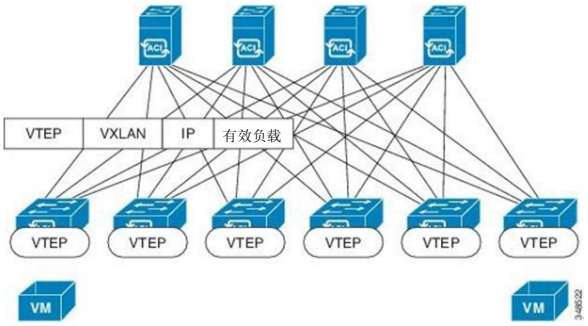
ACI 矩阵支持超过 64,000 专用租户网络。一个矩阵可以支持超过 100 万 IPv4/IPv6 端点，超过 64,000 租户，超过 200,000 10G 端口。ACI 矩阵在任何地点支持任何服务（物理或虚拟），无需额外的软件或硬件网关来连接物理和虚拟服务，该矩阵还可以使用通用路由封装（NVGRE）针对虚拟可扩展局域网（VXLAN）/VLAN/网络虚拟化将封装正常化。

ACI 矩阵将端点身份和相关策略从底层转发图中分离出来。它提供一个分布式第 3 层网关，确保第 3 层和第 2 层转发的最佳效果。该矩阵支持标准桥接和路由语义，没有标准位置约束（任何地方的任何 IP 地址），去掉了 IP 控制层地址解析协议（ARP）/通用属性注册协议（GARP）的洪泛要求。矩阵内的所有流量都封装在 VXLAN 之中。

解耦身份和位置

ACI 矩阵将租户端点地址、标识符从由端点定位符或 VXLAN 通道端点（VTEP）地址定义的端点位置中分离出来。下图显示了解耦身份和位置。

图 15：解耦身份和位置

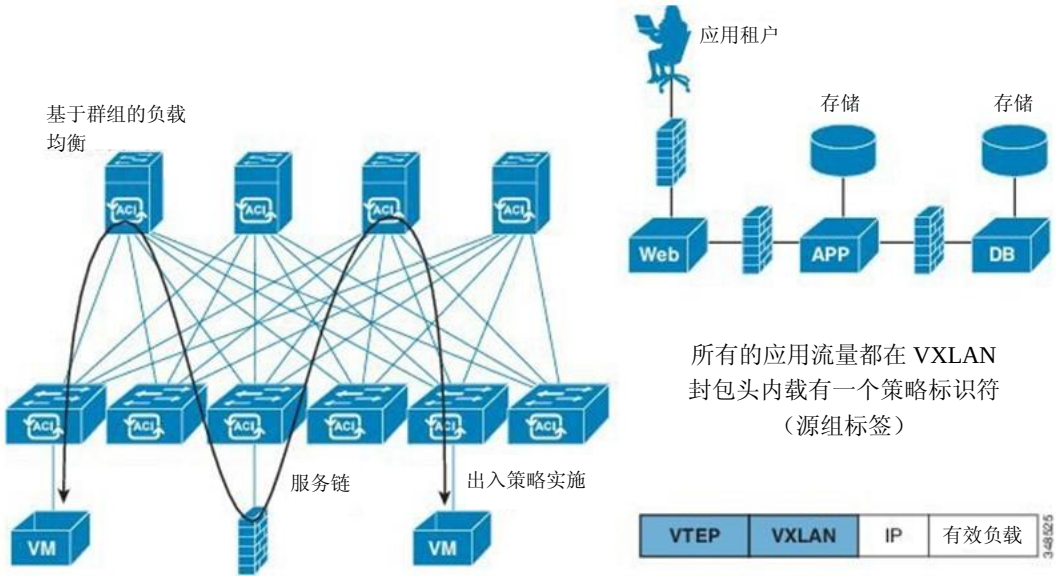


矩阵内的流量位于 VTEP 之中。内部租户 MAC 或 IP 地址的映射由 VTEP 通过分布式映射数据库执行。

策略识别和执行

应用策略通过一个同样在 VXLAN 数据包中运载的独特标签属性从转发中分离出来。策略识别在 ACI 矩阵的每个数据包中进行，这样能以一种完全分布式的方式一致地实施策略。下图对策略识别进行了说明。

图 16：策略识别和执行

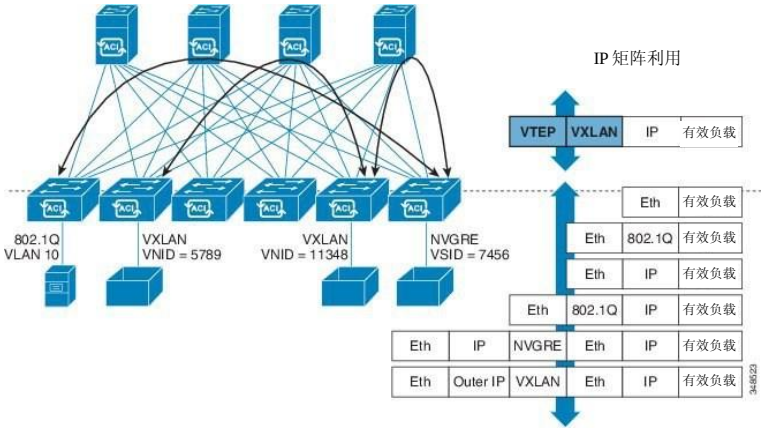


矩阵和接入策略控制内部矩阵和外部接入界面的操作。系统自动创建默认矩阵和接入策略。矩阵管理员（对整个矩阵拥有访问权的人员）可以根据他们的要求修改默认策略或创建全新策略。矩阵和接入策略可以支持不同的功能或协议。APIC 中的选择器使得矩阵管理员能够选择他们将要实施策略的节点和界面。

封装正常化

矩阵内的流量被封装为 VXLAN。外部 VLAN/VXLAN/NVGRE 标签在入口处被映射到内部 VXLAN 标签。下图对封装正常化进行了说明。

图 17：封装正常化



转发不局限于封装类型或封装覆盖网络。外部标识符按照边缘或边缘端口进行本地化，使得在必要时重新利用或翻译成为可能。桥接域转发策略可以被定义为在必要时提供标准 VLAN 行为。

原生 802.1p 和带标签的 EPG

遵守这些指导方针，确保需要未标记数据包的设备在连接 ACI 边缘交换机接入端口时按照预期运行。

备注 每个接入端口仅允许一个 802.1p EPG。

- 当某个接入端口配置一个原生 802.1p 模式的 EPG 时，它的数据包会退出那个未标记端口。
- 当某个接入端口配置多个 EPG 时，一个以原生 802.1p 模式，某些带有 VLAN 标记，所有退出那个接入端口的数据包都以下列方式标记：
  - 对于已原生 802.1p 模式配置的 EPG，数据包被标记为 VLNAN 0。
  - 对于其他 EPG，数据包退出时带有各自的 VLAN 标签。
- 当某个边缘交换机针对一个将要除去标签的 EPG 进行配置时，对于该 EPG 使用的每个端口，数据包都要退出不带标记的交换机。

备注 当以不带标签的方式部署 EPG 时，不要在同一个交换机的其他端口上以带标签的方式部署该 EPG

组播树拓扑

ACI 矩阵支持来自接入端口的单播、组播和广播流量转发。所有来自端点主机的多重目的地流量都在矩阵中作为组播流量进行运载。

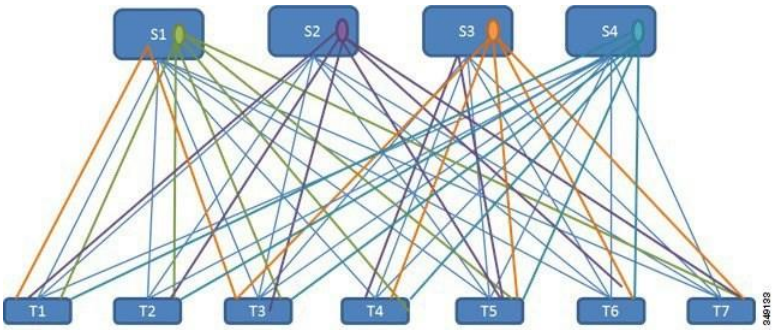
ACI 矩阵包含在 Clos 拓扑（名称取自 Charles Clos）中连接的核心和边缘交换机，在这里进入入口界面的流量可以通过任何可用的中级核心交换机路由到相关的出口交换机。边缘交换机拥有两种端口：连接边缘交换机的矩阵端口和连接服务器、服务设备、路由器和矩阵扩展器（FEX）等的接入端口。

架顶式（ToR）交换机是边缘交换机，连接在核心交换机上。边缘交换机不相互连接，而核心交换机仅与边缘交换机连接。在 Clos 拓扑中，每个较低层的交换机都与每个全网状拓扑中的顶层交换机连接。如果某个核心交换机发生故障，它只能在 ACI 矩阵中轻微地影响性能。选择数据路径的原则是流量负载在核心交换机之间均匀地分布。

ACI 矩阵使用转发标签（FTAG）树平衡多重目的地流量负载。在矩阵中所有多重目的地流量都以封装 IP 组播流量的方式转发。当把流量转发到核心交换机时入口边缘交换机会为流量分配一个 FTAG。分配 FTAG 时它在数据包中作为目标组播地址的一部分。在矩阵中，转发流量时伴随特定的 FTAG 树。核心交换机和任何中级边缘交换机都根据 FTAG ID 转发流量。每个 FTAG ID 创建一个转发树。任何两个节点之间每个 FTAG 仅由一个链接转发。由于使用多个 FTAG，平行链接可与每个 FTAG 配合使用，选择不同的转发链接。矩阵中 FTAG 树的数量越大，负载均衡潜力越佳。ACI 矩阵最多支持 12 个 FTAG。

下图显示了一个带有四个 FTAG 的拓扑结构。矩阵中的每个边缘交换机都直接或通过传输节点连接到每个 FTAG。一个 FTAG 固定在每个核心节点上。

图 18：组播树拓扑



如果边缘交换机直接连接核心交换机，那么它使用直接路径连接 FTAG 树。如果没有直接连接，边缘交换机使用连接到 FTAG 树的传输节点，如上图所示。虽然上图显示了作为 FTAG 树根的每个核心交换机，多个 FTAG 树根可以位于一个核心交换机上。

作为 ACI 矩阵的一部分，FTAG 根被放置在核心交换机上。APIC 通过核心交换机锚定的 FTAG 配置每个核心交换机。根的身份和 FTAG 的数量源自配置。APIC 指定将要使用的 FTAG 树的数量和每个树的根。矩阵中的拓扑结构每改变一次，FTAG 树就会重新计算一次。

根的布置由配置驱动，不会动态地重新植根于运行期间发生的事件（如核心交换机故障）上。通常，FTAG 配置是静态的。当因为管理员决定在剩余的或扩展的核心交换机组之间重新分配 FTAG 而添加或移除核心交换机时可以把 FTAG 从一个核心交换机重新锚定到另一个。

关于流量风暴控制

当数据包涌向 LAN 从而导致过多流量、降低网络性能时，我们称之为流量风暴。用户可以使用流量风暴控制策略防止第 2 层端口被广播、未知组播或未知单播流量风暴中断。

默认情况下，ACI 矩阵中不启用风暴控制。ACI 桥接域（BD）第 2 层位置单播洪泛在桥接域中默认为启用，但管理员可以将其禁用。在那种情况下，风暴控制策略仅应用于广播和未知组播流量。如果在桥接域中启用了第 2 层未知单播洪泛，那么除了广播和未知组播流量，风暴控制策略应用于第 2 层未知单播洪泛。

流量风暴控制（也称为“流量抑制”）允许用户在一秒的间隔内监控流入广播、组播和未知单播的流量水平。在此间隔内，流量水平（既可以表述为端口总可用带宽的百分比或给定端口上每秒钟的最大数据包）与用户配置的流量风暴控制水平进行对比。当入口流量达到端口上配置的流量风暴控制水平时，流量风暴控制会降低流量直到间隔结束。当超过风暴控制阈值时，管理员可以配置监控策略以提高默认值。

风暴控制指南

按照下列指南和限制配置流量风暴控制水平：

- 通常，矩阵管理员在下列接口上配置矩阵接入策略的风暴控制：
  - 常规中继（trunk）接口。
  - 单个边缘交换机上的直接端口通道。
  - 虚拟端口通道（两个边缘交换机上的一个端口通道）。
- 端口通道和虚拟端口通道风暴控制值（每秒钟的数据包或百分比）适用于端口通道的每个成员。不要在作为端口通道成员的接口上配置风暴控制。
- 在按可用带宽的百分比配置时，数值 100 表示没有流量风暴控制，而数值 0.01 表示抑制所有流量。

- 由于硬件的限制和不同尺寸的数据包的计数方法，百分比数值是一种约数。根据组成流入流量的帧大小的不同，实际实施的水平可能与配置水平相差若干个百分点。每秒封包数（PPS）换算为基于 256 比特的百分比。
- 极限突发是没有流量通过时最大积累速率。当流量开始时，首个间隔内不超过积累速率的所有流量都被允许。正在接下来的间隔内，只有不超过配置速率的流量才被允许。支持的最大数值是 65535 KB。如果配置的速率超过该数值，那么对于 PPS 和百分比该数值都是它的最大值。
- 在一个采用优化组播洪泛（OMF）模式的出口边缘交换机，不会应用流量风暴控制。
- 在一个采用非 OMF 模式的出口边缘交换机，会应用流量风暴控制。
- 在一个 FEX 边缘交换机上，在面向主机的接口上无法使用流量风暴控制。

负载均衡

ACI 矩阵提供多个负载均衡选项，用于平衡可用上行链接之间的流量。在每个流量按照 5 元组哈希分配到一个上行链路中的网络中，静态哈希负载均衡是传统负载均衡机制。这种负载均衡在大体均匀的可用链接之间分配数据流。通常，在有大量数据流时，数据流的均匀分配会使得带宽也均匀分配。但是，如果某几个数据流远大于其他的数据流，那么静态负载均衡可能会导致次优结果。

动态负载均衡（DLB）按照拥塞水平调节流量分配。它会测量各个可用路径的拥塞情况，然后把数据流放置在最不拥堵的路径中，使数据放置最优或接近最优。

可以通过配置 DLB 使用数据流或流簇的粒度将流量放置到可用的上行链路上。流簇是来自被大小合适的时间间隔分开的数据流的封包的突发。如果数据包两个突发之间的空闲间隔大于可用路径之间的延迟的最大差，那么第二个爆发（或流簇）可以沿着第一个之外的另一条路径发送，而不会对数据包再次排序。使用一个名为“流簇计时器”的计时器测量空闲间隔。流簇提供一个更高粒度的数据流替代品用于平衡负载而不会导致数据包再排序。

DLB 的运行模式要么激进要么保守。这些模式与用于流簇计数器的超时值有关。激进模式的流程超时是一个相对较小的数值。这个粒度非常细的负载均衡理想适用于流量分配，但可能发生数据包重新排序。但是，应用性能的总体获益等同于或优于保守模式。保守模式的流簇超时是一个更大的数值，可保证数据包不会被重新排序。折中方案是采用粒度更大的负载均衡，因为新的流簇机率更低。DLB 无法总能提供最优的负载均衡，但不会劣于静态哈希负载均衡。

当可用链接的数量因为链接下线或上线而变化时，ACI 矩阵会调整流量。矩阵在新链接组之间重新分配流量。

在负载均衡的所有模式中，不论静态或动态，流量只在符合等价多路径（ECMP）标准的上行链路或路径中发送；从路由角度看，这些路径相等且成本最低。

动态封包优先级（DPP）不是一种负载均衡技术，它使用 DLB 在交换机中采用的某些机制。DPP 配置不包括 DLB。按照 DPP 的规则，短数据流的优先级要高于长数据流；短数据流大约不超过 15 个封包。相比长数据流，短数据流对延迟更加敏感。DPP 可以改善应用总体的性能。

ACI 矩阵默认设置使用传统的静态哈希。静态哈希功能在上行链路之间向核心交换机分配来自边缘交换机的流量。当某个链路故障或上线时，所有链路上的流量按照新的上行链路数量进行重新分配。

端点保留

在交换机中保留缓存的端点 MAC 或 IP 地址能够提升性能。交换机会在端点激活时了解端点。本地端点位于本地交换机上。远程端点位于另一个交换机上，但缓存在本地。边缘交换机存储直接连接这些交换机（或通过某个直接连接第 2 层交换机或矩阵扩展器）的端点的、本地端点的连接矩阵上其他边缘交换机的端点（硬件中的远程端点）的本地和策略信息。交换机针对本地端点使用 32K 缓存条目，针对远程端点使用 64K 缓存条目。

在边缘交换机上运行的软件主动管理这些表项。对于本地连接的端点，在每个条目的保留计时器到期后软件超期清除这些条目。端点活动停止、端点位置移动到另一个交换机或生命周期状态变为下线时，会从交换机中删除端点条目。本地保留计时器的默认值为 15 分钟。在删除某个非活动条目时，边缘交换机向端点发送三个 ARP 请求，确认是否已经离开。对于远程连接的端点，交换机在进入非活动状态 3 分钟后删除条目。如果再次激活，远程端点会立即重新进入表项。若表项中没有远程端点，不会有性能代偿，除非策略在远程边缘交换机中执行，直到端点再次缓存。

端点保留计时器策略可被更改。配置静态端点 MAC 和 IP 地址可以通过将保留计时器设置为零永久性地将其存储在交换机缓存中。将某个条目的保留计时器设置为零意味着不会删除该条目。进行此项操作时必须十分小心。如果端点移动或其测量发生改变，必须通过 APIC 利用更新信息刷新该条目。当保留计时器不是零时，几乎立即在每个封包上检查并更新该信息而不会有 APIC 干预。

端点保留策略决定删除方式。对大多数操作使用默认策略算法。更改端点保留策略可以影响系统性能。如果某个交换机与数以千计的端点通讯，降低超时间隔增加了缓存窗口的数量，从而为大量活动端点提供支持。当端点超过 10,000 时，我们建议在多个交换机上分配端点。

ACI 矩阵的安全策略模型

ACI 矩阵的安全策略模型基于合约。这种方法解决了传统访问控制列表（ACL）的限制。合约包含了在端点组之间流量上执行的安全策略的规范。

EPG 通讯需要一个合约；没有合约不允许进行 EPG 到 EPG 的通讯。APIC 将整个策略模型（包括合约及其相关的 EPG）转化为每个交换机的物理模型。进入时，进入矩阵的每个封包都带有所需的策略详细信息。因为需要合约选择能够通过 EPG 之间的流量的类型，所以合约必须执行安全安全策略。在传统网络环境下，合约满足访问控制列表（ACL）处理的安全要求，因此合约是更加灵活、容易管理且全面的安全策略解决方案。

访问控制列表限制

传统的访问控制列表（ACL）带有若干 ACI 矩阵安全模式可以解决的限制。传统的 ACL 非常紧密地与网络拓扑耦合。它们通常在每个路由器或交换机入接口和出接口上配置，可以根据接口和预期流过这些接口的流量进行自定义。由于自定义的关系，它们常常无法跨接口使用，更不用说跨路由器或交换机。

传统的 ACL 非常复杂、神秘，因为它们包含了特定 IP 地址、子网和协议的列表，这些列表有些是被允许的，有些是不被允许的。这种复杂性意味着它们不易维护，而且管理员由于害怕造成问题而不愿意删除任何 ACL 列表规则，所以列表越来越臃肿。正因为这种复杂性，这些列表通常仅在网络中的特定分界点（如 WAN 和企业之间的分界或 WAN 和数据中心之间的分界）部署。在这种情况下，企业内部或对于数据中心包含的流量，ACL 的安全优势并没有被开发利用。

另一个问题是单个 ACL 中条目的数量可能大幅增加。用户经常希望创建一个 ACL，通过使用一组协议让一组资源与一组目标进行沟通。在最坏的情况下，如果 *N* 个资源与 *M* 个目标使用 *K* 个协议进行对话，那么 ACL 中可能有 *N***M***K* 行。ACL 必须列出针对每个协议与每个目标进行通讯的每个资源。在设备和协议不多的情况下 ACL 也可能非常庞大。

ACI 矩阵安全模型可以解决这些 ACL 问题。ACI 矩阵安全模型直接表达了管理员的意图。管理员使用合约、过滤器和标签管理的对象规定服务器组通讯的方式。这些被管对象不依赖于网络拓扑，因为不将它们应用于特定接口。它们知识网络必须执行的规则，无论服务器组连接的位置在哪里。这种拓扑独立性意味着可以在整个数据中心轻松部署和重复利用这些被管对象，不止作为特定分界点。

ACI 矩阵安全模型直接使用端点分组构造，因此让服务器组相互通讯的理念非常简单。一个单一规则可允许任意数量资源与同样任意数量的目标进行沟通。这种大小上的缩减显著改善规模和可维护性，也就是说，它们更加易于在数据中心使用。

合约包含安全策略规范

在 ACI 安全模型中，合约包含控制在 EPG 之间进行通讯的策略。合约规定允许通讯的内容，而 EPG 规定通讯的源地址和目标地址。合约连接 EPG，如下图所示。

EPG 1----------合约---------- EPG 2

EPG 1 中的端点能与 EPG 2 中的端点通讯，如果合约允许，反过来也可以。策略结构非常灵活。EPG 1 和 EPG 2 之间可以有很多合约，使用合约的 EPG 可以超过两个，而合约可以在多个 EPG 组之间再次使用。

EPG 和合约之间的关系中也存在方向性。EPG 可以提供或使用一个合约。提供合约的某个 EPG 通常是向一组租户设备提供某种服务的一组端点。该服务使用的协议在合约中进行了定义。使用合约的某个 EPG 通常是作为该服务租户的一组端点。当租户端点（服务使用方）试图连接某个服务端点（服务提供方）时，合约会检查连接是否被允许。除非另有规定，该合约将不允许服务器向租户发起连接。但是，EPG 之间的另一个合约可以比较容易地支持上述方向的连接。

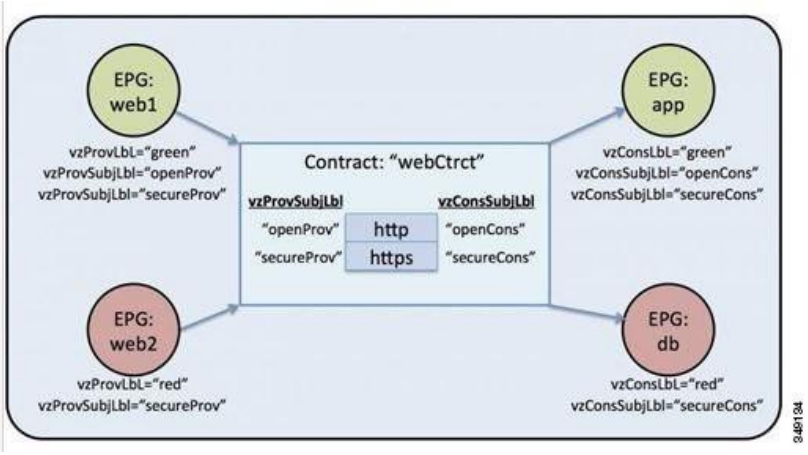
这种供给/使用关系通常用 EPG 和合约之间的箭头表示。注意下图中箭头的方向。

EPG 1 <--------使用-------- 合约 <--------提供-------- EPG 2

合约以层级形式构成。它包括一个或多个主题，每个主题包含一个或多个过滤器，每个过滤器能够定义一个或多个协议。

下图说明了合约如何控制 EPG 通讯。

图 19：合约决定 EPG 与 EPG 之间的通讯



例如，你可以定义一个名为“HTTP”、规定 TCP 端口 80 和端口 8080 的过滤器和另一个名为“HTTPS”、规定 TCP 端口 443 的过滤器。然后你可以创建一个拥有两个主题组、名为“webCtrct”的合约。openProv 和 openCons 是包含 HTTP 过滤器的主题。secureProv 和 secureCons 是包含 HTTPS 过滤器的主题。webCtrct 合约可以被用作允许两类 EPG——提供网页服务的 EPG 和包含希望使用该服务的端点的 EPG——之间的安全和不安全网页流量。

这些相同的构造也适用于控制虚拟机管理程序的策略。当把某个 EPG 置于虚拟机管理器（VMM）域的时候，APIC 把与 EPG 相关的所有策略都下载到接口与 VMM 域相连的边缘交换机。如需查看对 VMM 域的完整解释，请参考《ACI 基础知识》手册中的《虚拟机器管理器域》一章。当创建该策略时，APIC 将其推送到规定哪个交换机允许 EPG 端点连接性的 VMM 域中。VMM 域定义了允许 EPG 端点连接的交换机和服务器组。当某个端点上线时，它与对应的 EPG 发生联系。当该端点发送封包时，源 EPG 和目标 EPG 来自该封包，而相应合约规定的策略会被检查，看该封包是否被允许。如果“是”，封包会被转发。如果“否”，封包会被丢掉。

合约还支持更复杂的操作，不只是“允许”或“拒绝”。合约可以规定匹配给定主题的流量被重新发送给某个服务，被复制，或其 QoS 级别被修改。物理模型中的接入策略被预先填充之后，端点可以移动，新端点可以上线，通讯可以发生——即使 APIC 下线或由于其他原因而无法接入。APIC 不再是网络中的一个故障点。封包进入 ACI 矩阵时，在交换机中运行的物理模型可以执行安全策略。

安全策略执行

流量从前面板接口进入边缘交换机时，封包被标注为带有源 EPG 的 EPG。然后边缘交换机在租户空间内的封包目标 IP 地址上执行正向搜索。“命中”可能导致下列任何一种场景：

1 单播（/32）命中提供目标端点的 EPG 和目标端点所在的本地接口或远程边缘交换机 VTEP IP 地址。

2 子网前缀的单播（/32）命中提供目标子网前缀的 EPG 和目标子网前缀所在的本地接口或远程边缘交换机 VTEP IP 地址。

3 组播命中提供本地接收器的本地接口和外部目标 IP 地址，以便在矩阵和组播组 EPG 的 VXLAN 封装中使用。

备注 组播和外部路由器子网始终在入口边缘交换机上“命中”。一旦入口边缘交换机获悉目标 EPG，即开始执行安全策略。

转发表中的“失效”结果会导致封包被发送到核心交换机的转发代理。然后转发代理就会执行一个转发表搜索操作。如果“失效”，封包会被丢掉。如果“命中”，封包就会被发送到包含目标端点的出口边缘交换机。由于出口边缘交换机知道目标的 EPG，它会强制执行安全策略。出口边缘交换机也必须知道封包源的 EPG。矩阵首标使得这种过程成为可能，因为它把 EPG 从入口边缘交换机运载到出口边缘交换机。核心交换机在执行转发代理功能时会在封包中保留原始 EPG。

在出口核心交换机上，源 IP 地址、源 VTEP 和源 EPG 信息都通过学习被存储在本地转发表中。因为大多数数据流都具有双向性，一个返回封包会在数据流两端填充转发表，使得流量在两个方向经过入口过滤。

组播和 EPG 安全

组播流量会产生一个有趣的问题。对于单播流量，通过检查封包的目标地址而能够清楚地知道目标 EPG。但是，对于组播流量，目标地址是一个抽象的实体：组播组。封包的源地址从来都不是一个组播地址，因此源 EPG 的确定方式一如上文的单播示例。目标组的来源是组播的区别所在。

组播组有些独立于网络拓扑，因此允许将(S, G)和(*, G)静态配置到组绑定。当把组播组放置于转发表中时，对应组播组的 EPG 也被放置在转发表中。

备注 本文把“组播流”称为“组播组”。

边缘交换机总是把对应组播流的组视为目标 EPG 而不是源 EPG。在上文所示的接入控制矩阵中，当组播 EPG 为源地址时，行目录无效。发送到组播流的流量来自组播流的源地址或希望加入组播流的目标地址。组播流必须位于转发表中且流内没有分层寻址，因此在入口矩阵边缘组播流量存在访问控制。如此一来，IPv4 组播始终作为入口过滤被强制执行。

组播流的接收方在接收流量前必须首先加入组播流。发送 IGMP Join 请求时，组播接收方实际上是 IGMP 封包的源地址。目标地址被定义为组播组，而目标 EPG 检索自转发表。在路由器接收 IGMP Joint 请求的入口点上会施加访问控制。如果 Joint 请求被拒绝，接收端不会从特定组播流中接收任何流量。

组播 EPG 策略强制实施由边缘交换机按照上文所述的合约规则在入口处执行。EPG 捆绑的组播组由 APIC 推送到所有包含特定租户（VRF）的边缘交换机。

禁忌

虽然确保安全的正常流程仍旧适用，但会使用旨在保证任何安全惯例完整性的 ACI 策略模型辅助。在 ACI 策略模型方法中，所有通讯都必须符合下列条件：

- 通讯是否被允许取决于合约，而合约是模型中的被管对象。如果没有合约，EPG 间的通讯默认为禁用。
- 无法直接介入硬件；所有交互都通过策略模型接受管理。

禁忌是网络管理员可用来拒绝特定类别流量的模型中被合约管理的特殊对象。禁忌可用于丢掉匹配某一类型（任何 EPG、特定 EPG、匹配某一过滤器等）的流量。禁忌的规则在城规合约规则实施之前在硬件中使用。

第 5 章

矩阵配置

本章包括以下部分：

矩阵配置

相比传统的交换基础架构，思科以应用为中心的基础架构（ACI）自动化和自动配置提供了下列操作优势：

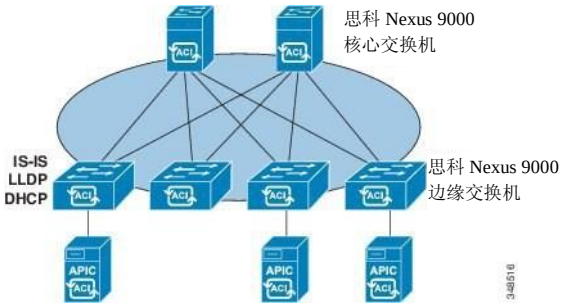
- 一个集群化的、逻辑上集中化但物理上分布式的 APIC 为整个矩阵提供了策略、引导程序和映像管理。
- APIC 启动拓扑自动发现、自动配置和基础架构寻址使用下列行业标准协议：中间系统到中间系统（IS-IS）、链路层发现协议（LLDP）和动态主机配置协议（DHCP）。

- APIC 提供了简单、自动、以策略为基础的配置和升级流程以及自动化的映像管理能力。
- APIC 提供了可扩展的配置管理功能。ACI 数据中心可以非常大，因此单独配置交换机或接口无法很好地扩展，即使使用脚本。APIC pod、控制器、交换机、模块和接口选择器（所有、系列、特定实例）在整个矩阵上实现了对称配置。应用对称配置时，管理员确定在一个策略组中与接口配置相关的交换机配置文件。

启动发现和配置

集群化的 APIC 控制器向矩阵提供 DHCP、引导程序配置和映像管理，从而实现自动化启动和升级。下图对启动发现进行了说明。

图 20：启动发现配置



思科 Nexus ACI 矩阵软件作为 ISO 映像捆绑，可以通过管理控制台安装在思科 APIC 服务器上。思科 Nexus ACI 软件 ISO 包含思科 APIC 映像、边缘节点的固件映像、核心节点的固件映像、默认矩阵基础架构策略和运行所需的协议。

ACI 矩阵引导程序顺序在所有交换机使用原厂安装的映像引导时开始。运行 ACI 固件和 APIC 的思科 Nexus 9000 系列交换机在引导过程中使用保留层。这种基础架构空间硬编码在交换机中。APIC 能够通过默认层连接边缘交换机，或使用一个本地重要的标识符。

ACI 矩阵使用一个基础架构空间，后者安全地孤立在矩阵中，是执行所有拓扑发现、矩阵管理和基础架构寻址的地方。矩阵中的 ACI 矩阵管理通讯通过基础架构内部私有 IP 地址空间进行。这种寻址方案允许 APIC 与矩阵节点和集群中的其他思科 APIC 控制器通讯。APIC 使用基于链路层发现协议（LLDP）的发现过程发现集群中其他思科 APIC 控制器的 IP 地址和节点信息。

APIC 集群发现过程如下：

- 思科 ACI 中的每个 APIC 都使用一个内部私有 IP 地址与 ACI 节点和集群中的其他 APIC 通讯。APIC 使用基于 LLDP 的发现过程发现集群中其他 APIC 控制器的 IP 地址。
- APIC 拥有一个设备向量（AV），该向量提供了从某个 APIC ID 到某个 APIC IP 地址和 APIC 通用唯一标识符（UUID）的映射。最初，每个 APIC 都从一个充满本地 IP 地址的 AV 开始，所有其他 APIC 插槽都标记为未知。

- 某个交换机重启时，边缘节点设备上的策略元素（PE）从 APIC 取得其 AV。然后交换机将此 AV 向所有相邻设备公布并向本地 AV 中的所有 APIC 报告本地 AV 和相邻设备 AV 的差别。

通过该过程，APIC 通过交换机学习到了 ACI 中的其他 APIC 控制器。在集群中验证了这些新发现的 APIC 之后，APIC 控制器更新它们本地的 AV 并用新的 AV 对交换机编程。然后交换机开始公布这个新的 AV。所有交换机拥有相同 AV，所有 APIC 控制器知道所有其他 APIC 控制器的 IP 地址之后，上述过程结束。

ACI 矩阵以层叠形式初启，从直接与 APIC 相连的边缘节点开始。LLDP 和控制平板的 IS-IS 融合与引导过程并行发生。ACI 矩阵使用基于 LLDP 和 DHCP 的矩阵发现功能自动发现矩阵交换机节点，分配基础架构 VXLAN 通道端点（VTEP）地址，在交换机上安装固件。在这个自动过程之前，必须在思科 APIC 控制器上执行最小限度的引导程序控制。

集群管理指南

APIC 集群包含多个 APIC 控制器，为操作员提供了针对 ACI 矩阵的统一实时监控、诊断和配置管理能力。为确保最佳的系统性能，在对 APIC 集群进行更改时遵循下列指南：

备注 在对集群进行更改前，务必核实其是否健康。在对集群实施计划之中的更改时，集群中的所有控制器都应当健康。如果集群中的一个或多个 APIC 控制器不健康，在采取行动前纠正该状况。解决 APIC 集群健康问题时，请参考“思科 APIC 故障排除指南”以获得更多信息。

管理集群时遵守下列通用指南：

- 忽略来自那些当前不在集群中的 APIC 的信息，这些 APIC 提供的集群信息不精确。
- 集群插槽中包含一个 APIC ChassisID。一旦用户配置一个插槽，在用户使用分配的 ChassisID 停止 APIC 之前，插槽处于不可使用状态。
- 如果 APIC 固件正在升级之中，在对集群进行其他更改之前，等待升级完成和集群完全就绪。

扩大 APIC 集群规模

扩大 APIC 集群规模时请遵守下列指南：

- 当矩阵工作负载不会受到集群扩展影响时，规划集群扩展的日程。
- 按照硬件安装指南中的说明逐步实施新的 APIC 控制器。确认与 PING 测试的带内连接性。
- 增加集群目标规模，使其与现有集群规模控制器计数和新控制器计数相等。例如，如果当前的集群规模控制器计数为 3 且你正在增加 3 个控制器，那么将新集群的目标规模设置为 6.集群开始按顺序增加规模，一次一个控制器，直到所有新控制都包含在了集群中。

备注 如果当前 APIC 控制器变得不可用，集群扩展停止。开始进行集群扩展之前解决这个问题。

- 根据增加每个设备时 APIC 必须同步的数据量，完成扩展所需的时间可能大于 10 分钟/台设备。成功扩展集群之后，APIC 操作规模等于目标规模。

备注 在对集群进行额外的更改之前，让 APIC 完成集群扩展。

在集群中更换 APIC 控制器

更换 APIC 控制器时遵守下列指南：

- 记录将要更换的 APIC 控制器的 ID 编号。
- 停止将要更换的 APIC 控制器。

备注 若在更换前未停止 APIC 控制器，集群将无法吸收替换后的控制器。

- 按照硬件安装指南中的说明逐步更换 APIC 控制器。确认与 PING 测试的带内连接性。
- 向 APIC 集群增加替换控制器时，向替换 APIC 控制器分配之前使用的 APIC 控制器 ID 编号。APIC 将使替换控制器与集群同步。

备注 如果当前 APIC 控制器变得不可用，集群同步停止。开始同步集群之前解决这个问题。

- 根据更换控制器时 APIC 必须同步的数据量，更换每个控制器所需的时间可能大于 10 分钟。替换控制器与集群成功同步后，APIC 运行规模和目标规模将保持不变。

备注 在对集群进行额外的更改之前，让 APIC 完成集群同步。

- 当矩阵工作负载不会受到集群同步影响时，规划集群更换的日程。

- UUID 和矩阵域名在重启时在 APIC 控制器中存留。但是，恢复出厂设置的重启会删除该信息。如果要将某个 APIC 控制器从一个矩阵移动到另一个，必须在将该控制器添加到另一个 ACI 矩阵之前恢复出厂设置。

减小 APIC 集群规模

按照下列指南减少 APIC 的集群规模并停止从集群中删除的 APIC 控制器：

备注 如果未能有序地停止并关闭减小集群中的 APIC 控制器，可能会发生无法预知的后果。不要让未识别的 APIC 控制器与矩阵连接。

- 减小集群尺寸会增加剩余 APIC 控制器上的负载。当矩阵工作负载不会受到集群同步影响时，规划缩小 APIC 控制器规模的日程。
- 把集群目标规模减小到另一个更低的数值。例如，如果当前集群的规模是 6 且你将删除 3 个控制器，那么将集群的目标规模减少到 3。
- 从现有集群中 ID 编号最高的控制器开始，逐一停止、关闭并断开 APIC 控制器，直到集群达到另一个更低的目标规模。

  停止并删除每个控制器后，APIC 使集群同步。
- 备注如果当前 APIC 控制器变得不可用，集群同步停止。开始同步集群之前解决这个问题。
- 根据删除控制器时 APIC 必须同步的数据量，为每个控制器停止和完成集群同步所需的时间可能大于 10 分钟。

备注 完成所有必需的停止步骤，在对集群进行额外的更改之前，让 APIC 完成集群同步。

矩阵组件

策略模型包含一个完整的矩阵实时组件，包括所有节点和接口。这种组件功能可以实现配置、故障排除、编辑和监控的自动化。

对于思科 ACI 矩阵交换机，矩阵成员节点组件包含识别节点 ID、序列号和名称的策略。第三方节点被记录为未被管理的矩阵节点。思科 ACI 交换机可被自动发现，或者它们的策略信息可被导入。策略模型也会维护矩阵成员节点的状态信息。

节点状态	状况
未知	无策略。所有节点都需要一个策略；没有策略，成员节点状态处于未知状态。
发现	显示节点正在被发现的暂态。
未被发现	节点有策略，但从未在矩阵中初启。
不受支持	节点是一个思科交换机，但不受支持。例如，固件版本不与 ACI 矩阵兼容。
停止	节点具有策略，已被发现，但某个用户将其禁用。该节点可被重新激活。 **备注** 如果在停止某个边缘交换机时指定清除选项，那么 APIC 会试图在边缘交换机和 APIC 上删除所有边缘交换机配置。如果边缘交换机不可及，那么只有 APIC 被清除。在这种情况下，用户必须通过重置手动清除边缘交换机。
未激活	该节点不可及。它已经被发现，但当前无法接入。例如，它可能被关闭，或者电缆被断开。
激活	该节点是矩阵的激活成员。

被禁用的接口可能被管理员列入黑名单或由于 APIC 检测到异常现象而被拆除。链路状态异常例如：

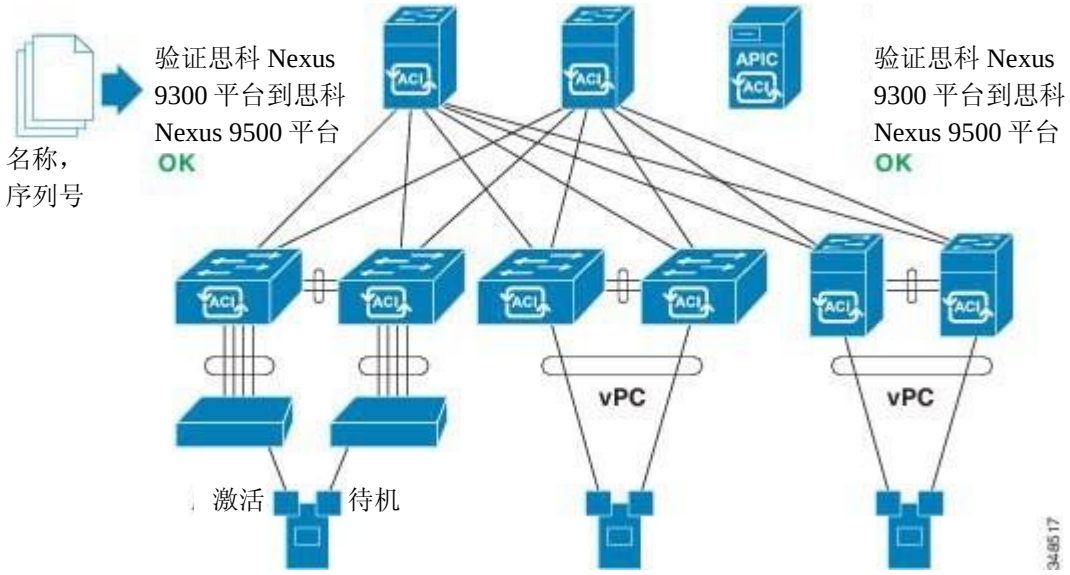
- 线路不匹配，如核心设备连接到核心设备、边缘节点设备连接到边缘节点设备、核心设备连接到边缘接入端口、核心设备连接到非 ACI 节点或边缘矩阵端口连接到非 ACI 设备。
- 矩阵名称不匹配。矩阵名称存储在每个 ACI 节点。如果某个节点移动到另一个矩阵而没有进行恢复出厂默认状态，那么它将保留矩阵名称。
- UUID 不匹配可导致 APIC 禁用节点。

备注 如果管理员使用 APIC 禁用核心设备上的所有边缘节点，那么需要重启核心设备才能恢复对核心设备的接入。

配置

APIC 配置方法自动通过合适的连接初启 ACI 矩阵。下图对矩阵配置进行了说明。

图 21：矩阵配置



链路层发现协议（LLDP）发现动态学习所有邻近的连接之后，会对比某个宽松的规范规则（如"LEAF 仅能连接 SPINE-L1-*"，或者"SPINE-L1-*可以连接 SPINE-L2-*或 LEAF"）对这些连接进行验证。如果出现规则不匹配的情况，那么会发生故障，连接被阻止。此外，会创建一个警报，表示需要注意连接。思科 ACI 矩阵管理员可以把所有矩阵节点的名称和序列号从文本文件导入 APIC 或允许矩阵自动发现序列号，然后使用 APIC GUI、命令行界面（CLI）或 API 把名称分配到节点。

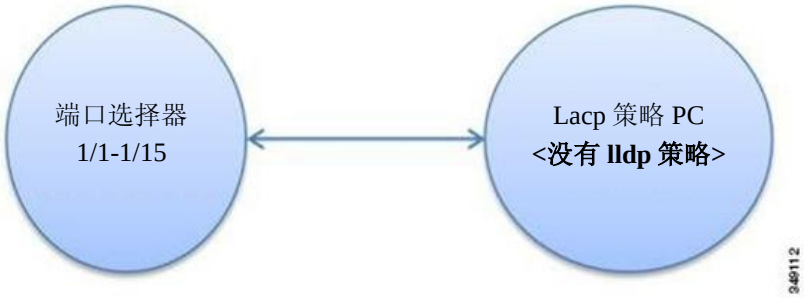
默认策略

APIC 默认策略值的初始值从加载到交换机的物理模型中取得。矩阵管理员可以修改默认策略。默认策略发挥多重功能：

1 允许矩阵管理员覆盖模型的默认值。

2 如果管理员没有提供明确的策略，APIC 会使用默认策略。管理员可以创建默认策略，除非管理员提供了明确的策略，否则 APIC 使用默认策略。

图 22：默认策略



例如，根据管理员采取或未采取的动作，APIC 会进行如下操作：

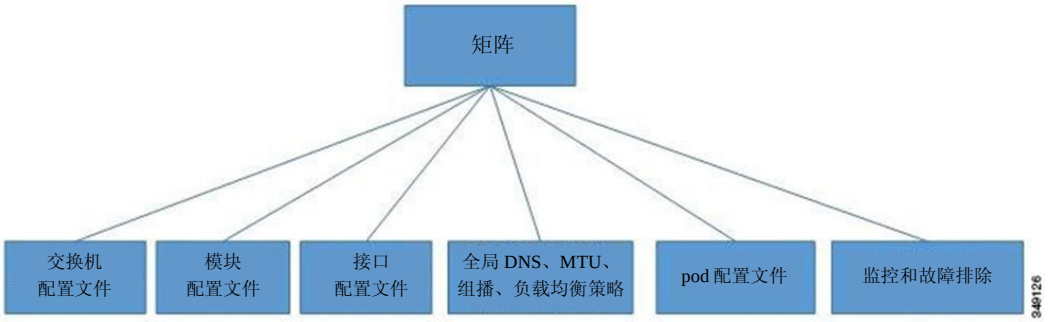
- 管理员没有为选定的端口指定 LLDP 策略，APIC 针对端口选择器中指定的端口使用默认 LLDP 接口策略。
- 如果管理员从端口选择器中删除端口，那么 APIC 对该端口使用默认策略。在上例中，如果管理员从端口选择器中删除端口 1/15，那么端口不再是端口通道的一部分，APIC 对该端口使用所有的默认策略。

当 ACI 矩阵升级时，现有策略默认值仍保留，即使在更新的版本中默认值发生改变。当节点首次连接 APIC 时，节点
在向该节点推送所有默认策略的 APIC 上注册自身信息。默认策略的任何变化都会推送到节点。

矩阵策略概览

矩阵策略控制内部矩阵接口的操作，启用连接核心和边缘交换机的各种功能、协议和接口。拥有矩阵管理员权限的管理员可以按照自己的要求创建新的矩阵策略。APIC 可以让管理员选择他们将要应用矩阵策略的 pod、交换机和接口。下图概括描述了矩阵策略模型。

图 23：矩阵策略概览



矩阵策略被划分到下列类别中：

- 交换机配置文件规定配置哪些交换机以及交换机的配置策略。
- 模块配置文件规定配置哪些核心交换机模块以及核心交换机的配置策略。
- 接口配置文件规定配置哪些矩阵接口以及接口的配置策略。
- 全局策略规定 DNS、矩阵 MTU 默认值、组播树和将要在整个矩阵中使用的负载均衡器配置。
- pod 配置文件规定日期、时间、SNMP、协作密钥服务器（COOP）、IS-IS 和边界网关协议（BGP）路由反射器策略。
- 监控和故障排除策略规定监控对象、阈值、处理故障和日志的方法和执行诊断的方法。

矩阵策略配置

矩阵策略配置连接核心和边缘交换机的接口。矩阵策略可以启用监控（统计信息收集和统计信息导出）、故障排除（按需诊断和 SPAN）、IS-IS、协作密钥服务器（COOP）、SNMP、边界网关协议（BGP）路由反射器、DNS 或网络时间协议（NTP）。

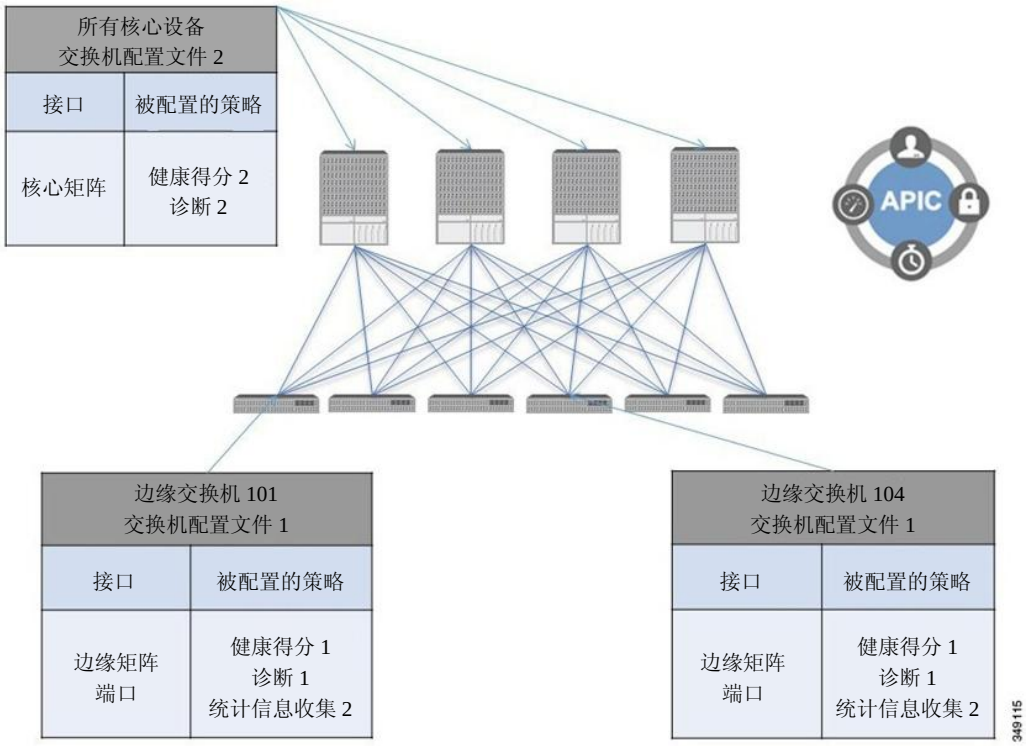
在整个矩阵中应用某个配置时，管理员只用一步即可将确定的策略组关联到交换机的接口上。这样，矩阵上的大量接口可以一次性配置；一次配置一个端口不可扩展。下图显示了上述过程如何在配置 ACI 矩阵中发挥作用。

矩阵策略配置

图 24：矩阵策略配置过程

下图说明了针对 ACI 矩阵应用交换机配置文件 1 和交换机配置文件 2 的结构。

图 25：矩阵交换机策略的应用

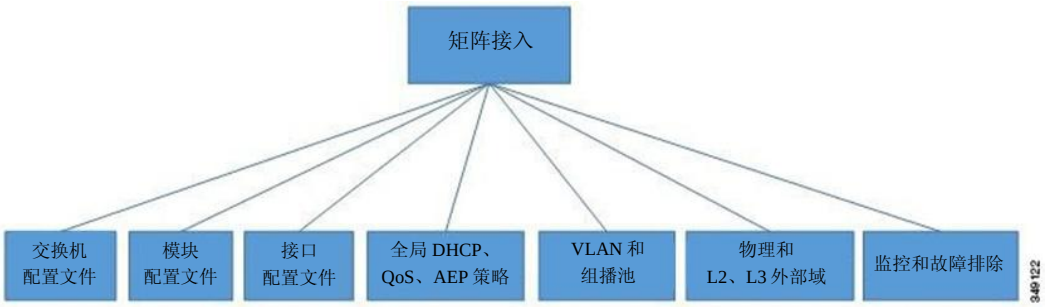


基础架构和范围的这种组合使得管理员能够以一种可扩展的方式管理矩阵配置。可以使用 REST API、CLI 或 GUI 实施这些配置。GUI 中的 Quick Start Fabric Interface Configuration 向导能够自动创建必要的基础对象用以实施此类策略。

访问策略概览

访问策略配置连接虚拟机器控制器、管理程序、主机、连接网络的存储、路由器或矩阵扩展器（FEX）接口。接口策略可以启用端口通道和虚拟端口通道的配置、协议（如链路层发现协议（LLDP）、思科发现协议（CDP）或链路聚合控制协议（LACP））和功能（如统计信息收集、监控和诊断）。下图概括描述了访问策略模型。

图 26：访问策略模型概览



访问策略被划分为下列类别：

- 交换机配置文件规定配置哪些交换机以及交换机的配置策略。
- 模块配置文件规定配置哪些边缘交换机接入卡和接入模块以及边缘交换机配置策略。
- 接口配置文件规定配置哪些接入接口以及接口的配置策略。
- 全局策略启用 DHCP 的配置、QoS 和可在整个矩阵使用的可连接接入实体（AEP）配置文件功能。AEP 配置文件提供了在一大组边缘端口上配置管理程序的策略并将一个虚拟机管理（VMM）域和物理网络基础架构关联起来。在第 2 层和第 3 层外部网络连接中也需要它们。
- 池指定了 VLAN、VXLAN 和组播地址池。池是能够被 VMM 和第 4 层到第 7 层服务等多个域使用的共享资源。池代表一系列流量封装标识符（如 VLAN、VNID 和组播地址）。
- 物理和外部域策略包括下列内容：
  - 外部桥接域第 2 层域配置文件包含连接到矩阵的桥接第 2 层网络使用的端口和 VLAN 规范。
  - 外部路由域第 3 层域配置文件包含连接到矩阵的路由第 3 层网络使用的端口和 VLAN 配置。
  - 物理域策略包含物理基础架构规范，如端口和 VLAN，由租户或服务器组使用。
- 监控和故障排除策略规定监控对象、阈值、处理故障和日志的方法和执行诊断的方法。

访问策略配置

访问策略配置不连接核心交换机的面向外部的接口。面向外部的接口连接虚拟机控制器和管理程序、主机、路由器或矩阵扩展器（FEX）等外部设备。接口策略使得管理员可以配置端口通道和虚拟端口通道、LLDP、CDP 或 LACP 等协议和监控或诊断等特性。附录 C 中给出了针对交换机接口、端口通道、虚拟端口通道和改变端口速度的样本 XML 策略。访问策略示例。

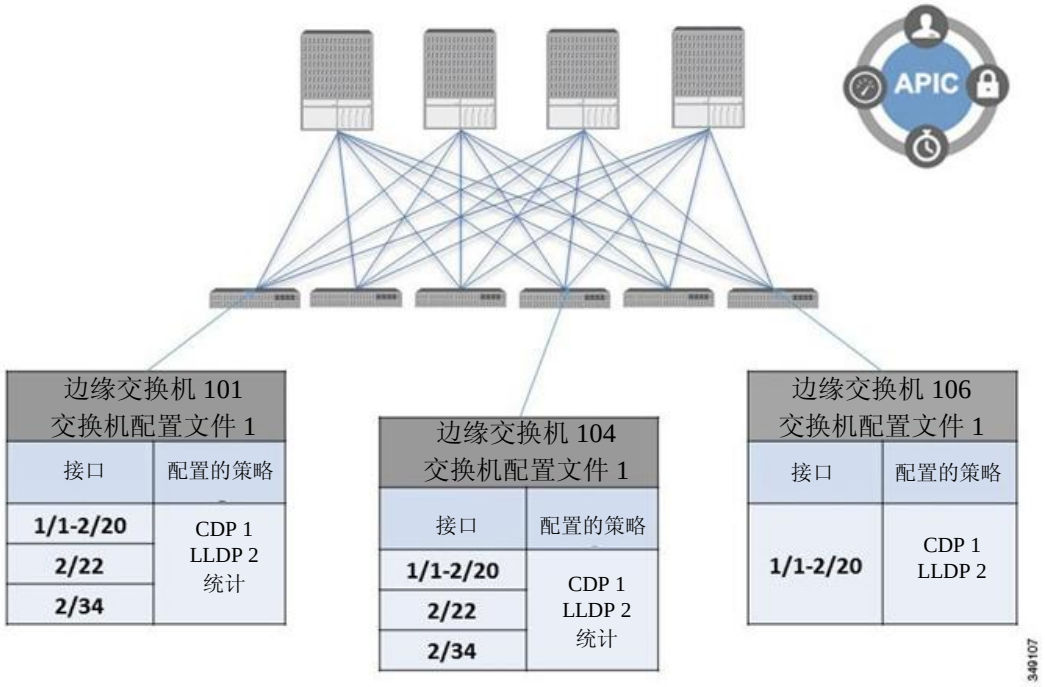
备注 租户网络策略与矩阵访问策略单独配置时，在租户策略以来的底层访问策略就绪之后，租户策略才被激活。

在可能非常多的交换机之间应用某种配置时，管理员确定在一个策略组中与接口配置关联的交换机配置文件。这样，矩阵上的大量接口可以一次性完成配置。交换机配置文件可能包含针对多个交换机或唯一特殊用途配置的对称配置。下图显示了配置通向 ACI 矩阵的接口的过程。

图 27：访问策略配置过程

下图说明了针对 ACI 矩阵应用交换机配置文件 1 和交换机配置文件 2 的结构。

图 28：应用接入交换机策略



基础架构和范围的这种组合使得管理员能够以一种可扩展的方式管理矩阵配置。可以使用 REST API、CLI 或 GUI 实施这些配置。GUI 中的 Quick Start Interface、PC、VPC Configuration 向导能够自动创建必要的基础对象用以实施此类策略。

调度员

日程表允许配置导入/导出或技术支持收集等操作在一个或多个指定时间窗口内发生。

一个日程表包含一组时间窗口（事件）。这些时间窗口可以是一次性的，或者可以在每周的指定时间和指定日期再次发生。在窗口中规定的选项，如时长或将要运行的任务的最大数量，决定某个已安排的任务的执行时间。例如，如果由于最大时长或任务的最大数量已经达到而致使某个更改无法在给定维护窗口内部署，那么该部署可以顺延到下一个时间窗口。

每个日程表定期查看 APIC 是否已经进入一个或多个维护窗口。如果已经进入，那么日程表执行符合维护策略中规定的限制的部署。

一个日程表包含一个或多个事件，这决定了与日程表关联的维护窗口。某个事件可以是下列中的一个：

- 一次性窗口—规定仅发生一次的某个日程表。窗口的最大时长或可以运行的任务的最大数量到达后，窗口关闭。

- 循环窗口—定义重复出现的日程表。任务的最大数量或窗口中指定的日期结束后，窗口关闭。

固件升级

APIC 上的策略管理固件升级流程的下列方面：

- 要使用的固件的版本。
- 从思科向 APIC 库下载固件映像。
- 兼容性强制执行。
- 被升级的对象：
  - 交换机
  - APIC
  - 兼容性目录
- 执行升级的时间。
- 如何处理故障（重试、中止、忽略等）。

每个固件映像中都包含是被支持类型和交换机型号的兼容性目录。APIC 管理固件映像、交换机型号和允许使用该固件映像的模型的目录。默认设置是在升级不符合兼容性目录时拒绝固件升级。

管理映像的 APIC 拥有一个针对兼容性目录、APIC 控制器固件映像和交换机映像的映像库。管理员可以通过创建映像源策略从外部 HTTP 服务器或 SCP 服务器向 APIC 映像库下载新固件映像。

APIC 上的固件组策略规定需要什么固件版本。

维护组策略规定升级固件的时间、升级的节点和如何处理故障。此外，维护组策略规定可以同时升级的节点组并向日程表分配这些维护组。节点组选项包括所有边缘节点、所有核心节点或属于矩阵一部分的节点集合。

APIC 控制器固件升级策略始终适用于集群中的所有节点，但升级总是每次完成一个节点。APIC GUI 提供关于固件升级的实时状态信息。

下图说明了 APIC 集群节点固件升级的过程。

图 29：APIC 集群控制器固件升级过程

APIC 以下列方式应用控制器固件升级策略：

- 控制器集群升级在周六午夜开始：
- 系统根据随新固件映像一同提供的兼容性目录检查现有固件的兼容性以升级到新版本。
- 升级时每次一个节点，直到集群中的所有节点都升级完毕。

备注 因为 APIC 是节点的复制集群，应尽量减少中断。在考虑规划 APIC 升级时管理员应当注意系统负载。

- ACI 矩阵，包括 APIC 在升级时继续运行。

备注 控制器以随机的顺序进行升级。每个 APIC 控制器升级使用大约 10 分钟。一旦控制器映像完成升级，它就会从集群中终止，集群中的其他 APIC 控制器仍在运行时该控制器映像会用更新的版本重启。控制器重启之后，它再次加入集群。然后集群融合，下一个控制器映像开始升级。如果集群没有立即融合且不完全合适，那么升级会等到集群融合并完全合适时再进行。在此过程中会显示一条“Waiting for Cluster Convergence”。

- 如果控制器节点升级失败，升级暂停，等待手动干预。

下图显示了升级所有 ACI 矩阵交换机节点固件时该过程的工作原理。

图 30：交换机固件升级过程

APIC 以下列方式应用交换机升级策略：

- APIC 在周六午夜开始升级。
- 系统根据随新固件映像一同提供的兼容性目录检查现有固件的兼容性以升级到新版本。
- 升级每次处理 5 个节点，直到所有指定的节点都升级完毕。

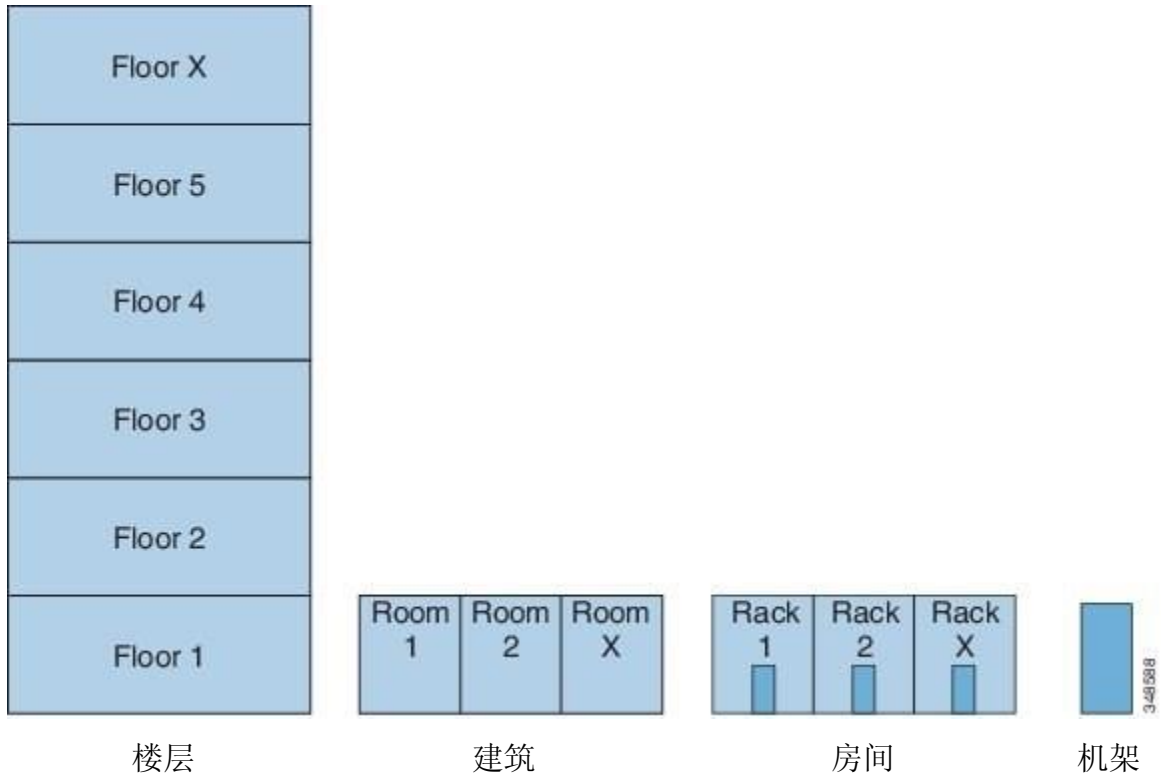
备注 固件升级导致交换机重启；重启可中断交换机运行数分钟。

- 如果交换机节点升级失败，升级暂停，等待手动干预。

地理位置

管理员使用地理位置策略确定 ACI 矩阵节点在数据中心的物理位置。下图举例说明了地址位置映射特性。

图 31：地理位置



例如，对于单个房间中的矩阵部署，管理员可以使用默认房间对象，然后创建一个或多个匹配交换机物理位置的机架。对于更大规模的部署，管理员可以创建一个或多个站点对象。每个站点可以包含一个或多个建筑物。每个建筑物有一层或多层。每个楼层有一个或多个房间，每个房间有一个或多个机架。最终每个机架可与一个或多个交换机关联。

第 6 章

网络和管理连接

本章包括以下部分：

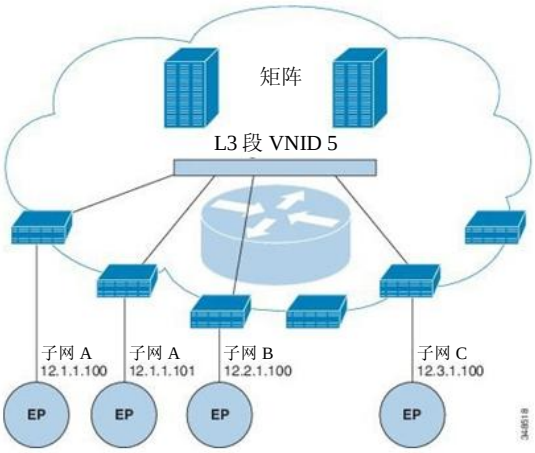
租户内部的路由选择

以应用为中心的基础架构（ACI）矩阵提供了租户默认网关功能和矩阵 VXLAN 网络之间的路由。对于每个租户，矩阵提供了跨越所有边缘交换机的虚拟默认网关，在连接端点的第一个边缘交换机的入口接口处租户与该交换机连接。每个入口接口都支持默认网关接口，矩阵中的所有入口界面共享同一个针对给定租户子网的路由器 IP 地址和 MAC 地址。

用于传输子网间租户流量的第 3 层 VNID

在 ACI 模型中，向 ACI 矩阵默认网关发送、到达矩阵入口的流量被路由到虚拟网络段（被称为“第 3 层 VNID”）。针对每个租户的三层地址域分配一个第 3 层 VNID。下图说明了在租户内部如何进行路由选择。

图 32：用于传输子网间租户流量的第 3 层 VNID



第 3 层 VNID 由 APIC 分配。使用第 3 层段的 VNID 传输穿过矩阵的流量。在出口边缘交换机，封包从第 3 层段 VNID 路由到出口子网的 VNID。

ACI 模型为在租户内路由的流量在矩阵内提供更加高效的转发。通过这种模型，两个虚拟机（VM）之间的流量属于同一个物理主机上的同一个租户，但位于不同的子网上。在（以最低的路径成本）被路由到正确的目标地址前，流量仅传输到入口交换机。在当前的虚拟机环境中，流量在被路由到正确的目标地址之前仅传输到边缘虚拟机（可能位于另一个物理服务器上）。

配置路由反射器

ACI 矩阵路由反射器使用多协议 BGP（MP-BGP）在矩阵内分布外部路由。在 ACI 矩阵内启用路由反射器时，矩阵管理员必须选择要充当路由反射器的核心交换机，并提供自主系统（AS）编号。一旦在 ACI 矩阵内启用路由反射器，管理员可以按照下列小节中的说明配置通往外部网络的连通性。

在把外部路由器与 ACI 矩阵连接时，矩阵基础架构管理员把核心节点配置为边界网关协议（BGP）路由反射器。出于冗余目的，被配置为路由反射器节点的核心设备不止一个（一个主反射器，一个辅助反射器）。

当租户需要把一个 WAN 路由器连接到 ACI 矩阵上时，基础架构管理员配置与作为 WAN 架顶（ToR）的 WAN 路由器连接的边缘节点（如下所述），并将此 WAN ToR 与一个路由反射器节点配对为 BGP 对等体。当在 WAN ToR 上配置路由反射器时，反射器能够在矩阵中公告这些租户路径。

每个边缘节点最多存储 4000 个路径。如果某个 WAN 路由器必须公告超过 4000 个路径，它必须与多个边缘节点配对。基础架构管理员使用它可以公告的路径（或路径前缀）配置每个配对的边缘节点。

基础架构管理员必须配置一个以下列方式连接到矩阵的外部 WAN 路由器：

1. 最多把两个核心节点配置为路由反射器。配置一个主路由反射器和一个辅助路由反射器，后者作为冗余。
2. 在 WAN ToR 上配置一个主路由反射器节点和一个辅助路由反射器节点，后者作为冗余。
3. 在 WAN ToR 上配置 ToR 负责公告的路径。这位可选操作，仅在知道租户路由器公告的路径超过 4000 时实施。

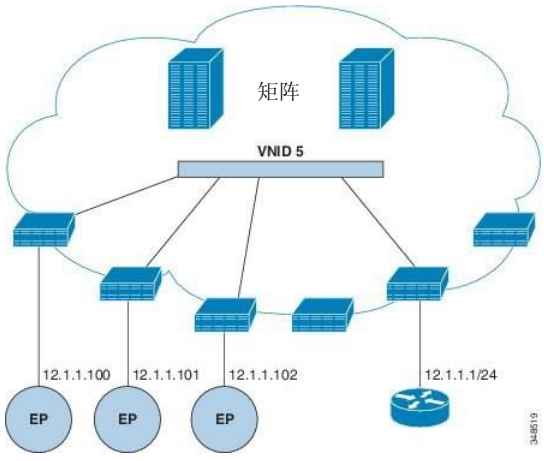
WAN 和其他外部网络

连接 WAN 的外部路由器和企业核心连接边缘交换机的前面板接口。连接外部路由器的边缘交换机接口可以被配置为桥接接口或路由对等物。

通往外部路由器的桥接接口

如下图所示，当边缘交换机接口被配置为桥接接口时，租户 VNID 的默认网关为外部路由器。

图 33：桥接外部路由器

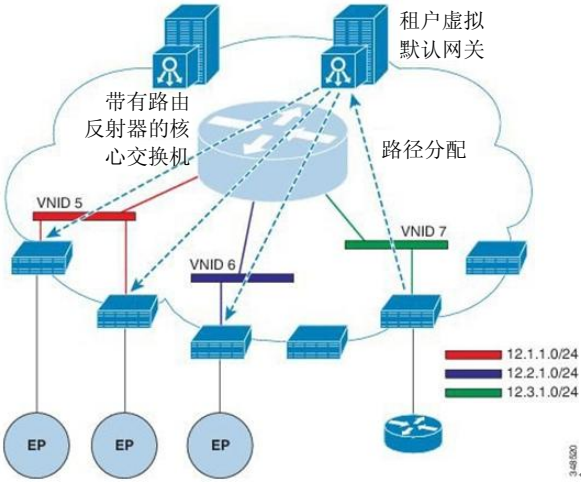


ACI 矩阵没有意识到外部路由器的存在，APIC 将边缘交换机静态地分配到它的 EPG。

路由器对等互连和路由分布式互连

如下图所示，当使用路由对等互连模型时，静态地把边缘交换机接口配置为外部路由器的对等端口。

图 34：路由器对等

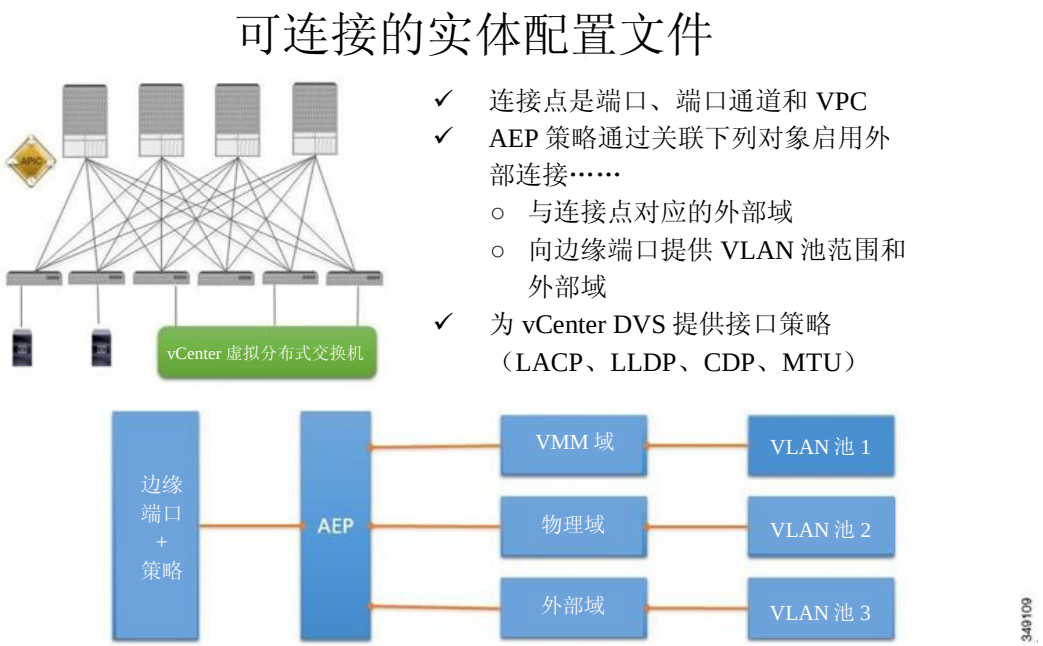


通过对等互连操作学习到的路径被发送到核心交换机。核心交换机充当路由反射器，向所有拥有属于相同租户的接口的边缘交换机分配外部路径。这些路径是“最长前缀匹配”（LPM）汇总的地址，通过外部路由器连接的远程边缘交换机的 VTEP IP 地址放置在边缘交换机的转发表中。WAN 路径没有转发代理。如果 WAN 路径不适合边缘交换机的转发表，那么流量会被丢弃。外部路由器不是默认网关，来自租户端点（EP）的封包被发送到 ACI 矩阵的默认网关。

连接实体配置文件

ACI 矩阵提供通过边缘端口与多种外部实体（如裸机服务器、管理程序、第 2 层交换机（如思科 UCS 矩阵互连）或第 3 层路由器（如思科 Nexus 7000 系列交换机））连接的多个连接点。这些连接点可以是物理端口、端口通道捆绑或边缘交换机上的虚拟端口通道捆绑（vPC），如下图所示。

图 35：可连接的实体配置文件



一个可连接实体配置文件（AEP）代表一组带有类似基础架构策略要求的外部实体。这种基础架构策略包括物理接口策略，如思科发现协议（CDP）、链路层发现策略（LLDP）、最大传输单元（MTU）或链路聚合控制协议（LACP）。

AEP 需要在边缘交换机上部署 VLAN。封装池（和关联的 VLAN）可以在边缘交换机上重复使用。AEP 向物理基础架构暗中提供 VLAN 池的范围。

备注 必须在各种配置情景下考虑下了 AEP 要求和依赖关系：

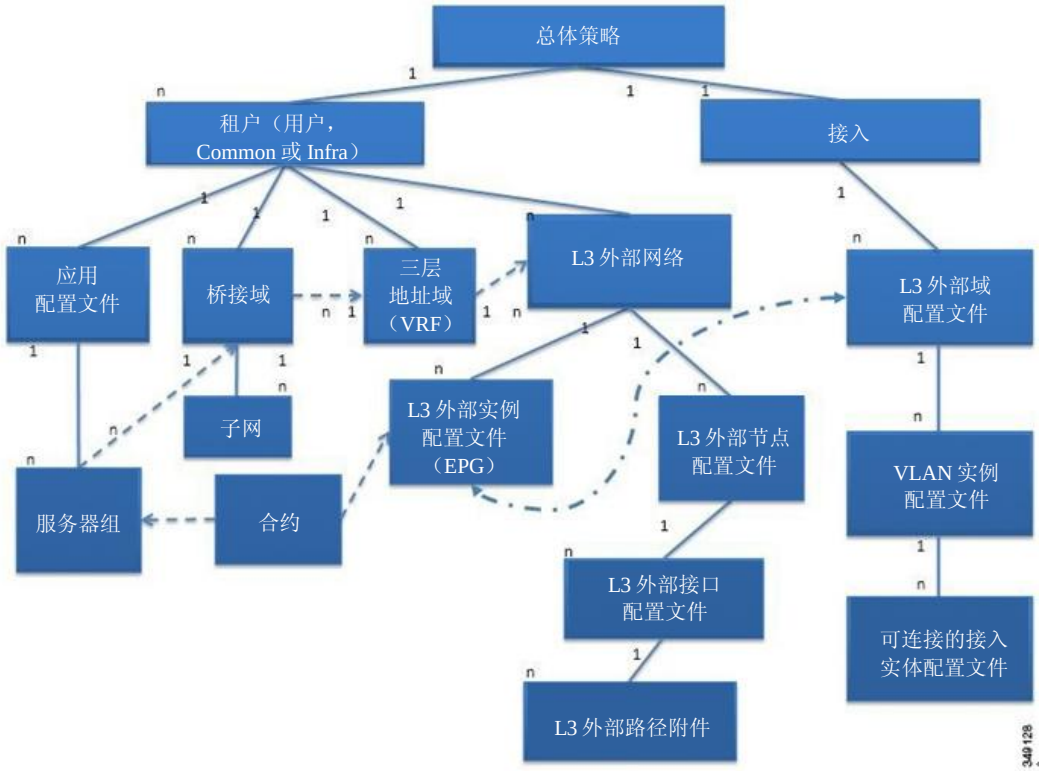
- AEP 在边缘交换机上提供 VLAN 池（和关联的 VLAN）时，服务器组（EPG）在端口上启用 VLAN。在端口上部署 EPG 之前，不会有流量流过。
- 部署 AEP VLAN 池之前，即使提供了 EPG，VLAN 也不会在边缘端口上启用。
- 在符合下列条件的边缘端口上提供或启用一个特定的 VLAN：该边缘端口基于静态绑定在某个边缘端口上的 EPG 事件，或基于来自 VMware vCenter 等外部控制器的 VM 事件。
- 边缘交换机不支持重叠的 VLAN 池。不同的重叠 VLAN 池不得与同一个 AEP 关联。

通往外部网络的桥接和路由连接

被外部网络管理的对象启用第 2 层和第 3 层通往外部网络的租户连接可以使用 GUI、CLI 或 REST API 配置通往外部网络的租户连接。附录 D：租户第 3 层外部网络策略示例包含一个样本 XML 策略。在矩阵中轻松的定位所有此类外部网络接入点时，第 2 层和第 3 层外部边缘节点可以被标记为“边界边缘节点”。

连接外部网络的租户路由连接性通过将某个矩阵接入（infraInfra）外部路由域（l3extDomP）与一个第 3 层外部外网（l3extInstP）的租户第 3 层外部实例配置文件（l3extInstP）EPG 关联启用，如下图所示。

图 36：通往外部网络的租户路由连接



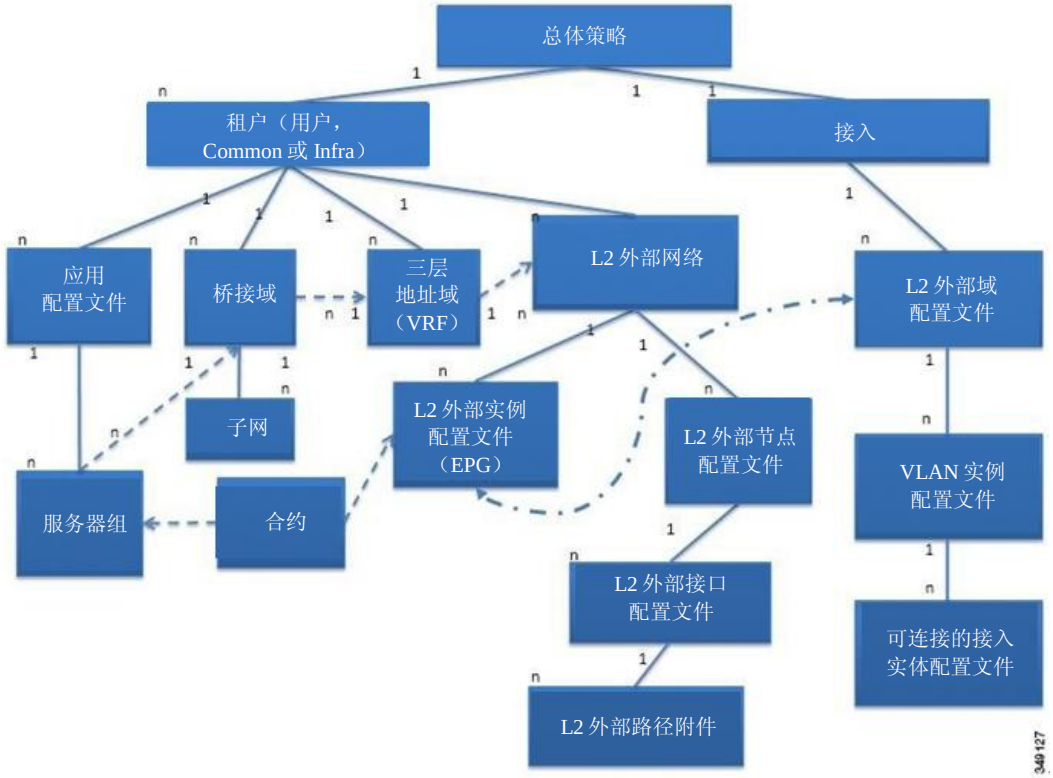
l3extOut 包括路由协议选项（BGP、OSPF 或两者都有）和特定于交换机的配置和特定于接口的配置。

备注 第 3 层外部网络包含路由协议（如 OSPF 连同其相关的三层地址域和区域 ID）、第 3 层外部接口配置文件包含必要的 OSPF 接口配置详细信息。启用 OSPF 时两者都需要。

l3extInstP EPG 通过合约向租户 EPG 披露外部网络。每个边缘交换机（节点）仅能配置一个外部网络。但是，外部网络配置可以通过把多个节点与 L3 外部节点配置文件关联被轻松地重复用于多个节点。可针对失败恢复和负载均衡配置使用相同配置文件的多个节点。

使用类似的流程配置通往外部网络的租户桥接连接性。可通过将矩阵接入（infraInfra）外部桥接域（L2extDomP）与第 2 层外部网络（l2extOut）的第 2 层外部实例配置文件（l2extInstP）EPG 关联起来启用租户第 2 层桥接外部网络连接性，如下图所示。

图 37：通往外部网络的租户桥接连接



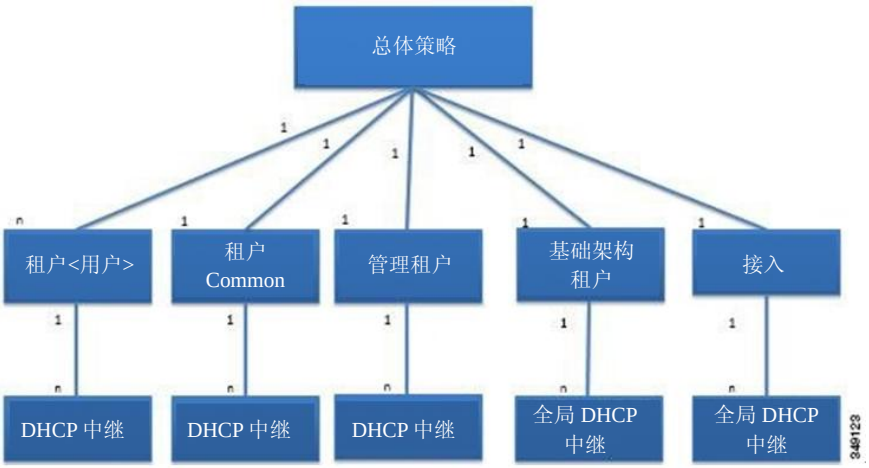
l2extOut 包含特定于交换机的配置和特定于接口的配置。l2extInstP EPG 通过合约向租户 EPG 披露外部网络。例如：某个包含一组网络连接存储设备的租户 EPG 可以通过合约按照第 2 层外部网络中包含的网络配置与 l2extInstP EPG 通讯。每个边缘交换机仅能配置一个外部网络。但是，外部网络配置可以通过把多个节点与 L2 外部节点配置文件关联被轻松地重复用于多个节点。可针对失败恢复和负载均衡配置使用相同配置文件的多个节点。

DHCP 中继

ACI 矩阵范围内的洪泛被默认禁用时，桥接域内的洪泛被默认启用。桥接域内的洪泛被默认启用，因此客户端可以连接相同 EPG 内的 DHCP 服务器。但是，当 DHCP 服务器与客户端位于不同的 EPG 或三层地址域中时，需要 DHCP 中继。而第 2 层洪泛被禁用时，需要 DHCP 中继。下

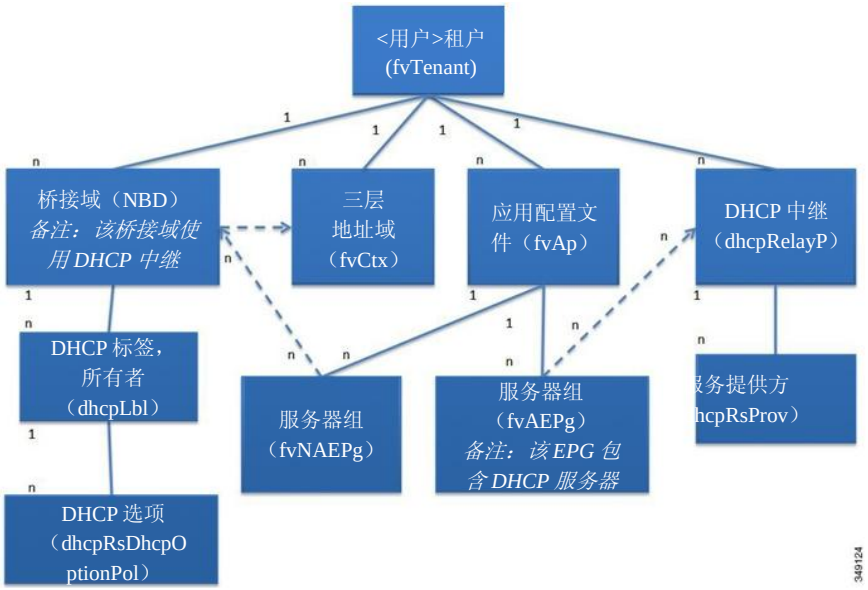
图显示了可以包含 DHCP 中继的管理信息树（MIT）中的被管对象：用户租户、普通租户、基础架构租户、管理租户和矩阵接入。

图 38：MIT 中的 DHCP 中继位置



下图显示了用户租户内 DHCP 中继对象的逻辑关系。

图 39：租户 DHCP 中继



DHCP 中继配置文件包含一个或多个服务提供方。EPG 包含一个或多个 DHCP 服务器，EPG 和 DHCP 中继之间的关系指定了 DHCP 服务器 IP 地址。服务使用方桥接域包含将服务提供方 DHCP 服务器与桥接域关联起来的 DHCP 标签。标签匹配使得桥接域可以使用 DHCP 中继。

备注 桥接域 DHCP 标签必须匹配 DHCP 中继的名称。

DHCP 标签对象也指定所有者。所有者可以是租户，也可以是接入基础架构。如果所有者是租户，ACI 矩阵首先会在租户内寻找匹配的 DHCP 中继。如果用户租户内没有匹配，ACI 矩阵会在普通租户内寻找。

DHCP 中继以下列两种模式之一运行：

- 可见—服务提供方的 IP 和子网被泄露到服务使用方的三层地址域中。DHCP 可见时，它是服务使用方三层地址域所特有的。
- 不可见—服务提供方的 IP 和子网没有被泄露到服务使用方的三层地址域中。

备注 当 DHCP 中继以不可见的模式运行时，服务提供方的桥接域必须与服务使用方在同一个边缘交换机上。

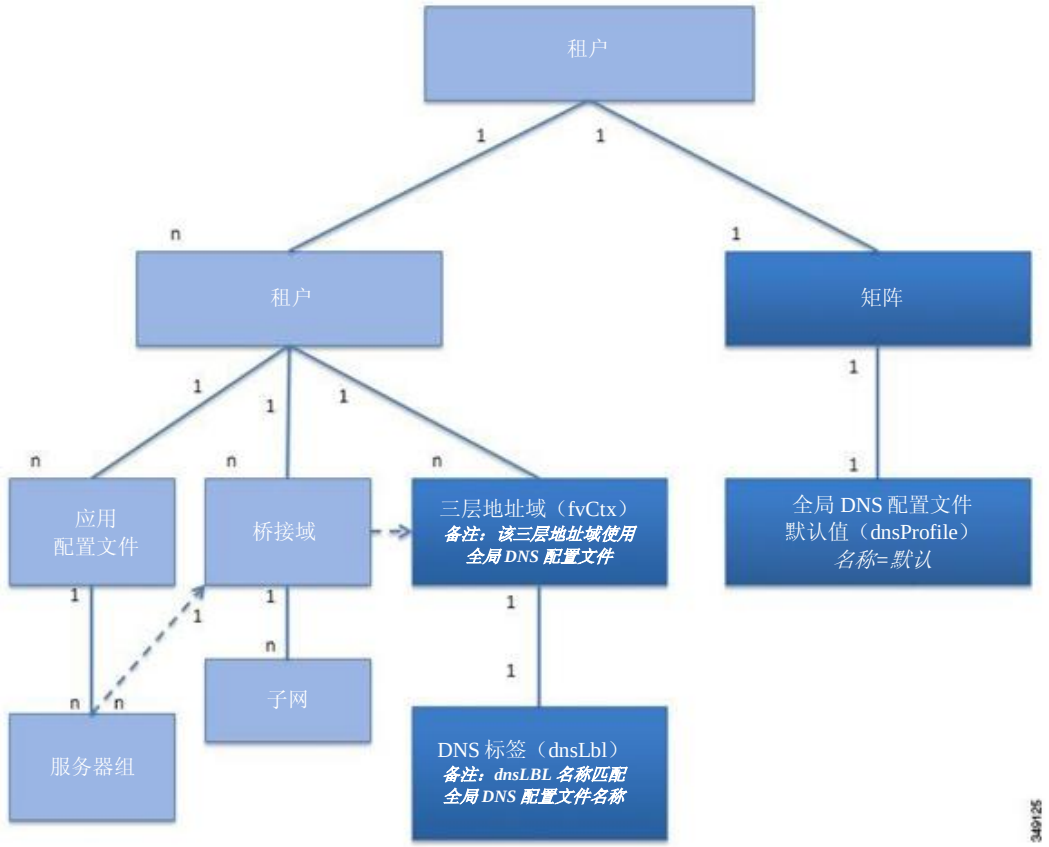
在以类似的方式配置租户和接入 DHCP 中继时，下列使用案例会发生相应的变化：

- 任何租户都可以使用普通租户 DHCP 中继。
- ACI 矩阵服务提供方选择性地向其他租户曝光基础架构租户 DHCP 中继。
- 矩阵接入（infraInfra）DHCP 中继可被任何租户使用且允许 DHCP 服务器拥有更加粒状的配置。在这种情况下，可以在相同的桥接域内为节点配置文件中的每个边缘交换机配置单独的 DHCP 服务器。

DNS

矩阵管理的对象中包含 ACI 矩阵 DNS 服务。可以通过矩阵接入矩阵全局默认 DNS 配置文件。下图显示了矩阵内 DNS 管理对象的逻辑关系。请参考附录 F：样本 DNS XMP 策略的 DNS。

图 40：DNS



三层地址域必须包含一个 dnsLBL 对象以便于使用全局默认 DNS 服务。标签匹配使得租户三层地址域可以使用全局 DNS 服务提供方。全局 DNS 配置文件为“默认”，因此三层地址域标签名称为“默认”（dnsLBL 名称 = 默认）。

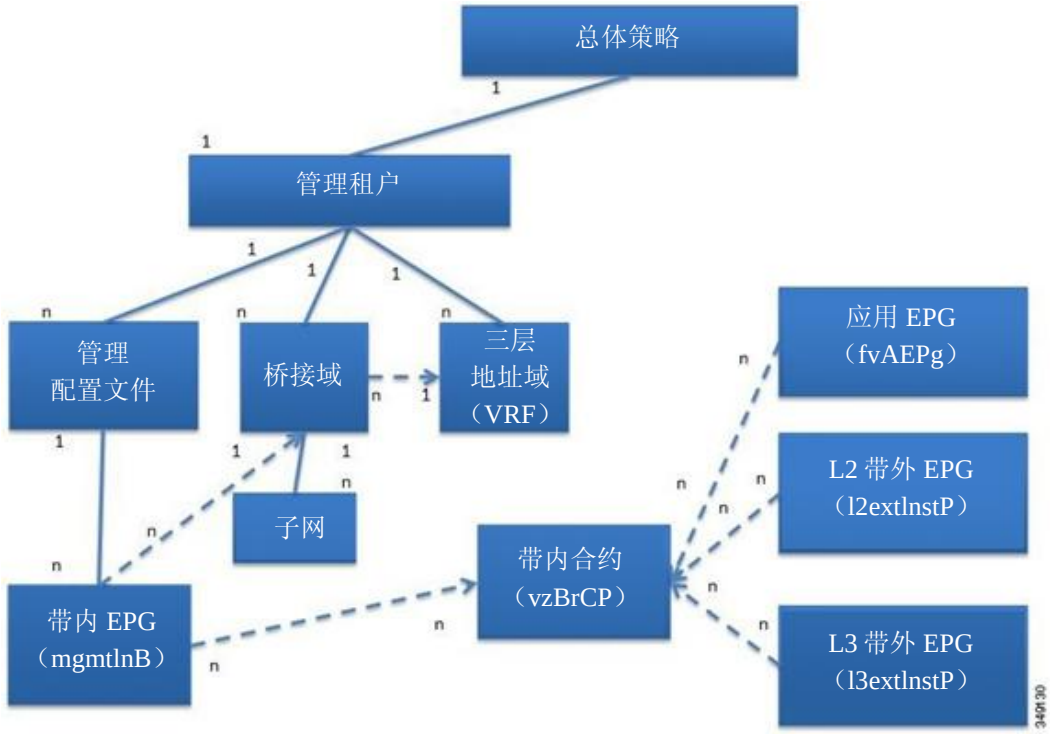
带内和带外管理访问

管理租户提供了一种方便的方式用于配置矩阵管理功能的接入。矩阵管理功能可以通过 APIC 使用，这些功能也可以直接通过带内和带外网络策略直接访问。

带内管理访问

下图概况描述了管理租户带内矩阵管理访问策略。

图 41：带内管理访问策略



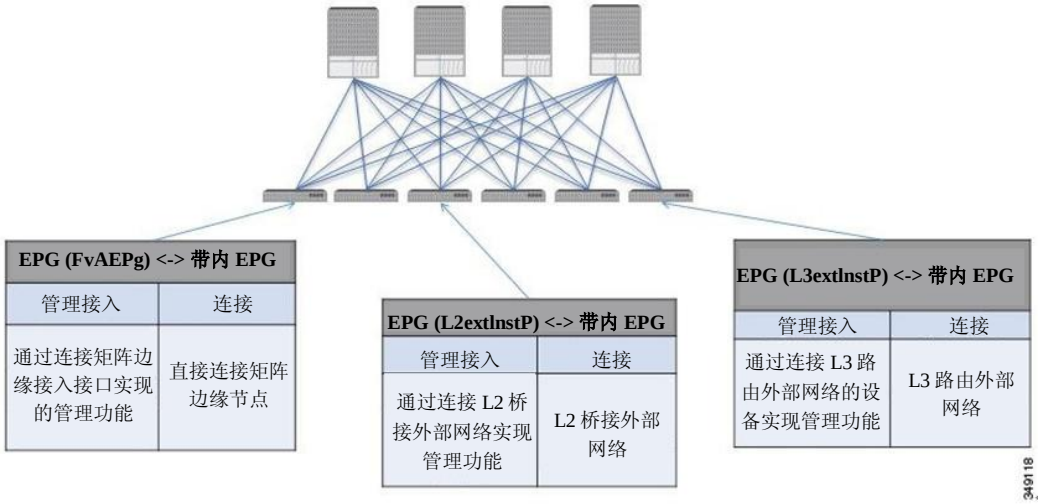
管理配置文件包含通过带内合约提供管理功能接入的带内 EPG MO（vzBrCP）。vzBrCP 启用 fvAEPg、l2extInstP 和 l3extInstP EPGs 以便使用带内 EPG。这把矩阵管理曝光于当地连接的设备、在第 2 层桥接外部网络上连接的设备以及第 3 层路由外部网络。如果服务使用方和服务提供方 EPG 位于不同的租户内，那么它们可以使用普通租户的桥接域和三层地址域。

认证、接入和审计日志适用于这些连接；任何试图通过带内 EPG 使用管理功能的用户都必须拥有相应的访问权限。

带内管理访问

下图显示了一种带内管理接入情形。

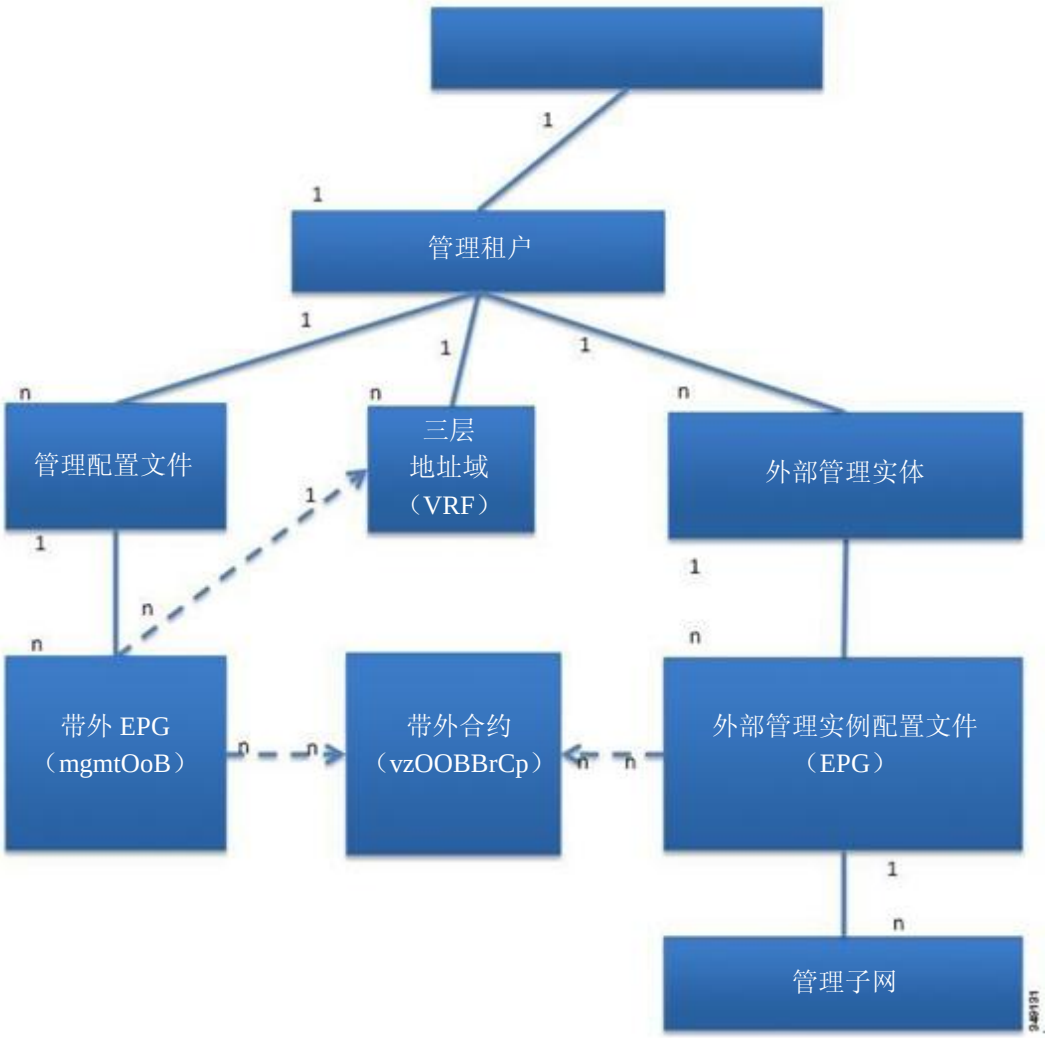
图 42：带内管理访问情形



带外管理访问

下图概况描述了管理租户带外矩阵管理访问策略。

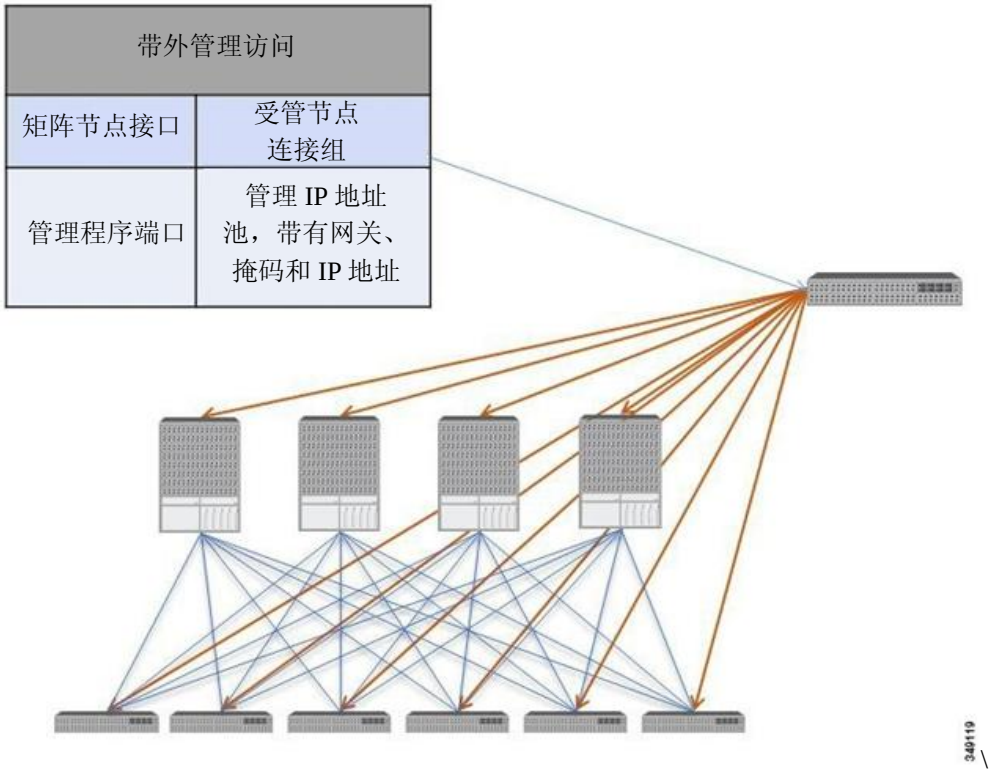
图 43：带外管理访问策略



管理配置文件包含通过带外合约提供管理功能接入的带外 EPG MO（vzOOBBrCp）。vzOOBBrCp 使得外部管理实例配置文件（mgmtExtInstP）EPG 可以使用带外 EPG。这把矩阵节点管理程序端口按照服务提供方的偏好曝光于本地或远程连接的设备。管理程序端口的带宽将会低于带内端口，当通过带内端口无法接入时管理程序端口支持直接接入矩阵节点。认证、接入和审计日志适用于这些连接；任何试图通过带外 EPG 使用管理功能的用户都必须拥有相应的访问权限。

下图显示了如何通过专用交换机整合带外管理接入。

图 44：带外接入场景



备注 某些服务提供方选择限制本地连接的带外连接，而其他服务提供方则选择启用来自外部网络的路由或桥接连接。服务提供方可以选择配置一组包含带内管理接入和带外管理接入的策略，既可以仅适用于本地设备，也可以既适用于本地设备，也适用于远程设备。

共享服务合约用法

配置共享服务合约时遵守下列指南。

- 带内服务器组和带外服务器组之间的合约（EPG）—当合约在带内和带外 EPG 之间配置时，适用于下列限制：
  - 两个 EPG 都应当位于同一个三层地址域中（VRF）。
  - 过滤器仅适用于来向。
  - 不支持第 2 层过滤器。

- QoS 不适用于带内第 4 层到第 7 层服务。
- 管理统计不可用。
- 不支持计算密集型流量的共享服务。

- 私有网络没有强制执行时，桥接域间的流量需要合约。
- 不支持基于前缀的 EPG。
  不支持对应第 3 层外部网络的共享服务。需要共享相同第 3 层三层地址域的 EPG 使用或提供外部第 3 层外部网络提供或使用的合约。
- 仅在子网不重叠、不重复的情况下支持共享服务。遵守下列指南：
  - 在 EPG 而非桥接域下为共享服务提供方配置子网。
  - 在共享相同三层地址域的 EPG 下配置的子网必须分离且不得重叠。
  - 从一个三层地址域泄露到另一个三层地址域的子网必须分离且不得重叠。
  - 从多个服务使用方网络泄露到一个三层地址域中或从一个三层地址域中泄露到多个服务使用方网络的子网必须分离且不得重复。

  如果两个服务使用方错误地与相同的子网配置，要解决这种状况可以删除两者的子网配置，然后重新正确地配置子网。
- 不要使用 AnyToProv 在服务提供方三层地址域中配置共享服务。APIC 会用错误拒绝该操作。
- 在提供共享服务时服务提供方的私有网络无法采用非强制执行模式。

第 7 章

用户访问、验证的记录

本章包括以下部分：

用户访问、验证和记录

APIC 策略管理思科 ACI 矩阵的访问、认证和记录（简称 AAA）功能。将用户权限、角色和域与访问权限继承集成之后，管理员可以在被管对象层级以一种非常颗粒化的方式配置 AAA 功能。可以使用 REST API、CLI 或 GUI 实施这些配置。

多租户支持

核心 APIC 内部数据访问控制系统可提供多租户隔离，并防止信息隐私在租户间泄露。读/写限制防止租户看到其他租户的配置、统计信息、错误或事件数据。除非管理员分配进行上述操作的权限，否则限制租户读取矩阵配置、策略、统计信息、错误或事件。

用户访问：角色、权限和安全域

APIC 按照用户角色通过基于角色的访问控制（RBAC）提供访问权。ACI 矩阵用户与下列项目关联：

- 角色集合
- 对于每种角色，一种权限类型：无访问权、只读或读写
- 识别用户可以访问的管理信息树（MIT）的各部分的一个或多个安全域标签

ACI 矩阵管理被管对象（MO）层级的访问权限。权限是允许或限制访问系统内特定功能的被管对象。例如，矩阵设备是一个权限位。这个位由 APIC 在物理矩阵中对应设备的所有对象上设置。

角色是权限位的集合。例如，因为使用权限位为“矩阵设备”和“租户安全”配置了“admin”角色，所以“admin” 角色可以访问矩阵设备和租户安全对应的所有对象。

安全域是一个与 ACI MI 对象层级中特定子树关联的一个标签。例如，默认租户“common” 带有一个域标签“共同” 类似地，一个特殊域标签“all” 包含整个 MIT 对象树。管理员用户可以向 MIT 对象层级分配自定义域。例如，向 solar 租户分配一个“solar”域标签。在 MIT 内，只有特定的对象可以被标记为安全域。例如，租户可以被标记为安全域，但租户内的对象不能。

如果虚拟机管理（VMM）域可以被标记为安全域，那么该安全域内包含的用户可以访问带有对应标签的 VMM 域。例如，如果一个名为“solar”的租户用名为“sun”的安全域进行标记且一个 VMM 域也用名为“sun”的安全域进行标记，那么 solar 租户内的用户可以根据自己的访问权限访问 VMM 域。

自定义 RBAC 规则

RBAC 规则可以让矩阵范围内的管理员在整个安全域提供访问权，否则这些域就会受到限制。使用 RBAC 规则曝光物理资源或共享原本由于位于不同安全域而无法使用的服务。RBAC 规则仅向目标资源提供读取权限。GUI RBAC 规则页的路径为 Admin => AAA => Security Management。可以在某个资源存在之前创建 RBAC 规则。关于对 RBAC 规则角色和权限（及其依赖性）的说明，请参考《管理信息模型》。

备注 用修改“all”域向用户提供使用用户安全域之外的资源的权限，是一种不良习惯。该用户可以访问为其他用户提供的资源。

跨安全域选择性地曝光物理资源

矩阵管理员使用 RBAC 规则有选择性地向用户曝光原本因为位于不同的安全域中而无法使用的物理资源。

例如，如果位于租户“solar”内的用户需要访问虚拟机管理（VMM）域，矩阵管理员可以创建一个 RBAC 规则用于上述目的。RBAC 规则包括两部分：定位要被访问的对象的可识别名（DN），包含要访问对象的用户的安全域的名称。在本示例中，当安全域 solar 中的指定用户被登录时，该规则向其提供访问 VMM 域及其在树中的子对象的权限。在向用户提供访问 VMM 域的多个安全域时，矩阵管理员将为包含 VMM 域和安全域 DN 的每个安全域创建一个 RBAC 规则。

备注 当 RBAC 规则向位于管理信息树不同部分的用户曝光某个对象时，无法通过横贯树结构使用 CLI 导航到此对象。但是，只要用户知道包含在 RBAC 规则中的对象的 DN，用户就能使用 CLI 通过 MO 寻找命令将其定位。

相关主题

RBAC 规则样本，第 167 页

允许跨安全域共享服务

矩阵管理员使用 RBAC 规则配置允许在租户间共享服务的跨租户 EPG 通讯。

相关主题

跨租户 EPG 通讯，第 20 页
RBAC 规则样本，第 167 页

APIC 本地用户

管理员可以选择不使用外部 AAA 服务器，而是在 APIC 上配置用户。这些用户被称作 APIC 本地用户。APIC 也允许管理员向在外部管理的认证“轻型目录访问协议”（LDAP）、RADIUS 或 TACACS+服务器上配置的用户提供访问权。用户可以术语不同的认证系统，可同时登录到 APIC。

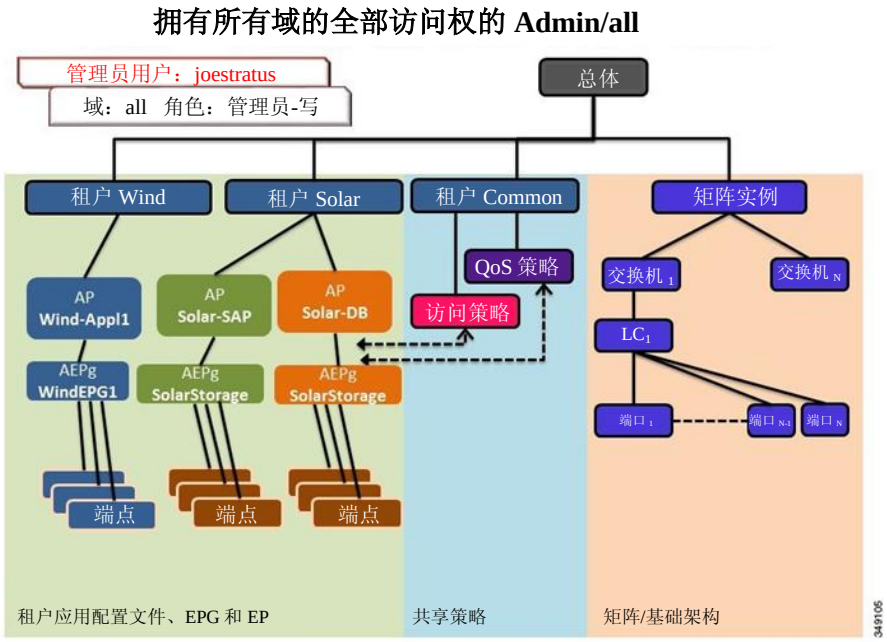
下图显示了在本地 APIC 认证数据库中配置拥有整个 ACI 矩阵全部访问权的管理员用户时该流程的工作方式。

图 45：APIC 本地用户配置流程

备注 安全域“all”表示整个被管信息树（MIT）。该域包含系统中的所有策略和 APIC 管理的所有节点。租户域包含素有用户和租户的被管对象。不应当向租户管理员授予访问“all”域的权限。

下图显示了管理员用户 Joe Stratus 对系统的访问权。

图 46：为“all”域配置管理员用户的结果

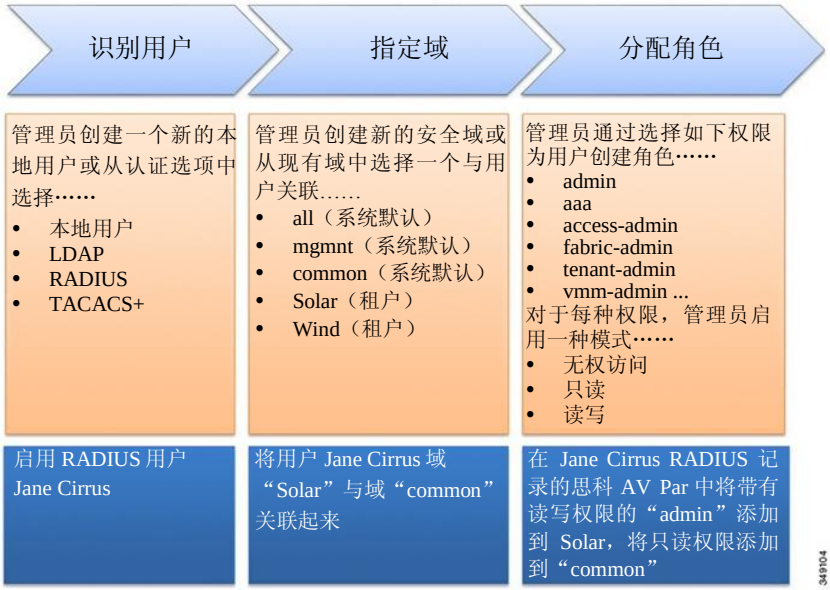


拥有读写“admin”权限的用户 Joe Stratus 被分配到给予它访问整个系统的全部权限的域“all”。

从外部管理的验证服务器用户

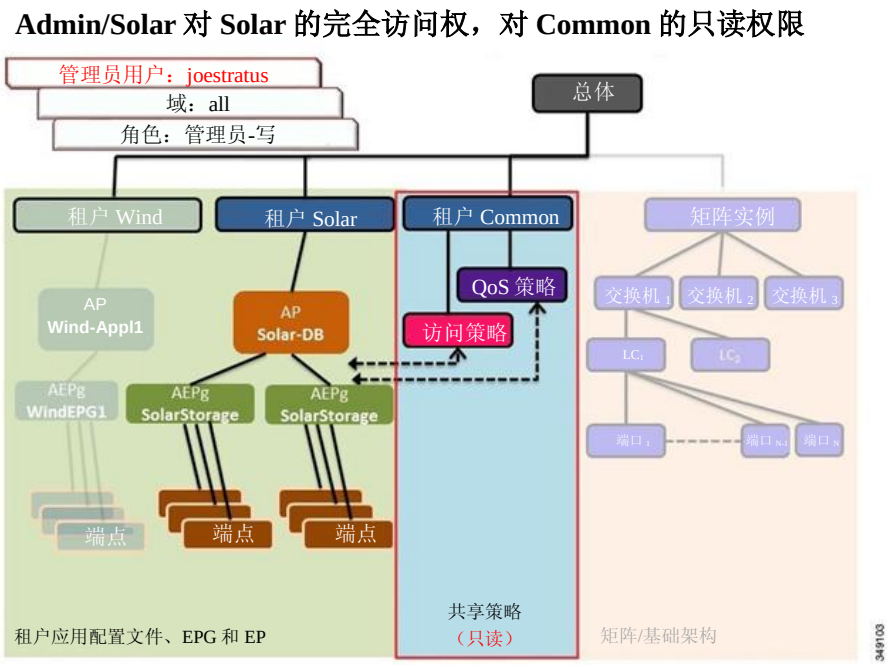
下图显示了在外部 RADIUS 服务器中配置对租户 Solar 拥有全部访问权的管理员用户时该过程的工作方式。

图 47：在外部认证服务器上配置用户的过程



下图显示了 admin 用户 Jane Cirrus 对系统的访问权。

图 48：为“all”域配置 Admin 用户的结果



在本例中，Solar 租户管理员对包含在 Solar 租户内的所有对象拥有完全访问权，对租户 Common 拥有只读权。租户 admin Jane Cirrus 对租户 Solar 拥有完全访问权，包括在租户 Solar 中创建新用户的能力。租户用户能够修改其拥有和控制的 ACI 矩阵的配置参数。他们还可以读取统计信息，对适用于他们的实体（如端点、服务器组和应用配置文件）的故障和事件进行监控。

上例中，在一个外部 RADIUS 认证服务器中对用户 Jane Cirrus 进行了配置。在某个外部认证服务器上配置一个 AV Pair 时，向现有用户记录添加一个思科 AV Pair。思科 AV Pair 为 APIC 上的用户指定基于角色的访问控制（RBAC）角色和权限。然后 RADIUS 服务器向 APIC 控制器通告用户权限。

在上例中，某个开放半径服务器（/etc/raddb/users）的配置如下：

janecirrus Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = solar/admin/,common//read-all(16001)" 本例包含下列元素：

- janecirrus 是租户管理员
- solar 是租户
- admin 是拥有写入权限的角色
- common 是所有用户都应当拥有只读访问权的租户 common 子树
- read-all 是带有读取权限的角色

思科 AV Pair 格式

思科 APIC 要求管理员在外部认证服务器上配置思科 AV Pair。为此，管理员向现有用户记录添加一个思科 AV Pair。思科 AV Pair 指定了 APIC 需要的 RBAC 规则和用户的权限。思科 AV Pair 规则对 RADIUS、LDAP 或 TACACS+而言相同。

思科 AV Pair 格式如下：

shell:domains = domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2, domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2 shell:domains = domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2, domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(16003)

第一个 av-pair 没有 UNIX 用户 ID，而第二个有。两个都正确。

APIC 支持下列正则表达式：

shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?),\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$ shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?),\\S+?/\\S*?/\\S*?){0,31})$

RADIUS

在 RADIUS 服务器上配置用户时，APIC 管理员必须使用 cisco-av-pair 属性配置所需的属性（shell:domains）。默认的用户角色是网络操作员。

SNMPv3 认证协议选项是 SHA 和 MD5。私有协议选项是 AES-128 和 DES。如果没有在 cisco-av-pair 属性中指定这些选项，那么 MD5 和 DES 为默认认证协议。

例如，SNMPv3 认证和私有协议属性可以被指定为：

snmpv3:auth=SHA priv=AES-128

类似地，域列表如下：

shell:domains="domainA domainB …"

TACACS+ 验证

终端访问控制器访问控制设备升级版（TACACS+）是思科设备支持的另一个远程 AAA 协议。TACACS+相比 RADIUS 认证具有下列优势：

- 提供独立的 AAA 设施例如，APIC 可在没有验证的情况下批准访问权。
- 使用 TCP 在 AAA 客户端和服务器之间发送数据，通过以连接为导向的协议实现可靠的传输。
- 加密交换机和 AAA 服务器之间的整个协议有效负载，确保更好地保密数据。RADIUS 仅对密码加密。
- 使用句法和配置上与 RADIUS 不同但 APIC 支持的 av-pairsshell:domains。

下面的 XML 示例对 ACI 矩阵进行了配置，10.193.208.9.

```
<aaaTacacsPlusProvider name="10.193.208.9"
                       key="test123"
                       authProtocol=”pap”/>
```

LDAP/主动式目录验证

与 RADIUS 和 TACACS+类似，LDAP 允许网络元素检索可用于验证并批准用户实施某些动作的 AAA 证书。管理员可以添加认证授权配置以启用 LDAPS（LDAP over SSL）信任并防止中间人攻击。

下面的 XML 示例对 ACI 矩阵进行了配置，以便与 IP 地址为 10.30.12.128 的 LDAP 服务提供方协同工作。

```
<aaaLdapProvider name="10.30.12.128"
                 rootdn="CN=Manager,DC=ifc,DC=com"
                 basedn="DC=ifc,DC=com"
                 SSLValidationLevel="strict"
                 attribute="CiscoAVPair"
                 enableSSL="yes"
                 filter="cn=$userid"
                 port="636" />
```

APIC Bash 外壳中的用户 ID

APIC 上 Linux shell 的用户 ID 在 APIC 中针对本地用户生成。对于在外部服务器上管理验证证书的用户，Linux 外壳的用户 ID 可以在 cisco-av-pair 中指定。在上面的 cisco-av-pair 中忽略（16001）是合法的，在这种情况下，远程用户取得默认 Linux 用户 ID 23999。Bash 会话过程中使用 Linux 用户 ID，因此可以强制执行标准 Linux 权限。用户创建的所有被管对象都被标记为被该用户的 Linux 用户 ID 创建。

下面的示例是一个用户 ID，与 APIC Bash 外壳中见到的相同：

```
admin@ifav17-ifc1:~> touch myfile
admin@ifav17-ifc1:~> ls -l myfile
-rw-rw-r-- 1 admin admin 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> ls -ln myfile
-rw-rw-r-- 1 15374 15374 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> id
uid=15374(admin) gid=15374(admin) groups=15374(admin)
```

登陆域

登陆域定义了某位用户的验证域。登陆域可被设置到 Local、LDAP、RADIUS 或 TACACS+验证机制。从 REST、CLI 或 GUI 访问系统时，APIC 允许用户从正确的验证域中作出选择。

例如，在 REST 场景下，用户名的前缀是一个字符串，这样完整的用户名如下：

```
apic:<domain>\<username>
```

如果从 GUI 访问系统，APIC 提供一个域的下拉列表供用户选择。如果没有指定 apic: domain ，默认验证域服务器用于查找用户名。

登陆域

第 8 章

虚拟机管理器域

本章包括以下部分：

虚拟机管理器域

APIC 是单一虚拟管理平台，可使包含访问策略和第 4-7 层服务的所有虚拟和物理负载的整个网络自动化。对于 VMware vCenter，虚拟分布式交换机（VDS）和端口组的所有网络功能都使用 APIC 执行。vCenter 管理员需要在 vCenter 中执行的唯一功能是将 vNIC 放置在 APIC 创建的相应组内。

VM 控制器—代表外部虚拟机管理系统，如 VMware vCenter、VMware vShield 和 Microsoft System Center Virtual Machine Manager（SCVMM）。

虚拟机管理器（VMM）域—带有类似网络策略要求的组 VM 控制器。例如，VM 控制器可以共享 VLAN 或虚拟可扩展局域网（简称 VXLAN）空间和应用服务器组（简称 EPG）。APIC 与控制器通信，以发布网络配置，例如接口组，随后应用到虚拟工作负载上。

备注 单个 VMM 域可以包含 VM 控制器的多个实例，但它们必须来自同一个厂家（如来自 VMware 或 Microsoft）。

在 VMM 域中配置 EPG—将应用配置文件 EPG 关联到 VMM 域，如下所示：

虚拟机管理器域

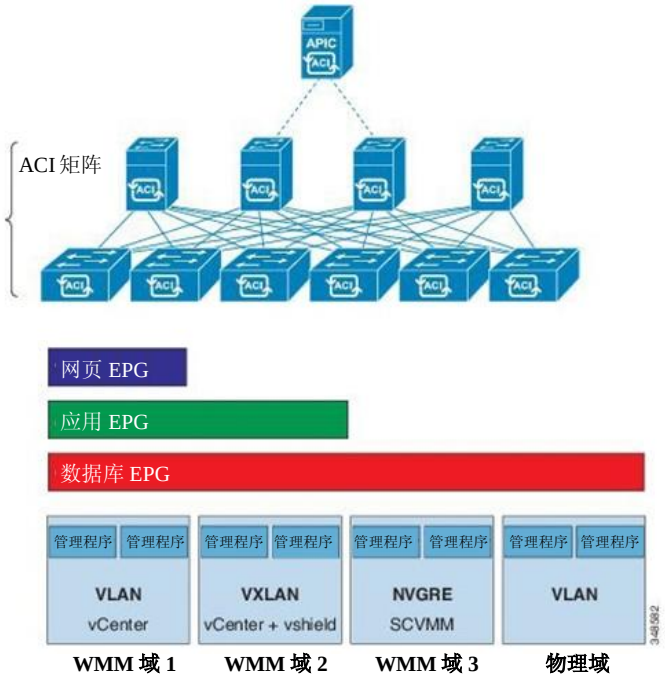
- APIC 把 EPG 作为 VM 控制器中的端口组进行推送。然后计算管理员把 vNIC 放置到这些端口组中。
- EPG 可以跨越多个 VMM 域，而一个 VMM 域可能包含多个 EPG。

EPG 在矩阵内的扩展能力—EPG 能够使用多个 VMM 域进行下列操作：

- 通过使用 APIC 自动管理的封装标识符识别 VMM 域内的 EPG，例如 VLAN、虚拟网络 ID（VNID 或 VXLAN）或虚拟子网标识符（NVGRE 的 VSID）。
- 一个 EPG 可以被映射到多个物理（对于裸机服务器）或虚拟域。它可以在每个域中使用不同的 VLAN、VNID、VSID ID 封装。
- 入口边缘交换机正常化并将封装（VLAN/VNID/VSID）从封包翻译为矩阵本地 VXLAN VNID（段 ID），这完成了 EPG 封装对边缘交换机的本地化。
- 可以在不同的边缘交换机上重复使用封装 ID。例如，基于 VLAN 的封装将 VMM 域内的 EPG 数量限制在 4096 以内。可以通过创建多个 VMM 域扩展 EPG 并在多个 VMM 域中关联相同的 EPG。

备注 多个 VMM 域如果没有重叠的 VLAN 池，那么它们可以连接相同的边缘交换机。如下图所示。类似地，可以在不同的域中使用相同的 VLAN 池，前提是它们不使用相同的边缘交换机。

图 49：多个 VMM 域和在矩阵内扩展 EPG



连接实体配置文件

ACI 矩阵提供通过边缘端口与多种外部实体（如裸机服务器、管理程序、第 2 层交换机（如思科 UCS 矩阵互连）和第 3 层路由器（如思科 Nexus 7000 系列交换机））连接的多个连接点。这些连接点可以是物理端口、端口通道或边缘交换机上的虚拟端口通道（vPC）。

一个**可连接实体配置文件**（AEP）代表一组带有类似基础架构策略要求的外部实体。这种基础架构策略包括物理接口策略，如思科发现协议（CDP）、链路层发现策略（LLDP）、最大传输单元（MTU）和链路聚合控制协议（LACP）。

虚拟机管理（VMM）域自动从与 AEP 关联的接口策略组中导出物理接口策略。

- 可以使用 AEP 覆盖策略为 VMM 域指定不同的物理接口策略。该策略在下列场景中非常实用：某个管理程序通过一个中级第 2 层节点与边缘交换机连接，在边缘交换机和管理程序物理接口处需要一个不同的策略。例如，你可以在边缘交换机和第 2 层节点之间配置 LACP。同时，你可以通过禁用 AEP 覆盖策略下面的 LACP 来禁用管理程序和第 2 层交换机之间的 LACP。

在边缘交换机上部署 VLAN 时需要 AEP。可以在不同的边缘交换机之间重复使用封装池（如 VLAN）。AEP 向物理基础架构暗中提供 VLAN 池（关联到域）的范围。

备注

- AEP 在边缘节点设备配置 VLAN 池（以及关联的 VLAN）。实际上并没有在端口上启用 VLAN。在端口上部署 EPG 之前，不会有流量流过。
- 如果没有使用 AEP 部署 VLAN 池，那么即使配置了 EPG，也无法在边缘端口上启用 VLAN。
  - 在符合下列条件的边缘端口上提供或启用一个特定的 VLAN：该边缘端口基于静态绑定在某个边缘端口上的 EPG 事件，或基于来自 VMware vCenter 等外部控制器的 VM 事件。
- 边缘交换机不支持重叠的 VLAN 池。不得将不同的重叠 VLAN 池与通过域进行关联的同样的 AEP 进行关联。

池

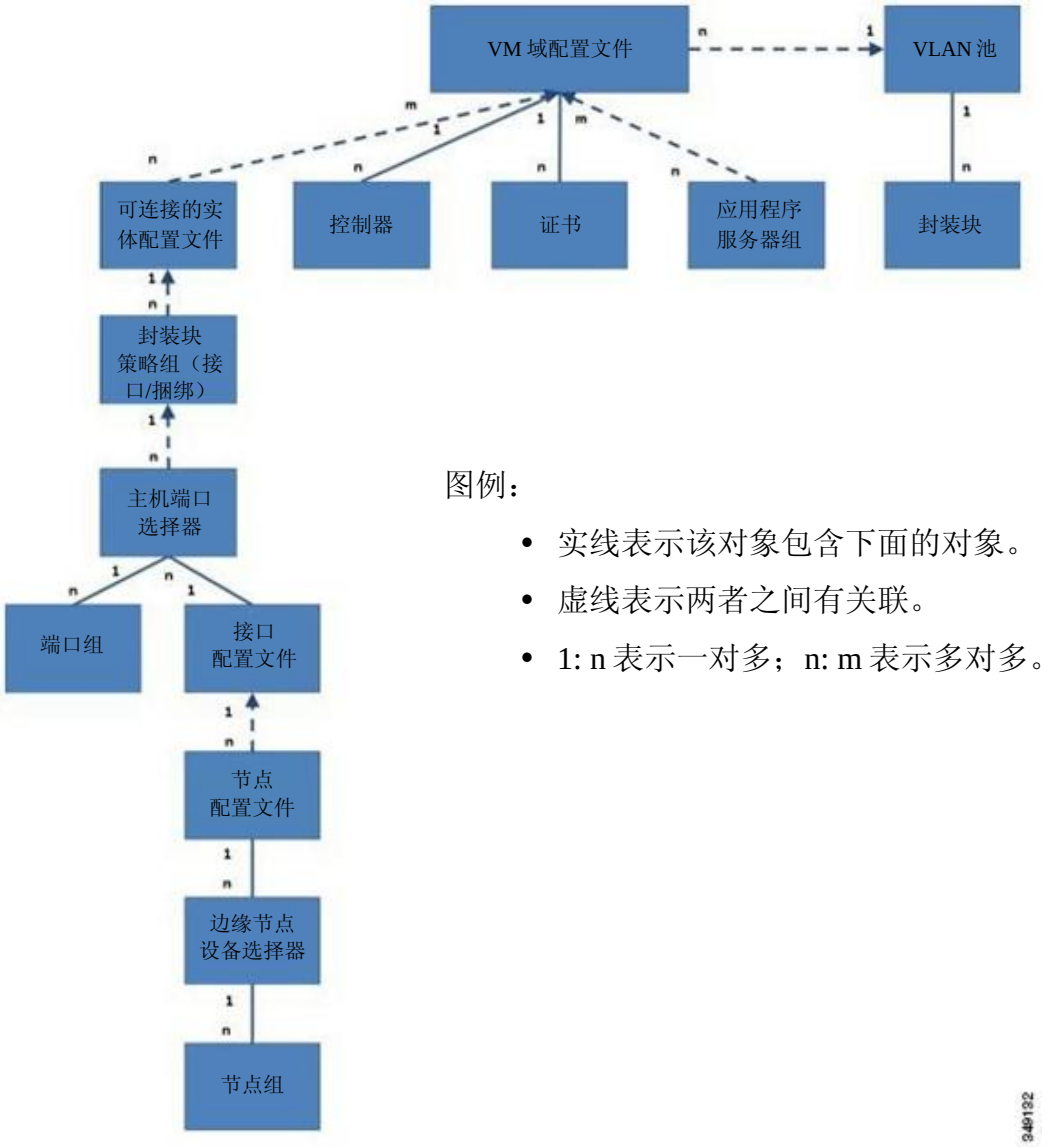
池代表一系列流量封装标识符（如 VLAN、VNID 和组播地址）。池是一种共享资源，可以被 VMM 等多个域和第 4-7 层服务使用。边缘交换机不支持重叠的 VLAN 池。不得将不同的重叠 VLAN 池与相同的可连接实体配置文件（AEP）进行关联。基于 VLAN 的池的两种类型如下：

- 动态池—由 APIC 在内部管理从而为服务器组（EPG）分配 VLAN。某个 vCenter 域仅能与一个动态池进行关联。
- 静态池— EPG 与域有关系，而域与池有关系。池包含一系列封装的 VLAN 和 VXLAN。对于静态 EPG 部署，用户定义接口和封装。封装必须位于与某个 EPG 关联的域关联的池内。

VMM 策略模型

ACI 矩阵 VM 网络允许管理员为虚拟机控制器配置连接策略。下图说明了 VM 网络策略模型的对象及其与 VM 域配置文件中其他对象之间的关系。

图 50：VMM 策略模型



VM 域配置文件包含下列 MO：

- **证书**—将用户与一个 VM 域关联起来。
- **控制器**—指定如何连接一个作为包含策略强制执行域一部分的 VMM 控制器。例如，控制器指定了与作为 VM 域一部分的 VMware vCenter 的连接。

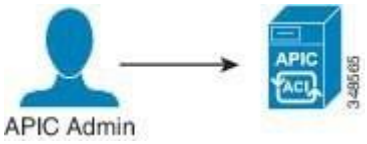
- **应用 EPG**—应用服务器组是一种策略，它规定了策略范围内端点之间的连接和可见性。
- **可连接实体配置文件**—提供了在一大组边缘端口部署管理程序的模板，也提供了 VM 域和物理网络基础架构之间的关联。可连接实体配置文件包含下列内容：
  - 策略组，指定了所用的接口策略。
  - 主机端口选择器，指定了要配置的端口和配置端口的方法。
  - 端口组，指定了一系列接口。
  - 接口配置文件，指定了接口配置。
  - 节点配置文件，指定了节点配置。
  - 边缘选择器，指定了将要配置哪个边缘节点。
  - 节点组，指定了一系列节点。
- **VLAN 池**—一个 VLAN 池指定了用于 VMM 域将要使用的 VLAN 封装的地址。

vCenter 域配置工作流程

1 APIC 管理员在 APIC 中配置 vCenter 域策略，如下图所示。APIC 管理员提供下列 vCenter 连接信息：

- vCenter IP 地址、vCenter 证书、VMM 域策略和 VMM 域 SPAN
- 策略（VLAN 池、域类型，如 VMware VDS、Cisco Nexus 1000V 交换机）
- 连接物理边缘接口（使用可连接实体配置文件）

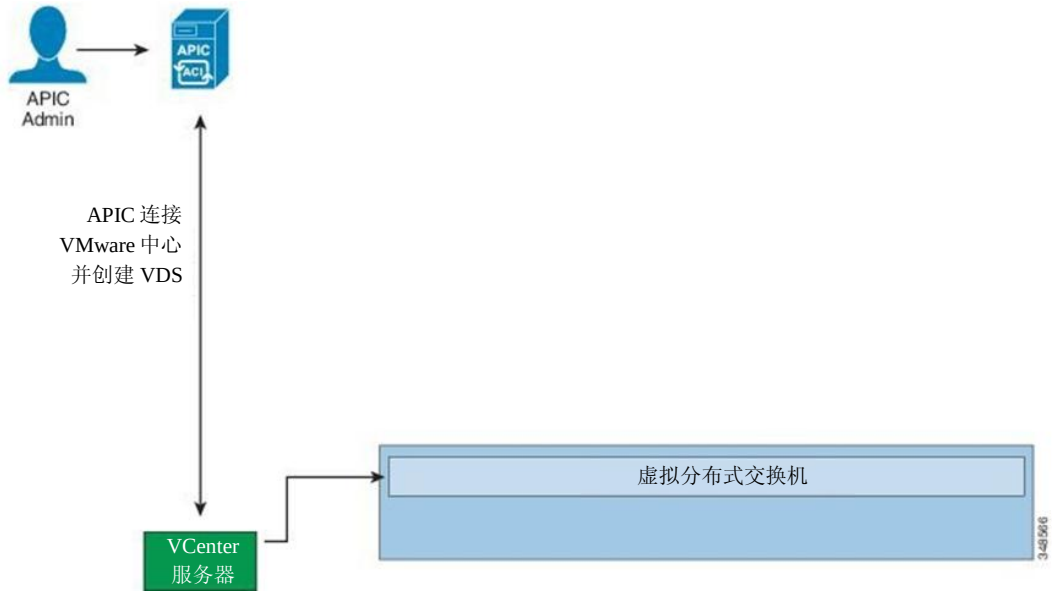
图 51：APIC 管理员配置 vCenter 域策略



APIC 自动连接 vCenter 并在 vCenter 下创建一个 VDS。如下图所示。

备注 VDS 这个名称是“VMM 域”和“数据中心”两个名词的串联

图 52：在 vCenter 下创建一个 VDS



2 APIC 管理员创建并将应用 EPG 关联到 VMM 域。

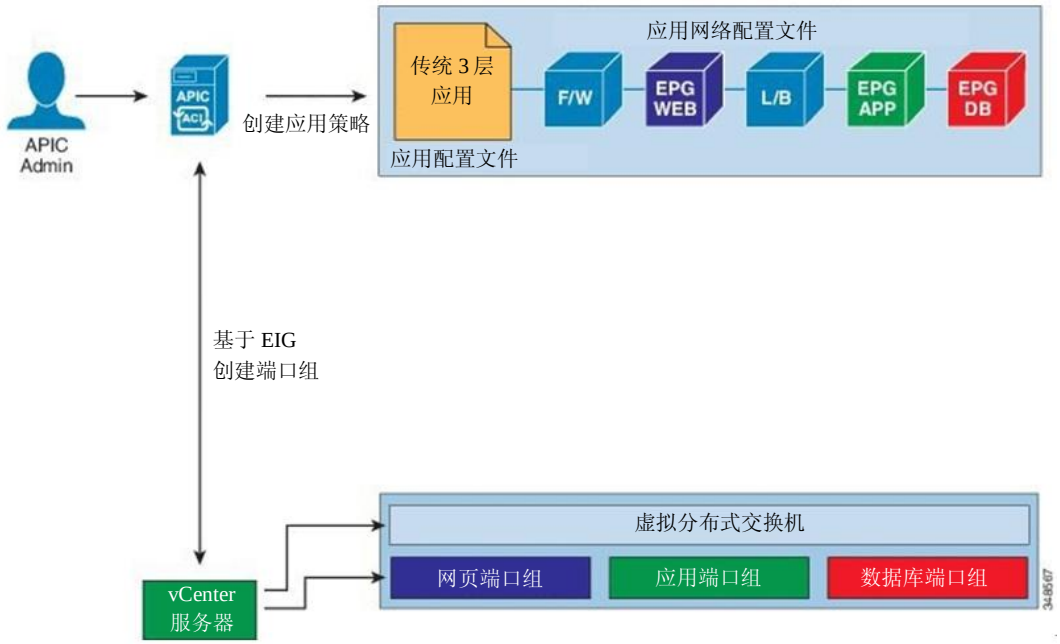
- APIC 在 VDS 下的 VMware vCenter 中自动创建端口组。
- 该过程在 VMware vCenter 中配置网络策略。

如下图所示。

备注

- 端口组名称是租户名称、应用配置文件名称和 EPG 名称的串联。
- 端口组在 VDS 下创建，由 APIC 在更早的时候创建。

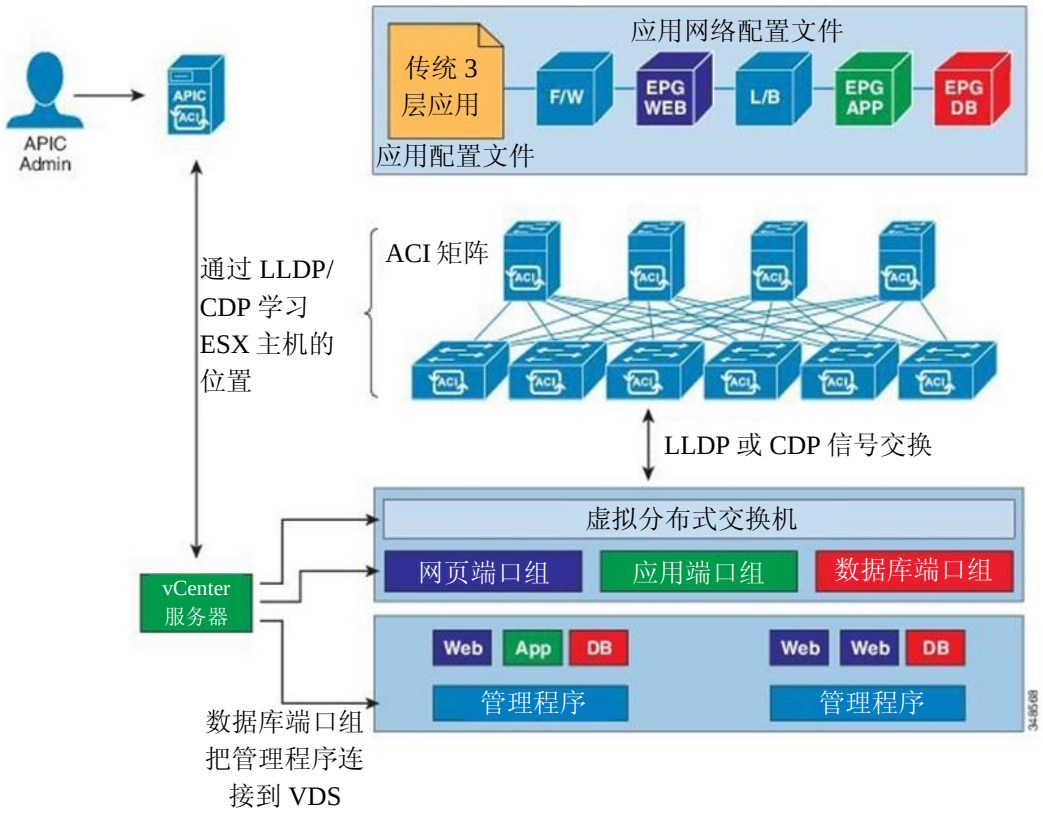
图 53：将应用 EPG 关联到 VMM 域



3 vCenter 管理员或计算管理工具箱 APIC VDS 添加 ESX 主机或管理程序，并在 APIC VDS 上分配作为上行链路的 ESX 主机管理程序端口。这些上行链路必须连接到 ACI 边缘交换机。

- APIC 通过管理程序的 LLDP 或 CDP 信息学习边缘连接的管理程序主机的位置，如下图所示。

图 54：使用管理工具把管理程序连接到 VDS

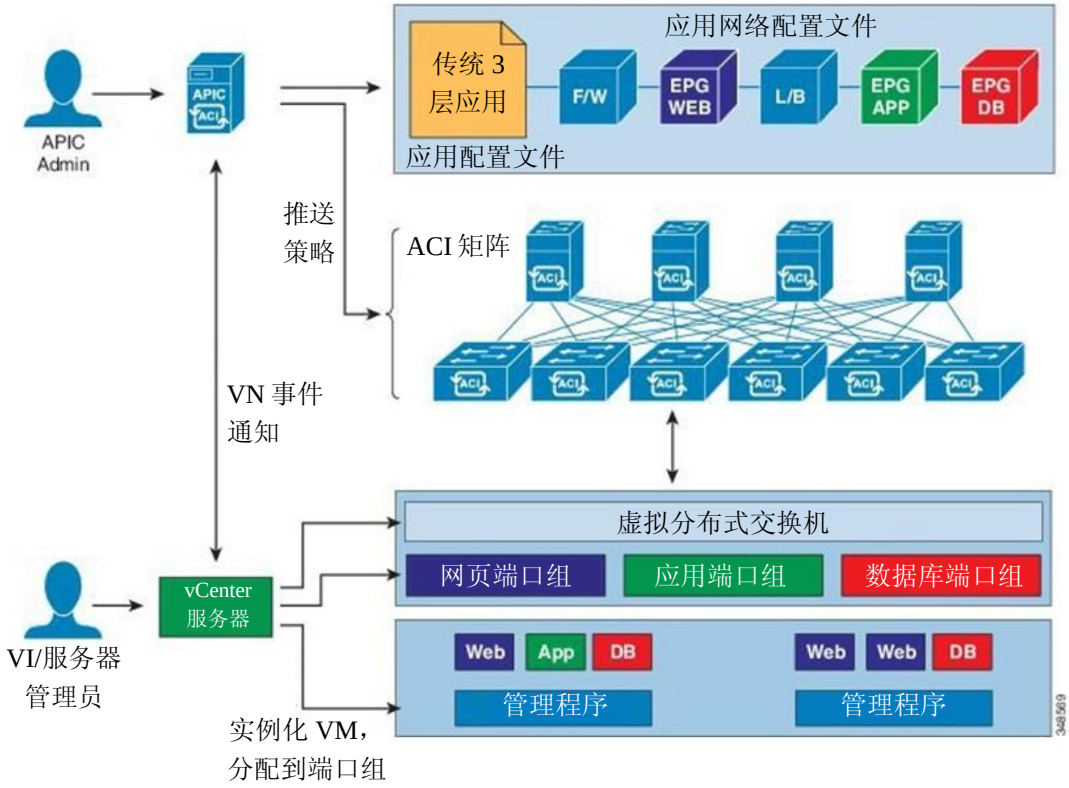


4 vCenter 管理员或计算管理工具实例化并将 VM 分配到端口组。

- APIC 基于 vCenter 时间学习 VM 放置。

- APIC 自动向 ACI 矩阵推送应用 EPG 及其关联的策略（如合约和过滤器）。如下图所示。

图 55：向 ACI 矩阵推送策略



vCenter 和 vShield 配置工作流程

该工作流程显示了 APIC 与 vShield Manager 结合，从而使用 VMware 提供的管理程序 VXLAN 功能的过程。

备注 APIC 控制并自动化在 vShield Manager 上准备和部署 VXLAN 的整个过程，这样用户就无需在 vShield Manager 上执行任何操作。

在开始配置之前，必须满足下列前提条件：

- 必须在 vShield Manager 中配置 vCenter 服务器 IP 地址。
- 必须把**矩阵基础设施 VLAN** 延伸到管理程序端口。把矩阵基础架构 VLAN 作为 VXLAN 数据封包的以太网首标中的外部 VLAN 使用。在为 VXLAN 准备 APIC VDS 时，APIC 自动把矩阵基础架构 VLAN 推送到 vShield Manager。

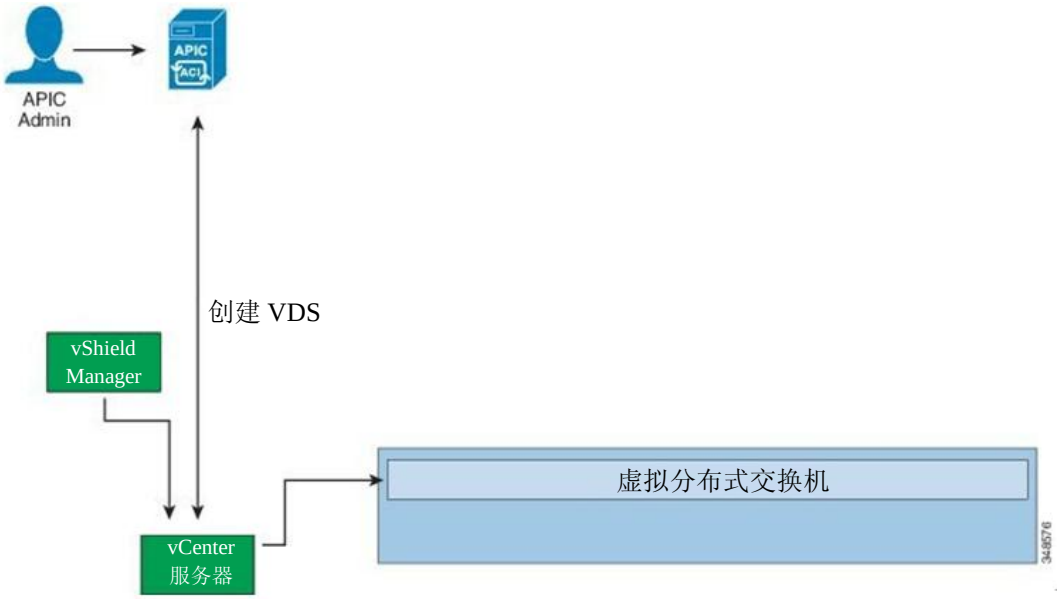
- 必须把矩阵基础设施 VLAN 延伸到管理程序端口，才能让数据路径工作。
  - 在边缘交换机面向租户的端口上，可以通过在 APIC 上创建一个可连接实体配置文件配置基础架构 VLAN。（关于创建可连接实体配置文件的信息，请参考《APIC 启动指南》。）
  - 如果管理程序和边缘交换机之间有任何中级第 2 层交换机，网络管理员必须在中级第 2 层节点上手动配置基础架构 VLAN。

1 APIC 管理员在 APIC 中配置 vCenter 和 vShield 域策略。

备注
- APIC 管理员必须在 APIC 上提供 vShield Manager 和 vCenter 服务器之间的关联。
- APIC 管理员必须提供 VXLAN 需要的段 ID 和组播地址池。vShield Manager 中的段 ID 不得与 APIC 上配置的其他 vShield Manager 中的池重叠。

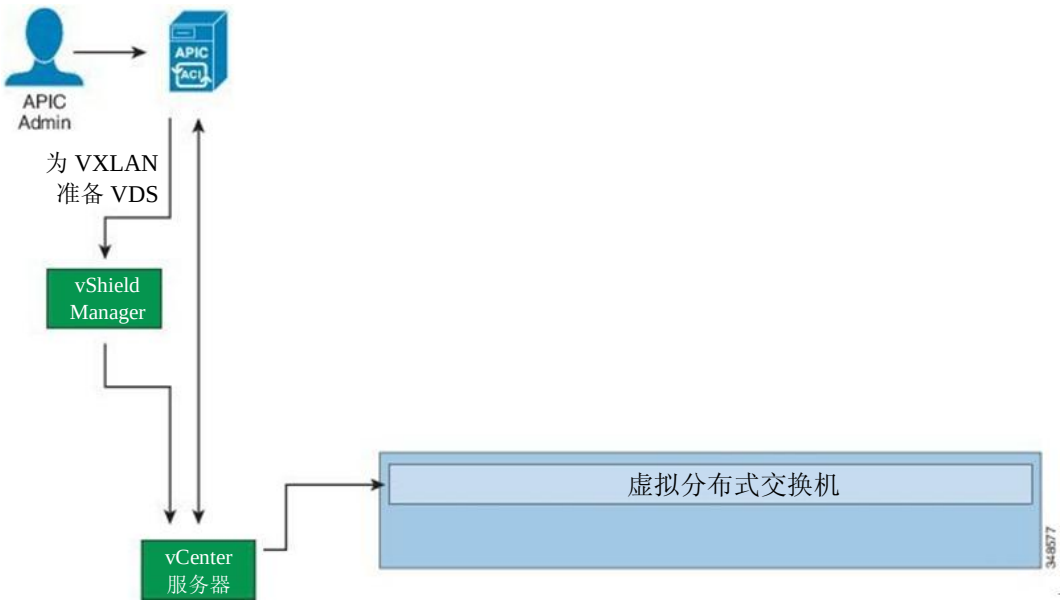
a APIC 连接 vCenter 并创建 VDS 如下图所示。

图 56：连接 vCenter 并创建 VDS



b APIC 连接至 vShield Manager，推送段 ID 和组播之地池并为 VXLAN 准备 VDS。如下图所示。

图 57：连接至 vShield Manager 并为 VXLAN 准备 VDS

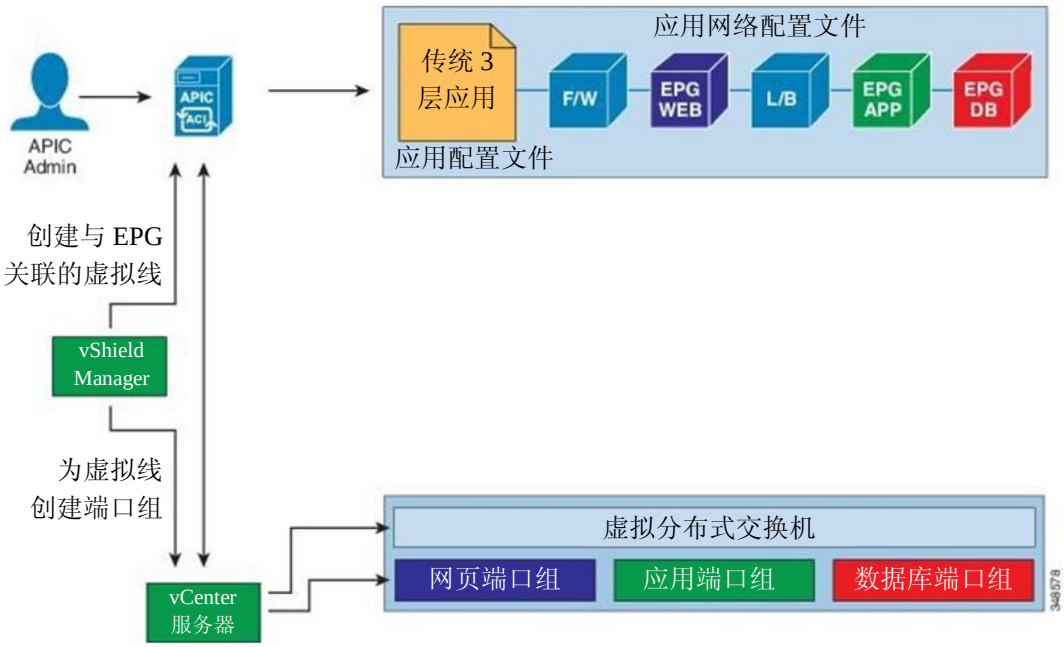


2 APIC 管理员创建应用配置文件和 EPG 并将其与 VMM 域关联。如下图所示。

- APIC 在 VDS 下的 vShield Manager 中自动创建虚拟线。
- APIC 从发送自 vShield Manager 的 VXLAN 虚拟线中读取段 ID 和组播地址。
- vShield Manager 推送虚拟线，作为 VDS 下 vCenter 服务器的端口组。

备注 虚拟线的名称是租户名称、应用配置文件名称和 EPG 名称的串联。

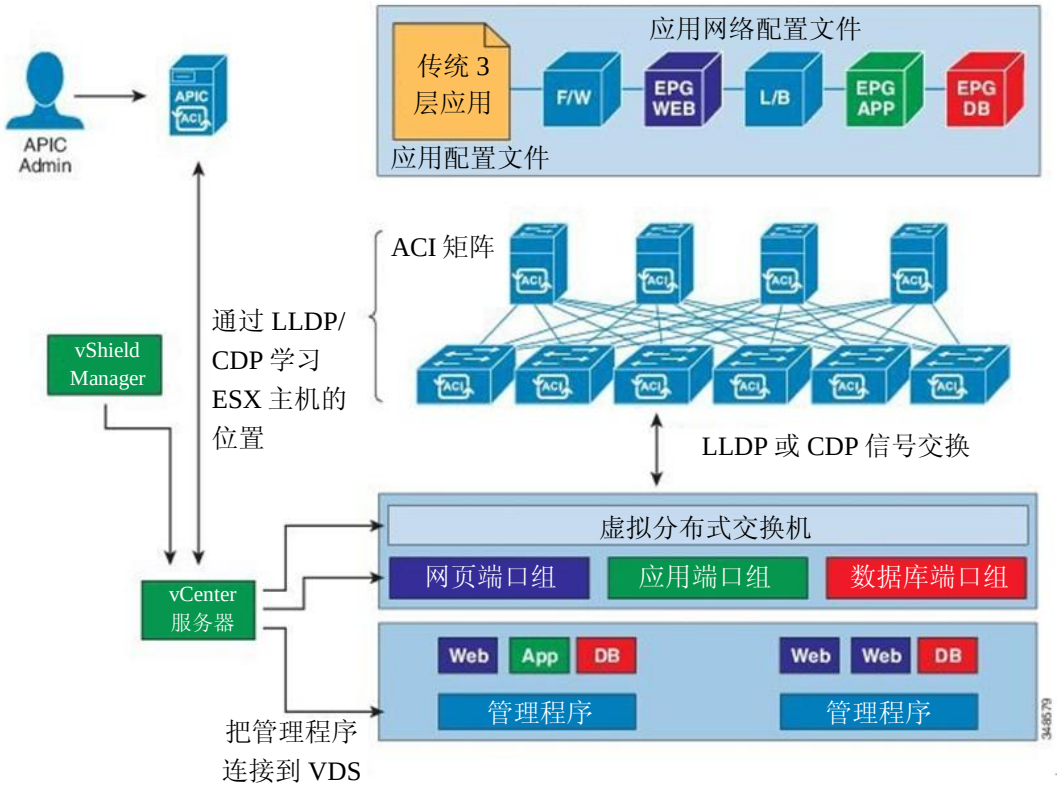
图 58：创建应用配置文件和 EPG



3 vCenter 管理员或计算管理工具将管理程序附加到 VDS。如下图所示。

• APIC 使用来自管理程序的 LLDP 或 CDP 信息学习边缘连接的管理程序主机的位置。

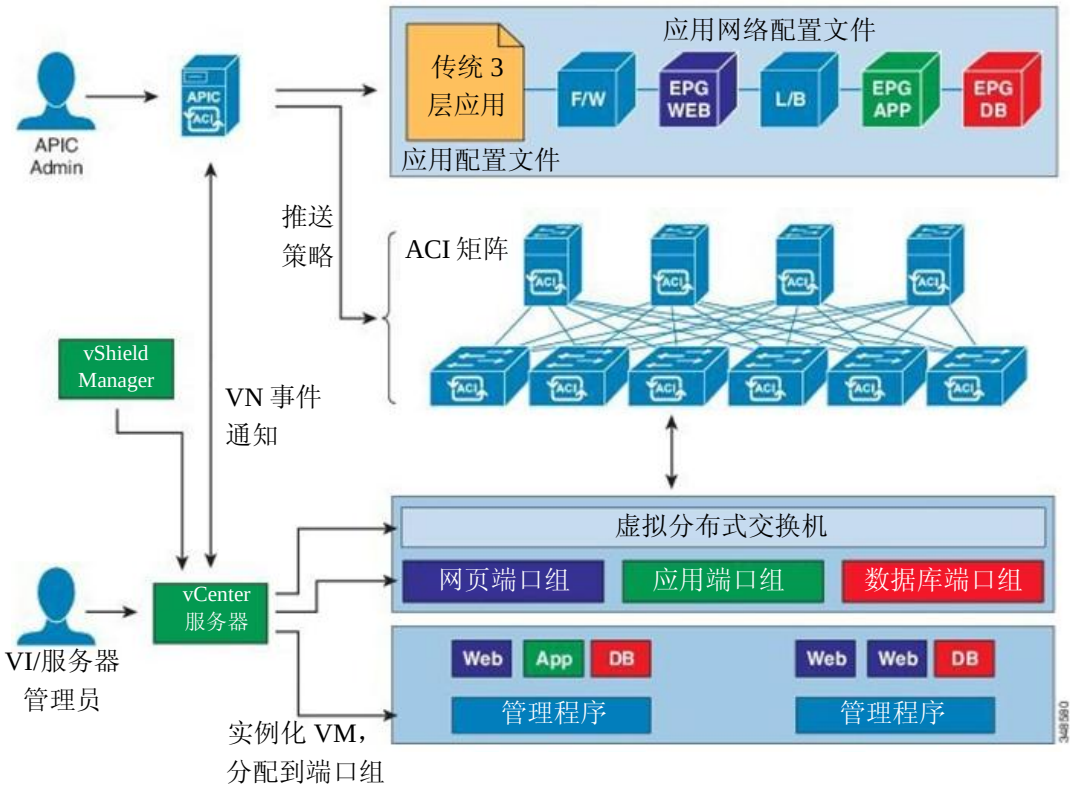
图 59：将管理程序附加到 VDS



4 vCenter 管理员或计算管理工具实例化并将 VM 分配到端口组。

APIC 向 ACI 矩阵自动推送策略。如下图所示。

图 60：向 ACI 矩阵推送策略



创建应用 EPG 策略解析和部署即时性

当某个 EPG 关联到一个 VMM 域时，管理可以选择解析和部署偏好用于指定推送策略的时间。

解析即时性

- 立即—规定一旦管理程序连接到 VDS 就把 EPG 策略（包括合约和过滤器）下载到关联的边缘交换机软件。LLDP or OpFlex 权限用于把管理程序解析到边缘节点附件。
- 按需—规定仅当某个 pNIC 附加到管理程序连接器且某个 VM 被放置到端点组中时，才把某个策略（如 VLAN、VXLAN 捆绑、合约或过滤器）推送到边缘端点。

部署即时性

一旦把策略下载到边缘软件，仪器即时性可以规定把策略推送到硬件策略 CAM 的时间。

- 立即—规定一旦把策略下载到边缘软件就在硬件策略 CAM 中对策略进行编程。
- 按需—规定仅当通过数据路径接收到第一个封包时才在硬件策略 CAM 中对策略进行编程。该过程有助于优化硬件空间。

关于删除 VMM 域的指南

遵照下面的顺序，确保删除 VMM 域的 APIC 请求自动触发关联的 VM 控制器从而正常地完成流程且 ACI 矩阵中没有“孤儿”EPG。

1. 在 VMWare vSphere 分布式交换机（VDS）或思科 AVS（AVS）中，VM 管理员从构成 ACI VMM 域的 VM 中删除所有 EPG 和 vtep 关联。完成该步骤后，VM 控制器触发 APIC 从 ACI 矩阵中删除这些 EPG。
2. 在 VM 控制器中，VM 管理员从构成 VMM 域中的虚拟交换机中删除所有虚拟适配器（vnics）。

备注 VM 管理员不应当删除虚拟交换机（如 VDS 或 AVS）；允许 APIC 在完成下面第 3 步后触发虚拟交换机的删除。如果管理员在 VMM 域在 APIC 删除之前从 VM 控制器中删除虚拟交换机，那么 APIC 中的 EPG 可能成为“孤儿”。

3. 在 APIC 中，删除 VMM 域。APIC 触发 VMM 域中虚拟交换机的删除，然后从 APIC 中删除 VMM 域。

如果违反了该顺序，VM 控制器也会删除与 APIC VMM 域关联的虚拟交换机。
VMM 域：在该场景下，VM 管理员必须从 VM 控制器中手动删除 EPG 和 vtep 关联，删除对应的 vnics，然后删除之前与 APIC VMM 域关联的虚拟交换机。

VMM 组件按需刷新

被激活的组件提供了一个手动激活选项，用于在 VMM 控制器和 APIC 之间拉动和再次同步组件。被激活的组件会从不同步的场景中即时恢复。被激活的组件仅适用于 vCenter VMM 控制器（scope:vm）。在正常场景下不需要它，使用它时应当谨慎，因为对 VMM 控制器而言组件同步是一种繁重的操作。

APIC 启动 vCenter 组件轮询。把主机、VM、DVS、上行链路端口组、NIC 等作为初始 VMM 控制器创建的一部分进行检索。通过事件订阅机制学习 vCenter 中的进一步变化。这允许 APIC VMM 管理器向 APIC 策略管理器发送端点附加/解除的升级信息，APIC 策略管理器下载更新的策略到 leave 节点交换机。

当发生过程重启、领导变化或背景周期 24 小时组件审计时，APIC 进行组件轮询，从而在 VMM 控制器和 APIC 之间保持 VMM 组件同步。偶尔 vCenter 不能提供 APIC 正确组件时间通知。在这种情况下，被激活的组件帮助 APIC 与 vCenter 保持同步。

关于向 ACI 带内 VLAN 迁移 Vcenter 管理程序 VMK0 的指南

按照下面的指南把默认 Vcenter 管理程序 VMK0 带外连接迁移到 ACI 带内端口。ACI 矩阵基础架构管理员通过必要的策略配置 APIC，然后 Vcenter 管理员把 VMK0 迁移到相应的 ACI 端口组。

创建必要的管理 EPG 策略

ACI 矩阵基础架构管理员创建下列管理租户和 VMM 域策略：

- 在 ACI 管理租户中创建一个管理 EPG。
- 将管理 EPG 与目标 VMM 域进行关联。如此一来，APIC 为管理 EPG 分配一个 VALN 并在 Vcenter DVS 下创建一个端口组。
- 注意 APIC 将哪一个 VLAN 分配到了管理 EPG。
- 找出 ESX 连接的 ACI 边缘交换机端口。
- 使用 APIC 分配给新创建的管理 EPG 的 vlan，创建管理 EPG 到这些端口的静态捆绑。

向带内 ACI VLAN 迁移 VMK0

Vcenter 默认在虚拟化层管理接口上配置默认 VMK0。上面创建的 ACI 策略允许 Vcenter 管理员向静态捆绑到 ACI 边缘交换机入站端口的端口组迁移默认 VMK0。该操作会释放管理程序管理端口。

第 9 章

第 4 至 7 层服务插入

本章包括以下部分：

第 4 至 7 层服务插入

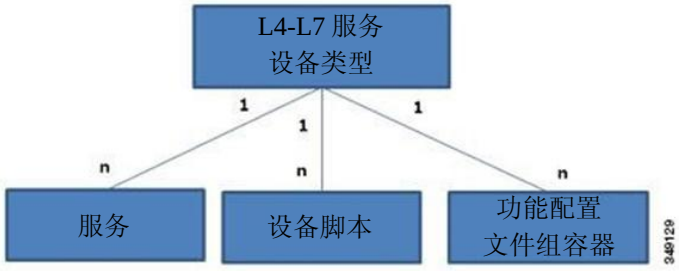
思科应用策略基础架构控制器（APIC）管理网络服务。策略被用于插入服务。APIC 服务集成提供了一个生命周期自动化框架，使得系统在服务上线或下线时能够动态响应。整个网络可用的共享服务由矩阵管理员进行管理。针对单一租户的服务由租户管理员进行管理。

APIC 提供自动化服务插入同时充当策略控制的中心点。APIC 策略既管理网络矩阵也管理服务设备。APIC 可以自动配置网络，从而让流量流过服务。APIC 也能按照应用的需求自动配置服务。传统的服务插入需要复杂的流量定向技术，而采用这种方法之后，组织可以让服务插入自动化，化解了上述挑战。

第 4 至 7 层策略模型

第 4 至 7 层服务设备类型策略包括重要的被管对象，包括包和设备脚本支持的服务。下图显示了第 4 至 7 层服务设备类型策略模型的对象。

图 61：第 4 至 7 层策略模型



第 4 至 7 层服务策略包含下列内容：

- **服务**—包含针对设备提供的所有功能（如 SSL 卸载和负载均衡）的元数据。MO 包含连接器名称、封装类型（如 VLAN 和 VXLAN）和任何接口标签。
- **设备脚本**—表示设备脚本处理程序，包含了关于脚本处理程序相关属性的元信息，包括名称、包名称和版本。
- **功能配置文件组容器**—包含服务设备类型可用功能的对象。功能配置文件包含组织到文件夹的设备支持的所有可配置参数。

服务图

思科以应用为中心的基础架构（ACI）把服务当作应用的一个完整部分。需要的任何服务都被视为一个来自思科应用策略基础架构（APIC）在 ACI 矩阵上被实例化的服务图。用户针对应用定义服务，同时服务图识别应用需要的网络或服务功能。每个功能都可被表示为一个节点。

在 APIC 中配置服务图后，APIC 按照服务图中规定的服务功能要求自动配置服务。APIC 还按照服务图中规定的服务功能的需要自动配置网络，这不需要对服务设备进行任何更改。

服务图可被表示为某个应用的两层或多层，层之间插入了相应的服务功能。

服务设备在图内执行服务功能。提供服务图所需服务时可能需要一个或多个服务设备。一个或多个服务功能可由单一的服务设备执行。

服务图和服务功能具备以下特性：

- 服务器组（EPG）可基于某个策略对发送的或接收的流量进行过滤，流量的子集可以被重新发送到服务图的不同边缘。

- 服务图的边缘具有方向性。
- 可以把 tap（基于硬件的封包复制服务）连接到服务图中的不同点。
- 基于策略，可以在相应的（物理或虚拟）设备上实现逻辑功能。
- 服务图支持边缘的拆分和结合，它不会将管理员限制在线性服务链中。
- 在被服务设备发送之后，可以在网络中对流量再次分类。
- 逻辑服务功能可以扩展或缩减，或能以集群或 1:1 激活-待机的高可用模式部署，取决于要求。

服务图配置参数

服务图可以具备设备包指定的配置参数。配置参数也可以由 EPG、应用配置文件或租户三层地址域指定。服务图内的功能节点可能需要一个或多个配置参数。可以锁定参数值避免发生改变。

当配置服务图时，指定配置参数的数值，APIC 会将参数传给设备包内的设备脚本。设备脚本将参数数据转换为下载到设备的配置。

服务图连接

服务图连接将一个功能节点连接到另一个功能节点。

自动服务插入

虽然传统的服务插入模式支持 VLAN 和虚拟路由和转发（VRF），应用策略基础架构控制器（APIC）会把服务插入和网络服务（如安全套层（SSL）卸载、服务器负载均衡（SLB）、网页应用防火墙（WAF）和防火墙）的配置自动化，同时充当策略模型的中心点。网络服务通常由服务设备（如应用交付控制器（ADC）和防火墙）提供。APIC 策略既管理网络矩阵也管理服务设备。APIC 可以自动配置网络，从而让流量流过服务。APIC 也可以按照应用的要求自动配置服务。传统的服务插入需要复杂的流量定向技术，而采用这种方法之后，组织可以让服务插入自动化，化解了上述挑战。

设备包

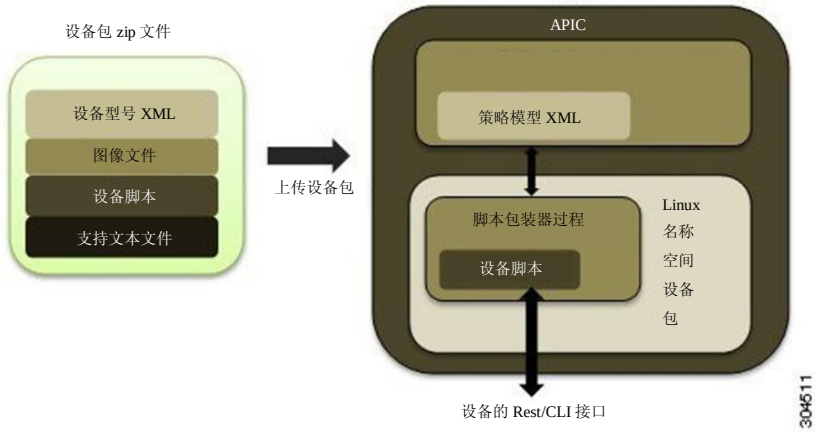
应用策略基础架构控制器（APIC）需要一个设备包用于配置和监控服务设备。设备包管理一组服务设备，为 APIC 提供关于这些设备信息，从而让 APIC 知道设备的属性和功能。设备包可以让管理员在避免中断的情况下添加、修改或删除 APIC 上的网络服务。可以通过上传设备包向 APIC 添加新的设备类型。

设备包是包含下列部分的 zip 文件：

设备规格	一个定义下列属性的 XML 文件： • 设备属性： ◦**型号**—设备的型号。 ◦**厂商**—设备的厂商。 ◦**版本**—设备的软件版本。 • 某个设备提供的功能，如负载均衡、内容交换和 SSL 终止。 • 针对每个功能的接口和网络连接信息。 • 设备配置参数。 • 针对每个功能的配置参数。
设备脚本	一个执行 APIC 和某个设备整合的 Python 脚本。APIC 事件被映射到设备脚本中定义的功能调用。
功能配置文件	一个带有厂商规定的默认值的参数配置文件。可以配置某个功能以使用这些默认值。
设备层级的配置参数	一个配置文件，指定了设备层级的某个设备需要的参数。该配置可以被一个或多个使用该设备的图共享。

下图借助设备包对 APIC 服务的自动化和插入基础架构进行了说明。

图 62：设备包基础架构



设备包可由设备厂商提供，或可由思科创建。

设备包可让管理员自动化下列服务的管理：

- 设备连接和断开连接
- 端点连接和断开连接
- 服务图提供
- 健康监控
- 警报、通知和日志
- 计数器

当设备包通过 GUI 或北行 APIC 接口上传时，APIC 为每个唯一的设备包创建一个命名空间。设备包的内容被解压并复制到命名空间。为设备包命名空间创建的文件结构如下：

```
root@apic1:/# ls
bin dbin dev etc fwk install images lib lib64 logs pipe sbin tmp usr util

root@apic1:/install# ls
DeviceScript.py DeviceSpecification.xml feature common images lib util.py
```

设备包的内容被复制到 install 目录之下。

APIC 可以解析设备型号。XML 文件中定义的被管对象被添加到由策略管理器维护的 APIC 的被管对象树中。

在设备包中定义的 Python 脚本在命名空间的脚本包装器过程中启动。禁止访问文件系统。Python 脚本可在/tmp 下创建临时文件，并能访问作为设备包一部分的任何文本文件。但是，Python 脚本不应当在文件中创建或存储任何持久性数据。

设备脚本可通过 ACI 日志框架生成调试日志。日志被写入 logs 目录下一个叫 debug.log 的循环文件。

设备包的多个版本可在 APIC 中共存，因为每个设备包版本都在自己的命名空间中运行。管理员可以选择一个特定版本用于管理一组设备。

关于设备

某个设备（也被称作逻辑设备）是一个或多个充当单个设备的具体设备。某个设备拥有逻辑接口，后者描述了设备的接口信息。在服务图模板渲染期间，功能节点连接器与逻辑接口关联起来。应用策略基础架构控制器（APIC）在服务图模板实例化和渲染过程中针对功能节点连接器分配网络资源（VLAN 或 VXLAN）并把网络资源编程到逻辑接口。

服务图模板使用一个基于管理员定义的选择策略（被称为“逻辑设备三层地址域”）的特定设备。

管理员可以在主备模式下最多设置两个具体设备。

关于具体设备

一个具体设备拥有两个具体接口。当把一个具体设备添加到逻辑设备上时，具体接口被映射到逻辑接口上。在服务图模板实例化的过程中，

在具体接口（基于这些接口与逻辑接口的关联）上对 VLAN 和 VXLAN 进行编程。

功能节点

功能节点代表单个服务功能。功能节点拥有功能节点连接器，后者表示服务功能的网络要求。

服务图内的功能节点可能需要一个或多个参数。GIA 参数可以由服务器组（EPG）、应用配置文件或租户三层地址域指定。参数也可以在管理员定义服务图时分批。可以锁定参数值避免发生改变。

功能节点连接器

功能节点连接器把某个功能节点连接到服务图并基于服务图连接器的子网与相应的桥接域和连接关联。每个连接器都与一个 VLAN 或 VXLAN 关联。连接器的每一端都被视为服务器组（EPG），把白名单下载到交换机用于启用两个功能节点之间的通讯。

终端节点

终端节点将一个服务图与合约连接起来。管理员可以通过把终端节点连接到某个合约在两个应用服务器组（EPG）之间为流量插入一个服务图。连接之后，合约的服务使用方 EPG 和服务提供方 EPG 之间的流量被重新发送到服务图。

关于权限

管理员可以向 APIC 内的角色分配权限。权限决定了允许某个角色执行的任务。管理员可以向管理员角色分配下列权限：

权限	说明
nw-svc-connectivity	• 创建一个管理 EPG。 • 创建一个通向其他对象的管理连接
nw-svc-policy	• 创建一个服务图 • 把一个服务图连接到一个应用 EPG 和一个合约 • 监控一个服务图
nw-svc-device	• 创建一个设备集群 • 创建一个具体设备 • 创建一个设备三层地址域

备注 只有基础架构管理员可以向 APIC 上传设备包。

服务自动化和配置管理

思科 APIC 可以选择性地为服务设备充当配置管理和自动化点，在服务设备和网络自动化之间发挥协调作用。思科 APIC 使用 Python 脚本与服务设备连接，在多种事件上调用特定于设备的 Python 脚本功能。

将设备脚本和定义服务设备支持的设备规格捆绑为设备包并将其安装在思科 APIC 上。设备脚本包装器使用自己的 REST 接口（首选）或 CLI 与设备连接，这取决于设备配置型号。

服务资源池

思科 ACI 矩阵可以在多个目标地址之间执行无状态的负载分布。借助这项功能，组织可以把物理和虚拟服务设备集合成服务资源池，后者可以按照功能或位置进一步集合。这些池可以视同标准的高可用性机制提供高可用性，或者它们可以被用作简单的状态服务引擎，如果发生故障，负载就会被重新分配到其他成员。任何一个选项会提供横向扩展，远远超出等价多路径（ECMP）、端口通道特性和服务应用集群，这需要共享状态。

服务资源池

如果服务设备不必与矩阵交互，思科 ACI 可以借助任何服务设备执行简单版本的资源池化，它可以执行涉及矩阵和服务设备之间协调的更加高级的池化。

第 10 章

管理工具

本章包括以下部分：

管理工具

思科以应用为中心的基础架构（ACI）工具帮助管理员、网络工程师和开发者开发、配置、调试和自动化租户与应用的部署。

关于管理 GUI

下列管理 GUI 特性提供了访问矩阵及其组件（边缘节点设备和核心设备）的权限：

- 基于通用网页标准（HTML5）。不需要任何安装程序或插件。
- 访问监控（统计、错误、事件、审计日志）、运行数据和配置数据。
- 通过单一登录机制访问 APIC 和核心设备和边缘节点设备。
- 使用第三方可用的相同的 RESTful API 与 APIC 进行通讯。

关于 CLI

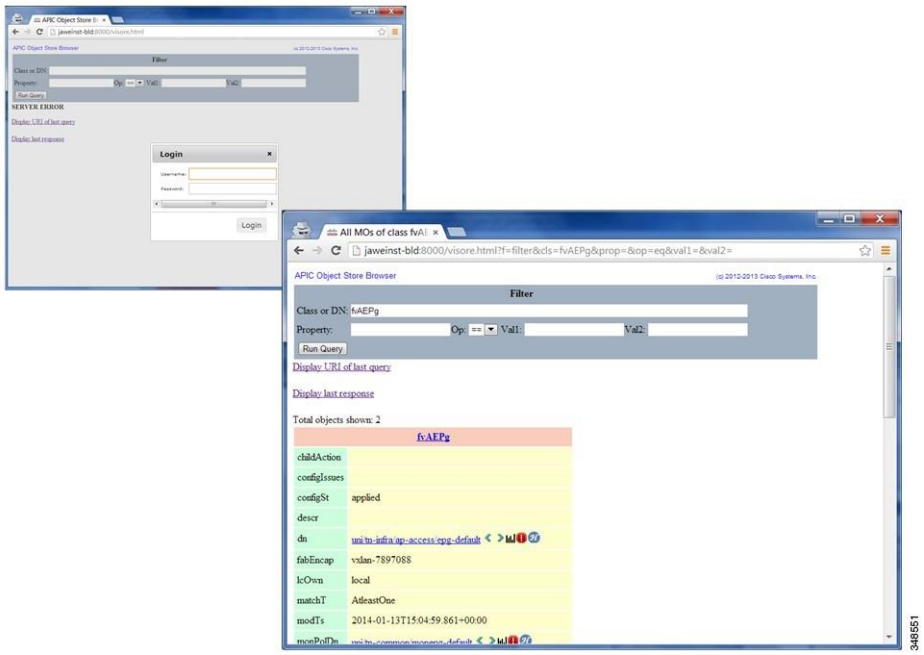
CLI 带有连接 APIC、边缘交换机和核心交换机的运行和配置接口：

- 在 Python 中从头开始部署；可以在 Python 解释器和 CLI 之间切换
- 延伸性的插件架构
- 基于三层地址域的对监控、运行和配置数据的访问权
- 通过 Python 命令或批脚本处理实现自动化

Visore 管理的对象查看器

Visore 是一种只读管理信息树（MIT）浏览器，如下所示。它通过可选过滤器支持可识别名（DN）和类查询。

图 63：Visore MO 查看器

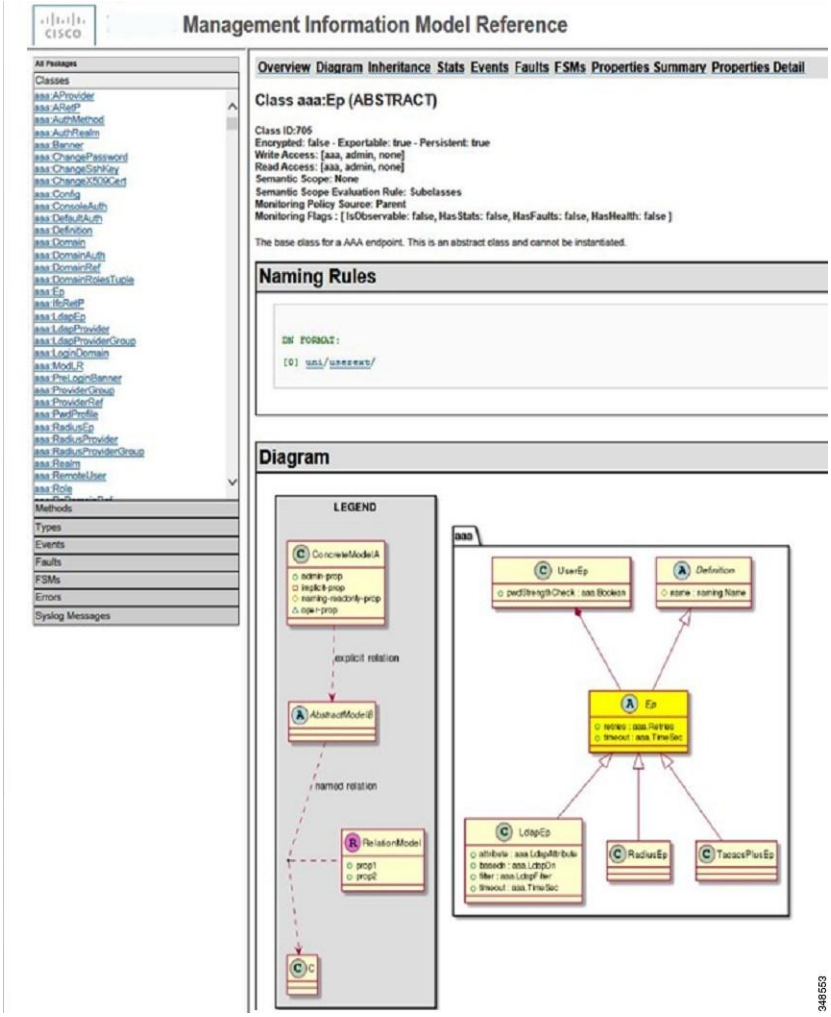


Visore 管理的对象查看器位于：http(s)://host[:port]/visore.html

管理信息模型参考

管理信息模型（MIM）包含系统及其属性中所有的被管对象。下图举例说明了管理员如何使用 MIM 研究 MIT 中的对象。

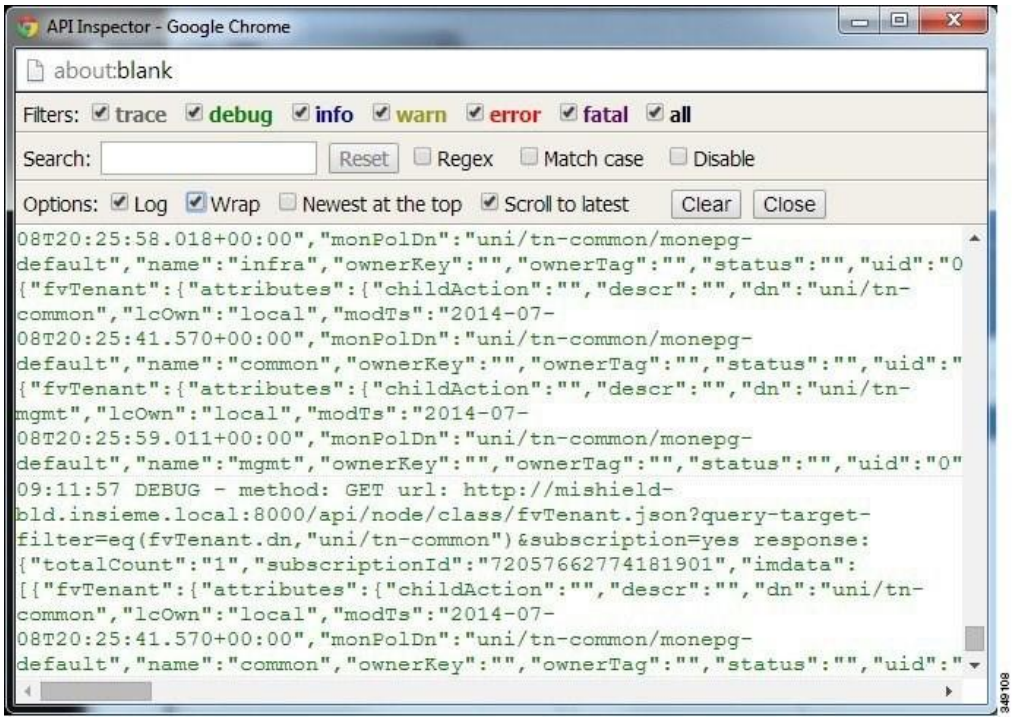
图 64：MIM 参考



API 检查器

APIC 检查器提供 APIC 处理用于执行 GUI 交互的 REST API 命令的实时显示。下图显示了 API 检查器在导航到 GUI 主租户版块时显示的 REST API 命令。

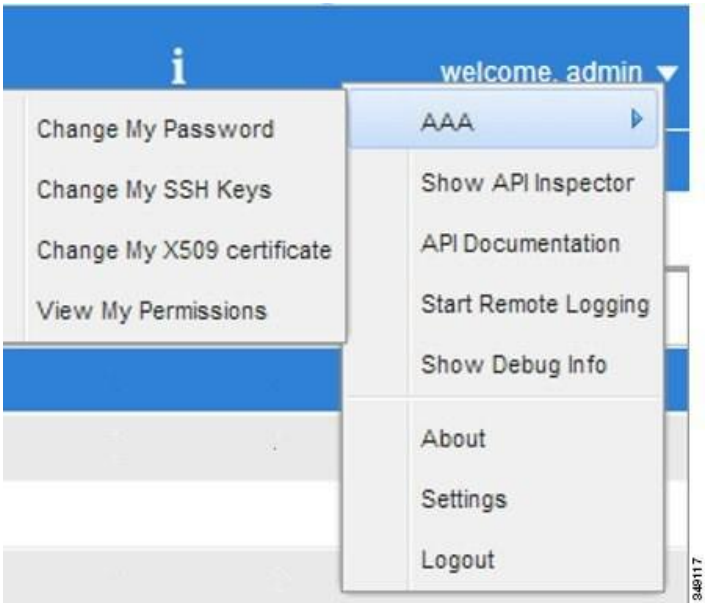
图 65：API 检查器



用户登录菜单选项

用户登录下拉菜单提供了几种配置、诊断、参考和首选项。下图显示了这个下拉菜单。

图 66：用户登录菜单选项



这些选项包括：

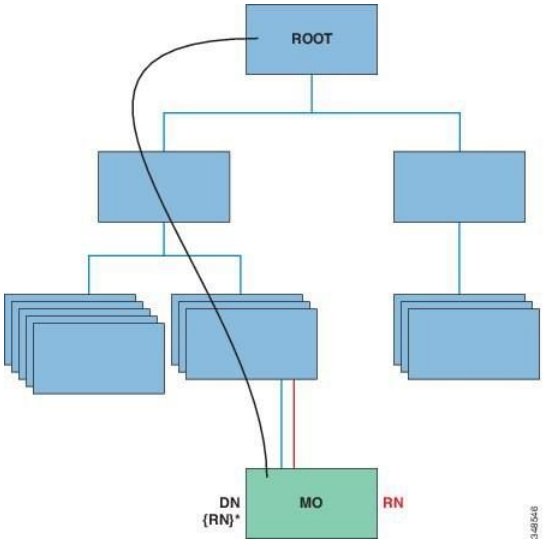
- 更改用户选项、SSH Key、X509 Certificate 和查看登录用户权限的 AAA 选项。
- 显示 API 检查器打开 API 检查器
- API 文档打开管理信息模型参考。
- 远程登录。
- 调试信息。
- 关于软件当前版本编号。
- 为使用 GUI 设置首选项。
- 注销以退出系统。

在 MIT 中定位对象

思科 ACI 使用基于信息模型的基础架构（管理信息树[MIT]），其中模型描述了可被管理流程控制的所有信息。对象实例被称作被管对象（MO）。

下图显示了可识别名称，它唯一性地表示任何给定的 MO 实例和相对名称，表示父 MO 之下本地的 MO。MIT 中的所有对象存在于根对象下面。

图 67：MO 的可识别和相对名称



系统中的每个 MO 都可通过一个独一无二的可识别名称（DN）进行标识。采用这种方法之后，可以在全局指代该对象。除了可识别名称，还可用相对名称（RN）指代每个对象。相对名称指明了某对象与其父对象之间的关系。任何给定对象的可识别名称都源自它附加在父对象可识别名称上的的相对名称。

DN 是相对名称的顺序，它以独一无二的方式对某个对象进行标识。

dn = {rn}/{rn}/{rn}/{rn}

dn =”sys/ch/lcslot-1/lc/leafport-1”

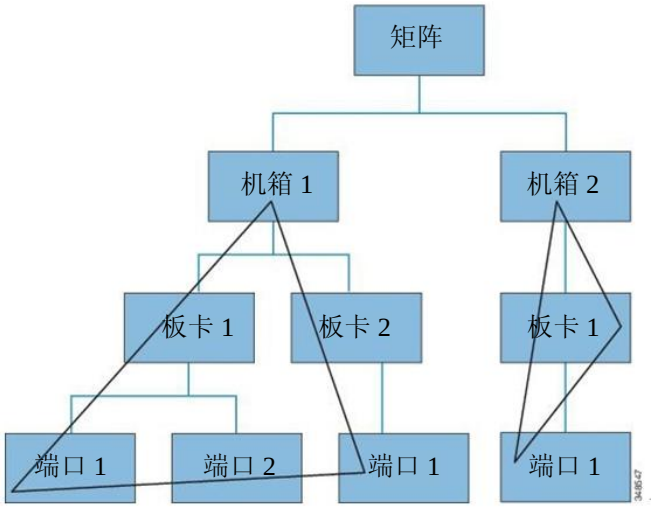
可识别名称直接映射到 URL。相对名称或者可识别名称可以用于访问某个对象，这取决于 MIT 当前的位置。

由于用于识别对象分类的树和属性系统具有层级属性，所以可以用多种方法对树进行查询从而获得被管对象的信息。可以通过可识别名称针对对象本身执行查询，也可以针对交换机机箱等对象分类，或在树级别用于发现对象的所有成员。

树级别的查询

下图显示在树级别查询的两个机箱。

图 68：树级别的查询

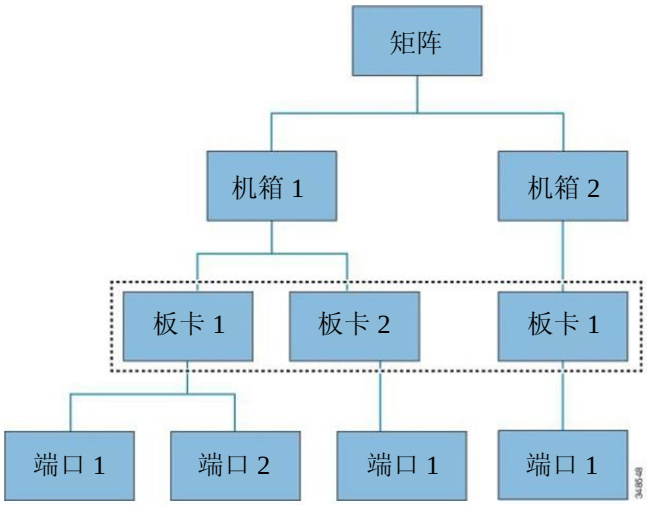


两个查询都会返回参考对象及其子对象。该过程非常有助于发现更大系统的组件。在本示例中，查询发现了给定交换机机箱的板卡和端口。

类级查询

下图显示了第二个查询类型：类级查询。

图 69：类级查询

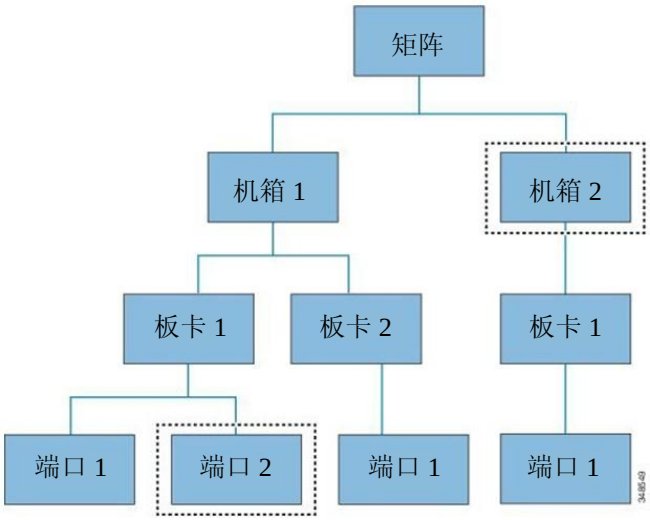


对象级查询会返回给定类的所有对象。这种方法非常有助于发现 MIT 中可用的某种类型的所有对象。在本例中，使用的类是板卡，它返回类型板卡的所有对象。

对象级查询

第三种查询类型是对象级查询。在对象级查询中，可识别名称用于返回某个特定对象。下图显示了两个对象级的查询：一个对应机箱 2 中的节点 1，一个对象是端口 2（板卡 1 机箱 1 的节点 1）。

图 70：对象级查询

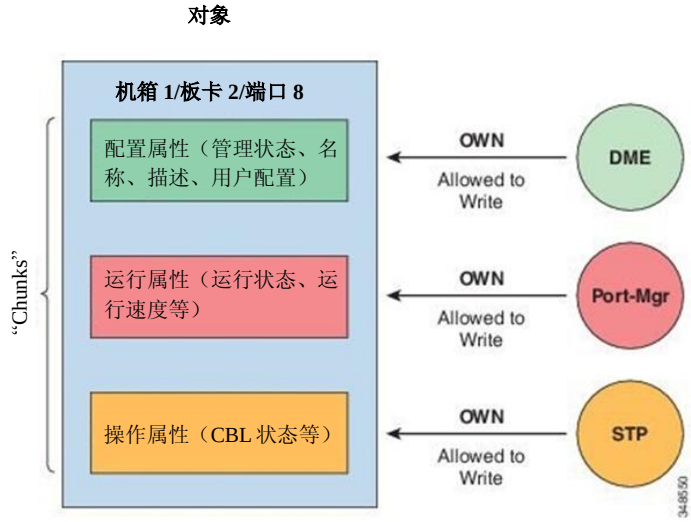


对于所有的 MIT 查询，管理员可以选择性地返回整个子树或部分子树。此外，系统中基于角色的访问控制（RBAC）机制指定返回哪些对象；仅返回用户有权查看的对象。

被管理对象的属性

思科 ACI 中被管理对象包含定义被管理对象的属性。被管理对象中的属性被分成被操作系统流程管理的组块。任何对象都有多个访问它的流程。所有这些属性都在运行时间编译，以单个对象的形式呈现到用户面前。下图举例说明了这种关系的属性。

图 71：被管理对象的属性



示例对象包含写入对象属性组块的三个流程。对象管理引擎（DME）是思科 APIC（用户）和对象之间的界面，端口管理器负责端口配置，二者与生成树协议（STP）都与该对象的组块交互。所有这些属性都在运行时间编译，以单个对象的形式呈现到用户面前。

通过 REST 接口接入对象数据

REST 是一种适用于万维网等分布式系统的软件基础架构风格。由于其更加简单的风格，REST 越来越多地替换了简单对象存取协议（SOAP）和网络服务描述语言（WSDL）等其他设计模型。思科 APIC 支持 REST 接口以便于进行针对整个思科 ACI 解决方案的编程访问。

思科 ACI 基于对象的信息模型使其非常适合于 REST 接口：URL 和 URI 直接映射到标记 MIT 上的对象的可识别名称，MIT 上的任何数据都可以被描述成一个自我包含的结构树文本文档，该文档采用 XML 或 JSON 编码。对象拥有通过可识别名称和属性识别的父子关系，通过创建、读取、更新和删除（CRUD）等一系列操作进行读取和修改。

可以使用标准 HTTP 命令在清晰定义的地址、REST URL 访问对象，用于检索和操作思科 APIC 对象数据。所用的 URL 格式可以表示如下：

<system>/api/[mo|class]/[dn|class][:method].[xml|json]?{options}

前面的 URL 的各种组件如下：

- system：系统标识符；一个 IP 地址或 DNS 可解析的主机名称
- mo | class：表示这是否是一个位于 MIT 中的被管理对象或分类级别的查询
- 分类：被查询对象的被管理对象分类（在信息模型中进行了规定）；分类名称表示为 <pkgName><ManagedObjectClassName>

- dn：被查询对象的可识别名称（MIT 中对象唯一的层级名称）
- method：对象上调用的方法的选择性表示；仅适用于 HTTP POST 请求
- xml | json：编码格式
- options:查询选项、过滤器和参数

有了为单个或一类带有 REST URL 的对象赋予地址的能力和对这些对象的访问权，管理员可以对整个对象树和整个系统进行完整的编程访问。

下面是 REST 查询的几个示例：

- 在租户 Solar 中找出所有 EPG 及其错误。
  http://192.168.10.1:7580/api/mo/uni/tn-solar.xml?query-target=subtree&target-subtree-class=fvAEPg&rsp-subtree-include=faults
- 经过过滤的 EPG 查询
  http://192.168.10.1:7580/api/class/fvAEPg.xml?query-target-filter=eq(fvAEPg.fabEncap,%20"vxlan-12780288")

配置导出/导入

所有 APIC 策略和配置数据都可以导出用于创建备份。如果导出策略允许把计划的备份或即时备份存放到远程服务器，那么这可以配置。可以配置计划的备份用于执行定期或循环的备份工作。默认情况下，可以备份所有的政策和租户，但管理员仅能有选择性地指定管理信息树的特定子树。可以通过导入策略把备份导入 APIC，这样可以让系统恢复到之前的配置。

配置数据库分片

APIC 集群使用一种被称作“分片”大型数据库技术。这种技术让 APIC 生成和处理的数据集变得可扩展且可靠。APIC 配置数据被分成逻辑上有界限的子集“分片”，类似于数据集分片。分片是数据管理的单位，APIC 以下列方式管理分片。

- 每个分片都有三个复制品。
- 分片在构成 APIC 集群的设备之间均匀的分布。

每个 APIC 设备上由于一个或多个分片。分片数据的分配基于预先设定的哈希功能，静态分片布局决定了分配到设备的分片。

配置导出

下图显示了上述过程如何在配置导出策略时发挥作用。

图 72：配置导出策略的工作流程

APIC 以下列方式应用该策略：

- 完整的系统配置备份每月执行一次。
- 该备份以 XML 格式存储在 BigBackup FTP 站点。
- 策略被触发（处于激活状态）。

配置导入

管理员可以创建一个导入策略，以下列两种模式之一导入：

- 尽力—忽略分片内无法导入的对象。如果将来配置的版本不与现有系统兼容，能够导入的正在导入的过程中不兼容的分片不能导入。
- 原子—在处理能够导入的分片时，忽略包含不能导入的对象的分片。如果将来配置的版本与现有系统不兼容，导入终止。

导入策略支持下列模式和类型组合：

- 尽力合并—导入的配置与现有配置合并，但忽略无法导入的对象。
- 原子合并—导入的配置与现有配置合并，但忽略包含无法导入的对象的分片。
- 原子替换—用导入的配置数据覆盖现有配置。删除存在于现有配置但不存在于导入配置的任何对象。把在现有配置中有“子女”但在导入配置中没有“子女”的对象从现有配置中删除。例如，如果现有配置有两个租户 solar 和 wind，但导入的备份配置在租户 wind 创建之前保存，那么租户 solar 从备份中恢复，而租户 wind 被删除。

下图显示了上述过程如何在配置导入策略时发挥作用。

图 73：配置导入策略的工作流程

APIC 以下列方式应用该策略：

- 创建一个策略用于把整个系统配置从阅读备份中恢复。
- 原子替换模式进行下列操作：
  - 覆盖现有配置。
  - 删除导入文件中不存在的现有配置对象。
  - 删除不存在的子对象。
- 策略未被触发（可用但处于未激活状态）。

技术支持、统计、核心

管理员可以配置 APIC 中的导出策略用于导出统计数据、技术支持收集、错误和事件，从而处理从矩阵（APIC 和交换机）到外部主机的核心文件和调试数据。导出数据能以多种格式存在，包括 XML、JSON、web socket、SCP 或 HTTP。导出数据支持订阅，支持流式传输、定期或按需。

管理员可以配置策略详细信息，如传输协议、压缩算法和传输频率。有权使用 AAA 的用户可以配置策略。实际传输的安全机制基于用户名和密码。在内部，某个策略元素负责数据的触发。

第 11 章

监控

本章包括以下部分：

故障、错误、事件、审计日志

备注 如需了解关于故障、事件、错误和系统消息的信息，请参考《思科 APIC 故障、事件和错误消息用户指南》和《思科 APIC 管理信息模型参考》，后者是一种基于网页的应用。

APIC 采用一系列的“管理对象”（MO）负责一个针对 ACI 矩阵系统管理和运行状态的全面的、现时的运行期描述。系统根据系统的运行期状态，或由系统或用户创建的管理策略生成的故障、错误、事件和审计日志数据。

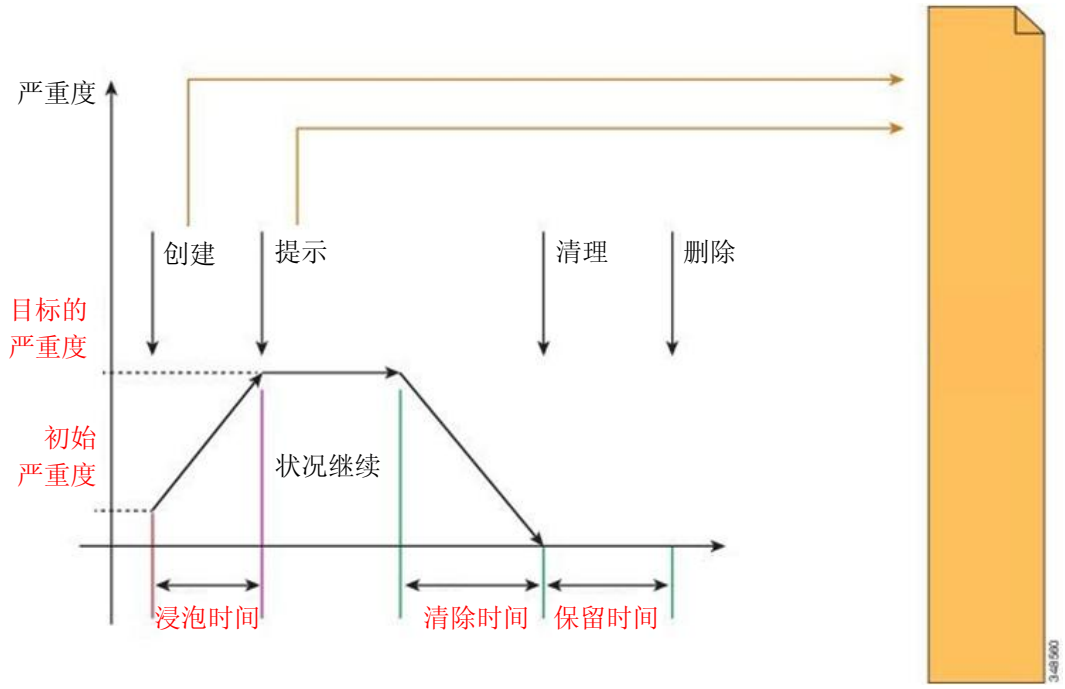
故障

根据系统运行时的状态，APIC 会自动检测异常并创建代表这些异常的故障对象。故障对象包含多种属性，目的是帮助用户诊断问题，评估问题影响并提供一个解决方案。

例如，如果系统检测到一个与端口有关的问题，如奇偶校验误差率较高，那么会自动创建一个故障对象并将其放置到管理信息树，作为端口对象的“子”。如果检测到多次的同样故障状态，不会创建额外的错误对象。

纠正触发故障的条件之后，按照故障生命周期策略的规定保留故障对象一段时间，最后删除。如下图所示。

图 74：故障生命周期



生命周期代表问题的当前状态。它开始于浸泡时间，即问题首次删除时，然后变为“提示”，如果问题持续存在则保持该状态。状况清除之后，它变为一种被称作“提示-清除”，在这种情况下我们认为状况依然潜在存在。然后让变为“清除时间”，最后是“保留”。此时，我们认为问题得到解决，故障对象被保留下来，以便让用户看到最近解决的问题。

每次生命周期过渡时，系统自动创建故障记录对象将其记录下来。故障记录一旦创建便无法修改，只有当故障数量超过故障保留策略规定的最大值时才被删除。

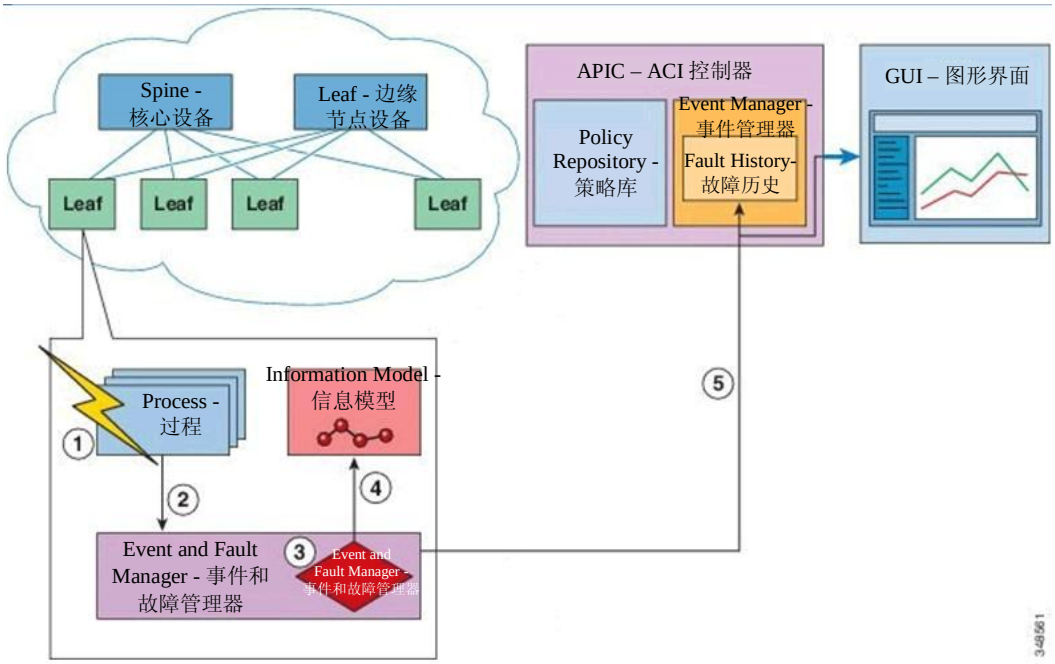
严重度是状况对系统服务能力的影响的一种估计。可能的值为警告、较小、重大和危急。如果故障的严重度为警告，表明潜在问题（例如包括配置不完整或不一致）当前不会影响任何已经部署的服务。如果故障严重度为较小和重大，那么所提供的服务可能恶化。“危急”表明一个重大中断正在严重地影响某个服务或对其产生完全损害。描述包含对事件的说明，目的是提供额外信息，帮助排除故障。

事件

事件记录是系统创建的对象，用于记录已经发生的、用户可能感兴趣的特定状况。这些记录包含了受影响的对象的完全合格域名（FQDN）、时间戳和对状况的描述。例如链接状态过渡、协议启动和停止以及新硬件组件删除。事件记录一旦创建便无法修改，只有当事件数量超过事件保留策略规定的最大值时才被删除。

下图显示了故障和事件报告的过程。

图 75：故障和事件报告/导出



1 过程检测到故障状况。

2 过程通知事件和故障管理器。

3 事件和故障管理器按照故障规则处理通知。

4 事件和故障管理器按照故障策略在 MIM 中创建故障实例并管理故障的生命周期。

5 事件和故障管理器向 APIC 和连接的客户通知状况变更。

6 事件和故障管理器触发额外的操作（如系统日志、自动通报等）。

错误

APIC 错误消息通常在 APIC GUI 和 APIC CLI 中显示。这些错误消息特定于用户正在执行的动作或用户正在配置或管理的对象。这些信息可能为：

- 信息性消息，提供关于被执行动作的帮助和提示
- 警告性消息，提供关于与对象相关的系统错误的信息，如用户正在配置或管理的用户账号或服务配置文件
- 有限状态机（FSM）状态消息，提供关于 FSM 阶段状态的信息

很多错误消息包含一个或多个变数。APIC 更改这些变数的信息是取决于信息的背景。某些信息可能由不止由一种错误类型产生。

审计日志

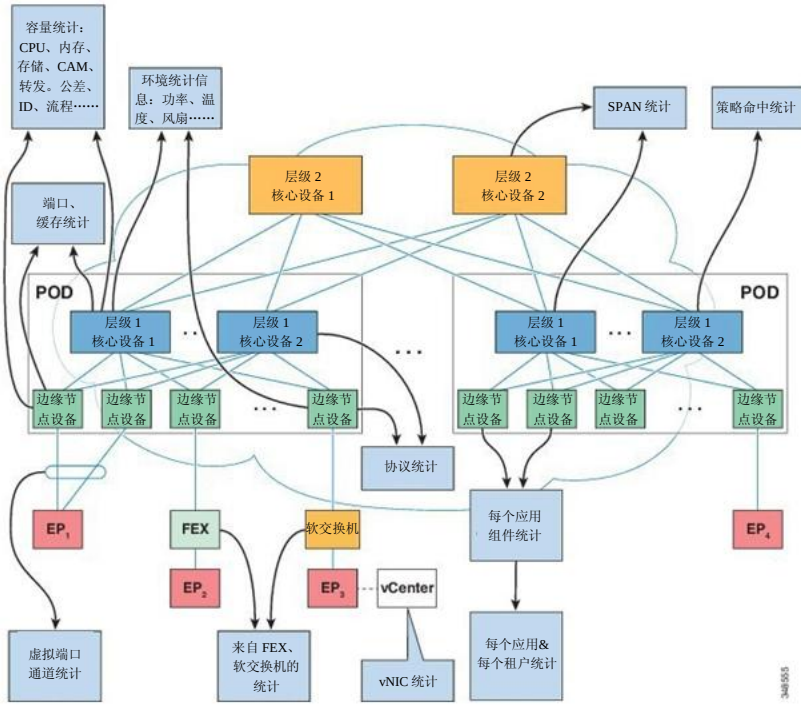
审计记录是系统创建用于记录用户发起的操作的对象，如登录/注销和配置更改。这些记录包含了正在执行该操作的用户的名称、时间戳、对操作的描述以及受影响对象的 FQDN（若适用）。审计记录一旦创建便无法修改，只有当审计数量超过审计保留策略规定的最大值时才被删除。

统计属性、层级、阈值和监控

统计信息可为趋势分析和故障排除提供支持。能够以持续或按需的方式配置统计信息收集用于收集统计信息。统计信息提供对被观测对象的实时测量。可以在累积计数器和计量器中收集统计信息。如下图所示。

策略会规定收集哪些信息，时间间隔为多少以及需要采取哪些动作。例如，如果在入口 VLAN 丢失的封包的阈值大于 1000/秒，策略会在 EPG 上提示一个故障。

图 76：统计信息的各种来源



统计信息的收集来自各种来源，包括接口、VLAN、EPG、应用配置文件、ACL 规则、租户或内部 APIC 过程。统计按照 5 分钟、15 分钟、1 小时、1 天、1 周、1 个月、1 季度或 1 年的取样间隔收集数据。更短的持续间隔对应更长的间隔。

可用的统计数据有多种，包括上次数值、累积、周期、变化率、趋势、最大值、最小值、平均值。收集和保留时间值可以更改。策略可以指定从系统当前状态还是从历史状态还是从两种来源收集统计信息。例如，某个策略可以规定在 1 个小时每隔 5 分钟收集一次历史统计信息。1 小时是一个移动的窗口。当第一个小时结束后，每 5 分钟添加的统计信息，最初 5 分钟的数据会被丢弃。

配置监控策略

管理员可以在下列四种广义范围内创建监控策略：

- 全矩阵内：包括矩阵和访问对象
- 接入层（也被称为基础架构）：访问端口、FEX、VM 控制器等
- 矩阵层：矩阵端口、卡片、机箱、风扇等
- 租户：EPG、应用配置文件、服务等

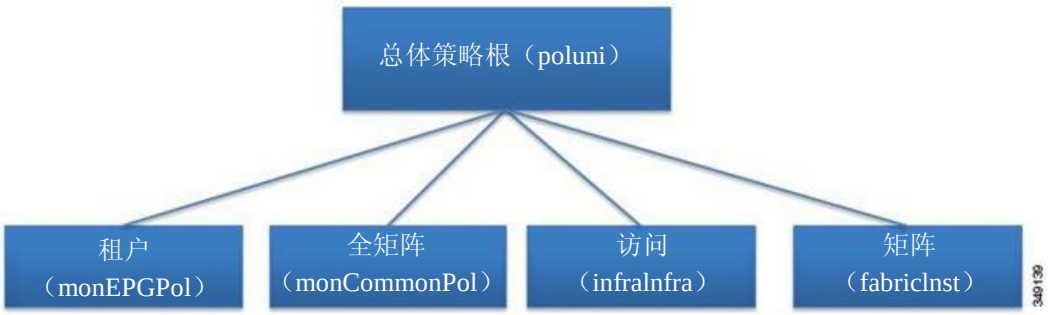
APIC 包括下列四类默认监控策略：

- monCommonPol (uni/fabric/moncommon)：适用于矩阵和接入层级
- monFabricPol (uni/fabric/monfab-default)：适用于矩阵层级
- monInfraPol (uni/infra/monifra-default)：适用于接入基础架构层级
- monEPGPol (uni/tn-common/monepg-default)：适用于租户层级

在四种监控策略中的每一种，默认策略可以被某个特定测量覆盖。例如，适用于 solar 租户（*tn-solar*）的某个监控策略可以覆盖针对 solar 租户的默认策略，而其他租户依然被默认策略监控。

下图四个对象中的每一个都包含监控目标。

图 77：默认监控策略的四个分类

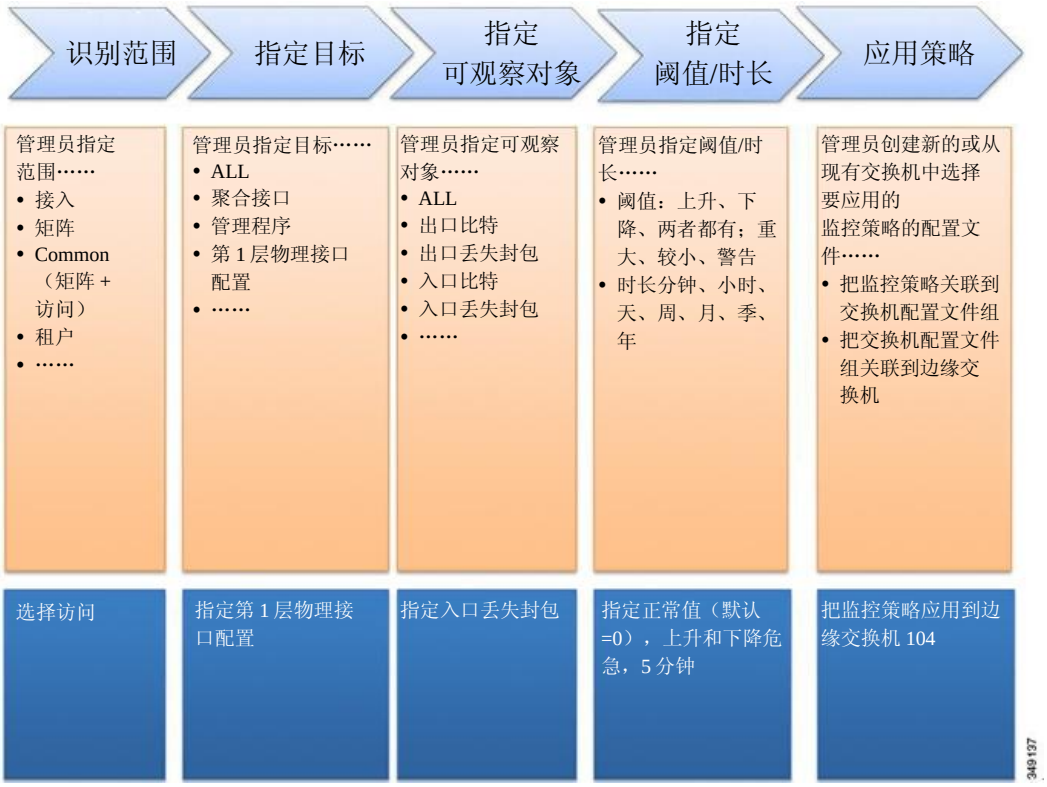


Infra 监控策略包含 monInfraTargets，矩阵监控策略包括 monFabTargets，租户监控策略包含 monEPGTargets。每个目标都代表本层级内对应分类的对象。例如，在 monInfra-default 监控策略下，

有一个目标代表面向 FEX 矩阵的端口。该目标中包含了关于如何监控这些面向 FEX 矩阵的端口的策略详细信息。上述目标下仅允许适用于每个目标的策略。请注意，并非所有可能的目标都默认自动创建。如果目标不在，管理员可以在策略下添加更多目标。

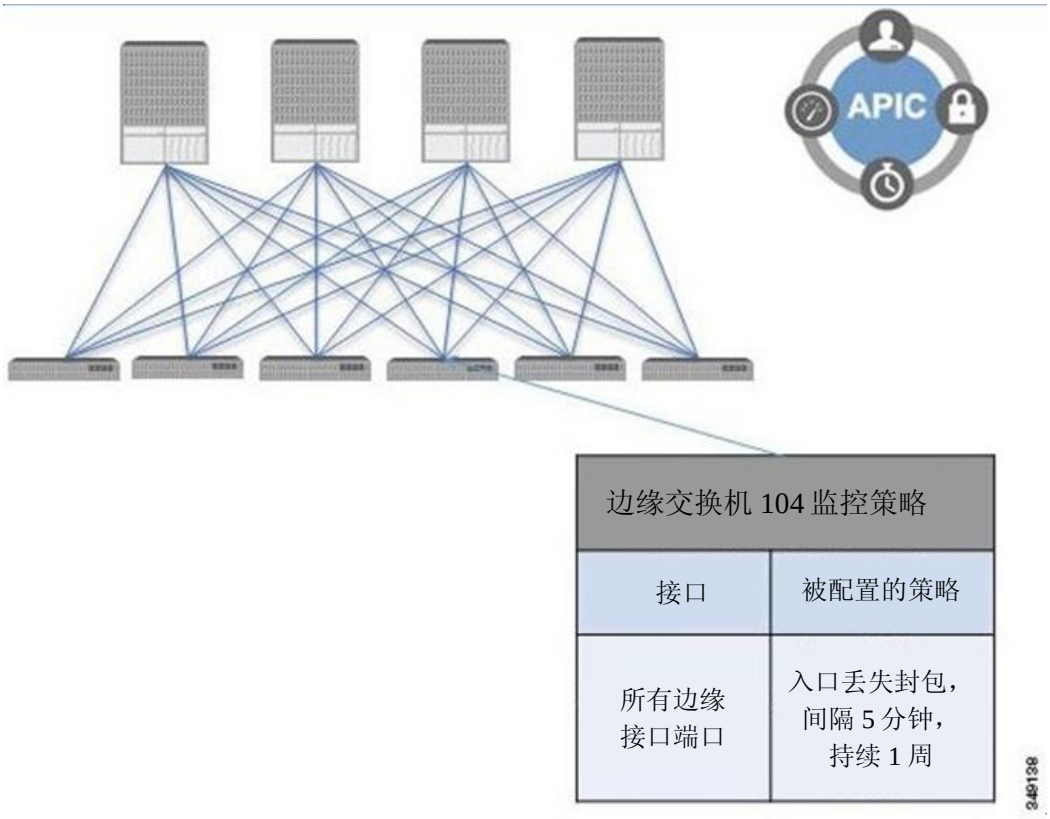
下图显示了上述过程如何在为统计信息配置矩阵监控策略时发挥作用。

图 78：配置访问监控策略的工作流程



APIC 应用监控策略的方法如下图所示：

图 79：访问监控样本策略的结果



配置监控策略可以根据其他系统操作来配置（如故障或健康得分）。监控策略的结构映射到该层级：

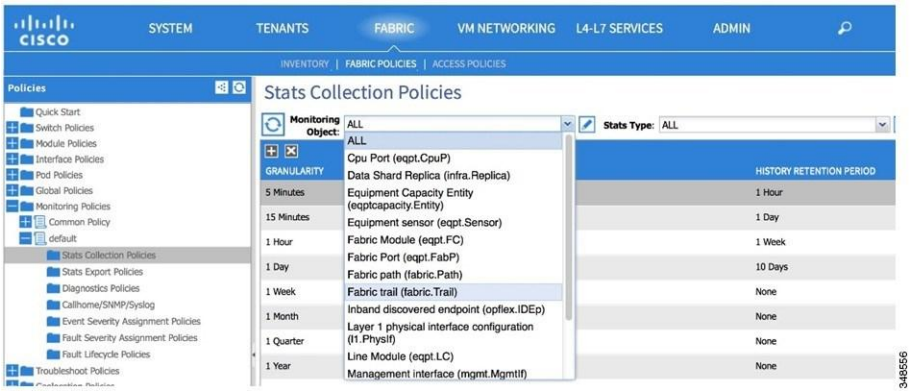
监控策略

- **统计导出**
- **收集规则**
- **监控目标**
  - **统计导出**
  - **收集规则**
  - **统计信息**
    - **收集规则**
    - **阈值规则**
    - **统计导出**

下图中的统计导出政策选项定义了将要被导出的统计信息的格式和目标地址。可以使用 FTP、HTTP 或 SCP 协议导出输出结果。支持 JSON 或 XML 格式。用户或管理员也可以选择压缩输出结果。可以在统计信息、监控目标或顶级监控策略下定义导出内容。更高层的统计导出定义具有优先权，除非有更底层的策略已经定义好。

如下图所示，监控策略通过使用选择器或关系被应用到特定的可监控的对象（如端口、卡片、EPG 和租户）或可监控对象组。

图 80：矩阵统计收集政策



监控策略规定下列内容：

- 收集统计信息并将其保留在历史中。
- 触发越限故障。
- 导出统计信息。

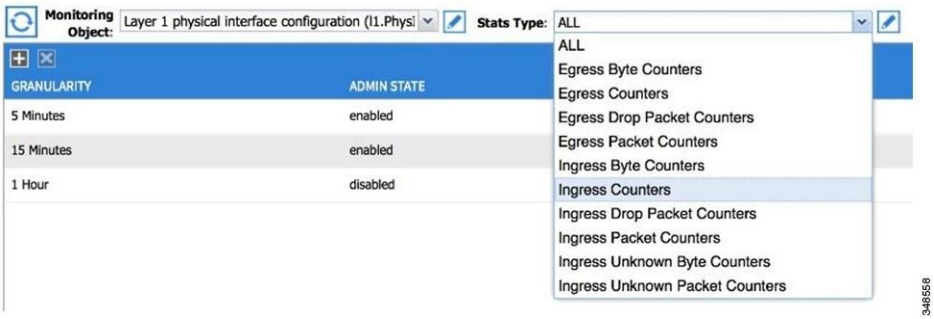
如下图所示，在每个取样间隔都定义收集规则。

图 81：统计监控间隔

它们配置如下内容：统计收集是打开还是关闭。如果定为打开的话、历史保留的时间多长。监控目标为可观察对象（如端口和 EPG）。

统计信息为统计计数器组（如入口计数器、出口计数器或丢失计数器）。

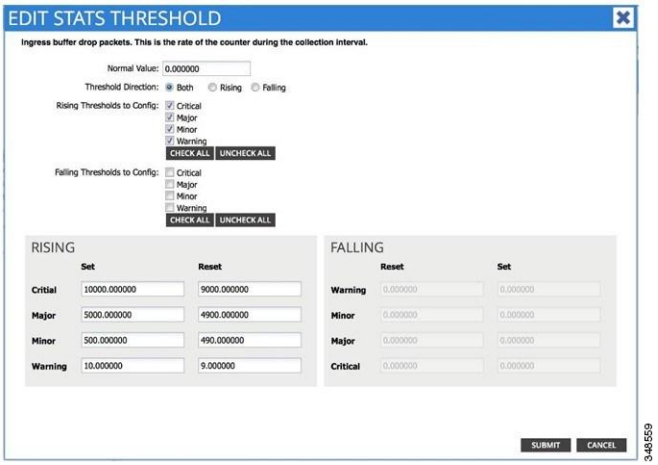
图 82：统计类型



可以在统计信息、监控对象或顶级监控策略下定义收集规则。收集规则更高层的定义具有优先权，除非有更底层的策略已经定义好。

如下图所示，阈值规则在收集规则之下定义，适用于在母收集规则中定义的对应取样间隔。

图 83：统计阈值



第 12 章

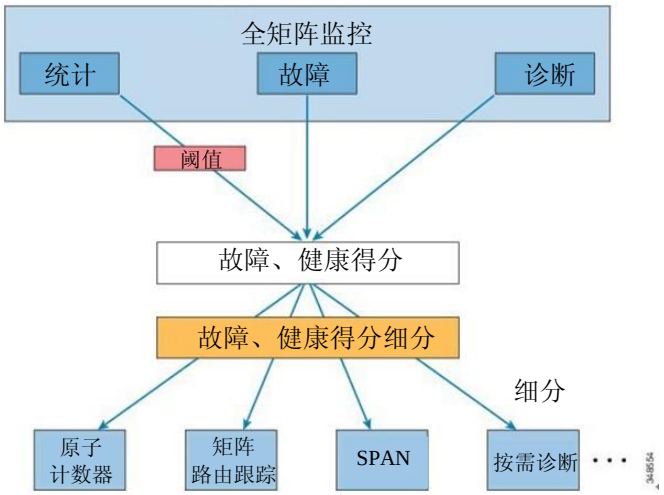
故障排除

本章包括以下部分：

故障排除

ACI 矩阵提供广泛的故障排除和监控工具，如下图所示。

图 84：故障排除



健康得分

ACI 矩阵使用策略模型把数据融合到健康得分中。可以针对多个方面（如系统、基础架构、租户、应用或服务）汇总监控得分。

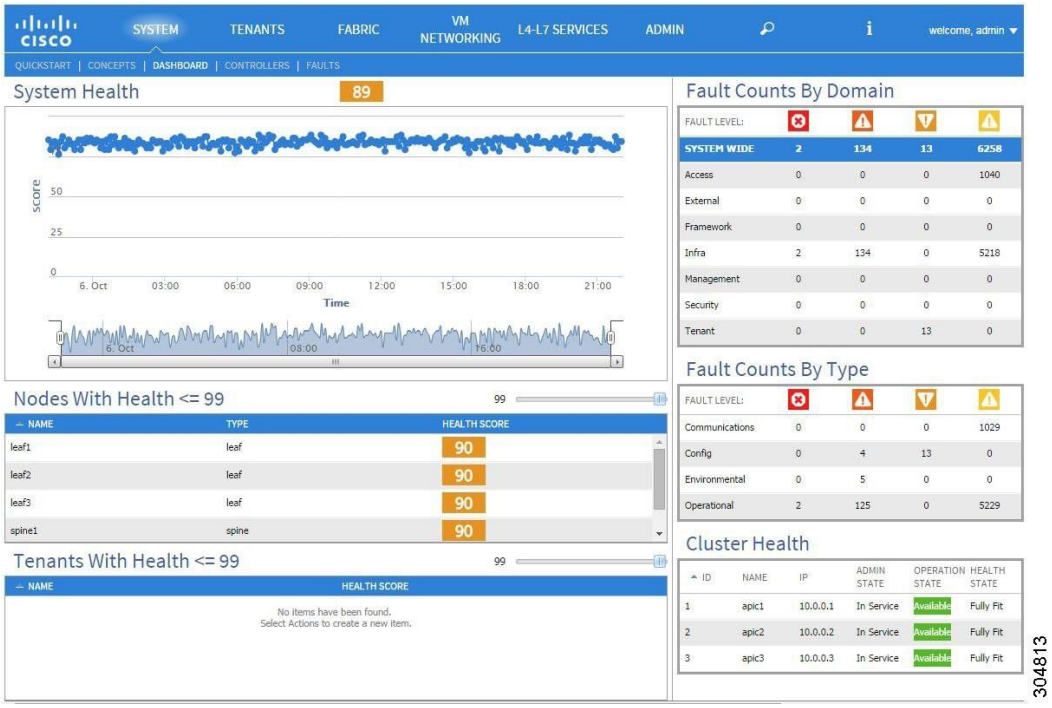
ACI 矩阵健康信息可用于查询系统的下列方面：

- 系统 — 汇总全系统的健康状况，包括 pod 健康得分、租户健康得分、系统故障计数（按照域和类型分）以及 APIC 集群健康得分。
- Pod — 汇总 pod（一组核心交换机和边缘交换机）的健康得分和全 pod 的故障计数（按照域和类型分）。
- 租户 — 汇总了租户的健康得分，包括对象（如特定于租户和全租户的故障计数（按照域和类型分））的性能数据，。
- 被管对象 — 被管对象（MO）的健康得分策略，其中被管对象包括独立的和相关的被管对象。管理员可以自定义这些策略。

系统和 Pod 健康得分

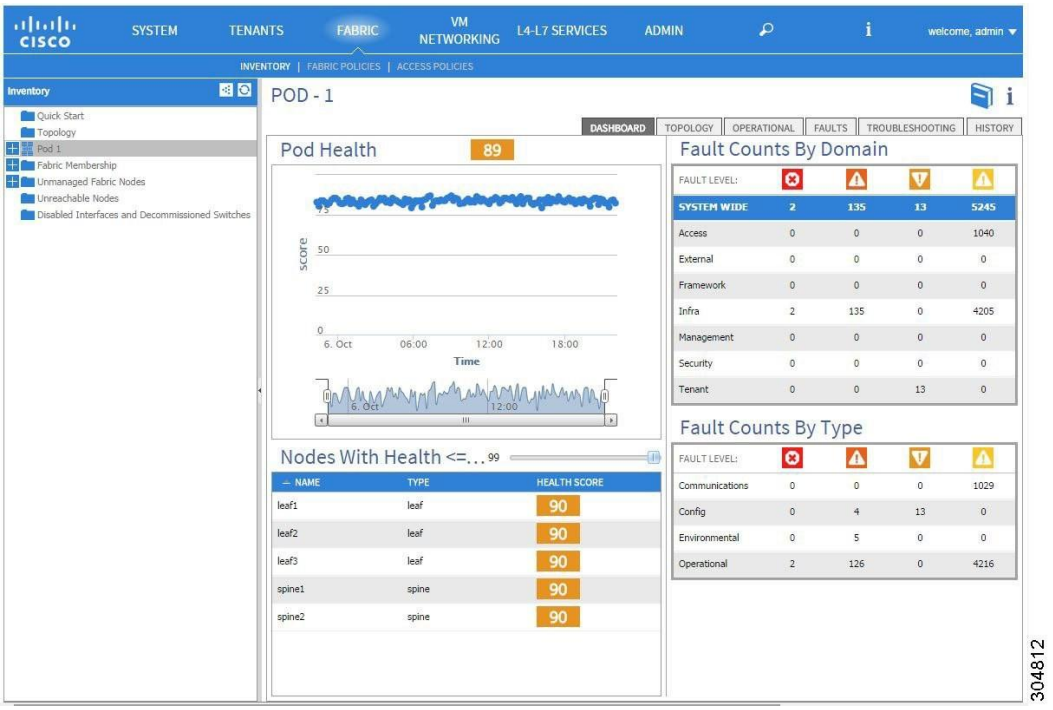
系统和 pod 健康得分基于边缘和核心交换机健康得分以及边缘交换机上学习到的端点的数量。GUI 系统仪表盘也按域的类型显示了全系统的故障计数以及每个节点管理员状态的 APIC 集群、运行状态和健康状态。

图 85：系统健康得分



pod 健康得分基于边缘交换机和核心交换机的健康得分以及边缘交换机上学习到的端点的数量。GUI 矩阵 pod 仪表盘屏幕也显示了全 pod 范围内按域和类型分类的故障计数。

图 86：Pod 健康得分



系统和 Pod 健康得分的计算方式相同。计算方法如下：边缘设备监控得分的加权平均值除以边缘交换机学习到的端点总数量，然后乘以核心系数（由核心数量及其健康得分求得）。

上述计算过程的等式如下。

图 87：系统和 Pod 健康得分计算

$$Health_{Fabric} = \frac{\sum_{i=1}^{N_{Leaf}} Health_{Leaf_i} \times Weight_{Leaf_i}}{\sum_{i=1}^{N_{Leaf}} Weight_{Leaf_i}} \times \left(1 - \left(1 - \frac{\sum_{i=1}^{N_{Spine}} Health_{Spine_i}}{N_{Spine} \times 100}\right)^{N_{Spine}}\right)$$

304814

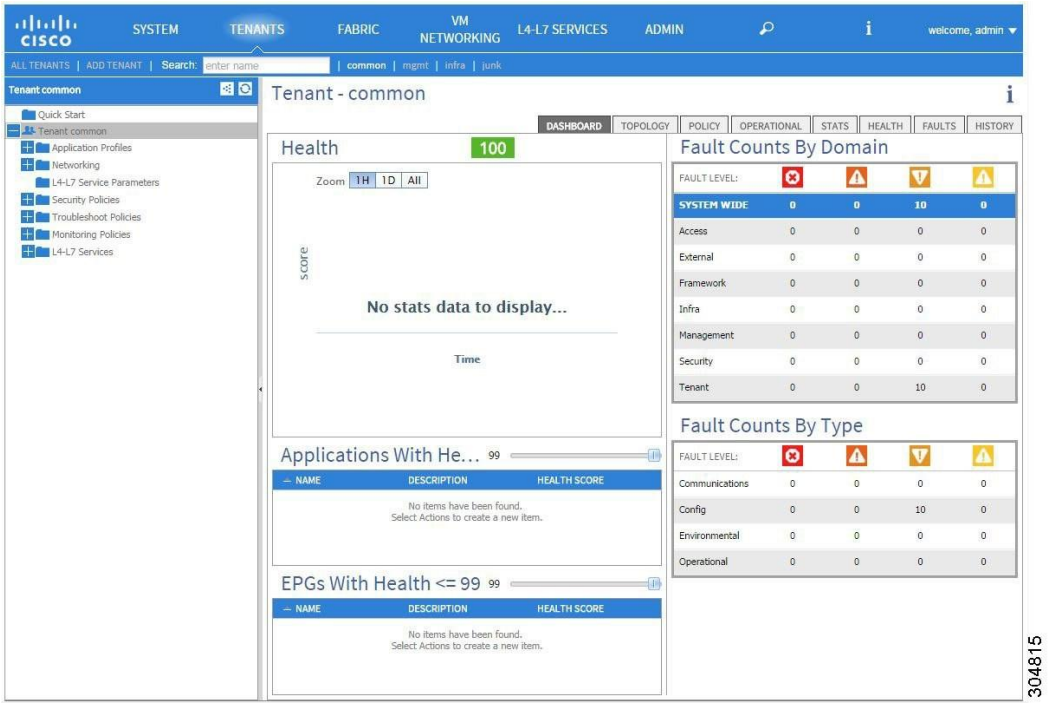
等式各部分对应的符号如下。

- $Health_{Leafi}$ 是边缘交换机的健康得分。
- $Weight_{Leafi}$ 边缘交换机上端点的数量。
- N_{Leaf} 矩阵上边缘交换机的数量。
- $Health_{Spinei}$ 核心交换机的健康得分。
- N_{Spine} 矩阵中核心交换机的数量。

租户健康得分

租户健康得分汇总了全租户内逻辑对象在它们所使用的整个基础架构的健康得分。GUI 租户仪表盘屏幕也显示了全租户范围内按域和类型分类的故障计数。

图 88：租户健康得分

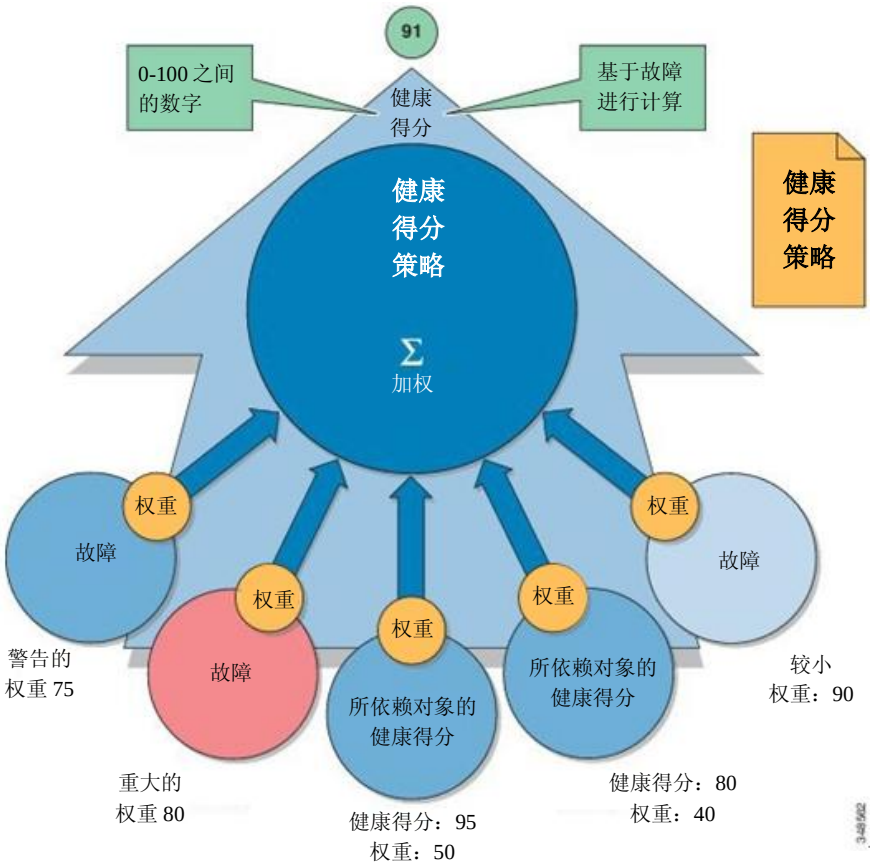


例如，某个 EPG 可能正在使用两个边缘交换机。每个边缘交换机可能包含一个已经部署的 EPG 组件。已经学习到的端点数量是一个权衡因素。每个端口学习到的端点的数量可能不同。因此，可以用下列算法得出 EPG 的健康得分：将每个 EPG 组件的健康得分相加，乘以边缘设备上学习到的端点的数量，然后除以 EPG 所用的整个边缘交换机上学习到的端点的总量。

MO 健康得分

每个被管对象（MO）都属于某个健康得分类别。默认情况下，MO 健康得分类别与其 MO 分类名称相同。

图 89：MO 健康得分



向每个健康得分类别分配一个影响层级。五个健康得分影响层级分别是：最大、高、中、低、无。例如，矩阵端口的默认影响层级为“最大”，边缘端口的默认影响层级为“高”。在计算父 MO 的健康得分时，可以通过分配健康得分影响“无”来排除某些类别的子 MO。用户可以配置对象之间的影响层级。但是，如果默认影响层级为“无”，那么管理员无法更改。

下面的因子是集中影响层级：

最大：100% 高：80% 中：50% 低：20% 无：0%

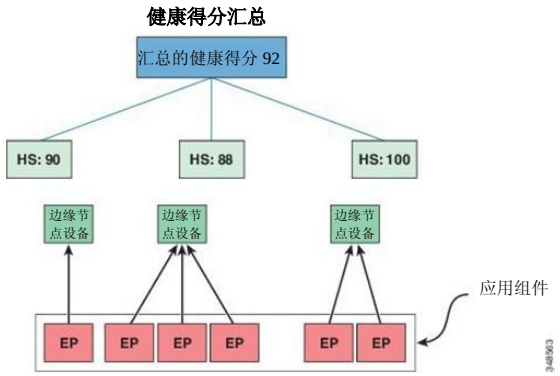
我们使用 Lp -Norm 方程计算类别健康得分。健康得分的补偿等于 100 减去健康得分。健康得分补偿代表属于某个给定类别的一组 MO 的整体健康得分补偿，是作为健康得分计算对象的 MO 的子或直系亲属。

MO 分类的健康得分类别可以通过策略进行更改。例如：某个边缘端口的默认健康得分类别是 eqpt:LeafP，矩阵接口的默认健康得分类别是 eqpt:FabP。但是，可以让包含边缘端口和矩阵端口的策略成为同一个类别（被称为“端口”）的一部分。

健康得分汇总和影响

某个应用组件的健康得分可以分布在多个边缘交换机之间，如下图所示。

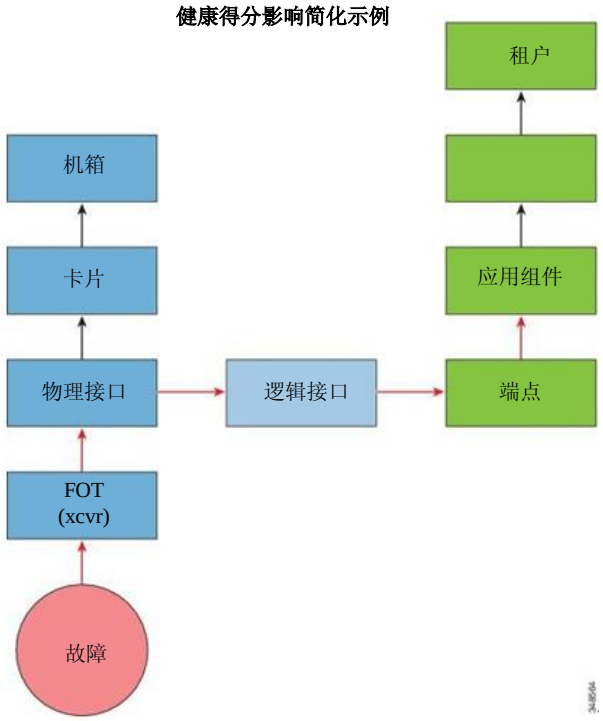
图 90：健康得分汇总



在 APIC 计算健康得分汇总。

如下图所示，硬件故障会影响应用组件的健康得分。

图 91：健康得分影响简化示例



原子计数器

原子计数器自动检测丢包和错误路由，这有助于快速调试和隔离应用连接问题。例如，管理员可以启用所有边缘交换机上的原子计数器以追踪从端点 1 到端点 2 的封包。如果任何核心设备和目标边缘节点设备之外的边缘节点设备有不为零的计数器，管理员可以向下挖掘。

在常规环境下，几乎不可能从裸机 NIC 监控通向特定 IP 地址（端点）或任何 IP 地址的流量数量。原子计数器允许管理员统计从裸机端点接收的封包数量，而不会干扰数据路径。此外，原子计数器可以监控送到端点或从端点或应用组发送的每个协议的流量。

边缘到边缘（TEP 到 TEP）原子计数器可以提供下列信息：

- 丢包计数、准许和过量封包
- 短期数据收集，如最后 30 秒，长期数据收集，如 5 分钟，15 分钟或更多
- 每个核心设备流量的分解
- 持续监控

租户原子计数器可以提供下列信息：

- 针对整个矩阵的流量、特定于应用的计数器，包括丢包、准许和过量封包
- 这些模式包括：
  - 端点到端点的 MAC 地址，或端点到端点的 IP 地址。注意，可能有多个 IP 地址与单个目标端点关联。
  - EPG 到 EPG，带有可选的细分
  - EPG 到端点
  - EPG 到*（任意）
  - 端点到外部 IP 地址

备注 当端点位于不同的租户或位于相同租户内不同的三层地址域时，不支持使用原子计数器。

多节点 SPAN

APIC 流量监控策略可以在核实的地方跨越策略，追踪每个应用组的所有成员以及它们连接的对象。如果任何成员发生移动，APIC 自动把策略推送到新的边缘节点设备。例如，如果某个端点 VMotion 移动到一个新的边缘节点设备，那么跨越配置会自动调整。

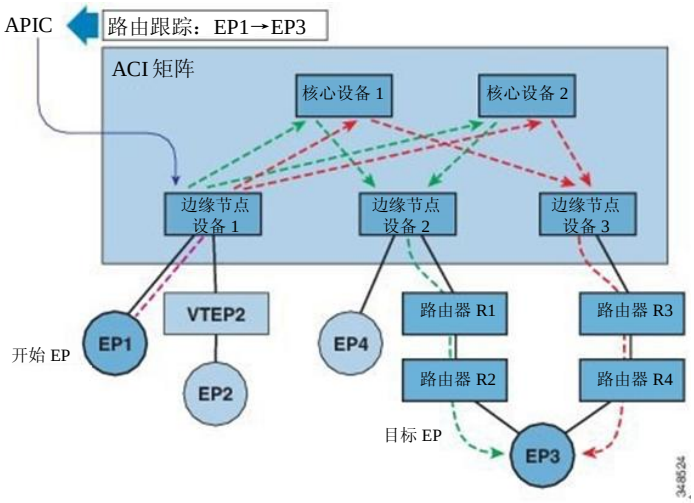
备注 SPAN（ERSPAN）的封装远程延伸时，注意租户 SPAN 使用 ERSPAN 类型 I，而矩阵 SPAN 使用 ERSPAN 类型 II。如需了解关于 ERSPAN 首标的信息，请参考

IETF Internet Draft：https://tools.ietf.org/html/draft-foschiano-erspan-00

ARP、ICMP Ping 和路由跟踪

默认网关 IP 地址的 ARP 在入口边缘交换机捕获。入口边缘交换机将 ARP 请求单播到目标地址，而目标地址发送 ARP 响应。

图 92：APIC 端点到端点路由跟踪



当中级跳出现在入口边缘路由器时，始于租户端点的路由跟踪显示了默认网关。

路由跟踪模式包括从端点到端点以及从边缘到边缘（TEP 到 TEP）。路由跟踪发现穿过矩阵的所有路径、外部端点的出口点，帮助找出是否有任何路径被堵塞。

附录 A

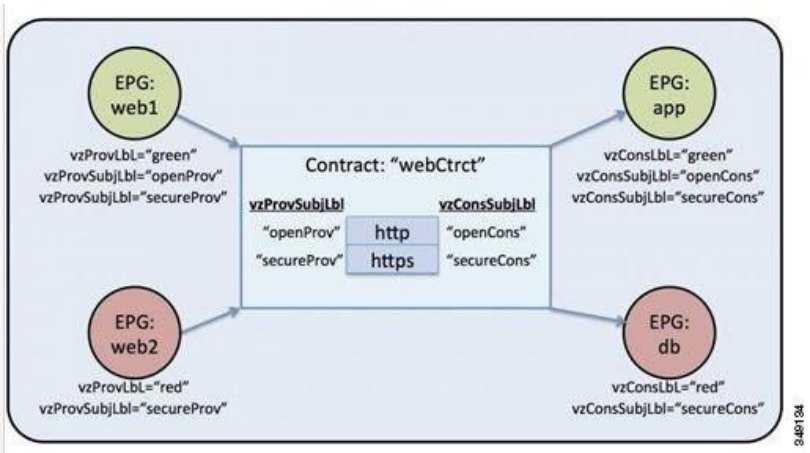
租户策略示例

本章包括以下部分：

租户策略示例概览

本附录使用 XML 术语对租户策略示例进行了描述。(http://en.wikipedia.org/wiki/XML#Key_terminology).本例展示了如何将基本的 APIC 策略模型结构翻译为 XML 代码。下图概括描述了租户策略示例。

图 93：EPG 和租户 Solar 中包含的合约



在图中，根据 webCtrct 合约和 EPG 标签，绿色的 EPG:web1 可以使用 http 和 https 与绿色的 EPG:app 通讯，红色的 EPG:web2 只能使用 https 与红色的 EPG:db 通讯。

租户策略示例 XML 代码

```
<polUni>
   <fvTenant name="solar">

      <vzFilter name="Http">
            <vzEntry name="e1" etherT="ipv4"
                  prot="tcp" dFromPort="80"
                  dToPort="80"/>
      </vzFilter>

      <vzFilter name="Https">
            <vzEntry name="e1" etherT="ipv4"
                  prot="tcp" dFromPort="443"
                  dToPort="443"/>
      </vzFilter>

      <vzBrCP name="webCtrct">
                  <vzSubj name="http" revFltPorts="true" provmatchT="All">
                     <vzRsSubjFiltAtt tnVzFilterName="Http"/>
                  <vzRsSubjGraphAtt graphName="G1" termNodeName="TProv"/>
                  <vzProvSubjLbl name="openProv"/>
                  <vzConsSubjLbl name="openCons"/>
            </vzSubj>
                  <vzSubj name="https" revFltPorts="true" provmatchT="All">
                   <vzProvSubjLbl name="secureProv"/>
                  <vzConsSubjLbl name="secureCons"/>
                     < vzRsSubjFiltAtt tnVzFilterName="Https"/>
                  <vzRsOutTermGraphAtt graphName="G2" termNodeName="TProv"/> </vzSubj>
      </vzBrCP>

      <fvCtx name="solarctx1"/>

      <fvBD name="solarBD1">
            <fvRsCtx tnFvCtxName="solarctx1" />
            <fvSubnet ip="11.22.22.20/24">
                  <fvRsBDSubnetToProfile tnL3extOutName="rout1"
                        tnRtctrlProfileName="profExport"/>
            </fvSubnet>
            <fvSubnet ip="11.22.22.211/24">
                  <fvRsBDSubnetToProfile tnL3extOutName="rout1"
                        tnRtctrlProfileName="profExport"/>
            </fvSubnet>
      </fvBD>

     <fvAp name="sap">
            <fvAEPg name="web1">
                  <fvRsBd tnFvBDName="solarBD1" />
                     <fvRsDomAtt tDn="uni/vmmp-VMware/dom-mininet" />
                     <fvRsProv tnVzBrCPName="webCtrct" matchT="All">
                        <vzProvSubjLbl name="openProv"/>
                         <vzProvSubjLbl name="secureProv"/>
                        <vzProvLbl name="green"/>
                  </fvRsProv>
            </fvAEPg>
            <fvAEPg name="web2">
                        <fvRsBd tnFvBDName="solarBD1" />
                           <fvRsDomAtt tDn="uni/vmmp-VMware/dom-mininet" />
                           <fvRsProv tnVzBrCPName="webCtrct" matchT="All">
                                 <vzProvSubjLbl name="secureProv"/>
                            <vzProvLbl name="red"/>
```

```
            </fvRsProv>
        </fvAEPg>
        <fvAEPg name="app">
            <fvRsBd tnFvBDName="solarBD1" />
                <fvRsDomAtt tDn="uni/vmmp-VMware/dom-mininet" />
              <fvRsCons tnVzBrCPName="webCtrct">
                    <vzConsSubjLbl name="openCons"/>
                     <vzConsSubjLbl name="secureCons"/>
                   <vzConsLbl name="green"/>
            </fvRsCons>
        </fvAEPg>
        <fvAEPg name="db">
            <fvRsBd tnFvBDName="solarBD1" />
                 <fvRsDomAtt tDn="uni/vmmp-VMware/dom-mininet" />
              <fvRsCons tnVzBrCPName="webCtrct">
                    <vzConsSubjLbl name="secureCons"/>
                  <vzConsLbl name="red"/>
            </fvRsCons>
        </fvAEPg>
    </fvAp>
  </fvTenant>
</polUni>
```

租户策略示例说明

本节对租户策略示例进行了详细解释。

总体策略

总体策略包含了规定每个租户策略的所有租户管理的对象。

<polUni>

第一行的启动标签<polUni>表示总体策略元素开始。该标签与策略末尾的</polUni> 对应。两者之间任何内容都是策略定义。

租户策略示例

<fvTenant> 表示租户元素的开始。

<fvTenant name="solar">

租户的所有策略都在该元素中进行了定义。本示例中租户的名称是“solar”。系统中租户的名称必须是唯一的。租户包含的主要组成部分有：过滤器、合约、外部网络、桥接域、包含服务器组（EPG）的应用配置文件。

过滤器

过滤器元素以<vzFilter> 标签开始，包含在<vzEntry> 标签中指明的元素。

过滤器

下例对“Http”和“Https”过滤器进行了定义。过滤器的第一个属性是其名称，名称属性的值是租户特有的字符串。这些名称可以在不同的租户中重复使用。在示例中，稍后在合约的主题元素中使用了这些过滤器。

```
<vzFilter name="Http">
      <vzEntry name="e1" etherT="ipv4" prot="tcp" dFromPort="80" dToPort="80"/>
</vzFilter>
<vzFilter name="Https">
      <vzEntry name="e1" etherT="ipv4" prot="tcp" dFromPort="443" dToPort="443"/>
</vzFilter>
```

本例对两个过滤器进行了定义：Http 和 Https。过滤器的第一个属性是其名称，名称属性的值是租户特有的字符串，即这些名称可以在不同的用户中重复使用。在示例中，稍后将在合约的主题元素中使用这些过滤器。

每个过滤器都有一个或多个条目，每个条目描述了一组第 4 层 TCP 或 UDP 端口编号。<vzEntry> 元素的某些可能的属性如下：

- name
- prot
- dFromPort
- dToPort
- sFromPort
- sToPort
- etherT
- ipFlags
- arpOpc
- tcpRules

在本例中，每个条目到的名称属性都已指明。名称是一个 ASCII 字符串，在过滤器内必须是唯一的，但可以在其他过滤器中重复使用。因为本例并不涉及之后的特定条目，所有只给它一个简单的名称“e1”。

下一个是 EtherType 属性 etherT。为它分配了值 ipv4 用于指明过滤器适用于 IPv4 封包。该属性还可以有其他可能的值。常见的包括 ARP、RARP，将来的版本中还有 IPv6。默认值未指明，因此为其分配一个数值非常重要。

紧随 EtherType 属性之后的是 prot 属性，被设置为 tcp 用于表明该过滤器适用于 TCP 流量。备选的协议属性包括 udp、icmp 和 unspecified （默认）。

在协议之后，目标 TCP 端口编号的分配范围在 80 至 80 之间（确切说是 TCP 端口 80），带有 dFromPort 和 dToPort 属性。如果范围的起始数字不同，那么它们会指定一系列端口编号。

在本例中，目标地址的端口编号通过属性 dFromPort 和 dToPort 指定。但是，当在合约中使用它们时，它们可以用于从 TCP 客户端到服务器的目标端口，作为返回流量的源端口。关于更多信息，请稍后参考本示例中的属性 revFltPorts 。

本质上第二个过滤器与之相同，不过针对的是端口 443.

合约中的对象通过目标可识别名 tDn 指代过滤器。tDn 名称的构成如下：uni/tn-<租户名称>/flt-<过滤器名称>。

例如，上文首个过滤的 tDn 是 uni/tn-coke/flt-Http。第二个过滤器的名称是 uni/tn-coke/flt-Https。在两种情况下，solar 来自于租户名称。

合约

合约元素带有 vzBrCP 标签，拥有一个名称属性。

```
<vzBrCP name="webCtrct">
        <vzSubj name="http" revFltPorts="true" provmatchT="All">
        <vzRsSubjFiltAtt tnVzFilterName="Http"/>
        <vzRsSubjGraphAtt graphName="G1" termNodeName="TProv"/>
        <vzProvSubjLbl name="openProv"/>
        <vzConsSubjLbl name="openCons"/>
    </vzSubj>
    <vzSubj name="https" revFltPorts="true" provmatchT="All">
        <vzProvSubjLbl name="secureProv"/>
        <vzConsSubjLbl name="secureCons"/>
        <vzRsFiltAtt tnVzFilterName="Https "/>
        <vzRsOutTermGraphAtt graphName="G2" termNodeName="TProv"/>
    </vzSubj>
</vzBrCP>
```

合约是 EPG 之间的策略元素。在生产和使用合约的 EPG 之间使用的所有过滤器都包含在合约中。合约元素带有 vzBrCP 标签，拥有一个名称属性。

关于可以在合约元素中使用的其他属性，请参考对象模型参考文档。本例带有一个名为 webCtrct 的合约。

合约包含多个主题元素，每个主题包含一组过滤器。在本例中，两个主题为 http 和 https。

合约稍后会被提供或使用它的 EPG 引用。EPG 通过下列方式通过合约的名称引用它：

uni/tn-[tenant-name]/brc-[contract-name]

tenant-name 是租户的名称，在本示例中是“solar”，contract-name 是合约的名称。对于本例，合约的 tDn 名称是 uni/tn-solar/brc-webCtrct。

主题

主题元素的开头是标签 vzSubj，带有三个属性： 名称、revFltPorts 和 matchT。名称就是主题的 ASCII 名称。

revFltPorts 是一个标记，它表明，主题过滤器内第 4 层源端口和目标端口应按照过滤器描述的规定正向（也就是从服务使用方到提供方 EPG）使用，反向时应当以相反的方式使用。在本例中，“http”主题包含“Http”过滤器，后者规定了 TCP 目标端口 80，但未指明源端口。由于 revFltPorts 标记被设置为“真”，所以策略将是 TCP **目标端口 80** 和任何从服务使用方到服务提供方的流量的源端口，策略将是 TCP 目标端口和从服务提供方到服务使用方的流量的**源端口 80**。前提是服务使用方发起 TCP 到服务提供方的连接（服务使用方是客户端，服务提供方是服务器）。

在未指明的情况下，revFltPrts 属性的默认值是“假”。

标签

匹配类型属性 provmatchT （针对提供方匹配）和 consmatchT （针对使用方匹配）决定在确定主题是否适用于规定的服务使用方和服务提供方时如何比较主题标签。可用下列匹配类型值：

- All
- AtLeastOne （默认）
- None
- ExactlyOne

在确定某个主题是否适用于提供方和使用方 EPG 之间的流量时，匹配属性决定在 EPG 中规定（或未规定）的主题标签应如何与主题中的标签进行比较。如果匹配属性值被设定为 All，那么它仅适用于下列提供方：即该提供方的主题标签 vzProvSubjLbl 匹配在主题中规定的所有 vzProvSubjLbl 标签。如果定义了两个标签，那么两者都必须在提供方中。如果提供方 EPG 拥有 10 个标签，只有主题中所有的提供方标签都存在，匹配即可确认。使用 vzConsSubjLbl 的使用方采用类似的标准。如果 matchT 属性值是 AtLeastOne，仅一个标签必须匹配。如果 matchT 属性是 None，只有当主题中没有提供方标签匹配提供方 EPG 的提供方标签时，匹配才发生，对使用方来说情况类似。

如果提供方或使用方没有任何主题标签且主题没有任何标签，那么只有 All、AtLeastOne 和 None 才发生匹配（如果你不使用标签，那么实用主题，matchT 属性无关紧要。）

在本例中没有显示的主题的可选属性为 prio ，这里指定了匹配过滤器的流量的优先级。可能的值为 gold、silver, bronze 或 unspecified（默认）。

在本例中，主题元素包含对过滤器元素、主题标签元素和图元素的引用。<vzRsSubjFiltAtt tDn=“uni/tn-coke/flt-Http”/> 是对之前定义的过滤器的引用。该元素通过 vzRsSubjFiltAtt 标签进行识别。

<vzRsSubjGraphAtt graphName=“G1” termNodeName=“TProv”/> 定义了一个终端连接。

<vzProvSubjLbl name=“openProv”/> 定义了一个服务提供方标签，名为“openProv”。标签用于判定或过滤哪个主题适用于哪个 EPG。这个标签是一个服务提供方标签，对应的服务使用方标签通过标签 vzConsSubjLbl 进行识别。这些标签与跟当前合约关联的提供方或使用方 EPG 对应的标签匹配。如果按照上文中的 matchT 标准出现了匹配，那么一个特别的主题适用于 EPG。如果没有出现匹配，则忽略该主题。

可以向某个主题添加多个服务提供方和使用方的主题标签，从而允许更加复杂的匹配标准。在本例中，每个主题的每个类型只有一个标签。但是，第一个主题上的标签不同于第二个主题，这使得可以根据对应 EPG 的标签以不同的方式处理这两个主题。主题元素内元素的顺序无关紧要。

三层地址域

三层地址域通过 fvCtx 标签识别，包含一个姓名属性。

```
<fvCtx name="solarctx1"/>
```

一个租户可以包含多个三层地址域。在本例中，租户使用一个名为“solartx1”的三层地址域。租户中的名称必须是唯一的。

三层地址域定义了第 3 层地址域。第 3 层域中所有的端点都必须拥有唯一的 IPv4 或 IPv6 地址，因为如果策略允许的话，可以直接在这些设备之间转发数据包。一个三层地址域相当于网络世界中的虚拟路由和转发（VRF）实例。

三层地址域确定了唯一的 IP 地址空间，而对应的子网在桥接域中进行了定义。一个或多个桥接域与一个三层地址域相关联。

桥接域

桥接域元素通过 fvBD 标签进行识别，带有一个名称属性。

```
<fvBD name="solarBD1">
      <fvRsCtx tnFvCtxName="solarctx1" />
      <fvSubnet ip="11.22.22.20/24">
            <fvRsBDSubnetToProfile
                  tnL3extOutName="rout1"
                  tnRtctrlProfileName="profExport" />
      </fvSubnet>
      <fvSubnet ip="11.22.23.211/24">
            <fvRsBDSubnetToProfile
                  tnL3extOutName="rout1"
                  tnRtctrlProfileName="profExport"/>
      </fvSubnet>
</fvBD>
```

在桥接域元素内定义了子网并引用了对应的第 3 层三层地址域。一个桥接域必须和一个三层地址域相连并至少有一个子网。

本例使用一个名为 “solarBD1”的桥接域。在本例中，使用 fvRsCtx 标记元素引用了“solarctx1”，为 tnFvCtxName 属性赋值“solarctx1”。名称来自于上文中定义的三层地址域。

子网包含在桥接域中，桥接域可以包含多个子网。本例对两个子网进行了定义。桥接域中使用的所有地址都必须属于子网中定义的一个地址范围中。但是，子网也可以是一个超网，它是一个非常大的子网，包含很多可能永远也无法用到的地址。指定一个涵盖所有当前、未来地址的大型子网可以简化桥接域的规范。然而，一个桥接域中不同的子网不得重叠，也不得跟与同一个三层地址域关联的其他桥接域中定义的子网重叠。子网可以跟与其他三层地址域关联的其他子网重叠。

上述子网是 11.22.22.xx/24 和 11.22.23.xx/24。但是，即使掩码规定仅使用 24，也会给予完整的 32 位地址，因为这个 IP 属性也指明了适用于那个子网的路由器的完整 IP。在第一种情况下，路由器的 IP 地址（默认网关）是 11.22.22.20，而第二个路由器的 IP 地址是 11.22.23.211。

条目 11.22.22.20/24 与下列内容等价，但形式更紧凑：

- 子网：11.22.22.00
- 子网掩码：255.255.255.0
- 默认网关：11.22.22.20

应用配置文件

配置文件的开头用 fvAp 标签标示，拥有一个名称属性。

```
<fvAp name="sap">
```

本例具有一个应用网络配置文件，名为“sap”。

配置文件是容纳 EPG 的一个容器。EPG 可以与同一个应用配置文件的其他 EPG 和其他应用配置文件的 EPG 通讯。应用配置文件只是一个用于容纳多个逻辑上相互关联的 EPG 的方便容器。它们的分类方式可以是：它们提供的应用，如 sap”；它们提供的功能，如“infrastructure”；它们在数据中心结构中的位置，如“DMZ”；或者管理员选择的任何组织原则。

应用配置文件包含的主要对象是一个服务器组（EPG）。在本例中，“sap” 应用配置文件包含 4 个 EPG：web1、web2、app 和 db。

端点和服务器组（EPG）

EPG 的开始部分是标签 fvAEPg，拥有一个名称属性。

```
<fvAEPg name="web1">
        <fvRsBd tnFvBDName="solarBD1" />
        <fvRsDomAtt tDn="uni/vmmp-VMware/dom-mininet" />
        <fvRsProv tnVzBrCPName="webCtrct" matchT ="All">
            <vzProvSubjLbl name="openProv"/>
            <vzProvSubjLbl name="secureProv"/>
            <vzProvLbl name="green"/>
        </fvRsProv>
</fvAEPg>
```

在策略模型中，EPG 是最重要的基本对象。它代表从策略角度看以同样方式对待的一系列端点。并非以单个的方式配置和管理这些端点，而是将其放在一个 EPG 中，作为一组整体管理。

EPG 对象是定义标签的位置，管理应用哪些策略以及哪些其他 EPG 可以与这个 EPG 通讯。它还包含了对与 EPG 内的端点关联的桥接域的引用，

还指明了这些端点与哪个虚拟机管理器（VMM）域关联。VMM 允许两个 VM 服务器之间的即时虚拟机移动性，且不会造成应用宕机时间。

示例中的首个 EPG 名为“web1”。EPG 内的 fvRsBd 规定了与其有关联的桥接域。桥接域用 tnFxBDName 属性的值识别。EPG 与上文“桥接域” 一节中指明的“solarBD1”桥接域关联。系统使用对桥接域的捆绑理解对 EPG 中的端点而言，何为默认网关的正确地址。这并不意味着端点都位于同一个子网，或者它们只能通过桥接进行通讯。端点的封包是被桥接还是被路由，取决于源端点将封包发送到默认网关或是所需的最终目标地址。如果源端点将封包发送到默认网关，封包会被路由。

EPG 所用的 VMM 域采用 fvRsDomAtt 标签进行识别。该元素引用了在另一个地方定义的 VMM 域对象。VMM 域对象采用 tDn 名称属性进行识别。本例表明，只有一个名为“uni/vmmp-VMware/dom-mininet”的 VMM 域。

“web1”EPG 中的下一个元素规定了该 EPG 提供了哪个合约并采用 fvRsProv 进行识别。如果“web1”要提供多个合约，那么会有多个 fvRsProv 元素。类似地，如果它要使用一个或多个合约，那么也会有 fvRsCons 元素。

fvRsProv 元素有一个必要的属性，是正在被提供的合约的名称。“web1” 提供之前定义的合约“webCtrct”，名为 tDn=“uni/tn-coke/brc-webCtrct”。

下一个属性是 matchT 属性，对于匹配服务提供方或使用方的标签而言，它的语义与在合约中针对主题标签的行为一样（它的值可以是 All、AtLeastOne 或 None）。该标准适用于服务提供方的标签，与对应的服务使用方标签类似。标签匹配意味着服务使用方和提供方可以通讯，前提是两者之间的合约允许这样做。换句话说，合约必须允许通讯，而按照服务提供方规定的匹配标准服务使用方和提供方的标签必须匹配。

服务使用方没有对应的匹配标准。所用的匹配类型总是由服务提供方决定。

在服务提供方元素 fvRsProv 中，管理员需要指明将要使用的标签。有两种标签，服务提供方标签和服务提供方主题标签。服务提供方标签 vzProvLbl 用于匹配其他 EPG 中的服务使用方标签，这些 EPG 使用之前描述过的 matchT 标准。服务提供方主题标签 vzProvSubjLbl 用于匹配合约中规定的主题标签。标签的唯一属性是其名称属性。

在“web1” EPG 中，两个服务提供方主题标签，openProv 和 secureProv，被指明用于匹配“webCtrct”合约的“http” 和“https” 的主题。一个服务提供方标签“green” 被指明带有一个匹配标准 All，该标准与“App” EPG 中同样的标签匹配。

示例中的下一个 EPG，“web2”，非常类似于“web1” ，只是只有一个 vzProvSubjLbl，且标签本身不同。

第三个 EPG 名为“app”，定义如下：

```
<fvAEPg name="app">
        <fvRsBd tnFvBDName="solarBD1" />
        <fvRsDomAtt tDn="uni/vmmp-VMware/dom-mininet" />
        <fvRsCons tnVzBrCPName="webCtrct">
                <vzConsSubjLbl name="openCons"/>
                <vzConsSubjLbl name="secureCons"/>
                <vzConsLbl name="green"/>
        </fvRsCons>
</fvAEPg>
```

第一部分几乎与“web1” EPG 相同。主要的不同是 EPG 是“webCtrct” 的使用方，带有相应的服务提供方标签和服务提供方主题标签。句法几乎相同，只是在标签中“Prov” 被“Cons” 替换。FvRsCons 元素中没有匹配属性，因为提供方指定了把提供方与使用方标签匹配的匹配类型。

在最后一个 EPG 中，“db” 非常类似于“app” EPG，因为前者只是一个单纯的服务使用方。

在本例中，EPG 是单个合约的使用方或提供方，通常情况下，EPG 可以同时是多个合约的提供方和多个合约的使用方。

闭合

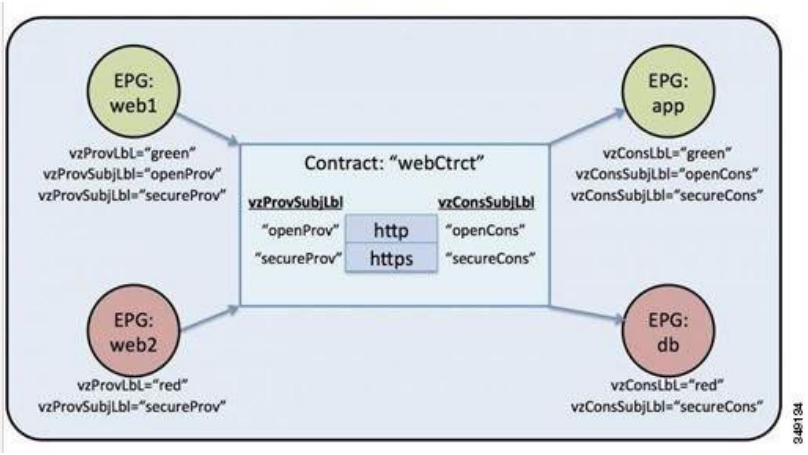
```
        </fvAp>
    </fvTenant>
</polUni>
```

最后几行构成了完整的策略。

示例租户策略的作用

下图显示了合约控制服务器组（EPG）通讯的过程。

图 94：标签和合约决定 EPG 与 EPG 之间的通讯



四个 EPG 名为 EPG:web1、EPG:web2、EPG:app 和 EPG:db。EPG:web1 和 EPG:web2 提供了一个名为 webCtrct 的合约。EPG:app 和 EPG:db 使用同一个合约。

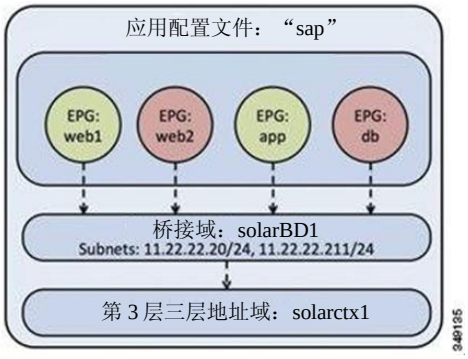
EPG:web1 只能与 EPG:app 通讯，EPG:web2 只能与 EPG:db 通讯。这种交互通过提供方和使用方标签“green” 和“red”控制。

EPG:web1 与 EPG:app 通讯时使用 webCtrct 合约。EPG:app 可发起与 EPG:web1 之间的连接，因为前者使用 EPG:web1 提供的合约。

EPG:web1 和 EPG:app 可用于通讯是 http 和 https，因为 EPG:web1 拥有提供方主题标签“openProv”，而 http 主题也有这一标签。EPG:web1 拥有提供方主题标签“secureProv”，与主题 https 相同。类似地，EPG:app 拥有主题 http 和 https 拥有的主题标签“openCons” 和“secureCons”。

EPG:web2 与 EPG:db 通讯时，两者只能使用 https 主题，因为只有 https 主题带有提供方和使用方主题标签。EPG:db 可以发起与 EPG:web2 之间的 TCP 连接，因为 EPG:db 使用 EPG:web2 提供的合约。

图 95：桥接域、子网和第 3 层三层地址域



示例策略以下列方式指明了 EPG、应用配置文件、桥接域和第 3 层三层地址域之间的关系：EPGs EPG:web1、EPG:web2、EPG:app 和 EPG:db 都是名为“sap”的应用配置文件的成员。这些 EPG 也与桥接域“solarBD1”关联。solarBD1 拥有两个子网 11.22.22.XX/24 和 11.22.23.XX/24。四个 EPG 中的端点必须位于两个子网的范围内。两个子网中默认网关的 IP 地址是 11.22.22.20 和 11.22.23.211。solarBD1 桥接域与“solarctx1” 第 3 层三层地址域连接。策略所有的详细信息都包含在了名为“solar”的租户中。

示例租户策略的作用

附录 B

标签匹配

本章包括以下部分：

标签匹配

标签匹配用于确定合约的哪些主题与合约的给定提供方或使用方一同使用，它们用于确定哪些使用方和提供方可以通讯。

匹配类型或算法由 matchT 属性确定，可以取下列值：

- All
- AtLeastOne （默认）
- None
- ExactlyOne

查看互相匹配的提供方标签 vzProvLbl 和使用方标签 vzConsLbl 时，matchT 由提供方 EPG 决定。

在拥有主题的 EPG 中查看相互匹配的提供方或使用方主题标签 vzProvSubjLbl、vzConsSubjLbl 时，matchT 由主题决定。

下表给出了所有 EPG 提供方和使用方匹配类型及其结果的简单示例。在该表中，[ ]条目表示没有标签。

matchT	vzProvLbl	vzConsLbl	结果
All	LabelX, LabelY	LabelX, LabelY	匹配
All	LabelX, LabelY	LabelX, LabelZ	不匹配
All	LabelX, LabelY	LabelX	不匹配
All	LabelX	LabelX, LabelY	匹配

标签匹配

matchT	vzProvLbl	vzConsLbl	结果
All	[ ]	LabelX	不匹配
All	LabelX	[ ]	不匹配
All	[ ]	[ ]	不匹配
AtLeastOne	LabelX, LabelY	LabelX	匹配
AtLeastOne	LabelX, LabelY	LabelZ	不匹配
AtLeastOne	LabelX	[ ]	不匹配
AtLeastOne	[ ]	LabelX	不匹配
AtLeastOne	[ ]	[ ]	匹配
None	LabelX	LabelY	匹配
None	LabelX	LabelX	不匹配
None	LabelX, LabelY	LabelY	不匹配
None	LabelX	LabelX, LabelY	不匹配
None	[ ]	LabelX	匹配
None	LabelX	[ ]	匹配
None	[ ]	[ ]	匹配
ExactlyOne	LabelX	LabelX	匹配
ExactlyOne	LabelX, LabelY	LabelX, LabelY	不匹配
ExactlyOne	LabelX, LabelZ	LabelX, LabelY	匹配
ExactlyOne	LabelX	LabelY	不匹配
ExactlyOne	[ ]	LabelX	不匹配
ExactlyOne	LabelX	[ ]	不匹配
ExactlyOne	[ ]	[ ]	匹配

相同的逻辑也适用于主题标签。合约中的主题标签将位于第二栏，而 EPG 主题标签将位于第三栏。

附录 C

接入策略示例

本章包括以下部分：

应用于多个交换机的单端口通道配置

样本 XML 策略在边缘交换机 17 上创建了一个端口通道，在边缘交换机 18 上创建了第二个端口通道，在边缘交换机 20 上创建了第三个端口通道。在每个边缘交换机上，相同的接口都将是端口通道的一部分（接口 1/10 到 1/15，1/20 到 1/25）。所有这些端口通道都具备相同的配置。

```
<infraInfra dn="uni/infra">

    <infraNodeP name=”test">
            <infraLeafS name="leafs" type="range">
                <infraNodeBlk name="nblk” from_=”17" to_=”18” />
                <infraNodeBlk name="nblk” from_=”20" to_=”20” />
        </infraLeafS>
            <infraRsAccPortP tDn="uni/infra/accportprof-test"/>
</infraNodeP>

    <infraAccPortP name="test">
            <infraHPortS name="pselc" type="range">
              <infraPortBlk name="blk1” fromCard="1"
                    toCard="1" fromPort="10" toPort=”15” />
              <infraPortBlk name="blk2" fromCard="1" toCard="1"
                    fromPort=”20" toPort=”25” />
              <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-bndlgrp"/>
        </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
```

```
                    <infraAccBndlGrp name="bndlgrp" lagT="link">
                        <infraRsHIfPol tnFabricHIfPolName=“default"/>
                      <infraRsCdpIfPol tnCdpIfPolName=”default”/>
                       <infraRsLacpPol tnLacpLagPolName=”default"/>
                </infraAccBndlGrp>
            </infraFuncP>

</infraInfra>
```

应用于多个交换机的双端口通道配置

样本 XML 策略在边缘交换机 17 上创建了两个端口通道，在边缘交换机 18 上创建了第二个端口通道，在边缘交换机 20 上创建了第三个端口通道。在每个边缘交换机上，相同的接口都将是端口通道的一部分（对于端口通道 1，接口 1/10 到 1/15；对于端口通道 2，1/20 到 1/25）。策略使用两个交换机组，因为每个交换机组可能仅包含一组连续的交换机 ID。所有这些端口通道都具备相同的配置。

备注 即使端口通道配置相同，示例使用了两个不同的接口策略组。每个接口策略组都代表交换机上的一个端口通道。与给定接口策略组关联的所有接口都是相同端口通道的一部分。

```
<infraInfra dn="uni/infra">

    <infraNodeP name=”test">
            <infraLeafS name="leafs" type="range">
              <infraNodeBlk name="nblk”
                    from_=”17" to_=”18”/>
              <infraNodeBlk name="nblk”
                    from_=”20" to_=”20”/>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-test1"/>
        <infraRsAccPortP tDn="uni/infra/accportprof-test2"/>
    </infraNodeP>

    <infraAccPortP name="test1">
        <infraHPortS name="pselc" type="range">
            <infraPortBlk name="blk1”
                fromCard="1" toCard="1"
                fromPort="10" toPort=”15”/>
            <infraRsAccBaseGrp
                tDn="uni/infra/funcprof/accbundle-bndlgrp1"/>
         </infraHPortS>
    </infraAccPortP>

    <infraAccPortP name="test2">
            <infraHPortS name="pselc" type="range">
              <infraPortBlk name="blk1”
                    fromCard="1" toCard="1"
                    fromPort=“20" toPort=”25”/>
              <infraRsAccBaseGrp
        tDn="uni/infra/funcprof/accbundle-bndlgrp2" /> </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
            <infraAccBndlGrp name="bndlgrp1" lagT="link">
```

```
                    <infraRsHIfPol tnFabricHIfPolName=“default"/>
                  <infraRsCdpIfPol tnCdpIfPolName=”default”/>
                   <infraRsLacpPol tnLacpLagPolName=”default"/>
            </infraAccBndlGrp>

                <infraAccBndlGrp name="bndlgrp2" lagT="link">
                   <infraRsHIfPol tnFabricHIfPolName=“default"/> <infraRsCdpIfPol
                   tnCdpIfPolName=”default”/>
                          <infraRsLacpPol tnLacpLagPolName=”default"/>
                </infraAccBndlGrp>
        </infraFuncP>

</infraInfra>
```

跨两个交换机的单虚拟端口通道

跨两个交换机创建虚拟端口通道的两个步骤如下：

- 创建一个 fabricExplicitGEp：该策略指明了组成一对后形成虚拟端口通道的边缘交换机。
- 使用基础架构选择器指明接口配置。

APIC 执行 fabricExplicitGEp 的几种验证，验证失败后会提示故障。某个边缘交换机仅能与另一个边缘交换机配对。APIC 拒绝违反该规则的任何配置。创建 fabricExplicitGEp 时，管理员必须提供将要配对的两个边缘交换机的 ID。APIC 拒绝违反该规则的任何配置。创建 fabricExplicitGEp 时，两个交换机必须都处于良好的工作状态。如果一个交换机没有满足上述条件，APIC 接受配置，但会提示故障。两个交换机必须都是边缘交换机。如果一个或两个交换机 ID 对应某个核心交换机，APIC 接受配置，但会提示故障。

```
<fabricProtPol pairT="explicit">
<fabricExplicitGEp name="tG" id="2">
   <fabricNodePEp id=”18”/>
   <fabricNodePEp id=”25"/>
          </fabricExplicitGEp>
    </fabricProtPol>
```

两个交换机选定端口组上的一个虚拟端口通道

该策略在边缘交换机 18 和 25 上创建了一个虚拟端口通道，使用边缘交换机 18 上的接口 1/10 到 1/15 以及边缘交换机 25 上的接口 1/20 到 1/25。

```
<infraInfra dn="uni/infra">

        <infraNodeP name=”test1">
            <infraLeafS name="leafs" type="range">
                <infraNodeBlk name="nblk”
                      from_=”18" to_=”18”/>
            </infraLeafS>
                  <infraRsAccPortP tDn="uni/infra/accportprof-test1"/>
        </infraNodeP>

        <infraNodeP name=”test2">
            <infraLeafS name="leafs" type="range">
```

```
                <infraNodeBlk name="nblk”
                    from_=” 25" to_=” 25” />
            </infraLeafS>
                <infraRsAccPortP tDn="uni/infra/accportprof-test2"/>
        </infraNodeP>

        <infraAccPortP name="test1">
            <infraHPortS name="pselc" type="range">
                <infraPortBlk name="blk1”
                    fromCard="1" toCard="1"
                    fromPort="10" toPort=” 15” />
                <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-bndlgrp" />
            </infraHPortS>
        </infraAccPortP>

        <infraAccPortP name="test2">
            <infraHPortS name="pselc" type="range">
                <infraPortBlk name="blk1”
                    fromCard="1" toCard="1"
                    fromPort=“20" toPort=” 25” />
            <infraRsAccBaseGrp
                tDn="uni/infra/funcprof/accbundle-bndlgrp" />
            </infraHPortS>
        </infraAccPortP>

        <infraFuncP>
            <infraAccBndlGrp name="bndlgrp" lagT=” node">
                <infraRsHIfPol tnFabricHIfPolName=“default"/>
                <infraRsCdpIfPol tnCdpIfPolName=” default” />
                <infraRsLacpPol tnLacpLagPolName=” default"/>
            </infraAccBndlGrp>
        </infraFuncP>

</infraInfra>
```

设置接口速度

该策略设置了一系列接口的端口速度。

```
<infraInfra dn="uni/infra">

        <infraNodeP name=” test1">
            <infraLeafS name="leafs" type="range">
                <infraNodeBlk name="nblk” from_=” 18" to_=” 18” />
            </infraLeafS>
            <infraRsAccPortP tDn="uni/infra/accportprof-test1"/>
        </infraNodeP>

        <infraNodeP name=” test2">
            <infraLeafS name="leafs" type="range">
                <infraNodeBlk name="nblk” from_=” 25" to_=” 25” />
            </infraLeafS>
            <infraRsAccPortP tDn="uni/infra/accportprof-test2"/>
        </infraNodeP>

        <infraAccPortP name="test1">
            <infraHPortS name="pselc" type="range">
                <infraPortBlk name="blk1”
                    fromCard="1" toCard="1"
                    fromPort="10" toPort=” 15” />
```

```
                <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-bndlgrp" />
            </infraHPortS>
        </infraAccPortP>

        <infraAccPortP name="test2">
            <infraHPortS name="pselc" type="range">
                <infraPortBlk name="blk1”
                    fromCard="1" toCard="1"
                    fromPort=“20" toPort=”25” />
                <infraRsAccBaseGrp
                    tDn="uni/infra/funcprof/accbundle-bndlgrp" />
            </infraHPortS>
        </infraAccPortP>

        <infraFuncP>
            <infraAccBndlGrp name="bndlgrp" lagT=” node">
                <infraRsHIfPol tnFabricHIfPolName=“default"/>
                <infraRsCdpIfPol tnCdpIfPolName=” default” />
                <infraRsLacpPol tnLacpLagPolName=” default"/>
            </infraAccBndlGrp>
        </infraFuncP>

</infraInfra>
```

设置接口速度

附录 D

租户第 3 层外部网络策略示例

本章包括以下部分：

租户外部网络策略示例

下列 XML 代码举例说明了租户第 3 层外部网络策略。

```
<polUni>

 <fvTenant name='t0'>
      <fvCtx name="o1">
                <fvRsOspfCtxPol tnOspfCtxPolName="ospfCtxPol"/>
      </fvCtx>
      <fvCtx name="o2"> </fvCtx>

      <fvBD name="bd1">
            <fvRsBDToOut tnL3extOutName='T0-o1-L3OUT-1'/>
            <fvSubnet ip='10.16.1.1/24' scope='public'/>
            <fvRsCtx tnFvCtxName="o1"/>
      </fvBD>

      <fvAp name="AP1">
            <fvAEPg name="bd1-epg1">
                  <fvRsCons tnVzBrCPName="vzBrCP-1">
                  </fvRsCons>
                  <fvRsProv tnVzBrCPName="vzBrCP-1">
                  </fvRsProv>
                  <fvSubnet ip='10.16.2.1/24' scope='private'/>
                  <fvSubnet ip='10.16.3.1/24' scope='private'/>
                  <fvRsBd tnFvBDName="bd1"/>
                  <fvRsDomAtt tDn="uni/phys-physDomP"/>
                  <fvRsPathAtt
                        tDn="topology/pod-1/paths-101/pathep-[eth1/40]"
                        encap='vlan-100' mode='regular'
                        instrImedcy='immediate' />
            </fvAEPg>

            <fvAEPg name="bd1-epg2">
                  <fvRsCons tnVzBrCPName="vzBrCP-1">
```

租户外部网络策略示例

```
                    </fvRsCons>
                    <fvRsProv tnVzBrCPName="vzBrCP-1">
                    </fvRsProv>
                    <fvSubnet ip='10.16.4.1/24' scope='private'/>
                    <fvSubnet ip='10.16.5.1/24' scope='private'/>
            <fvRsBd tnFvBDName="bd1"/>
            <fvRsDomAtt tDn="uni/phys-physDomP"/>
            <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/41]"
                    encap='vlan-200' mode='regular'
                    instrImedcy='immediate'/>
    </fvAEPg>
</fvAp>

<l3extOut name="T0-o1-L3OUT-1">

    <l3extRsEctx tnFvCtxName="o1"/>
    <ospfExtP areaId='60'/>
    <l3extInstP name="l3extInstP-1">
        <fvRsCons tnVzBrCPName="vzBrCP-1">
        </fvRsCons>
        <fvRsProv tnVzBrCPName="vzBrCP-1">
        </fvRsProv>
        <l3extSubnet ip="192.5.1.0/24" />
        <l3extSubnet ip="192.5.2.0/24" />
        <l3extSubnet ip="192.6.0.0/16" />
        <l3extSubnet ip="199.0.0.0/8" />
    </l3extInstP>

    <l3extLNodeP name="l3extLNodeP-1">
        <l3extRsNodeL3OutAtt tDn=“topology/pod-1/node-101" rtrId="10.17.1.1">
            <ipRouteP ip="10.16.101.1/32">
                    <ipNexthopP nhAddr="10.17.1.99"/>
            </ipRouteP>
            <ipRouteP ip="10.16.102.1/32">
                    <ipNexthopP nhAddr="10.17.1.99"/>
            </ipRouteP>
            <ipRouteP ip="10.17.1.3/32">
                    <ipNexthopP nhAddr="10.11.2.2"/>
            </ipRouteP>
        </l3extRsNodeL3OutAtt >

        <l3extLIfP name='l3extLIfP-1'>
            <l3extRsPathL3OutAtt tDn=“topology/pod-1/paths-101/pathep-[eth1/25]"
                    encap='vlan-1001'
                    ifInstT='sub-interface'
                    addr="10.11.2.1/24"
                    mtu="1500"/>
            <ospfIfP>
                    <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
            </ospfIfP>
        </l3extLIfP>
    </l3extLNodeP>
</l3extOut>

<ospfIfPol name="ospfIfPol" />
<ospfCtxPol name="ospfCtxPol" />

<vzFilter name="vzFilter-in-1">
        <vzEntry name="vzEntry-in-1"/>
</vzFilter>
<vzFilter name="vzFilter-out-1">
        <vzEntry name="vzEntry-out-1"/>
```

```
  </vzFilter>

  <vzBrCP name="vzBrCP-1">
       <vzSubj name="vzSubj-1">
            <vzInTerm>
                        <vzRsFiltAtt tnVzFilterName="vzFilter-in-1"/>
            </vzInTerm>
            <vzOutTerm>
                  <vzRsFiltAtt tnVzFilterName="vzFilter-out-1"/> </vzOutTerm>
       </vzSubj>
        </vzBrCP>
  </fvTenant>
</polUni>
```

租户外部网络策略示例

附录 E

DHCP 中继策略示例

本章包括以下部分：

第 2 层和第 3 层 DHCP 中继样本策略

该样本策略举例说明了服务使用方租户 L3extOut DHCP 中继配置。

```
<polUni>
      <!-- 服务使用方租户 2 -->
  <fvTenant dn="uni/tn-tenant1"
    name="tenant1">
     <fvCtx name="dhcp"/>

            <!-- DHCP 租户桥接域 -->
    <fvBD name="cons2">
                  <fvRsBDToOut tnL3extOutName='L3OUT'/>
      <fvRsCtx tnFvCtxName="dhcp" />
                  <fvSubnet ip="20.20.20.1/24"/>
                  <dhcpLbl name="DhcpRelayP" owner="tenant"/>
    </fvBD>
            <!-- L3Out EPG DHCP -->
    <l3extOut name="L3OUT">
              <l3extRsEctx tnFvCtxName="dhcp"/>
              <l3extInstP name="l3extInstP-1">
                        <!-- 通往 L3out、发送流量的授权路由 -->
                        <l3extSubnet ip="100.100.100.0/24" />
              </l3extInstP>
              <l3extLNodeP name="l3extLNodeP-pc">
              <!-- 节点上 VRF 外部回路接口 -->
                    <l3extRsNodeL3OutAtt
                          tDn="topology/pod-1/node-1018"
                          rtrId="10.10.10.1" />
                    <l3extLIfP name='l3extLIfP-pc'>
                          <l3extRsPathL3OutAtt
                                tDn="topology/pod-1/paths-1018/pathep-[eth1/7]"
                                encap='vlan-900'
                                ifInstT='sub-interface'
                                addr="100.100.100.50/24"
                                mtu="1500"/>
```

```
                    </l3extLIfP>
                </l3extLNodeP>
            </l3extOut>
        <!-- 静态 DHCP 租户配置 -->
        <fvAp name="cons2">
            <fvAEPg name="APP">
              <fvRsBd tnFvBDName="cons2"/>
                    <fvRsDomAtt tDn="uni/phys-mininet"/>
                       <fvRsPathAtt tDn="topology/pod-1/paths-1017/pathep-[eth1/3]"
                              encap="vlan-1000" instrImedcy='immediate'
                              mode='native'/>
            </fvAEPg>
    </fvAp>
            <!-- DHCP 服务器配置 -->
            <dhcpRelayP
                    name="DhcpRelayP"
                    owner="tenant" mode="visible">
                    <dhcpRsProv
                          tDn="uni/tn-tenant1/out-L3OUT/instP-l3extInstP-1"
                          addr="100.100.100.1"/>
            </dhcpRelayP>
  </fvTenant>
</polUni>
```

该样本策略举例说明了服务使用方租户 L2extOut DHCP 中继配置。

```
<fvTenant dn="uni/tn-dhcpl2Out"
    name="dhcpl2Out"> <fvCtx
    name="dhcpl2Out"/>
                    <!-- 桥接域 -->

    <fvBD name="provBD">
      <fvRsCtx tnFvCtxName="dhcpl2Out" />
                                  <fvSubnet ip="100.100.100.50/24" scope="shared"/>
    </fvBD>

    <!-- 服务使用方桥接域 -->
    <fvBD name="cons2">
      <fvRsCtx tnFvCtxName="dhcpl2Out" />
                                      <fvSubnet ip="20.20.20.1/24"/>
                                     <dhcpLbl name="DhcpRelayP" owner="tenant"/>
    </fvBD>

                    <vzFilter name='t0f0' >
                    <vzEntry name='t0f0e9'></vzEntry>
                    </vzFilter>

                    <vzBrCP name="webCtrct" scope="global">
                    <vzSubj name="app">
                                        <vzRsSubjFiltAtt tnVzFilterName="t0f0"/>
                          </vzSubj>
                    </vzBrCP>

          <l2extOut name="l2Out">
                  <l2extLNodeP name='l2ext'>
                <l2extLIfP name='l2LifP'>
                   <l2extRsPathL2OutAtt tDn="topology/pod-1/paths-1018/pathep-[eth1/7]"/>
                    </l2extLIfP>
                        </l2extLNodeP>
                <l2extInstP name='l2inst'>
                                            <fvRsProv tnVzBrCPName="webCtrct"/>
```

```
                </l2extInstP>
            <l2extRsEBd tnFvBDName="provBD" encap='vlan-900'/>
        </l2extOut>

        <fvAp name="cons2">
            <fvAEPg name="APP">
              <fvRsBd tnFvBDName="cons2" />
                  <fvRsDomAtt tDn="uni/phys-mininet" />
                      <fvRsBd tnFvBDName="SolarBD2" />
                      <fvRsPathAtt tDn="topology/pod-1/paths-1018/pathep-[eth1/48]"
encap="vlan-1000" instrImedcy='immediate' mode='native'/>
                </fvAEPg>
        </fvAp>
                <dhcpRelayP name="DhcpRelayP" owner="tenant" mode="visible">
                <dhcpRsProv tDn="uni/tn-dhcpl2Out/l2out-l2Out/instP-l2inst" addr="100.100.100.1"/>
            </dhcpRelayP>
    </fvTenant>
```

第 2 层和第 3 层 DHCP 中继样本策略

附录 F

DNS 策略示例

本章包括以下部分：

DNS 策略示例

dnsProfile 的样本 XML：

```
<!-- /api/policymgr/mo/.xml -->
<polUni>
<fabricInst>
<dnsProfile name="default">
        <dnsProv addr="172.21.157.5" preferred="yes"/>
        <dnsDomain name="insieme.local" isDefault="yes"/>
        <dnsRsProfileToEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>
</dnsProfile>
</fabricInst>
</polUni>
```

dns 标签的样本 xml：

```
<!-- /api/policymgr/mo/.xml -->
<polUni>
<fvTenant name=’t1’ >
   <fvCtx name=’ctx0’ >
      <dnsLbl name=’default’ />
   </fvCtx>
</fvTenant>
</polUni>
```

DNS 策略示例

附录 G

RBAC 规则样本

RBAC 规则样本

下面 JSON 样本文件中的 RBAC 规则支持针对 VMM 域资源的跨租户访问和租户访问。使用方需要的资源是 uni/tn-prov1/brc-webCtrct 和 vmmp-Vmware/dom-Datacenter。

下面两条 RBAC 规则支持使用方租户在下面 JSON 文件中进行提供方邮递员查询。

```
<aaaRbacEp>
    <aaaRbacRule objectDn="uni/vmmp-VMware/dom-Datacenter" domain="cons1"/>
    <aaaRbacRule objectDn="uni/tn-prov1/brc-webCtrct" domain="cons1"/>
</aaaRbacEp>
```

下面的 JSON 文件包含这两个 RBAC 规则：

```
{"id":"ac62a200-9210-f53b-7114-a8f4cffb9a36","name":"SharedContracts","timestamp":1398806919868,"requests":
[{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"2dfc75cc-431e-e136-622c-a577ce7622d8", "name":"login as prov1",
"description":"",
"url":"http://http://solar.local:8000/api/aaaLogin.json",
"method":"POST",
"headers":"", "data":
"{\"aaaUser\":{\"attributes\":{\"name\": \"prov1\", \"pwd\": \"secret!\"}}}",
"dataMode":"raw","timestamp":0,"version":2,"time":1398807562828},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"56e46db0-77ea-743f-a64e-c5f7b1f59807", "name":"Root login",
"description":"",
"url":"http://http://solar.local:8000/api/aaaLogin.json",
"method":"POST",
"headers":"", "data":
"{\"aaaUser\":{\"attributes\":{\"name\": \"admin\", \"pwd\": \"secret!\"}}}",
"dataMode":"raw","timestamp":0,"responses":[],"version":2},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"804893f1-0915-6d35-169d-3af0eb3e64ec", "name":"consumer tenant
only",
"description":"",
"url":"http://http://solar.local:8000/api/policymgr/mo/uni/tn-cons1.xml",
"method":"POST",
"headers":"", "data":
"<fvTenant name=\"cons1\">
    <aaaDomainRef name=\"cons1\"/>\n
```

RBAC 规则样本

```
</fvTenant>\n",
"dataMode":"raw","timestamp":0,"version":2,"time":1398968007487},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"85802d50-8089-bf8b-4481-f149bec258c8", "name":"login as cons1",
"description":"",
"url":"http://solar.local:8000/api/aaaLogin.json",
"method":"POST",
"headers":"", "data":
"{\"aaaUser\":{\"attributes\":{\"name\": \"cons1\", \"pwd\": \"secret!\"}}}",
"dataMode":"raw","timestamp":0,"version":2,"time":1398807575531},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"a2739d92-5f9d-f16c-8894-0f64b6f967a3",
"name":"consumer",
"description":"",
"url":"http://solar.local:8000/api/policymgr/mo/uni/tn-cons1.xml", "method":"POST","headers":"","data":
"<fvTenant name=\"cons1\" status=\"modified\">\n
      <fvCtx name=\"cons1\"/>\n
        <!-- bridge domain -->\n
            <fvBD name=\"cons1\">\n
   <fvRsCtx tnFvCtxName=\"cons1\" />\n
          <fvSubnet ip=\"10.0.2.128/24\" scope='shared'/>\n </fvBD>\n
   \n <!-- DNS Shared Service Contract Interface-->\n
   <vzCPIf name=\"consIf\">\n
              <vzRsIf tDn=\"uni/tn-prov1/brc-webCtrct\" >\n </vzRsIf>\n
   </vzCPIf>\n \n
  <fvAp name=\"cons1\">\n
       <fvAEPg name=\"APP\">\n
              <fvRsBd tnFvBDName=\"cons1\" />\n
     <fvRsNodeAtt tDn=\"topology/pod-1/node-101\" encap=\"vlan-4000\" instrImedcy=\"immediate\" mode=\"regular\"/>\n
      <fvRsDomAtt tDn=\"uni/vmmp-VMware/dom-Datacenter\"/>\n
      <fvRsConsIf tnVzCPIfName=\"consIf\"/>\n
   </fvAEPg>\n
      </fvAp>\n
</fvTenant>\n",
"dataMode":"raw","timestamp":0,"version":2,"time":1398818639692},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"c0bd866d-600a-4f45-46ec-6986398cbf78", "name":"provider tenant only",
"description":"",
"url":"http://solar.local:8000/api/policymgr/mo/uni/tn-prov1.xml",
"method":"POST",
"headers":"", "data":
"<fvTenant name=\"prov1\"><aaaDomainRef name=\"prov1\"/> \n
</fvTenant>\n",
"dataMode":"raw","timestamp":0,"version":2,"time":1398818137518},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"d433a213-e95d-646d-895e-3a9e2e2b7ba3", "name":"create RbacRule",
"description":"",
"url":"http://solar.local:8000/api/policymgr/mo/uni.xml",
"method":"POST",
"headers":"", "data":
"<aaaRbacEp>\n
      <aaaRbacRule objectDn=\"uni/vmmp-VMware/dom-Datacenter\" domain=\"cons1\"/>\n
      <aaaRbacRule objectDn=\"uni/tn-prov1/brc-webCtrct\" domain=\"cons1\"/>\n
</aaaRbacEp>\n",
"dataMode":"raw","timestamp":0,"version":2,"time":1414195420515},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"d5c5d580-a11a-7c61-34ac-cbdac249157f",
"name":"provider",
"description":"",
"url":"http://solar.local:8000/api/policymgr/mo/uni/tn-prov1.xml",
     "method":"POST",
     "headers":"", "data":
     "<fvTenant name=\"prov1\" status=\"modified\">\n
```

RBAC 规则样本


```
        <fvCtx name=\"prov1\"/>\n
   \n <!-- bridge domain -->\n
          <fvBD name=\"prov1\">\n
            <fvRsCtx tnFvCtxName=\"prov1\" />\n
     </fvBD>\n \n
     <vzFilter name='t0f0' >\n
           <vzEntry etherT='ip' dToPort='10' prot='6' name='t0f0e9' dFromPort='10'>
         </vzEntry>\n
     </vzFilter>\n \n
     <vzFilter name='t0f1'>\n
           <vzEntry etherT='ip' dToPort='209' prot='6' name='t0f1e8' dFromPort='109'>
         </vzEntry>\n
   </vzFilter>\n \n
       <vzBrCP name=\"webCtrct\" scope=\"global\">\n
         <vzSubj name=\"app\">\n
           <vzRsSubjFiltAtt tnVzFilterName=\"t0f0\"/>\n
     <vzRsSubjFiltAtt tnVzFilterName=\"t0f1\"/>\n
             </vzSubj>\n
       </vzBrCP>\n \n
  <fvAp name=\"prov1AP\">\n
         <fvAEPg name=\"Web\">\n
           <fvRsBd tnFvBDName=\"prov1\" />\n
                    <fvRsNodeAtt tDn=\"topology/pod-1/node-17\" encap=\"vlan-4000\"
instrImedcy=\"immediate\" mode=\"regular\"/>\n
                    <fvRsProv tnVzBrCPName=\"webCtrct\"/>\n
      <fvRsDomAtt tDn=\"uni/vmmp-VMware/dom-Datacenter\"/>\n
                    <fvSubnet ip=\"10.0.1.128/24\" scope='shared'/>\n </fvAEPg>\n
  </fvAp>\n
</fvTenant>\n",
"dataMode":"raw","timestamp":0,"version":2,"time":1398818660457},

{"collectionId":"ac62a200-9210-f53b-7114-a8f4cffb9a36","id":"e8866493-2188-8893-8e0c-4ca0903b18b8",
"name":"add user prov1",
"description":"",
"url":"http://solar.local:8000/api/policymgr/mo/uni/userext.xml",
"method":"POST",
"headers":"", "data":
"<aaaUserEp>\n
         <aaaUser name=\"prov1\" pwd=\"secret!\">
           <aaaUserDomain name=\"prov1\">
                 <aaaUserRole name=\"tenant-admin\" privType=\"writePriv\"/>
                 <aaaUserRole name=\"vmm-admin\" privType=\"writePriv\"/>
    </aaaUserDomain>
    </aaaUser>\n
          <aaaUser name=\"cons1\" pwd=\"secret!\">
      <aaaUserDomain name=\"cons1\">
      <aaaUserRole name=\"tenant-admin\" privType=\"writePriv\"/>
      <aaaUserRole name=\"vmm-admin\" privType=\"writePriv\"/>
      </aaaUserDomain>
    </aaaUser>\n
          <aaaDomain name=\"prov1\"/>\n
      <aaaDomain name=\"cons1\"/>\n
</aaaUserEp>\n",
"dataMode":"raw","timestamp":0,"version":2,"time":1398820966635}]}
```

RBAC 规则样本

附录 H

术语表

本章包括以下部分：

术语表

以应用为中心的基础设施（ACI）—ACI 是一个全局数据中心基础架构，支持集中自动化，带有策略驱动的应用配置文件。

应用策略基础架构控制器（APIC）—管理可扩展多租户矩阵的 ACI 基础架构的关键组件。APIC 控制器包括了一个复制的同步群集控制器，提供针对多租户结构的管理、策略编程、应用部署和健康监控。**服务提供方**—使用某个服务的服务器组（EPG）。**三层地址域**—定义一个三层地址域。**合约**—指明 EPG 之间的通讯是何种类型，如何发生。**可识别名称（DN）**—描述 MO 并在 MIT 中定位 MO 的独特名称。

服务器组（EPG）—EPG 是一个管理对象，它是一个包含一系列端点的指定逻辑单元。端点是直接或间接与网络相连的设备。它们有地址（标识）、位置、属性（例如版本或补丁级别），可以是实物，也可以是虚拟的。端点的示例包括服务器、虚拟机、存储、或是互联网上的客户端。

过滤器—一个 TCP/IP 首标字段，如第 3 层协议类型、第 4 层端口等，用在合约中定义 EPG 之间的入站或出站通讯。

标签—仅具备一个属性的被管对象，一个名称。标签能够分类，哪些对象可以或是不可以与其他对象通信。**被管对象（MO）**—对矩阵资源的一种抽象。

管理信息树（MIT）—包含矩阵所有被管对象的一个具有层级结构的信息树。

外部网络—一个被管对象，定义通往矩阵之外的网络的连接。

策略—被指明的实体，包含了控制系统行为某些方面的类属规范。例如，一个第 3 层外部网络策略会包含 BGP 协议，用于在把矩阵连接至一个外部第 3 层网络时启用 BGP 路由功能。

配置文件—被指明的实体，包含用于执行一个或多个策略实例的必要的配置详细信息。例如，路由策略的一个交换机节点配置文件可能包含所有特定于交换机的、执行 BGP 所需的配置详情。

术语表

服务提供方—一个提供服务的 EPG。**主题**—包含在合约中指明可通讯哪些信息以及如何通讯这些信息的被管对象。

目标 DN（tDn）—一个明示的引用，规定了一个源 MO 和目标 MO 特定实例之间的关系。目标实例通过一个在关系源（Rs）MO 中明确设置的目标 DN 属性（tDn）识别。

租户—一个应用策略的逻辑容器，使得管理员能够执行基于域的接入控制。租户代表一个从策略角度来看的独立单元，但并不代表一个专用网络。租户可以代表服务供应者设置中的客户，企业设置中的组织机构或域，或只是一组方便分组的策略。租户包含的主要元素是过滤器、合约、外层、桥接域和包含 EPG 的应用配置文件。