シスコアプリケーションセントリックインフラストラクチャの基本

初版：2014 年 08 月 01 日

最終更新：2016 年 04 月 20 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次

はじめに

この前書きは、次の項で構成されています。

対象読者

このガイドは、次の 1 つ以上に責任を持つ、専門知識を備えたデータセンター管理者を主な対象にしています。

- 仮想マシンのインストールと管理
- サーバ管理
- スイッチおよびネットワークの管理

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
italic	イタリック体の文字は、ユーザが値を入力する引数です。

表記法	説明
[x]	省略可能な要素（キーワードまたは引数）は、角カッコで囲んで示しています。
[x | y]	いずれか 1 つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。
{x | y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波カッコで囲み、縦棒で区切って示しています。
[x {y | z}]	角カッコまたは波カッコが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角カッコ内の波カッコと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
`variable`	ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

例では、次の表記法を使用しています。

表記法	説明
`screen フォント`	スイッチが表示する端末セッションおよび情報は、screen フォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
イタリック体の *screen* フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
< >	パスワードのように出力されない文字は、山カッコ（< >）で囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

このマニュアルでは、次の表記法を使用しています。

（注） 「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

警告 安全上の重要な注意事項

「危険」の意味です。人身事故を予防するための注意事項が記述されています。装置の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止策をとるよう努めてください。警告の各国語版を参照するには、各注意事項の番号と、装置に付属の「Translation Safety Warnings」の番号を照らし合せてください

これらの注意事項を保管しておいてください。

関連資料

シスコ アプリケーション セントリック インフラストラクチャ（ACI）のマニュアル

ACI のマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

シスコ アプリケーション セントリック インフラストラクチャ（ACI）シミュレータのマニュアル

Cisco ACI Simulator のマニュアルは、次の URL から入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html。

Cisco Nexus 9000 シリーズ スイッチのマニュアル

Cisco Nexus 9000 シリーズ スイッチのマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html

Cisco Application Virtual Switch のマニュアル

Cisco Application Virtual Switch（AVS）のマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html

シスコ アプリケーション セントリック インフラストラクチャ（ACI）と OpenStack の統合に関するマニュアル

Cisco ACI と OpenStack の統合に関するマニュアルは、次の URL から入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、apic-docfeedback@cisco.com までご連絡ください。ご協力をよろしくお願いいたします。

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手、Cisco Bug Search Tool（BST）の使用、サービス要求の送信、追加情報の収集の詳細については、『*What's New in Cisco Product Documentation*』を参照してください。このドキュメントは、次から入手できます。 http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』では、シスコの新規および改訂版の技術マニュアルの一覧を、RSS フィードとして購読できます。また、リーダー アプリケーションを使用して、コンテンツをデスクトップに配信することもできます。RSS フィードは無料のサービスです。

第 1 章

新機能および変更された機能に関する情報

この章の内容は、次のとおりです。

新機能および変更された機能に関する情報

次の表は、この最新リリースに関するマニュアルでの主な変更点の概要を示したものです。ただし、このリリースに関するガイドの変更点や新機能の中には、一部、この表に記載されていないものもあります。

新機能および変更された機能に関する情報

表 *1* : ***Cisco APIC 1.3(x)*** およびスイッチ ***11.3(x)*** リリースの新機能と変更された動作

機能	説明	参照先
-- マイクロセグメンテーション	マイクロセグメンテーションは、仮想マシン属性、IP アドレス、MAC アドレスに基づいて、複数の EPG からのエンドポイントを、1 つのマイクロセグメント化 EPG に関連付けます。仮想マシン属性に含まれる属性は、次のとおりです。VNic ドメイン名、VM 識別子、VM 名、ハイパーバイザ識別子、VMM ドメイン、データセンター、オペレーティング システム、カスタム属性。ベア メタルおよび VM エンドポイントのための EPG 内分離と組み合わせると、マイクロセグメンテーションを行うことにより、ポリシーに基づいて自動化された完全なエンドポイント分離をアプリケーション層内で実行することができます。	--マイクロセグメンテーション, （30 ページ）
-- バグ修正	タグ付き EPG のトピックの更新	-- ネイティブ 802.1p とタグ付き EPG, （44 ページ）

表 *2* : ***Cisco APIC*** リリース ***1.2(2x)*** の新機能と変更された動作

機能	説明	参照先
-- EPG 内拒否	EPG 内拒否ポリシーにより、仮想エンドポイントまたは物理エンドポイントが完全に分離されます。完全分離モードで稼働している EPG 内のエンドポイント間の通信は許可されません。	--EPG 内エンドポイントの分離, （30 ページ）
-- データ プレーン ポリシング	データ プレーン ポリシング（DPP）を使用して、ACI ファブリック アクセス インターフェイスの帯域幅使用量を管理します。	--データ プレーン ポリシング, （112 ページ）

機能	説明	参照先
-- BGP 属性の設定	ルート制御コンテキストは照合対象を指定し、スコープは設定対象を指定します。	--ルートのインポートとエクスポート、ルート集約、ルートコミュニティの一致, （103 ページ）
-- BGP および OSPF 集約	ルート集約ポリシーにより、ボーダー リーフ スイッチとそのネイバー リーフ スイッチの間でルートを効率的に共有することができます。	
-- EIGRPv6	EIGRPv6 のサポートが有効になりました。	--EIGRP プロトコルのサポート, （132 ページ）
-- DSCP マーキング	以前は DSCP マーキングを L3Out でしか設定できませんでしたが、コントラクト、サブジェクト、イン ターム、アウト タームでも設定可能になりました。	--802.1p サービス クラスの保持, （76 ページ）
-- 双方向フォワーディング検出	双方向フォワーディング検出（BFD）を使用して、ピアリング ルータの接続をサポートするように設定された ACI ファブリック境界リーフ スイッチ間の転送パスのサブセカンド障害検出時間を提供します。	--双方向フォワーディング検出, （112 ページ）
-- 管理インターフェイスの IPv6 サポート	すべての ACI ファブリックと APIC インターフェイスに対する無制限の IPv6 サポート。IPv4 または IPv6 またはデュアル スタック構成がサポートされます。管理インターフェイスで IPv4 アドレスのみを許可する要件は適用されなくなりました。	--IPv6 のサポート, （113 ページ）
-- BGP ダイナミック ネイバー、ルート ダンプニング、重み属性、remove-private-as -- OSPF のネーム ルックアップ、プレフィクス抑制、タイプ 7 変換	BGP および OSPF オプションのサポートの拡張。	--プロトコルによるルート ピアリング, （125 ページ）

新機能および変更された機能に関する情報

機能	説明	参照先
-- 設定ゾーン	設定ゾーンは ACI ファブリックを複数のゾーンに分割します。これらのゾーンは、別々のタイミングで設定を変更をして更新することができます。これにより、障害のある設定がファブリック全体に導入されるリスクが限定され、トラフィックが中断したり、さらにはファブリックがダウンしたりする可能性が抑えられます。	-- 設定ゾーン，（81 ページ）
-- アップリンク障害検出のためのポート トラッキング ポリシー	リーフ スイッチから 1 つ以上のスパイン スイッチへのアップリンク障害が検出されると、ファブリック リンク ステート トラッキングにより、リンクがダウンしたことがアクセス ポート接続デバイスに通知されます。	--アップリンク障害検出のためのポート トラッキング ポリシー，（75 ページ）

表 *3*：*Cisco APIC* リリース *1.2(1x)* の新機能と変更された動作

機能	説明	参照先
-- IP ベースの EPG	IP ベースの EPG は、最長プレフィックス照合（LPM）分類ではサポートできない、多数の EPG が必要となる設定に適しています。	--エンドポイント グループ，（27 ページ）
-- EPG 下のパブリック サブネットのサポート	共有サービスを提供する EPG は、サブネットを（ブリッジ ドメイン下ではなく）その EPG 下で設定する必要があります。そのスコープは advertised externally および shared between VRFs に設定する必要があります。	--ブリッジ ドメインとサブネット，（24 ページ）

機能	説明	参照先
-- 共有レイヤ 3 Out	共有レイヤ 3 Out 設定は、外部ネットワークへのルーテッド接続を共有サービスとして提供します。l3extInstP EPG は、外部ネットワークへのルーテッド接続を提供します。これは、任意のテナント（user、common、infra、または mgmt.）の共有サービスとしてプロビジョニングできます。	--共有レイヤ 3 Out, （108 ページ）
-- バグ修正	サブネット ルート エクスポートおよびルート インポートの設定オプションの説明を改善。	--ルートのインポートとエクスポート、ルート集約、ルート コミュニティの一致, （103 ページ）
-- 課金用レイヤ 3 ルート インターフェイスの統計情報	APIC は、共有サービスとしての外部ネットワークへのルーテッド接続用に設定されたポート（`l3extInstP` EPG）からバイト カウントとパケット カウントでの課金統計情報を収集するように設定できます。	-- 共有サービスとしての外部ネットワークへのルーテッド接続の課金と統計情報, （140 ページ）
-- 最大プレフィクス数の設定	BGP l3extOut 接続のテナント ネットワーキング プロトコル ポリシーは、最大プレフィックス制限を使用して設定できます。これにより、ピアから受信されるルート プレフィックスの数を監視し、制限することができます。	--外部ネットワークへのルーテッド接続のためのレイヤ 3 Out, （101 ページ）
-- L3Out スケールの入力ポリシー適用	入力ベースのポリシーの適用により、レイヤ 3 Out トラフィックのポリシー適用を出力方向および入力方向に定義することが可能になります。ダイレクト サーバ リターン（DSR）および属性 EPG には入力ベースのポリシー適用が必要です。	
-- 重み付けのあるスタティック ルート	ACI ファブリック内のスタティック ルートのプリファレンスは、コスト拡張コミュニティを使用して MP-BGP で送信されます。	--スタティック ルートのプリファレンス, （102 ページ）

新機能および変更された機能に関する情報

機能	説明	参照先
-- IPv4 の共通パーベイシブ ゲートウェイと IPv4 のセカンダリ IP アドレス	ブリッジドメインごとに IPv4 共通ゲートウェイを使用して複数の ACI ファブリックを設定できます。	--共通パーベイシブ ゲートウェイ, （94 ページ）
-- ファブリック セキュア モード	ファブリック セキュア モードでは、ファブリック機器に物理的にアクセスできるユーザが管理者による手動の認証を受けずにファブリックにスイッチまたは APIC コントローラを追加するのを防止できます。	--ファブリック セキュア モード, （82 ページ）
-- CoS（802.1p）	ACI ファブリックは、ファブリック内で 802.1p サービス クラス（CoS）を維持することができます。ファブリック グローバル QoS ポリシーの dot1p-preserve オプションを有効にすることで、ACI ファブリックを入力し中継するパケット 802.1p 値の保持が保証されます。	--802.1p サービス クラスの保持, （76 ページ）

表 *4* : ***Cisco APIC*** リリース ***1.1(2x)*** の新機能と変更された動作

機能	説明	参照先
-- APICのコンフィギュレーション ファイルの AES 暗号化	ACIファブリックは、コンフィギュレーションエクスポート/インポート ファイル内のセキュア プロパティの AES 暗号化をサポートします。	-- 設定ファイルの暗号化, （183 ページ） --セキュア プロパティ, （261 ページ）
-- アップデートおよびバグ修正	ラベルの一致の付録の更新。 保持ポリシー ガイドラインの追加。 共通テナント内の L3extOut を介したテナントブリッジドメインパブリック サブネットのアドバタイジングのサポートに関する更新。	--ラベルの一致, （225 ページ） --エンドポイントの保持, （49 ページ） --外部ルータへのブリッジドインターフェイス, （100 ページ）

表 *5* : ***Cisco APIC*** リリース ***1.1(1x)*** の新機能と変更された動作

機能	説明	参照先
-- IPv6 サポート	ACI ファブリックは、テナント アドレッシング、コントラクト、共有サービス、ルーティング、レイヤ 4 ～レイヤ 7 のサービス、トラブルシューティングに関して IPv6 をサポートします。ACI ファブリック インターフェイスは、リンク ローカル、グローバル ユニキャスト、マルチキャスト IPv6 アドレスで設定できます。	--IPv6 のサポート, （113 ページ）
-- トランジット ルーティング	ACI ファブリックは、必須の EIGRP、eBGP、OSPF プロトコル サポートを含む中継ルーティングをサポートします。それにより、境界ルータが他のルーティング ドメインとの双方向再配布を実行することが可能になります。	--ACI トランジット ルーティング, （119 ページ）
-- EIGRP	ACI ファブリックは、IPv4 に対してのみ L3 外部での EIGRP プロトコルをサポートします。	--EIGRP プロトコルのサポート, （132 ページ）
-- EBGP	ACI ファブリックは、IPv4 と IPv6 の両方に対して L3 外部での eBGP をサポートします。	--プロトコルによるルート ピアリング, （125 ページ）
-- ホスト vPC FEX	ACI ファブリックは、Cisco ファブリック エクステンダ（FEX）サーバ側仮想ポート チャネル（VPC）（別名 FEX ストレート VPC）をサポートします。	-- FEX 仮想ポート チャネル, （74 ページ）
-- ブリッジ ドメイン単位のマルチキャスト/ブロードキャスト パケット制御	管理者は、これらのパケットの動作をブリッジ ドメインごとに制御できます。	--ブリッジ ドメインとサブネット, （24 ページ）

新機能および変更された機能に関する情報

機能	説明	参照先
-- サービス アプライアンスとのルート ピアリング	ルート ピアリングは、L4-L7 サービス デバイスで OSPF および BGP のピアリングを設定し、接続先の ACI リーフ ノードとルートを交換できるようにするために使用されます。	--トランジット ルーティングの使用例, （120 ページ） --L4-L7 ルート ピアリングの設定, （251 ページ）
-- ポート単位の VLAN	同じリーフ スイッチ上の複数のポートの（複数のブリッジ ドメイン上の）複数の EPG に同じ VLAN ID を設定できます。管理者は、同じスイッチ上のすべてのポートに同じ VLAN ID を設定することが可能になりました。	-- ポート単位の VLAN, （99 ページ）
-- ループ検出	リーフ スイッチ アクセス ポートに接続されたレイヤ 2 ネットワーク セグメントのループを ACI ファブリックで検出できるようになりました。	--ループ検出, （51 ページ）
-- スケール トポロジのためのアトミック カウンタのパス モード		--アトミック カウンタ, （208 ページ）
-- さまざまな更新およびバグ修正	vzAny 概要の追加。 アカウンティング。 デフォルト ポリシー。 コントラクト範囲。 ネットワーク ドメイン。 VMM ドメインの概念を更新し、新しい拡張版『ACI Virtualization Guide』に手順を移動。	--vzAny とは, （34 ページ） --アカウンティング, （139 ページ） --デフォルト ポリシー, （37 ページ） --コントラクト, （31 ページ） -- ネットワーク ドメイン, （96 ページ） --Cisco ACI の VM ネットワーキングによる複数ベンダーの Virtual Machine Manager のサポート, （151 ページ）

表 *6*：*Cisco APIC* リリース *1.0(3x)* の新機能と変更された動作

機能	説明	参照先
--マルチサイト ストレッチ ファブリック	マルチサイト ストレッチ ファブリックのサポートを実装。	--ストレッチ ACI ファブリックの設計の概要，（65 ページ）
--エンドポイント保持のトピックの更新	ブリッジ ドメイン フラッディングの動作を明確に説明（BD内の複数のリーフ スイッチにまたがる EPG サブネット内のエンドポイントの位置を更新）。	--エンドポイントの保持，（49 ページ）
-- フィルタのトピックの更新	フィルタの `matchT All` オプション使用時のベスト プラクティスのガイドラインを提供。	--EPG 通信を制御するラベル、フィルタ、およびサブジェクト，（32 ページ）
-- ストーム制御	レイヤ 2 ストーム制御を実装。	--トラフィック ストーム制御について，（47 ページ）
-- AAA VMM ドメインのタグ	VMM ドメインをセキュリティ ドメインとしてタグ付けすることができます。それにより、セキュリティ ドメインに含まれるユーザが VMM ドメインを表示できるようなります。	--ユーザ アクセス：ロール、権限、セキュリティ ドメイン，（138 ページ）
-- アトミック カウンタのエンドポイント ツー IP アドレス オプション	送信先 MAC アドレスまたは IP アドレスを選択できます。	--アトミック カウンタ，（208 ページ）
-- VMM ドメインのガイドラインを削除	推奨されるワークフロー シーケンスを特定。	--「Virtual Machine Manager のドメイン」の章の VMM ドメインを削除するためのガイドラインのトピックを参照してください。
-- カスタムの RBAC 規則	カスタムの RBAC 規則を開発するためのユース ケース シナリオとガイドラインを特定。	--カスタムの RBAC 規則，（141 ページ） --サンプルの RBAC 規則，（247 ページ）
-- ヘルス スコアの計算	システム、ポッド、テナント、MO レベルのヘルス スコアの計算方法を特定。	--ヘルス スコア，（202 ページ）

新機能および変更された機能に関する情報

機能	説明	参照先
-- マルチノード SPAN ERSPAN のガイドラインおよびヘッダータイプ	ERSPAN を使用するための ERSPAN ヘッダー タイプとガイドラインを特定。	--マルチノード SPAN, （209 ページ）
-- EPG のタグなしおよびタグ付き VLAN ヘッダー	タグなし EPG VLAN の使用のガイドラインおよび制限事項について説明。	--エンドポイント グループ, （27 ページ）
-- ブリッジ ドメイン レガシー モード	レガシー モード ブリッジ ドメイン設定のガイドラインを提供。	--ブリッジ ドメインとサブネット, （24 ページ）
-- AAA LDAP および TCACS+ の設定例の更新	AAA LDAP および TCACS+ の設定例を追加。	--LDAP/Active Directory の認証, （148 ページ） --TACACS+ 認証, （148 ページ）
-- 設定インポート/エクスポートのベスト エフォート、アトミック、マージ、置換オプションの更新	設定インポート/エクスポート ポリシーの強化について説明。	--エクスポート/インポートの設定, （182 ページ）
-- ワイプ オプションでのデコミッションの更新	コミッション リーフ スイッチをワイプ オプションで使用するためのガイドラインの提供。	--ファブリック インベントリ, （62 ページ）
-- DHCP リレーのトピックの更新	DHCP リレーとの関係を確立する際に 1 つのブリッジ ドメインに 1 つのサブレットを設定するための要件に関するガイドラインを提供。	--DHCP リレー, （85 ページ）
-- 読みやすさの向上や画像中の単語のスペルミス修正のための、さまざまなテキスト編集	複数のトピックにおける信頼性の向上と詳細の追加。	「基本」、「プロビジョニング」、「ネットワーキング」の章を参照してください。

第 2 章

Cisco Application Centric Infrastructure

この章の内容は、次のとおりです。

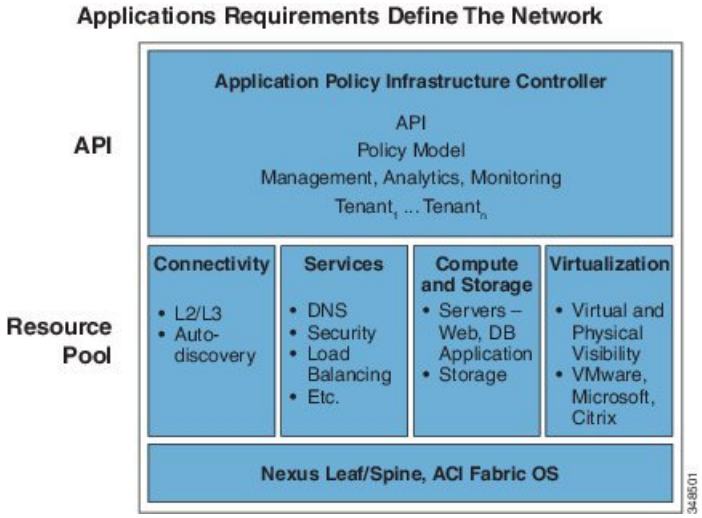
Cisco Application Centric Infrastructure の概要

Cisco アプリケーション セントリック インフラストラクチャ（ACI）では、アプリケーション要件によってネットワークを定義できます。このアーキテクチャにより、アプリケーションの展開ライフサイクル全体が簡素化、最適化、および促進されます。

Cisco Application Policy Infrastructure Controller について

Cisco Application Policy Infrastructure Controller（APIC）API により、アプリケーションはネットワーク、コンピューティング、およびストレージ機能を含む、安全な共有の高パフォーマンス リソース プールと直接接続することができます。次の図は、APIC の概要について説明します。

図 *1*：*APIC* の概要



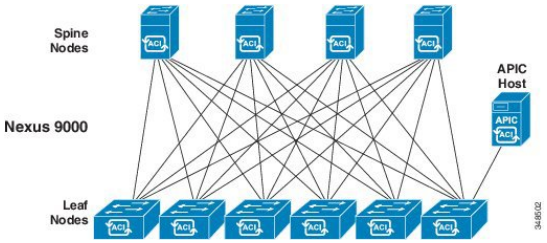
APIC は、スケーラブルな ACI のマルチテナント ファブリックを管理します。APIC は、ファブリックの自動化と管理、ポリシープログラミング、アプリケーション展開、およびヘルスモニタリングの統合ポイントを提供します。複製同期されたクラスタ化コントローラとして実装されるAPICにより、パフォーマンスが最適化され、アプリケーションがあらゆる場所でサポートされ、物理および仮想インフラストラクチャの統合操作が提供されます。APICにより、ネットワーク管理者はアプリケーションの最適なネットワークを容易に定義できます。データセンターのオペレータは、アプリケーションがどのようにネットワーク リソースを消費するかを確認でき、アプリケーションとインフラストラクチャの問題を簡単に切り分けて解決できます。また、リソースの使用パターンをモニタおよびプロファイリングできます。

シスコ アプリケーション セントリック インフラストラクチャ ファブリックの概要

シスコアプリケーションセントリックインフラストラクチャ ファブリック（ACI）のファブリックには、APIC がリーフ/スパイン ACI のファブリック モードで稼働する Cisco Nexus 9000 シリーズ スイッチが含まれます。これらのスイッチは、各リーフ ノードを各スパイン ノードに接続することで、「ファット ツリー」ネットワークを形成します。他のすべてのデバイスは、リーフ ノードに接続します。APIC は、ACI ファブリックを管理します。APIC に対する推奨される最小構成は、3 つの複製されたホストのクラスタです。APIC ファブリック管理機能は、ファブリック

のデータ パスでは動作しません。次の図は、リーフ/スパイン ACI ファブリックの概要を示します。

図 *2*：*ACI* ファブリックの概要

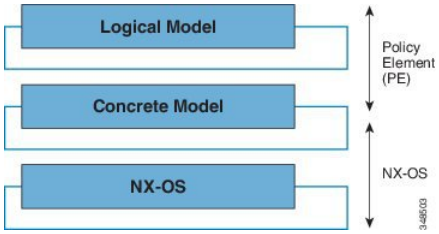


ACI ファブリックでは、高帯域幅リンク（40 Gbps、将来の機能としては 100 Gbps）で一貫した低遅延転送が提供されます。同じリーフ スイッチ上で送信元と宛先を持つトラフィックはローカルで処理され、他のトラフィックはすべて入力リーフから出力リーフへスパイン スイッチを経由して伝送されます。このアーキテクチャは、物理的な観点から 2 つのホップのように見えますが、ファブリックは単一のレイヤ 3 スイッチとして動作するため、実際には単一のレイヤ 3 ホップとなります。

ACI ファブリック オブジェクト指向のオペレーティング システム（OS）は、Cisco Nexus 9000 シリーズの各ノードで動作します。これにより、システムの設定可能な各要素のオブジェクトのプログラミングが可能になります。

ACI ファブリック OS は、ポリシーを APIC から物理インフラストラクチャで動作する具象モデルにレンダリングします。具象モデルはコンパイルされたソフトウェアに類似していて、スイッチのオペレーティング システムが実行できるモデルの形式です。次の図は、論理モデルと具象モデルおよびスイッチ OS との関係を示します。

図 *3*：具象モデルにレンダリングされる論理モデル



すべてのスイッチ ノードには、具象モデルの完全なコピーが含まれます。管理者が APIC で設定を表すポリシーを作成すると、APIC は論理モデルを更新します。次に APIC は、十分に精緻化されたポリシーを作成する中間ステップを実行し、そのポリシーは、具象モデルが更新されるすべてのスイッチ ノードにプッシュされます。

（注） Cisco Nexus 9000 シリーズ スイッチは唯一具象モデルを実行できます。各スイッチには、具象モデルのコピーがあります。APICがオフラインになると、ファブリックは動作し続けますが、ファブリック ポリシーへの変更はできません。

APICは、ファブリックのアクティブ化、スイッチ ファームウェアの管理、ネットワーク ポリシーの設定およびインスタンス化に関与します。APICはファブリックに対する一元化されたポリシーとネットワーク管理エンジンとして機能する一方で、転送トポロジを含むデータ パスから完全に削除されます。したがって、ファブリックは APIC との通信が失われてもトラフィックを転送できます。

Cisco Nexus 9000 シリーズ スイッチでは、モジュラ型および固定型の 1、10、40 ギガビット イーサネット スイッチ設定が提供され、現在の Cisco Nexus スイッチでは Cisco NX-OS スタンドアロン モードとして動作し互換性と一貫性が実現され、ACI モードでは APIC のアプリケーション ポリシーに基づくサービスおよびインフラストラクチャの自動化機能を最大限に活用できます。

ファブリックがどのように動作するかを決定する

ACI ファブリックにより、顧客はクラウド導入に対しスケーラブルで高パフォーマンスのネットワーク、コンピューティングおよびストレージ リソースを自動化し、調整することができます。ACI ファブリックがどのように動作するかを定義するキー プレーヤーには次が含まれます。

- IT プランナー、ネットワーク エンジニア、およびセキュリティ エンジニア
- APIC API 経由でシステムにアクセスする開発者
- アプリケーションおよびネットワーク管理者

Representational State Transfer（REST）アーキテクチャは、クラウド コンピューティングをサポートする重要な開発手法です。ACI API は、REST ベースです。ワールドワイド ウェブは、REST アーキテクチャ スタイルに適合するシステムの最大実装を表します。

クラウド コンピューティングは、規模とアプローチの点で従来のコンピューティングとは異なります。従来の環境には、大幅な運用コストを消費する関連するスキル セットともにソフトウェアおよび保守の要件が含まれます。クラウド アプリケーションは、急激に低下している費用曲線に沿って展開される大規模なインフラストラクチャによってサポートされるシステム設計を使用します。このインフラストラクチャ タイプでは、システム管理者、開発チームおよびネットワーク技術者が協力してより価値のある貢献を行います。

従来の設定では、コンピューティング リソースおよびエンドポイントへのネットワーク アクセスは、仮想 LAN（VLAN）またはロード バランサやファイアウォールなどの堅く定義されたネットワーク サービス経由でトラフィックを強制するマルチプロトコル ラベル スイッチング（MPLS）などの厳格なオーバーレイを通じて管理されます。APICは、プログラマビリティと中央管理を目的に設計されています。ネットワークを抽象化することで、ACI ファブリック上でオペレータはネットワークのリソースをスタティック方式の代わりに動的にプロビジョニングできます。その結果、導入までの時間（市場投入までの時間）が月単位または週単位から分単位に短縮できます。

仮想または物理スイッチ、アダプタ、ポリシー、およびその他のハードウェアおよびソフトウェア コンポーネントの設定変更は、API コールにより数分で行うことができます。

従来の方式からクラウド コンピューティング方式への変換では、データセンターからの柔軟でスケーラブルなサービスへの要求が増大します。これらの変更には、この変換を有効にするためにスキルの高いスペシャリストの大規模プールが要求されます。APICは、プログラマビリティと中央管理を目的に設計されています。APIC の主な機能は、REST と呼ばれる Web API です。APIC REST API は、JavaScript Object Notation（JSON）または Extensible Markup Language（XML）のドキュメントを含む HTTP または HTTPS メッセージを受け入れて返します。現在、多くの Web 開発者が RESTful 方式を使用しています。ネットワーク全体で Web API を採用することで、企業はサービスを容易に開発し他の内部または外部のプロバイダーと組み合わせることができます。このプロセスにより、ネットワークは提供時に静的なリソースの複雑な組み合わせからサービスの動的な交換に変換されます。

第 3 章

ACI ポリシー モデル

この章の内容は、次のとおりです。

ACI ポリシー モデルについて

ACI ポリシー モデルにより、アプリケーション要件のポリシーの指定を行えます。APIC は、ファブリック インフラストラクチャにポリシーを自動的にレンダリングします。ユーザまたはプロセスがファブリック内のオブジェクトへの管理上の変更を開始すると、APIC は最初にポリシー モデルにその変更を適用します。このポリシー モデルの変更により、実際の管理対象エンドポイントへの変更がトリガーされます。この方法を、モデル方式フレームワークといいます。

ポリシー モデルの主な特性

ポリシー モデルの主な特性には次のものがあります。

- モデル駆動型アーキテクチャとして、ソフトウェアはシステム（モデル）の管理および動作状態の完全表記を維持します。モデルはファブリック、サービス、システム動作、およびネットワークに接続された仮想および物理デバイスに均一に適用されます。
- 論理ドメインと具象ドメインが区別されます。論理的な設定は、使用可能な物理リソースに関連するポリシーを適用することで具体的な設定にレンダリングされます。具象エンティティに対して設定は行われません。具象エンティティは、APIC ポリシー モデルの変更の副作用として明示的に設定されます。具象エンティティは、物理的にすることができますが、必ずしもそうする必要はありません（仮想マシンまたは VLAN など）。
- システムは、新しいデバイスを含めるようにポリシー モデルが更新されるまで、新たに接続されたデバイスとの通信を禁止します。
- ネットワーク管理者は、論理的および物理的なシステム リソースを直接設定しませんが、システム動作のさまざまな面を制御する（ハードウェアに依存しない）論理的な設定と APIC ポリシーを定義します。

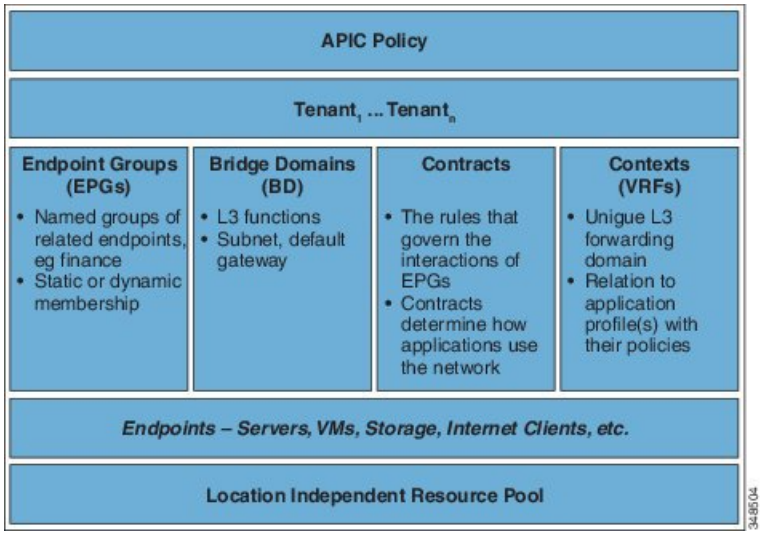
モデル内の管理対象オブジェクトを操作することで、エンジニアは独立した個々のコンポーネントの設定を管理することから開放されます。これらの特性により、自動化と柔軟なワークロードのプロビジョニングが可能になり、インフラストラクチャ内のワークロードをどこでも検索できるようになります。ネットワーク接続されたサービスは簡単に展開でき、APIC により自動化フレームワークが提供され、それらのネットワーク接続されたサービスのライフサイクルを管理できます。

論理構造

ポリシー モデルは、インフラストラクチャ、認証、セキュリティ、サービス、アプリケーション、診断など、ファブリック全体を管理します。ポリシー モデルの論理構造は、ファブリックの

機能のニーズをファブリックがどのように満たすかを定義します。次の図は、ACI ポリシー モデルの論理構造の概要を示します。

図 *4*：*ACI* ポリシー モデルの論理構造の概要



ファブリック全体またはテナントの管理者は、アプリケーションまたは共有リソースの要件を含む事前定義されたポリシーを作成します。これらのポリシーは、アプリケーション、ネットワーク接続サービス、セキュリティポリシー、およびテナントサブネットのプロビジョニングを自動化し、管理者をインフラストラクチャの構成要素ではなくアプリケーションの観点から、リソース プールにアプローチするポジションにします。アプリケーションは、ネットワーキングの動作を誘導する必要があり、その逆ではありません。

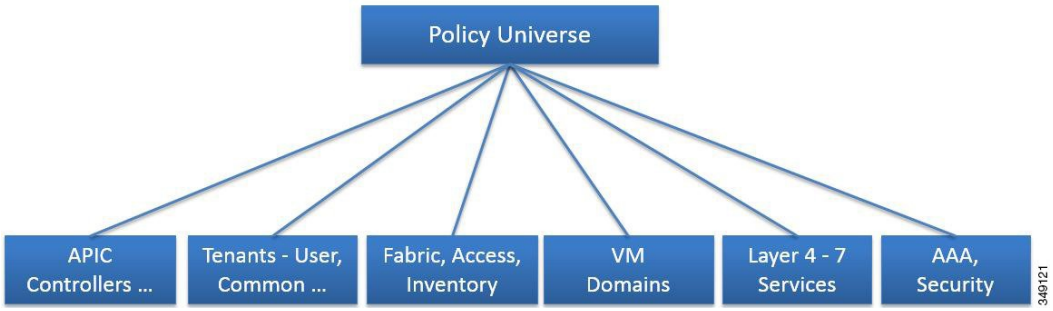
Cisco ACI ポリシー管理情報モデル

ファブリックは、階層型管理情報ツリー（MIT）で表示できる管理情報モデル（MIM）に記録される物理および論理コンポーネントから構成されます。情報モデルは、APIC で実行するプロセスによって保存され管理されます。OSI 共通管理情報プロトコル（CMIP）および他の X.500 バリアントと同様に、APIC によって、MIT の階層構造内のオブジェクトの場所に応じて継承できるオブジェクトのプロパティとして管理可能な特性を示すことにより、管理対象リソースの制御が可能になります。

ツリーの各ノードは、管理対象オブジェクト（MO）またはオブジェクトのグループを表します。MO はファブリック リソースの抽象化です。MO は、スイッチ、アダプタなどの具象オブジェク

ト、またはアプリケーションプロファイル、エンドポイントグループ、または障害などの論理オブジェクトを表すことができます。次の図は、MIT の概要について説明します。

図 *5*：*Cisco ACI* ポリシー管理情報モデルの概要



階層構造は、最上位（ルート）でポリシー ユニバースから始まり、親と子ノードが含まれます。ツリー内の各ノードは MO で、ファブリック内の各オブジェクトには、オブジェクトを説明しツリー内の場所を検索する一意な識別名（DN）があります。

次の管理対象オブジェクトには、システムの動作を管理するポリシーが含まれます。

- APIC コントローラは、マルチテナント ファブリックの管理、ポリシー プログラミング、アプリケーション展開、およびヘルス モニタリングを提供する複製同期されたクラスタ化コントローラを構成します。
- テナントは、ポリシーのコンテナで、管理者はドメインベースのアクセス コントロールを実行できます。システムにより、次の 4 種類のテナントが提供されます。
  - ユーザ テナントは、ユーザのニーズに応じて管理者によって定義されます。アプリケーション、データベース、Web サーバ、ネットワーク接続ストレージ、仮想マシンなどのリソースの動作を管理するポリシーが含まれます。
  - 共通テナントは、システムによって提供されますが、ファブリックの管理者が設定できます。ファイアウォール、ロード バランサ、レイヤ 4 ～ レイヤ 7 のサービス、侵入検知アプライアンスなどのすべてのテナントにアクセス可能なリソースの動作を管理するポリシーが含まれます。
  - インフラストラクチャ テナントは、システムによって提供されますが、ファブリックの管理者が設定できます。ファブリック VXLAN オーバーレイなどのインフラストラクチャ リソースの動作を管理するポリシーが含まれます。また、ファブリック プロバイダーはリソースを 1 つ以上のユーザ テナントに選択的に展開できます。インフラストラクチャ テナント ポリシーは、ファブリックの管理者が設定できます。
  - 管理テナントは、システムによって提供されますが、ファブリックの管理者が設定できます。ファブリック ノードのインバンドおよびアウトオブバンドの設定に使用するファブリック管理機能の動作を管理するポリシーが含まれます。管理テナントには、スイッチの管理ポートを介したアクセスを提供するファブリック データ パスの外部にある APIC/fabric 内部通信用のプライベートなアウトオブバウンド アドレス空間が含まれま

す。管理テナントにより、仮想マシン コントローラとの通信の検出と自動化が可能になります。

- アクセス ポリシーは、ストレージ、コンピューティング、レイヤ 2 およびレイヤ 3（ブリッジドおよびルーテッド）接続、仮想マシン ハイパーバイザ、レイヤ 4 ～ レイヤ 7 のデバイスなどのリソースへの接続を提供するスイッチ アクセス ポートの動作を管理します。テナントが Cisco Discovery Protocol（CDP）、Link Layer Discovery Protocol（LLDP）、Link Aggregation Control Protocol（LACP）、またはスパニング ツリーなどのデフォルトのリンクで提供される設定以外のインターフェイス設定を必要とする場合、管理者はアクセス ポリシーを設定して、リーフ スイッチのアクセス ポートでそのような設定を有効にする必要があります。
- ファブリック ポリシーは、ネットワーク タイムプロトコル（NTP）のサーバ同期、Intermediate System-to-Intermediate System Protocol（IS-IS）、ボーダー ゲートウェイ プロトコル（BGP）のルート リフレクタ、ドメイン ネーム システム（DNS）などの機能を含む、スイッチ ファブリック ポートの動作を管理します。ファブリック MO には、電源、ファン、シャーシなどのオブジェクトが含まれます。
- 仮想マシン（VM）ドメインは、同様のネットワーキング ポリシー要件を持つVM コントローラをグループ化します。VM コントローラは、VLAN または Virtual Extensible Local Area Network（VXLAN）の領域およびアプリケーション エンドポイント グループ（EPG）を共有できます。APIC は VM コントローラと通信し、のちに仮想ワークロードに適用されるポート グループなどのネットワーク設定を公開します。
- レイヤ 4 ～ レイヤ 7 のサービス統合ライフサイクルの自動化フレームワークにより、サービスがオンラインまたはオフラインになったときにシステムは動的に応答することができます。ポリシーは、サービス デバイス パッケージとインベントリ管理機能を提供します。
- アクセス、認証、およびアカウンティング（AAA）ポリシーは、Cisco ACI ファブリックのユーザ権限、ロール、およびセキュリティ ドメインを管理します。

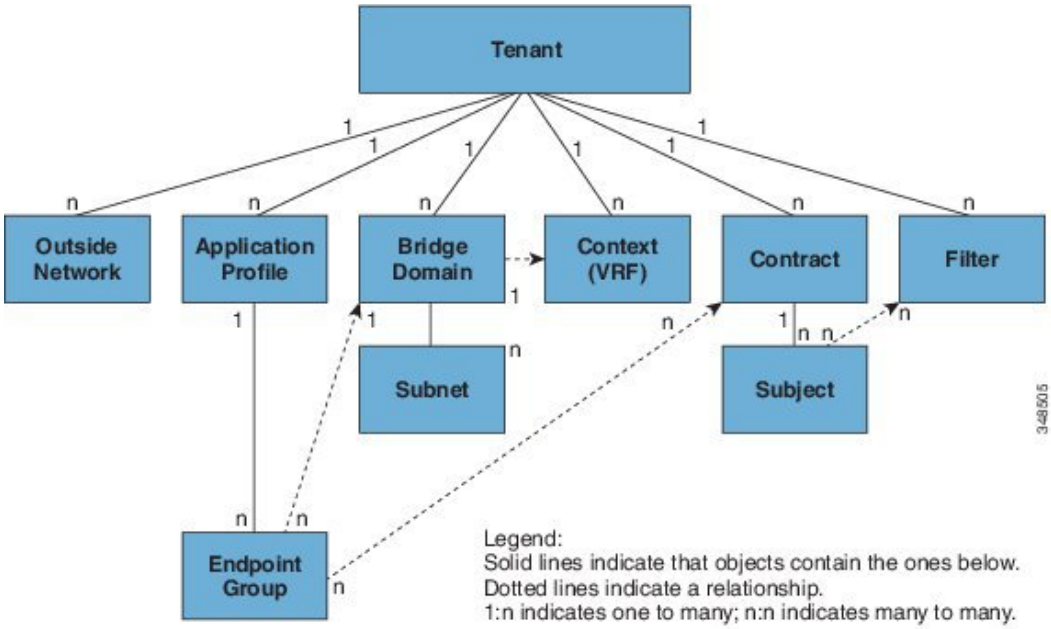
階層型ポリシー モデルは、REST API インターフェイスとうまく適合します。呼び出されると、API は MIT 内のオブジェクトで読み取りまたは書き込みを行います。URL は、MIT 内のオブジェクトを識別する識別名に直接マッピングします。MIT 内のデータは、XML または JSON でエンコードされた自己完結型の構造化ツリー テキスト ドキュメントとして説明できます。

テナント

テナント（`fvTenant`）は、アプリケーション ポリシーの論理コンテナで、管理者はドメインベースのアクセス コントロールを実行できます。テナントはポリシーの観点から分離の単位を表しますが、プライベート ネットワークは表しません。テナントは、サービス プロバイダーの環境では

お客様を、企業の環境では組織またはドメインを、または単にポリシーの便利なグループ化を表すことができます。次の図は、管理情報ツリー（MIT）のテナント部分の概要を示します。

図 *6*：テナント



テナントは相互に分離することも、リソースを共有することもできます。テナントが含む主要な要素は、フィルタ、コントラクト、外部ネットワーク、ブリッジ ドメイン、コンテキスト、およびエンドポイント グループ（EPG）を含むアプリケーション プロファイルです。テナントのエンティティはそのポリシーを継承します。テナントには、1 つ以上の仮想ルーティングおよび転送（VRF）インスタンスまたはコンテキストを含めることができます。各コンテキストは、複数のブリッジ ドメインに関連付けることができます。

（注） テナント ナビゲーション パス の下の APIC GUI では、コンテキスト（VRF）はプライベート ネットワークと呼ばれます。

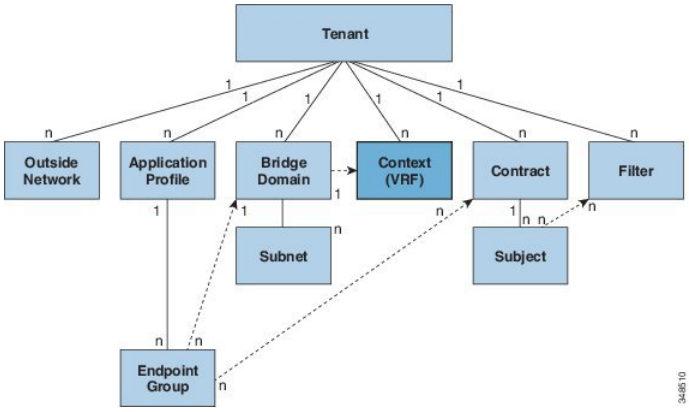
テナントはアプリケーション ポリシーの論理コンテナです。ファブリックには複数のテナントを含めることができます。レイヤ 4 ～ 7 のサービスを展開する前に、テナントを設定する必要があります。ACI ファブリックは、テナント ネットワークに対して IPv4、IPv6、およびデュアル スタック構成をサポートします。

コンテキスト

コンテキスト（`fvCtx`）とはテナントのネットワークのことです（APIC GUI ではプライベート ネットワークと呼んでいます）。テナントには、複数のコンテキストを含めることができます。

コンテキストは、一意なレイヤ 3 フォワーディングおよびアプリケーション ポリシー ドメインです。次の図は、管理情報ツリー（MIT）内のコンテキストの場所とテナントの他のオブジェクトとの関係を示します。

図 7: コンテキスト



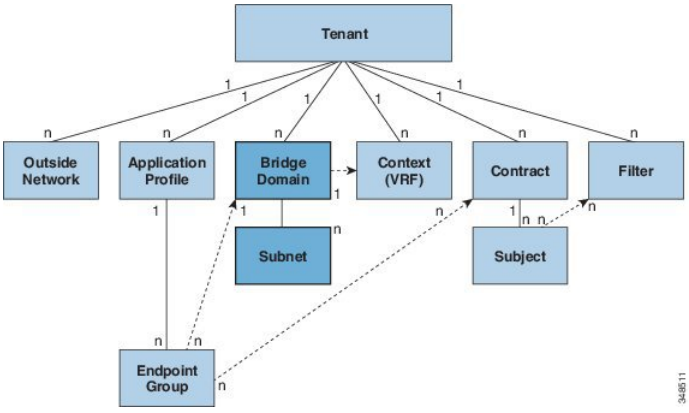
コンテキストは、レイヤ 3 のアドレス ドメインを定義します。1 つ以上のブリッジ ドメインがコンテキストに関連付けられます。レイヤ 3 ドメイン内のすべてのエンドポイントが一意の IP アドレスを持っている必要があります。なぜなら、ポリシーで許可されている場合にこれらのデバイス間でパケットを直接転送できるためです。テナントには、複数のコンテキストを含めることができます。管理者が論理デバイスを作成した後、管理者はデバイス クラスタの選択基準ポリシーを提供する論理デバイス コンテキストを作成できます。論理デバイスは、コントラクト名、グラフ名、またはグラフ内の関数ノード名に基づいて選択できます。

（注） コンテキストは、ネットワーキング ワールドの仮想ルーティングおよび転送（VRF）インスタンスに相当します。APIC GUI では、コンテキストを「プライベート ネットワーク」と呼んでいます。

ブリッジ ドメインとサブネット

ブリッジ ドメイン（fvBD）は、ファブリック内のレイヤ 2 フォワーディングの構造を表します。次の図は、管理情報ツリー（MIT）内のブリッジ ドメインの場所とテナントの他のオブジェクトとの関係を示します。

図 *8*：ブリッジ ドメイン



ブリッジ ドメインは、コンテキストにリンクされ、それに関連付けられた少なくとも 1 個のサブネット（fvSubnet）が必要です。ブリッジ ドメインは、このようなフラッディングがイネーブルの場合に、一意のレイヤ 2 MAC アドレス空間およびレイヤ 2 フラッド ドメインを定義します。コンテキストが一意の IP アドレス空間を定義する一方で、そのアドレス空間は複数のサブネットで構成できます。それらのサブネットは、対応するコンテキストを参照する 1 つ以上のブリッジ ドメインで定義されます。

ブリッジ ドメイン下または EPG 下のサブネットのオプションは次のとおりです。

- *Public*：サブネットをルーテッド接続にエクスポートできます。
- *Private*：サブネットはテナント内にのみ適用されます。
- *Shared*：共有サービスの一部として、同じテナントまたは他のテナントのマルチ コンテキスト（VRF）に対してサブネットの共有やエクスポートを行うことができます。共有サービスの例としては、異なるテナントの別のコンテキスト（VRF）に存在する EPG へのルーテッド接続などがあります。これにより、トラフィックがコンテキスト（VRF）間で双方向に移動することが可能になります。共有サービスを提供する EPG は、サブネットを（ブリッジ ドメイン下ではなく）その EPG 下で設定する必要があります。そのスコープは advertised externally および shared between VRFs に設定する必要があります。

（注） 共有サブネットは、通信に含まれるコンテキスト（VRF）全体で一意でなければなりません。EPG 下のサブネットがレイヤ 3 外部ネットワーク共有サービスを提供する場合、このようなサブネットは、ACI ファブリック内全体でグローバルに一意である必要があります。

ブリッジ ドメイン パケットの動作は次の方法で制御できます。

パケット タイプ	モード
ARP （注） `limitIpLearnToSubnets` を `fvBD` で設定すると、ブリッジ ドメインの設定済みサブネット内または共有サービス プロバイダーである EPG サブネット内に IP が存在する場合のみ、エンドポイントの学習がブリッジ ドメインに限定されます。	ユニキャスト/フラッド
未知のユニキャスト	プロキシ/フラッド
未知の IP マルチキャスト	フラッド モード：パケットは入力およびボーダー リーフ スイッチ ノードでのみフラッディングされます。N9K-93180YC-EX では、パケットは、ブリッジ ドメインが導入されているすべてのノードでフラッディングされます。 OMF モード：1 リーフあたり 50 のブリッジ ドメインのみサポートされます。この制限は N9K-93180YC-EX には該当しません。
L2マルチキャスト、ブロードキャスト、リンクローカル （注） 次のプロトコルはブリッジ ドメインで常にフラッディングされるため、ブリッジ ドメイン モードの設定は適用されません。OSPF/OSPFv6、BGP、EIGRP、LACP、CDP、LLDP、ISIS、IGMP、PIM、ST-BPDU、ARP/GARP、RARP、ND。	BD-flood：ブリッジ ドメインのフラッド encap-flood：カプセル化フラッド drop：パケットをドロップします

ブリッジドメインは複数のスイッチにまたがることができます。ブリッジドメインには複数のサブネットを含めることができますが、サブネットは単一のブリッジ ドメイン内に含まれます。ブリッジドメイン（fvBD）の `limitIPLearnToSubnets` プロパティが `yes` に設定されていると、ブリッ

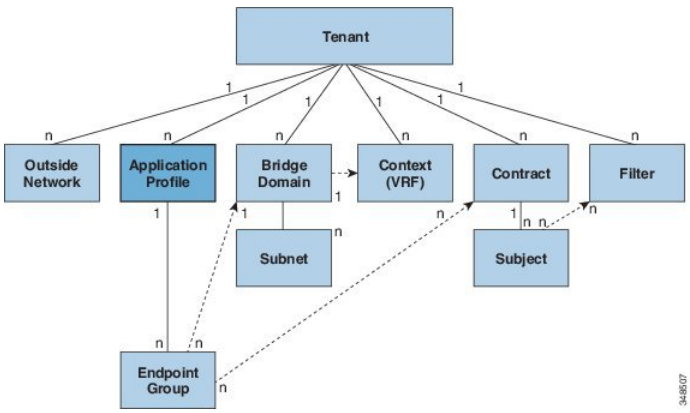
ジ ドメインの設定済みサブネットのいずれかの中に IP アドレスがあるとき、または EPG が共有サービス プロバイダーである場合には EPG サブネット内に IP アドレスがあるときのみ、ブリッジ ドメイン内でエンドポイントの学習が行われます。サブネットは複数の EPG にまたがることができ、1 つ以上の EPG を 1 つのブリッジ ドメインまたはサブネットに関連付けることができます。ハードウェアのプロキシ モードでは、異なるブリッジ ドメインのエンドポイントがレイヤ 3 のルックアップ動作の一部として学習されると、そのエンドポイントに ARP トラフィックが転送されます。

（注） ブリッジ ドメイン レガシー モードでは、ブリッジ ドメインごとに 1 つの VLAN のみ許可されます。ブリッジ ドメイン レガシー モードを指定すると、ブリッジ ドメインを参照するすべての EPG にブリッジ ドメインのカプセル化が使用されます。EPG のカプセル化が定義されていても、それは無視されます。ユニキャスト ルーティングはブリッジ ドメイン レガシー モードには適用されません。レガシーまたは通常モードを組み合わせて操作する複数のブリッジ ドメインをリーフ スイッチに設定することができます。ただし、一度ブリッジ ドメインを設定すると、そのモードを切り替えることはできません。

アプリケーション プロファイル

アプリケーションプロファイル（fvAp）は、アプリケーション要件をモデル化します。アプリケーションプロファイルは、EPG をグループ化する便利な論理コンテナです。次の図は、管理情報ツリー（MIT）内のアプリケーション プロファイルの場所と、テナント内の他のオブジェクトとの関係を示します。

図 *9*: アプリケーション プロファイル



アプリケーション プロファイルには、1 つ以上の EPG が含まれます。最新のアプリケーションには、複数のコンポーネントが含まれます。たとえば、e-コマース アプリケーションには、Web サーバ、データベース サーバ、ストレージ エリア ネットワーク内にあるデータ、および金融取引を可能にする外部リソースへのアクセスが必要となる場合があります。アプリケーション プロファ

イルには、アプリケーションの機能の提供に論理的に関連する必要な数の（またはそれ以下の）EPG が含まれます。

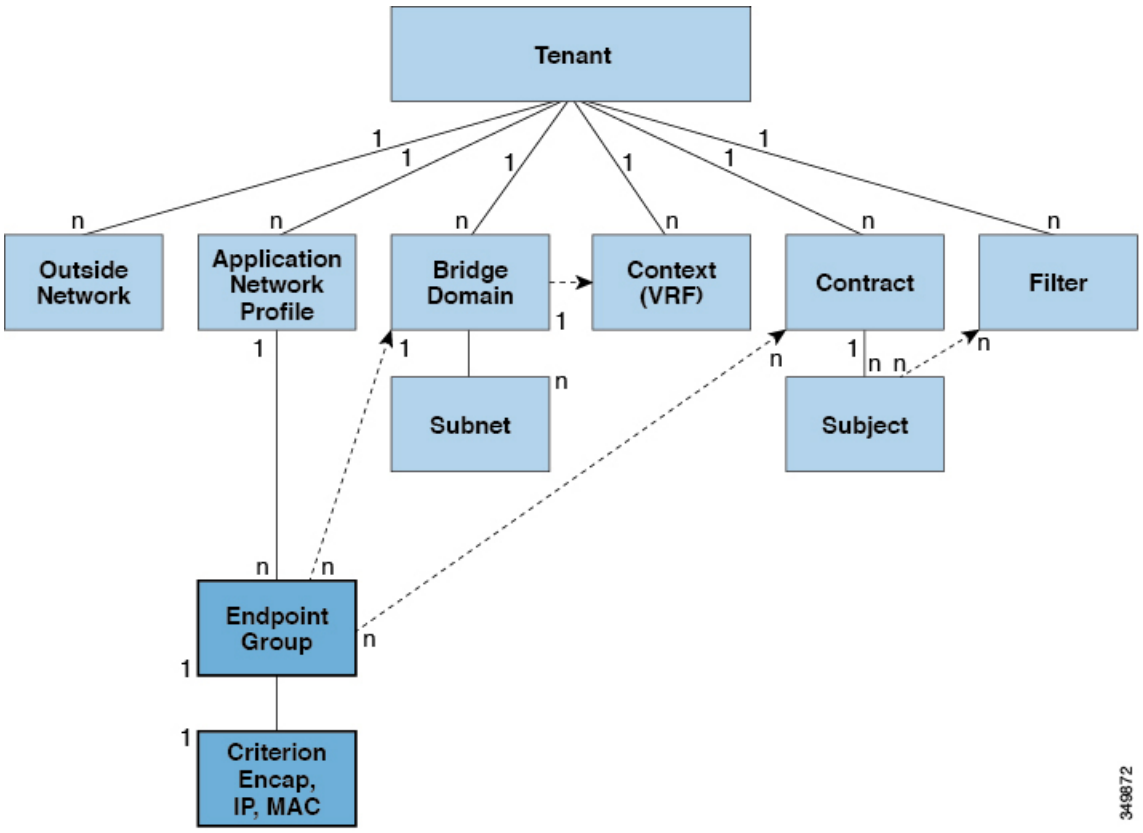
EPG は次のいずれかに従って組織化できます。

- 提供するアプリケーション（付録 A の例にある sap など）
- 提供する機能（インフラストラクチャなど）
- データセンターの構造内の場所（DMZ など）
- ファブリックまたはテナントの管理者が使用することを選択した組織化の原則

エンドポイント グループ

エンドポイント グループ（EPG）は、ポリシー モデルの最も重要なオブジェクトです。次の図は、管理情報ツリー（MIT）内のアプリケーション EPG の場所とテナント内の他のオブジェクトとの関係を示します。

図 *10*：エンドポイント グループ



EPG は、エンドポイントの集合を含む名前付き論理エンティティである管理対象オブジェクトです。エンドポイントは、ネットワークに直接的または間接的に接続されるデバイスです。エンド

ポイントには、アドレス（ID）、ロケーション、属性（バージョンやパッチ レベルなど）があり、物理または仮想にできます。エンドポイントのアドレスを知ることで、他のすべてのIDの詳細にアクセスすることもできます。EPG は、物理および論理トポロジから完全に分離されます。エンドポイントの例には、インターネット上のサーバ、仮想マシン、ネットワーク接続ストレージ、またはクライアントが含まれます。EPG内のエンドポイント メンバシップは、ダイナミックまたはスタティックにできます。

EPGには、セキュリティ、仮想マシンのモビリティ（VMM）、QoS、レイヤ4～レイヤ7サービスなどの共通のポリシー要件を持つエンドポイントが含まれます。エンドポイントは個別に設定および管理されるのではなく、EPG内に配置され、グループとして管理されます。ACIファブリックには、次のタイプの EPG を含めることができます。

- アプリケーション エンドポイント グループ（`fvAEPg`）
- レイヤ 2 外部外側ネットワーク インスタンスのエンドポイント グループ（`l2extInstP`）
- レイヤ 3 外部外側ネットワーク インスタンスのエンドポイント グループ（`l3extInstP`）
- アウトオブバンド（`mgmtOoB`）またはインバンド（`mgmtInB`）アクセス用の管理エンドポイント グループ。

ポリシーはEPGに適用されます。個々のエンドポイントに適用されることは絶対にありません。EPG は、APIC において管理者により静的に設定されるか、vCenter または OpenStack などの自動システムによって動的に設定されます。

（注） EPGがスタティック バインディング パスを使用する場合、このEPGに関連付けられるカプセル化 VLAN はスタティック VLAN プールの一部である必要があります。IPv4/IPv6 デュアルスタック設定の場合、IP アドレスのプロパティは `fvStCEp` MO の `fvStIp` 子プロパティに含まれます。IPv4 および IPv6 アドレスをサポートする複数の `fvStIp` を 1 つの `fvStCEp` オブジェクト下に追加できます。ACI を、IPv4 のみのファームウェアから、IPv6 をサポートするバージョンのファームウェアにアップグレードすると、既存の IP プロパティが `fvStIp` MO にコピーされます。

EPG の設定内容にかかわらず、含まれるエンドポイントに EPG ポリシーが適用されます。

ファブリックへの WAN ルータ接続は、スタティック EPG を使用する設定の 1 つの例です。ファブリックへの WAN ルータ接続を設定するには、関連付けられている WAN サブネット内のエンドポイントを含む `l3extInstP` EPG を管理者が設定します。ファブリックは、エンドポイントの接続ライフサイクルが経過する間に、検出プロセスを通してEPGのエンドポイントについて学習します。エンドポイントを学習すると、ファブリックは、それに基づいて `l3extInstP` EPGポリシーを適用します。たとえば、WAN接続クライアントがアプリケーション（`fvAEPg`）EPG内でサーバとの TCP セッションを開始すると、`l3extInstP` EPG は、`fvAEPg` EPG Web サーバとの通信が始まる前に、そのクライアント エンドポイントにポリシーを適用します。クライアント サーバ TCP セッションが終わり、クライアントとサーバの間の通信が終了すると、そのエンドポイントはもうファブリック内に存在しません。

（注） リーフ スイッチが EPG 下のスタティック バインディング（リーフ）用に設定されている場合は、次の制限が適用されます。

- スタティック バインディングをスタティック パスで上書きすることはできません。
- そのスイッチのインターフェイスをルーテッド外部ネットワーク（L3out）設定に使用することはできません。
- そのスイッチのインターフェイスに IP アドレスを割り当てることはできません。

VMware vCenter への仮想マシン管理接続は、ダイナミック EPG を使用する設定の 1 つの例です。ファブリックで仮想マシン管理ドメインが設定されると、vCenter は、必要に応じて仮想マシン エンドポイントを開始、移動、シャットダウンさせることのできる EPG の動的設定をトリガーします。

通常はカプセル化ベースの EPG が使用されますが、最長プレフィックス照合（LPM）分類によるサポートが不可能な多数の EPG が必要となる設定には、IP ベースの EPG が適しています。IP ベースの EPG は、LPM 分類とは異なり、各 EPG にネットワーク/マスク範囲を割り当てる必要がありません。また、それぞれの IP ベースの EPG が固有のブリッジ ドメインを持つ必要もありません。IP ベースの EPG を設定する手順は、Cisco AVS vCenter 設定で使用される仮想 IP ベースの EPG の設定手順に類似しています。

IP ベースの EPG に関する次のガイドラインと制約事項に従ってください。

- IP ベースの APIC EPG は、APIC 1.1(2x) および ACI スイッチ 11.1(2x) リリース以降（Donner-C ベースの ToR スイッチを含む）でサポートされていますIP ベースの EPG をサポートしていない旧型のスイッチでその導入を試みると、APIC でエラーが発生します。
- IP ベースの EPG は、特定の IP アドレス、IP アドレスの範囲、あるいはその両方の組み合わせに設定できます。
- IP ベースの EPG は、次のシナリオではサポートされません。
  - スタティック EP 設定との組み合わせ。
  - 外部、インフラストラクチャ テナント（infra）、およびレイヤ 2 のみのブリッジ ドメイン設定はブロックされません。このようなケースではレイヤ 3 学習がないため、効力はありません。
  - 同じプレフィクスを共有サービスと IP ベースの EPG に使用することはできません。
  - IP ベースの EPG は、ダイナミック EPG ではサポートされません。サポートされるのはスタティック EPG だけです。IP ベースの EPG プレフィクスがブリッジ ドメイン サブネットの範囲外で設定されても障害は発生しません。ブリッジ ドメイン サブネットと IP ベースの EPG はどのような順序で設定してもよいので、これは、エラー状態にはなりません。IP ベースの EPG プレフィクスの最終的な設定をすべてのブリッジ ドメイン サブネットの外で設定すると、設定は影響力を持たず、エラー状態としてのフラグは付けられません。

マイクロセグメンテーション

マイクロセグメンテーションは、仮想マシン属性、IPアドレス、MACアドレスに基づいて、複数のEPGからのエンドポイントを、1つのマイクロセグメント化EPGに関連付けます。仮想マシン属性に含まれる属性は、次のとおりです。VNic ドメイン名、VM 識別子、VM 名、ハイパーバイザ識別子、VMM ドメイン、データセンター、オペレーティング システム、カスタム属性。

マイクロセグメンテーションの特長には、次のものがあります。

- ライン レートが強化されたステートレスなホワイト リスト ネットワーク アクセス セキュリティ
- ダイナミック レイヤ4～レイヤ7サービス投入およびチェーニングによる、マイクロセグメント単位の細かいセキュリティ自動化。
- 幅広い仮想スイッチ環境における、ハイパーバイザに依存しないマイクロセグメンテーション。
- 問題のある VM を隔離セキュリティ ゾーンに容易に移動できる ACI ポリシー。
- ベア メタルおよび VM エンドポイントのための EPG 内分離と組み合わせると、マイクロセグメンテーションを行うことにより、ポリシーに基づいて自動化された完全なエンドポイント分離をアプリケーション層内で実行することができます。

どのEPGに関しても、ACIファブリック入力リーフ スイッチは、入力ポートに関連付けられたポリシーに従って、パケットを EPG に分類します。マイクロセグメント化 EPG は、マイクロセグメント化 EPG ポリシーで指定された IP アドレス、MAC アドレス、VM 属性に基づいて取得された個々の仮想エンドポイントまたは物理エンドポイントに、ポリシーを適用します。

EPG 内エンドポイントの分離

EPG 内エンドポイント分離ポリシーにより、仮想エンドポイントまたは物理エンドポイントが完全に分離されます。分離が適用されているEPG内でのエンドポイント間通信は許可されません。分離が適用された EGP では、多くのクライアントが共通サービスにアクセスするときに必要な EPG カプセル化の数が削減されますが、相互通信は許可されません。

EPGは、すべてのACIネットワーク ドメインで分離を適用されるか、またはどのドメインでも適用されません。ACI ファブリックは接続エンドポイントに直接分離を実装しますが、ファブリックに接続されているスイッチはプライマリ VLAN（PVLAN）タグに従って分離規則を認識します。

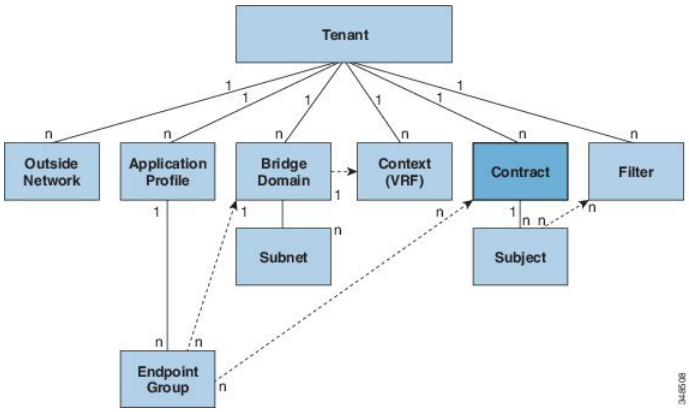
（注） EPG 内エンドポイント分離を適用して EPG が設定されている場合は、以下の制限が適用されます。

- 分離を適用された EPG 全体のすべてのレイヤ 2 エンドポイント通信は、ブリッジ ドメイン内にドロップされます。
- 分離を適用された EPG 全体のすべてのレイヤ 3 エンドポイント通信は、同じサブネット内にドロップされます。

コントラクト

EPG に加えて、コントラクト（vzBrCP）はポリシー モデルの主要オブジェクトです。EPG は唯一、コントラクトのルールに従って他の EPG と通信できます。次の図は、管理情報ツリー（MIT）内のコントラクトの場所とテナントの他のオブジェクトとの関係を示します。

図 *11*：コントラクト



管理者は、コントラクトを使用して許可されるプロトコルやポートを含む EPG 間を通過できるトラフィックのタイプを選択します。コントラクトがなければ、EPG 間通信はデフォルトでディセーブルになります。EPG 内の通信に必要なコントラクトはありません。EPG 内の通信は常に暗黙的に許可されています。

コントラクトは、次のタイプのエンドポイント グループの通信を管理します。

- ACI ファブリック アプリケーション EPG （fvAEPg）間、テナント内およびテナント間の両方

（注） 共有サービス モードの場合、コントラクトはテナント間通信に必要です。テナント コンテキストがポリシーを適用していなくても、コントラクトがコンテキスト間でスタティック ルートを指定するために使用されます。

- ACI ファブリック アプリケーション EPG とレイヤ 2 外部外側ネットワークのインスタンス EPG （l2extInstP）間
- ACI ファブリック アプリケーション EPG とレイヤ 3 外部外側ネットワークのインスタンス EPG （l3extInstP）間
- ACI ファブリック アウトオブバンド（mgmtOoB）またはインバンド（mgmtInB）管理 EPG 間

コントラクトは、プロバイダー、コンシューマ、またはその両方とラベル付された EPG 間の通信を制御します。EPG プロバイダーは、コンシューマ EPG が従う必要のあるコントラクトを公開します。EPG とコントラクトの関係は、プロバイダーまたはコンシューマです。EPG がコントラクトを提供すると、通信が提供されたコントラクトに準拠している限り、その EPG との通信は他の EPG から開始できます。EPG がコントラクトを消費すると、消費する EPG のエンドポイントが、そのコントラクトを提供している EPG の任意のエンドポイントとの通信を開始する場合があります。

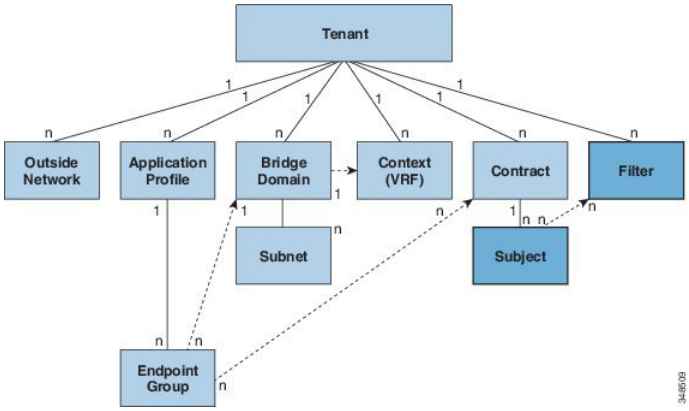
（注） EPG は同じコントラクトを提供および消費できます。EPG は複数のコントラクトを同時に提供および消費することもできます。

EPG 通信を制御するラベル、フィルタ、およびサブジェクト

ラベル、サブジェクト、およびフィルタの管理対象オブジェクトにより、さまざまなアプリケーションまたはサービスの提供要件を満たすための EPG とコントラクト間の混合と照合が可能にな

ります。次の図は、管理情報ツリー（MIT）内のアプリケーション サブジェクトおよびフィルタの場所と、テナント内の他のオブジェクトとの関係を示します。

図 *12*：ラベル、サブジェクトおよびフィルタ



コントラクトには、複数の通信ルールを含めることができ、複数の EPG は複数のコントラクトを消費および提供できます。ラベルは、EPG の特定のペア間で通信が行われるときにどのルールが適用されるかを管理します。ポリシーの設計者は、複雑な通信ポリシーを簡潔に表現でき、アプリケーションの複数のインスタンス間でこれらのポリシーを再利用できます。たとえば、付録 A のサンプル ポリシーでは、HTTP または HTTPS が必要な異なる EPG 間でどのように通信が発生するかを識別するために、同一のコントラクトがどのようにラベル、サブジェクトおよびフィルタを使用するかが示されています。

ラベル、サブジェクトおよびフィルタは次のオプションに従って EPG 通信を定義します。

- ラベルは、プロパティ（名前）を 1 つだけ持つ管理対象オブジェクトです。ラベルにより、互いに通信できるオブジェクトとできないオブジェクトを分類できます。ラベルの一致は最初に行われます。ラベルが一致しない場合、他のコントラクトまたはフィルタ情報は処理されません。ラベルの一致属性は、次の値のいずれかになります。AtLeastOne（デフォルト）、All、None または Exactly One。付録 B は、すべてのラベルの一致タイプとその結果のシンプルな例を示します。

（注） ラベルは、EPG、コントラクト、ブリッジドメイン、DHCP リレーポリシー、および DNS ポリシーなどのさまざまなプロバイダーおよびコンシューマの管理対象オブジェクトに適用できます。ラベルはオブジェクト タイプ間では適用されません。アプリケーション EPG のラベルは、ブリッジ ドメインのラベルと関連がありません。

ラベルは、互いに通信できる EPG コンシューマと EPG プロバイダーを決定します。ラベルの一致により、コントラクトのどのサブジェクトがそのコントラクトの所定の EPG プロバイダーまたは EPG コンシューマに使用できるかが決定されます。

ラベルには次の 2 つのタイプがあります。

- EPG に適用されるサブジェクト ラベル。サブジェクト ラベルの一致により、EPG はコントラクト内のサブジェクトのサブセットを選択することができます。
- EPG に適用されるプロバイダー/コンシューマ ラベル。プロバイダー/コンシューマのラベルの一致により、コンシューマ EPG はプロバイダー EPG を選択でき、その逆も可能です。

- フィルタは、レイヤ 2 ～ レイヤ 4 フィールド、レイヤ 3 プロトコル タイプなどの TCP/IP ヘッダー フィールド、レイヤ 4 ポートなどです。関連するコントラクトに従って、EPG プロバイダーは、IN および OUT 両方の方向でプロトコルおよびポートを決定します。コントラクトのサブジェクトは、コントラクトを提供する側と消費する側のEPGの間に適用されるフィルタ（およびその方向）への関連付けが含まれています。

（注） コントラクトフィルタの一致タイプが`All`である場合は、コンテキスト（VRF）非適用モードを使用することがベスト プラクティスになります。特定の状況下では、これらのガイドラインに従わないと、このコンテキスト（VRF）における EPG 間トラフィックがコントラクトによって拒否されるという結果を招く可能性があります。

- サブジェクトはコントラクトに含まれています。コントラクト内の 1 つ以上のサブジェクトがフィルタを使用して、通信できるトラフィックのタイプと発生の仕方を指定します。たとえば、HTTPS メッセージの場合、サブジェクトはその方向と許可される IP アドレス タイプ（たとえば IPv4）、HTTP プロトコル、およびポートを指定するフィルタを指定します。サブジェクトは、フィルタを単方向にするか双方向にするかを決定します。単方向フィルタは 1 方向で使用されます。単方向フィルタは、IN または OUT の通信を定義しますが、両方に対して同じではありません。双方向フィルタは両方に対して同じで、IN および OUT の通信を定義します。

vzAny とは

`vzAny` とは、基本的に、各 EPG に個別のコントラクト関係を作成するのではなく、コンテキスト（`fvCtx`）内のすべてのエンドポイント グループ（EPG）を 1 つ以上のコントラクト（`vzBrCP`）に関連付けられる、便利な管理対象オブジェクトです。

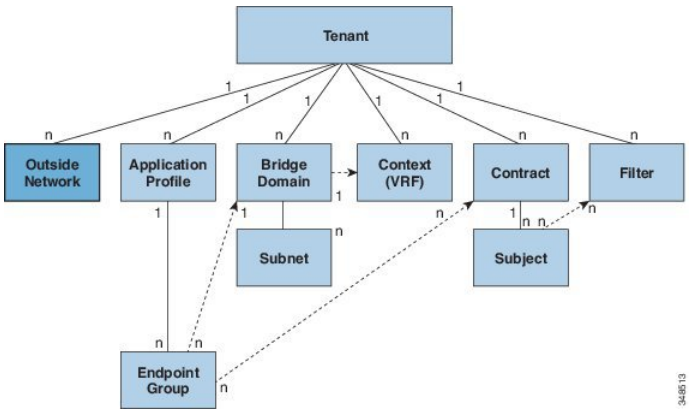
Cisco ACI ファブリックでは、コントラクトのルールにより、EPG は他の EPG としか通信できません。EPG とコントラクトの関係によって、EPG がコントラクトのルールに定義された通信を提供するのか、消費するのか、あるいは提供も消費も行うのかが指定されます。コンテキスト中のすべての EPG にコントラクトのルールを動的に適用することで、`vzAny` により EPG とコントラクトとの関係を設定するプロセスが自動化されます。新しい EPG がコンテキストに追加されるたびに、vzAny コントラクト ルールが自動的に適用されます。`vzAny` と EPG の「1 対すべて」の関係は、コンテキスト中のすべての EPG にコントラクトのルールを適用するための最も効率的な方法です。

（注） テナント下の APIC GUI では、コンテキストはプライベート ネットワーク（テナント内のネットワーク）と呼ばれます。コンテキストは、仮想ルーティングおよび転送（VRF）インスタンスに相当します。

外部ネットワーク

外部ネットワークのオブジェクト ポリシーは、外部への接続を制御します。テナントには、複数の外部ネットワーク オブジェクトを含めることができます。次の図は、管理情報ツリー（MIT）内の外部ネットワークの場所とテナントの他のオブジェクトとの関係を示します。

図 *13*：外部ネットワーク



外部ネットワーク ポリシーは、外部のパブリック/プライベート ネットワークと ACI ファブリック間の通信を制御する関連するレイヤ 2（`l2extOut`）またはレイヤ 3（`l3extOut`）プロパティを指定します。WAN およびエンタープライズ コアに接続するルータや既存のレイヤ 2 スイッチなどの外部デバイスは、リーフ スイッチの前面パネルのインターフェイスに接続します。このような接続を提供するリーフ スイッチは、ボーダー リーフとして知られています。外部デバイスに接続するボーダー リーフ スイッチ インターフェイスは、ブリッジドまたはルーテッド インターフェイスとして設定できます。ルーテッド インターフェイスの場合、スタティックまたはダイナミック ルーティングを使用できます。ボーダー リーフ スイッチは、標準のリーフ スイッチのすべての機能を実行することもできます。

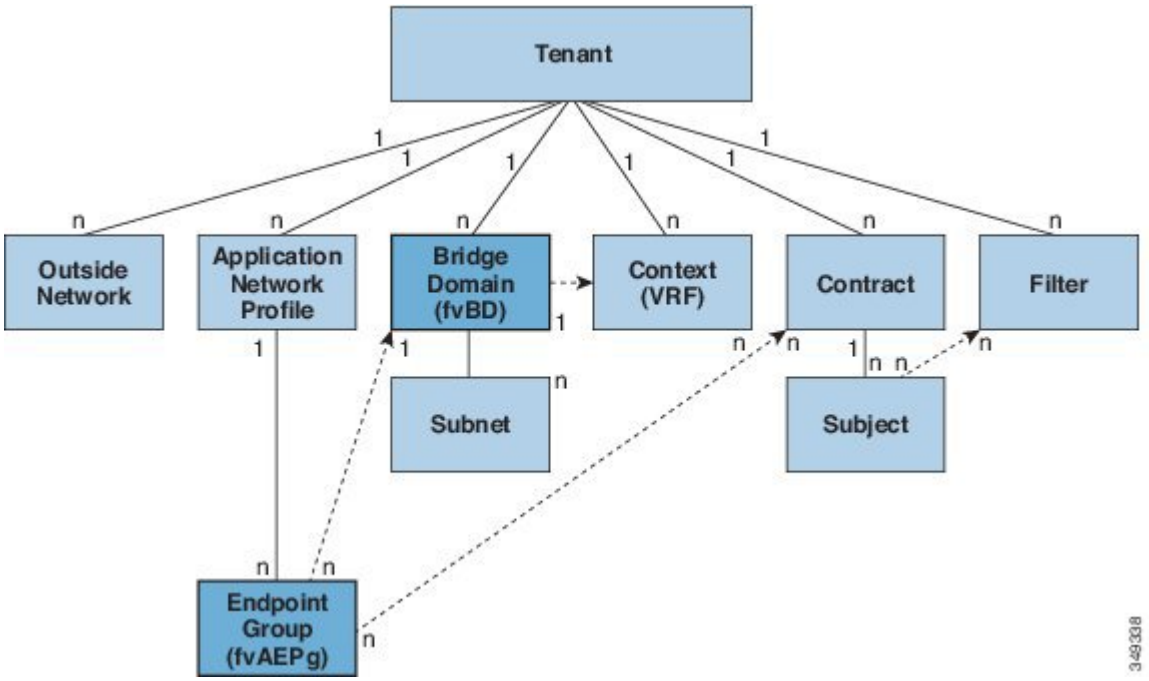
管理対象オブジェクトの関係とポリシー解決

関係管理対象オブジェクトは、抑制（親/子）の関係を共有しない管理対象オブジェクトのインスタンス間の関係を表します。MO の関係は、次の 2 つの方法のいずれかでソース MO とターゲット MO の間に確立されます。

- 明示的な関係（fvRsPathAtt）は、ターゲット MO のドメイン名（DN）に基づいて関係を定義します。
- 名前付きの関係は、ターゲット MO の名前に基づいて関係を定義します。

次の図の点線は、いくつかの一般的な MO の関係を示します。

図 *14*：*MO* の関係



たとえば、EPG とブリッジ ドメイン間の点線は、これら 2 つの MO 間の関係を定義します。この図では、EPG（fvAEPg）には、ターゲットのブリッジ ドメイン MO（fvDB）の名前が付いた関係 MO（fvRsBD）が含まれます。たとえば、実稼働がブリッジ ドメイン名（tnFvBDName=production）である場合、関係の名前は実稼働（fvRsBdName=production）になります。

名前付き関係に基づくポリシー解決の場合は、一致する名前を持つターゲット MO が現在のテナントに見つからない場合、ACI ファブリックは共通のテナントで解決を試行します。たとえば、ユーザのテナント EPG がテナントに存在しないブリッジ ドメインを対象とした関係 MO を含んでいた場合、システムは共通のテナントでその関係の解決を試行します。名前付き関係が現在のテナントまたは共通のテナントで解決できない場合、ACI ファブリックは、デフォルト ポリシーに解決を試行します。デフォルト ポリシーが現在のテナントに存在する場合、それが使用されます。存在しない場合、ACI ファブリックは共通のテナントでデフォルト ポリシーを検索します。ブリッジドメイン、コンテキストおよびコントラクト（セキュリティポリシー）の名前付き関係はデフォルトに解決されません。

デフォルト ポリシー

ACI ファブリックは、そのコア機能の多くにデフォルトのポリシーを含んでいます。これらのデフォルト ポリシーの例は次のとおりです。

- ブリッジ ドメイン（共通テナント内）
- レイヤ 2 およびレイヤ 3 プロトコル
- ファブリック初期化、デバイス検出、配線検出
- ストーム制御とフラッディング
- Virtual port channel（仮想ポート チャネル）
- スイッチ バッファにおける学習済みエンドポイントのキャッシングおよびエージング アウトのためのエンドポイント保持
- ループ検出
- モニタリングと統計情報

次のシナリオでは、共通のポリシー解決動作について説明します。

- 設定では、明示的にデフォルト ポリシーを参照します。現在のテナントにデフォルト ポリシーが存在する場合は、そのポリシーが使用されます。それ以外の場合は、共通テナントのデフォルト ポリシーを使用します。
- 設定では、現在のテナントまたは共通テナントに存在しない指定ポリシー（デフォルト以外）を参照します。現在のテナントにデフォルト ポリシーが存在する場合は、そのポリシーが使用されます。それ以外の場合は、共通テナントのデフォルト ポリシーを使用します。

（注） これは、テナント内のコンテキスト（プライベート ネットワーク）またはブリッジ ドメインには適用されません。

- 設定では、どのポリシー名も参照しません。デフォルト ポリシーが現在のテナントに存在する場合はそのポリシーが使用されます。それ以外の場合は、共通テナントのデフォルト ポリシーを使用します。

（注） ブリッジ ドメインとコンテキストに関しては、共通テナントの接続インストルメンテーション ポリシー（`fvConnInstrPol`）に適切なブリッジ ドメインまたはコンテキストのフラグがセットされている場合のみ、これが適用されます。これにより、意図しない EPG がテナント共通サブネット内に導入されることを防ぎます。

デフォルトポリシーは変更または削除できます。デフォルトポリシーを削除すると、ポリシー解決プロセスの異常終了を招く可能性があります。

（注） デフォルト ポリシーを使用する設定を実行する際の混乱を避けるため、デフォルト ポリシーのドキュメントの変更が行われました。デフォルト ポリシーを削除する前に、現在または将来の設定がデフォルトポリシーに依存していないことを確認してください。たとえば、デフォルトのファームウェア更新ポリシーを削除すると、今後、問題の有るファームウェア更新が発生する可能性があります。

ポリシー モデルは、オブジェクトが自身の下に関係管理対象オブジェクト（MO）を持つことによって別のポリシーを使用していることや、関係 MO が名前によってターゲット ポリシーを参照することを指定します。この関係が、名前による明示的なポリシー参照を行わない場合には、システムは、デフォルトと呼ばれるポリシーを解決しようとします。ブリッジドメイン（BD）およびコンテキスト/VRF（Ctx）は、この規則の例外です。

エンドポイント グループ（EPG）には、`tnFvBDName` というプロパティを持つ BD（`fvRsBd`）との関係があります。これが設定されていない場合（`tnVfBDName=""`）には、接続インストルメンテーションポリシー（`fvConnInstrPol`）がこのようなケースのための動作を取得します。このポリシーは、すべての EPG ケース（VMM、ベアメタル、l2ext、l3ext）に適用されます。インストルメンテーション ポリシーは、`bdctrl` プロパティを使用してデフォルトの BD ポリシーの使用の有無を制御し、`ctxCtrl` プロパティを使用してデフォルトの Ctx（VRF）ポリシーの使用の有無を制御します。次のオプションは、両方に共通です。

- *do not instrument*：リーフ スイッチは、デフォルトのポリシーを使用しません。
- *instrument-and-no-route*：ポリシーをインストルメント化し、ルーティングを有効にしません。
- *instrument-and-route*：ポリシーをインストルメント化し、ルーティングを有効にします。

トランス テナント EPG 通信

あるテナントのEPGが、共有テナントに含まれるコントラクトインターフェイスを介して、他のテナントのEPGと通信することが可能です。コントラクトインターフェイスは、異なるテナントに含まれる EPG によってコントラクト消費インターフェイスとして使用できる MO です。インターフェイスへの関連付けによって、EPG は共有テナントに含まれるコントラクトへのインターフェイスによって表されるサブジェクトを消費します。テナントは第 3 位で定義された単一のコントラクトに参加できます。より厳しいセキュリティ要件は、テナントが互いに完全に独立したままになるようにテナント、コントラクト、サブジェクトおよびフィルタの方向を定義することで満たすことができます。

共有サービス コントラクトの設定時は、次のガイドラインに従ってください。

- インバンド EPG とアウトオブバンド EPG の間でコントラクトが設定される場合、以下の制限が適用されます。
  - 両方の EPG は同じコンテキスト（VRF）にする必要があります。

- フィルタは、着信方向のみに適用されます。
- レイヤ 2 フィルタはサポートされません。
- QoS は、インバンド レイヤ 4 ～ レイヤ 7 のサービスには適用されません。
  - 管理統計情報は利用できません。
  - CPU 宛てトラフィックの共有サービスはサポートされません。

- プライベート ネットワークを適用しない場合、コントラクトがブリッジ間ドメインのトラフィックに必要です。
- プレフィクスベースの EPG はサポートされません。共有サービスはレイヤ 3 外部外側ネットワークではサポートされません。レイヤ 3 外部外側ネットワークによって提供または消費されるコントラクトは、同じレイヤ 3 コンテキストを共有する EPG により消費または提供される必要があります。
- 共有サービスは、重複しないサブネットのみでサポートされます。共有サービスのサブネットを設定するときは、以下のガイドラインに従ってください。
  - 共有サービス プロバイダーのサブネットは、ブリッジ ドメイン下ではなく EPG 下で設定します。
  - 同じコンテキストを共有する EPG で設定されたサブネットは、統合および重複してはなりません。
  - あるコンテキストから他のコンテキストへ漏れたサブネットは統合および重複してはなりません。
  - 複数のコンシューマ ネットワークからあるコンテキストへ漏れたサブネットまたはその逆で漏れたサブネットは統合および重複してはなりません。

  （注） 2 人のコンシューマが誤って同じサブネットに設定されている場合は、両方のサブネットの設定を削除してこの状態からリカバリし、その後サブネットを正しく再設定します。

- プロバイダー コンテキストで共有サービスを `AnyToProv` に設定しないでください。APIC はこの設定を拒否し、エラーが発生します。
- 共有サービスを提供している間は、プロバイダーのプライベート ネットワークは非強制モードにできません。

タグ

オブジェクト タグにより、API 操作が簡素化されます。API 操作では、識別名（DN）の代わりにタグ名でオブジェクトまたはオブジェクトのグループを参照できます。タグは、タグ付けするアイテムの子オブジェクトです。名前以外に他のプロパティはありません。

オブジェクトのグループに記述名を割り当てる際にタグを使用します。同じタグ名を複数のオブジェクトに割り当てることができます。複数のタグ名を 1 つのオブジェクトに割り当てることができます。たとえば、すべての Web サーバの EPG への簡易な検索可能アクセスをイネーブルにするには、このようなすべての EPG に Web サーバ タグを割り当てます。ファブリック全体の Web サーバ EPG は、Web サーバ タグを参照することで検索できます。

第 4 章

ACI ファブリックの基本

この章の内容は、次のとおりです。

ACI ファブリックの基本について

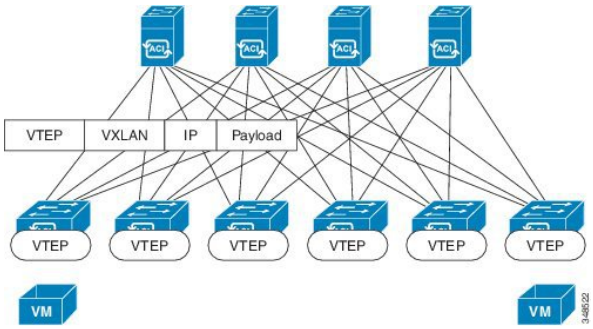
ACI ファブリックは、64,000 以上の専用テナント ネットワークをサポートしています。単一のファブリックは、100 万以上の IPv4/IPv6 エンドポイント、64,000 以上のテナント、および 200,000 以上の 10G ポートをサポートできます。ACI ファブリックにより、物理サービスと仮想サービス間を接続する追加のソフトウェアやハードウェア ゲートウェイを必要とすることなくサービス（物理または仮想）がどこでも可能になり、Virtual Extensible Local Area Network（VXLAN）/VLAN/Network Virtualization using Generic Routing Encapsulation（NVGRE）のカプセル化が正規化されます。

ACI ファブリックは、基盤となる転送グラフからエンドポイント アイデンティティおよび関連するポリシーを分離します。また、最適なレイヤ 3 およびレイヤ 2 フォワーディングを保証する分散レイヤ 3 ゲートウェイが提供されます。ファブリックは、一般的な場所の制約（あらゆる場所の IP アドレス）なしで標準のブリッジングおよびルーティングのセマンティックをサポートし、IP コントロール プレーンの Address Resolution Protocol（ARP）/Gratuitous Address Resolution Protocol（GARP）に関するフラッディング要件を削除します。ファブリック内のすべてのトラフィックは、VXLAN 内にカプセル化されます。

ID と場所の分離

ACI ファブリックは、テナント エンドポイント アドレスとその識別子をそのロケータまたは VXLAN トンネルエンドポイント（VTEP）のアドレスで定義されるエンドポイントの場所から切り離します。次の図は、分離された ID および場所を示します。

図 *15*：*ID* と場所の分離



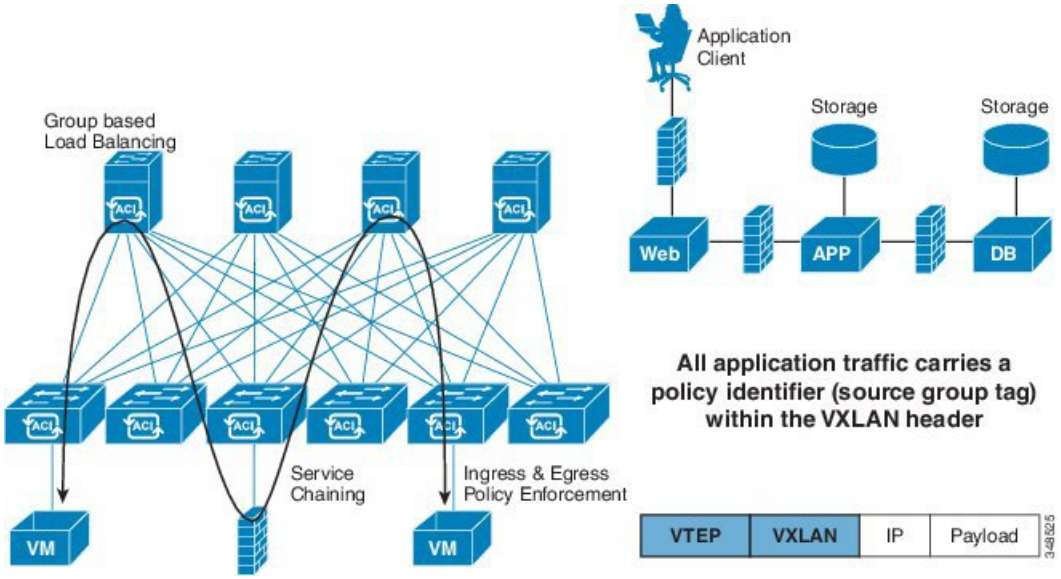
ファブリック内の転送は VTEP 間で行われます。ある場所への内部テナント MAC または IP アドレスのマッピングは、分散マッピング データベースを使用して VTEP によって実行されます。ルックアップが完了したら、VTEP は、宛先リーフ上の VTEP を宛先アドレス（DA）として、VXLAN 内でカプセル化された元データ パケットを送信します。その後パケットは宛先リーフでカプセル化を解除され、受信側ホストに送信されます。このモデルにより、ループの回避にスパニングツリー プロトコルを使用することなく、フル メッシュのループフリー トポロジを確保することが可能になります。

ポリシー ID と適用

アプリケーション ポリシーは、VXLAN パケットで送信される個別のタギング属性を使用して転送から分離されます。ポリシー ID は、ACI ファブリック内のすべてのパケットで送信され、完全

に分散した形でポリシーの一貫した適用を行うことができます。次の図は、ポリシーIDを示します。

図 *16*：ポリシー *ID* と適用

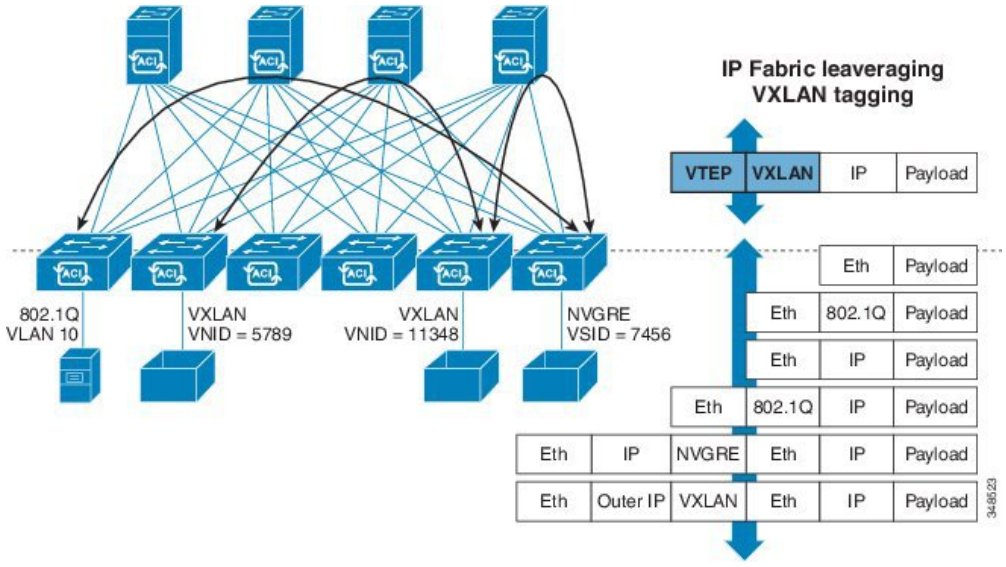


ファブリックおよびアクセスポリシーは、内部のファブリックインターフェイスおよび外部のアクセス インターフェイスの動作を管理します。システムは、デフォルトのファブリックおよびアクセス ポリシーを自動的に作成します。ファブリックの管理者（ファブリック全体へのアクセス権がある者）は、要件に応じてデフォルトのポリシーを変更したり、新しいポリシーを作成できます。ファブリックおよびアクセス ポリシーにより、さまざまな機能やプロトコルを有効にできます。APICのセレクタにより、ファブリックの管理者は、ポリシーを適用するノードおよびインターフェイスを選択できます。

カプセル化の正規化

ファブリック内のトラフィックは、VXLAN としてカプセル化されます。外部の VLAN/VXLAN/NVGRE タグは、内部の VXLAN タグへのイングレスでマッピングされます。次の図は、カプセル化の正規化を示します。

図 *17*：カプセル化の正規化



転送は、カプセル化のタイプまたはカプセル化のオーバーレイ ネットワークによって制限または制約されません。外部識別子は、リーフまたはリーフ ポートにローカライズされ、必要に応じて再利用または変換できます。ブリッジ ドメインのフォワーディング ポリシーは、必要な場合に標準の VLAN 動作を提供するために定義できます。

ネイティブ 802.1p とタグ付き EPG

タグ付けされていないパケットを必要とするデバイスが ACI リーフ スイッチのアクセス ポートに接続されたときに想定どおり動作することを保証するために、次のガイドラインに従ってください。

（注） 1 つのアクセス ポートにつき、ネイティブ 802.1p EPG は 1 つのみ許可されます。

- アクセス ポートにネイティブ 802.1p モードの EPG を 1 つ設定すると、そのパケットはタグなしの状態でポートを退出します。

- アクセス ポートに複数の EPG（1 つのネイティブ 802.1p モードの EPG と、いくつかの VLAN タグ付き EPG）を設定すると、そのアクセス ポートから退出するすべてのパケットに、次の方法でタグが付けられます。
  - タグなしの EPG パケットは、タグなしのままアクセス ポートを退出します。
  - 他の EPG の場合、パケットは、それぞれの VLAN タグを付けられた状態で退出します。
- ある EPG が使用するすべてのポートについて、この EPG にタグ付けしないようリーフ スイッチを設定すると、パケットはタグなしの状態でスイッチを退出します。
- EPG に QoS を設定すると、デフォルト値は QoS クラス 3 となります。コントラクト内で QoS を設定する際には、QoS クラスを明示的に設定する必要があります。コントラクト内で明示的に指定された QoS タギングは、デフォルトの EPG QoS タギングよりも優先されます。

（注） EPG をタグなしとして導入する際は、その EPG を同じスイッチの他のポート上にタグ付きとして導入することは避けてください。

マルチキャスト ツリー トポロジ

ACI ファブリックは、アクセス ポートからのユニキャスト、マルチキャスト、およびブロードキャスト トラフィックの転送をサポートします。エンドポイント ホストからのすべてのマルチデスティネーション トラフィックは、ファブリックにマルチキャスト トラフィックとして伝送されます。

ACI ファブリックは、入力インターフェイスに入るトラフィックを使用可能な中間ステージのスパイン スイッチを介して関連する出力スイッチにルーテッドできる Clos トポロジ（Charles Clos にちなんで名付けられた）に接続されるスパインおよびリーフ スイッチで構成されます。リーフ スイッチには次の 2 種類のポートがあります。スパイン スイッチに接続するためのファブリック ポートと、サーバ、サービス アプライアンス、ルータ、ファブリック エクステンダ（FEX）などを接続するアクセス ポートです。

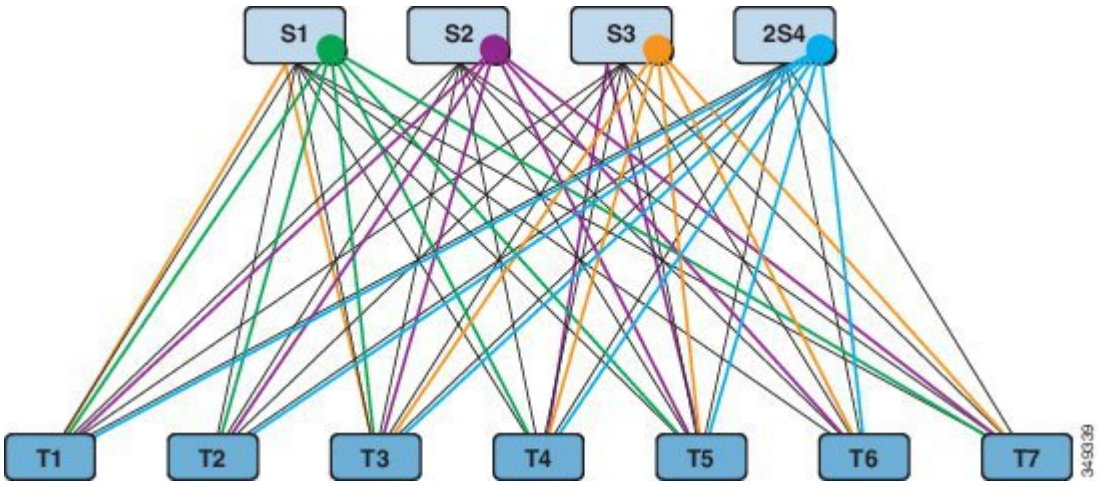
トップ オブ ラック（ToR）スイッチはリーフ スイッチで、スパイン スイッチに接続されます。リーフ スイッチは互いに接続されず、スパイン スイッチはリーフ スイッチのみに接続します。この Clos トポロジでは、すべての下位層のスイッチがフルメッシュ トポロジの最上位層のスイッチにそれぞれ接続されます。スパイン スイッチが故障すると、ACI ファブリック全体のパフォーマンスだけがわずかに低下します。データ パスは、トラフィック負荷がスパイン スイッチ間で均等に分散されるように選択されます。

ACI ファブリックは、Forwarding Tag（FTAG）ツリーを使用してバランス マルチデスティネーション トラフィックをロードします。すべてのマルチデスティネーション トラフィックは、ファブリック内でカプセル化された IP マルチキャスト トラフィックの形式で転送されます。入力リーフは、FTAG をスパインに転送するときにトラフィックに割り当てます。FTAG は宛先マルチキャスト アドレスの一部としてパケットに割り当てられます。ファブリックでは、トラフィックは指定された FTAG ツリーに沿って転送されます。スパインおよび中間リーフ スイッチは、FTAG ID

に基づいてトラフィックを転送します。転送ツリーは、FTAG ID 1 つにつき 1 個構築されます。任意の 2 つのノード間で、FTAG 1 つにつきリンク 1 つだけが転送されます。複数の FTAG を使用することで、転送に異なるリンクを使用している各 FTAG でパラレル リンクを使用できます。ファブリック内の FTAG ツリーの数が多いほど、ロード バランシングの効果が大きい可能性があるということになります。ACI ファブリックは、最大 12 個の FTAG をサポートします。

次の図は、4 つの FTAG によるトポロジを示します。ファブリック内のすべてのリーフ スイッチは、各 FTAG に直接または中継ノードを介して接続されます。1 つの FTAG が各スパイン ノードに根付いています。

図 *18*：マルチキャスト ツリー トポロジ



リーフ スイッチはスパインへの直接接続性がある場合、直接パスを使用して FTAG ツリーに接続します。直接リンクがない場合、リーフ スイッチは上記の図に示すように FTAG ツリーに接続されている中継ノードを使用します。図には、各スパインが 1 つの FTAG ツリーのルートとして示されていますが、複数の FTAG ツリー ルートを 1 つのノード上に置くことができます。

ACI ファブリック起動検出プロセスの一環として、FTAG ルートはスパイン スイッチに配置されます。APIC は、各スパイン スイッチをスパインがアンカーする FTAG で設定します。ルートの ID と FTAG の数は設定から取得されます。APIC は、使用される FTAG ツリーの数と各ツリーに対するルートを指定します。FTAG ツリーは、ファブリックでトポロジの変更があるたびに再計算されます。

ルートの配置は誘導される設定で、スパイン スイッチの障害などのランタイム イベントで動的に再度ルート付けされることはありません。通常、FTAG設定はスタティックです。スパイン スイッチの追加または削除時は、管理者がスパイン スイッチの残りのセットまたは拡張セット間でFTAGを再配布することを決める可能性があるため、FTAG はあるスパインから別のスパインへ再アンカーできます。

トラフィック ストーム制御について

トラフィック ストームは、パケットが LAN でフラッディングする場合に発生するもので、過剰なトラフィックを生成し、ネットワークのパフォーマンスを低下させます。トラフィック ストーム制御ポリシーを使用すると、物理インターフェイス上におけるブロードキャスト、未知のマルチキャスト、または未知のユニキャストのトラフィック ストームによって、レイヤ 2 ポート経由の通信が妨害されるのを防ぐことができます。

デフォルトでは、ストーム制御は ACI ファブリックでは有効になっていません。ACI ブリッジ ドメイン（BD）レイヤ 2 の未知のユニキャストのフラッディングは BD 内でデフォルトで有効になっていますが、管理者が無効にすることができます。その場合、ストーム制御ポリシーはブロードキャストと未知のマルチキャストのトラフィックにのみ適用されます。レイヤ 2 の未知のユニキャストのフラッディングが BD で有効になっている場合、ストーム制御ポリシーは、ブロードキャストと未知のマルチキャストのトラフィックに加えて、レイヤ 2 の未知のユニキャストのフラッディングに適用されます。

トラフィック ストーム制御（トラフィック抑制ともいいます）を使用すると、着信するブロードキャスト、マルチキャスト、未知のユニキャストのトラフィックのレベルを 1 秒間隔でモニタできます。この間に、トラフィック レベル（ポートで使用可能な合計帯域幅のパーセンテージ、または特定のポートで許可される 1 秒あたりの最大パケット数として表されます）が、設定したトラフィック ストーム制御レベルと比較されます。入力トラフィックが、ポートに設定したトラフィック ストーム制御レベルに到達すると、トラフィック ストーム制御機能によってそのインターバルが終了するまでトラフィックがドロップされます。管理者は、ストーム制御しきい値を超えたときにエラーを発生させるようにモニタリング ポリシーを設定できます。

ストーム制御のガイドライン

以下のガイドラインと制約事項に従って、トラフィック ストーム制御レベルを設定してください。

- 通常、ファブリック管理者は以下のインターフェイスのファブリック アクセス ポリシーでストーム制御を設定します。
  - 標準トランク インターフェイス。
  - 単一リーフ スイッチ上のダイレクト ポート チャネル。
  - バーチャル ポート チャネル（2 つのリーフ スイッチ上のポート チャネル）。
- ポート チャネルおよびバーチャル ポート チャネルでは、ストーム制御値（1 秒あたりのパケット数またはパーセンテージ）はポート チャネルのすべての個別メンバーに適用されます。ポート チャネルのメンバーであるインターフェイスには、ストーム制御を設定しないでください。

（注） APIC 1.3(x) 以降のスイッチ ハードウェアおよびスイッチ 11.3(x) リリースでは、ポート チャネルの設定に関し、集約されるポートのトラフィック抑制が最大で設定値の 2 倍となることがあります。新しいハードウェア ポートは、内部的に slice-0 と slice-1 の 2 グループに細分されます。スライシング マッピングを確認するには、`vsh_lc` コマンド `show platform internal hal l2 port gpd` を使い、`slice 0` または `slice 1` を `Sl` 列で探します。ポートチャネル メンバーが slice-0 と slice-1 の両方にある場合は、各スライスごとに公式が計算されるため、許容されるストーム制御トラフィックが設定値の倍になることがあります。

- 使用可能な帯域幅のパーセンテージで設定する場合、値 100 はトラフィック ストーム制御を行わないことを意味し、値 0.01 はすべてのトラフィックを抑制します。
- ハードウェアの制限およびさまざまなサイズのパケットのカウント方式が原因で、レベルのパーセンテージは概数になります。着信トラフィックを構成するフレームのサイズに応じて、実際に適用されるパーセンテージ レベルと設定したパーセンテージ レベルの間には、数パーセントの誤差がある可能性があります。1 秒あたりのパケット数（PPS）の値は、256 バイトに基づいてパーセンテージに変換されます。
- 最大バーストは、通過するトラフィックがないときに許可されるレートの最大累積です。トラフィックが開始されると、最初の間隔では累積レートまでのすべてのトラフィックが許可されます。後続の間隔では、トラフィックは設定されたレートまでのみ許可されます。サポートされる最大数は 65535 KB です。設定されたレートがこの値を超えると、PPS とパーセンテージの両方についてこの値で制限されます。
- 累積可能な最大バーストは 512 MB です。
- 最適化されたマルチキャスト フラッディング（OMF）モードの出力リーフ スイッチでは、トラフィック ストーム制御は適用されません。
- OMF モードではない出力リーフ スイッチでは、トラフィック ストーム制御が適用されます。
- FEX のリーフ スイッチでは、ホスト側インターフェイスにはトラフィック ストーム制御を使用できません。

ロード バランシング

ACI ファブリックでは、利用可能なアップリンク リンク間のトラフィックを平衡化するためのロード バランシング オプションがいくつか提供されます。スタティック ハッシュ ロード バランシングは、各フローが 5 タプルのハッシュに基づいてアップリンクに割り当てられるネットワークで使用される従来のロードバランシング機構です。このロードバランシングにより、使用可能なリンクにほぼ均等な流量が分配されます。通常、流量が多いと、流量の均等な分配により帯域幅も均等に分配されます。ただし、いくつかのフローが残りよりも多いと、スタティック ロード バランシングにより完全に最適ではない結果がもたらされる場合があります。

ダイナミック ロードバランシング（DLB）により、輻輳レベルに従ってトラフィックの割り当てが調整されます。DLB では、使用可能なパス間の輻輳が測定され、輻輳状態が最も少ないパスにフローが配置されるので、データが最適またはほぼ最適に配置されます。

DLB は、フローまたはフローレットの粒度を使用して使用可能なアップリンクにトラフィックを配置するように設定できます。フローレットは、時間の大きなギャップによって適切に区切られるフローからのパケットのバーストです。パケットの 2 つのバースト間のアイドル間隔が使用可能なパス間の遅延の最大差より大きい場合、2 番目のバースト（またはフローレット）を 1 つ目とは異なるパスに沿ってパケットのリオーダーなしで送信できます。このアイドル間隔は、フローレット タイマーと呼ばれるタイマーによって測定されます。フローレットにより、パケット リオーダーを引き起こすことなくロード バランシングに対する粒度の高いフローの代替が提供されます。

DLB動作モードは積極的または保守的です。これらのモードは、フローレット タイマーに使用するタイムアウト値に関係します。アグレッシブ モードのフローレット タイムアウトは比較的小さい値です。この非常に精密なロード バランシングはトラフィックの分配に最適ですが、パケット リオーダーが発生する場合があります。ただし、アプリケーションのパフォーマンスに対する包括的なメリットは、保守的なモードと同等かそれよりも優れています。保守的なモードのフローレット タイムアウトは、パケットが並び替えられないことを保証する大きな値です。新しいフローレットの機会の頻度が少ないので、トレードオフは精度が低いロード バランシングです。DLB は常に最も最適なロード バランシングを提供できるわけではありませんが、スタティック ハッシュ ロード バランシングより劣るということはありません。

ACI ファブリックは、リンクがオフラインまたはオンラインになったことで使用可能なリンク数が変化すると、トラフィックを調整します。ファブリックは、リンクの新しいセットでトラフィックを再分配します。

スタティックまたはダイナミックのロード バランシングのすべてのモードでは、トラフィックは、Equal Cost Multipath（ECMP）の基準を満たすアップリンクまたはパス上でのみ送信され、これらのパスはルーティングの観点から同等で最もコストがかかりません。

ロード バランシング技術ではありませんが、Dynamic Packet Prioritization（DPP）は、スイッチで DLB と同じメカニズムをいくつか使用します。DPP の設定は DLB 専用です。DPP は、長いフローよりも短いフローを優先します。短いフローは約 15 パケット未満です。短いフローは、長いフローよりも遅延に敏感であるため、DPP はアプリケーション全体のパフォーマンスを向上できます。

ACIファブリックのデフォルト設定では、従来の静的なハッシュが使用されます。静的なハッシュ機能により、アップリンク間のトラフィックがリーフ スイッチからスパイン スイッチに分配されます。リンクがダウンまたは起動すると、すべてのリンクのトラフィックが新しいアップリンク数に基づいて再分配されます。

エンドポイントの保持

スイッチでキャッシュ エンドポイントの MAC アドレスと IP アドレスを保持することで、パフォーマンスが向上します。スイッチは、アクティブになるときにエンドポイントについて学習します。ローカル エンドポイントはローカル スイッチにあります。リモート エンドポイントは他のスイッチにありますが、ローカルでキャッシュされます。リーフ スイッチは、直接（または直接接続さ

れたレイヤ 2 スイッチまたはファブリック エクステンダを通じて）接続されたエンドポイント、ローカルエンドポイント、およびファブリックの他のリーフ スイッチに接続されたエンドポイント（ハードウェアのリモート エンドポイント）に関する場所とポリシーの情報を保存します。スイッチは、ローカル エンドポイントには 32 Kb エントリ キャッシュを、リモート エンドポイントには 64 Kb エントリ キャッシュを使用します。

リーフ スイッチで稼働するソフトウェアは、これらのテーブルを能動的に管理します。ローカル的に接続されたエンドポイントでは、ソフトウェアは各エントリの保持タイマーの期限切れ後にエントリをエージングアウトします。エンドポイントエントリは、エンドポイントのアクティビティが終了するとスイッチ キャッシュからプルーニングされ、エンドポイントの場所が他のスイッチに移動するか、またはライフサイクルの状態がオフラインに変わります。ローカル保持タイマーのデフォルト値は 15 分です。非アクティブのエントリを削除する前に、リーフ スイッチはエンドポイントに 3 つの ARP 要求を送信し、実際になくなっているかを確認します。スイッチが ARP 応答を受信しない場合、エントリがプルーニングされます。リモートで接続されたエンドポイントの場合、スイッチは非アクティブになってから 3 分後にエントリをエージング アウトします。リモート エンドポイントは、再度アクティブになるとテーブルにすぐに再入力されます。

（注） エンドポイントが再度キャッシュされるまでリモート リーフ スイッチで適用されるポリシー以外にテーブルにリモート エンドポイントがなくても、パフォーマンスのペナルティはありません。

ブリッジ ドメインのサブネットが適用に設定されている場合は、エンドポイントの保持ポリシーは次のように動作します。

- ブリッジ ドメインのサブネットに含まれていない IP アドレスを持つ新しいエンドポイントは学習されません。
- デバイスがトラッキングに応答しない場合、すでに学習済みのエンドポイントは、エンドポイント保持キャッシュからエージ アウトします。

この適用プロセスは、サブネットがブリッジ ドメイン下で定義されているかどうか、またはサブネットが EPG 下で定義されているかどうかにかかわらず、同様に動作します。

エンドポイントの保持タイマー ポリシーは変更できます。スタティック エンドポイントの MAC および IP アドレスを設定すると、保持タイマーをゼロに設定することで、スイッチ キャッシュに永久的に保存できます。エントリの保持タイマーをゼロに設定することは、それが自動削除されないことを意味します。この操作は慎重に行う必要があります。エンドポイントが移動したりポリシーが変化する場合は、APIC を介してエントリを手動で最新情報に更新する必要があります。保持タイマーがゼロ以外の場合、この情報は APIC の介入なしで各パケットで確認され瞬時に更新されます。

ブリッジ ドメイン（BD）は、複数のリーフ スイッチ（たとえば、リーフ スイッチ 1、リーフ スイッチ 2、リーフ スイッチ 3、リーフ スイッチ 4）にまたがることのできるフラッド ドメインです。BD には、VLAN とみなすことのできる EPG が 1 つ以上含まれます。EPG は、BD のリーフ スイッチ（たとえばリーフ スイッチ 1 とリーフ スイッチ 2）でのみ導入できます。パケットが入力リーフ スイッチに到達したとき、入力リーフ スイッチには、どのリーフ スイッチが出力リーフ スイッチなのかわかりません。したがって、パケットは BD 内のすべてのリーフ スイッチ（こ

の例ではリーフ スイッチ1、リーフ スイッチ2、リーフ スイッチ3、リーフ スイッチ4）にフラッディングされます。このフラッディングは、EPG 内のエンドポイントを学習する必要があるときに発生します。それに伴って、リーフスイッチエンドポイント保持テーブルが更新されます。この例の EPG はリーフ スイッチ3 またはリーフ スイッチ4 に関連していないため（これらのスイッチ上では VLAN が非該当）、これらのリーフ スイッチはこれらのパケットをドロップします。

（注） ブリッジ ドメインに存在しないホストはこれらのパケットを受信しません。

エンドポイントの保持ポリシーは、プルーニングがどのように行われるかを決定します。ほとんどの場合、デフォルトのポリシー アルゴリズムが使用されます。エンドポイントの保持ポリシーを変更すると、システム パフォーマンスに影響を与える場合があります。何千ものエンドポイントと通信するスイッチの場合、エージング間隔を短くすると、多数のアクティブなエンドポイントをサポートするのに使用可能なキャッシュ ウィンドウの数が増えます。エンドポイントの数が 10,000 を超える場合は、複数のスイッチにエンドポイントを分散させることを推奨します。

デフォルトのエンドポイント保存ポリシーの変更に関する次のガイドラインに従ってください。

- リモート バウンス間隔 =（リモート エージ X 2）+ 30 秒
  - 推奨デフォルト値：
    - ローカル エージ = 900 秒
    - リモート エージ = 300 秒
    - バウンス エージ = 630 秒
- アップグレードに関する考慮事項：リリース 1.0(1k) より前のバージョンにアップグレードする際には、共通テナント下のエンドポイント保持ポリシー（`epRetPol`）のデフォルト値が次のとおりであることを確認してください。バウンス エージ = 660 秒

ループ検出

ACI ファブリックは、ACI アクセス ポートに接続されているレイヤ 2 ネットワーク セグメントでループを検出できるグローバルデフォルトループ検出ポリシーを提供します。これらのグローバルポリシーはデフォルトで無効になっていますが、ポート レベルのポリシーはデフォルトで有効になっています。グローバルポリシーを有効にすると、個々のポート レベルで無効にしていない限り、すべてのアクセス ポート、仮想ポート、仮想ポート チャネルでポリシーが有効になります。

ACI ファブリックは、スパニング ツリー プロトコル（STP）には参加しません。代わりに、配線ミスプロトコル（MCP）を実行してループを検出します。MCP は外部レイヤ 2 ネットワークで実行される STP を補完するように機能し、アクセス ポートが受信するブリッジ プロトコル データ ユニット（BPDU）パケットを処理します。

（注） VPC で ACI ファブリックに接続されスパニング ツリーを実行している外部スイッチからのインターフェイスは、loop_inc 状態になる可能性があります。この問題は、外部スイッチからのポート チャネルのフラッピングによって解決します。BPDU Filter を有効にするか、外部スイッチのループガードを無効にすると、問題を防止できます。

ファブリック管理者は、ACI ファブリックによって開始された MCP パケットを識別するために MCP が使用するキーを提供します。管理者は、MCP ポリシーがループを識別する方法、ループに対するアクション（syslog のみ、またはポートを無効にする）を選択できます。

VM 移動などのエンドポイント移動は、それ自体が正常でも、頻度が高く、移動の間隔が短ければ、ループ症状を示す可能性があります。個別のグローバルデフォルトエンドポイント移動ループ検出ポリシーを使用できますが、デフォルトでは無効になっています。管理者は、ループ検出時の対処方法を選択できます。

また、エラーディセーブル回復ポリシーにより、ループ検出および BPDU ポリシーによって無効にされたポートを、管理者が設定した間隔が経過した後に有効にすることができます。

ACI ファブリック ネットワーク アクセス セキュリティ ポリシー モデル

ACI のファブリック セキュリティ ポリシー モデルはコントラクトに基づいています。このアプローチにより、従来のアクセス コントロール リスト（ACL）の制限に対応できます。コントラクトには、エンドポイント グループ間のトラフィックで適用されるセキュリティ ポリシーの仕様が含まれます。

EPG 通信にはコントラクトが必要です。EPG/EPG 通信はコントラクトなしでは許可されません。APIC は、コントラクトや関連する EPG などのポリシー モデル全体を各スイッチの具象モデルにレンダリングします。入力時に、ファブリックに入るパケットはすべて、必要なポリシーの詳細でマークされます。EPG の間を通過できるトラフィックの種類を選択するためにコントラクトが必要とされるので、コントラクトはセキュリティ ポリシーを適用します。コントラクトは、従来のネットワーク設定でのアクセス コントロール リスト（ACL）によって扱われるセキュリティ要件を満たす一方で、柔軟性が高く、管理が容易な、包括的なセキュリティ ポリシー ソリューションです。

アクセス コントロール リストの制限

従来のアクセス コントロール リスト（ACL）には、ACI ファブリック セキュリティ モデルが対応する多数の制限があります。従来の ACL は、ネットワーク トポロジと非常に強固に結合されています。それらは通常、ルータまたはスイッチの入力および出力インターフェイスごとに設定され、そのインターフェイス、およびそれらのインターフェイスを流れることが予期されるトラフィックに合わせてカスタマイズされます。このカスタマイズにより、それらは多くの場合イン

ターフェイス間で再利用できません。もちろんこれはルータまたはスイッチ間にも当てはまります。

従来の ACL は、非常に複雑で曖昧です。なぜなら、そのリストには、許可された特定の IP アドレス、サブネット、およびプロトコルのリストと、明確に許可されていない多くのものが含まれているためです。この複雑さは、問題が生じるのを管理者が懸念してACLルールを削除するのを躊躇するため、維持が困難で、多くの場合は増大するだけということを意味します。複雑さは、それらが通常 WAN と企業間または WAN とデータセンター間の境界などのネットワーク内の特定の境界ポイントでのみ配置されていることを意味します。この場合、ACL のセキュリティのメリットは、エンタープライズ内またはデータセンターに含まれるトラフィック向けには生かされません。

別の問題として、1 つの ACL 内のエントリ数の大幅増加が考えられます。ユーザは多くの場合、一連の送信元が一連のプロトコルを使用して一連の宛先と通信するのを許可するACLを作成します。最悪の場合、*N* の送信元が *K* のプロトコルを使用して *M* の宛先と対話する場合、ACL に *N*M*K* の行が存在する場合があります。ACLは、プロトコルごとに各宛先と通信する各送信元を一覧表示する必要があります。また、ACL が非常に大きくなる前に多くのデバイスやプロトコルを取得することはありません。

ACI ファブリック セキュリティ モデルは、これらの ACL の問題に処理します。ACI ファブリック セキュリティ モデルは、管理者の意図を直接表します。管理者は、連絡先、フィルタ、およびラベルの管理対象オブジェクトを使用してエンドポイントのグループがどのように通信するかを指定します。これらの管理対象オブジェクトは、ネットワークのトポロジに関連していません。なぜなら、それらは特定のインターフェイスに適用されないためです。それらは、エンドポイントのこれらのグループの接続場所に関係なく、ネットワークが強要しなければならない簡易なルールです。このトポロジの独立性は、これらの管理対象オブジェクトが特定の境界ポイントとしてだけではなくデータセンター全体にわたって容易に配置して再利用できることを意味します。

ACIファブリック セキュリティ モデルは、エンドポイントのグループ化コンストラクトを直接使用するため、サーバのグループが相互に通信できるようにするための概念はシンプルです。1 つのルールにより、任意の数の送信元が同様に任意の数の宛先と通信することを可能にできます。このような簡略化により、そのスケールと保守性が大幅に向上します。つまり、データセンター全体でより簡単に使用できることにもつながります。

セキュリティ ポリシー仕様を含むコントラクト

ACIセキュリティ モデルでは、コントラクトにEPG間の通信を管理するポリシーが含まれます。コントラクトは通信内容を指定し、EPG は通信の送信元と宛先を指定します。コントラクトは次のように EPG をリンクします。

EPG 1 ---------- コントラクト ---------- EPG 2

コントラクトで許可されていれば、EPG 1 のエンドポイントは EPG 2 のエンドポイントと通信でき、またその逆も可能です。このポリシーの構造には非常に柔軟性があります。たとえば、EPG 1 と EPG 2 間には多くのコントラクトが存在でき、1 つのコントラクトを使用する EPG が 3 つ以上存在でき、コントラクトは複数の EPG のセットで再利用できます。

またEPGとコントラクトの関係には方向性があります。EPGはコントラクトを提供または消費できます。コントラクトを提供するEPGは通常、一連のクライアントデバイスにサービスを提供す

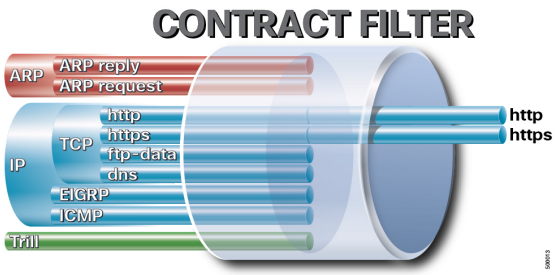
る一連のエンドポイントです。そのサービスによって使用されるプロトコルはコントラクトで定義されます。コントラクトを消費する EPG は通常、そのサービスのクライアントである一連のエンドポイントです。クライアントエンドポイント（コンシューマ）がサーバエンドポイント（プロバイダー）に接続しようとすると、コントラクトはその接続が許可されるかどうかを確認します。特に指定のない限り、そのコントラクトは、サーバがクライアントへの接続を開始することを許可しません。ただし、EPG 間の別のコントラクトが、その方向の接続を簡単に許可する場合があります。

この提供/消費の関係は通常、EPG とコントラクト間を矢印を使って図で表されます。次に示す矢印の方向に注目してください。

EPG 1 <------- 消費 -------- コントラクト <------- 提供 -------- EPG 2

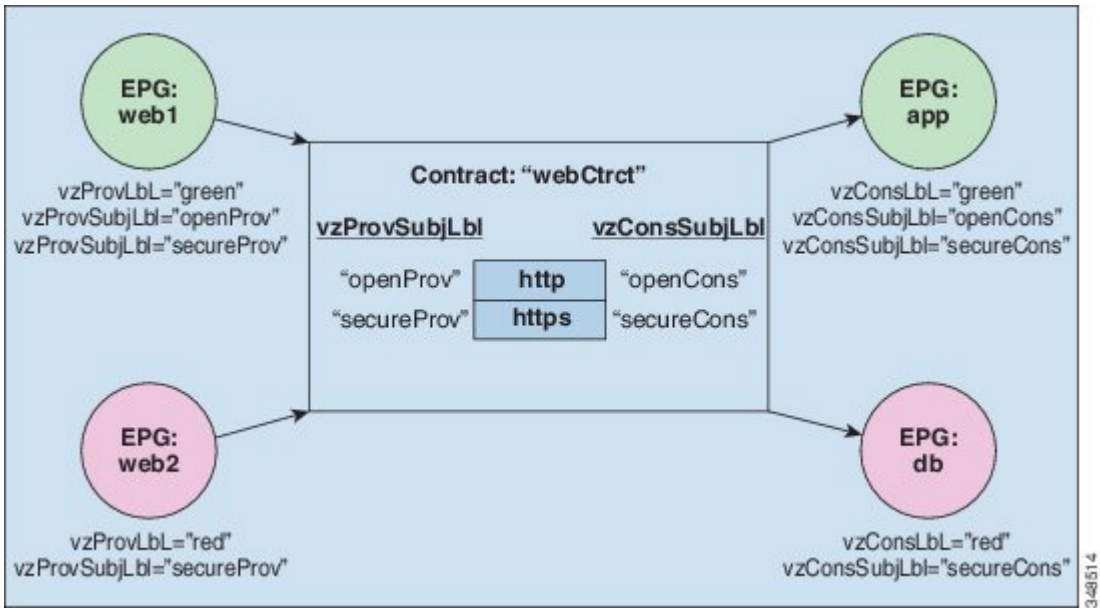
コントラクトは階層的に構築されます。1 つ以上のサブジェクトで構成され、各サブジェクトには 1 つ以上のフィルタが含まれ、各フィルタは 1 つ以上のプロトコルを定義できます。

図 *19*：コントラクト フィルタ



次の図は、コントラクトが EPG の通信をどのように管理するかを示します。

図 *20*： *EPG/EPG* 通信を決定するコントラクト



たとえば、TCP ポート 80 とポート 8080 を指定する HTTP と呼ばれるフィルタと、TCP ポート 443 を指定する HTTPS と呼ばれる別のフィルタを定義できます。その後、2 セットのサブジェクトを持つ webCtrct と呼ばれるコントラクトを作成できます。openProv と openCons are は HTTP フィルタが含まれるサブジェクトです。secureProv と secureCons は HTTPS フィルタが含まれるサブジェクトです。この webCtrct コントラクトは、Web サービスを提供する EPG とそのサービスを消費するエンドポイントを含む EPG 間のセキュアな Web トラフィックと非セキュアな Web トラフィックの両方を可能にするために使用できます。

これらの同じ構造は、仮想マシンのハイパーバイザを管理するポリシーにも適用されます。EPG が Virtual Machine Manager（VMM）のドメイン内に配置されると、APIC は EPG に関連付けられたすべてのポリシーを VMM ドメインに接続するインターフェイスを持つリーフ スイッチにダウンロードします。VMM ドメインの完全な説明については、『ACI の基本』マニュアルの「Virtual Machine Manager のドメイン」の章を参照してください。このポリシーが作成されると、APIC は EPG のエンドポイントへの接続を可能にするスイッチを指定する VMM ドメインにそれをプッシュ（あらかじめ入力）します。VMM ドメインは、EPG 内のエンドポイントが接続できるスイッチとポートのセットを定義します。エンドポイントがオンラインになると、適切な EPG に関連付けられます。パケットが送信されると、送信元 EPG および宛先 EPG がパケットから取得され、対応するコントラクトで定義されたポリシーでパケットが許可されたかどうかが確認されます。許可された場合は、パケットが転送されます。許可されない場合は、パケットはドロップされます。

コントラクトは、許可や拒否よりも複雑なアクションも許可します。コントラクトは、所定のサブジェクトに一致するトラフィックをサービスにリダイレクトしたり、コピーしたり、その QoS レベルを変更したりできることを指定可能です。具象モデルでアクセス ポリシーをあらかじめ入力すると、APIC がオフラインまたはアクセスできない場合でも、エンドポイントは移動でき、新しいエンドポイントをオンラインにでき、通信を行うことができます。APIC は、ネットワークの単一の障害発生時点から除外されます。ACI ファブリックにパケットが入力されると同時に、セキュリティ ポリシーがスイッチで実行している具象モデルによって適用されます。

セキュリティ ポリシーの適用

トラフィックは前面パネルのインターフェイスからリーフ スイッチに入り、パケットは送信元 EPG の EPG でマーキングされます。リーフ スイッチはその後、テナント エリア内のパケットの宛先 IP アドレスでフォワーディング ルックアップを実行します。ヒットすると、次のシナリオのいずれかが発生する可能性があります。

1. ユニキャスト（/32）ヒットでは、宛先エンドポイントの EPG と宛先エンドポイントが存在するローカル インターフェイスまたはリモート リーフ スイッチの VTEP IP アドレスが提供されます。
2. サブネット プレフィクス（/32 以外）のユニキャスト ヒットでは、宛先サブネット プレフィクスの EPG と宛先サブネット プレフィクスが存在するローカル インターフェイスまたはリモート リーフ スイッチの VTEP IP アドレスが提供されます。
3. マルチキャスト ヒットでは、ファブリック全体の VXLAN カプセル化とマルチキャスト グループの EPG で使用するローカル レシーバのローカル インターフェイスと外側の宛先 IP アドレスが提供されます。

（注） マルチキャストと外部ルータのサブネットは、入力リーフ スイッチでのヒットを常にもたらします。セキュリティ ポリシーの適用は、宛先 EPG が入力リーフ スイッチによって認識されるとすぐに発生します。

転送テーブルの誤りにより、パケットがスパイン スイッチの転送プロキシに送信されます。転送プロキシはその後、転送テーブル検索を実行します。これが誤りである場合、パケットはドロップされます。これがヒットの場合、パケットは宛先エンドポイントを含む出力リーフ スイッチに送信されます。出力リーフ スイッチが宛先の EPG を認識するため、セキュリティ ポリシーの適用が実行されます。出力リーフ スイッチは、パケット送信元の EPG を認識する必要があります。ファブリック ヘッダーは、入力リーフ スイッチから出力リーフ スイッチに EPG を伝送するため、このプロセスをイネーブルにします。スパイン スイッチは、転送プロキシ機能を実行するときに、パケット内の元の EPG を保存します。

出力リーフ スイッチでは、送信元 IP アドレス、送信元 VTEP、および送信元 EPG 情報は、学習によってローカルの転送テーブルに保存されます。ほとんどのフローが双方向であるため、応答パケットがフローの両側で転送テーブルに入力し、トラフィックが両方向で入力フィルタリングされます。

マルチキャストおよび EPG セキュリティ

マルチキャスト トラフィックでは、興味深い問題が起こります。ユニキャスト トラフィックでは、宛先 EPG はパケットの宛先の検査からはっきり知られています。ただし、マルチキャスト トラフィックでは、宛先は抽象的なエンティティ、マルチキャスト グループです。パケットの送信元はマルチキャスト アドレスではないため、送信元 EPG は以前のユニキャストの例と同様に決定されます。宛先グループの起源はマルチキャストが異なる場所です。

マルチキャスト グループが、ネットワーク トポロジから若干独立しているので、グループ バインディングへの (S, G) および (*, G) の静的設定は受け入れ可能です。マルチキャスト グループが転送テーブルにある場合、マルチキャスト グループに対応する EPG は、転送テーブルにも配置されます。

（注） このマニュアルでは、マルチキャスト グループとしてマルチキャスト ストリームを参照します。

リーフ スイッチは、マルチキャスト ストリームに対応するグループを常に宛先 EPG と見なし、送信元 EPG と見なすことはありません。前述のアクセス コントロール マトリクスでは、マルチキャスト EPG が送信元の場合は行の内容は無効です。トラフィックは、マルチキャスト ストリームの送信元またはマルチキャスト ストリームに加わりたい宛先からマルチキャスト ストリームに送信されます。マルチキャスト ストリームが転送テーブルにある必要があり、ストリーム内に階層型アドレッシングがないため、マルチキャスト トラフィックは、入力ファブリックの端でアクセスが制御されます。その結果、IPv4 マルチキャストは入力フィルタリングとして常に適用されます。

マルチキャストストリームの受信側は、トラフィックを受信する前にマルチキャストストリームに最初に加わる必要があります。IGMP Join 要求を送信すると、マルチキャスト レシーバは実際に IGMP パケットの送信元になります。宛先はマルチキャスト グループとして定義され、宛先 EPG は転送テーブルから取得されます。ルータが IGMP Join 要求を受信する入力点で、アクセス制御が適用されます。Join要求が拒否された場合、レシーバはその特定のマルチキャストストリームからトラフィックを受信しません。

マルチキャストEPGへのポリシーの適用は、前述のようにコントラクトのルールに従ってリーフスイッチにより入力時に発生します。また、EPGバインディングに対するマルチキャスト グループは、APIC によって特定のテナント（VRF）を含むすべてのリーフ スイッチにプッシュされます。

タブー

セキュリティを確保する通常のプロセスも適用されますが、ACI ポリシー モデルは、どのようなセキュリティ プラクティスが採用されても完全性を確保するのに役立ちます。ACI ポリシー モデルのアプローチでは、すべての通信がこれらの条件に準拠する必要があります。

- 通信は、モデルの管理対象オブジェクトであるコントラクトに基づいてのみ許可されます。コントラクトがなければ、EPG 間通信はデフォルトでディセーブルになります。
- ハードウェアへのダイレクト アクセスはなく、すべてのインタラクションはポリシー モデルを通じて管理されます。

タブーは、ネットワーク管理者がトラフィックの特定のクラスを拒否するために使用できるモデル内の特別なコントラクト管理対象オブジェクトです。タブーコントラクトは、パターンに一致するトラフィック（EPG、フィルタに一致する特定の EPG など）をドロップするために使用できます。タブーコントラクトのルールはハードウェアで通常のコントラクトのルールが適用される前の場合に適用されます。

タブー

第 5 章

ファブリック プロビジョニング

この章の内容は、次のとおりです。

ファブリック プロビジョニング

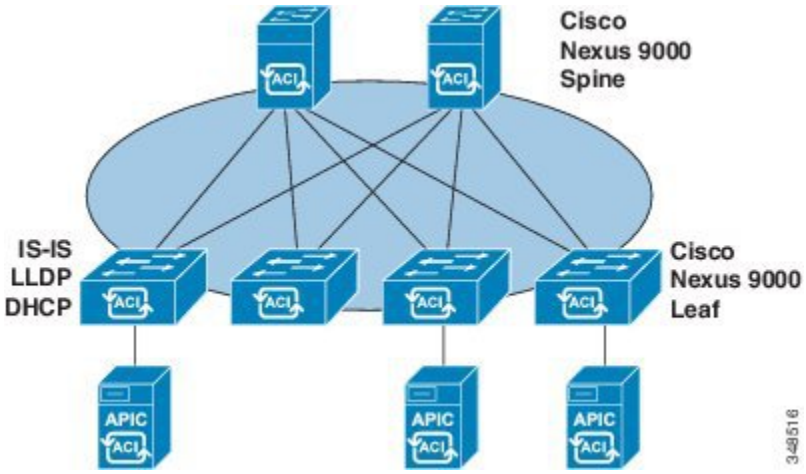
Ciscoアプリケーションセントリック インフラストラクチャ（ACI）の自動化とセルフプロビジョニングにより、従来のスイッチング インフラストラクチャに勝るこれらの操作上のメリットがもたらされます。

- クラスタ化され論理的に一元化されたが物理的に分散されている APIC では、ファブリック全体にポリシー、ブートストラップおよびイメージ管理が提供されます。
- APIC 起動トポロジの自動検出、自動設定、およびインフラストラクチャ アドレッシングでは、次の業界標準のプロトコルが使用されます。Intermediate System-to-Intermediate System（IS-IS）、リンク層検出プロトコル（LLDP）、ダイナミック ホスト コンフィギュレーション プロトコル（DHCP）。
- APIC では、シンプルで自動化されたポリシーベースのプロビジョニングとアップグレードのプロセス、および自動イメージ管理が提供されます。
- APIC では、スケーラブルな設定管理が提供されます。ACI のデータセンターは非常に規模が大きい場合があるため、スイッチまたはインターフェイスを個別に設定すると、スクリプトを使用しても十分に拡張しません。APICポッド、コントローラ、スイッチ、モジュール、およびインターフェイス セレクタ（すべて、範囲、特定のインスタンス）により、ファブリック全体の対称設定が可能になります。対称設定を適用するには、管理者がインターフェイス コンフィギュレーションを単一のポリシー グループに関連付けるスイッチ プロファイルを定義します。すると、個別の設定を必要とせずに、そのプロファイル内のすべてのインターフェイスに設定が迅速に導入されます。

スタートアップ検出と設定

クラスタ化された APIC コントローラでは、ファブリックに DHCP、ブートストラップ コンフィギュレーションおよびイメージ管理が提供され、自動化された起動およびアップグレードが可能になります。次の図は、スタートアップ検出を示します。

図 *21*：スタートアップ検出の設定



Cisco Nexus ACI ファブリック ソフトウェアは ISO イメージとしてバンドルされ、Cisco Integrated Management Controller（CIMC）上の KVM インターフェイスを通じて Cisco APIC サーバにインストールできます。Cisco Nexus ACI Software ISO には、Cisco APIC イメージ、リーフ ノードのファームウェア イメージ、スパイン ノードのファームウェア イメージ、デフォルトのファブリック インフラストラクチャ ポリシーおよび操作に必要なプロトコルが含まれます。

ACI ファブリックのブートストラップ シーケンスは、すべてのスイッチで出荷時にインストールされたイメージによってファブリックが起動されると開始されます。ACI ファームウェアと APIC を実行する Cisco Nexus 9000 シリーズ スイッチは、ブート プロセスに予約済みのオーバーレイを使用します。このインフラストラクチャ スペースはスイッチ上でハードコードされています。APIC はデフォルトのオーバーレイを通じてリーフに接続できます。または、ローカルで有効な ID を使うことができます。

ACI ファブリックはインフラストラクチャ スペースを使用します。インフラストラクチャ スペースはファブリック内でセキュアに隔離され、ここですべてのトポロジディスカバリ、ファブリック管理、インフラストラクチャ アドレッシングが行われます。ファブリック内の ACI ファブリック管理コミュニケーションは、内部のプライベート IP アドレスを通じてインフラストラクチャ スペース内で行われます。このアドレッシング方式によって、APIC はクラスタ内のファブリック ノードおよび他の Cisco APIC コントローラとの通信を行えます。APIC は、Link Layer Discovery Protocol（LLDP）ベースの検出プロセスを使用してクラスタ内の他の Cisco APIC コントローラの IP アドレスとノード情報を検出します。

次に、APIC クラスタ検出プロセスについて説明します。

- Cisco ACI の各 APIC は、内部のプライベート IP アドレスを使用してクラスタ内の ACI ノードおよび他の APIC と通信します。APIC は、LLDP ベースの検出プロセスを通じてクラスタ内の他の APIC コントローラの IP アドレスを検出します。
- APIC は、APIC ID から APIC IP アドレスと APIC の Universally Unique Identifier（UUID）にマッピングを提供するアプライアンス ベクトル（AV）を維持します。最初に、各 APIC がローカルの IP アドレスで満たされた AV から開始し、他のすべての APIC スロットが不明としてマークされます。
- スイッチの再起動後、リーフのポリシー要素（PE）が APIC からその AV を取得します。スイッチはその後、この AV をすべてのネイバーにアドバタイズし、ローカル AV とネイバーの AV 間の不一致をローカル AV のすべての APIC にレポートします。

このプロセスを使用して、APIC はスイッチを介して ACI の他の APIC コントローラについて学習します。クラスタ内のこれらの新しく検出された APIC コントローラを検証した後、APIC コントローラはローカル AV を更新して、スイッチを新しい AV でプログラミングします。その後、スイッチはこの新しい AV のアドバタイズを開始します。このプロセスは、すべてのスイッチが同一の AV を持ち、すべての APIC コントローラが他のすべての APIC コントローラの IP アドレスを認識するまで継続します。

（注） クラスタへの変更を開始する前に、必ずその状態を確認してください。クラスタに対して計画した変更を実行するときは、クラスタ内のすべてのコントローラが正常である必要があります。クラスタ内の 1 つ以上の APIC コントローラが正常でない場合は、クラスタの変更を進める前にその状況を修復してください。また、APIC に追加されたクラスタ コントローラが APIC クラスタ内の他のコントローラと同じファームウェア バージョンを実行しているか確認してください。APIC クラスタ変更の正常な完了を確保するために参照するガイドラインについては、記事『KB: Cisco ACI APIC Cluster Management』を参照してください。

ACI ファブリックは、APIC に直接接続しているリーフ ノードから順にカスケード式に起動されます。LLDP およびコントロール プレーン IS-IS コンバージェンスは、このブート プロセスと並行して行われます。ACI ファブリックは LLDP および DHCP ベースのファブリック検出機能を使用して、ファブリック スイッチ ノードの検出、インフラストラクチャの VXLAN トンネル エンドポイント（VTEP）アドレスの割り当て、スイッチへのファームウェアのインストールを自動的に行います。この自動プロセスの前に、Cisco APIC コントローラ上で最小限のブートストラップ設定を行う必要があります。APIC コントローラが接続され、その IP アドレスが割り当てられた後で、任意の APIC コントローラのアドレスを Web ブラウザに入力すると、APIC GUI にアクセスできるようになります。APIC GUI は、HTML5 を実行します。Java をローカルにインストールする必要はなくなります。

ファブリック インベントリ

ポリシー モデルには、すべてのノードおよびインターフェイスを含むファブリックの完全なリアルタイム インベントリが含まれます。このインベントリ機能により、プロビジョニング、トラブルシューティング、監査、およびモニタリングを自動化できます。

Cisco ACI のファブリック スイッチの場合は、ファブリック メンバーシップのノード インベントリに、ノードID、シリアル番号および名前を識別するポリシーが含まれます。サードパーティのノードは、管理対象外のファブリック ノードとして記録されます。Cisco ACI のスイッチは自動的に検出することができ、またはポリシー情報をインポートできます。ポリシー モデルは、ファブリック メンバー ノードのステータス情報も保持します。

ノードのステータス	状態
Unknown	ポリシーが存在しません。すべてのノードにはポリシーが必要で、ポリシーがない場合はメンバ ノードのステータスは不明となります。
Discovering	ノードが検出され、ホストトラフィックを待機していることを示す一時状態。
Undiscovered	ノードにはポリシーがありますが、ファブリックで提示されたことはありません。
Unsupported	ノードは Cisco のスイッチですが、サポートされていません。たとえば、ファームウェアのバージョンが ACI のファブリックと互換性がありません。
Decommissioned	ノードはポリシーを持っており、検出されましたが、ユーザがこれを無効にしました。ノードを再びイネーブルにすることができます。 （注） リーフをデコミッションしているときにワイプ オプションを指定すると、APIC は、リーフ スイッチおよび APIC の両方からすべてのリーフ スイッチ設定を削除しようとします。リーフ スイッチが到達可能でない場合には、APIC だけが削除されます。この場合、ユーザはリセットによって手動でリーフ スイッチをワイプする必要があります。
Inactive	ノードが到達不能です。検出されましたが、現在アクセスできません。たとえば、電源がオフになっているか、ケーブルが切断されている可能性があります。
Active	ノードはファブリックのアクティブ メンバです。

無効のインターフェイスは、管理者によってブラックリスト化されたものや、APIC が異常を検出するため取り除かれたものである可能性があります。リンク ステート異常の例を次に示します。

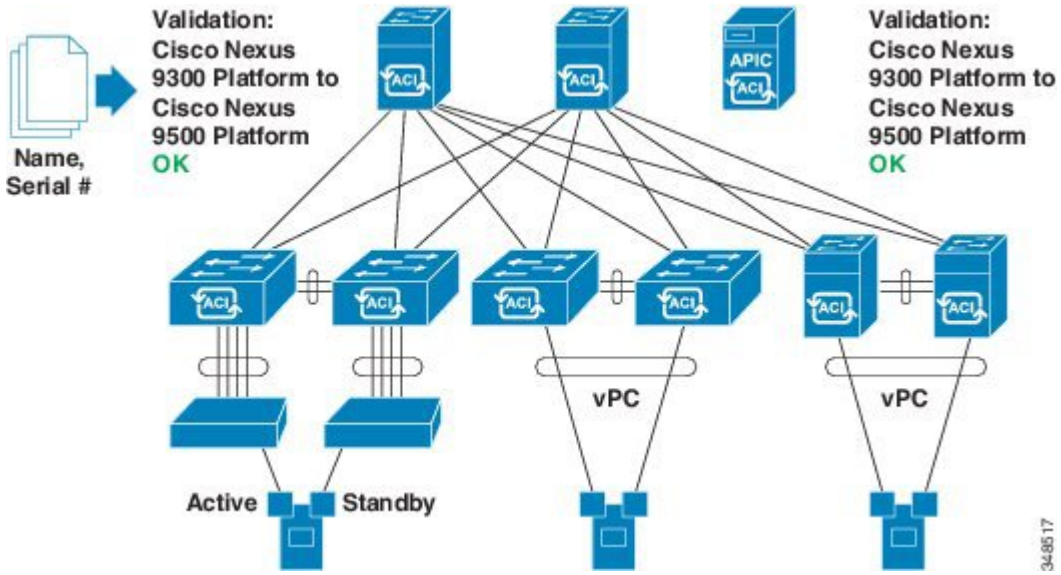
- スパインに接続されているスパイン、リーフに接続されているリーフ、リーフ アクセス ポートに接続されているスパイン、非ACI ノードに接続されているスパイン、または非ACI デバイスに接続されているリーフ ファブリック ポートなどの配線の不一致。
- ファブリック名の不一致。ファブリック名は各ACI ノードに保存されます。工場出荷時のデフォルト状態にリセットされることなくノードが別のファブリックに移動される場合、ファブリック名が保持されます。
- UUID の不一致によって APIC がノードをディセーブルにします。

（注） 管理者がAPICを使用してスパインのすべてのリーフ ノードをディセーブルにすると、スパインへのアクセスを回復するためにスパインの再起動が必要です。

プロビジョニング

APIC プロビジョニング方式により、適切な接続を通じて ACI ファブリックが自動的に起動します。次の図は、ファブリックのプロビジョニングを示します。

図 *22*：ファブリック プロビジョニング



Link Layer Discovery Protocol（LLDP）ディスカバリが隣接するすべての接続を動的に学習した後、これらの接続は緩やかなルールに照らし合わせて検証できます。たとえば、「LEAF can connect to only SPINE-L1-*」または「SPINE-L1-* can connect to SPINE-L2-* or LEAF」などと指定できます。規則への不一致が見つかった場合、リーフが別のリーフに接続したりスパインが別のスパインに接続したりすることは許可されないため、エラーが発生して接続がブロックされます。また、接続に注意が必要であることを示すアラームが作成されます。Cisco ACI ファブリックの管理者は、テキスト ファイルからすべてのファブリック ノードの名前とシリアル番号を APIC にインポートすることができ、または APIC GUI、コマンドライン インターフェイス（CLI）または API を使用してシリアル番号を自動的に検出し、名前をノードに割り当てることをファブリックに許可できます。APIC は SNMP によって検出可能です。これには、次の asysobjectId があります。

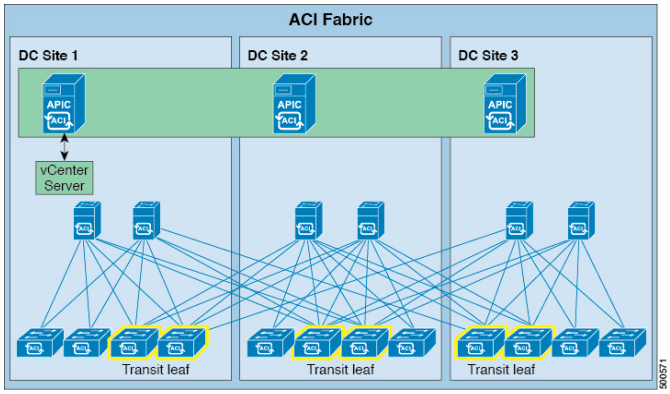
```
ciscoACIController OBJECT IDENTIFIER ::= { ciscoProducts 2238 }
```

ストレッチ ACI ファブリックの設計の概要

ストレッチ ACI ファブリックは、一部メッシュ構造の設計になっており、複数の場所に分散している ACI リーフ/スパイン スイッチを接続します。通常の ACI ファブリック実装は単一サイトで行われ、フル メッシュ設計によってファブリック内の各リーフ スイッチを各スパイン スイッチに接続します。これにより最大のスループットおよびコンバージェンスが得られます。マルチサイトのシナリオでは、フル メッシュ接続が不可能な場合や、膨大なコストがかかる場合があります。サイト、建物、または会議室が複数ある場合、距離の広がりが大きいために、十分なファイバ接続が得られなかったり、サイト間の各スパイン スイッチ/リーフ スイッチ接続にコストがかかりすぎてしまう可能性があります。

次の図は、ストレッチ ファブリックのトポロジを示しています。

図 *23* : *ACI* ストレッチ ファブリック トポロジ



ストレッチ ファブリックは単一 ACI ファブリックです。サイトは、管理ドメイン 1 つと、可用性ゾーン 1 つです。管理者は、サイトを 1 つのエンティティとして管理できます。任意の APIC コントローラ ノードで行った設定変更が、サイト全体のデバイスに適用されます。ストレッチ ACI ファブリックは、VM ライブ マイグレーション機能をサイト全体で保持します。現在、ストレッチ ファブリック設計は 3 つのサイトで検証されています。

ストレッチ ACI ファブリックに関するドキュメント

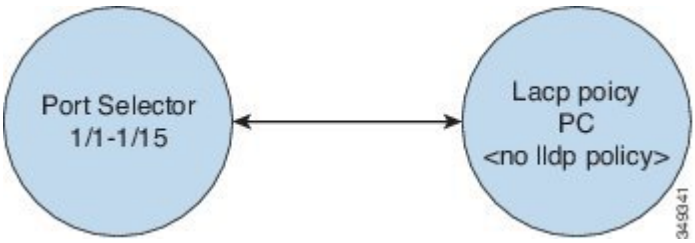
テクニカル ノート『KB Stretched ACI Fabric Design Overview』では、ACI ファブリックを複数のサイトで運用するためのトラフィック フロー、APIC クラスタ冗長性、運用上の考慮事項に関する設計ガイドラインが提供されています。

デフォルト ポリシー

APICデフォルト ポリシー値の初期値は、スイッチにロードされる具象モデルから取得されます。ファブリックの管理者は、デフォルト ポリシーを変更できます。デフォルト ポリシーは、次の複数の目的に使用されます。

1. ファブリックの管理者がモデル内のデフォルト値を上書きできます。
2. 管理者が明示ポリシーを提供しない場合、APIC はデフォルトのポリシーを適用します。管理者はデフォルトのポリシーを作成でき、管理者が明示ポリシーを提供しない限り、APIC はそのポリシーを使用します。

図 *24*：デフォルト ポリシー



たとえば、管理者が行うアクションまたは行わないアクションに応じて、APIC は次を実行します。

- 管理者が選択したポートに対して LLDP ポリシーを指定しないため、APIC はポート セレクタに指定されたポートに対しデフォルトの LLDP インターフェイス ポリシーを適用します。
- 管理者がポート セレクタからポートを削除すると、APIC はそのポートにデフォルト ポリシーを適用します。この例では、管理者がポート 1/15 をポート セレクタから削除すると、そのポートはポート チャネルの一部ではなくなり、APIC はそのポートにすべてのデフォルト ポリシーを適用します。

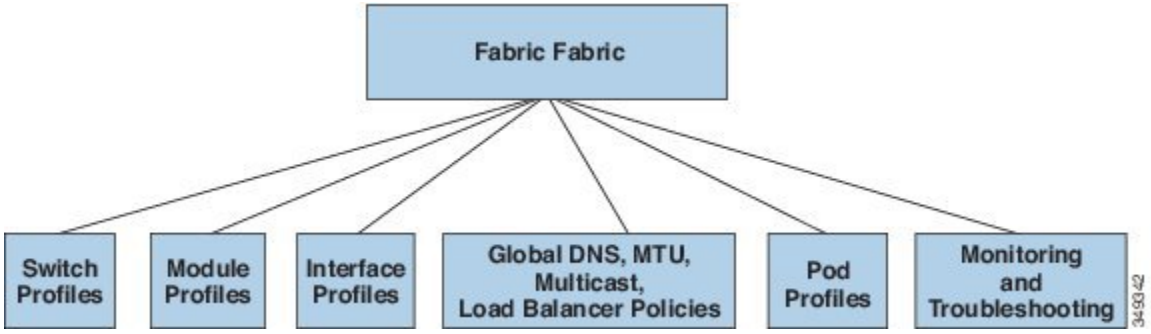
ACI ファブリックをアップグレードした場合、デフォルト値が新しいリリースで変更されても既存のポリシーのデフォルト値が保持されます。ノードが APIC に初めて接続されると、ノードはそれ自体をすべてのデフォルト ポリシーをノードにプッシュする APIC に登録します。デフォルト ポリシーでのすべての変更がノードにプッシュされます。

ファブリック ポリシーの概要

ファブリック ポリシーは、内部のファブリック インターフェイスの操作を管理し、スパインおよびリーフ スイッチを接続するさまざまな機能、プロトコル、およびインターフェイスの設定を可能にします。ファブリックの管理者権限を持つ管理者は、要件に応じて新しいファブリック ポリシーを作成できます。APIC では、管理者はファブリック ポリシーを適用するポッド、スイッチ

およびインターフェイスを選択できます。次の図は、ファブリックのポリシー モデルの概要を示します。

図 *25*：ファブリック ポリシーの概要



ファブリック ポリシーは、次のカテゴリにグループ化されます。

- スイッチ プロファイルは、設定するスイッチとスイッチの設定ポリシーを指定します。
- モジュール プロファイルは、設定するスパイン スイッチ モジュールとスパイン スイッチの設定ポリシーを指定します。
- インターフェイスプロファイルは、設定するファブリック インターフェイスとインターフェイスの設定ポリシーを指定します。
- グローバル ポリシーは、DNS、ファブリック MTU のデフォルト、マルチキャスト ツリー、およびファブリック全体で使用するロード バランサの設定を指定します。
- ポッド プロファイルは、日時、SNMP、カウンシル オブ オラクル プロトコル（COOP）、IS-IS、およびボーダー ゲートウェイ プロトコル（BGP）のルート リフレクタ ポリシーを指定します。
- モニタリングおよびトラブルシューティングのポリシーは、モニタする対象、しきい値、障害とログの処理方法、および診断方法を指定します。

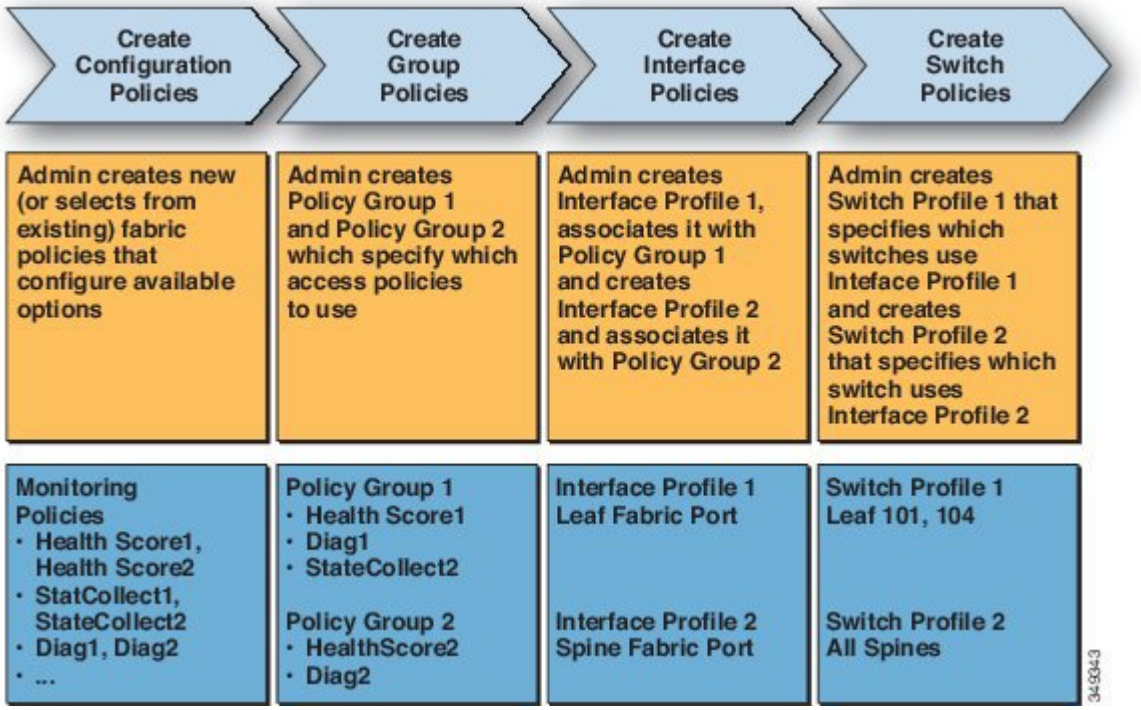
ファブリック ポリシーの設定

ファブリック ポリシーは、スパインおよびリーフ スイッチに接続するインターフェイスを設定します。ファブリック ポリシーは、モニタリング（統計情報収集および統計情報のエクスポート）、トラブルシューティング（オンデマンド診断と SPAN）、IS-IS、カウンシル オブ オラクル プロトコル（COOP）、SNMP、ボーダー ゲートウェイ プロトコル（BGP）のルート リフレクタ、DNS、またはネットワーク タイム プロトコル（NTP）などの機能を有効にできます。

ファブリック全体で設定を適用するには、管理者がポリシーの定義済みグループをスイッチ上のインターフェイスに単一段階で関連付けます。このようにして、ファブリック上の多数のインター

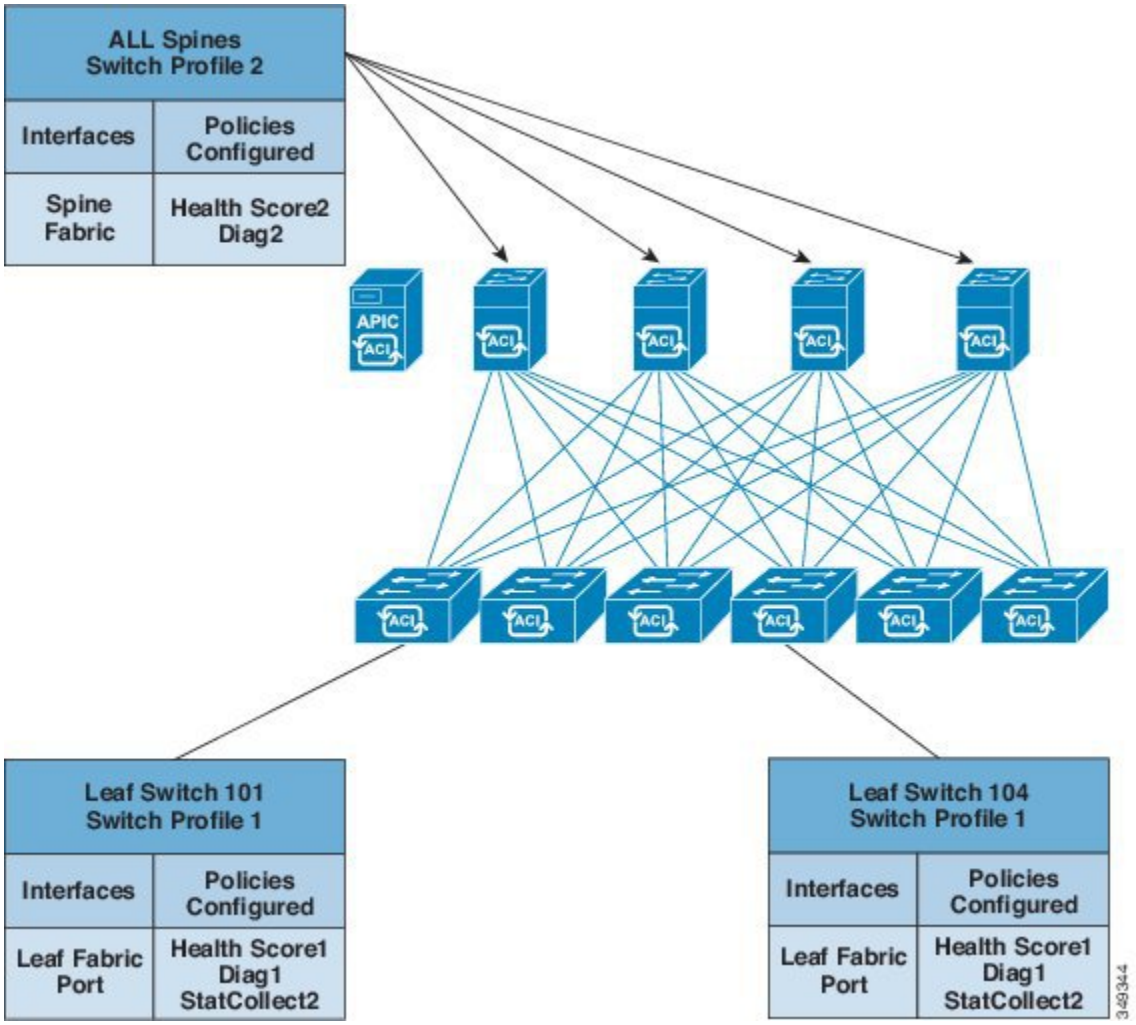
フェイスを一度に設定できます。1 個のポートを一度に設定することはスケーラブルではありません。次の図は、ACI ファブリックを設定するプロセスがどのように動作するかを示します。

図 *26*：ファブリック ポリシーの設定プロセス



次の図は、ACI ファブリックにスイッチ プロファイル 1 およびスイッチ プロファイル 2 を適用した結果を示します。

図 *27*：ファブリック スイッチ ポリシーのアプリケーション



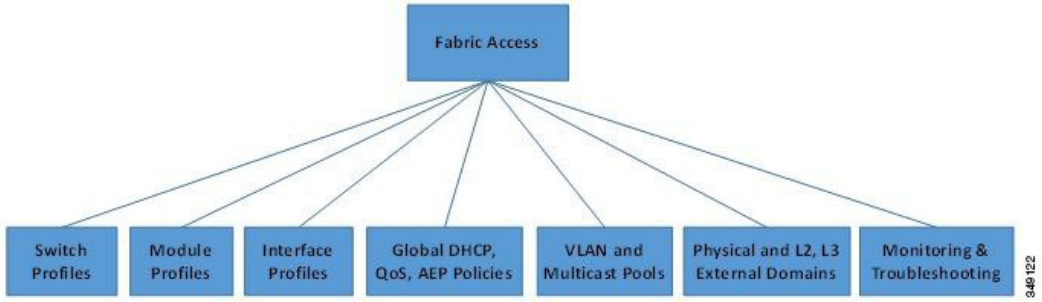
インフラストラクチャと範囲を組み合わせることにより、管理者はスケーラブルな方法でファブリック構成を管理することができます。これらの設定は、REST API、CLI、または GUI を使用して実行できます。GUI 内の [Quick Start Fabric Interface Configuration] ウィザードでは、そのようなポリシーの実行に必要な基盤となるオブジェクトが自動的に作成されます。

アクセス ポリシーの概要

アクセスポリシーは、仮想マシンコントローラおよびハイパーバイザなどのデバイスに接続する外向きインターフェイス、ホスト、ネットワーク接続ストレージ、ルータ、またはファブリックエクステンダ（FEX）インターフェイスを設定します。アクセス ポリシーにより、ポート チャネ

ルおよび仮想ポート チャネル、Link Layer Discovery Protocol（LLDP）、Cisco Discovery Protocol（CDP）、または Link Aggregation Control Protocol（LACP）などのプロトコル、および統計情報収集、監視、および診断などの機能の設定が可能になります。次の図は、アクセスポリシーモデルの概要を示します。

図 *28*：アクセス ポリシー モデルの概要



アクセス ポリシーは、次のカテゴリにグループ化されます。

- スイッチ プロファイルは、設定するスイッチとスイッチの設定ポリシーを指定します。
- モジュール プロファイルは、設定するリーフ スイッチのアクセス カードおよびアクセス モジュールとリーフ スイッチの設定ポリシーを指定します。
- インターフェイス プロファイルは、設定するアクセス インターフェイスとインターフェイスの設定ポリシーを指定します。
- グローバル ポリシーにより、ファブリック全体に使用できる DHCP、QoS、および接続可能アクセスエンティティ（AEP）のプロファイル機能の設定が可能になります。AEPプロファイルは、リーフ ポートの大規模セットでハイパーバイザ ポリシーを展開するためのテンプレートを提供し、仮想マシン管理（VMM）のドメインと物理ネットワーク インフラストラクチャを関連付けます。また、レイヤ 2 およびレイヤ 3 の外部ネットワークの接続にも必要となります。
- プールは、VLAN、VXLAN およびマルチキャスト アドレス プールを指定します。プールは、VMM などの複数のドメインおよびレイヤ 4 ～ レイヤ 7 のサービスで消費できる共有リソースです。プールは、トラフィックのカプセル化 ID の範囲を表します（たとえば、VLAN ID、VNID、マルチキャスト アドレスなど）。
- 物理および外部ドメイン ポリシーには、次のものが含まれます。
  - 外部ブリッジド ドメインのレイヤ 2 ドメイン プロファイルには、ファブリックに接続されたブリッジド レイヤ 2 ネットワークが使用するポートおよび VLAN の仕様が含まれます。
  - 外部ルーテッド ドメインのレイヤ 3 ドメイン プロファイルには、ファブリックに接続されたルーテッド レイヤ 3 ネットワークが使用するポートおよび VLAN の仕様が含まれます。

- 物理ドメインポリシーには、テナントまたはエンドポイントグループで使用されるポートや VLAN などの物理インフラストラクチャの仕様が含まれます。

- モニタリングおよびトラブルシューティングのポリシーは、モニタする対象、しきい値、障害とログの処理方法、および診断方法を指定します。

アクセス ポリシーの設定

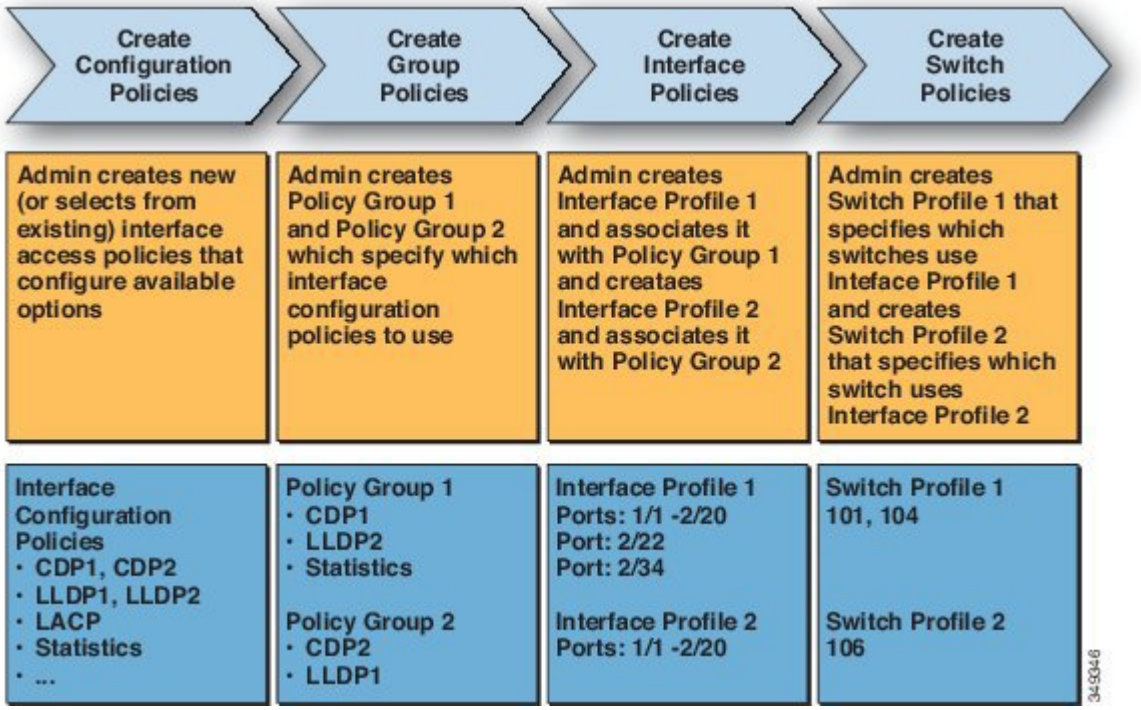
アクセス ポリシーは、スパイン スイッチに接続していない外向きインターフェイスを設定します。外向きインターフェイスは、仮想マシン コントローラなどの外部デバイス、ハイパーバイザ、ホスト、ルータ、またはファブリック エクステンダ（FEX）と接続します。アクセス ポリシーにより、管理者はポート チャネルおよび仮想ポート チャネル、LLDP、CDP、LACP などのプロトコル、モニタリングや診断などの機能を設定することができます。スイッチ インターフェイス、ポートチャネル、仮想ポートチャネル、およびインターフェイス速度の変更に関するサンプルの XML ポリシーを付録 C「アクセス ポリシーの例」に示します。

（注） テナント ネットワーク ポリシーがファブリックのアクセス ポリシーと別に設定される一方で、テナント ポリシーが依存する基盤となるアクセス ポリシーが整わないとテナント ポリシーはアクティブ化されません。

潜在的に多数のスイッチ間で設定を適用するためには、管理者は、単一のポリシー グループのインターフェイスコンフィギュレーションを関連付けるスイッチプロファイルを定義します。このようにして、ファブリック上の多数のインターフェイスを一度に設定できます。スイッチ プロ

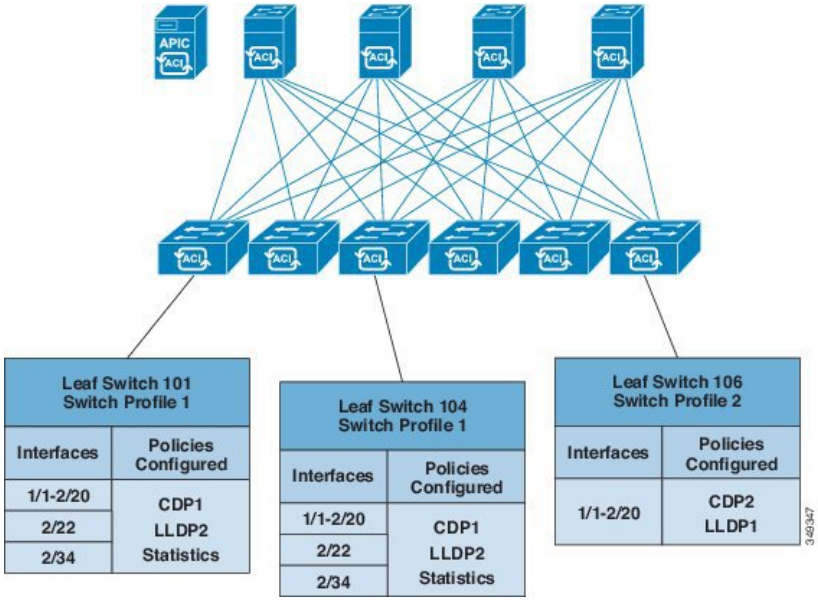
ファイルには、複数のスイッチに対する対称設定や一意の特殊用途設定を含めることができます。次の図は、ACI ファブリックへのアクセス設定のプロセスを示します。

図 *29*：アクセス ポリシーの設定プロセス



次の図は、ACI ファブリックにスイッチ プロファイル 1 およびスイッチ プロファイル 2 を適用した結果を示します。

図 *30*：アクセス スイッチ ポリシーの適用



インフラストラクチャと範囲を組み合わせることにより、管理者はスケーラブルな方法でファブリック構成を管理することができます。これらの設定は、REST API、CLI、または GUI を使用して実行できます。GUI 内の [Quick Start Interface]、[PC]、[VPC Configuration] ウィザードでは、そのようなポリシーの実行に必要な基盤となるオブジェクトが自動的に作成されます。

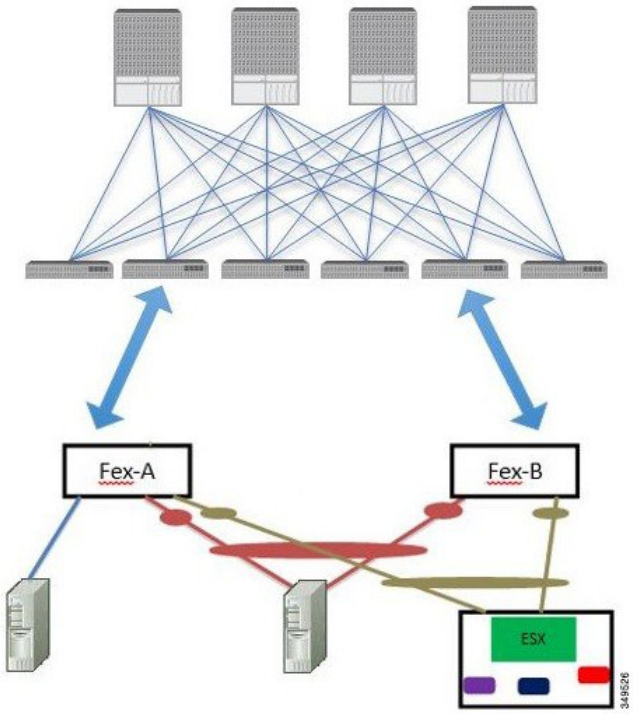
ポート チャネルと仮想ポート チャネル アクセス

アクセス ポリシーにより、管理者はポート チャネルおよび仮想ポート チャネルを設定することができます。スイッチ インターフェイス、ポート チャネル、仮想ポート チャネル、およびインターフェイス速度の変更に関するサンプルの XML ポリシーを複数のスイッチに適用される単一のポート チャネルの設定に示します。

FEX 仮想ポート チャネル

v1.1 リリースの時点で、ACI ファブリックは、Cisco ファブリック エクステンダ（FEX）サーバ側仮想ポート チャネル（VPC）（別名 FEX ストレート VPC）をサポートします。付録 D：FEX ポリシーの例に XML FEX ポリシーの例を掲載しています。

図 *31*：サポートされている *FEX VPC* トポロジ



サポートされている FEX VPC ポート チャネル トポロジは次のとおりです。

- FEX の背後にある VTEP および非 VTEP の両方のハイパーバイザ。
- ACI ファブリックに接続された 2 つの FEX に接続されている仮想スイッチ（AVS または VDS など）。（物理的な FEX ポートに直接接続されている VPC はサポートされません。VPC はポート チャネルでのみサポートされます）

（注） Mac に IP を通知するために GARP をプロトコルとして使用している場合、同じ FEX 上の異なるインターフェイスに変更をバインディングするには、ブリッジ ドメインのモードをフラッドに設定し、CPU への GARP トラップを無効にする必要があります。

アップリンク障害検出のためのポートトラッキングポリシー

アップリンク障害検出は、ファブリック アクセス グローバル ポート トラッキング ポリシーで有効化できます。ポート トラッキング ポリシーは、リーフ スイッチとスパイン スイッチ間のリンクの状態を監視します。有効なポート トラッキング ポリシーがトリガーされると、リーフ スイッチは、EPGによって導入されたスイッチ上のすべてのアクセス インターフェイスをダウンさせます。

（注） 拡張 GUI では、ポート トラッキングは [Fabric] > [Access Policies] > [Port Tracking] の下にあります。ベーシック GUI では、ポート トラッキングは [System] > [Port Tracking] の下にあります。

各リーフ スイッチから各スパイン スイッチへのアップリンク接続の許容数はリーフ スイッチのモデルによって異なり、6、8、または12となります。ポート トラッキング ポリシーは、ポリシーをトリガーするアップリンク接続の数と、指定のアップリンク数を超えた後にリーフ スイッチ アクセス ポートを復旧させる遅延タイマーを指定します。

ポート トラッキング ポリシーの動作の例を次に示します。

- 各リーフ スイッチからスパイン スイッチへのアクティブなアップリンク接続の数は、最大 6 つです。
- ポート トラッキング ポリシーは、ポリシーをトリガーする各リーフ スイッチのアクティブなアップリンク接続のしきい値を 2 に指定します。
- リーフ スイッチからスパイン スイッチへのアクティブなアップリンク接続数が 2 まで減少すると、ポート トラッキング ポリシーがトリガーされます。
- 各リーフ スイッチはそのアップリンク接続を監視し、ポリシーで指定されたしきい値に従ってポート トラッキング ポリシーをトリガーします。
- アップリンク接続が復旧すると、リーフ スイッチは遅延タイマーの時間が満了するのを待ち、その後、そのアクセス ポートを復旧させます。これにより、ファブリックには、リーフ スイッチ アクセス ポートでトラフィックが再開する前に再コンバージェンスできる時間が確保されます。大きなファブリックでは、遅延タイマーの時間を長めに設定することが必要な場合があります。

（注） このポリシーの設定には注意が必要です。ポート トラッキング設定において、ポート トラッキングをトリガーするアクティブなスパイン リンク数を過剰に大きく設定すると、すべてのリーフ スイッチ アクセス ポートがダウンします。

802.1p サービス クラスの保持

ACI ファブリックは、ファブリック内で 802.1p サービス クラス（CoS）を維持することができます。ファブリック グローバル QoS ポリシーの `dot1p-preserve` オプションを有効にすることで、ACI ファブリックを入力し中継するパケット 802.1p 値の保持が保証されます。次に示す 801.1p CoS 保持のガイドラインおよび制限事項に従ってください。

- 現在のリリースでは、VLAN ヘッダー内の 802.1p 値のみ保持可能です。DEI ビットは保持されません。
- 次の設定オプションを有効にすると、802.1P の保持は行われません。
  - QoS を設定するコントラクト。
  - ダイナミック パケットのプライオリティが有効。
  - 発信インターフェイスが FEX 上にある。
  - DSCP QoS ポリシーが VLAN EPG 上で設定され、パケットに IP ヘッダーがある。次の項目について、最内部～最外部の優先順位を付けて、DSCP マーキングをフィルタ レベルで設定できます。
    - コントラクト
    - サブジェクト
    - インターム
    - アウトターム

（注） コントラクトに vzAny を指定した場合、vzAny はそのコンテキストのすべての EPG の集合であり、EPG の固有の設定が適用できないため、外部 EPG DSCP の値は考慮されません。EPG 固有のターゲット DSCP 値が必要な場合は、外部 EPG で vzAny を使用しないでください。

  - 複数ポッドの QoS が有効。

スケジューラ

スケジュールにより、設定のインポート/エクスポートまたはテクニカルサポートの収集などの操作を 1 つ以上の指定した時間帯に発生させることができます。

スケジュールには、一連のタイム ウィンドウ（オカレンス）が含まれます。これらのウィンドウは、1 回だけ発生させるか、または毎週指定した日時に繰り返し発生させることができます。期間や実行するタスクの最大数などのウィンドウで定義されているオプションにより、スケジュール設定されたタスクの実行時期が決定されます。たとえば、最大時間長またはタスク数に達した

ため特定のメンテナンス時間帯に変更を展開できない場合、この展開は次のメンテナンス時間に持ち越されます。

各スケジュールは、APICが1つまたは複数のメンテナンス時間帯に入っているかどうか、定期的に確認します。入っている場合、スケジュールはメンテナンス ポリシーで指定された制限に対し適切な展開を実行します。

スケジュールには、スケジュールに関連付けられたメンテナンス時間帯を決定する 1 つ以上のオカレンスが含まれています。オカレンスは次のいずれかになります。

- [One-time Window]：一度だけ行うスケジュールを定義します。これらの時間帯は、その時間帯の最大時間長まで、または時間帯の中で実行可能なタスクの最大数に達するまで継続されます。
- [Recurring Window]：繰り返すスケジュールを定義します。この時間帯は、タスクの最大数に達するまで、または時間帯に指定された日の終わりに達するまで継続します。

ファームウェア アップグレード

APIC 上のポリシーは、ファームウェア アップグレード プロセスの次の項目を管理します。

- 使用するファームウェアのバージョン。
- シスコから APIC リポジトリへのファームウェア イメージのダウンロード。
- 互換性の適用。
- アップグレードするもの：
  - スイッチ
  - APIC
  - 互換性カタログ
- アップグレードを実行する時期。
- 障害の処理方法（再試行、一時停止、無視など）。

各ファームウェアイメージには、サポートされるタイプおよびスイッチモデルを識別する互換性カタログが含まれます。APICは、ファームウェアイメージ、スイッチタイプ、およびそのファームウェア イメージを使用することを許可されるモデルのカタログを保持しています。デフォルトの設定では、互換性カタログに適合しない場合、ファームウェアの更新が拒否されます。

イメージ管理を実行する APIC には、互換性カタログ、APIC コントローラのファームウェア イメージおよびスイッチ イメージのイメージ リポジトリがあります。管理者は、イメージ ソース ポリシーを作成することで外部 HTTP サーバまたは SCP サーバから新しいファームウェア イメージを APIC イメージ リポジトリにダウンロードできます。

APIC 上のファームウェア グループ ポリシーは、必要なファームウェア バージョンを定義します。

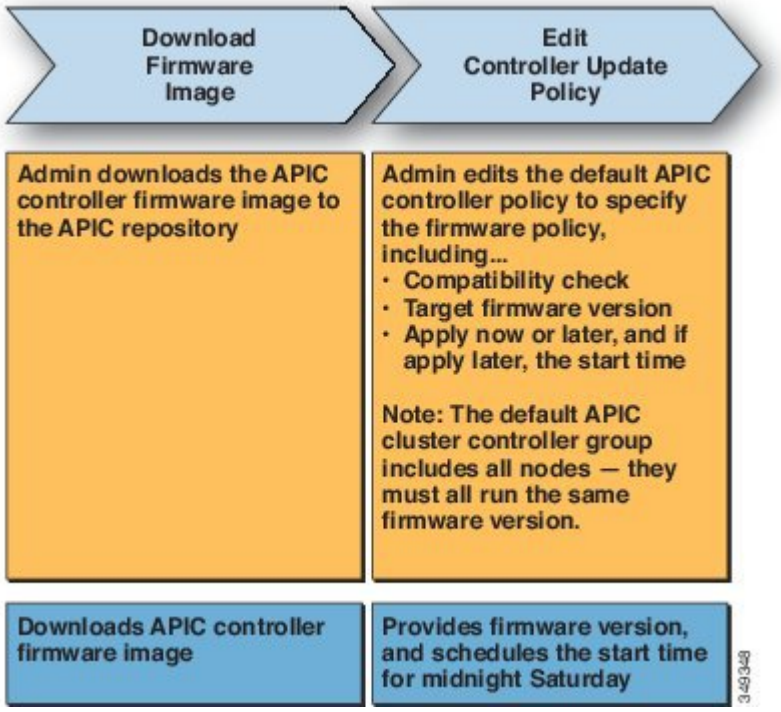
メンテナンス グループ ポリシーは、ファームウェアをアップグレードする時期、アップグレードするノード、および障害の処理方法を定義します。また、メンテナンス グループ ポリシーは、同時にアップグレードできるノードのグループを定義して、それらのメンテナンス グループをスケジュールに割り当てます。ノード グループ オプションには、すべてのリーフ ノード、すべてのスパイン ノード、またはファブリックの一部であるノードのセットが含まれます。

APIC コントローラのファームウェア アップグレード ポリシーは、クラスタ内のすべてのノードに常に適用されますが、アップグレードは常に一度に 1 つのノードに実行されます。APIC GUI により、ファームウェア アップグレードに関するリアルタイムのステータス情報が提供されます。

（注） 定期的アップグレードまたは1度だけのアップグレードのスケジュールに過去の日時が設定されている場合、スケジューラはただちにアップグレードをトリガーします。

次の図は、APIC クラスタ ノードのファームウェア アップグレードのプロセスを示します。

図 *32* : *APIC* クラスタ コントローラのファームウェア アップグレードのプロセス



APICは、次のようにこのコントローラのファームウェアアップグレードポリシーを適用します。

- 管理者がコントローラ更新ポリシーの開始時刻を土曜日の午前 0 時に設定したので、APIC は土曜日の午前 0 時にアップグレードを開始します。
- システムは、既存のファームウェアの互換性を確認し、新しいファームウェア イメージで提供される互換性カタログに従って、新しいバージョンにアップグレードします。

- アップグレードは、クラスタ内のすべてのノードがアップグレードされるまで、一度に 1 個のノードずつ行われます。

（注） APIC はノードの複製クラスタであるため、中断は最小限に抑えるべきです。管理者は、APIC のアップグレードのスケジューリングを検討する場合、システムの負荷を意識する必要があります。また、アップグレードがメンテナンス期間中に行われるよう計画する必要があります。

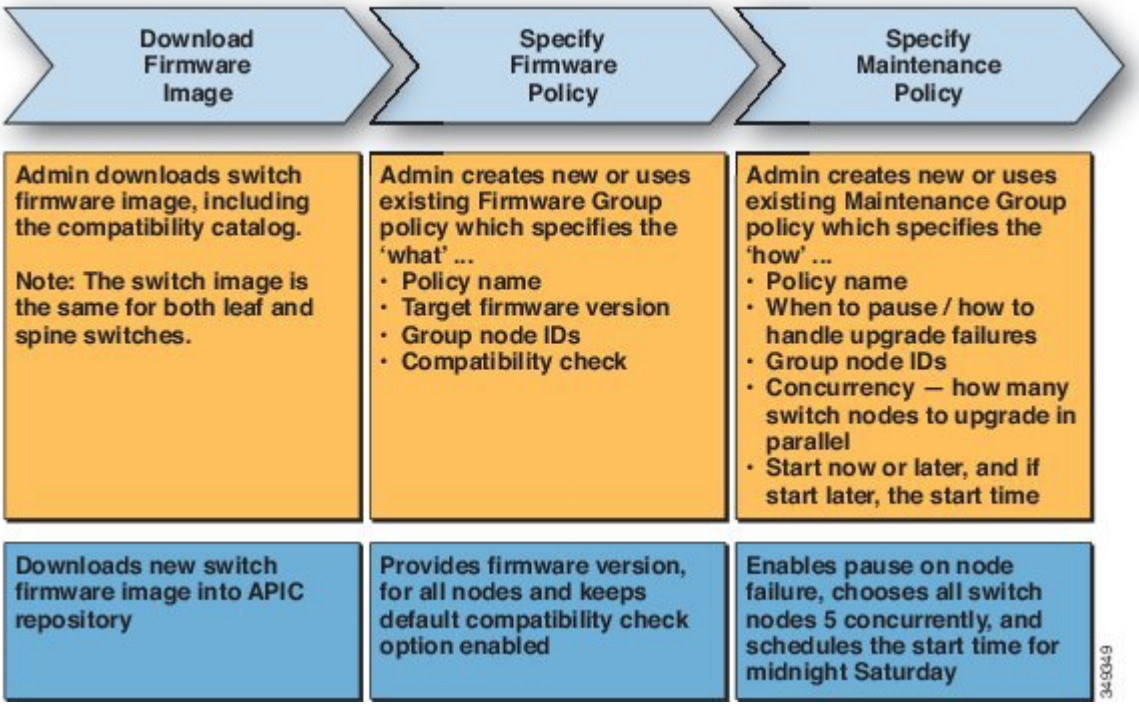
- APIC を含む ACI ファブリックは、アップグレードが進行中でも動作し続けます。

（注） ランダムにコントローラのアップグレード。各 APIC コントローラはアップグレードに約 10 分かかります。コントローラのイメージがアップグレードされた場合、クラスタからドロップし、クラスタ内の他 APIC のコントローラがまだ動作している間に、新しいバージョンで再起動します。コントローラが再起動すると、クラスタに再び参加します。その後、クラスタが収束し、次のコントローラのイメージのアップグレードを開始します。クラスタがすぐに収束せず、完全に適合しない場合は、クラスタが収束し完全に適合するまでアップグレードは待機します。この期間中、「Waiting for Cluster Convergence」メッセージが表示されます。

- コントローラ ノードのアップグレードが失敗すると、アップグレードが一時停止し、手動による介入が行われるまで待機します。

次の図は、すべての ACI ファブリック スイッチ ノードのファームウェアをアップグレードするプロセスがどのように動作するかを示します。

図 *33*：スイッチ ファームウェアのアップグレード プロセス



APIC は、次のようにこのスイッチ アップグレード ポリシーを適用します。

- 管理者がコントローラ更新ポリシーの開始時刻を土曜日の午前 0 時に設定したので、APIC は土曜日の午前 0 時にアップグレードを開始します。
- システムは、既存のファームウェアの互換性を確認し、新しいファームウェア イメージで提供される互換性カタログに従って、新しいバージョンにアップグレードします。
- 指定されたすべてのノードのアップグレードが終わるまで、一度に 5 個ずつのノードがアップグレードされます。

（注） ファームウェアのアップグレードにより、スイッチがリブートします。リブートにより数分間スイッチの操作が中断される場合があります。ファームウェアのアップグレードはメンテナンス期間中にスケジューリングしてください。

- スイッチ ノードのアップグレードが失敗すると、アップグレードが一時停止し、手動による介入が行われるまで待機します。

ファームウェア アップグレードの実行に関する詳細な手順については、『Cisco APIC Firmware Management Guide』を参照してください。

設定ゾーン

設定ゾーンは ACI ファブリックを複数のゾーンに分割します。これらのゾーンは、別々のタイミングで設定を変更をして更新することができます。これにより、障害のある設定がファブリック全体に導入されるリスクが限定され、トラフィックが中断したり、さらにはファブリックがダウンしたりする可能性が抑えられます。管理者は、あまり重要ではないゾーンに設定を導入した後に、適切であることを確認してから重要なゾーンに導入することができます。

設定ゾーンの動作は次のポリシーによって指定します、

- `infrazone:ZoneP` は、システム アップグレードに自動的に作成されます。削除または変更することはできません。
- `infrazone:Zone` には、1 つ以上のノード グループ（`NodeGrp`）が含まれます。ノードは 1 個のゾーン（`infrazone:Zone`）だけに所属できます`NodeGrp` には、名前および導入モードという 2 つのプロパティがあります。導入モード プロパティは次のとおりです。
  - `enabled`：保留中の更新がただちに送信されます。
  - `disabled`：新しいアップデートが保留されます。

    （注） disabled の設定ゾーンでは、ノードのアップグレード、ダウングレード、コミッション、デコミッションは行わないでください。

  - `triggered`：保留中の更新がただちに送信され、導入モードが `triggered` への変更前の値に自動的にリセットされます。

所定のノード セットでポリシーを作成、変更、または削除されると、ポリシーが導入されている各ノードに更新が送信されます。ポリシーのクラスと `infrazone` 設定に基づいて、次のような処理が行われます。

- `infrazone` 設定に従わないポリシーの場合、APIC がすべてのファブリック ノードにただちに更新を送信します。
- `infrazone` 設定に従うポリシーの場合は、`infrazone` 設定に従って更新が続行します。
  - ノードが `infrazone:Zone` に含まれている場合、更新は、ゾーンの導入モードが有効に設定されていればただちに送信されます。それ以外では更新は保留になります。
  - ノードが `infrazone:Zone` に含まれている場合は、すぐに更新が実行されます。これは ACI ファブリックのデフォルトの動作です。

ファブリック セキュア モード

ファブリックセキュアモードでは、ファブリック機器に物理的にアクセスできるユーザが管理者による手動の認証を受けずにファブリックにスイッチまたは APIC コントローラを追加するのを防止できます。リリース 1.2(1x) 以降、ファームウェアでは、ファブリックのスイッチとコントローラに有効なシスコ デジタル署名付き証明書と関連付けられた有効なシリアル番号があることを確認します。この検証は、このリリースへのアップグレード時、またはファブリックの最初のインストール時に行われます。この機能のデフォルト設定は、非制限モードです。既存のファブリックは、リリース 1.2(1) へのアップグレード後も継続して実行されます。ファブリック全体のアクセス権限を持つ管理者は、strict モードを有効にする必要があります。この 2 つの動作モードの要約を、以下の表に示します。

非制限モード（デフォルト）	**Strict モード**
1 つ以上のスイッチに無効な証明書がある場合でも既存のファブリックは正常に動作することができます。	有効なシスコ シリアル番号および SSL 証明書を持つスイッチのみが許可されます。
シリアル番号に基づく認証は適用されません。	シリアル番号認証が適用されます。
自動検出されたコントローラとスイッチはシリアル番号の認証を適用せずにファブリックに参加することができます。	コントローラとスイッチがファブリックに参加するには管理者が手動で承認する必要があります。

拡張 GUI を使用して、[Fabric] > [Inventory] > [Fabric Membership] 画面に SSL 証明書のステータスを表示します。strict モードを有効にするには、[System] > [Controllers] > [Cluster as Seen by Node] 画面の [Properties] セクションの [ACI Fabric Internode Secure Authentication Communications] フィールドで strict を選択します。新しいコントローラを受け入れるには、コントローラを右クリックして、[System]> [Controllers] > [Cluster as Seen by Node] 画面の [Unauthorized Controllers] セクションで `[accept controller]` を選択します。

この例に示すように、CLI を使用して、show switch コマンドでファブリックの SSL ステータスを確認します。

```
ifav74-ifc1# show switch
 ID    Name         Address          In-Band Address  OOB Address      Flags  Serial Number
 ----  -----------  ---------------  ---------------  ---------------  -----  ----------------

 102   ifav74-spine1 10.0.176.94      10.10.10.5       172.23.55.123    asiv   FGE1733000M
 108   ifav74-leaf2  10.0.176.93      10.10.10.4       172.23.49.218    aliv   SAL17299LWU
 109   ifav74-leaf1  10.0.176.95      10.10.10.3       172.23.49.219    aliv   SAL17299LWG

Flags - a:Active | l/s:Leaf/Spine | v:Valid Certificate | i:In-Service
```

リーフ スイッチの認証を確認するには、この例に示すように、cat /mit/sys/lldp/inst/if-\[eth1--46\]/ctrlradj/summary コマンドを使用します。

```
ifav74-leaf2# cat /mit/sys/lldp/inst/if-\[eth1--46\]/ctrlradj/summary
# Controller Adjacency
authCookie   : 315369427
childAction    :
<snip>
```

```
rn              : ctrlradj
status          :
verified      : yes
```

コントローラを承認または拒否するには、この例に示すように、system controller-id コマンドを使用します。

```
ifav74-ifc1(config)# system controller-id FCH1750V025 ?
 approve  Approve controller
 reject   Reject controller
```

strict モードを有効にするか、非制限モードに戻すには、この例に示すよう、system fabric-security-mode コマンドを使用します。

```
ifav74-ifc1(config)# system fabric-security-mode
 permissive  permissive
 strict      strict
```

REST API を使用し、次のコマンドをポストすることで strict モードを有効にします。

```
<!-- /api/node/mo/uni.xml? -->
  <pkiFabricCommunicationEp mode="strict"/>
```

REST API を使用し、次のコマンドをポストすることで非制限モードに切り替えます。

```
<!-- /api/node/mo/uni.xml? -->
  <pkiFabricCommunicationEp mode="permissive"/>
```

次の REST コマンドを使用してコントローラを承認します。

```
<!-- /api/mo/uni/controller.xml? -->
  <fabricNodeIdentPol>
      <fabricCtrlrIdentP serial="TEP-1-1"/>
  </fabricNodeIdentPol>
```

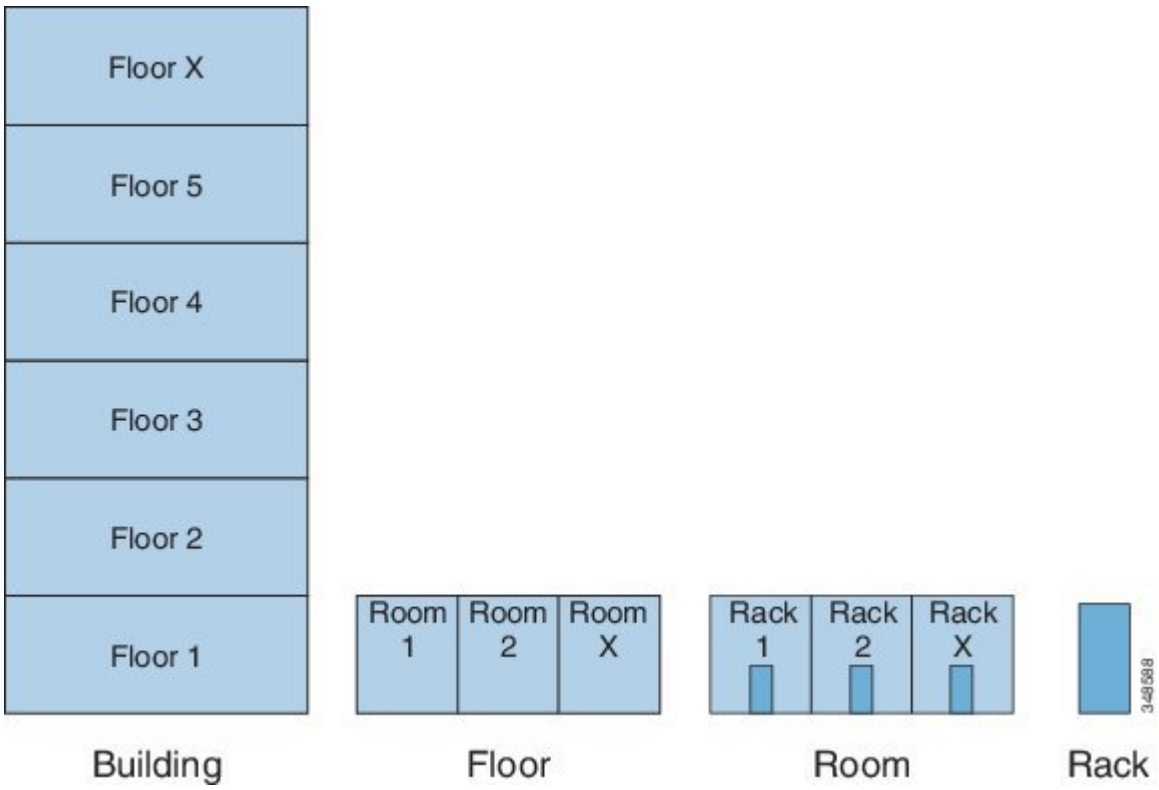
次の REST コマンドを使用してコントローラを拒否します。

```
<!-- /api/mo/uni/controller.xml? -->
  <fabricNodeIdentPol>
      <fabricCtrlrIdentP serial="FCH1750V025" reject="yes"/>
  </fabricNodeIdentPol>
```

位置情報

管理者は、位置情報ポリシーを使用して、データセンター施設内のACIファブリック ノードの物理ロケーションをマッピングします。次の図は、位置情報マッピング機能の例を示します。

図 *34*：位置情報



たとえば、単一の部屋でのファブリック展開の場合は、管理者がデフォルトのルーム オブジェクトを使用して、スイッチの物理ロケーションに一致する1つ以上のラックを作成します。大規模な展開の場合、管理者は1つ以上のサイト オブジェクトを作成できます。各サイトには、1つ以上の建物を含めることができます。各建物には、1つ以上のフロアがあります。各フロアには1つ以上の部屋があり、各部屋には1つ以上のラックがあります。最後に、各ラックは1つ以上のスイッチに関連付けることができます。

第 6 章

ネットワーキングと管理接続

この章の内容は、次のとおりです。

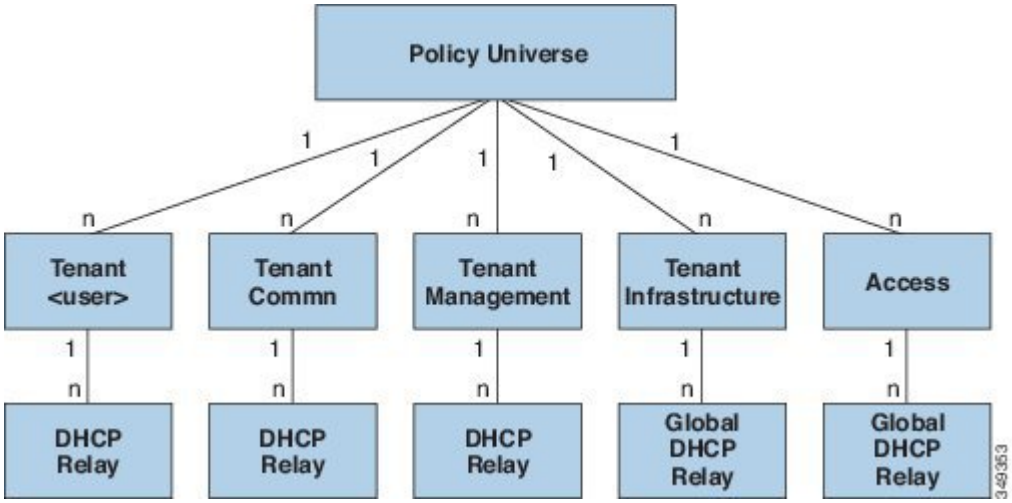
DHCP リレー

ACI のファブリック全体のフラッディングはデフォルトでディセーブルになっている一方で、ブリッジドメイン内のフラッディングはデフォルトでイネーブルになっています。ブリッジドメイン内のフラッディングがデフォルトでイネーブルになっているため、クライアントは同じ EPG 内の DHCP サーバに接続できます。ただし、DHCP サーバがクライアントとは別の EPG またはコンテキストにある場合は、DHCP リレーが必要です。また、レイヤ 2 フラッディングがディセーブルの場合、DHCP リレーが必要です。

（注） ACI ファブリックは DHCP リレーとして動作するときに、DHCP オプション 82（DHCP Relay Agent Information Option）を、クライアントの代わりに中継する DHCP 要求に挿入します。応答（DHCP オファー）がオプション 82 なしで DHCP サーバから返された場合、その応答はファブリックによってサイレントにドロップされます。ACI が DHCP リレーとして動作するときは、ACI ファブリックに接続されたノードを計算するために IP アドレスを提供している DHCP サーバはオプション 82 をサポートする必要があります。Windows 2003 および 2008 はオプション 82 をサポートしていませんが、Windows 2012 はサポートしています。

次の図は、DHCP リレー（ユーザ テナント、共通テナント、インフラストラクチャ テナント、管理テナントおよびファブリック アクセス）を含むことができる管理情報ツリー（MIT）内の管理対象オブジェクトを示します。

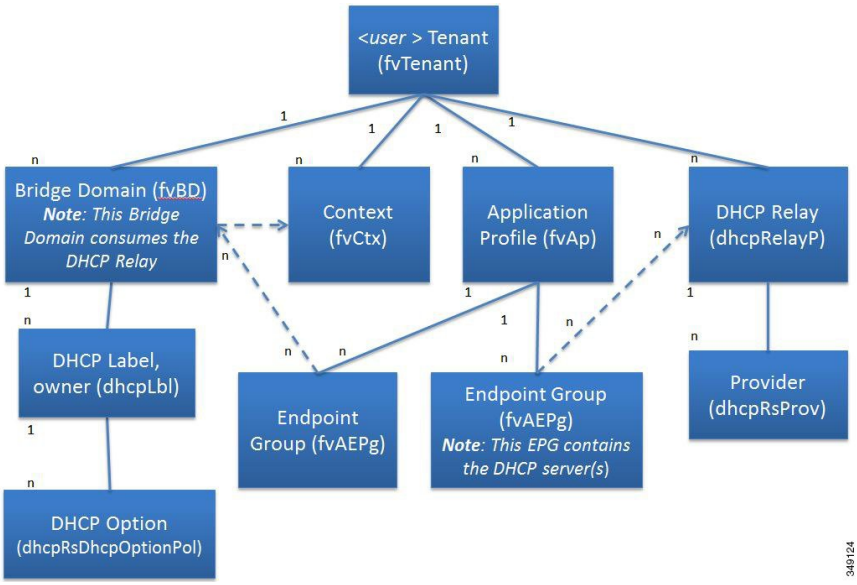
図 *35*：*MIT* 内の *DHCP* リレーの場所



（注） DHCP リレーはブリッジ ドメインごとに 1 つのサブネットに制限されます。

次の図は、ユーザ テナント内の DHCP リレー オブジェクトの論理関係を示します。

図 *36*：テナント *DHCP* リレー



DHCP リレー プロファイルには、1 つ以上のプロバイダーが含まれます。EPG には 1 つ以上の DHCP サーバが含まれ、EPG と DHCP リレーの関係は DHCP サーバの IP アドレスを指定します。コンシューマ ブリッジ ドメインには、プロバイダーの DHCP サーバをブリッジ ドメインと関連付ける DHCP ラベルが含まれます。ラベルの一致により、ブリッジ ドメインは DHCP リレーを消費できます。

（注） ブリッジ ドメインの DHCP ラベルは、DHCP リレーの名前と一致する必要があります。

DHCP ラベル オブジェクトは、所有者も指定します。所有者には、テナントまたはアクセス インフラストラクチャを指定できます。所有者がテナントの場合、ACI ファブリックは最初にテナント内で一致する DHCP リレーを検索します。ユーザ テナント内で一致するものが見つからなかった場合、ACI ファブリックは次に共通テナント内を検索します。

DHCP リレーは、次の 2 つのモードのいずれかで動作します。

- 可視：プロバイダーの IP およびサブネットは、コンシューマのコンテキストにリークされます。DHCP リレーが表示されているときは、コンシューマのコンテキストに限定されます。
- 非可視：プロバイダーの IP およびサブネットは、コンシューマのコンテキストにリークされません。

（注） DHCP リレーが非可視モードで動作している場合、プロバイダーのブリッジ ドメインはコンシューマと同じリーフ スイッチ上にある必要があります。

テナントおよびアクセスの DHCP リレーが同じ方法で設定されている一方で、以下の使用例はそれに応じて異なります。

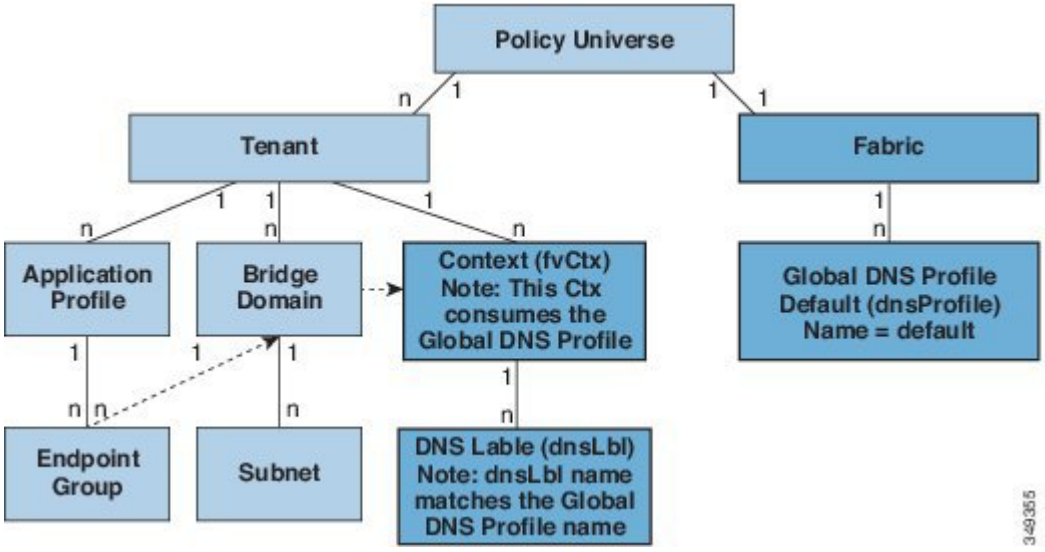
- 共通テナントの DHCP リレーは、どのテナントでも使用できます。
- インフラストラクチャ テナントの DHCP リレーは、ACI ファブリックのサービス プロバイダーによって他のテナントに選択的に公開されます。
- ファブリック アクセス（`infraInfra`）の DHCP リレーは、どのテナントでも使用でき、DHCP サーバのより細かい設定が可能になります。この場合、同じブリッジ ドメイン内の別個の DHCP サーバをノードプロファイルの各リーフ スイッチ用にプロビジョニングすることができます。

DNS

ACI ファブリックの DNS サービスは、ファブリックの管理対象オブジェクトに含まれます。ファブリックのグローバル デフォルト DNS プロファイルには、ファブリック全体でアクセスできま

す。次の図は、ファブリック内のDNS管理対象オブジェクトの論理関係を示します。付録F「DNS for sample DNS XMP policies（サンプルの DNS XMP ポリシー用の DNS）」を参照してください。

図 *37* : *DNS*



コンテキストには、グローバル デフォルト DNS サービスを使用するために `dnsLBL` オブジェクトを含める必要があります。ラベルの一致により、テナント コンテキストはグローバル DNS プロバイダーを消費することができます。グローバル DNS プロファイルの名前が「default」なので、コンテキスト ラベル名は「default」になります（`dnsLBL name = default`）。

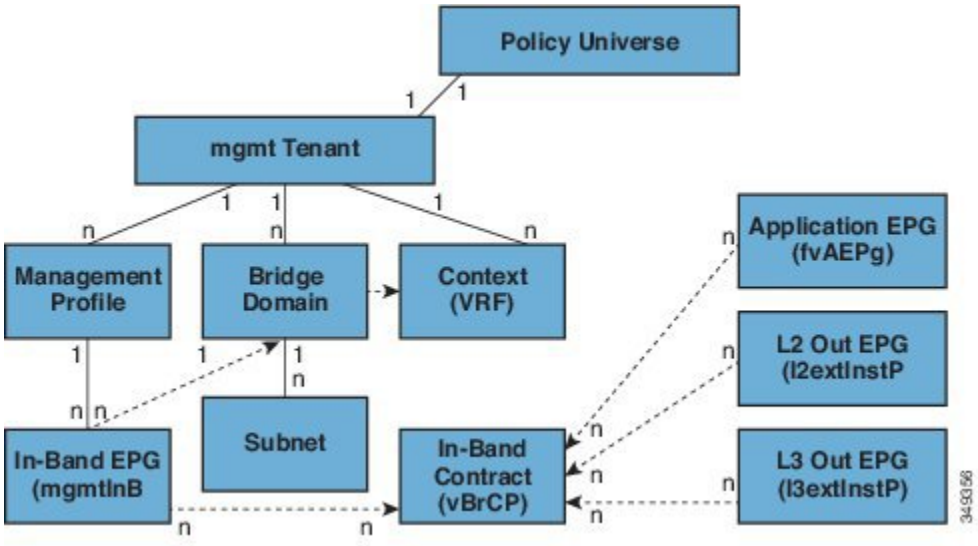
インバンドおよびアウトオブバンド管理アクセス

管理テナントでは、ファブリック管理機能へのアクセスを設定するための便利な方法が提供されます。APICを介してファブリック管理機能にアクセスできると同時に、インバンドおよびアウトオブバンドのネットワーク ポリシー経由で直接アクセスすることもできます。

インバンド管理アクセス

次の図は、管理テナントのインバンド ファブリック管理アクセス ポリシーの概要を示します。

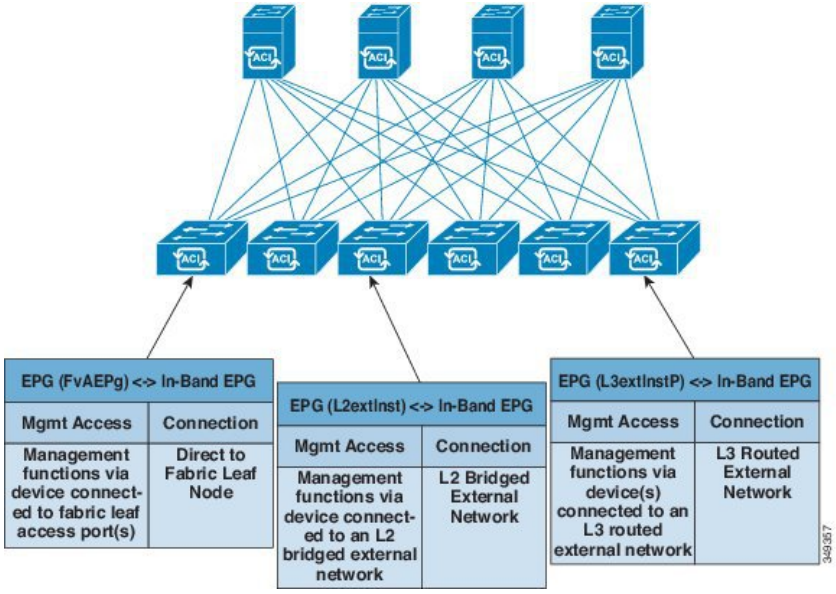
図 *38*：インバンド管理アクセス ポリシー



管理プロファイルには、インバンド コントラクト（`vzBrCP`）を介した管理機能へのアクセスを提供するインバンドEPG MO が含まれます。`vzBrCP` は、`fvAEPg`、`l2extInstP`、および `l3extInstP` EPG がインバンドEPGを消費することを可能にします。これにより、ローカルで接続されたデバイスや、レイヤ 2 ブリッジド外部ネットワークおよびレイヤ 3 ルーテッド外部ネットワーク経由で接続されたデバイスにファブリック管理が提供されます。コンシューマおよびプロバイダーEPGが異なるテナントにある場合は、共通テナントからブリッジ ドメインおよびコンテキストを使用できます。認証、アクセス、および監査のロギングはこれらの接続に適用され、インバンドEPGを通して管理機能にアクセスしようとするユーザには適切なアクセス権が必要です。

次の図は、インバンド管理のアクセス シナリオを示します。

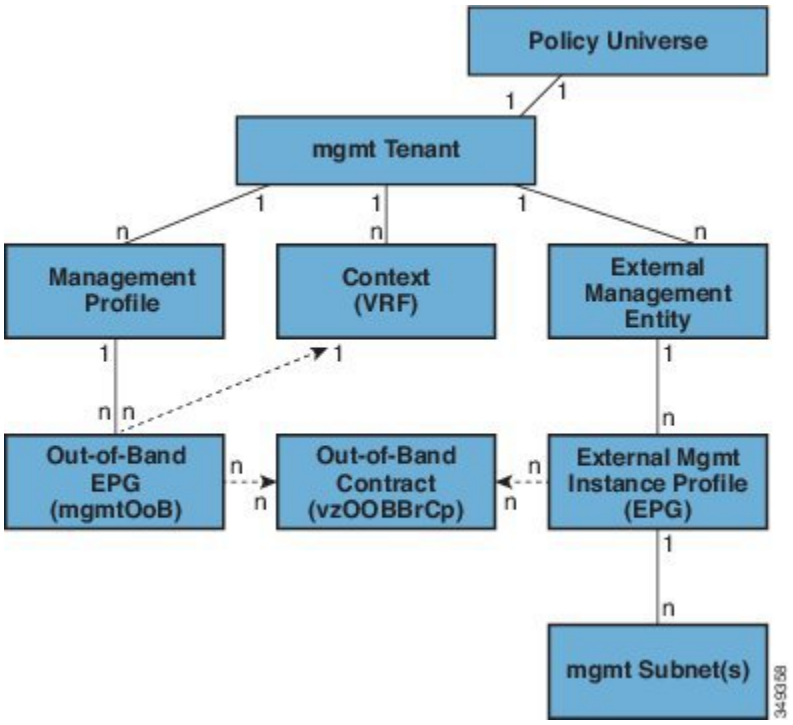
図 *39*：インバンド管理のアクセス シナリオ



アウトオブバンド管理アクセス

次の図は、管理テナントのアウトオブバンドファブリック管理アクセスポリシーの概要を示します。

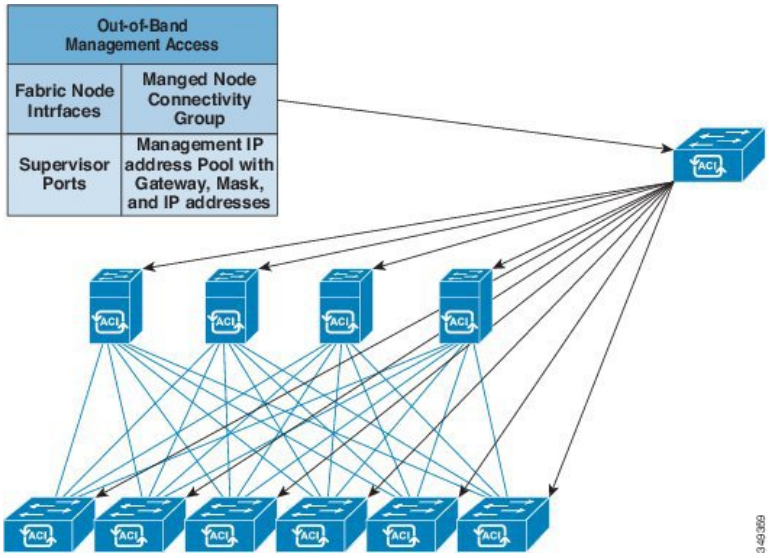
図 *40*：アウトオブバンド管理アクセス ポリシー



管理プロファイルには、アウトオブバンドコントラクト（vzOOBBrCP）を介した管理機能へのアクセスを提供するアウトオブバンド EPG MO が含まれます。vzOOBBrCP により、外部管理インスタンス プロファイル（mgmtExtInstP）EPG はアウトオブバンド EPG を消費できます。これにより、サービス プロバイダーのプリファレンスに応じて、ローカルまたはリモートで接続されたデバイスにファブリック ノードのスーパーバイザ ポートが公開されます。スーパーバイザ ポートの帯域幅がインバンドポート未満である間は、インバンドポートを介したアクセスが利用できない場合、スーパーバイザポートがダイレクトアクセスを提供できます。認証、アクセス、および監査のロギングはこれらの接続に適用され、アウトオブバンド EPG を通して管理機能にアクセスしようとするユーザには適切なアクセス権が必要です。管理者が外部管理インスタンス プロファイルを設定すると、それにより、アウトオブバンドアクセスを許可されるデバイスのサブネットの範囲が指定されます。この範囲内にないデバイスは、アウトオブバンド アクセスができません。

次の図は、アウトオブバンド管理アクセスを専用スイッチを通じてどのように統合できるかについて示します。

図 *41*：アウトオブバンド アクセスのシナリオ



サービス プロバイダーによってはローカル接続へのアウトオブバンド接続を制限するように選択します。また、外部ネットワークからルーテッドまたはブリッジド接続を有効にすることを選択するサービス プロバイダーも存在します。また、サービス プロバイダーはローカル デバイスのみ、またはローカルおよびリモート デバイス両方に対するインバンドおよびアウトオブバンド管理アクセスの両方を含む一連のポリシーを設定することを選択することもできます。

（注） APIC リリース 1.2(2) 以降では、アウトオブバンド管理ノード EPG でコントラクトが提供されると、アウトオブバンド ノード管理アドレスで設定されるローカル サブネットが、デフォルトの APIC アウトオブバンド コントラクト送信元アドレスになります。以前は、任意のアドレスをデフォルトの APIC アウトオブバンド コントラクト送信元アドレスにすることが可能でした。

テナント内のルーティング

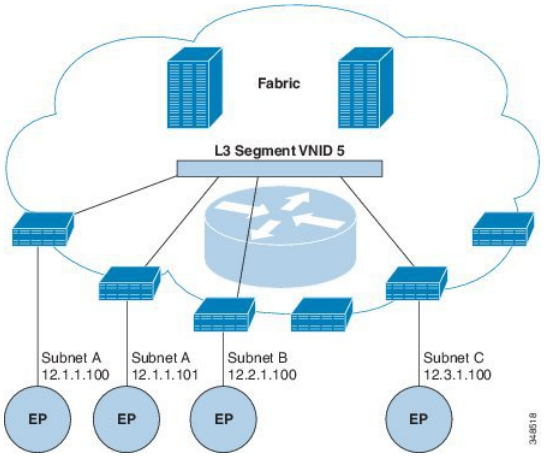
アプリケーション セントリック インフラストラクチャ（ACI）のファブリックでは、テナントのデフォルト ゲートウェイ機能が提供され、ファブリックの Virtual Extensible Local Area（VXLAN）ネットワーク間のルーティングが行えます。各テナントについて、APIC でサブネットが作成されるたびに、ファブリックは仮想デフォルト ゲートウェイまたはスイッチ仮想インターフェイス（SVI）を提供します。これは、そのテナント サブネットの接続エンドポイントがあるすべてのスイッチにわたります。各入力インターフェイスはデフォルトのゲートウェイ インターフェイス

をサポートし、ファブリック全体のすべての入力インターフェイスは任意のテナント サブネットに対する同一のルータの IP アドレスと MAC アドレスを共有します。

Intersubnet のテナント トラフィックを転送するために使用されるレイヤ 3 VNID

ACI モデルでは、ACI ファブリックのデフォルト ゲートウェイに送信されるファブリックのイングレスに到達するトラフィックは、レイヤ 3 VNID として知られる仮想ネットワーク セグメントにルーティングされます。単一のレイヤ 3 VNID が、各テナント コンテキストに割り当てられます。次の図は、テナント内のルーティングがどのように行われるかを示します。

図 *42*：*Intersubnet* のテナント トラフィックを転送するレイヤ *3 VNID*



レイヤ 3 VNID は、APIC によって割り当てられます。ファブリックを経由するトラフィックは、レイヤ 3 セグメントの VNID を使用して転送されます。出力リーフ スイッチでは、パケットはレイヤ 3 セグメントの VNID から出力サブネットの VNID にルーティングされます。

ACI モデルでは、テナント内でルーティングされるトラフィックのファブリックで非常に効率的な転送が提供されます。たとえば、同じ物理ホストの同じテナントに属するがサブネットは異なる 2 つの仮想マシン（VM）間のトラフィックでは、（最小パス コストを使用して）正しい宛先にルーティングされる前の移動先は入力スイッチのみです。現在の VM 環境では、トラフィックは正しい宛先にルーティングされる前に、（異なる物理サーバ上にあると思われる）エッジ VM に伝送されます。

ルート リフレクタの設定

ACI ファブリックのルート リフレクタは、マルチプロトコル BGP（MP-BGP）を使用してファブリック内に外部ルートを配布します。ACI ファブリックでルート リフレクタをイネーブルにするには、ファブリックの管理者がルート リフレクタになるスパイン スイッチを選択して、自律シス

テム（AS）番号を提供する必要があります。ルート リフレクタが ACI ファブリックでイネーブルになると、管理者は次の項で説明するように、外部ネットワークへの接続を設定できます。

ACI ファブリックに外部ルータを接続するには、ファブリック インフラストラクチャの管理者がボーダー ゲートウェイ プロトコル（BGP）のルート リフレクタとしてスパイン ノードを設定します。冗長性のために、複数のスパインがルータ リフレクタ ノードとして設定されます（1 台のプライマリ リフレクタと 1 台のセカンダリ リフレクタ）。

テナントが ACI ファブリックに WAN ルータを接続する必要がある場合は、インフラストラクチャの管理者が WAN ルータが WAN のトップ オブ ラック（ToR）として接続されるリーフ ノードを（以下の通りに）設定し、この WAN ToR を BGP ピアとしてルート リフレクタ ノードの 1 つと組み合わせます。ルート リフレクタが WAN ToR に設定されていると、ファブリックにテナント ルートをアドバタイズできます。

各リーフ ノードには最大 4000 のルートを保存できます。WAN ルータが 4000 を超えるルートをアドバタイズしなければならない場合、複数のリーフ ノードとピアリングする必要があります。インフラストラクチャの管理者は、ペアになったリーフ ノードそれぞれをアドバタイズできるルート（またはルート プレフィクス）で設定します。

インフラストラクチャの管理者は、次のようにファブリックに接続されている外部 WAN ルータを設定する必要があります。

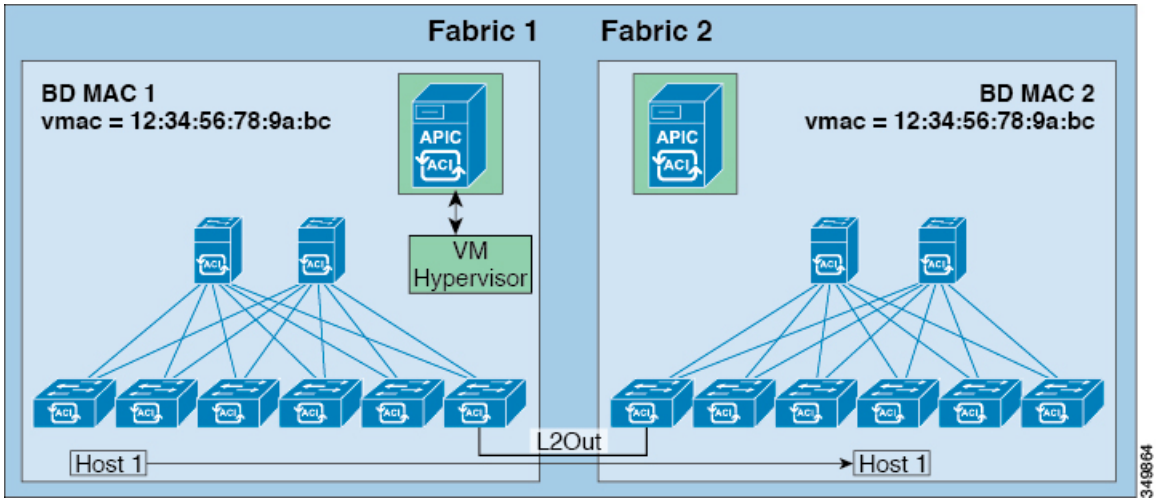
1. ルート リフレクタとして最大 2 つのスパイン ノードを設定します。冗長性のために、プライマリおよびセカンダリ ルート リフレクタを設定します。
2. WAN ToR で、プライマリおよびセカンダリ ルート リフレクタのノードを設定します。
3. WAN ToR で、ToR がアドバタイズを担当するルートを設定します。これは任意で、テナント ルータが 4000 を超えるルートをアドバタイズすることがわかっている場合にのみ行う必要があります。

共通パーベイシブ ゲートウェイ

ブリッジ ドメインごとに IPv4 共通ゲートウェイを使用して複数の ACI ファブリックを設定できます。これにより、1 つ以上の仮想マシン（VM）または従来のホストを、ホストがその IP アドレスを保持したままファブリック間で移動できます。ファブリック間の VM ホストの移動は、VM ハイパーバイザによって自動的に行うことができます。ACI ファブリックは、同じ場所に配置することも、複数のサイト間でプロビジョニングすることもできます。ACI ファブリック間のレイ

ヤ 2 接続は、ローカル リンクか、ルーテッド WAN リンクになります。次の図は、基本的な共通パーベイシブ ゲートウェイ トポロジを示しています。

図 *43* : *ACI* 複数ファブリック共通パーベイシブ ゲートウェイ



ブリッジ ドメインごとの共通パーベイシブ ゲートウェイの設定要件は、次のとおりです。

- 各ファブリックのブリッジ ドメイン MAC（*mac*）値は一意である必要があります。

（注） デフォルトのブリッジ ドメイン MAC（*mac*）アドレス値はすべての ACI ファブリックで同じです。共通パーベイシブ ゲートウェイでは、管理者は、ブリッジ ドメイン MAC（*mac*）値が各 ACI ファブリックで一意になるように設定する必要があります。

- ブリッジ ドメインの仮想 MAC（*vmac*）アドレスとサブネットの仮想 IP アドレスは、ブリッジ ドメインのすべての ACI ファブリックで同じにする必要があります。複数のブリッジ ドメインを、接続されている ACI ファブリック間で通信するように設定できます。仮想 MAC アドレスと仮想 IP アドレスは、ブリッジ ドメイン間で共有できます。

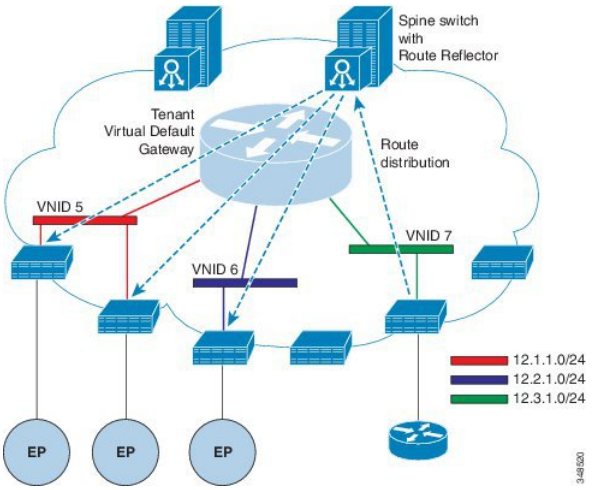
WAN およびその他の外部ネットワーク

WAN およびエンタープライズ コアに接続する外部ルータは、リーフ スイッチの前面パネルのインターフェイスに接続します。外部ルータに接続するリーフ スイッチ インターフェイスは、ブリッジ インターフェイスまたはルーティング ピアとして設定できます。

ルータ ピアリングおよびルート配布

次の図に示すように、ルーティング ピア モデルを使用すると、リーフ スイッチ インターフェイスが外部ルータのルーティング プロトコルとピアリングするように静的に設定されます。

図 *44*：ルータのピアリング



ピアリングによって学習されるルートは、スパイン スイッチに送信されます。スパイン スイッチはルート リフレクタとして動作し、外部ルートを同じテナントに属するインターフェイスを持つすべてのリーフ スイッチに配布します。これらのルートは、最長プレフィクス照合（LPM）により集約されたアドレスで、外部ルータが接続されているリモートのリーフ スイッチの VTEP IP アドレスが含まれるリーフ スイッチの転送テーブルに配置されます。WAN ルートには転送プロキシはありません。WAN ルートがリーフ スイッチの転送テーブルに適合しない場合、トラフィックはドロップされます。外部ルータがデフォルト ゲートウェイではないため、テナントのエンドポイント（EP）からのパケットはACIファブリックのデフォルト ゲートウェイに送信されます。

ネットワーク ドメイン

ファブリック管理者は、ポート、プロトコル、VLAN プール、およびカプセル化を設定するドメイン ポリシーを作成します。これらのポリシーは、単一テナント専用にすることも、共有することもできます。ファブリック管理者が ACI ファブリック内にドメインを設定すると、テナント管理者はテナント エンドポイント グループ（EPG）をドメインに関連付けることができます。

以下のネットワーク ドメイン プロファイルを設定できます。

- VMM ドメイン プロファイル（`vmmDomP`）は、仮想マシンのハイパーバイザ統合のために必要です。
- 物理ドメイン プロファイル（`physDomP`）は、ベア メタル サーバ接続と管理アクセスに使用します。

- ブリッジド外部ネットワーク ドメイン プロファイル（l2extDomP）は通常、ACI ファブリックのリーフ スイッチにブリッジド外部ネットワーク トランク スイッチを接続するために使用されます。
- ルーテッド外部ネットワーク ドメイン プロファイル（l3extDomP）は、ACI ファブリックのリーフ スイッチにルータを接続するために使用されます。

ドメインはVLANプールに関連付けられるように設定されます。その後、EPGは、ドメインに関連付けられている VLAN を使用するように設定されます。

（注） EPG ポートと VLAN の設定は、EPG が関連付けられているドメイン インフラストラクチャ設定で指定されている設定に一致する必要があります。一致しない場合、APIC でエラーが発生します。そのようなエラーが発生した場合は、ドメインインフラストラクチャ設定がEPGポートと VLAN の設定に一致していることを確認してください。

接続可能エンティティ プロファイル

ACI ファブリックにより、リーフ ポートを通してベア メタル サーバ、仮想サーバ、ハイパーバイザ、レイヤ 2 スイッチ（たとえば、Cisco UCS ファブリック インターコネクト）、またはレイヤ 3 ルータ（たとえば、Cisco Nexus 7000 シリーズ スイッチ）などのさまざまな外部エンティティに接続する複数の接続ポイントが提供されます。これらの接続ポイントは、リーフ スイッチ上の物理ポート、FEX ポート、ポート チャネル、またはバーチャル ポート チャネル（vPC）にすることができます。

接続可能エンティティ プロファイル（AEP）は、同様のインフラストラクチャ ポリシー要件を持つ外部エンティティのグループを表します。インフラストラクチャ ポリシーは、Cisco Discovery Protocol（CDP）、Link Layer Discovery Protocol（LLDP）、最大伝送単位（MTU）、Link Aggregation Control Protocol（LACP）などのさまざまなプロトコル オプションを設定する物理インターフェイス ポリシーで構成されます。

AEPは、リーフ スイッチでVLANプールを展開するのに必要です。カプセル化ブロック（および関連 VLAN）は、リーフ スイッチで再利用可能です。AEP は、VLAN プールの範囲を物理インフラストラクチャに暗黙的に提供します。

次の AEP の要件と依存関係は、さまざまな設定シナリオ（ネットワーク接続や VMM ドメインなど）でも考慮する必要があります。

- AEPは許容されるVLANの範囲を定義しますが、それらのプロビジョニングは行いません。EPG がポートに展開されていない限り、トラフィックは流れません。AEP で VLAN プールを定義しないと、EPG がプロビジョニングされても VLAN はリーフ ポートでイネーブルになりません。
- リーフ ポートで静的にバインディングしている EPG イベントに基づいて、または VMware vCenter や Microsoft Azure Service Center Virtual Machine Manager（SCVMM）などの外部コントローラからの VM イベントに基づいて、特定の VLAN がリーフ ポート上でプロビジョニングされるかイネーブルになります。

Virtual Machine Manager（VMM）ドメインは、AEP のインターフェイス ポリシー グループから物理インターフェイス ポリシーを自動的に取得します。

AEP のオーバーライド ポリシーを VMM ドメイン用の別の物理インターフェイス ポリシーを指定するために使用できます。このポリシーは、VM コントローラが中間レイヤ 2 ノードを介してリーフ スイッチに接続され、異なるポリシーがリーフ スイッチおよび VM コントローラの物理ポートで要求される場合に役立ちます。たとえば、リーフ スイッチとレイヤ 2 ノード間で LACP を設定できます。同時に、AEP オーバーライド ポリシーで LACP をディセーブルにすることで、VM コントローラとレイヤ 2 スイッチ間の LACP をディセーブルにできます。

外部ネットワークへのブリッジドおよびルーテッド接続

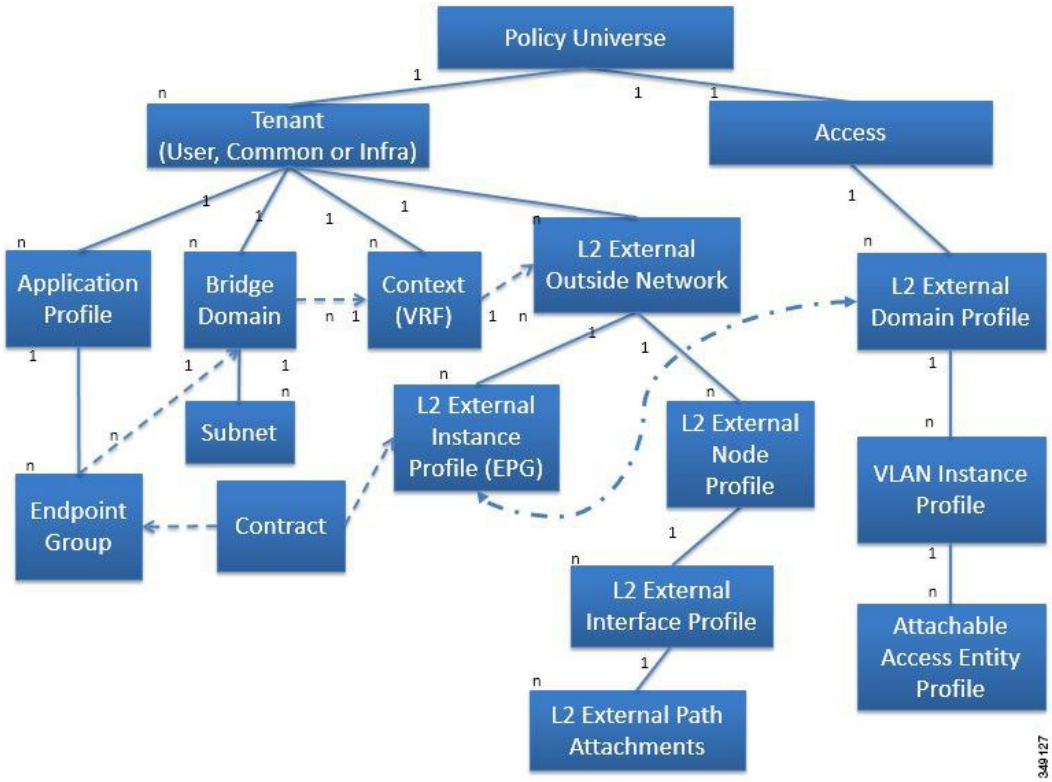
外部ネットワークの管理対象オブジェクトにより、外部ネットワークへのレイヤ 2 およびレイヤ 3 のテナント接続が可能になります。GUI、CLI、または REST API は、外部ネットワークへのテナント接続を設定するために使用できます。付録 E「テナント レイヤ 3 の外部ネットワーク ポリシーの例」には、サンプルの XML ポリシーが含まれます。ファブリック内のそのような外部ネットワーク アクセス ポイントすべてを簡単に検索するために、レイヤ 2 およびレイヤ 3 の外部リーフ ノードを「ボーダー リーフ ノード」としてタグ付けできます。

外部ネットワークへのブリッジド接続のためのレイヤ 2 Out

外部ネットワークへのテナント レイヤ 2 ブリッジド接続は、次の図に示すようにファブリック アクセス（`infraInfra`）外部ブリッジド ドメイン（`L2extDomP`）をレイヤ 2 外部外側ネットワーク

（l2extOut）のレイヤ 2 外部インスタンス プロファイル（l2extInstP）に関連付けることによって有効になります。

図 *45*：外部ネットワークへのテナント ブリッジド接続



l2extOut には、スイッチ固有の設定およびインターフェイス固有の設定が含まれます。l2extInstP EPG は、コントラクトを通してテナント EPG に外部ネットワークを公開します。たとえば、ネットワーク接続ストレージ デバイスのグループを含むテナント EPG は、レイヤ 2 外部外側ネットワークに含まれるネットワーク構成に応じてコントラクトを介して l2extInstP EPG と通信できます。リーフ スイッチ 1 個につき設定できる外部ネットワークは 1 つのみです。ただし、外部ネットワーク設定は、複数のノードを L2 外部ノード プロファイルに関連付けることで複数のノードに容易に再利用できます。同じプロファイルを使用する複数のノードをフェールオーバーやロード バランシングのために設定できます。

ポート単位の VLAN

v1.1 リリースより前の ACI バージョンでは、特定のカプセル化 VLAN はリーフ スイッチ上の単一の EPG だけにマッピングされます。同じリーフ スイッチ上に同じ VLAN カプセル化を持つ第 2 の EPG があると、ACI でエラーが発生します。

v1.1 リリース以降では、EPGS がそれぞれ違ったブリッジ ドメインに関連付けられている場合に限り、同じ VLAN カプセル化を持つ複数の EPG を特定のリーフ スイッチ ポート（または FEX ポート）で導入できるようになりました。この設定は、EPG が同じブリッジ ドメインに属してい

る場合は無効になります。これは、外部レイヤ 3 外部接続用に設定されたポートには適用されません。

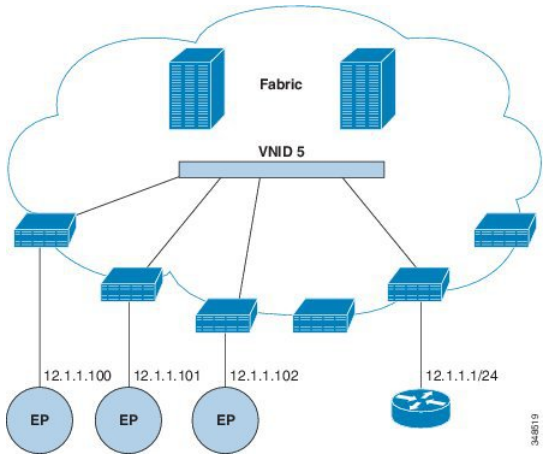
新しいインターフェイス ポリシー（l2IfPol）では、同一の VLAN カプセル化を、同じアクセス ポート（および FEX ポート）や 1 つのリーフ スイッチ内の異なるアクセス ポート（および FEX ポート）上の、異なるブリッジ ドメインに属する、複数の EPG と一緒に使用できます。l2IfPol ポリシーの vlanScope 属性は、global または portlocal です。このポリシーは infraInfra に含まれており、infraAccGrp からの関係があります。以前のファームウェア バージョンから v1.1 以降のファームウェア バージョンへのアップグレード時に、vlanScope が global であるデフォルトの l2IfPol ポリシーが作成され、すべてのスイッチ ポートによって消費されます。同じ VLAN カプセル化を持つ異なるブリッジ ドメインに関連付けられた EPG は、異なる物理ドメインおよび異なる名前空間プールに関連付ける必要があります。1 つのブリッジ ドメインに属している 2 つの EPG は、リーフ スイッチで同じカプセル化値を共有できません。

入力および出力の両方向で個別の（ポート、VLAN）変換エントリの割り当てが可能なのは、vlanScope が portlocal に設定されているポートだけです。vlanScope が portlocal に設定されているポートでは、各 VLAN は一意でなければなりません。ポート P1 および VLAN V1 が与えられると、2 番目の P1V1 設定は失敗します。

外部ルータへのブリッジド インターフェイス

次の図に示すように、リーフ スイッチのインターフェイスがブリッジドインターフェイスとして設定されている場合、テナント VNID のデフォルト ゲートウェイが外部ルータとなります。

図 *46*：ブリッジド外部ルータ

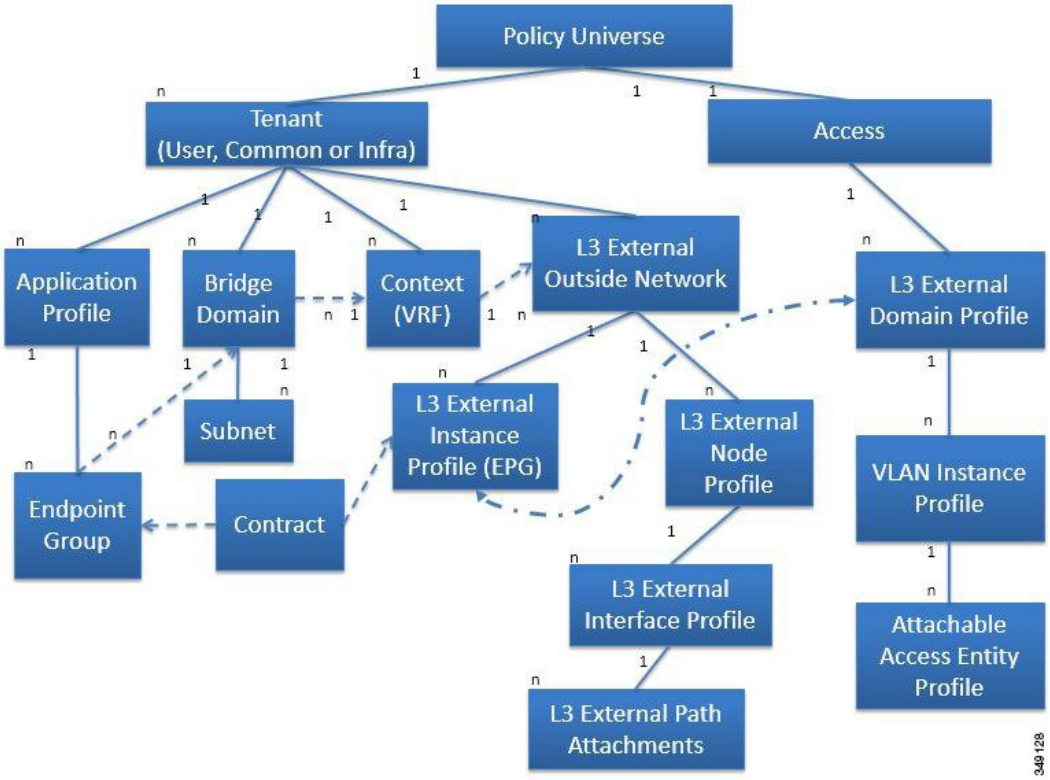


ACI ファブリックは、外部ルータの存在を認識せず、APIC はリーフ スイッチのインターフェイスを EPG に静的に割り当てます。

外部ネットワークへのルーテッド接続のためのレイヤ 3 Out

外部ネットワークへのルーテッド接続は、次の図に示すようにファブリック アクセス（infraInfra）外部ルーテッド ドメイン（l3extDomP）をレイヤ 3 外部外側ネットワーク（l3extOut）のテナント レイヤ 3 外部インスタンス プロファイル（l3extInstP）に関連付けることによって有効になります。

図 47：外部ネットワークへのテナント ルーテッド接続



l3extOut には、ルーティング プロトコル オプション（BGP、OSPF または両方）とスイッチ固有の設定およびインターフェイス固有の設定が含まれます。レイヤ 3 外部外側ネットワークにルーティング プロトコル（たとえば、関連するコンテキストとエリア ID を含む OSPF）が含まれる一方で、レイヤ 3 外部インターフェイスのプロファイルには必要な OSPF インターフェイス設定の詳細が含まれます。いずれも OSPF のイネーブル化に必要です。

（注） テナント ブリッジ ドメインには、共通テナントでプロビジョニングされている l3extOut によってアドバタイズされたパブリック サブネットを含めることができます。

l3extInstP EPG は、コントラクトを通してテナント EPG に外部ネットワークを公開します。たとえば、Web サーバのグループを含むテナント EPG は、レイヤ 3 外部外側ネットワークに含まれるネットワーク設定に応じてコントラクトを介して l3extInstP EPG と通信できます。外部ネット

ワーク設定は、複数のノードを L3 外部ノード プロファイルに関連付けることで複数のノードに容易に再利用できます。同じプロファイルを使用する複数のノードをフェールオーバーやロードバランシングのために設定できます。現在、コンテキスト（VRF）ごとに最大 3 つのレイヤ 3 外部ルーテッド接続を設定できます。各スイッチでマルチ コンテキスト（VRF）を設定できます。拡張性に関する情報については、現行の『Verified Scalability Guide for Cisco ACI』を参照してください。

リリース 1.2(1) 以降、入力ベースのポリシーの適用により、レイヤ 3 Out トラフィックのポリシー適用を出力方向および入力方向に定義することが可能になっています。デフォルトでは入力になっています。リリース 1.2(1) 以降へのアップグレード中に、既存のレイヤ 3 Out 設定が出力に設定され、既存の設定の動作と一致するようになります。特別なアップグレード手順を計画する必要はありません。アップグレード後、管理者がグローバル プロパティ値を入力に変更します。変更すると、システムが、規則とプレフィックス エントリをプログラムし直します。規則は出力リーフから削除され、入力リーフ上に既存の規則がない場合は、入力リーフ上にインストールされます。既存の設定がない場合、`Actrl` プレフィックス エントリが入力リーフ上にインストールされます。ダイレクト サーバ リターン（DSR）および属性 EPG には入力ベースのポリシー適用が必要です。vzAny とタブーは、入力ベースのポリシー適用を無視します。入力には中継規則が適用されます。

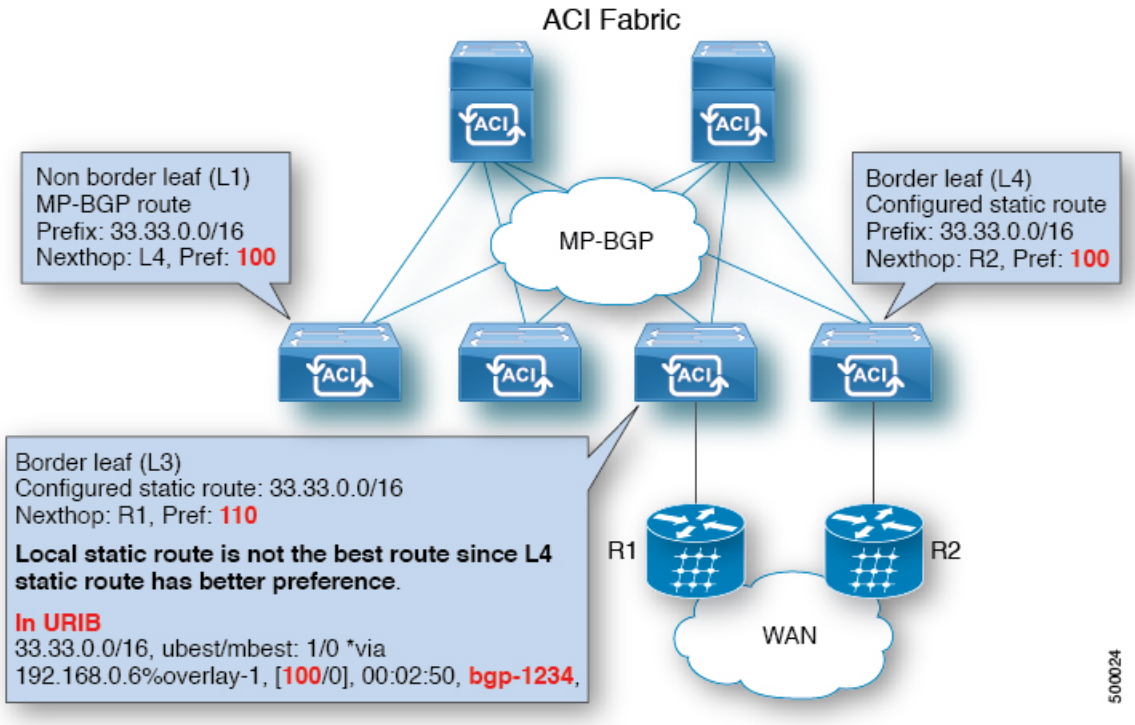
（注） リリース 1.2（1x）以降、BGP `l3extOut` 接続のテナント ネットワーキング プロトコル ポリシーは、最大プレフィックス制限を使用して設定できます。これにより、ピアから受信されるルート プレフィックスの数をモニタし、制限することができます。最大プレフィックス制限を超えると、ログ エントリの記録、それ以降のプレフィックスの拒否、固定期間中にカウントがしきい値未満になった場合の接続の再起動、または接続のシャットダウンを行うことができます。一度に 1 つのオプションだけを使用できます。デフォルト設定では 20,000 プレフィックスに制限され、その後は新しいプレフィックスは拒否されます。拒否オプションが導入されると、BGP は設定されている制限よりも 1 つ多くプレフィックスを受け入れ、APIC でエラーが発生します。

スタティック ルートのプリファレンス

ACI ファブリック内のスタティック ルートのプリファレンスは、コスト拡張コミュニティを使用して MP-BGP で送信されます。

次の図は、ACI ファブリックがスタティック ルートのプリファレンスをリーフ スイッチ間でそのまま維持することにより、ルート選択がこのプリファレンスに基づいて行われる様子を示しています。

図 *48*：スタティック ルートのプリファレンス



この図では、リーフ スイッチ4（L4）からリーフ スイッチ3（L3）に向かう MP-BGP ルートがローカル スタティック ルートより優先されています。スタティック ルートは、管理者が設定したプリファレンスでユニキャスト ルーティング情報ベース（URIB）にインストールされます。ACI の非境界リーフ スイッチでは、スタティック ルートはリーフ スイッチ4（L4）をネクストホップとしてインストールされます。L4 でネクストホップを使用できない場合は、L3 スタティック ルートがファブリック内で最善のルートになります。

（注） リーフ スイッチのスタティック ルートが `next hop Null 0` で定義されている場合、MP-BGP はそのルートをファブリック内の他のリーフ スイッチにアドバタイズしません。

ルートのインポートとエクスポート、ルート集約、ルート コミュニティの一致

サブネット ルート エクスポートまたはルート インポート設定オプションは、次に説明するスコープおよび集約オプションに従って指定できます。

ルーティング対象サブネットについては、以下のスコープ オプションが使用可能です。

- エクスポート ルート制御サブネット：エクスポート ルート方向を制御します。
- インポート ルート制御サブネット：インポート ルート方向を制御します。

（注） 現時点では、インポート ルート制御は BGP でのみサポートされています。

- 外部 EPG 用外部サブネット（セキュリティ インポート サブネット）：外部 EPG サブネットについてルート方向を制御します。
- 共有ルート制御サブネット：共有サービス設定において、この特性が有効になっているサブネットのみ、コンシューマ EPG コンテキスト（VRF）にインポートされます。コンテキスト（VRF）間の共有サービスのルート方向を制御します。
- 共有セキュリティ インポート サブネット：インポート対象サブネットに共有コントラクトを適用します。デフォルトの仕様では、外部 EPG 用外部サブネットが設定されています。

ルート対象サブネットを集約することができます。集約が設定されていない場合は、サブネットが正確に照合されます。たとえば、サブネットが 11.1.0.0/16 の場合、11.1.1.0/24 ルートにはポリシーが適用されず、ルートが 11.1.0.0/16 である場合のみ適用されます。すべてのサブネットを 1 つずつ定義する作業は面倒でエラーが発生しやすいので、それを回避するために、サブネットのセットを 1 つのエクスポート、インポートまたは共有ルート ポリシーに集約することができます。現時点では、0/0 サブネットのみ集約可能です。0/0 に集約を指定すると、次の選択オプションに基づき、すべてのルートがインポート、エクスポート、および異なるコンテキスト（VRF）と共有（異なるコンテキストへリーク）されます。

- 集約エクスポート：コンテキスト（VRF）（サブネット0/0）のすべての中継ルートをエクスポートします。
- 集約インポート：所定の L3 Out ピア（サブネット0/0）のすべて着信ルートをインポートします。

（注） 現時点では、集約インポート ルート制御は、BGP でのみサポートされます。

- 集約共有ルート：1 つのコンテキスト（VRF）で学習されているルートを別のコンテキスト（VRF）にリークする必要がある場合、サブネットと正確に一致させるか、またはサブネット マスクに従った集約方法でリークできます。集約共有ルートでは、複数のサブネット マスクを使用して、コンテキスト（VRF）間でリークさせるルート グループを指定できます。たとえば、10.1.0.0/16 と 12.1.0.0/16 を指定してこれらのサブネットを集約することができます。あるいは、0/0 を使用すると、複数のコンテキスト（VRF）のすべてのサブネット ルートを共有できます。

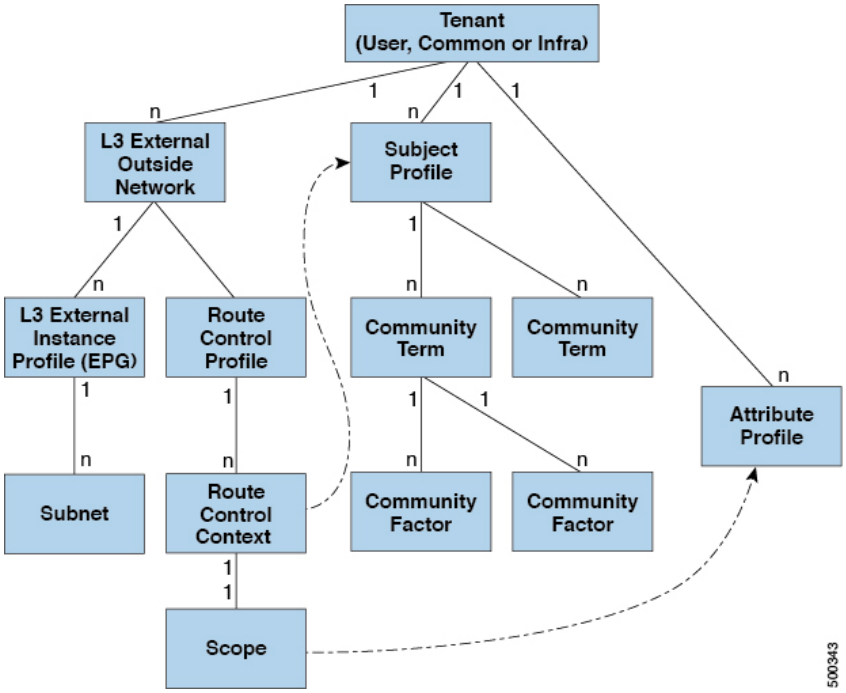
ルート集約では、多数の具体的なアドレスを 1 つのアドレスに置き換えることで、ルート テーブルが簡素化します。たとえば、10.1.1.0/24、10.1.2.0/24、10.1.3.0/24 は 10.1.0.0/16 に置き換えられます。ルート集約ポリシーにより、ボーダー リーフ スイッチとそのネイバー リーフ スイッチの間でルートを効率的に共有することができます。BGP、OSPF、あるいは EIGRP のルート集約ポリシーは、ブリッジ ドメインまたは中継サブネットに適用されます。OSPF では、エリア間ルー

ト集約と外部ルート集約がサポートされます。集約ルートはエクスポートされます。ファブリック内でのアドバタイズは行われません。上記の例では、ルート集約ポリシーが適用されると、EPGが 10.1.0 サブネットを使用し、10.1.0.0/16 の範囲全体がすべての隣接リーフ スイッチと共有されます。

（注） 同じリーフ スイッチで 2 つの L3Out ポリシーに OSPF を設定している場合（1 つはレギュラーで、もう 1 つはバックボーン）には、コンテキスト（VRF）内の全エリアに集約が適用されるため、片方の L3Out で設定されているルート集約ポリシーが両方の L3Out ポリシーに適用されます。

次の図に示すように、ルート制御プロファイルは、プレフィックス ベースおよびコミュニティ ベースの一致に基づいて、ルート マップを取得します。

図 *49* : ルート コミュニティ マッチング



ルート制御プロファイル（`rtctrtlProfile`）は、許可される対象を指定します。ルート制御コンテキストは一致対象を指定し、スコープは設定すべき対象を指定します。サブジェクト プロファイルには、コミュニティ マッチの仕様が含まれます。これは複数の L3 外部ネットワーク（`l3extOut`）で使用可能です。サブジェクト プロファイル（`SubjP`）には、それぞれ 1 つまたは複数のコミュニティ ファクタ（コミュニティ）を含む複数のコミュニティ タームを含めることができます。これにより、次のブール演算を指定することができます。

- 複数コミュニティ ターム間の論理的 `OR`
- 複数コミュニティ ターム間の論理的 `AND`

たとえば、北東と呼ばれるコミュニティ タームに、それぞれ多くのルートを含む複数のコミュニティが含まれているとします。また、南東という別のコミュニティ タームにも、さまざまなルートが多数含まれているとします。管理者は、そのどちらかあるいは両方を一致させることを選択できます。コミュニティ ファクタ タイプには、レギュラーまたは拡張を使用できます。拡張タイプのコミュニティ ファクタを使用する際には、仕様間の重複がないよう注意することが必要です。

ルート制御プロファイルのスコープ部分は、属性プロファイル（rtctrlAttrP）を参照して、適用すべき設定アクション（プリファレンス、ネクスト ホップ、コミュニティなど）を指定します。ルートを l3extOut から学習した場合は、ルートの属性を変更できます。

上の図は、l3extOut に rtctrtlProfile が含まれているケースを示しています。rtctrtlProfile はテナントの下にも配置できます。この例では、l3extOut に、自身をテナント下の rtctrtlProfile と関連付ける相互リーク関係ポリシー（L3extRsInterleakPol）が設定されています。これにより、rtctrtlProfile を複数の l3extOut 接続に再利用できるとともに、ファブリックによって BGP 属性が付与された OSPF からファブリックが学習したルートを追跡できるようになります（BGP はファブリック内で使用されます）。L3Out 下で定義された rtctrtlProfile の優先順位は、テナント下で定義された rtctrtlProfile よりも高くなります。

rtctrtlProfile には、組み合わせ可能およびグローバルという 2 つのモードがあります。デフォルトの組み合わせ可能モードでは、パーベイシブ サブネット（fvSubnet）および外部サブネット（l3extSubnet）に一致/設定メカニズムを組み合わせてルート マップをレンダリングします。グローバル モードはテナント内のすべてのサブネットに適用され、そのほかのポリシー属性の設定が無効になります。グローバル rtctrtlProfile では、明示的な（0/0）サブネットを定義しなくても、すべての動作が許可されます。グローバル rtctrtlProfile は、コミュニティやネクスト ホップといった異なるサブネット属性を使用してマッチングが行われる非プレフィックス ベースの一致ルールと一緒に使用されます。1 つのテナント下で複数の rtctrtlProfile ポリシーを設定できます。

rtctrtlProfile ポリシーによって、デフォルト インポートおよびデフォルト エクスポートのルート制御の拡張が可能になります。集約インポートあるいはエクスポート ルートを伴う L3Out には、サポート対象デフォルト エクスポート/デフォルト インポートおよびサポート対象 0/0 集約ポリシーを指定するインポート/エクスポート ポリシーを設定できます。すべてのルート（着信または発信）に rtctrtlProfile ポリシーを適用するには、一致ルールのないグローバル デフォルト rtctrtlProfile を定義します。

（注） 1 つのスイッチ上で複数の l3extOut 接続を設定することは可能ですが、スイッチは 1 つのルート マップしか持つことができないため、スイッチで設定されているすべての L3Out が同じ rtctrtlProfile を使用する必要があります。

プロトコル相互リーク/再配布ポリシーは、ACI ファブリック BGP ルートにリークした外部学習ルートを制御します。設定属性はサポートされています。これらのポリシーは、L3Out 単位、VRF 単位、ノード単位でサポートされます。相互リークポリシーは、L3Out 内のルーティング プロトコルによって学習されたルートに適用されます。現在、相互リーク/再配布ポリシーは、OSPF v2 および v3 でサポートされます。ルート制御ポリシー rtctrtlProfile は、相互リーク ポリシーによって消費される場合、グローバルとして定義する必要があります。

共有サービス コントラクトの使用

共有サービスを使用すると、テナントの分離とセキュリティ ポリシーを維持したままテナント間で通信できます。外部ネットワークへのルーテッド接続は、複数のテナントが使用する共有サービスの例です。

共有サービス コントラクトの設定時は、次のガイドラインに従ってください。

- さまざまなコンテキスト（VRF）にサブネットをエクスポートする共有サービスでは、EPG にサブネットを定義し、スコープを *advertised externally* および *shared between VRFs* に設定する必要があります。
- プライベートネットワークを適用しない場合、ブリッジ間ドメインのトラフィックにコントラクトは不要です。
- コンテキスト（VRF）が適用されていない場合でも、共有サービスのコンテキスト（VRF）間トラフィックにはコントラクトが必要です。
- 共有サービスを提供している間は、プロバイダーEPGのコンテキスト（VRF）は非強制モードにできません。
- 共有サービスは、重複しないサブネットのみでサポートされます。共有サービスのサブネットを設定するときは、以下のガイドラインに従ってください。
  - 共有サービス プロバイダーのサブネットは、ブリッジ ドメイン下ではなく EPG 下で設定します。
  - 同じコンテキストを共有する EPG で設定されたサブネットは、統合および重複してはなりません。
  - あるコンテキストから他のコンテキストへ漏れたサブネットは統合および重複してはなりません。
  - 複数のコンシューマ ネットワークからあるコンテキストへ漏れたサブネットまたはその逆で漏れたサブネットは統合および重複してはなりません。

（注） 2人のコンシューマが誤って同じサブネットに設定されている場合は、両方のサブネットの設定を削除してこの状態からリカバリし、その後サブネットを正しく再設定します。

- プロバイダー コンテキストで共有サービスを `AnyToProv` に設定しないでください。APIC はこの設定を拒否し、エラーが発生します。
- インバンド EPG とアウトオブバンド EPG の間でコントラクトが設定される場合、以下の制限が適用されます。
  - 両方の EPG は同じコンテキスト（VRF）にする必要があります。
  - フィルタは、着信方向のみに適用されます。

- レイヤ 2 フィルタはサポートされません。
- QoS は、インバンド レイヤ 4 ～ レイヤ 7 のサービスには適用されません。
- 管理統計情報は利用できません。
- CPU 宛てトラフィックの共有サービスはサポートされません。

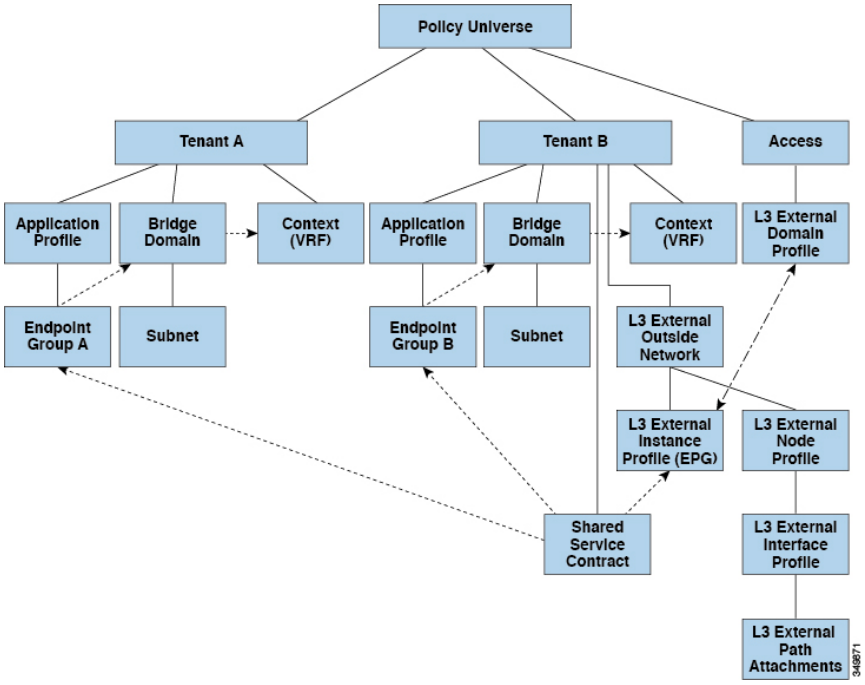
共有レイヤ 3 Out

共有レイヤ 3 Out 設定は、外部ネットワークへのルーテッド接続を共有サービスとして提供します。`l3extInstP` EPG は、外部ネットワークへのルーテッド接続を提供します。これは、任意のテナント（user、common、infra、または mgmt.）の共有サービスとしてプロビジョニングできます。リリース 1.2(1x) より前では、この設定は user テナントと common テナントでのみサポートされていました。任意のテナントの EPG が、`l3extInstP` EPG がファブリック内のどこにプロビジョニングされているかには関係なく、共有サービス コントラクトを使用してその `l3extInstP` EPG に接続できます。これにより、外部ネットワークへのルーテッド接続のプロビジョニングが簡単になります。複数のテナントが、外部ネットワークへのルーテッド接続用に単一の `l3extInstP` EPG を共有できます。`l3extInstP` EPG を共有すると、単一の共有 `l3extInstP` EPG を使用する EPG の数には関係なくスイッチ上で使用されるセッションは 1 つのみであるため、より効率的になります。

（注） l3extInstP EPG 共有サービス コントラクトを使用するすべてのスイッチは、APIC 1.2（1x）およびスイッチ 11.2（1x）の各リリース以降で使用可能なハードウェアおよびソフトウェアのサポートを必要とします。詳細については、『Firmware Management Guide and Release Notes』というマニュアルを参照してください。

次の図は、共有 l3extInstP EPG 用に設定された主なポリシー モデル オブジェクトを示しています。

図 *50*：共有レイヤ *3 Out* ポリシー モデル



共有レイヤ 3 Out 設定について、以下のガイドラインと制限事項に注意してください。

- テナント制限なし：テナント A と B は、任意の種類のテナント（user、common、infra、mgmt.）です。共有 l3extInstP EPG がテナント common にある必要はありません。
- EPG の柔軟な配置：上の図の EPG A と EPG B は異なるテナントにあります。EPG A と EPG B で同じブリッジ ドメインとコンテキストを使用することはできますが、それは必須ではありません。EPG A と EPG B は異なるブリッジ ドメインおよび異なるコンテキストにありますが、同じ l3extInstP EPG を共有しています。
- サブネットは、*private*、*public*、または *shared* です。レイヤ 3 Outside 外部ネットワークのコンシューマ EPG またはプロバイダー EPG は *shared* に設定されている必要があります。レイヤ 3 Outside 外部ネットワークにエクスポートされるサブネットは、*public* に設定されている必要があります。
- 共有サービス コントラクトは、共有レイヤ 3 Out サービスを提供する l3extInstP EPG が含まれているテナントからエクスポートされます。共有サービス コントラクトは、共有サービスを使用する EPG が含まれているテナントにインポートされます。
- 共有 L3 Out では禁止コントラクトを使用しないでください。この設定はサポートされません。
- l3extInstP は共有サービス プロバイダーとしてサポートされますが、l3extInstP 以外のコンシューマのみに限定されます（Layer3Out EPg = l3extInstP である場合）。

- 中継ルーティングは共有サービスではサポートされません。つまり、異なるVRF内の2つのレイヤ3 Outが共有サービス機能を使用して互いに通信することはできません。
- トラフィック フラップ：`l3instP` EPGが、`l3instP` サブセットのスコープ プロパティを共有ルート制御（*shared-rctrl*）または共有セキュリティ（*shared-security*）に設定して外部サブネット0.0.0.0/0を使用して設定されると、コンテキスト（VRF）はグローバル `pcTag` を使用して再配置されます。これにより、そのVRF内のすべての外部トラフィックがフラップされます（VRFがグローバル `pcTag` を使用して再配置されるため）。
- 共有レイヤ3 Outのプレフィックスは一意である必要があります。同じコンテキスト（VRF）の同じプレフィックスを使用した、複数の共有レイヤ3 Out設定は動作しません。VRFにリークする外部サブネット（外部プレフィックス）が一意であることを確認してください（同じ外部サブネットが複数の `l3instP` に属することはできません）。プレフィックスprefix1を使用したレイヤ3 Outside設定（たとえば、L3Out1）と、同様にプレフィックスprefix1を使用した2番目のレイヤ3 Outside設定（たとえば、L3Out2）が同じコンテキスト（VRF）に属すると、動作しません（導入されるpcTagは1つのみであるため）。
- 許可されないトラフィック：無効な設定で、共有ルート制御（shared-rtctrl）に対する外部サブネットのスコープが、共有セキュリティ（shared-security）に設定されているサブネットのサブセットとして設定されている場合、トラフィックは許可されません。たとえば、以下の設定は許可されません。
  - *shared rtctrl*：10.1.1.0/24, 10.1.2.0/24
  - *shared security*：10.1.0.0/16

  この場合、10.1.1.0/24および10.1.2.0/24の各プレフィックスがドロップ ルールを使用してインストールされているため、宛先IP 10.1.1.1を使用して非境界リーフに到達するトラフィックはドロップされます。トラフィックは許可されません。そのようなトラフィックは、shared-rtctrlプレフィックスをshared-securityプレフィックスとしても使用するように設定を修正することで、有効にすることができます。
- 不注意によるトラフィック フロー：次の設定シナリオを避けることで、不注意によるトラフィック フローを予防します。
  - **ケース 1** 設定の詳細：
    - コンテキスト（VRF）1を使用したレイヤ3 Outside設定（たとえばL3Out1）はprovider1と呼ばれます。
    - コンテキスト（VRF）2を使用した2番目のレイヤ3 Outside設定（たとえばL3Out2）はprovider2と呼ばれます。
    - L3Out1 VRF1はデフォルトルートをインターネットにアドバタイズします=0.0.0.0/0 = *shared-rtctrl*、*shared-security*。
    - L3Out2 VRF2は特定のサブネットをDNSおよびNTPにアドバタイズします=192.0.0.0/8 = *shared-rtctrl*。
    - L3Out2 VRF2には特定のサブネット192.1.0.0/16があります= *shared-security*。

- **バリエーション A**：EPG トラフィックが複数のコンテキスト（VRF）に向かいます。
  - EPG1 と L3Out1 の間の通信は *allow_all* コントラクトによって制御されます。
  - EPG1 と L3Out2 の間の通信は *allow_all* コントラクトによって制御されます。

    結果：EPG1 から L3Out2 へのトラフィックも 192.2.x.x に向かいます。
- **バリエーション B**：EPG は 2 番目の共有レイヤ 3 Out の *allow_all* コントラクトに従います。
  - EPG1 と L3Out1 の間の通信は *allow_all* コントラクトによって制御されます。
  - EPG1 と L3Out2 の間の通信は *allow_icmp* コントラクトによって制御されます。

    結果：EPG1 -> L3Out2 -> 192.2.x.x のトラフィックは *allow_all* コントラクトに従います。

- **ケース 2** 設定の詳細：
  - レイヤ 3 Out インスタンス プロファイル（l3instP）は、1 つの共有プレフィックスとその他の非共有プレフィックスを持っています。
  - `src = non-shared` で到達するトラフィックは、EPG に向かうことが許可されます。
    - **バリエーション A**：意図しないトラフィックが EPG を通過します。
      - レイヤ 3 Out（l3instP）EPG トラフィックは、以下のプレフィックスを持っているレイヤ 3 Out を通過します。
        - `192.0.0.0/8 = import-security、shared-rtctrl`
        - `192.1.0.0/16 = shared-security`
        - EPG は `1.1.0.0/16 = shared` となっています。

          結果：192.2.x.x からのトラフィックも EPG に向かいます。
    - **バリエーション B**：意図しないトラフィックが EPG を通過します。共有レイヤ 3 Out に到達するトラフィックは、コンテキスト（VRF）に応じて通過できます。
      - 共有レイヤ 3 Out のコンテキスト（VRF）は、`pcTag = prov vrf` の EPG と *allow_all* のコントラクトを持っています。
      - EPG は `<subnet> = shared` となっています。

        結果：レイヤ 3 Out に到達するトラフィックは EPG を通過することができます。

双方向フォワーディング検出

双方向フォワーディング検出（BFD）を使用して、ピアリング ルータの接続をサポートするように設定されたACIファブリック境界リーフ スイッチ間の転送パスのサブセカンド障害検出時間を提供します。

BFD は、次のような場合に特に役立ちます。

- ルータ同士の間に直接的な接続がない場合に、レイヤ 2 デバイスまたはレイヤ 2 クラウド経由でピアリング ルータが接続されているとき。転送パスに障害があっても、ピア ルータにはそれがわからない可能性があります。プロトコルの制御に利用できるメカニズムは hello タイムアウトだけですが、タイムアウトまでには数十秒、さらには数分の時間がかかる場合があります。BFD では、障害を 1 秒未満で検出することが可能です。
- 信頼できる障害検出に非対応の物理メディア（共有イーサネットなど）経由でピアリング ルータが接続されているとき。この場合も、ルーティング プロトコルは、時間のかかる hello タイマーに頼るしかありません。
- 1 組のルータの間で多くのプロトコルが実行されているとき、各プロトコルは、独自のタイムアウトでリンク障害を検出する独自の hello メカニズムを持っています。BFD は、すべてのプロトコルに均一のタイムアウトを指定し、それによってコンバージェンス時間の一貫性を保ち、予測可能にします。

次に示す BFD の設定のガイドラインおよび制限事項に従ってください。

- リーフ スイッチ/スパイン スイッチ間の BFD はサポートされません。
- VPC ピア間の BFD はサポートされません。
- マルチホップ BFD はサポートされません。
- ループバック アドレス ピアでの iBGP 上の BFD はサポートされません。
- インターフェイス ポリシーで BFD サブインターフェイス最適化を有効化できます。このフラグを 1 つのサブインターフェイスに立てることにより、その物理インターフェイス上のすべてのサブインターフェイスの最適化が有効になります。
- BGP プレフィクス ピアの BFD はサポートされません。

データ プレーン ポリシング

データ プレーン ポリシング（DPP）を使用して、ACI ファブリック アクセス インターフェイスの帯域幅使用量を管理します。DPP ポリシーは出力トラフィック、入力トラフィック、またはその両方に適用できます。DPP は特定のインターフェイスのデータ レートを監視します。データ レートがユーザ設定値を超えると、ただちにパケットのマーキングまたはドロップが発生します。ポリシングではトラフィックがバッファリングされないため、伝搬遅延への影響はありません。トラフィックがデータ レートを超えた場合、ACI ファブリックは、パケットのドロップか、パケット内 QoS フィールドのマーキングのどちらかを実行できます。

（注） 出力データ プレーン ポリサーは、スイッチ仮想インターフェイス（SVI）ではサポートされません。

DPP ポリシーは、シングルレート、デュアルレート、カラー対応のいずれかになります。シングルレート ポリシーは、トラフィックの認定情報レート（CIR）を監視します。デュアルレート ポリサーは、CIR と最大情報レート（PIR）の両方を監視します。また、システムは、関連するバースト サイズもモニタします。指定したデータ レート パラメータに応じて、適合（グリーン）、超過（イエロー）、違反（レッド）の 3 つのカラー、つまり条件が、パケットごとにポリサーによって決定されます。

通常、DPP ポリシーは、サーバやハイパーバイザなどの仮想または物理デバイスへの物理または仮想レイヤ 2 接続に適用されます。ルータについてはレイヤ 3 接続で適用されます。リーフ スイッチ アクセス ポートに適用された DPP ポリシーは、ACI ファブリックのファブリック アクセス（`infraInfra`）部分で設定します。設定はファブリック管理者が行う必要があります。ボーダー リーフ スイッチ アクセス ポート（`l3extOut` または `l2extOut`）上のインターフェイスに適用される DPP ポリシーは、ACI ファブリックのテナント（`fvTenant`）部分で設定します。テナント管理者がその設定を行うことができます。

各状況に設定できるアクションは 1 つだけです。たとえば、DPP ポリシーを最大 200 ミリ秒のバーストで、256,000 bps のデータ レートに適合させることが可能です。この場合、システムは、このレートの範囲内のトラフィックに対して適合アクションを適用し、このレートを超えるトラフィックに対して違反アクションを適用します。カラー対応ポリシーは、トラフィックが以前にカラーによってすでにマーキングされているものと見なします。次に、このタイプのポリサーが実行するアクションの中で、その情報が使用されます。

IPv6 のサポート

ACI ファブリックは、インバンドおよびアウトオブバンド インターフェイス、テナント アドレッシング、コントラクト、共有サービス、ルーティング、レイヤ 4 ～レイヤ 7 のサービス、トラブルシューティングに関して次の IPv6 をサポートします。

- IPv6 アドレス管理、パーベイシブ ソフトウェア仮想インターフェイス（SVI）、ブリッジ ドメイン サブネット、外部ネットワーク外部インターフェイス アドレス、ロード バランサまたは侵入検知などの共有サービスへのルート。
- ルータ アドバタイズメント（RA）とルータ要求（RS）と呼ばれる ICMPv6 メッセージを使用したネイバー探索、および重複アドレス検出（DAD）。
- ステートレス アドレス自動設定（SLAAC）および DHCPv6
- ブリッジ ドメインの転送。
- トラブルシューティング（「トラブルシューティング」の章のアトミック カウンタ、SPAN、iping6、itraceroute のトピックを参照してください）。

- IPv4 のみ、IPv6 のみ、またはインバンドおよびアウトオブバンド インターフェイスのデュアル スタック設定。

現在の ACI ファブリック Ipv6 導入に関する制限は以下のとおりです。

- マルチキャスト リスナー検出（MLD）スヌーピングはサポートされません。
- IPv6 マネジメントでは、スタティック アドレスのみ許可されます。ダイナミック IPv6 プールは IPv6 管理ではサポートされません。
- ファブリック内では IPv6 トンネル インターフェイス（Intra-Site Automatic Tunnel Addressing Protocol、6to4 など）はサポートされません。ファブリック経由で実行される IPv6 トンネル トラフィックはファブリックに対して透過的です。

ACIファブリックインターフェイスは、リンクローカル、グローバルユニキャスト、マルチキャスト IPv6 アドレスで設定できます。

（注） このマニュアルでは多くの例で IPv4 アドレスを使用していますが、IPv6 アドレスも使用できます。

グローバルユニキャストアドレスはパブリックインターネット全範囲でルーティングできます。これはルーティング ドメイン内ではグローバルに一意のアドレスとなります。リンクローカルアドレス（LLA）にはリンクローカル スコープがあり、リンク（サブネット）上では一意です。LLA をサブネット間でルーティングすることはできません。これらは、ネイバー探索や OSPF などの制御プロトコルによって使用されます。マルチキャスト アドレスは、複数のエンドポイントにパケットを配信するために、ネイバー探索プロトコルなどの IPv6 制御プロトコルによって使用されます。これらは、プロトコル コンポーネントによって自動的に生成され、設定することはできません。

グローバル ユニキャスト アドレス

管理者は、フル 128 ビット IPv6 グローバル ユニキャスト アドレスを、圧縮形式または非圧縮形式で、インターフェイス上に手動で指定できます。たとえば、「2001:0000:0000:0001:0000:0000:0000:0003」または「2001:0:0:1:0:0:0:3」、「2001:0:0:1::3」といった形式でアドレスを指定できます。ACI ファブリック ネーミング プロパティでは、IPv6 アドレスは常に圧縮形式で表されます。上記の例での相対名は「2001:0:0:1::3」となります。管理者は、必要に応じてアドレスのマスク長を任意に選択できます。

また、管理者は、ACI ファブリック IPv6 グローバル ユニキャスト アドレスを EUI-64 形式で指定できます。RFC2373 で指定されているように、拡張固有識別子（EUI）により、ホストは一意の 64 ビット IPv6 EUI-64 インターフェイス識別子を自身に割り当てることができます。EUI-64 形式の IPv6 アドレスは、インターフェイス MAC アドレスを 128 ビット IPv6 グローバル ユニキャスト アドレス内に組み込むことによって得られます。IPv6 にこの機能があるため、手動設定や DHCP は必要なくなります。EUI-64 形式で指定されるブリッジ ドメインまたはレイヤ 3 インターフェイスの IPv6 アドレスは、次のようにして形成されます。<IPv6 prefix>::/<mask>/eui64（マスクは 64 以下）。たとえば、管理者が「2002::/64/eui64」と指定すると、スイッチは

「2002::222:bdff:fef8:19ff/64」というアドレスを割り当てます。スイッチは、インターフェイス MAC アドレスを使用して EUI-64 アドレスを作成します。形成された IPv6 アドレスは、`ipv6If` オブジェクトの `operAddr` フィールドに格納されます。

（注） EUI-64 形式はパーベイシブ ブリッジ ドメインとレイヤ 3 インターフェイス アドレスにしか使用できません。外部サーバ アドレスなどファブリックの他の IP フィールドや、DHCP リレーには使用できません。

ブリッジ ドメイン サブネットおよびレイヤ 3 外部インターフェイス IP アドレスは、マスクが /1 から /127 までの範囲内にある IPv6 グローバル アドレスとして設定できます。ブリッジ ドメインには複数の IPv4 および IPv6 サブネットを含めることができます。同じ L3 外部インターフェイス上で IPv4 アドレスと IPv6 アドレスをサポートするには、複数のインターフェイス プロファイルを管理者によって作成します。EPG または外部 EpP がスイッチに導入されると、同等のブリッジ ドメイン/L3 インターフェイスに手動で設定されたリンクローカル アドレスまたはサブネット/アドレス フィールドの IPv6 アドレスのプレゼンスにより、スイッチ内に `ipv6If` インターフェイスが作成されます。

リンクローカル アドレス

1 つのインターフェイスに 1 つのリンクローカル アドレス（LLA）を割り当てることができます。LLA は、自動生成するかまたは管理者が設定できます。デフォルトでは、スイッチによって ACI リンクローカルアドレスが EUI-64 形式で自動生成されます。自動生成されたリンクローカル アドレスをスイッチで上で生成させるには、管理者が少なくとも 1 つのグローバル アドレスをインターフェイス上で設定する必要があります。自動生成されたアドレスは、`ipv6If` MO の `operllAddr` フィールドに保存されます。パーベイシブ SVI の場合、使用される MAC アドレスは、設定済みのインターフェイス MAC アドレスと同じです。他のタイプのインターフェイスではバックプレーンの MAC アドレスが使用されます。管理者は、フル 128 ビット IPv6 リンクローカル アドレスを、圧縮形式または非圧縮形式で、インターフェイス上に手動で指定できます。

（注） スイッチ ハードウェア テーブルは、コンテキスト（VRF）1 つあたり 1 つの LLA に限定されます。

パーベイシブ ブリッジ ドメインそれぞれに 1 つの IPv6 リンクローカル アドレスを設定できます。このアドレスは、管理者によって設定できます。あるいは、1 つも提供されていない場合にはスイッチによって自動設定することもできます。自動設定の場合、スイッチは、MAC アドレスを IPv6 アドレスにエンコードして一意のアドレスを作成する Modified EUI-64 形式で、アドレスを形成します。パーベイシブ ブリッジ ドメインは、すべてのリーフ ノードで 1 つのリンクローカル アドレスを使用します。

外部 SVI および VPC メンバーでは、すべてのリーフ ノードにそれぞれ固有のリンクローカル アドレスがあります。リンクローカル アドレスは、インターフェイスのライフサイクル中いつでも、手動（ゼロ以外の、手動で指定されたリンクローカル アドレス）または自動（指定リンクローカル アドレスを手動でゼロに設定することによる）に変更できます。管理者が指定するリン

クローカル アドレスは、IPv6 リンクローカル形式（FE80::/10）に準拠する必要があります。IPv6 インターフェイス MO（ipv6If）は、インターフェイスで最初のグローバル アドレスが作成されるときか、管理者が手動でリンクローカル アドレスを設定するときの、どちらか早いほうのタイミングで、スイッチ上で作成されます。管理者が指定したリンクローカル アドレスは、論理モデルでは、ブリッジ ドメインとレイヤ 3 インターフェイス オブジェクトの llAddr プロパティで表されます。スイッチによって使用されるリンクローカルアドレス（llAddr から作成、または llAddr がゼロである場合に自動生成）は、対応する ipv6If オブジェクトのプロパティ operLlAddr で表されます。リンクローカルアドレスの重複などの運用上のリンクローカルアドレス関連エラーは、重複アドレス検出プロセス中にスイッチによって検出され、ipv6If オブジェクトの operStQual フィールドに記録されるか、必要に応じて障害を発生させます。llAddr フィールド以外では、アドレスのルーティングができないため、リンクローカル アドレス（FE80::/10）が APIC 内の他の IP アドレス フィールド（外部サーバ アドレスまたはブリッジ ドメイン サブネットなど）における有効なアドレスとなることはできません。

スタティック ルート

ACI IPv6 スタティック ルートは、アドレスおよびプレフィクス形式の設定の違いを除き、IPv4 の場合と同様にサポートされます。IPv6 スタティック ルート モジュールが扱う一般的なスタティック ルートのタイプは次のとおりです。

- ローカル ルート：インターフェイスで設定された任意の /128 アドレスが、CPU を参照するローカル ルートに向かいます。
- 直接ルート：パーベイシブ BD で設定されたアドレスの場合は、ポリシー要素が、スパイン上の IPv4 プロキシ トンネルの宛先を参照するサブネット ルートをプッシュします。非パーベイシブ レイヤ 3 外部インターフェイスで設定されたアドレスの場合は、IPv6 マネージャ モジュールが、CPU を参照するサブネット ルートを自動的にプッシュします。
- PE からプッシュされるスタティック ルート：外部接続に使用されます。このようなルートのネクスト ホップ IPv6 アドレスは、外部ルータ上の直接接続サブネット上に存在するか、直接接続サブネット上で実際のネクスト ホップに解決可能な再帰的ネクスト ホップとして存在することができます。IFC モデルではインターフェイスをネクスト ホップとして扱うことができないので注意してください（ただし、スイッチではサポートされています）。テナント間で共有サービスを有効化するために使用される場合、スタティック ルートが存在する共有サービス用のネクスト ホップは、共有サービス コンテキスト（VRF）内にあります。これは、入力リーフ スイッチ上にルートがインストールされるテナント コンテキストとは異なります。

ネイバー探索

IPv6 ネイバー探索（ND）は、ノードのアドレスの自動設定、リンク上の他のノードの探索、他のノードのリンク層アドレスの判別、重複アドレスの検出、使用可能なルータと DNS サーバの検出、アドレス プレフィックスの探索、および他のアクティブなネイバー ノードへのパスに関する到達可能性情報の維持を担当します。

ND 固有のネイバー要求/ネイバー アドバタイズメント（NS/NA）およびルータ要求/ルータ アドバタイズメント（RS/RA）パケット タイプは、物理、L3 Sub-if、および SVI（外部およびパーベイシブ）を含むすべての ACI ファブリックのレイヤ 3 インターフェイスでサポートされます。RS/RA パケットはすべての L3 インターフェイスの自動設定に使用されますが、パーベイシブ SVI の場合にのみ設定できます。ACI のブリッジ ドメイン ND は常にフラッド モードで動作します。ユニキャスト モードはサポートされません。

ACI ファブリック ND サポートに含まれるもの：

- インターフェイス ポリシー（`nd:IfPol`）は、NS/NA メッセージに関する ND タイマーと動作を制御します。
- ND プレフィックス ポリシー（`nd:PfxPol`）は、RA メッセージを制御します。
- ND の IPv6 サブネット（fv:Subnet）の設定。
- 外部ネットワークの ND インターフェイス ポリシー。
- 外部ネットワークの設定可能 ND サブネットおよびパーベイシブ ブリッジ ドメインの任意サブネット設定はサポートされません。

設定可能なオプションは次のとおりです。

- 隣接関係
  - 設定可能な静的隣接関係：（<vrf、L3Iface、ipv6 アドレス> --> MAC アドレス）
  - 動的隣接関係：NS/NA パケットの交換によって学習
- インターフェイス単位
  - ND パケットの制御（NS/NA）
    - ネイバー要求間隔
    - ネイバー要求再試行回数
  - RA パケットの制御
    - RA の抑制
    - RA MTU の抑制
    - RA 間隔、RA 最小間隔、再送信時間
- プレフィックス単位（RA でアドバタイズ）の制御
  - ライフタイム、優先ライフタイム
  - プレフィックス制御（自動設定、リンク対象）

重複アドレス検出

重複アドレス検出（DAD）は、設定中のアドレスをすでに使用しているリンク上の任意の他ノードを検出します。DAD は、リンクローカル アドレスとグローバル アドレスの両方に実行されます。設定済みのアドレスそれぞれが、次の DAD 状態を維持します。

- `NONE`：DAD を試みる前の、アドレスが最初に作成されたときの状態。
- `VALID`：重複アドレスを検出することなくアドレスが正常に DAD プロセスを渡したことを表す状態。
- `DUP`：アドレスがリンク上で重複として検出されたことを表す状態。

DAD の状態が VALID である限り、どの設定済みアドレスでも IPv6 トラフィックの送受信に使用できます。

ステートレス アドレス自動設定（SLAAC）および DHCPv6

次のホスト設定がサポートされます。

- SLAAC のみ
- DHCPv6 のみ
- SLAAC + DHCPv6 ステートレスは、アドレス設定には SLAAC のみ使用しますが、DNS やその他の機能には DHCPv6 を使用します。

DHCP リレーでは IPv6 アドレスがサポートされます。コンテキスト間の DHCPv6 リレー（VRF）および VLAN と VXLAN との間の DHCP リレーがサポートされています。DHCPv4 は DHCPv6 と連動します。

第 7 章

ACI トランジット ルーティング、ルート ピアリング、および EIGRP サポート

この章の内容は、次のとおりです。

ACI トランジット ルーティング

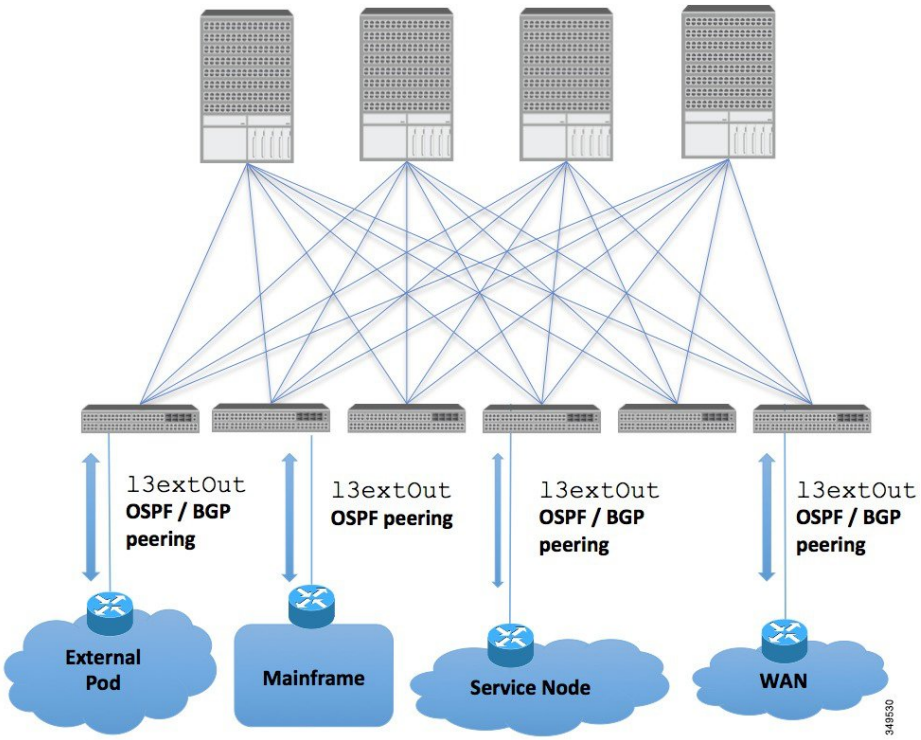
ACI ファブリックは、境界ルータが他のドメインとの双方向再配布を実行できるようにする、トランジット ルーティングをサポートします。トランジット再配布をブロックする ACI ファブリックの以前のリリースのスタブ ルーティング ドメインとは異なり、双方向再配布では、1 つのルーティング ドメインから別のルーティング ドメインにルーティング情報を渡します。そのような再配布により、ACI ファブリックはさまざまなルーティング ドメイン間の完全な IP 接続を提供します。これにより、ルーティング ドメイン間のバックアップ パスを有効にすることで冗長接続を提供することもできます。

最適でないルーティングや、ルーティング ループというさらに重大な問題を回避するように、トランジット再配布ポリシーを設計してください。通常、トランジット再配布は、元のトポロジとリンク状態情報を維持せず、ディスタンス ベクター方式で外部ルートを再配布します（リンクステート プロトコルの場合でもルートはベクター プレフィックスと関連距離としてアドバタイズされます）。このような状況では、ルータが想定外のルーティング ループを形成して、パケットを宛先に配信できなくなる可能性があります。

トランジット ルーティングの使用例

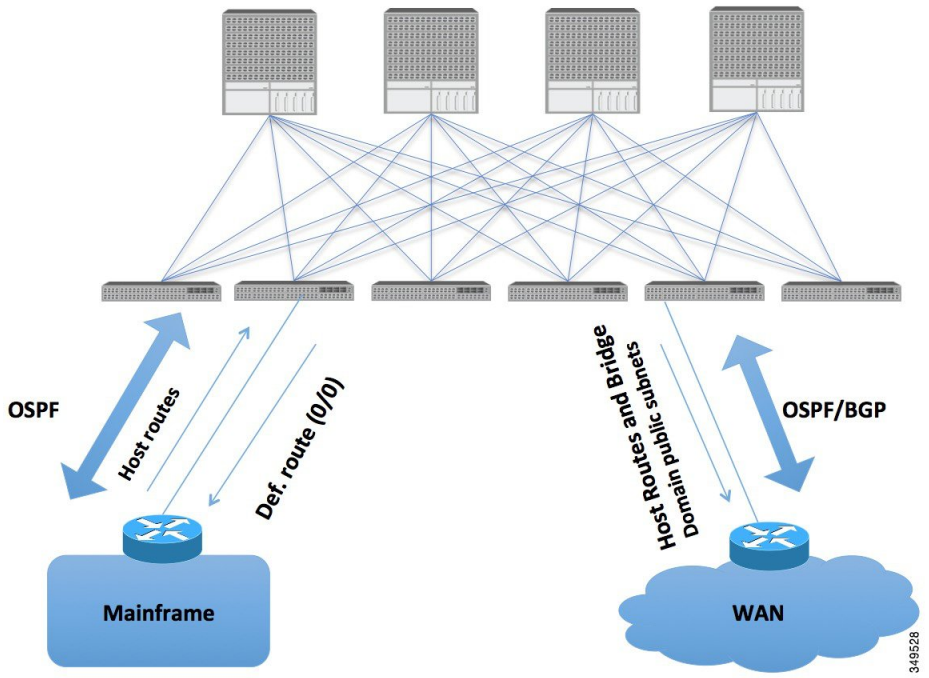
外部ポッド、メインフレーム、サービス ノード、WAN ルータなどの複数のレイヤ 3 ドメインがACIファブリックとピアリングして、それらの間のトランジット機能を提供することができます。

図 *51*：レイヤ *3* ドメイン間のトランジット ルーティング



メインフレームは、論理パーティション（LPAR）および仮想 IP アドレッシング（VIPA）の要件に対応する標準 IP ルーティング プロトコルを実行する IP サーバとして機能します。

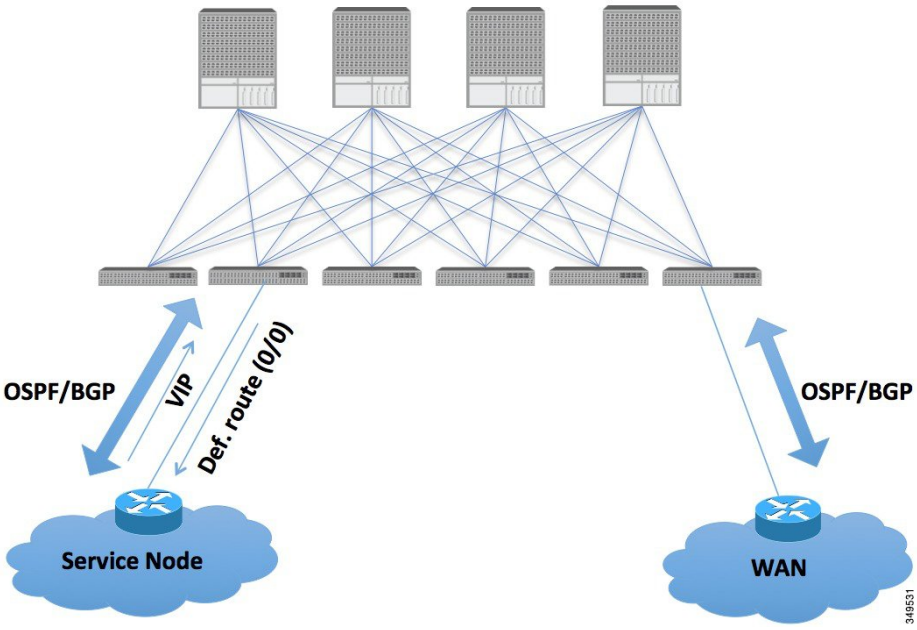
図 *52*：メインフレームのトランジット接続



メインフレームでは、ACI ファブリックが WAN ルータを介した外部ドメインおよびファブリック内の East-West トラフィックのトランジット ドメインである必要がありますが、ホスト ルートがファブリックにプッシュされ、それらのルートがファブリック内および外部インターフェイスに配布されます。

サービス ノードは ACI ファブリックとピアリングし、外部 WAN インターフェイスに再配布される仮想 IP（VIP）ルートをアドバタイズすることができます。

図 *53*：サービス ノードのトランジット接続

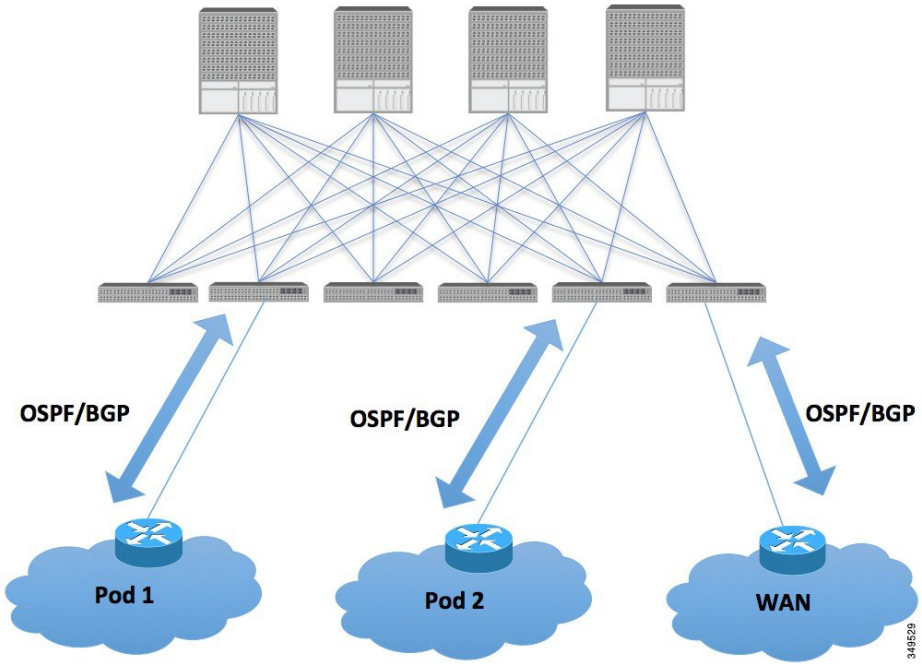


VIP は、特定のサイトやサービスの外部向けの IP アドレスです。VIP は、サービス ノードの背後にある 1 つ以上のサーバまたはノードに関連付けられています。

ACI ファブリックは、外部接続および POD（ポッド）間の相互接続のトランジットとして機能します。クラウドのプロバイダーは、顧客データ センター内に管理対象リソース POD を導入でき

ます。責任分界点は、OSPF および BGP とファブリックとのピアリングが行われている L3Out にすることができます。

図 *54*：複数ポッドのトランジット接続



このようなシナリオでは、ポリシーは責任分界点で管理され、ACI ポリシーを設定する必要はありません。

L4-L7 ルート ピアリングは、ファブリックをトランジットとして使用する特殊なケースであり、ACI ファブリックは他の POD（ポッド）に対する OSPF および BGP のトランジット ドメインの役目を果たします。ルート ピアリングは、L4-L7 サービス デバイスで OSPF および BGP のピアリングを設定し、接続先の ACI リーフ ノードとルートを交換できるようにするために使用されます。ルート ピアリングの一般的な使用例として、SLB VIP が OSPF および iBGP を介して ACI ファブリック外のクライアントにアドバタイズされるルート ヘルス インジェクションがあります。このシナリオの設定のウォークスルーについては、付録 H を参照してください。

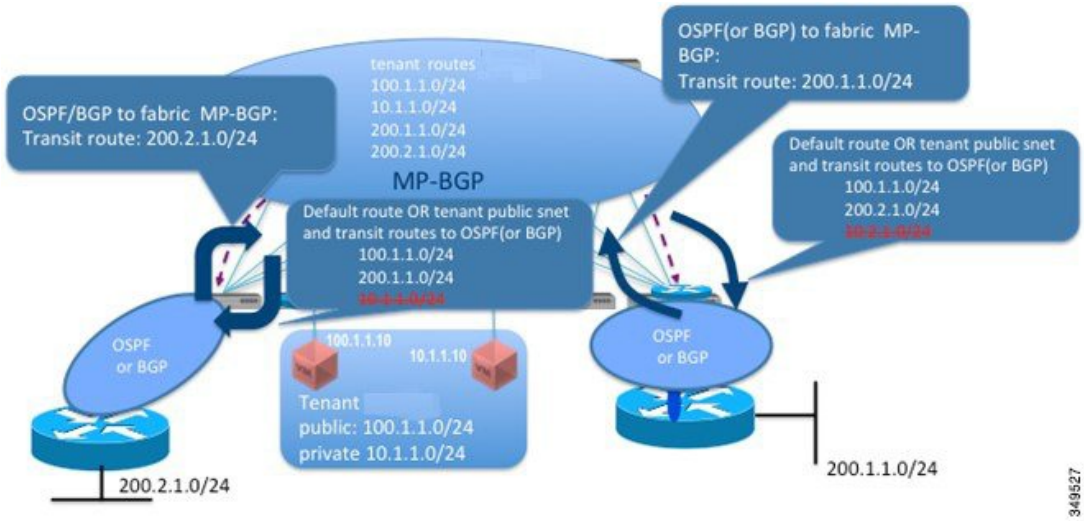
ACI ファブリック ルート ピアリング

ファブリックとのレイヤ 3 接続およびピアリングは、レイヤ 3 外部ネットワーク（`l3extOut`）インターフェイスを使用して設定されます。ルート再配布と着信/発信フィルタリング ルールを伴ったピアリング プロトコル設定は、`l3extOut` と関連付けられます。ACI ファブリックは、外部ピアへの大規模ルータとしてではなく、別々のレイヤ 3 ドメイン間の中継として出現します。1 つの `l3extOut` でのピアリングの考慮事項が他の `l3extOut` ポリシーのピアリングの考慮事項に影響するわけではありません。ACI ファブリックは、MP-BGP を使用して外部ルートをファブリック内に配布します。

ルートの再配布

外部ピアからの着信ルートは、着信フィルタリング ルールに基づき、MP-BGP を使用して ACI ファブリックに再配布されます。WAN接続の場合は、中継ルートまたは外部ルートのいずれかである可能性があります。MP-BGP は、テナントが配置されているすべてのリーフ（他のボーダー リーフを含む）にルートを配付します。

図 *55*：ルートの再配布



着信ルートのフィルタリングルールは、外部ピアによって`l3extOut`インターフェイス上のファブリックにアドバタイズされるルートのサブセットを選択します。インポート フィルタのルート マップは、プレフィックスベースのEPGでプレフィクスを使用することによって生成されます。インポート フィルタ リストはMP-BGPにのみ関連付けられて、ファブリックに配付されるプレフィクスを制限します。また、セット アクションをインポート ルート マップに関連付けることもできます。

発信方向では、管理者が、デフォルトルートまたは中継ルートおよびブリッジ ドメインパブリック サブネットをアドバタイズできます。デフォルト ルート アドバタイズメントが有効でない場合、管理者の設定に従って、発信ルート フィルタリングがルートを選択的にアドバタイズします。

現在、ルートマップは、外部ルータにアドバタイズされるブリッジ ドメインのパブリック サブネットを示すため、テナント単位のプレフィックスリストとともに作成されます。また、プレフィックスリストはすべての中継ルートが外部ルータにアドバタイズされるように作成する必要があります。中継ルートのプレフィックスリストは管理者によって設定されます。デフォルトの動作では、外部ルータへの中継ルートのアドバタイズメントはすべて拒否されます。

中継ルートに関連付けられたルート マップには次のオプションを使用できます。

- *Permit-all*：すべての中継ルートを外部に再配布およびアドバタイズします。
- *Match prefix-list*：中継ルートのサブセットのみ外部に再配布およびアドバタイズされます。

- *Match prefix-list* および *set action*：set action を中継ルートのサブセットと関連付けることで、特定の属性を持つルートへのタグ付けが可能になります。

ブリッジ ドメイン パブリック サブネットと中継ルート プレフィクスは、プレフィクス リストとしては別々であっても、異なるシーケンス番号を付けて単一のルート マップに統合することができます。中継ルートおよびブリッジ ドメイン パブリック サブネットは同じプレフィクスを持つ想定にはなっていないため、プレフィックス リスト マッチは互いに排反します。

プロトコルによるルート ピアリング

ルート ピアリングは、BGP と OSPF を組み合わせたケースとして、およびスタティック ルートとともに、プロトコル ベースで設定できます。

OSPF	BGP
接続を有効にして冗長性を提供するために、さまざまなホスト タイプが OSPF を必要とします。これらには、たとえばファブリック内および WAN へのレイヤ 3 中継として ACI を使用するサービス ノード、外部ポッド、メインフレームなどがあります。このような外部デバイスは、OSPF を実行している非境界リーフを介してファブリックとピアリングします。理想的には、OSPF エリアを Not-So-Stubby Area（NSSA）つまり完全スタブエリアとして設定し、デフォルト ルートは受信するが全域ルーティングには参加しないようにします。既存の導入において管理者がルーティング設定の変更を望まない場合は、スタブエリアの設定は必須ではありません。 2 つのファブリック リーフの間に OSPF 隣接関係が確立されることはありません。ただし、同じ外部 SVI インターフェイスを共有している場合は除きます。	外部 POD とサービス ノードはファブリックとの BGP ピアリングを使用できます。BGP ピアは `l3extOut` に関連付けられており、`l3extOut` ごとに複数の BGP ピアを設定することができます。BGP ピアには、OSPF、EIGRP、接続されたインターフェイス、スタティック ルート、またはループバック経由で到達できます。外部ルータとのピアリングには iBGP/eBGP を使用します。ファブリック内への外部ルートの配付には MP-BGP が使用されるため、外部ルータからの BGP ルート属性は保持されます。 同じ値を持つ推移的 BGP 拡張コミュニティと非推移的 BGP 拡張コミュニティの両方へのマッチを含む設定はサポートされません。このような設定は APIC によって拒否されます。

OSPF	BGP
OSPF ルート再配布 OSPF の default-information originate 機能により、外部ルータへのデフォルト ルートが生成されます。メインフレーム、外部ポッド、サービス ノードとピアリングする場合にはこのモードが推奨されます。 default-information originate がない場合は、redistribute static および redistribute BGP が OSPF ドメイン上で設定されて、それぞれスタティック ブリッジ ドメイン パブリック サブネットと中継ルートをアドバタイズします。ルートマップは、発信フィルタリングの再配布ポリシーに関連付けられています。外部 WAN ルータとピアリングする場合にはこのモードが推奨されます。着信方向では、OSPF ルートが MP-BGP 経由で ACI ファブリックに再配布されます。	**BGP ルート再配布** 発信方向では、default-originate ポリシーにより、BGP がピアごとにデフォルト ルートを生成します。ローカル ルーティング テーブルにデフォルト ルートが存在しない場合でも、このデフォルト ルートが BGP ピアに挿入されます。default-originate ポリシーが設定されていない場合には、ブリッジ ドメイン パブリック サブネットに対してスタティック再配布が有効になります。BGP によるアドバタイジングには MP-BGP からの中継ルートを使用できます。これらのルートは、発信フィルタリング ポリシーに基づき、条件付きで外部にアドバタイズされます。 着信方向では、着信フィルタリング ルールに基づき、MP-BGP が、アドバタイズされたルートを使用してファブリックを再配布します。BGP が外部ピアリングに使用される場合、ルートのすべての BGP 属性がファブリック全体で維持されます。
OSPF ルート フィルタリング 外部ピアから受け入れ可能なリンクステート アドバタイズメント（LSA）の数を制限するよう OSPF を設定することで、不正な外部ルータによるルート テーブルの過剰使用を避けることができます。 発信方向では、OSPF ドメイン レベルで redistribute static および redistribute BGP が設定されます。ルートマップは、ブリッジ ドメイン パブリック サブネットと中継ルートをフィルタリングするように設定されています。必要に応じて、ルートマップの一部のプレフィックスに set アクションを設定してルート タグを追加することもできます。発信フィルタ リストを使用し、これを OSPF エリアと関連付けることで、エリア間プレフィックスもフィルタリングされます。	**BGP ルート フィルタリング** BGP 着信ルート フィルタリングは、ピアごとにルートマップを使用して適用されます。ルートマップは、中継ルートがファブリックに入ることを許されるフィルタリング方向と同じ方向に、*peer-af* レベルで設定されます。 発信方向では、スタティック ルートは *dom af* レベルで BGP に再配布されます。外部 BGP ピアリング セッションには MP-BGP からの中継ルートが使用できます。ルートマップは、外方向の *peer-af* レベルで、パブリック サブネットおよび選択された外部中継ルートだけを許可するように設定されます。必要に応じて、選択したプレフィックスのコミュニティ値をアドバタイズする set アクションが、ルート マップ上で設定されています。 ブリッジ ドメイン パブリック サブネットと中継ルート プレフィクスは、プレフィクス リストとしては別々であっても、*peer-af* レベルで、異なるシーケンス番号を付けて単一のルートマップに統合することができます。

OSPF	BGP
OSPF のネーム ルックアップ、プレフィクス抑制、タイプ 7 変換 ルータ ID のネーム ルックアップを有効にしてプレフィクス抑制を実行するよう OSPF を設定することができます。 変換後のタイプ 5 LSA での OSPF フォワーディング アドレス抑制機能では、NSSA ABR でタイプ 7 LSA がタイプ 5 LSA に変換されます。ただし、フォワーディング アドレスとして、タイプ 7 LSA で指定されたものではなく 0.0.0.0 が使用されます。この機能を使用すると、フォワーディング アドレスをバックボーンにアドバタイズしないよう設定されているルータが、転送されたトラフィックを、変換を行う NSSA ASBR に渡すようになります。	**BGP ダイナミック ネイバー サポートとプライベート AS コントロール** 特定のネイバー アドレスを提供するのではなく、ダイナミック ネイバーのダイナミックな範囲のアドレスを提供することができます。 プライベート自律番号（AS）は 64512 ～ 65535 です。これらはグローバル BGP テーブルにリークできません。次のバリエーションに従って、プライベート AS 番号をピアごとの AS パスから削除し、eBGP ピアのみで使用することができます。 • `remove Private AS`：AS パスがプライベート AS 番号のみの場合、削除します。 • `remove All`：AS パスがプライベートおよびパブリック AS 番号の両方の場合、削除します。 • `replace As`：プライベート AS をローカル番号に置き換えます。 （注） `remove all` と `replace AS` は、`remove private as` が設定されている場合のみ設定できます。

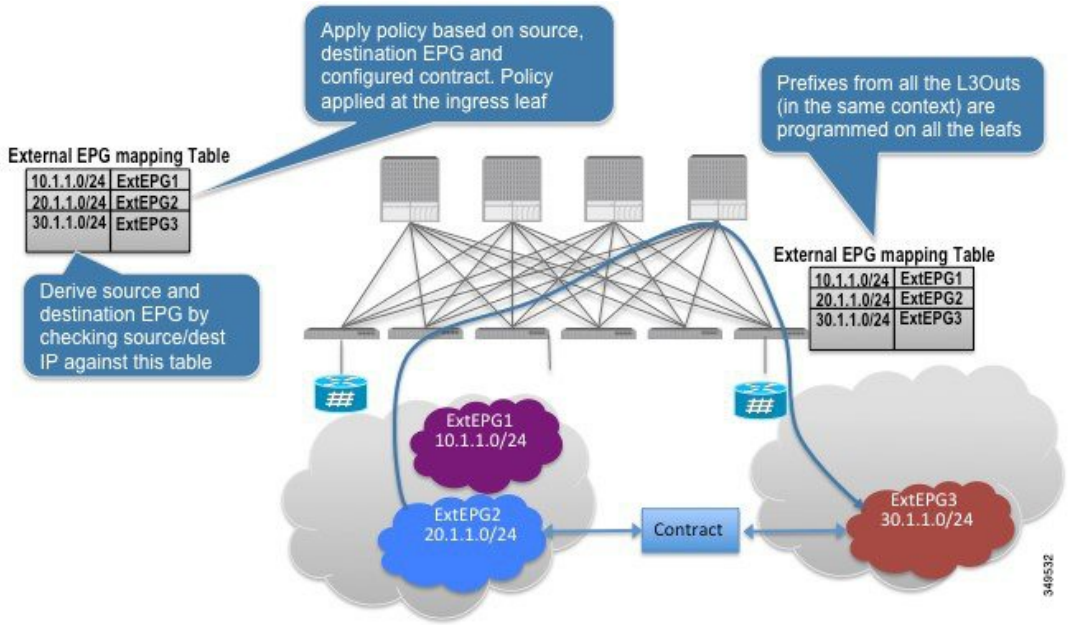
BGP ダンプニングでは、境界リーフ スイッチ（BLs）に接続されている外部ルータから受信したフラッピング e-BGP ルートのファブリックへの伝搬を最小限に抑えます。外部ルータから頻繁にフラッピングするルートは設定された条件に基づいて BLs で抑制され、iBGP ピア（ACI スパイン スイッチ）への再配布が禁止されます。抑制されたルートは設定された時間条件の後で再使用されます。各フラップは e-BGP ルートに 1000 のペナルティを科します。フラップのペナルティが定義された抑制限度のしきい値（デフォルトは 2000）に達すると、e-BGP ルートは抑制としてマーキングされます。抑制されたルートは他の BGP ピアにはアドバタイズされません。ペナルティは、半減期（デフォルトは 15 分）ごとに半分に減少します。抑制されたルートは、ペナルティが指定された再利用の限度（デフォルトは 750）を下回ると再利用されます。抑制されたルートは、指定された最大抑制時間（最大 45 分間）の間最大限に抑制されます。

最適なパスを選択するには、BGP 重み属性を使用します。重みは、ローカルでルータに割り当てられます。値は、特定のルータにのみ有効です。ルート更新によって値が伝搬されたり伝送されたりすることはありません。重みは、0 ～ 65,535 で指定できます。デフォルトでは、ルータの発信元であるパスの重みは 32,768、他のパスの重みは 0 に設定されています。同じ宛先に対して複数のルートがある場合、より高い重み値のルートが優先されます。重みは、ネイバーの下またはルート マップの下で設定できます。

BGP ピアリングは、通常、ネイバーのループバック アドレスに設定されます。このような場合、ループバックの到達範囲は、OSPF を通じて静的に設定またはアドバタイズされます。より一般的なのは後者のほうです。ループバック インターフェイスは、パッシブ インターフェイスとして設定され、OSPF エリアに追加されます。OSPF にアタッチされている再配布ポリシーはありません。ルート再配布とルート フィルタリングの実装には、BGP を使用します。

外部ルートをそれぞれのテナント内のボーダー リーフ上のスタティック ルートとしてプログラムすることもできます。外部ルートがボーダー リーフ上のスタティック ルートとしてプログラムされている場合、ピアリング プロトコルは必要はありません。外部スタティック ルートは、インポート フィルタリングに基づき、MP-BGP を通してファブリック内の他のリーフに再配布されます。リリース 1.2(1x) 以降、ACI ファブリック内のスタティック ルートのプリファレンスは、コスト拡張コミュニティを使用して MP-BGP で送信されます。レイヤ 3 外部接続では、レイヤ 4 からの MP-BGP ルートがローカル スタティック ルートより優先されます。ルートは、管理者が設定したプリファレンスでユニキャスト ルーティング情報ベース（URIB）にインストールされます。ACI の非境界リーフ スイッチでは、ルートはレイヤ 4 をネクストホップとしてインストールされます。レイヤ 4 でネクストホップを使用できない場合は、レイヤ 3 スタティック ルートがファブリック内で最善のルートになります。

図 *56*：中継のためのスタティック ルート ポリシー モデル



`l3extOut` 接続では、IP プレフィクスに基づいて外部エンドポイントが外部 EPG にマッピングされます。管理者は、異なる外部エンドポイント グループに異なるポリシー処理が必要かどうかに基づいて、それぞれの `l3extOut` 接続ごとに、1 つまたは複数の外部 EPG の作成を選択できます。

それぞれの外部 EPG は class-id と関連付けられています。外部 EPG の各プレフィクスは、対応する class-id を抽出するようにハードウェアでプログラムされています。プレフィックスは、コンテキストによってのみ認定され、導入先の `l3extOut` インターフェイスからは認定されません。

同じコンテキスト内のすべての l3extOut ポリシーからのプレフィクスの結合が、l3extOut ポリシーが導入されているすべてのリーフでプログラムされます。パケット内の送信元および宛先 IP アドレスに対応する送信元および宛先 class-id は入力リーフで抽出され、設定済みのコントラクトに基づき、入力リーフ自体でポリシーが適用されます。異なる 2 つの l3extOut インターフェイスの 2 つのプレフィックス間のトラフィックをコントラクトが許可している場合、l3extOut インターフェイス間では、（設定されたプレフィクスに属する）送信元および宛先 IP アドレスを任意に組み合せたパケットが許可されます。EPG 間にコントラクトがない場合には、トラフィックは入力リーフでドロップされます。

l3extOut ポリシーが導入されているすべてのリーフでプレフィクスがプログラムされるので、プレフィックスベースの EPG でサポートされるプレフィクスの総数は、ファブリックで 1K に制限されています。

同じコンテキスト内にある異なる l3extOut インターフェイス上で、重複するサブネットまたは同等のサブネットを設定することはできません。重複するサブネットまたは同等のサブネットが必要になった場合は、適切なエクスポートプレフィクスを付けて、単一の l3extOut を中継に使用します。

中継ルート制御

インポート対象になっている場合には l3extInstP に対してルート中継がインポートに定義され、エクスポート対象になっている場合には別の l3extInstP に対して別のルート中継がエクスポートに定義されます。

ファブリック内の 1 つまたは複数のノードに複数の l3extOut ポリシーを配置できるので、プロトコルのさまざまな組み合わせがサポートされます。プロトコルの組み合わせはすべて、複数の l3extOut ポリシーを使用して 1 つのノードに配置することも、または複数の l3extOut ポリシーを使用して複数のノードに配置することも可能です。同じファブリック内の異なる l3extOut ポリシーに 3 つ以上のプロトコルを配置することもできます。

エクスポート ルート マップは、IPv4 または IPv6 プレフィックス リストのマッチから構成されます。各 IPv4/IPv6 プレフィックス リストは、コンテキスト内のブリッジ ドメイン パブリック サブネットプレフィクスと、外部にアドバタイズする必要のあるエクスポートプレフィクスのセットから構成されます。

ルート制御ポリシーは、l3extOut ポリシーで定義され、l3extOut に関連付けられたプロパティおよび関係によって制御されます。ルート制御方向を適用するには、l3extOut の enforceRtctrl プロパティを使用します。デフォルトでは、外方向（エクスポート）で制御が適用され、内方向（インポート）ではすべてが許可されます。インポートおよびエクスポートされたルート（l3extSubnet）は、l3extInstP で定義されます。すべてのルートのデフォルト スコープはインポートです。これらは、プレフィックス ベースの EPG を形成するルート/プレフィックスです。

スコープがインポートになっているすべてのルートは、インポートルート マップを形成し、BGP によってインポートを制御するために使用されます。スコープがエクスポートになっているすべてのルートは、ルート マップを形成し、OSPF および BGP によってエクスポートを制御するために使用されます。

インポートとエクスポートのルート制御ポリシーは、異なるレベルで定義されます。IPv6 ではすべての IPv4 ポリシー レベルがサポートされます。l3extInstP および l3extSubnet での定義により追加された関係は、インポートを制御します。

デフォルト ルート リークは、l3extOut の下の l3extDefaultRouteLeakP MO の定義によって有効になります。

l3extDefaultRouteLeakP には、OSPF の場合はエリアごとに、BGP の場合はピアごとに、コンテキスト（ドメイン）スコープまたは l3-out を設定できます。

これらの設定ルールは、ルート制御を提供します。

- rtctrlSetPref
- rtctrlSetRtMetric
- rtctrlSetRtMetricType

rtctrlSetComm 用の追加の構文は次のとおりです。

- no-advertise
- no-export
- no-peer

BGP

ACI ファブリックは、外部ルータとの BGP ピアリングをサポートします。BGP ピアは l3extOut ポリシーに関連付けられており、l3extOut ごとに複数の BGP ピアを設定することができます。BGP は、l3extOut の下で bgpExtP MO を定義することにより l3extOut レベルで有効化できます。

（注） l3extOut ポリシーにルーティング プロトコル（たとえば、関連するコンテキストを含む BGP）が含まれる一方で、レイヤ 3 外部インターフェイスのプロファイルには必要な BGP インターフェイス設定の詳細が含まれます。いずれも BGP の有効化に必要です。

BGP ピアには、OSPF、EIGRP、接続されたインターフェイス、スタティック ルート、またはループバック経由で到達できます。外部ルータとのピアリングには iBGP/eBGP を使用できます。ファブリック内への外部ルートの配付には MP-BGP が使用されるため、外部ルータからの BGP ルート属性は保持されます。

l3extOut に関連付けられたコンテキストに関して IPv4 アドレス ファミリや IPv6 アドレス ファミリを有効にするには、これで十分です。スイッチ上で有効になるアドレス ファミリは、bgpPeerP ポリシーで l3extOut のために定義した IP アドレス タイプによって決まります。ポリシーは省略可能です。定義しない場合はデフォルトが使用されます。ポリシーはテナントに対して定義され、名前で参照されるコンテキストによって使用できます。

ボーダー リーフ スイッチでプロトコルを有効にするには、少なくとも 1 つのピア ポリシーを定義する必要があります。ピア ポリシーは 2 つの場所で定義できます。

- l3extRsPathL3OutAtt の下：送信元インターフェイスとして物理インターフェイスが使用されます。

- l3extLNodeP の下：送信元インターフェイスとしてループバック インターフェイスが使用されます。

OSPF

接続を有効にして冗長性を提供するために、さまざまなホスト タイプが OSPF を必要とします。これらには、たとえばファブリック内および WAN へのレイヤ 3 中継として ACI ファブリックを使用するサービス ノード、外部ポッド、メインフレームデバイスなどがあります。このような外部デバイスは、OSPF を実行している非境界リーフを介してファブリックとピアリングします。デフォルト ルートは受信し、全域ルーティングには参加しないよう、OSPF エリアを NSSA（スタブ）エリアとして設定します。通常は、既存のルーティングの導入によって設定の変更が回避されるため、スタブ エリアの設定は必須ではありません。

OSPF は、l3extOut の下で ospfExtP 管理対象オブジェクトを設定することで有効になります。ボーダー リーフ上で設定されている OSPF IP アドレス ファミリ バージョンは、OSPF インターフェイス IP アドレスに設定されているアドレス ファミリによって決まります。

（注） l3extOut ポリシーにルーティング プロトコル（たとえば、関連するコンテキストとエリア ID を含む OSPF）が含まれる一方で、レイヤ 3 外部インターフェイスのプロファイルには必要な OSPF インターフェイス設定の詳細が含まれます。いずれも OSPG の有効化に必要です。

すべての OSPF コンテキスト レベルのポリシーは、fvRsCtxToOspfCtxPol の関係を使用して設定します。関係はアドレス ファミリごとに設定できます。設定しない場合はデフォルトのパラメータが使用されます。

OSPF エリアは ospfExtP 管理対象オブジェクトで設定されます。ここでは、必要に応じて IPv6 エリアのプロパティも公開されます。

デフォルト ポリシー動作

任意の 2 つのプレフィックス ベース EPG 間にコントラクトがない場合、未知の発信元と未知の宛先プレフィクス間のトラフィックはドロップされます。これは、未知の発信元および未知の宛先プレフィクスに異なる class-id を暗黙的にプログラムすることによって達成されます。class-id が異なるため、これにはクラス不等規則が適用され、パケットは拒否されます。クラス不等ドロップ規則は、既知の送信元/宛先 IP と未知の送信元/宛先 IP との間で送られるパケットを拒否する役割も担っています。

デフォルト動作のこの変更に伴い、catch-all（0/0）エントリのための class-id プログラミングは、次の例に示すように変更されました。

- 未知の発信元 IP = EPG1
- 未知の宛先 IP = EPG2
- 未知の発信元 IP <-> 未知の宛先 IP => クラス不等規則 => ドロップ
- ユーザ設定デフォルト プレフィクス（0/0）= EPG3 および（10/8）= EPG4。EPG3 と EPG4 間のコントラクトは許可に設定されます

- プログラムされた規則
  - EPG1 <-> EPG4 => クラス不等規則 => ドロップ
  - EPG4 <-> EPG2 => クラス不等規則 => ドロップ

EIGRP プロトコルのサポート

EIGRP プロトコルは、ACI ファブリック内の他のルーティング プロトコルと同様にモデル化されています。サポートされる機能は次のとおりです。

- IPv4 および IPv6 ルーティング
- 各アドレス ファミリのコンテキストおよびインターフェイス制御
- ノード間の OSPF による再配布
- コンテキストごとのデフォルト ルート リーク ポリシー
- パッシブ インターフェイスおよびスプリット ホライズンのサポート
- エクスポートされたルートにタグを設定するためのルート マップ制御
- EIGRP インターフェイス ポリシーの帯域幅および遅延設定オプション

次の機能はサポートされていません。

- ノード間の EIGRP による再配布
- スタブ ルーティング
- BGP 接続に使用される EIGRP
- 同じノード上の複数の EIGRP L3Out
- 認証サポート
- サマリー プレフィックス
- インターフェイスごとのインポートおよびエクスポート用配布リスト

EIGRP の機能は、次のように大きく分類できます。

- プロトコル ポリシー
- L3Out 設定
- インターフェイス設定
- ルート マップ サポート
- デフォルト ルート サポート
- 中継サポート

EIGRP サポートを提供するプライマリ管理対象オブジェクトには次のものがあります。

- `eigrpCtxAfPol`：`fvTenant`（テナント/プロトコル）下で設定されたコンテキスト ポリシー。
- `fvRsCtxToEigrpCtxAfPol`：所定のアドレス ファミリ（IPv4 または IPv6）についての、コンテキストから `eigrpCtxAfPol` への関係。関係は、アドレス ファミリごとに 1 つのみ存在できます。
- `eigrpIfPol`：`fvTenant` で設定される EIGRP インターフェイス ポリシー。
- `eigrpExtP`：L3Out 上で EIGRP のフラグを有効にします。
- `eigrpIfP`：`l3extLIfP` に接続された EIGRP インターフェイス プロファイル。
- `eigrpRsIfPol`：EIGRP インターフェイス プロファイルから `eigrpIfPol` への関係。
- `Defrtleak`：`l3extOut` 下のデフォルト ルート リーク ポリシー。

テナント下では次の EIGRP プロトコル ポリシーがサポートされます。

- **インターフェイス ポリシー（`eigrpIfPol`）**：インターフェイス上の所定のアドレス ファミリに適用される設定が含まれます。インターフェイス ポリシーでは次の設定が可能です。
  - 秒単位の *hello* 間隔
  - 分単位の *hold* 間隔
  - 次のインターフェイス制御フラグのうち 1 つ以上。
    - スプリット ホライズン
    - パッシブ
    - ネクスト ホップ セルフ

- **コンテキスト ポリシー（`eigrpCtxAfPol`）**：所定のコンテキスト内の所定のアドレス ファミリの設定が含まれます。EIGRP コンテキスト ポリシーは、テナント プロトコル ポリシー下で設定し、テナント下の 1 つ以上のコンテキストに適用できます。EIGRP コンテキスト ポリシーは、アドレス ファミリ単位コンテキストの関係によって、コンテキスト上で有効にできます。所定のアドレス ファミリーに関係がない場合、あるいは関係に記述されている EIGRP コンテキスト ポリシーがな存在しない場合は、共通テナント下に作成されたデフォルトのコンテキスト ポリシーが、そのアドレス ファミリに使用されます。

  コンテキスト ポリシーでは次の設定が可能です。
  - 内部ルートのアドミニストレーティブ ディスタンス
  - 外部ルートのアドミニストレーティブ ディスタンス
  - 最大許容 ECMP パス数
  - アクティブ タイマー間隔
  - メトリック バージョン（32 ビット/64 ビット メトリック）

EIGRP L3out 設定

EIGRP は、リーフ スイッチで設定されているファブリックのパブリック サブネット、接続ルート、スタティック ルート、中継ルートのアドバタイジングに使用するメイン プロトコルです。

任意の `l3extOut` ルーテッド ドメインの EIGRP に関する有効/無効フラグがあります。

（注） EIGRP に使用されるタグである自律システム番号。BGP で使用されるファブリック ASN とは同じではありません。

EIGRP は、同じ L3Out 上の BGP または OSPF とともに有効にすることはできません。

次の EIGRP 中継シナリオがサポートされます。

- 1 つのノード上の L3Out 内で実行する EIGRP と、別のノード上の別の L3Out 内で実行する OSPF。

  （注） 同じコンテキスト（VRF）内の同じノード上の複数の EIGRP L3Out はサポートされません。

- スタティック ルート中継への EIGRP。

EIGRP インターフェイス プロファイル

インターフェイス上で EIGRP を有効にするには、[L3Out] -> [Node] -> [Interface] と階層をたどって、インターフェイス プロファイル下で EIGRP プロファイルを設定する必要があります。EIGRP プロファイルには、テナント内で有効にされている EIGRP インターフェイス ポリシーとの関係があります。テナント内に関係またはインターフェイス ポリシーが存在しない場合は、共通テナントのデフォルト EIGRP インターフェイス ポリシーが使用されます。EIGRP は、インターフェイス プロファイルに含まれているすべてのインターフェイスで有効になっています。たとえば、インターフェイス プロファイルに含まれる VPC、L3 ポート、サブインターフェイス、ポート上の外部 SVI、ポート チャネルなどです。

ポリシーモデルのルートマップのインフラストラクチャと設定は、すべてのプロトコルに共通です。ルート マップのセット アクションは、BGP、OSPF、EIGRP をカバーする上位集合アクションです。EIGRP プロトコルは、相互リーク/再配布に使用するルート マップの *set tag* オプションをサポートしています。これらのルート マップはコンテキスト単位で設定します。L3Out に IPv4 と IPv6 の両方のインターフェイスがある場合、相互リーク ポリシーは、そのコンテキストの IPv4 と IPv6 の両方のアドレス ファミリに適用されます。

（注） 現時点では、コンテキスト レベルのルート マップはサポートされていますが、インターフェイス ルート マップはサポートされていません。

L3Out 上のデフォルト ルート リーク ポリシーは、設定の点ではプロトコルに依存しません。デフォルト ルート リーク ポリシーで有効になっているプロパティは、個別のプロトコルの上位集合です。デフォルト ルート リークでサポートされている設定は次のとおりです。

- *Scope*：コンテキストが、EIGRP でサポートされている唯一のスコープになります。
- Always：スイッチは、ルーティング テーブルに存在している場合のみデフォルト ルートをアドタイズするか、無差別にアドバタイズします。
- *Criteria*：唯一（only）あるいは追加（in-addition）。only オプションを使用すると、EIGRP はデフォルト ルートだけをアドバタイズします。in-addition では、デフォルト ルートとともにパブリック サブネットと中継ルートもアドバタイズされます。

デフォルト ルート リーク ポリシーは、アドレス ファミリ単位のコンテキストごとにドメイン内で有効にすることができます。

デフォルトで、適切なルート マップを伴うプロトコル再配布相互リークポリシーが、有効なすべての設定にセットアップされています。管理者が中継ルーティングを有効にするのは、純粋に、*scope=export-route control* である `l3extInstP` サブネットを作成することによって、同一コンテキスト内の 2 つの L3Out 間で特定のルートを送信できるようにしたいという理由からです。`l3extInstP` サブネットのスコープのほかには、中継ケースを扱うための特別なプロトコル固有の設定はありません。プロトコル固有であるスコープのほか、デフォルト ルートリーク ポリシーの他のパラメータが、すべてのプロトコルに共通です。

異なるノード中継シナリオでは、別の L3Out 上の OSPF が EIGRP でサポートされています。

次に示す EIGRP のガイドラインおよび制限事項に従ってください。

- 現時点では、同じリーフ スイッチ上の複数の EIGRP L3Out はサポートされていません。
- すべてのルートが、EIGRP を使用する L3Out 上でインポートされます。インポート サブネット スコープは、EIGRP が L3Out 上のプロトコルである場合、GUI で使用することはできません。

第 8 章

ユーザ アクセス、認証およびアカウンティング

この章の内容は、次のとおりです。

ユーザ アクセス、認証およびアカウンティング

APIC ポリシーは、Cisco ACI ファブリックのアクセス、認証、およびアカウンティング（AAA）機能を管理します。ユーザ権限、ロールおよびドメインとアクセス権限の継承を組み合わせることにより、管理者は非常に細分化された方法で管理対象オブジェクト レベルで AAA 機能を設定することができます。これらの設定は、REST API、CLI、または GUI を使用して実行できます。

マルチテナントのサポート

コア APIC 内部データ アクセス コントロール システムにより、マルチテナント分離が提供され、テナント間での個人情報の漏洩が防止されます。読み取り/書き込みの制約により、テナントによる他のテナントの設定、統計情報、障害、またはイベント データの参照が防止されます。管理者によって読み取り権限が割り当てられない限り、テナントはファブリックの設定、ポリシー、統計情報、障害、またはイベントの読み取りが制限されます。

ユーザアクセス：ロール、権限、セキュリティドメイン

APIC では、ロールベース アクセス コントロール（RBAC）を介してユーザのロールに従ってアクセスが提供されます。ACI ファブリック ユーザは、次に関連付けられています。

- ロールのセット
- ロールごとの権限タイプ：アクセスなし、読み取り専用、または読み取り/書き込み
- ユーザがアクセスできる管理情報ツリー（MIT）の一部を識別する 1 つ以上のセキュリティ ドメイン タグ

ACI ファブリックは、管理対象オブジェクト（MO）レベルでアクセス権限を管理します。権限は、システム内の特定の機能に対するアクセスを許可または制限する MO です。たとえば、ファブリック機器は権限ビットです。このビットは、物理ファブリックの機器に対応するすべてのオブジェクト上で APIC によって設定されます。

ロールは権限ビットの集合です。たとえば、「管理者」ロールが「ファブリック機器」と「テナント セキュリティ」に対する権限ビットに設定されていると、「管理者」ロールにはファブリックの機器とテナント セキュリティに対応するすべてのオブジェクトへのアクセス権があります。

セキュリティ ドメインは、ACI MIT オブジェクト階層の特定のサブツリーに関連付けられたタグです。たとえば、デフォルトのテナント「common」にはドメイン タグ `common` が付いています。同様に、特殊なドメイン タグ `all` の場合、MIT オブジェクト ツリー全体が含まれます。管理者は、MIT オブジェクト階層にカスタム ドメイン タグを割り当てることができます。たとえば、管理者は「solar」という名前のテナントにドメイン タグ「solar」を割り当てることができます。MIT 内では、特定のオブジェクトだけがセキュリティ ドメインとしてタグ付けできます。たとえば、テナントはセキュリティ ドメインとしてタグ付けすることができますが、テナント内のオブジェクトはできません。

ユーザを作成してロールを割り当てても、アクセス権は有効になりません。1 つ以上のセキュリティ ドメインにそのユーザを割り当てることも必要です。デフォルトでは、ACI ファブリックには事前作成された以下の 2 つの特殊なドメインが含まれています。

- `All`：MIT 全体へのアクセスを許可
- `Infra`：ファブリック アクセス ポリシーなどの、ファブリック インフラストラクチャのオブジェクトおよびサブツリーへのアクセスを許可

（注） ユーザのクレデンシャルが許可しない管理対象オブジェクトの読み取り操作の場合、「DN/Class Unauthorized to read」ではなく「DN/Class Not Found」というエラーが返されます。ユーザのクレデンシャルが許可しない管理対象オブジェクトへの書き込み操作の場合、「HTTP 401 Unauthorized」というエラーが返されます。GUI では、ユーザのクレデンシャルが許可しないアクションの場合、表示されないか、またはグレー表示されます。

事前に定義された一連の管理対象オブジェクト クラスをドメインに関連付けることができます。これらのクラスがオーバーラップすることはできません。ドメインの関連付けをサポートするクラスの例：

- レイヤ 2 およびレイヤ 3 のネットワークで管理されたオブジェクト
- ネットワーク プロファイル（物理、レイヤ 2、レイヤ 3、管理など）
- Quality of Service（QoS）ポリシー

ドメインに関連付けることができるオブジェクトが作成されると、ユーザは、ユーザのアクセス権の範囲内でオブジェクトにドメインを割り当てる必要があります。ドメインの割り当てはいつでも変更できます。

仮想マシン管理（VMM）ドメインがセキュリティ ドメインとしてタグ付けされている場合、セキュリティ ドメイン内のユーザは、対応するタグ付き VMM ドメインにアクセスできます。たとえば、solar という名前のテナントに sun というセキュリティ ドメインのタグが付いており、VMM ドメインにも sun というセキュリティ ドメインのタグが付いている場合、solar テナント内のユーザは各自のアクセス権限に従って VMM ドメインにアクセスできます。

アカウンティング

ACI ファブリック アカウンティングは、障害およびイベントと同じメカニズムで処理される以下の 2 つの管理対象オブジェクト（MO）によって処理されます。

- `aaaSessionLR` MO は、APIC およびスイッチでのユーザ アカウントのログイン/ログアウト セッション、およびトークン更新を追跡します。ACI ファブリック セッション アラート機能は、次のような情報を保存します。
  - ユーザ名
  - セッションを開始した IP アドレス
  - タイプ（telnet、https、REST など）
  - セッションの時間と長さ
  - トークン更新：ユーザ アカウントのログイン イベントは、ユーザ アカウントが ACI ファブリックの権利を行使するために必要な、有効なアクティブ トークンを生成します。

（注） トークンはログインに関係なく期限切れになります。ユーザはログアウトできますが、トークンは含まれているタイマー値の期間に従って期限切れになります。

- aaaModLR MO は、ユーザがオブジェクトに対して行う変更、およびいつ変更が発生したかを追跡します。

aaaSessionLR と aaaModLR 両方のイベント ログは、APIC シャードに保存されます。データがプリセットされているストレージ割り当てサイズを超えると、先入れ先出し方式でレコードを上書きします。

（注） APIC クラスタ ノードを破壊するディスク クラッシュや出火などの破壊的なイベントが発生した場合、イベント ログは失われ、イベント ログはクラスタ全体で複製されません。

aaaModLR MO と aaaSessionLR MO は、クラスまたは識別名（DN）でクエリできます。クラスのクエリは、ファブリック全体のすべてのログ レコードを提供します。ファブリック全体のすべての aaaModLR レコードは、GUI の [Fabric] -> [Inventory] -> [pod-1] -> [history] -> [audit log] セクションで取得できます。[GUI] => [History] => [Log] オプションを使用すると、GUI コンテキストで識別された特定のオブジェクトのイベント ログを表示できます。

標準の syslog、callhome、REST クエリ、および CLI エクスポート メカニズムは、aaaModLR MO と aaaSessionLR MO のクエリ データで完全にサポートされます。このデータをエクスポートするデフォルト ポリシーはありません。

APIC には、一連のオブジェクトまたはシステム全体のデータの集約を報告する、事前設定されたクエリはありません。ファブリック管理者は、aaaModLR および aaaSessionLR のクエリ データを定期的に syslog サーバにエクスポートするエクスポート ポリシーを設定できます。エクスポートされたデータを定期的にアーカイブし、システムの一部またはシステム ログ全体のカスタム レポートを生成するために使用できます。

共有サービスとしての外部ネットワークへのルーテッド接続の課金と統計情報

APIC は、共有サービスとしての外部ネットワークへのルーテッド接続用に設定されたポート（l3extInstP EPG）からバイト カウントとパケット カウントでの課金統計情報を収集するように設定できます。任意のテナントの任意の EPG が、外部ネットワークへのルーテッド接続用に l3extInstP EPG を共有できます。課金統計情報は、共有サービスとして l3extInstP EPG を使用する任意のテナント内の EPG ごとに収集できます。l3extInstP がプロビジョニングされているリーフ スイッチは課金統計情報を APIC に転送し、そこで課金情報が集約されます。定期的に課金統計情報をサーバにエクスポートするようにアカウンティング ポリシーを設定できます。

カスタムの RBAC 規則

RBAC 規則により、ファブリック全体の管理者は、本来はブロックされるはずのセキュリティ ドメイン間アクセスを許可することができます。RBAC 規則を使用して、別のセキュリティ ドメインにあるため他の方法ではアクセス不可能なサービスを共有したり物理リソースを公開したりできます。RBAC 規則では、ターゲット リソースへの読み取りアクセスのみ許可されます。[Admin] > [AAA] > [Security Management] と移動すると、GUI RBAC 規則のページがあります。RBAC 規則は、リソースが存在する前から作成できます。RBAC 規則、ロール、権限（およびその依存関係）の記述については、管理情報モデルのリファレンスで説明されています。

（注） ユーザに対し、そのユーザのセキュリティ ドメインの外にあるリソースへのアクセス許可を付与するように「全」ドメインを変更することは好ましくありません。そのようなユーザは、他のユーザ向けにプロビジョニングしたリソースにアクセスできるようになってしまいます。

複数のセキュリティ ドメイン間で物理リソースを選択的に公開する

ファブリック全体の管理者は、RBAC 規則を使用して、異なるセキュリティ ドメインにあるため他の方法ではアクセス不可能なユーザに対し、物理リソースを選択的に公開します。

たとえば、ソーラーというテナントのユーザが仮想マシン管理（VMM）ドメインへのアクセスを必要とする場合、ファブリック全体の管理者によって、これを許可する RBAC 規則を作成することができます。RBAC 規則は、次の 2 つの部分から構成されます。アクセス対象オブジェクトを検索する識別名（DN）と、オブジェクトにアクセスするユーザを含むセキュリティ ドメインの名前です。したがって、この例では、ソーラーというセキュリティ ドメイン内の指定ユーザがログインすると、このルールにより、VMM ドメインおよびツリーの内の子オブジェクトすべてへのアクセスが許可されます。VMM ドメインへのアクセスを複数のセキュリティ ドメイン内のユーザに許可するには、ファブリック全体の管理者は、セキュリティ ドメインそれぞれについて、VMM ドメインの DN とセキュリティ ドメインを含む RBAC 規則を作成します。

（注） 管理情報ツリー内の異なる部分に存在するユーザに対し、RBAC 規則によりオブジェクトを公開することは可能ですが、CLI の使用によってツリーの構造を横断することでそのようなオブジェクトに移動することはできません。ただし、RBAC 規則に含まれるオブジェクトの DN をユーザが把握していれば、ユーザは MO 検索コマンドにより、CLI を使用してそれを見つけることができます。

関連トピック

サンプルの RBAC 規則, （247 ページ）

複数のセキュリティ ドメイン間でのサービス共有を有効にする

ファブリック全体の管理者は、RBAC 規則を使用して、テナント間の共有サービスを可能にするトランステナント EPG 通信をプロビジョニングします。

APIC ローカル ユーザ

管理者は、外部 AAA サーバを使用しないことを選択し、APIC 自体でユーザを設定することができます。これらのユーザは、APIC ローカル ユーザと呼ばれます。

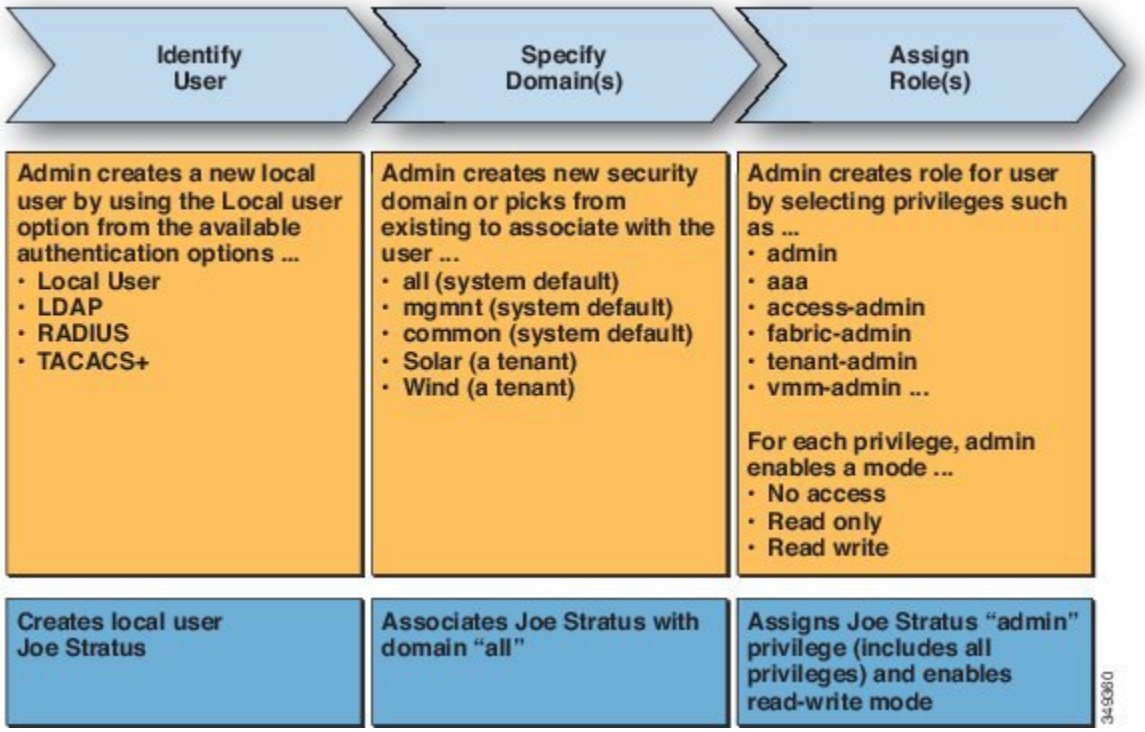
ユーザがパスワードを設定する時点で、APIC は以下の基準に対してパスワードを検証します。

- パスワードの最小長は 8 文字です。
- パスワードの最大長は 64 文字です。
- 連続して繰り返される文字は 3 文字未満です。
- 小文字、大文字、数字、記号の文字種のうち少なくとも 3 種類の文字が含まれている必要があります。
- 簡単に推測できるパスワードは使用しません。
- ユーザ名やユーザ名を逆にしたものは使用できません。
- cisco、isco、またはこれらの文字列の並べ替えを変化させたものや、それらの文字の大文字化の変更により取得される変形語であってはなりません。

また APIC により、管理者は、外部で管理されている認証 Lightweight Directory Access Protocol（LDAP）、RADIUS、または TACACS+ サーバで設定されたユーザにアクセス権を付与できます。ユーザは、異なる認証システムに属し、APIC に同時にログインできます。

次の図は、ACI ファブリック全体へのフル アクセス権があるローカル APIC 認証データベース内の管理ユーザを設定するプロセスがどのように動作するかを示します。

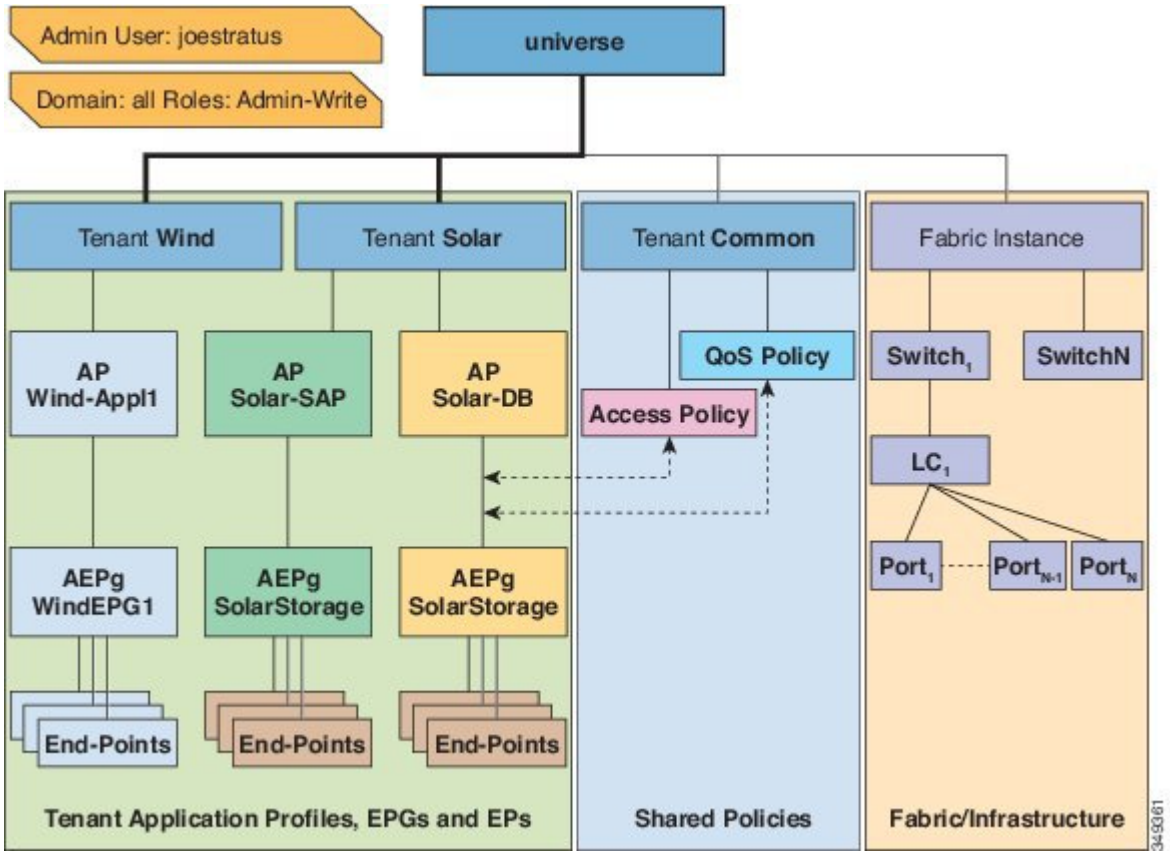
図 *57* : *APIC* ローカル ユーザの設定プロセス



（注） セキュリティ ドメイン「all」は、管理情報ツリー（MIT）全体を表します。このドメインには、システム内のすべてのポリシーと APIC よって管理されるすべてのノードが含まれます。テナント ドメインには、テナントのすべてのユーザおよび管理対象オブジェクトが含まれます。テナント管理者には、「all」ドメインへのアクセス権を付与しないでください。

次の図は、管理ユーザ Joe Stratus が持つシステムへのアクセス権を示します。

図 *58*：「*all*」ドメインへ管理ユーザを設定した結果

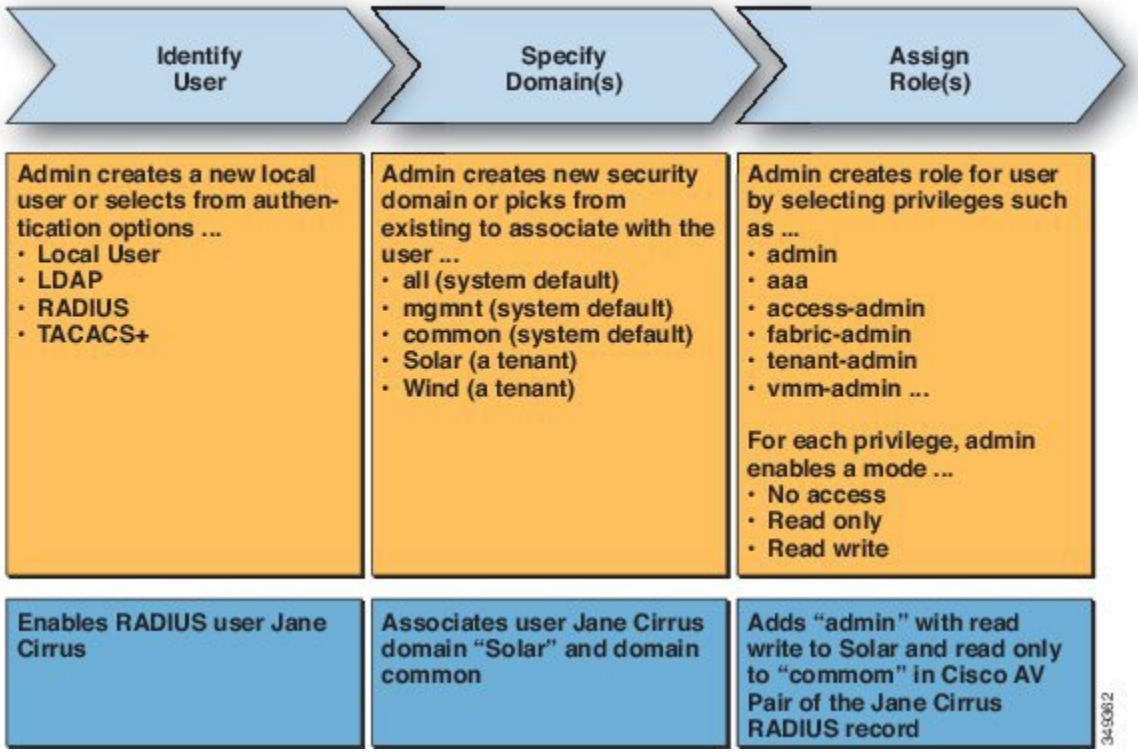


読み取り/書き込み「管理者」権限を持つユーザ Joe Stratus は、ドメイン「all」に割り当てられ、システム全体へのフル アクセス権が与えられます。

外部管理されている認証サーバのユーザ

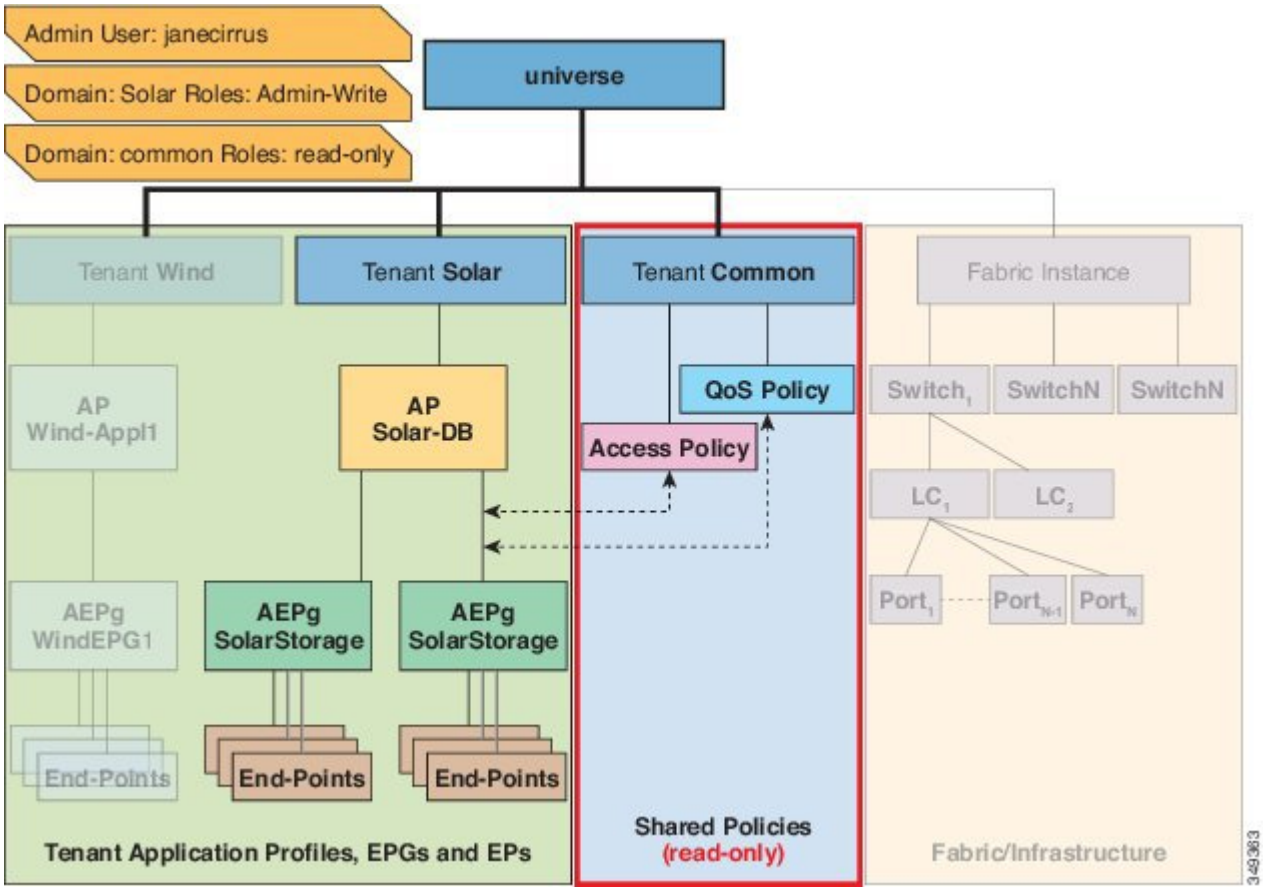
次の図は、テナント Solar へのフル アクセス権がある外部 RADIUS サーバ内の管理ユーザを設定するプロセスがどのように動作するかを示します。

図 *59*：外部認証サーバでのユーザ設定のプロセス



次の図は、管理ユーザ Jane Cirrus が持つシステムへのアクセス権を示します。

図 *60*：テナント *Solar* へ管理ユーザを設定した結果



この例では、Solar テナントの管理者には、Solar テナントに含まれるすべてのオブジェクトへのフル アクセス権と、テナント Common への読み取り専用アクセス権があります。テナント管理者 Jane Cirrus には、テナント Solar へのフル アクセス権があり、テナント Solar で新しいユーザを作成する機能などがあります。テナント ユーザは、自身が所有し制御する ACI ファブリックの設定パラメータを変更できます。また、エンドポイント、エンドポイント グループ（EPG）およびアプリケーション プロファイルなどの適用されるエンティティ（管理対象オブジェクト）の統計情報の読み取り、障害およびイベントのモニタもできます。

上記の例では、ユーザ Jane Cirrus は外部 RADIUS 認証サーバで設定されました。外部認証サーバで AV ペアを設定するには、既存のユーザ レコードに Cisco AV ペアを追加します。Cisco AV ペアは、APIC 上のユーザに対するロールベース アクセス コントロール（RBAC）のロールと権限を指定します。次に RADIUS サーバは、ユーザ権限を APIC コントローラに伝播します。

上記の例のオープン RADIUS サーバ（/etc/raddb/users）の設定は次のとおりです。

```
janecirrus Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = solar/admin/,common//read-all(16001)"
```

この例には、次の要素が含まれています。

- janecirrus はテナント管理者です。
- solar はテナントです。
- admin は書き込み権限があるロールです。
- commonは、テナント共通サブツリーで、すべてのユーザがそのサブツリーへの読み取り専用アクセス権を持っています。
- read-all は、読み取り権限があるロールです。

Cisco AV ペアの形式

Cisco APIC では、管理者が外部認証サーバで Cisco AV ペアを設定する必要があります。これを行うには、管理者は既存のユーザ レコードに Cisco AV ペアを追加します。Cisco AV ペアは、ユーザの RBAC ロールおよび権限に必要な APIC を指定します。Cisco AV ペアの形式は、RADIUS、LDAP、または TACACS+ のものと同じです。

Cisco AV ペアの形式は次のとおりです。

```
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)
```

最初の AV ペアの形式には UNIX ユーザ ID がなく、2 番目のものにはあります。どちらも正しいです。

APIC は、次の正規表現をサポートしています。

```
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

RADIUS サーバでユーザを設定するには、APIC 管理者は cisco-av-pair 属性を使用して必要な属性（shell:domains）を設定する必要があります。デフォルトのユーザ ロールは、network-operator です。

SNMPv3 認証プロトコルに指定できるオプションは、SHA と MD5 です。プライバシー プロトコルに指定できるオプションは、AES-128 と DES です。これらのオプションが cisco-av-pair 属性で指定されていない場合は、MD5 および DES がデフォルトの認証プロトコルとなります。

たとえば、SNMPv3 認証とプライバシー プロトコルの属性は次のように指定できます。

```
snmpv3:auth=SHA priv=AES-128
```

同様に、ドメインのリストは次のとおりです。

```
shell:domains="domainA domainB …"
```

TACACS+ 認証

Terminal Access Controller Access Control device Plus（TACACS+）は、シスコ デバイスでサポートされる別のリモート AAA プロトコルです。TACACS+ には、RADIUS 認証にはない次の利点があります。

- 独立した AAA ファシリティを提供する。たとえば、APICは、認証を行わずにアクセスを許可できます。
- AAA クライアントとサーバ間のデータ送信に TCP を使用しているため、コネクション型プロトコルによる確実な転送が可能になります。
- スイッチと AAA サーバ間でプロトコル ペイロード全体を暗号化して、高度なデータ機密性を実現します。RADIUS はパスワードのみを暗号化します。
- 構文と設定が RADIUS と異なる av-pairs を使用しますが、APIC は `shell:domains` をサポートします。

次に示す XML の例では、IP アドレス 10.193.208.9 の TACACS+ プロバイダーを ACI ファブリックに使用させるよう設定が行われています。

（注） この例では IPv4 アドレスを使用していますが、IPv6 アドレスも使用できます。

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

LDAP/Active Directory の認証

RADIUS および TACACS+ と同様、LDAP により、ネットワーク要素はユーザを認証し、特定のアクションの実行を許可するために使用できる AAA クレデンシャルを取得できます。追加された認証局の設定は管理者によって実行でき、LDAPS（SSL 経由の LDAP）の信頼性をイネーブルにし、中間者攻撃を防ぐことができます。

次に示す XML の例では、ACI ファブリックが IP アドレス 10.30.12.128 の LDAP プロバイダーを使用するように設定しています。

（注） この例では IPv4 アドレスを使用していますが、IPv6 アドレスも使用できます。

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

（注） LDAP 設定のベストプラクティスは、属性文字列として AciCiscoAVPair を使用することです。これにより、オブジェクト識別子（OID）の重複を許可しない LDAP サーバ制限に関連した問題が回避されます。つまり、ciscoAVPair OID がすでに使用されている場合です。

APIC Bash シェルのユーザ ID

APIC での Linux シェル用のユーザ ID は、ローカル ユーザ用に APIC 内で生成されます。認証クレデンシャルが外部サーバで管理されているユーザは、Linux シェル用のユーザ ID を cisco-av-pair で指定できます。上記の cisco-av-pair の（16001）を省略することは、リモート ユーザがデフォルトの Linux ユーザ ID 23999 を取得すれば可能です。Linux ユーザ ID がバッシュ セッション中に使用され、標準の Linux 権限が適用されます。また、ユーザが作成するすべての管理対象オブジェクトは、そのユーザの Linux ユーザ ID によって作成されたとマークされます。

次に、APIC Bash シェルに表示されるユーザ ID の例を示します。

```
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)
```

ログイン ドメイン

ログイン ドメインは、ユーザの認証ドメインを定義します。ログイン ドメインは、ローカル、LDAP、RADIUS、または TACACS+ 認証メカニズムを設定できます。REST、CLI、または GUI からシステムにアクセスすると、APIC によりユーザは正しい認証ドメインを選択できます。

たとえば、REST シナリオでは、完全なログイン ユーザ名が次のように表示されるようにユーザ名の頭に文字列が付きます。

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

システムに GUI からアクセスする場合は、APIC により選択するユーザのドメインのドロップダウン リストが提供されます。`apic: domain` が指定されない場合は、デフォルトの認証ドメイン サーバがユーザ名の検索に使用されます。

ACI バージョン 1.0(2x) 以降、APIC のログイン ドメイン フォールバックのデフォルトはローカルになっています。デフォルト認証とコンソール認証方法がどちらも非ローカルの方法に設定されており、両方の非ローカル方法がローカル認証に自動的にフォールバックしない場合でも、APIC にはローカル認証を使用してアクセスすることができます。

APIC フォールバック ローカル認証にアクセスするには、次の文字列を使用します。

- GUI からは、*apic:fallback\\username* を使用します。
- REST API からは、*apic#fallback\\username* を使用します。

（注） フォールバック ログイン ドメインは変更しないでください。変更すると、システムからロックアウトされる可能性があります。

第 9 章

Virtual Machine Manager のドメイン

この章の内容は、次のとおりです。

Cisco ACI の VM ネットワーキングによる複数ベンダーの Virtual Machine Manager のサポート

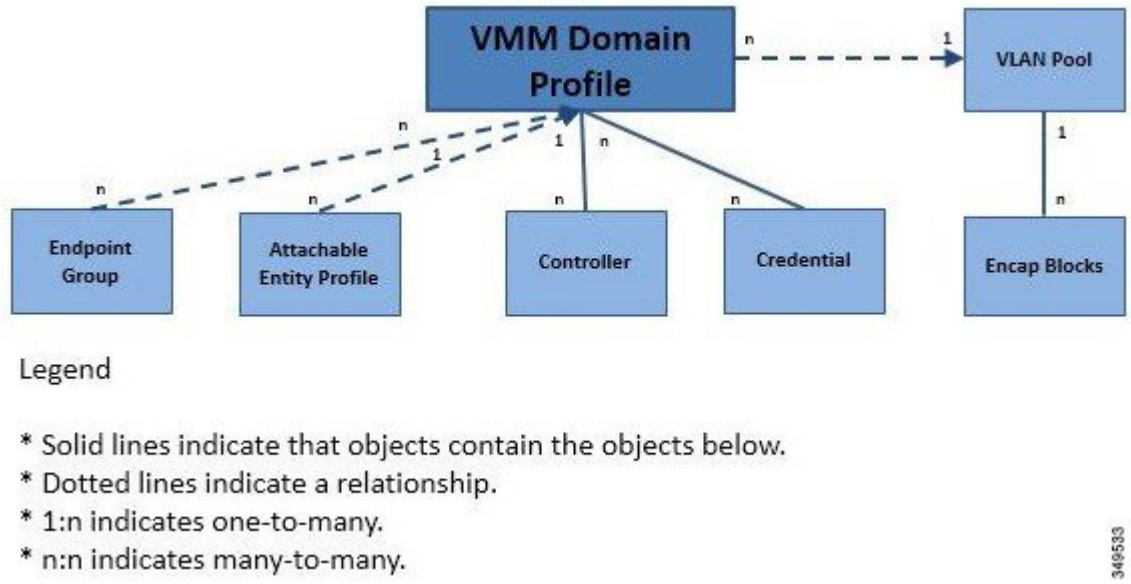
Cisco ACI の仮想マシン ネットワーキングは、複数ベンダーによるハイパフォーマンスでスケーラブルな仮想データセンターインフラストラクチャへのプログラム可能自動アクセスのハイパーバイザを提供します（確認済みの相互運用可能な製品の最新のリストについては、「Virtualization Compatibility List Solution Overview」を参照してください）。プログラム可能性と自動化は、スケーラブルなデータ センター仮想化インフラストラクチャにおける重要な機能です。ACI オープン REST API により、ポリシー モデル ベースの ACI ファブリックのオーケストレーションと仮想マシン（VM）を統合できます。ACI VM ネットワーキングにより、複数のベンダーのハイパーバイザで管理される仮想ワークロードと物理ワークロードの両方にわたって一貫してポリシーを適用できます。接続可能エンティティ プロファイルにより、VM モビリティと ACI ファブリック内の任意の場所のワークロードの配置を簡単に実現できます。ACI APIC コントローラにより、トラブルシューティング、アプリケーションのヘルス スコア、および仮想化モニタリングが一元化され

ます。手動設定および手動作業における間違いを削減または排除することにより、ACI のマルチハイパーバイザ VM 自動化は、非常に多くの VM を仮想化データ センターが信頼性とコスト効率を保ちながらサポートすることを可能にします。

VMM ドメイン ポリシー モデル

VMM ドメイン プロファイル（vmmDomP）は、仮想マシン コントローラを ACI ファブリックに接続させる接続ポリシーを指定します。次の図は、vmmDomP ポリシーの概要を示しています。

図 *61*：*VMM* ドメイン ポリシー モデルの概要



Virtual Machine Manager ドメインの主要コンポーネント

ACI ファブリック Virtual Machine Manager（VMM）ドメインにより、管理者は仮想マシン コントローラの接続ポリシーを設定できます。ACI VMM ドメイン ポリシーの基本的なコンポーネントは次のとおりです。

- **Virtual Machine Manager ドメイン プロファイル**：同様のネットワーキング ポリシー要件を持つ VM コントローラをグループ化します。たとえば、VM コントローラは VLAN プールとアプリケーション エンドポイント グループ（EPG）を共有できます。APIC はコントローラと通信し、のちに仮想ワークロードに適用されるポート グループなどのネットワーク設定を公開します。VMM ドメイン プロファイルには、次の基本コンポーネントが含まれます。
  - **クレデンシャル**：有効な VM コントローラ ユーザ クレデンシャルを APIC VMM ドメインと関連付けます。

- **コントローラ**：ポリシーの適用ドメインの一部である VM コントローラへの接続方法を指定します。たとえば、コントローラは VMM ドメインの一部である VMware vCenter への接続を指定します。

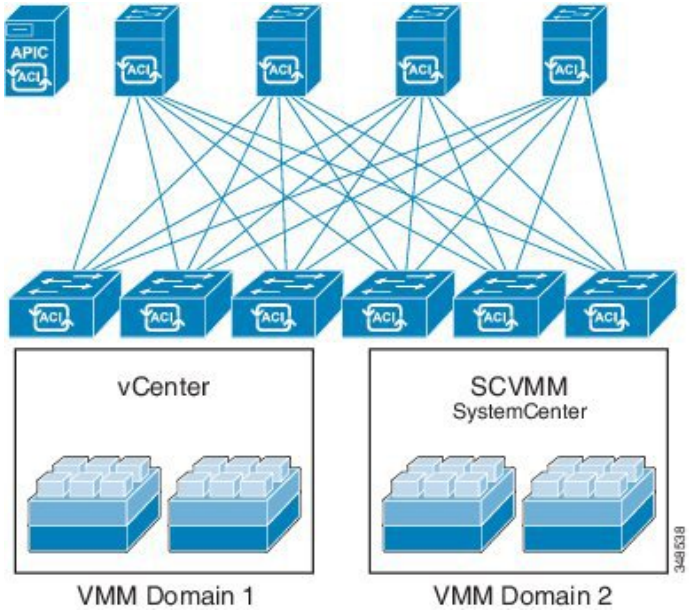
（注） 1 つのドメインに VM コントローラの複数のインスタンスを含めることができますが、それらは同じベンダーのものである必要があります（VMware または Microsoft など）。

- **EPG の関連付け**：エンドポイント グループは VMM ドメイン ポリシーの範囲内のエンドポイント間の接続と可視性を調整します。VMM ドメイン EPG は次のように動作します。
  - APIC は、これらの EPG をポート グループとして VM コントローラにプッシュします。
  - 1 つの EPG は、複数の VMM ドメインをカバーでき、1 つの VMM ドメインには複数の EPG を含めることができます。
- **接続可能エンティティ プロファイルの関連付け**：VMM ドメインを物理ネットワーク インフラストラクチャと関連付けます。接続可能エンティティ プロファイル（AEP）は、多数のリーフ スイッチ ポートで VM コントローラ ポリシーを展開するための、ネットワーク インターフェイステンプレートです。AEP は、使用できるスイッチやポートおよびその設定方法を指定します。
- **VLAN プールの関連付け**：VLAN プールは、VMM ドメインが消費する VLAN カプセル化に使用する VLAN ID または範囲を指定します。

Virtual Machine Manager のドメイン

APIC VMM ドメイン プロファイルは、VMM ドメインを定義するポリシーです。VMM ドメイン ポリシーは APIC で作成され、リーフ スイッチにプッシュされます。

図 *62*：*ACI VMM* ドメイン *VM* コントローラの統合



VMM ドメインは以下を提供します。

- 複数の VM コントローラ プラットフォームに対してスケーラブルな耐障害性サポートを可能にする、ACI ファブリックの共通レイヤ
- ACI ファブリック内の複数のテナントに対する VMM サポート

VMM ドメインには、VMware vCenter や Microsoft SCVMM Manager などの VM コントローラと、VM コントローラと対話するための ACI API に必要なクレデンシャルが含まれます。VMM ドメインはドメイン内の VM モビリティを実現できますが、ドメイン間は実現できません。単一の VMM ドメイン コントローラに VM コントローラの複数のインスタンスを含めることはできますが、同じタイプである必要があります。たとえば、1 つの VMM ドメインに、それぞれが複数の VM を実行する複数のコントローラを管理する多くの VMware vCenter を含めることができますが、SCVMM Manager も含めることはできません。VMM ドメインはコントローラ要素（pNIC、vNIC、VM 名など）をインベントリに含め、コントローラにポリシーをプッシュして、ポート グループなどの必要な要素を作成します。ACI VMM ドメインは VM モビリティなどのコントローラ イベントを監視し、状況に応じて応答します。

VMM ドメイン VLAN プールの関連付け

VLAN プールは、トラフィック VLAN ID のブロックを表します。VLAN プールは共有リソースで、VMM ドメインおよびレイヤ 4 ～レイヤ 7 のサービスなど、複数のドメインで使用できます。

各プールには、作成時に定義された割り当てタイプ（静的または動的）があります。割り当てタイプによって、含まれる ID が APIC で自動割り当てに使用されるか（動的）、管理者によって明示的に設定されるか（静的）が決まります。デフォルトでは、VLAN プールに含まれるすべてのブロックの割り当てタイプはプールと同じですが、ユーザは動的プールに含まれるカプセル化ブロックの割り当てタイプを静的に変更できます。これを行うと、動的割り当てからそれらが除外されます。

VMM ドメインは、1 つの動的 VLAN プールにのみ関連付けることができます。デフォルトでは、VMM ドメインに関連付けられた EPG への VLAN ID の割り当ては、APIC によって動的に行われます。動的割り当てがデフォルトであり、推奨設定ですが、管理者は代わりに EPG に静的に VLAN ID を割り当てることができます。この場合、使用する ID は VMM ドメインに関連付けられている VLAN プールのカプセル化ブロックから選択し、その割り当てタイプを静的に変更する必要があります。

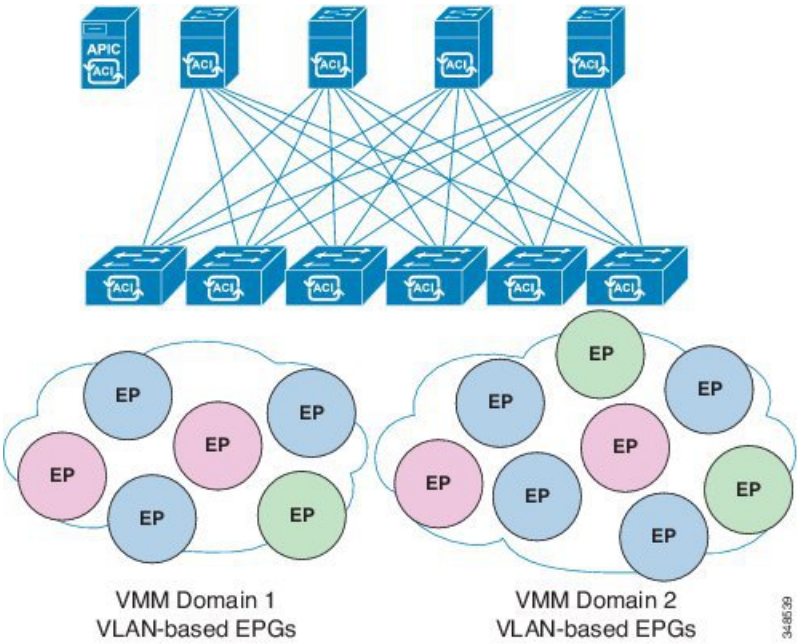
APIC は、リーフ ポート上の VMM ドメイン VLAN を EPG イベントに基づいてプロビジョニングします（リーフ ポート上の静的バインドまたは VMware vCenter や Microsoft SCVMM などのコントローラからの VM イベントに基づいて）。

VMM ドメイン EPG の関連付け

ACI ファブリックは、Microsoft Azure などのオーケストレーション コンポーネントによって自動的に、またはその設定を作成する APIC 管理者によって、VMM ドメインにテナント アプリケー

ションプロファイル EPG を関連付けます。1 つの EPG は、複数の VMM ドメインをカバーでき、1 つの VMM ドメインには複数の EPG を含めることができます。

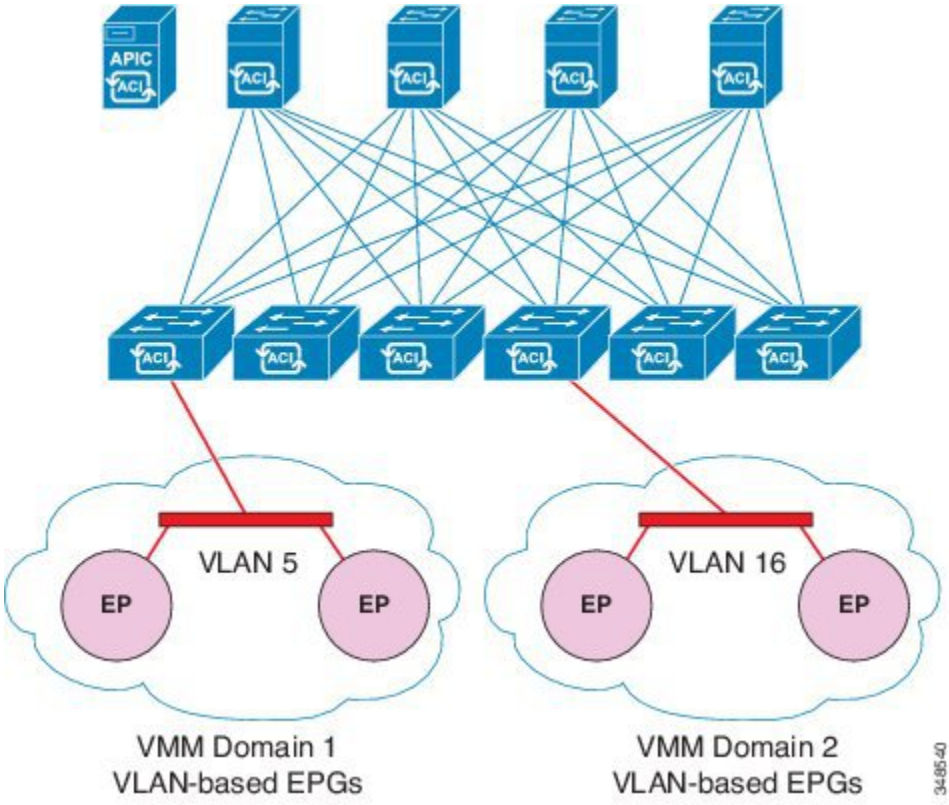
図 *63* : *VMM* ドメイン *EPG* の関連付け



上の図では、同じ色のエンド ポイント（EP）は同じエンド ポイント グループに属しています。たとえば、緑色のすべての EP は 2 つの異なる VMM ドメインに含まれていますが同じ EPG に属しています。

仮想ネットワークと VMM ドメイン EPG 機能の情報については、Cisco ACI ドキュメントの最新の『Verified Scalability Guide』を参照してください。

図 *64*：*VMM* ドメイン *EPG VLAN* の消費



（注） 同じポートに重複する VLAN プールがない場合は、複数の VMM ドメインを同じリーフ スイッチに接続できます。同様に、リーフ スイッチの同じポートを使用していない場合は、同じ VLAN プールを異なるドメイン間で使用できます。

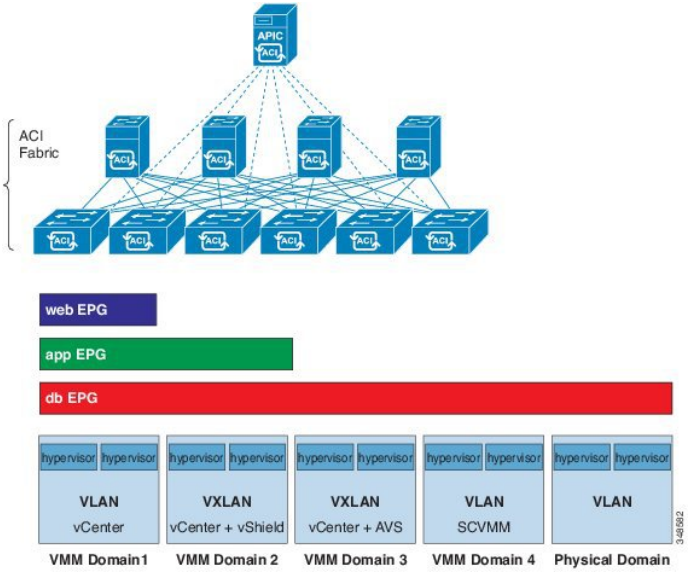
EPG は複数の VMM ドメインを次のように使用できます。

- VMM ドメイン内の EPG は、APIC によって自動的に管理されるか管理者によって固定で選択されたカプセル化識別子を使用して識別されます。一例は、VLAN、仮想ネットワーク ID（VNID）です。
- EPG は複数の物理ドメイン（baremetal サーバの場合）または仮想ドメインにマッピングできます。各ドメインで異なる VLAN または VNID カプセル化を使用できます。

（注） デフォルトで、APIC は動的に EPG の VLAN の割り当てを管理します。VMware DVS 管理者は、EPG に対して特定の VLAN を設定できます。その場合は、VLAN は VMM ドメインに関連付けられたプール内のスタティック割り当てブロックから選択します。

アプリケーションは、複数の VMM ドメインに導入できます。

図 *65*：ファブリック内の複数の *VMM* ドメインと *EPG* の増大



VMM ドメイン内の VM のライブ マイグレーションがサポートされていても、VMM ドメイン間の VM のライブ マイグレーションはサポートされません。

EPG ポリシーの解決および展開の緊急度

EPG が VMM ドメインに関連付けられるたびに、管理者は解決と展開の優先順位を選択して、ポリシーをいつリーフ スイッチにプッシュするかを指定できます。

解決の緊急性

- [Pre-provision]：VM コントローラが仮想スイッチ（たとえば、VMware VDS や Cisco AVS など）に接続される前でもポリシー（たとえば、VLAN、VXLAN バインディング、コントラクト、フィルタなど）をリーフ スイッチにダウンロードすることにより、スイッチの設定を事前プロビジョニングすることを指定します。

  事前プロビジョニングの重要度を使用する場合、ポリシーは CDP または LLDP のネイバーシップには関係なくダウンロードされます。VMM スイッチに接続されたホストがない。

- [Immediate]：VM コントローラが仮想スイッチに接続すると EPG ポリシー（コントラクトおよびフィルタを含む）が関連付けられているリーフ スイッチ ソフトウェアにダウンロードされるよう指定します。VM コントローラ/リーフ ノード接続を解決するために LLDP または OpFlex 権限が使用されます。

  VMM スイッチにホストを追加すると、ポリシーがリーフにダウンロードされます。ホストからリーフへの CDP または LLDP のネイバーシップが必要です。

- [On Demand]：VM コントローラが仮想スイッチに接続され、VM がポート グループ（EPG）に配置されている場合にのみ、ポリシー（たとえば、VLAN、VXLAN バインディング、コントラクト、フィルタ）がリーフ ノードにプッシュされるよう指定します。

  ホストが VMM スイッチに追加され、仮想マシンをポート グループ（EPG）に配置する必要がある場合、ポリシーがリーフにダウンロードされます。ホストからリーフへの CDP または LLDP のネイバーシップが必要です。

  即時とオンデマンドの両方において、ホストおよびリーフが LLDP または CDP のネイバーシップを失うと、ポリシーは削除されます。

展開の緊急性

ポリシーがリーフ ソフトウェアにダウンロードされると、展開の緊急度でポリシーをいつハードウェア ポリシー CAM にプッシュするかを指定できます。

- [Immediate]：ポリシーがリーフ ソフトウェアでダウンロードされるとすぐにポリシーがハードウェアのポリシー CAM でプログラムされるよう指定します。
- [On Demand]：最初のパケットがデータ パス経由で受信された場合にのみポリシーがハードウェアのポリシー CAM でプログラムされるよう指定します。このプロセスは、ハードウェアの領域を最適化するのに役立ちます。

VMM ドメインを削除するためのガイドライン

次の手順に従って、VMM ドメインを自動的に削除する APIC リクエストによって関連する VM コントローラ（VMware vCenter または Microsoft SCVMM）がトリガーされ、プロセスが正常に完了すること、および ACI ファブリックに孤立した EPG が残されないことを確認します。

1. VM 管理者は、APIC によって作成されたすべての VM を、ポート グループ（VMware vCenter の場合）または VM ネットワーク（SCVMM の場合）からデタッチする必要があります。

   Cisco AVS の場合、VM 管理者は Cisco AVS に関連付けられている vmk インターフェイスも削除する必要があります。

2. ACI 管理者は、APIC で VMM ドメインを削除します。APIC は、VMware VDS または Cisco AVS または SCVMM 論理スイッチおよび関連するオブジェクトの削除をトリガーします。

（注） VM 管理者が仮想スイッチまたは関連オブジェクト（ポート グループまたは VM ネットワークなど）を削除することはできません。上記のステップ 2 の完了時に、APIC に仮想スイッチの削除を許可します。VMM ドメインが APIC で削除される前に VM 管理者が VM コントローラから仮想スイッチを削除した場合、EPG は APIC で孤立する可能性があります。

このシーケンスに従わない場合、VM コントローラは APIC VMM ドメインに関連付けられている仮想スイッチを削除します。このシナリオでは、VM 管理者は VM コントローラから VM および vtep アソシエーションを手動で削除してから、以前に APIC VMM ドメインに関連付けられていた仮想スイッチを削除します。

第 10 章

レイヤ 4 ～ レイヤ 7 のサービスの挿入

この章の内容は、次のとおりです。

レイヤ 4 ～ レイヤ 7 のサービスの挿入

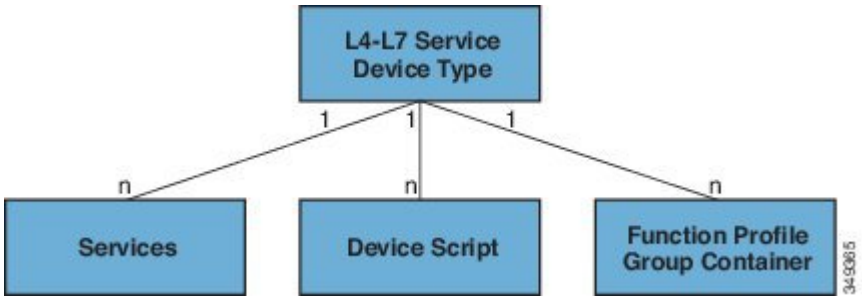
Cisco Application Policy Infrastructure Controller（APIC）は、ネットワーク サービスを管理します。ポリシーは、サービスを挿入するために使用されます。APIC サービス統合により、ライフサイクルの自動化フレームワークが提供され、サービスがオンラインまたはオフラインになった場合にシステムが動的に応答できます。ファブリック全体で使用可能な共有サービスは、ファブリックの管理者によって管理されます。単一のテナント向けのサービスは、テナントの管理者によって管理されます。

APICは、ポリシー制御の中心点として機能すると同時に、自動サービス挿入を提供します。APICポリシーは、ネットワーク ファブリックとサービス アプライアンスの両方を管理します。APICは、トラフィックがサービスを通って流れるように、ネットワークを自動的に設定できます。また、APICはアプリケーション要件に従ってサービスを自動的に設定できます。このアプローチにより、組織はサービス挿入を自動化し、従来のサービス挿入の複雑なすべてのトラフィック誘導技術の管理に伴う課題を排除できます。

レイヤ 4 ～ レイヤ 7 のポリシー モデル

レイヤ 4 ～ レイヤ 7 のサービス デバイス タイプ ポリシーには、パッケージおよびデバイス スクリプトでサポートされるサービスなどの主要な管理対象オブジェクトが含まれます。次の図は、レイヤ 4 ～ レイヤ 7 のサービス デバイス タイプ ポリシー モデルのオブジェクトを示します。

図 *66*：レイヤ *4* ～ レイヤ *7* のポリシー モデル



レイヤ 4 ～ レイヤ 7 のサービス ポリシーには次のものが含まれます。

- **サービス**：SSLオフロードやロードバランシングなどのデバイスによって提供されるすべての機能のメタデータが含まれます。このMOには、コネクタの名前、VLANやVXLANなどのカプセル化のタイプ、およびインターフェイス ラベルが含まれます。
- **デバイス スクリプト**：名前、パッケージ名、バージョンなどのスクリプト ハンドラの関連属性に関するメタ情報を含むデバイス スクリプト ハンドラを表します。
- **機能プロファイル グループ コンテナ**：サービス デバイス タイプで使用可能な機能を含むオブジェクト。機能プロファイルには、フォルダに編成されたデバイスでサポートされる設定可能なすべてのパラメータが含まれます。

サービス グラフ

Cisco Application Centric Infrastructure（ACI）は、アプリケーションの欠くことのできない一部としてサービスを扱います。必要とされるすべてのサービスが、Cisco Application Policy Infrastructure Controller（APIC）から ACI ファブリックでインスタンス化されるサービス グラフとして扱われます。ユーザは、アプリケーションに対してサービスを定義し、サービス グラフはアプリケー

ションが必要とする一連のネットワークまたはサービス機能を識別します。各機能はノードとして表されます。

グラフが APIC に設定されると、APIC はサービス グラフに明記されたサービス機能の要件に従って、サービスを自動的に設定します。APIC はまた、サービス グラフで指定されるサービス機能のニーズに応じてネットワークを自動的に設定しますが、これによってサービス デバイスでの変更は要求されません。

サービス グラフは、アプリケーションの複数の階層として表され、適切なサービス機能が間に挿入されます。

サービス アプライアンス（デバイス）は、グラフ内でサービス機能を実行します。1 つ以上のサービス アプライアンスが、グラフに必要なサービスをレンダリングするために必要になることがあります。1 つ以上のサービス機能が単一のサービス デバイスで実行できます。

サービス グラフおよびサービス機能には、次の特性があります。

- エンドポイント グループ（EPG）で送信または受信されたトラフィックはポリシーに基づいてフィルタリングでき、トラフィックのサブセットはグラフ内の異なるエッジにリダイレクトできます。
- サービス グラフのエッジには方向性があります。
- タップ（ハードウェア ベースのパケット コピー サービス）は、サービス グラフの異なるポイントに接続できます。
- 論理機能は、ポリシーに基づいて適切な（物理または仮想）デバイスでレンダリングできます。
- サービス グラフでは、エッジの分割と結合がサポートされ、管理者は線形サービス チェーンに制限されません。
- トラフィックは、サービス アプライアンスが発信した後にネットワーク内で再度分類できます。
- 論理サービス機能は、要件に応じて、拡張や縮小が可能で、クラスタ モードまたは 1:1 アクティブ/スタンバイ ハイアベイラビリティ モードで展開できます。

サービス グラフ コンフィギュレーション パラメータ

サービス グラフには、デバイス パッケージで指定されたコンフィギュレーション パラメータを割り当てることができます。コンフィギュレーション パラメータは、EPG、アプリケーション プロファイルまたはテナント コンテキストでも指定できます。サービス グラフ内の機能ノードでは、1 つ以上のコンフィギュレーション パラメータが必要になる場合があります。パラメータ値は変更がさらに加えられるのを防ぐためにロックできます。

サービス グラフを設定し、コンフィギュレーション パラメータの値を指定すると、APIC はそのパラメータをデバイス パッケージ内にあるデバイス スクリプトに渡します。デバイス スクリプトは、パラメータ データをデバイスにダウンロードされる設定に変換します。

サービス グラフ接続

サービス グラフ接続は、1 つの機能ノードを別の機能ノードに接続します。

自動サービス挿入

VLAN および仮想ルーティングおよび転送（VRF）スイッチングは、従来のサービス挿入モデルによってサポートされますが、Application Policy Infrastructure Controller（APIC）はポリシー制御の中心点として機能する一方で、サービス挿入とセキュア ソケット レイヤ（SSL）オフロード、サーバ ロード バランシング（SLB）、Web アプリケーション ファイアウォール（WAF）およびファイアウォールなどのネットワーク サービスのプロビジョニングを自動化できます。ネットワーク サービスは通常、Application Delivery Controller（ADC）やファイアウォールなどのサービス アプライアンスによってレンダリングされます。APIC ポリシーは、ネットワーク ファブリックとサービス アプライアンスの両方を管理します。APIC は、トラフィックがサービスを通って流れるように、ネットワークを自動的に設定できます。APICは、アプリケーション要件に従ってサービスを自動的に設定することもでき、それにより組織はサービス挿入を自動化し、従来のサービス挿入の複雑な技術の管理に伴う課題を排除できます。

デバイス パッケージ

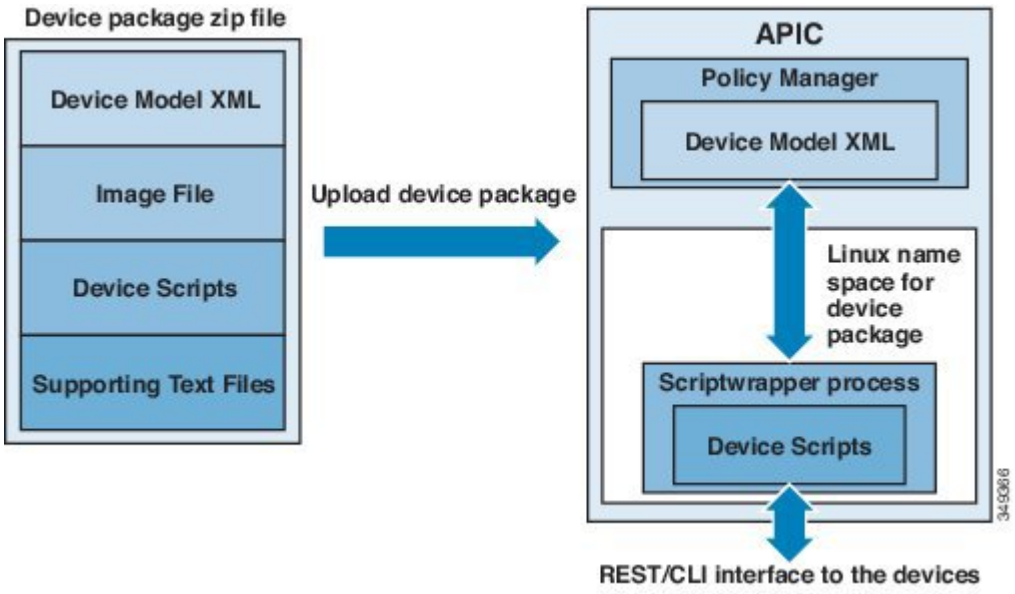
Application Policy Infrastructure Controller（APIC）は、サービス デバイスの設定およびモニタリングにデバイス パッケージを必要とします。デバイス パッケージは、サービス デバイスのクラスを管理して、デバイスが何であるか、およびデバイスで何が実行できるかを APIC が認識できるように APIC にデバイスの情報を提供します。管理者は、デバイス パッケージを使用することにより、APIC 上のネットワーク サービスを中断なく追加、変更、削除することができます。APIC への新しいデバイス タイプの追加は、デバイスパッケージをアップロードすることで実行できます。

デバイス パッケージは次の項目を含む `zip` ファイルです。

デバイス仕様	次のプロパティを定義する XML ファイル： • デバイス プロパティ： ◦ [Model]：デバイスのモデル。 ◦ [Vendor]：デバイスのベンダー。 ◦ [Version]：デバイスのソフトウェア バージョン。 • ロードバランシング、コンテンツ切り替え、およびSSL終端などの、デバイスによって提供される機能。 • 各機能のインターフェイスおよびネットワーク接続情報。 • デバイス設定パラメータ。 • 各機能の設定パラメータ。
デバイス スクリプト	APIC とデバイス間の統合を実行する Python スクリプト。APIC イベントは、デバイス スクリプトで定義した機能呼び出しにマッピングされます。
機能プロファイル	ベンダーによって指定されたデフォルト値を持つパラメータのプロファイル。これらのデフォルト値を使用するように機能を設定できます。
デバイスレベル設定パラメータ	デバイス レベルでデバイスに必要なパラメータを指定するコンフィギュレーション ファイル。設定は、デバイスを使用している 1 つ以上のグラフで共有できます。

次の図に、デバイス パッケージによる APIC サービスの自動化と挿入アーキテクチャを示します。

図 *67*：デバイス パッケージ アーキテクチャ



デバイス パッケージは、デバイス ベンダーが提供するか、またはシスコが作成できます。

デバイス パッケージにより、管理者は次のサービスの管理を自動化することができます。

- デバイスの接続と切断
- エンドポイントの接続と切断
- サービス グラフのレンダリング
- ヘルス モニタリング
- アラーム、通知、ロギング
- カウンタ

デバイス パッケージが GUI またはノースバウンド APIC インターフェイス経由でアップロードされると、APIC はそれぞれ一意なデバイス パッケージのネームスペースを作成します。デバイス パッケージの内容は、解凍されネーム スペースにコピーされます。デバイス パッケージのネームスペース用に作成されるファイル構造は次のとおりです。

```
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
```

デバイス パッケージのコンテンツは `install` ディレクトリにコピーされます。

APIC がデバイス モデルを解析します。XML ファイルで定義される管理対象オブジェクトは、ポリシー マネージャによって維持される APIC の管理対象オブジェクト ツリーに追加されます。

デバイス パッケージで定義される Python スクリプトは、ネームスペースのスクリプト ラッパー プロセス内で開始されます。ファイル システムへのアクセスは制限されます。Python スクリプトは、/tmp に一時ファイルを作成でき、デバイス パッケージの一部としてバンドルされたテキスト ファイルにアクセスできます。ただし、Python スクリプトではファイル内に永続データを作成または保存しないでください。

デバイス スクリプトは、ACI ロギング フレームワークを通してデバッグ ログを生成できます。ログは、logs ディレクトリ下の debug.log という循環型ファイルに書き込まれます。

各デバイス パッケージのバージョンは自身のネームスペースで動作するため、デバイス パッケージの複数のバージョンが APIC 上に共存できます。管理者は、一連のデバイスを管理するための特定のバージョンを選択できます。

デバイス クラスタについて

デバイス クラスタ（別名論理デバイス）は、単一のデバイスとして機能する 1 つ以上の具象デバイスです。デバイス クラスタには、デバイス クラスタのインターフェイス情報を説明するクラスタ（論理）インターフェイスがあります。サービス グラフのテンプレート レンダリング中に、機能ノード コネクタはクラスタ（論理）インターフェイスに関連付けられます。Application Policy Infrastructure Controller（APIC）は、サービス グラフのテンプレート インスタンス化およびレンダリング中に機能ノード コネクタにネットワーク リソース（VLAN または Virtual Extensible Local Area Network（VXLAN）を割り当て、クラスタ（論理）インターフェイスにネットワーク リソースをプログラミングします。

サービス グラフ テンプレートは、管理者が定義するデバイス選択ポリシー（論理デバイス コンテキストと呼ばれます）に基づく特定のデバイスを使用します。

管理者は、アクティブ/スタンバイ モードで最大 2 つの具象デバイスをセットアップできます。

デバイス クラスタをセットアップするには、次のタスクを実行する必要があります。

1. ファブリックに具象デバイスを接続します。
2. デバイス クラスタに管理 IP アドレスを割り当てます。
3. APIC にデバイス クラスタを登録します。APIC は、デバイス パッケージからのデバイス仕様を使用してデバイスを検証します。

具象デバイスについて

具象デバイスには、具象インターフェイスがあります。具象デバイスが論理デバイスに追加されると、具象インターフェイスは論理インターフェイスにマッピングされます。サービス グラフのテンプレート インスタンス化時に、VLAN および VXLAN は、論理インターフェイスとのアソシエーションに基づいた具象インターフェイス上でプログラミングされます。

機能ノード

機能ノードは、単一サービス機能を表します。機能ノードには、サービス機能のネットワーク要件を表す機能ノード コネクタがあります。

サービス グラフ内の機能ノードは、1つ以上のパラメータが必要になる場合があります。パラメータは、エンドポイント グループ（EPG）、アプリケーション プロファイル、またはテナント コンテキストで指定できます。パラメータは、管理者がサービス グラフを定義した時点で割り当てることもできます。パラメータ値は変更がさらに加えられるのを防ぐためにロックできます。

機能ノード コネクタ

機能ノード コネクタは、サービス グラフに機能ノードを接続し、グラフのコネクタ サブネットに基づいて適切なブリッジ ドメインと接続と関連付けられます。各コネクタは、VLAN または Virtual Extensible LAN（VXLAN）に関連付けられます。コネクタの両側がエンドポイント グループ（EPG）として扱われ、ホワイトリストがスイッチにダウンロードされ、2つの機能ノード間の通信がイネーブルになります。

端末ノード

端末ノードはサービス グラフとコントラクトを接続します。管理者は、コントラクトに端末ノードを接続することにより、2 つのアプリケーション エンドポイント グループ（EPG）間のトラフィックに対しサービス グラフを挿入できます。接続されると、コントラクトのコンシューマ EPG とプロバイダー EPG 間のトラフィックはサービス グラフにリダイレクトされます。

権限について

管理者は、APIC でロールに権限を付与できます。権限は、ロールが実行できるタスクを決定します。管理者は、管理者のロールに次の権限を付与できます。

特権	説明
nw-svc-connectivity	• 管理 EPG の作成 • 他のオブジェクトに管理接続を作成
nw-svc-policy	• サービス グラフの作成 • アプリケーション EPG およびコントラクトへのサービス グラフのアタッチ • サービス グラフのモニタ

特権	説明
nw-svc-device	• デバイス クラスタの作成 • 具象デバイスの作成 • デバイス コンテキストの作成

（注） インフラストラクチャの管理者だけがデバイス パッケージを APIC にアップロードできます。

サービスの自動化と構成管理

Cisco APIC は、サービス デバイスの構成管理と自動化のポイントとして任意に動作でき、ネットワーク自動化とのサービス デバイスの調整を行うことができます。Cisco APIC は、さまざまなイベントで Python スクリプトを使用してサービス デバイスと連動し、デバイス固有の Python スクリプト機能を呼び出します。

デバイス スクリプトとサービス デバイスでサポートされる機能を定義するデバイスの仕様は、デバイス パッケージとしてまとめられ、Cisco APIC にインストールされます。デバイス スクリプト ハンドラは、デバイス設定モデルに基づいてその REST インターフェイス（推奨）または CLI を使用してデバイスとやりとりします。

サービス リソースのプーリング

Cisco ACI ファブリックは、多数の宛先間で非ステートフル負荷分散を実行できます。この機能により、組織は物理および仮想サービス デバイスをサービス リソース プールにグループ化でき、機能や場所によってさらにグループ化できます。これらのプールは、標準のハイアベイラビリティ メカニズムを使用することでハイ アベイラビリティを提供するか、または障害が発生した場合に、他のメンバーに負荷が再分散された状態で簡易なステートフル サービス エンジンとして使用できます。どちらのオプションでも、等コスト マルチパス（ECMP）、ポート チャネル機能および共有状態を必要とするサービス アプライアンスのクラスタリングの現在の制限をはるかに超える横方向のスケールアウトが提供されます。

サービス デバイスがファブリックとやりとりする必要がない場合、Cisco ACI はサービス デバイスを使用して簡易バージョンのリソース プーリングを実行できます。また、ファブリックとサービス デバイス間の調整を伴うより高度なプーリングも実行できます。

第 11 章

管理ツール

この章の内容は、次のとおりです。

管理ツール

Cisco アプリケーション セントリック インフラストラクチャ（ACI）のツールは、ファブリックの管理者、ネットワーク エンジニア、および開発者がテナントおよびアプリケーションの導入を開発、設定、デバッグおよび自動化するのに役立ちます。

管理 GUI について

次の管理 GUI の機能により、ファブリックおよびそのコンポーネント（リーフとスパイン）にアクセスできます。

- 世界共通の Web 標準（HTML5）に基づく。インストーラまたはプラグインは必要ありません。

- モニタリング（統計情報、障害、イベント、監査ログ）、操作および設定データへのアクセス。
- シングル サインオン メカニズムによる APIC とスパインおよびリーフ スイッチへのアクセス。
- サードパーティが使用できる同じ RESTful API を使用した APIC との通信。

CLI について

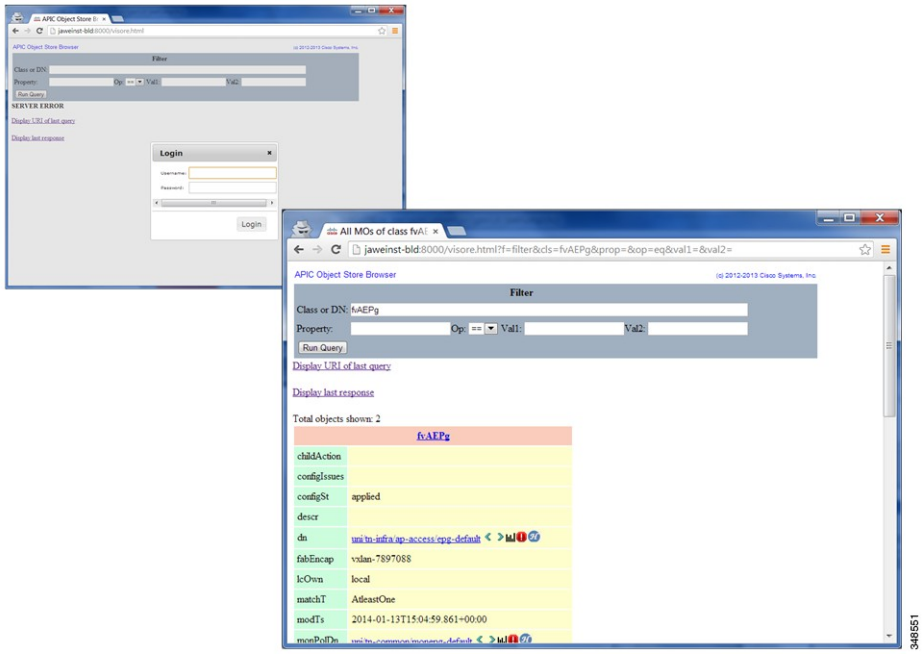
CLI は、APIC、リーフおよびスパイン スイッチへの操作インターフェイスおよび設定インターフェイスを特徴としています。

- Python で初めから実行され、Python インタープリタと CLI 間で切替えることができます。
- 拡張性のプラグイン アーキテクチャ
- 監視データ、操作データおよび構成データへのコンテキストベースのアクセス
- Python コマンドまたはバッチ スクリプティングによる自動化

Visore 管理対象オブジェクト ビューア

Visore は、下の図に示すように、読み取り専用の管理情報ツリー（MIT）ブラウザです。これにより、オプションのフィルタを使用して、識別名（DN）とクラスのクエリが可能になります。

図 *68*：*Visore MO* ビューア

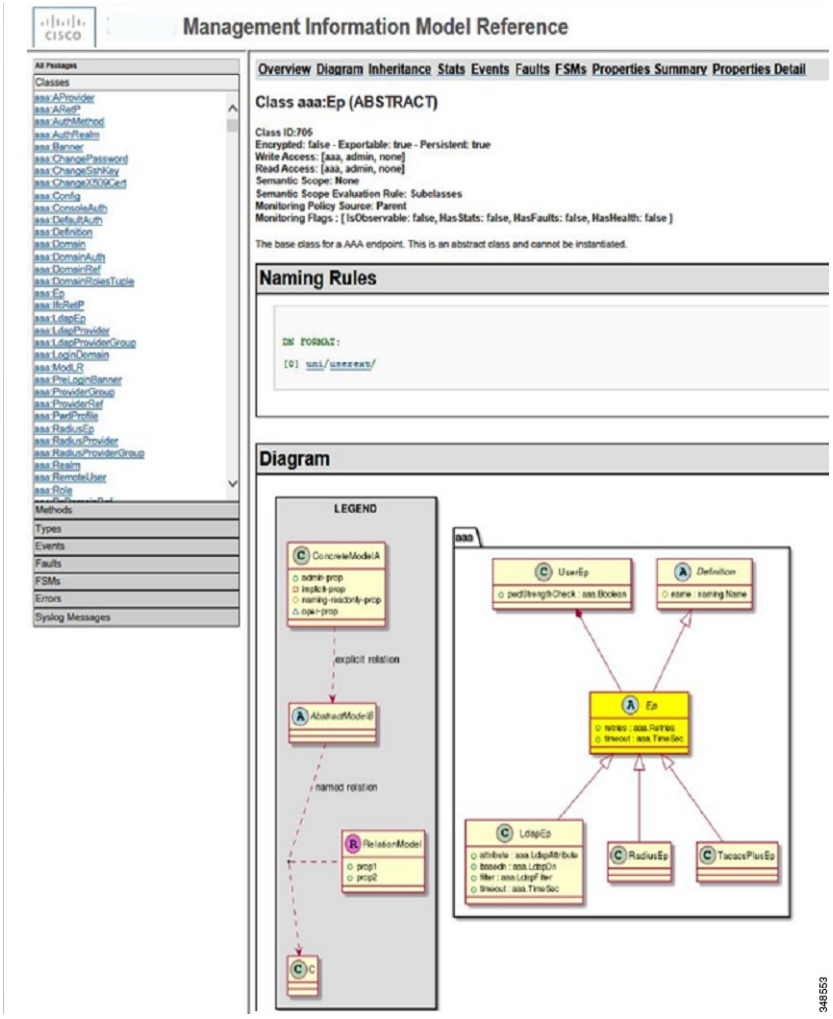


Visore 管理対象オブジェクト ビューアは次の場所にあります。`http(s)://host[:port]/visore.html`

管理情報モデルのリファレンス

管理情報モデル（MIM）には、システム内のすべての管理対象オブジェクトとそのプロパティが含まれます。MIT内のオブジェクトを検索するために管理者がどのようにMIMを使用できるかに関する例については、次の図を参照してください。

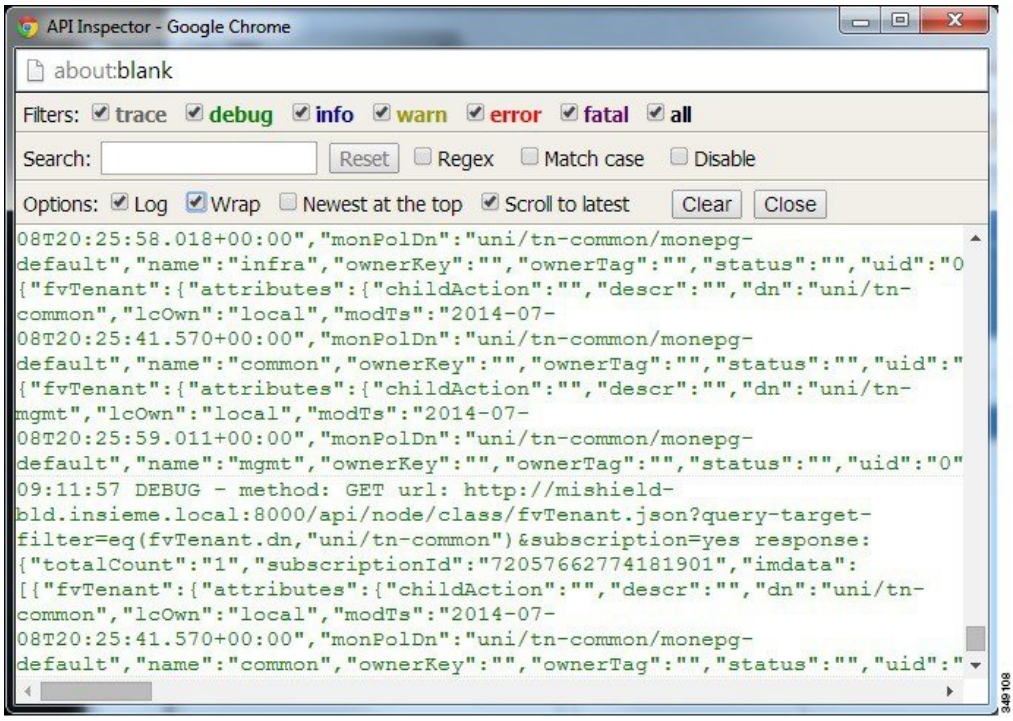
図 *69*：*MIM* リファレンス



API インスペクタ

API インスペクタでは、APIC が GUI インタラクションを実行するために処理する REST API コマンドのリアルタイム表示が提供されます。下の図は、API インスペクタが GUI の主要テナントのセクションに移動する場合に表示する REST API コマンドを示します。

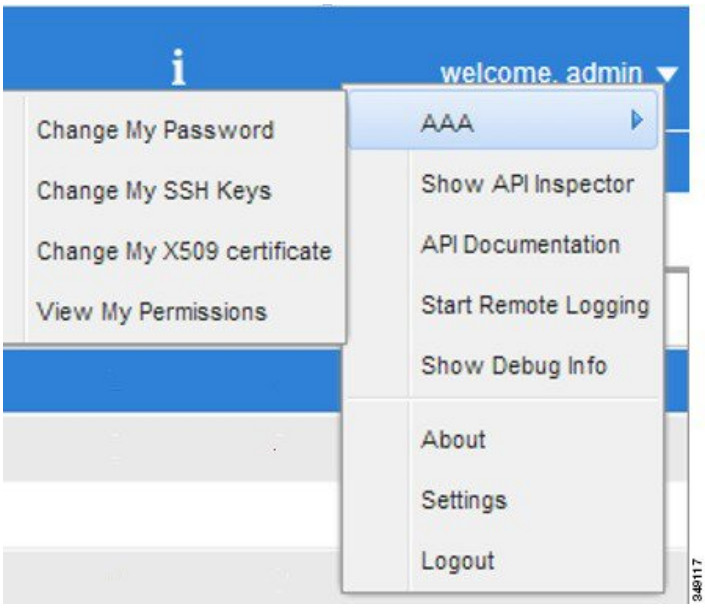
図 *70*：*API* インスペクタ



ユーザ ログインのメニュー オプション

ユーザ ログインのドロップダウン メニューにより、複数の設定、診断、参照およびプリファレンスのオプションが提供されます。次の図は、このドロップダウン メニューを示します。

図 *71*：ユーザ ログインのメニュー オプション



オプションには次のものが含まれます。

- ユーザ パスワード、SSH キー、X509 証明書を変更、およびログインしたユーザの権限を表示するための AAA オプション。

（注） すべてのデバイスのシステム クロックが正確であることを保証するため、ACI ファブリックは、アクティブなネットワーク タイム プロトコル（NTP）ポリシーで設定する必要があります。そうしないと、同期していない証明書がノードで拒否される可能性があります。

- [Show API Inspector] では、API インスペクタが開きます。
- [API Documentation] では、管理情報モデルの参照を開きます。
- リモート ロギング。
- デバッグ情報。
- ソフトウェアの現在のバージョン番号について。
- GUI を使用するためのプリファレンスの設定。

- システムを終了するためのログアウト。

GUI および CLI のバナーのカスタマイズ

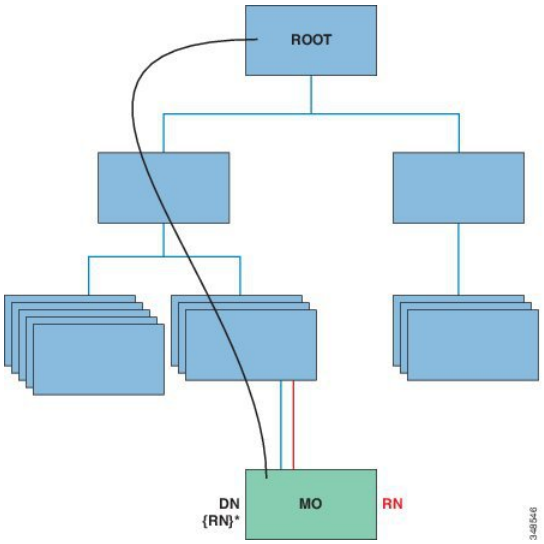
GUI および CLI バナーは、GUI の [Admin] > [AAA] > [Security management] セクションにあります。ユーザ ログイン認証の前に CLI バナーが表示されます。CLI バナーは、コンソールにそのまま表示されるテキスト ベースの文字列です。ユーザ ログイン認証の前に GUI バナーが表示されます。GUI バナーは URL です。URL は、iFrame に置かれることを許可する必要があります。URL の `x-frame-option` が `deny` または `sameorigin` に設定されていると、ユーザ認証の前にログイン URL が表示されません。

MIT 内のオブジェクトの検索

Cisco ACI は情報モデルベースのアーキテクチャ（管理情報ツリー（MIT））を使用しており、管理プロセスによって制御できるすべての情報がモデルによって説明されます。オブジェクト インスタンスは管理対象オブジェクト（MO）と呼ばれます。

次の図は、任意の MO インスタンスを一意的に表す識別名と、親 MO の下にある MO をローカル的に表す相対名を示します。MIT 内のオブジェクトはすべて、ルート オブジェクトの下に存在します。

図 *72*：*MO* の識別名と相対名



システム内のすべての MO は固有の識別名（DN）によって識別されます。このアプローチにより、グローバルにオブジェクトを参照できます。またオブジェクトの識別名のほか、各オブジェクトを相対名（RN）で参照することもできます。相対名は、親オブジェクトに対して相対的にオブジェクトを識別します。指定されたオブジェクトの識別名は、親オブジェクトの識別名に相対名を加えることで取得できます。

DN は、オブジェクトを一意的に識別する一連の相対名です。

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

```
dn =”sys/ch/lcslot-1/lc/leafport-1”
```

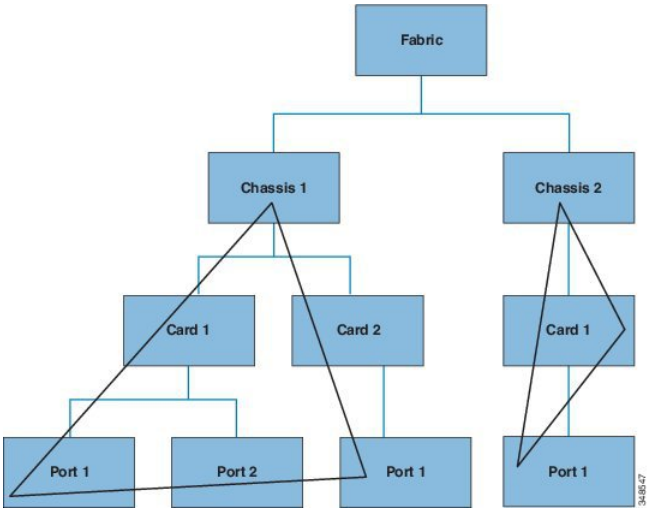
識別名はURLに直接マッピングされます。MIT内におけるオブジェクトの現在位置に応じて、相対名または識別名のいずれかを使用してオブジェクトにアクセスできます。

ツリーは階層型で構成され、属性システムを使用してオブジェクト クラスを識別できるため、さまざまな方法で管理対象オブジェクトの情報を取得するためにツリー内を照会できます。クエリは、識別名を使用してオブジェクト自体に対して実行するか、スイッチ シャーシなどのオブジェクトのクラスに対して実行するか、ツリー レベルで実行してオブジェクトのすべてのメンバーを検出できます。

ツリーレベルのクエリ

次の図は、クエリ対象の 2 つのシャーシをツリー レベルで示しています。

図 *73*：ツリーレベルのクエリ

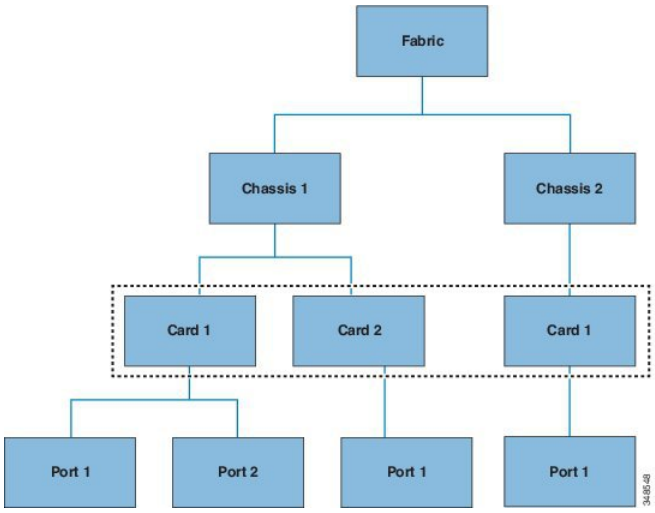


どちらのクエリも、参照されたオブジェクトと、その子オブジェクトを返します。このアプローチは、大規模なシステムのコンポーネントを検出するために役立ちます。この例では、クエリにより指定されたスイッチ シャーシのカードとポートが検出されます。

クラスレベル クエリ

次の図は、2 番目のクエリ タイプ、クラスレベル クエリを示します。

図 *74*：クラスレベル クエリ



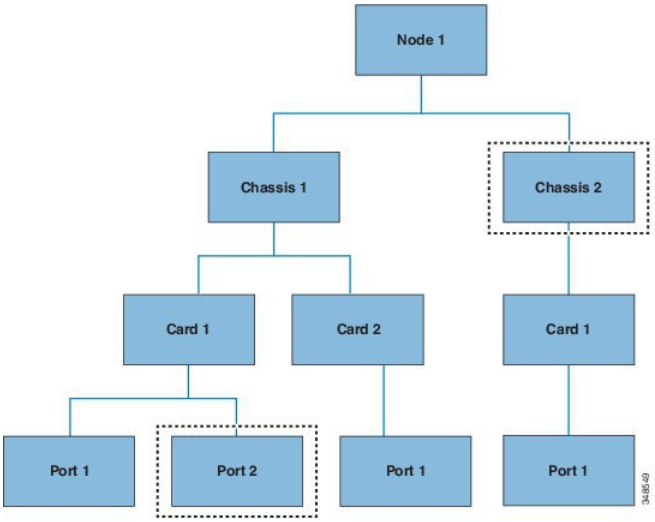
クラスレベルクエリは、任意のクラスのオブジェクトをすべて返します。このアプローチは、MIT で使用できる特定のタイプのオブジェクトをすべて検出する場合に役立ちます。この例で使用しているクラスはカードで、カード タイプのすべてのオブジェクトを返します。

オブジェクトレベル クエリ

3 つ目のクエリ タイプはオブジェクトレベル クエリです。オブジェクトレベル クエリでは、識別名を使用して特定のオブジェクトを返します。次の図は、2 つのオブジェクトレベル クエリを示

しており、1 つはノード 1/シャーシ 2、もう 1 つはノード 1/シャーシ 1/カード 1/ポート 2 を照会しています。

図 *75*：オブジェクトレベル クエリ



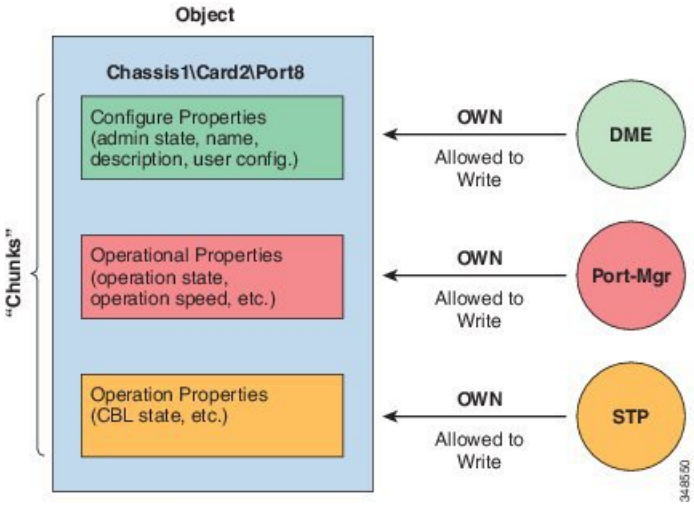
すべての MIT クエリで、管理者はサブツリー全体またはサブツリーの一部を返すよう選択できます。また、システム内のロールベース アクセス コントロール（RBAC）メカニズムによって、返されるオブジェクトが決まります。必ず、ユーザが表示権限を持つオブジェクトのみが返されます。

管理対象オブジェクトのプロパティ

Cisco ACI の管理対象オブジェクトには、管理対象オブジェクトを定義するプロパティが含まれています。管理対象オブジェクトのプロパティはチャンクに分割され、オペレーティング システム内でプロセスによって管理されます。オブジェクトには、複数のプロセスがアクセスする場合が

あります。これらのプロパティはすべて実行時にまとめてコンパイルされ、単一のオブジェクトとしてユーザに提示されます。次の図は、この関係の例を示します。

図 *76*：管理対象オブジェクトのプロパティ



オブジェクトの例には、オブジェクト内のプロパティ チャンクに書き込むプロセスが 3 つあります。Cisco APIC（つまりユーザ）とオブジェクトとの間のインターフェイスとなるデータ管理エンジン（DME）、ポートの構成を処理するポート マネージャ、およびスパニング ツリー プロトコル（STP）のすべてが、このオブジェクトのチャンクとやりとりします。APIC は、実行時にコンパイルされる単一のエンティティとしてオブジェクトをユーザに提示します。

REST インターフェイスによるオブジェクト データへのアクセス

REST は、World Wide Web などの分散型システム用ソフトウェア アーキテクチャの形式で、形式がシンプルであるため、Simple Object Access Protocol（SOAP）や Web サービス記述言語（WSDL）など、その他の設計モデルに代わって採用される機会が増えています。Cisco APIC は REST インターフェイスをサポートしており、Cisco ACI ソリューション全体へのプログラムを通じたアクセスを実現します。

Cisco ACI のオブジェクトベース情報モデルは、REST インターフェイスに非常にうまく適合しています。URL と URI は識別名に直接マッピングされ、MIT 上のオブジェクトを識別でき、MIT 上のデータを XML または JSON 形式でエンコードされた自己完結型の構造化テキスト ツリー ドキュメントとして記述できます。オブジェクトには、識別名とプロパティを使用して識別される親子関係があり、この関係は一連の作成、読み取り、更新、および削除（CRUD）操作によって読み取りと変更が可能です。

オブジェクトにアクセスするには、明確に定義されたアドレスである REST URL を使用します。Cisco APIC オブジェクト データを取得および操作するには標準の HTTP コマンドを使用します。使用できる URL の形式は次のとおりです。

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

URL の前に指定する各構成要素は、次のとおりです。

- system：システム識別子、IP アドレスまたは DNS で解決可能なホスト名
- mo | class：これが MIT 内の MO かまたはクラスレベルのクエリかどうかの表示
- class：照会するオブジェクトの MO クラス（情報モデルでの指定に従う）。クラス名は、<pkgName><ManagedObjectClassName> で表されます。
- dn：照会するオブジェクトの識別名（MIT 内のオブジェクトの一意の階層名）
- method：オブジェクトに対して呼び出すメソッドの指定（オプション）。HTTP POST リクエストにのみ適用されます。
- xml | json：エンコード形式
- options：クエリ オプション、フィルタ、引数

REST URL で個々のオブジェクトまたはオブジェクト クラスのアドレスを指定してアクセスできる機能により、管理者はオブジェクト ツリー全体、つまりシステム全体にプログラムを通じて完全にアクセスできます。

次に、REST クエリの例を示します。

- テナント solar 下のすべての EPG と障害を検索する。

  http://192.168.10.1:7580/api/mo/uni/tn-solar.xml?query-target=subtree&target-subtree-class=fvAEPg&rsp-subtree-include=faults

- フィルタされた EPG クエリ

  http://192.168.10.1:7580/api/class/fvAEPg.xml?query-target-filter=eq(fvAEPg.fabEncap,%20"vxlan-12780288")

エクスポート/インポートの設定

すべての APIC ポリシーおよび設定データは、バックアップの作成のためにエクスポートできます。これは、エクスポート ポリシーを使用して設定でき、リモート サーバにスケジュール バックアップまたは即時バックアップできます。スケジュール バックアップは、定期バックアップ ジョブまたは繰り返しバックアップ ジョブを実行するように設定できます。デフォルトでは、すべてのポリシーとテナントがバックアップされますが、管理者は管理情報ツリーの特定のサブツリーだけを選択して指定できます。バックアップは、インポート ポリシーによって APIC にインポートでき、システムを以前の設定に復元できます。

コンフィギュレーション データベースのシャーディング

APIC クラスタは、シャーディングと呼ばれる大規模データベース テクノロジーを使用します。このテクノロジーにより、APICが生成および処理するデータセットの拡張性と信頼性が向上します。APIC構成のデータは、論理的制限のあるサブセットに分割されます。このサブセットはシャードと呼ばれ、データベースのシャードと類似しています。シャードはデータ管理の単位となり、APIC はシャードを次のように管理します。

- 各シャードには 3 つの複製があります。

- シャードは、APIC クラスタを構成する複数のアプライアンスに均等に配分されます。

各 APIC アプライアンスに 1 つ以上のシャードが配置されます。シャード データの割り当ては所定のハッシュ関数に基づいており、スタティック シャードのレイアウトによってアプライアンスへのシャード割り当てが決定します。

設定ファイルの暗号化

リリース 1.1(2) 以降、AES-256 暗号化を有効にすることにより APIC 設定ファイルを暗号化できます。AES 暗号化はグローバル設定オプションです。すべてのセキュア プロパティは AES 構成設定に従っています。テナント設定などの ACI ファブリック設定のサブセットを AES 暗号化を使用してエクスポートするが、ファブリック設定の残りの部分は暗号化しないということはできません。セキュア プロパティのリストについては、『Cisco Application Centric Infrastructure Fundamentals』の「Appendix K: Secure Properties」を参照してください。

APIC は、16 ～ 32 文字のパスフレーズを使用して AES-256 キーを生成します。APIC GUI では、AES パスフレーズのハッシュを表示します。このハッシュを使用して、2 つの ACI ファブリックで同じパスフレーズが使用されているかどうかを確認できます。このハッシュをクライアント コンピュータにコピーして、別の ACI ファブリックのパスフレーズ ハッシュと比較できます。これにより、それらのハッシュが同じパスフレーズを使用して生成されたかどうかを確認できます。ハッシュを使用して、元のパスフレーズまたは AES-256 キーを再構築することはできません。

暗号化された設定ファイルを使用する際は、次のガイドラインに従ってください。

- AES 暗号化設定オプションを使用しているファブリックに古い ACI 設定をインポートするための後方互換性がサポートされています。

（注） 逆の互換性はサポートされていません。AES 暗号化が有効になっている ACI ファブリックからエクスポートされた設定を古いバージョンの APIC ソフトウェアにインポートすることはできません。

- ファブリック バックアップ設定のエクスポートを実行するときは、必ず AES 暗号化を有効にします。これにより、ファブリックを復元するときに、設定のすべてのセキュア プロパティが正常にインポートされるようになります。

（注） AES 暗号化を有効にせずにファブリック バックアップ設定がエクスポートされると、どのセキュア プロパティもエクスポートに含まれません。そのような暗号化されていないバックアップにはセキュア プロパティは何も含まれていないため、そのようなファイルをインポートしてシステムを復元すると、ファブリックの管理者およびすべてのユーザがシステムからロックアウトされてしまう可能性があります。

- 暗号化キーを生成する AES パスフレーズは、ACI 管理者やその他のユーザが復元したり読み取ったりすることはできません。AES パスフレーズは保存されません。APIC は、AES パス

フレーズを使用して AES キーを生成した後でそのパスフレーズを廃棄します。AES キーはエクスポートされません。AES キーは、エクスポートされず、REST API を使用して取得できないため、復元できません。

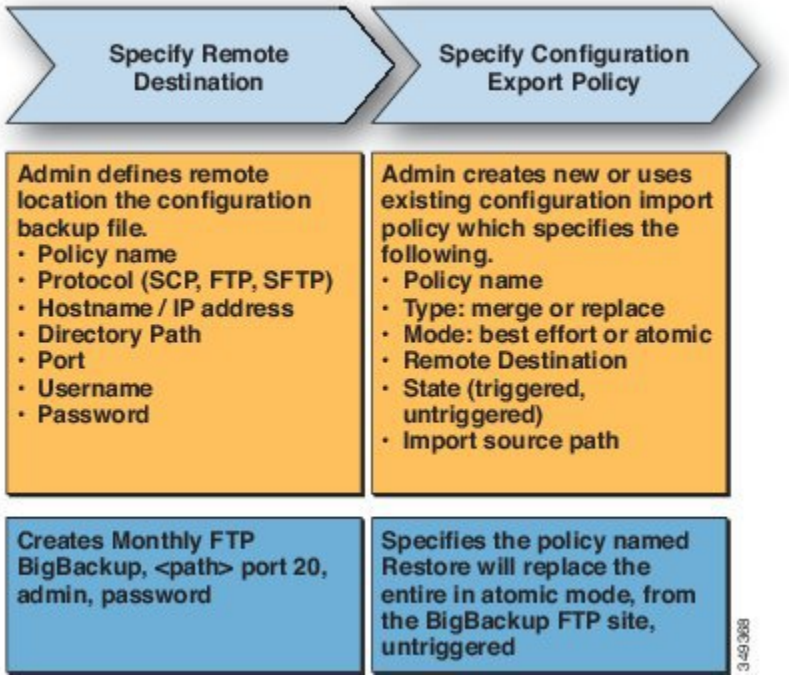
- 同じ AES-256 パスフレーズは、常に同じ AES-256 キーを生成します。設定のエクスポートファイルは、同じ AES パスフレーズを使用する他の ACI ファブリックにインポートできます。
- トラブルシューティングを目的として、セキュアプロパティの暗号化データが含まれていない設定ファイルをエクスポートします。設定のエクスポートを実行する前に一時的に暗号化をオフにすると、エクスポートされた設定からすべてのセキュアプロパティ値が削除されます。すべてのセキュアプロパティが削除されたそのような設定ファイルをインポートするには、インポート マージ モードを使用します。インポート置換モードは使用しません。インポート マージ モードを使用すると、ACI ファブリック内の既存セキュア プロパティが保持されます。
- デフォルトでは、APICは復号できないフィールドが含まれているファイルの設定のインポートを拒否します。この設定をオフにするときは注意してください。このデフォルト設定がオフになっているときに設定のインポートが適切に実行されないと、ファブリックのAES暗号化設定に一致しない設定ファイルのインポート時にACIファブリックのすべてのパスワードが削除される可能性があります。

（注） このガイドラインに従わないと、ファブリック管理者を含むすべてのユーザがシステムからロックアウトされる可能性があります。

設定のエクスポート

次の図は、エクスポート ポリシーを設定するプロセスがどのように動作するかを示します。

図 77：エクスポート ポリシーを設定するワークフロー



APIC はこのポリシーを次のように適用します。

- 完全なシステム構成のバックアップは月に一度実行されます。
- バックアップは BigBackup FTP サイトに XML 形式で保存されます。
- ポリシーがトリガーされます（有効です）。

設定のインポート

管理者は、次の 2 つのモードのいずれかで、インポートを実行するインポート ポリシーを作成できます。

- best-effort：インポートできないシャード内のオブジェクトを無視します。着信した設定のバージョンが既存システムと互換性がある場合、互換性のないシャードはインポートされず、インポート可能なシャードについてはインポート処理が進められます。

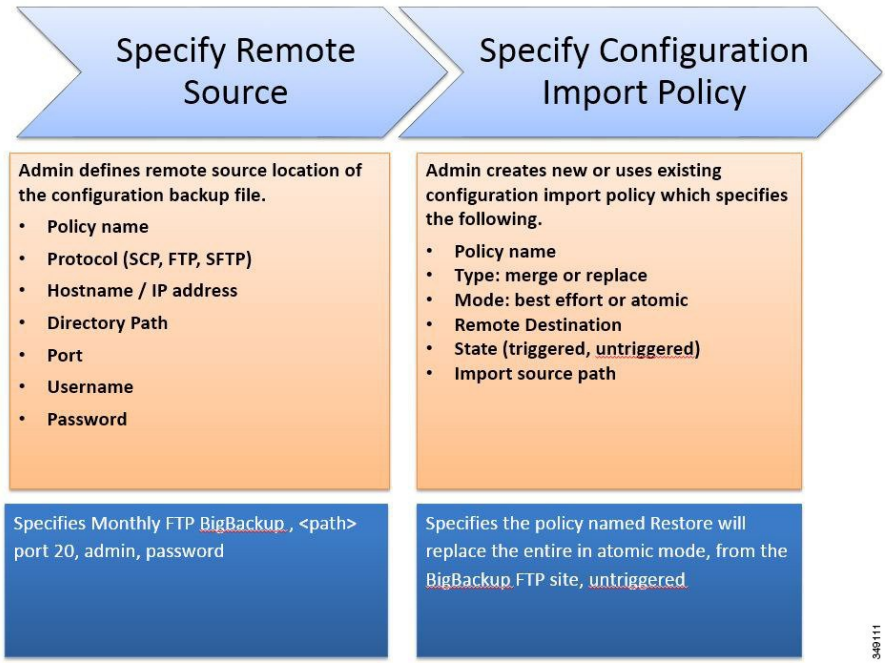
- Atomic：インポートできないオブジェクトを含むシャードは無視し、インポート可能なシャードについては処理が進められます。着信した設定のバージョンに既存のシステムとの互換性がない場合は、インポートが終了します。

インポート ポリシーでサポートされるモードおよびタイプの組み合わせは次のとおりです。

- Best-effort Merge：インポートした設定が既存の設定にマージされますが、インポートできないオブジェクトは無視されます。
- Atomic Merge：インポートした設定が既存の設定にマージされますが、インポートできないオブジェクトを含むシャードは無視されます。
- Atomic Replace：インポートした設定データで既存の設定を上書きします。既存の設定には存在するがインポートした設定には存在しないオブジェクトは、すべて削除されます。既存の設定内には子オブジェクトを持っているが着信したインポート済み設定内には持っていないオブジェクトは、既存の設定から削除されます。たとえば、2つのテナント、ソーラー、ウィンドを持った既存の設定において、テナントのウィンドが作成されたタイミングがインポート対象バックアップ設定の保存より後だった場合には、テナントのソーラーはバックアップから復元されますが、テナントのウィンドは削除されます。

次の図は、インポート ポリシーを設定するプロセスがどのように動作するかを示します。

図 *78*：インポート ポリシーを設定するワークフロー



APIC はこのポリシーを次のように適用します。

- 毎月のバックアップから完全なシステム構成の復元を実行するためのポリシーが作成されます。

- Atomic replace モードでは、次の処理が行われます。
  - 既存の設定を上書きする。
  - インポートしたファイルに存在しない既存の設定のオブジェクトを削除する。
  - 存在しない子オブジェクトを削除する。
- ポリシーはトリガーされません（使用できますが、アクティブ化されていません）。

テクニカル サポート、統計情報、コア

管理者は、APIC 内で、コア ファイルとデバッグ データを処理するために、統計情報、テクニカル サポートの収集、障害およびイベントをファブリック（APIC およびスイッチ）から外部ホストにエクスポートするようエクスポート ポリシーを設定できます。エクスポートはXML、JSON、Web ソケット、SCP、HTTP などのさまざまな形式にできます。エクスポートはサブスクライブでき、定期的またはオンデマンドでストリーミングできます。

（注） 統計情報エクスポート ポリシーの最大数は、テナント数とほぼ同じです。各テナントには複数の統計情報エクスポート ポリシーを設定でき、複数のテナントで同じエクスポート ポリシーを共有することができますが、ポリシーの総数はテナント数とほぼ同数に制限されます。

管理者は、転送プロトコル、圧縮アルゴリズム、転送の頻度などポリシーの詳細を設定できます。ポリシーは、AAA を使用して認証されたユーザによって設定できます。実際の転送のセキュリティ メカニズムは、ユーザ名とパスワードに基づいています。内部的に、ポリシー要素はデータのトリガーを処理します。

第 12 章

モニタリング

この章の内容は、次のとおりです。

障害、エラー、イベント、監査ログ

（注） 障害、イベント、エラー、システム メッセージに関する情報については、Web ベースのアプリケーションである『*Cisco APIC Faults, Events, and System Messages Management Guide*』および『*Cisco APIC Management Information Model Reference*』を参照してください。

APIC は、MO の集合形式で ACI ファブリック システムの管理および操作状態の包括的な現在のランタイム表現を維持します。システムは、これらのプロセスを管理するためにシステムとシステムおよびユーザが作成するポリシーのランタイム状態に従って、障害、エラー、イベント、および監査ログ データを生成します。

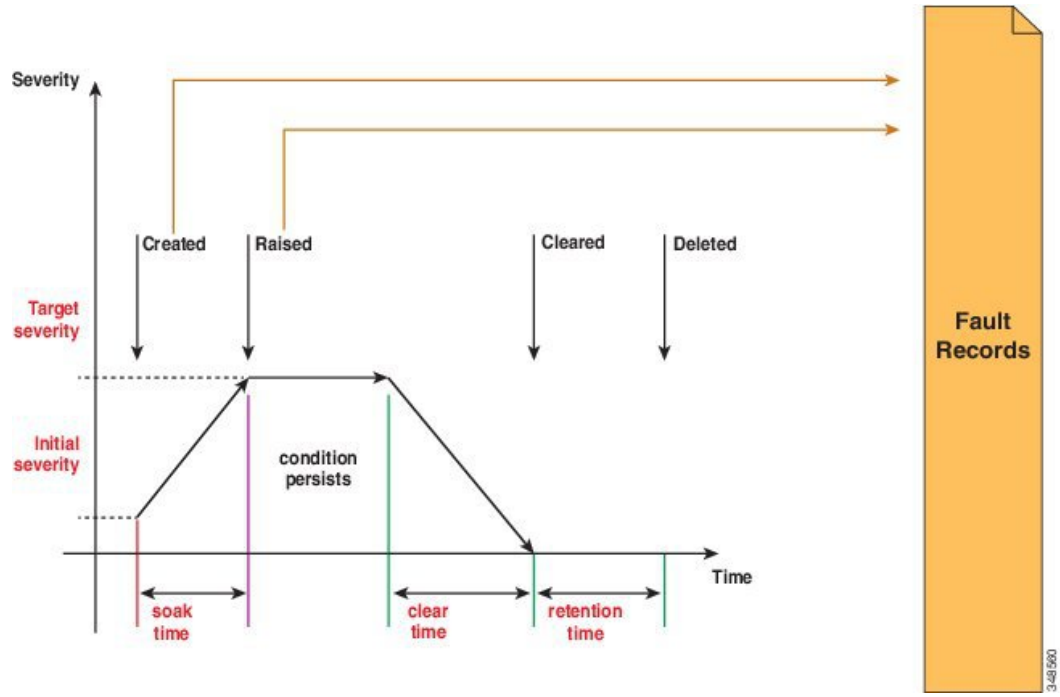
障害

システムの実行時の状態に基づいて、APIC は自動的に異常を検出し、障害を表す障害オブジェクトを作成します。障害オブジェクトには、ユーザが問題を診断してその影響を評価するのに役立ち、解決策を提供するように作られているさまざまなプロパティが含まれます。

たとえば、高いパリティ エラー率などポートに関連する問題をシステムが検出すると、障害オブジェクトが自動的に作成され、ポート オブジェクトの子として管理情報ツリー（MIT）内に配置されます。同じ状況が複数回検出される場合、障害オブジェクトの追加インスタンスは作成され

ません。障害を引き起こした状況が修正された後、障害オブジェクトは障害のライフサイクル ポリシーで指定された一定期間保存され、最終的に削除されます。次の図を参照してください。

図 *79*：障害のライフサイクル



ライフサイクルは問題の現在の状態を表します。サイクルは問題が最初に検出されると、そのソーク時間で開始され、提起された状態へと変わって、問題がまだ存在するとその状態のままになります。状態がクリアされると、「raised-clearing」と呼ばれるステータスに移行します。そのステータスでは、その状態がまだ存在する可能性があると見なされます。次に、「clearing time」に移行し、最終的に「retaining」に移行します。この時点で、問題は解決されたと見なされ、ユーザが最近解決された問題を確認できるようにする目的のために障害オブジェクトは保持されます。

ライフサイクルの移行が発生するたびに、システムは自動的にそれを記録する障害記録オブジェクトを作成します。障害レコードは、作成後は変更されることはなく、レコード数が障害保持ポリシーで指定された最大値を超えた場合にのみ削除されます。

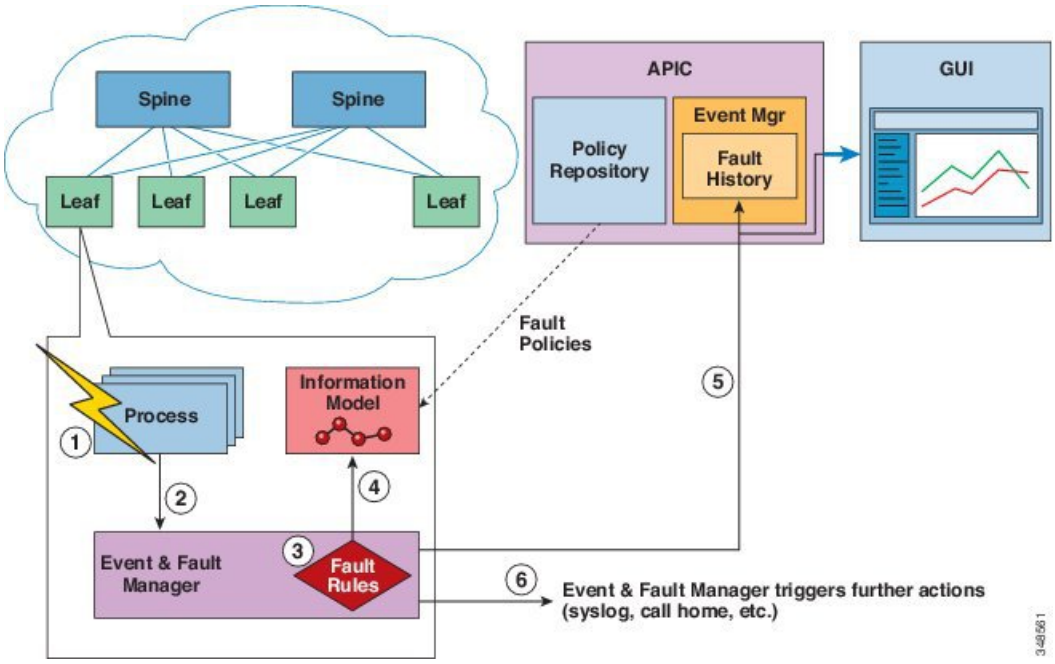
重大度は、サービスを提供するシステムの機能に対するその状態の影響の概算値です。考えられる値は、Warning、Minor、Major および Critical です。Warning に相当する重大度の障害は、導入されているサービスには現在影響を与えていない潜在的な問題を示します（たとえば、不完全または矛盾した設定など）。Minor および Major の障害は、提供されるサービスが低下する可能性があることを示します。Critical は、大規模な停電がサービスを著しく低下させていたり、同時にサービスが悪化していることを意味します。説明には、追加情報を提供したりトラブルシューティングに役立てるために用意された人間に解読可能な問題の説明が含まれます。

イベント

イベント レコードは、ユーザにとって重要な可能性がある特定の状態の発生を記録するためにシステムによって作成されるオブジェクトです。レコードには、影響を受けるオブジェクトの完全修飾ドメイン名（FQDN）、タイムスタンプおよび状態の説明が含まれます。例には、リンク状態遷移、プロトコルの開始と停止、および新しいハードウェア コンポーネントの検出が含まれます。イベント レコードは、作成後は変更されることはなく、レコード数がイベント保持ポリシーで指定された最大値を超えた場合にのみ削除されます。

次の図は、障害とイベントに関するレポートを作成するプロセスを示します。

図 *80*：障害およびイベントのレポート/エクスポート



1. プロセスが障害のある状態を検出します。
2. プロセスが Event and Fault Manager に通知します。
3. Event and Fault Manager は障害ルールに従って通知を処理します。
4. Event and Fault Manager は、MIM で障害インスタンスを作成し、障害ポリシーに従ってそのライフ サイクルを管理します。
5. Event and Fault Manager は、APIC および接続されたクライアントに状態遷移を通知します。
6. Event and Fault Manager は、追加のアクションをトリガーします（syslog や Call Home など）。

エラー

APIC エラー メッセージは通常、APIC GUI および APIC CLI に表示されます。これらのエラー メッセージは、ユーザが実行している処理、あるいはユーザが設定または管理しているオブジェクトに固有のものです。これらのメッセージには次のようなものがあります。

- 情報メッセージ。実行している処理のヘルプおよびヒントを提供します。
- 警告メッセージ。ユーザが設定または管理しているオブジェクト（ユーザアカウントやサービス プロファイルなど）に関連するシステム エラーの情報を提供します。
- Finite State Machine（FSM）のステータス メッセージ。FSM 段階のステータスに関する情報を提供します。

多くのエラー メッセージには、1 つまたは複数の変数が含まれます。これらの変数を置き換えるために APIC が使用する情報は、メッセージのコンテキストによって決まります。一部のメッセージは、複数のタイプのエラーによって生成される場合があります。

監査ログ

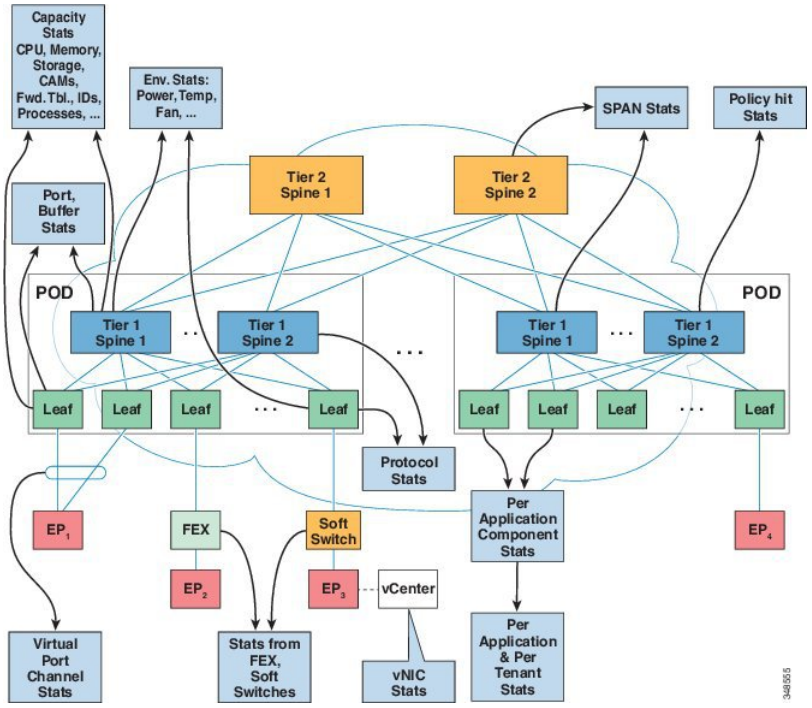
監査レコードは、ログイン/ログアウトや構成の変更などのユーザが開始するアクションを記録するためにシステムにより作成されるオブジェクトです。レコードには、アクションを実行したユーザの名前、タイムスタンプ、アクションの説明、また該当する場合は影響を受けたオブジェクトの FQDN が含まれます。監査レコードは、作成後は変更されることはなく、レコード数が監査保持ポリシーで指定された最大値を超えた場合にのみ削除されます。

統計情報プロパティ、階層、しきい値およびモニタリング

統計情報により、トレンド分析とトラブルシューティングが可能になります。統計情報収集は、収集を継続的にまたはオンデマンドで行うように設定できます。統計情報により、監視対象オブジェクトのリアルタイム測定が提供されます。統計情報は、累積カウンタとゲージで収集できます。次の図を参照してください。

ポリシーは、収集する統計情報の種類、間隔、実行するアクションを定義します。たとえば、入力 VLAN でドロップされたパケットのしきい値が毎秒 1000 を超える場合、ポリシーは EPG 上で 1 つの障害を生成することができます。

図 *81*：統計情報のさまざまな送信元



統計情報データは、インターフェイス、VLAN、EPG、アプリケーションプロファイル、ACL ルール、テナント、内部 APIC プロセスなどのさまざまな送信元から収集されます。統計情報は、5 分、15 分、1 時間、1 日、1 週間、1 ヵ月、4 半期、または 1 年のサンプリング間隔でデータを蓄積します。短い期間の間隔によって、長い間隔が与えられます。

さまざまな統計情報プロパティを利用でき、[last value]、[cumulative]、[periodic]、[rate of change]、[trend]、[maximum]、[min]、[average] などがあります。収集/保持時間は設定できます。ポリシーは、統計情報をシステムの現在の状態から収集するか、履歴的に蓄積するか、またはその両方かを指定できます。たとえば、ポリシーは、履歴統計を 1 時間にわたって 5 分間隔で収集するように指定できます。1 時間は移動ウィンドウです。1 時間が経過すると、次の 5 分間の統計情報が追加され、一番最初の 5 分間に収集されたデータが放棄されます。

（注） 5 分間分の粒度サンプル レコードの数は最大 12（1 時間分の統計情報）に限定されます。他のサンプル間隔の場合はすべて、サンプル レコード 1,000 個までに限定されます。たとえば、1 時間の粒度統計情報は 41 日間まで保持できます。

モニタリング ポリシーの設定

管理者は、次の 4 つの広い範囲でモニタリング ポリシーを作成できます。

- ファブリック全体：ファブリック オブジェクトとアクセス オブジェクトの両方が含まれます。
- アクセス（別名インフラストラクチャ）：アクセス ポート、FEX、VM コントローラなど
- ファブリック：ファブリック ポート、カード、シャーシ、ファンなど
- テナント：EPG、アプリケーション プロファイル、サービスなど

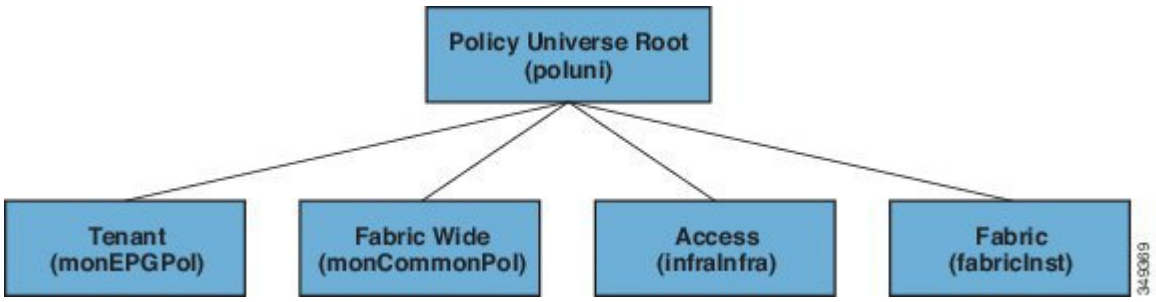
APIC には、デフォルトのモニタリング ポリシーの次の 4 つのクラスが含まれます。

- `monCommonPol`（uni/fabric/moncommon）：ファブリック インフラストラクチャ階層とアクセス インフラストラクチャ階層の両方に適用されます。
- `monFabricPol`（uni/fabric/monfab-default）：ファブリック階層に適用されます。
- `monInfraPol`（uni/infra/monifra-default）：アクセス インフラストラクチャ階層に適用されます。
- `monEPGPol`（uni/tn-common/monepg-default）：テナント階層に適用されます。

モニタリング ポリシーの 4 つのクラスそれぞれにおいて、デフォルト ポリシーは特定のポリシーによって上書きできます。たとえば、Solar テナント（*tn-solar*）に適用されたモニタリング ポリシーは、他のテナントがまだデフォルト ポリシーによってモニタされている一方で、Solar テナントのデフォルト ポリシーを上書きします。

次の図の 4 つのオブジェクトのそれぞれには、モニタリングのターゲットが含まれます。

図 *82*：デフォルト モニタリング ポリシーの *4* つのクラス



インフラ モニタリング ポリシーには `monInfra` ターゲットが含まれ、ファブリック モニタリング ポリシーには `monFab` ターゲットが含まれ、テナント モニタリング ポリシーには `monEPG` ターゲットが含まれます。各ターゲットは、この階層内のオブジェクトの対応するクラスを表します。たとえば、`monInfra-default` モニタリング ポリシーには、FEX ファブリック対面ポートを表すターゲットがあります。これらの FEX ファブリック対面ポートのモニタリング方法に関するポリシーの詳細はこのターゲットに含まれています。ターゲットに適用できるポリシーのみがそのターゲッ

ト下で許可されます。考えられるターゲットすべてがデフォルトで自動作成されるわけではないことに注意してください。管理者は、ターゲットがない場合にポリシー下でターゲットを追加できます。

共通モニタリング ポリシー（monCommonPol）にはグローバルなファブリック全体のスコープが含まれ、APIC コントローラを含むファブリック内のすべてのノードで自動的に導入されます。共通モニタリング ポリシーの下にあるあらゆる発信元（syslog、callhome、SNMP など）は、すべての障害、イベント、監査、ヘルスのオカレンスをキャプチャします。1 つの共通モニタリング ポリシーで、ファブリック全体を監視します。Syslog および SNMP の重大度または callhome の緊急度のしきい値は、ファブリック管理者が適切と判断する詳細レベルに応じて設定できます。

複数のモニタリング ポリシーを使用してファブリック内の個々の部分を独立して監視することができます。たとえば、グローバル モニタリング ポリシー下のソースはグローバル ビューを反映します。それとは別の、一部ノードだけに導入されているカスタム モニタリング ポリシー下のソースは、電源を綿密に監視します。あるいは、さまざまなテナントの特定の障害またはイベントのオカレンスをリダイレクトして特定のオペレータに通知することも可能です。

他のモニタリング ポリシーの下にあるソースは、比較的小さなスコープ内で障害、イベント、および監査をキャプチャします。モニタリング ポリシーの直下にあるソースは、スコープ内のすべてのオカレンス（たとえばファブリックやインフラなど）をキャプチャします。ターゲットの下にあるソースは、ターゲットに関連するすべてのオカレンス（たとえば電源用の eqpt:Psu）をキャプチャします。障害/イベント重大度割り当てポリシーの下にあるソースは、障害/イベント コードによって識別される特定の障害やイベントに一致するオカレンスのみキャプチャします。

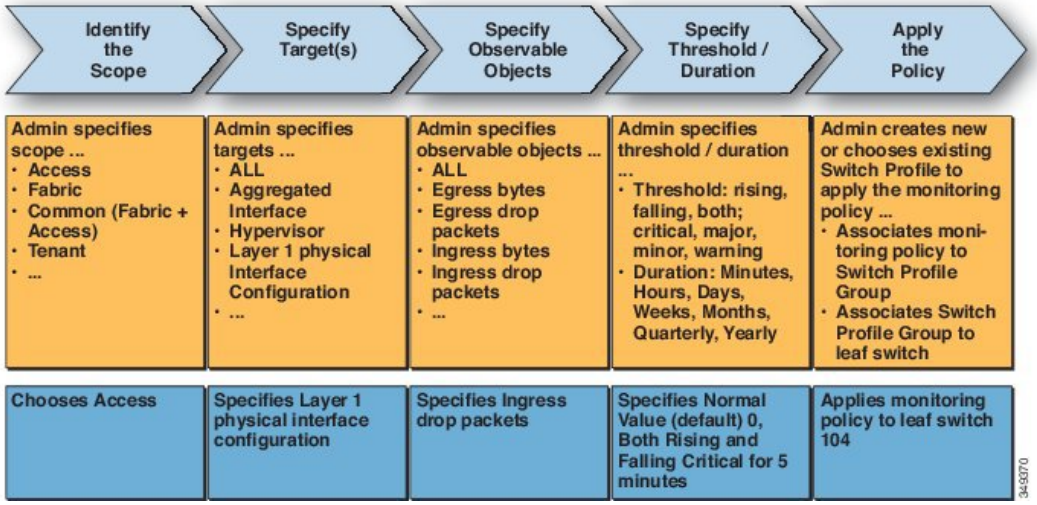
障害/イベント/監査が生成されると、該当するすべてのソースが使用されます。たとえば、次のような設定について考えてみましょう。

- Syslog グループ 4 を参照する Syslog ソース 4 が障害 F0123 に定義されています。
- Syslog グループ 3 を参照する Syslog ソース 3 が電源ターゲット（eqpt:Psu）定義されています。
- Syslog グループ 2 を参照する Syslog ソース 2 がスコープ インフラに定義されています。
- Syslog グループ 1 を参照する Syslog ソース 1 が共通モニタリング ポリシーに定義されています。

スコープ インフラ内のクラス eqpt Psu の MO で障害 F0123 が発生すると、メッセージの重大度がそれぞれの送信元および宛先に定義された最小値以上であることを前提として、syslog グループ 1 ～ 4 内のすべての宛先に syslog メッセージが送信されます。この例では syslog 設定について述べていますが、callhome 設定と SNMP 設定も同じように機能します。

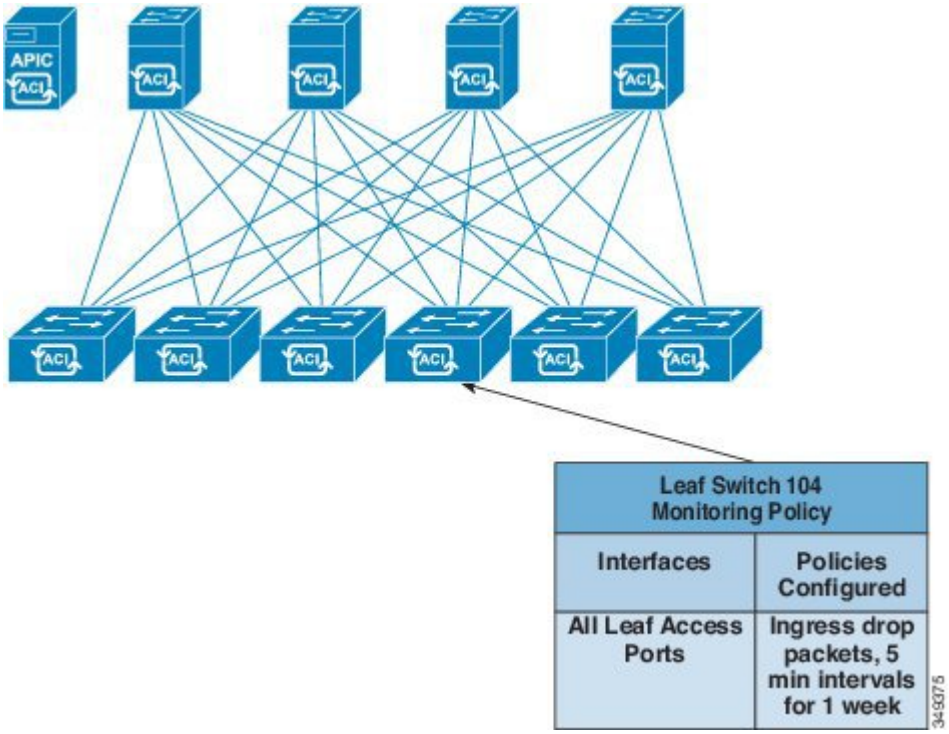
次の図は、統計情報用のファブリック モニタリング ポリシーを設定するプロセスがどのように動作するかを示します。

図 *83* : アクセス モニタリング ポリシーを設定するワークフロー



APIC は、次の図に示すように、このモニタリング ポリシーを適用します。

図 *84* : サンプルのアクセス モニタリング ポリシーの結果



モニタリング ポリシーは、障害やヘルス スコアなどの他のシステム操作に対しても設定できます。この階層へのモニタリング ポリシー マップの構造

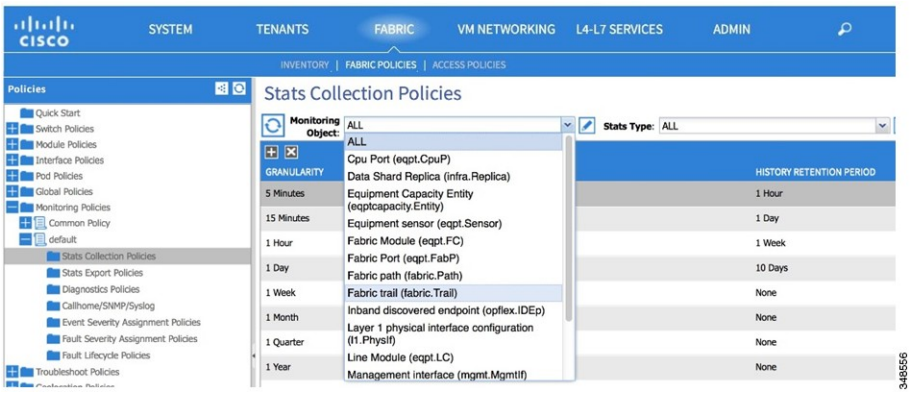
モニタリング ポリシー

- **統計情報のエクスポート**
- **収集ルール**
- **モニタリング ターゲット**
  - **統計情報のエクスポート**
  - **収集ルール**
  - **統計情報**
    - **収集ルール**
    - **しきい値ルール**
    - **統計情報のエクスポート**

次の図の [Statistics Export policies] オプションは、エクスポートする統計情報の形式と宛先を定義します。出力は、FTP、HTTP、またはSCPプロトコルを使用してエクスポートできます。形式はJSONまたはXMLです。ユーザまたは管理者は、出力を圧縮することもできます。エクスポートは、[Statistics]、[Monitoring Targets] または最上位のモニタリング ポリシー下で定義できます。統計情報のエクスポートの上位レベルの定義は、定義された下位レベルのポリシーが存在しない限り優先されます。

次の図に示すように、モニタリング ポリシーは、セレクタまたは関係を使用して、特定の監視可能なオブジェクト（ポート、カード、EPG、テナントなど）または監視可能なオブジェクトのグループに適用されます。

図 *85*：ファブリック統計情報収集ポリシー



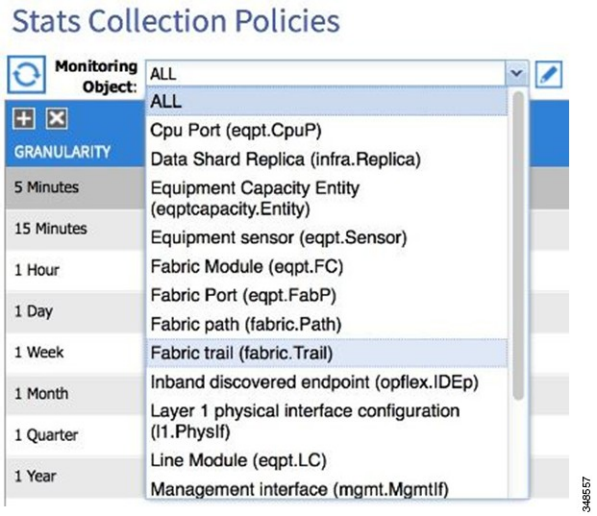
モニタリング ポリシーは次を定義します。

- 統計情報が収集され、履歴に保持されます。

- しきい値超過障害がトリガーされます。
- 統計情報がエクスポートされます。

次の図に示すように、収集ルールは、サンプリング間隔ごとに定義されます。

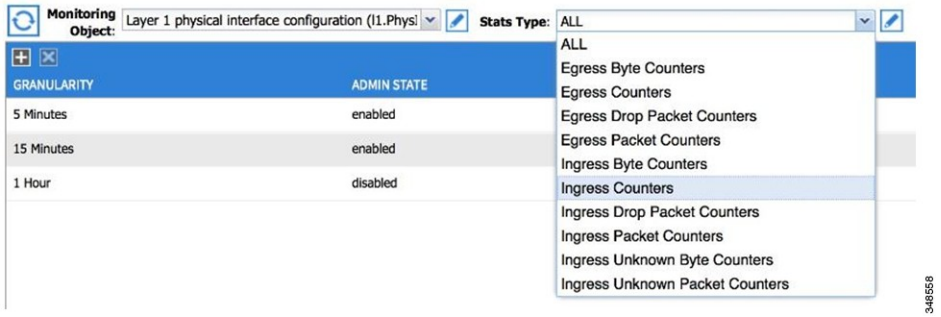
図 *86*：統計情報モニタリング間隔



情報統計の収集をオンまたはオフにする必要があるかどうか、またオンにした場合、履歴保持期間をどうするべきかを設定します。モニタリング ターゲットは、監視可能なオブジェクトに相当します（ポートや EPG など）。

統計情報は、統計カウンタのグループに相当します（入力カウンタ、出力カウンタ、またはドロップ カウンタなど）。

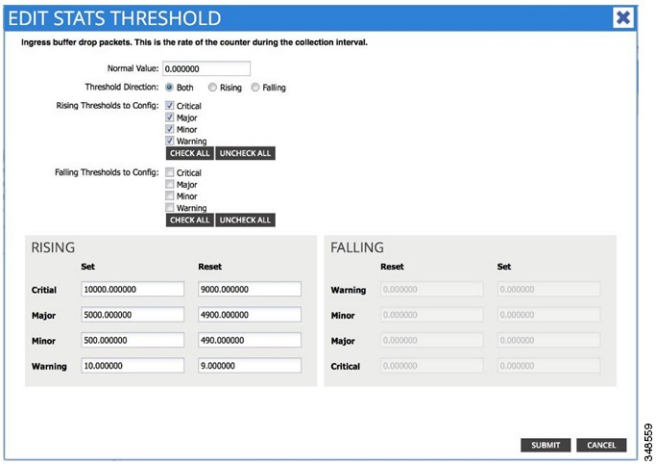
図 *87*：統計情報タイプ



収集ルールは、[Statistics]、[Monitoring Targets] または最上位のモニタリング ポリシー下で定義できます。収集ルールの上位レベルの定義は、定義された下位レベルのポリシーが存在しない限り優先されます。

次の図に示すように、しきい値ルールは収集ルール下で定義され、親収集ルールで定義される対応するサンプリング間隔に適用されます。

図 *88*：統計情報しきい値



第 13 章

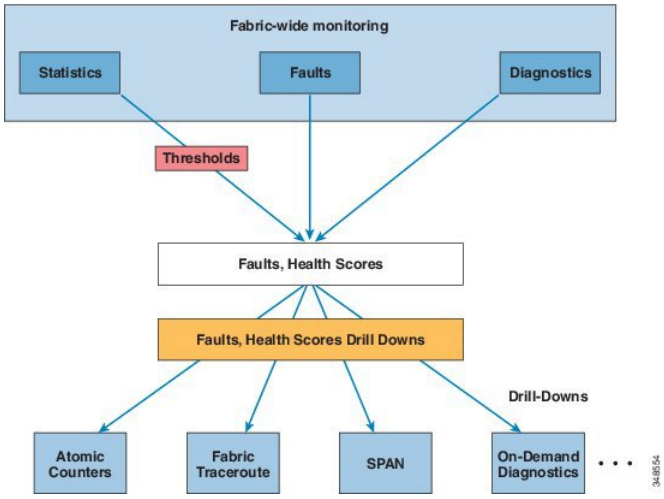
トラブルシューティング

この章の内容は、次のとおりです。

トラブルシューティング

ACI ファブリックでは、次の図に示すように広範なトラブルシューティングとモニタリングのツールが提供されます。

図 *89*：トラブルシューティング



ヘルス スコア

APIC ファブリックは、ポリシー モデルを使用してデータをヘルス スコアに組み入れます。ヘルス スコアは、システム、インフラストラクチャ、テナント、アプリケーション、またはサービスなどのさまざまな領域に集約できます。

ACI ファブリック ヘルス情報は、システムの次の表示画面で見ることができます。

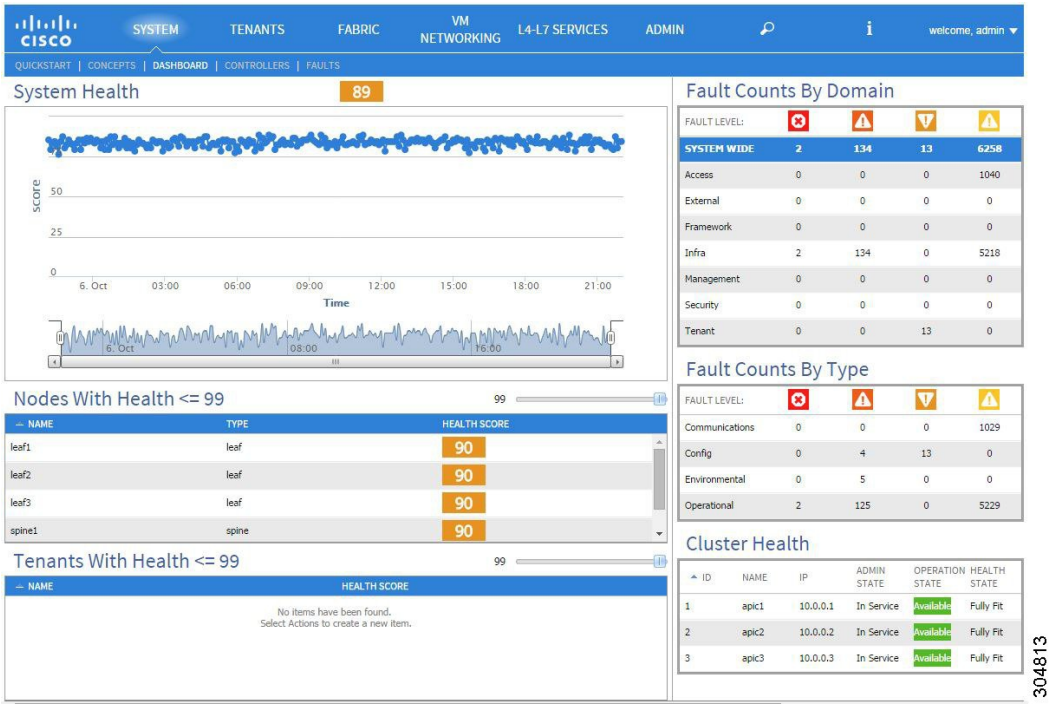
- [System]：ポッドのヘルス スコア、テナントのヘルス スコア、ドメインおよびタイプごとのシステム エラー数、APIC クラスタ ヘルス状態など、システム全体の健全性の集約。
- [Pod]：ポッド（スパインおよびリーフ スイッチのグループ）のヘルス スコアの集約、ドメインおよびタイプごとのポッド全体のエラー数。
- [Tenant]：テナント固有のアプリケーションおよび EPG などのオブジェクトのパフォーマンス データを含むテナントのヘルス スコアの集約、ドメインおよびタイプごとのテナント全体のエラー数。
- [Managed Object]：管理対象オブジェクト（MO）（独立 MO および関連 MO を含む）のヘルス スコア ポリシー。これらのポリシーは、管理者によりカスタマイズできます。

システムおよびポッドのヘルス スコア

システムおよびポッドのヘルス スコアは、リーフ/スパイン スイッチのヘルス スコアと、リーフ スイッチで学習されたエンドポイントの数に基づいています。GUI システム ダッシュボードに

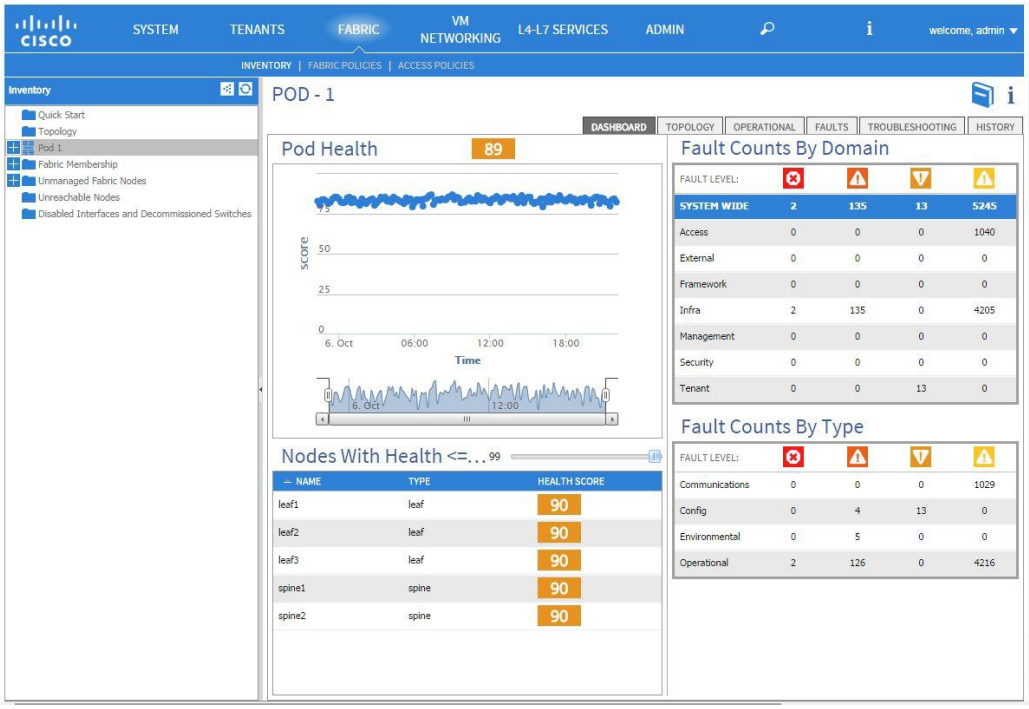
も、ドメイン タイプごとのシステム全体のエラー数とともに、APIC クラスタのノード単位の管理状態、稼働状態、ヘルス状態が表示されます。

図 *90*：システムのヘルス スコア



ポッドのヘルス スコアは、リーフ/スパイン スイッチのヘルス スコアと、リーフ スイッチで学習されたエンドポイントの数に基づいています。GUI ファブリック ポッド ダッシュボード画面でも、ポッド全体のエラーがドメインやタイプごとに表示されます。

図 *91*：ポッドのヘルス スコア



304812

システムとポッドのヘルス スコアは、同じ方法で計算されます。計算は、リーフ ヘルス スコアの加重平均を、リーフ スイッチの学習済みエンドポイントの総数で除算し、その結果にスパイン数とそのヘルス スコアから算出したスパイン係数を乗算することによって行います。

次の式に、この計算方法を示します。

図 *92*：システムおよびポッドのヘルス スコアの計算

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

304814

次の凡例では、方程式コンポーネントについて定義します。

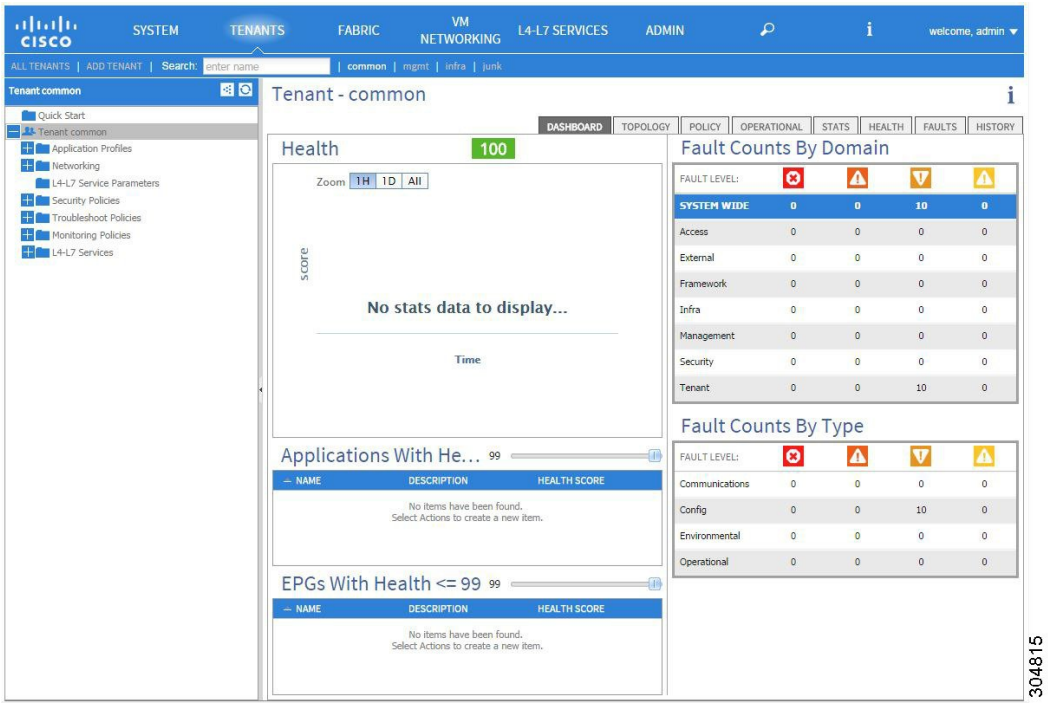
- $Health_{Leaf_i}$ はリーフ スイッチのヘルス スコアです
- $Weight_{Leaf_i}$ はリーフ スイッチ上のエンドポイントの数です。
- N_{Leaf} はファブリック内のリーフ スイッチの数です。
- $Health_{Spine_i}$ はスパイン スイッチのヘルス スコアです

- N_{Spine} はファブリック内のスパイン スイッチの数です。

テナントのヘルス スコア

テナントのヘルス スコアは、その時に使用しているインフラストラクチャ間でのテナント全体の論理オブジェクトのヘルス スコアを集約します。GUI のテナント ダッシュボード画面でも、テナント全体のエラーの数がドメインやタイプごとに表示されます。

図 *93*：テナントのヘルス スコア

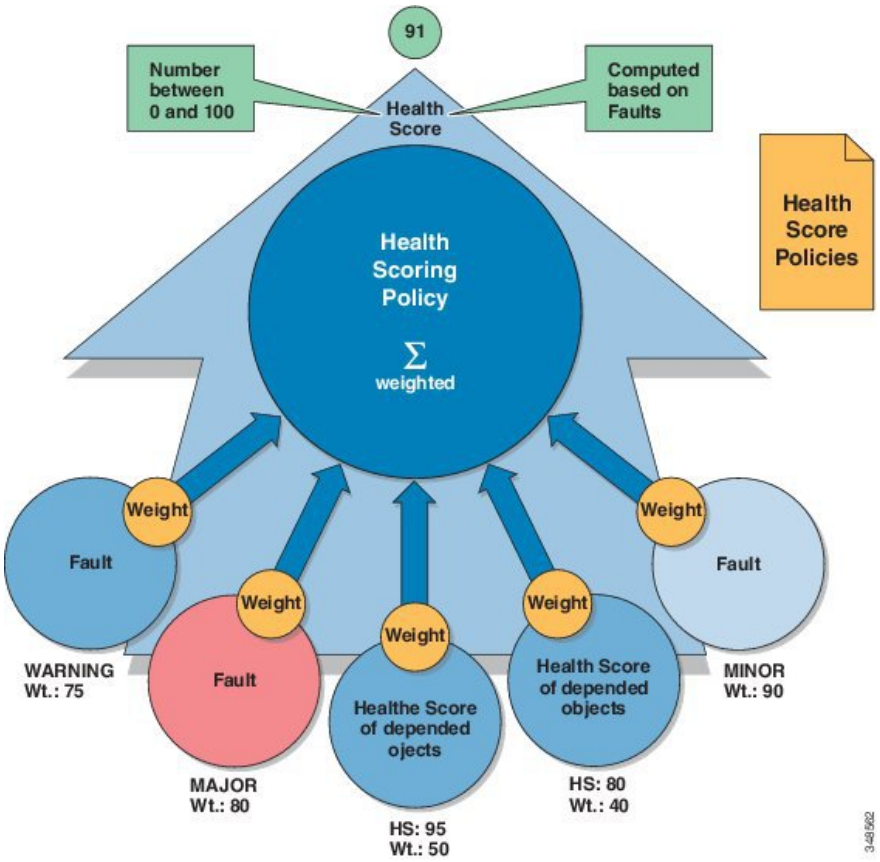


たとえば、EPG が 2 つのリーフ スイッチのポートを使用しているとします。各リーフ スイッチには、配置済みの EPG コンポーネントが含まれています。学習したエンドポイントの数は重み係数となります。学習したエンドポイントの数は、それぞれのポートで異なっている可能性があります。したがって、EPG のヘルス スコアは、各 EPG コンポーネントのヘルス スコアに、そのリーフで学習されたエンドポイントの数をかけ、EPG が使用しているリーフ スイッチ全体の学習済みエンドポイントの総数で割ることにより、算出されます。

MO のヘルス スコア

各管理対象オブジェクト（MO）は、ヘルス スコアのカテゴリーに属しています。デフォルトでは、MO のヘルス スコアのカテゴリは MO のクラス名と同じです。

図 *94*：*MO* のヘルス スコア



各ヘルス スコア カテゴリには影響レベルが割り当てられます。ヘルス スコアの 5 つの影響レベルは、Maximum、High、Medium、Low および None です。たとえば、ファブリック ポートのデフォルトの影響レベルは Maximum で、リーフ ポートのデフォルトの影響レベルは High です。子 MO の特定のカテゴリは、ヘルス スコアの影響レベル None を割り当てることで、親 MO のヘルス スコアの計算から除外できます。これらのオブジェクト間の影響レベルは、ユーザが設定できます。ただし、デフォルトの影響レベルが None の場合は、管理者はこれを上書きできません。

次の係数は、さまざまな影響レベルです。

Maximum：100% High：80% Medium：50% Low：20% None：0%

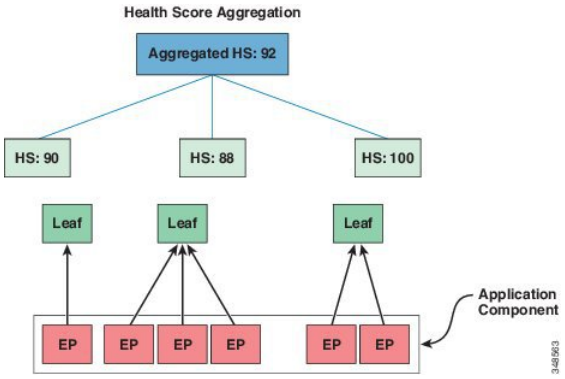
カテゴリ ヘルス スコアは、Lp ノルム式を使用して計算されます。ヘルス スコア ペナルティは、100 - ヘルス スコアと等しくなります。ヘルス スコア ペナルティは、所定のカテゴリに属し、ヘルス スコアが計算される MO の子または直接親族である MO のセットの全体的なヘルス スコア ペナルティを表します。

MO クラスのヘルス スコアのカテゴリは、ポリシーを使用して変更できます。たとえば、リーフ ポートのデフォルトのヘルス スコア カテゴリは `eqpt:LeafP` で、ファブリック ポートのデフォルトのヘルス スコア カテゴリは `eqpt:FabP` です。ただし、リーフ ポートとファブリック ポートの両方を含むポリシーは、ポートと呼ばれる同じカテゴリの一部になるように作成できます。

ヘルス スコアの集約と影響

アプリケーション コンポーネントのヘルス スコアは、次の図に示すように複数のリーフ スイッチに分散できます。

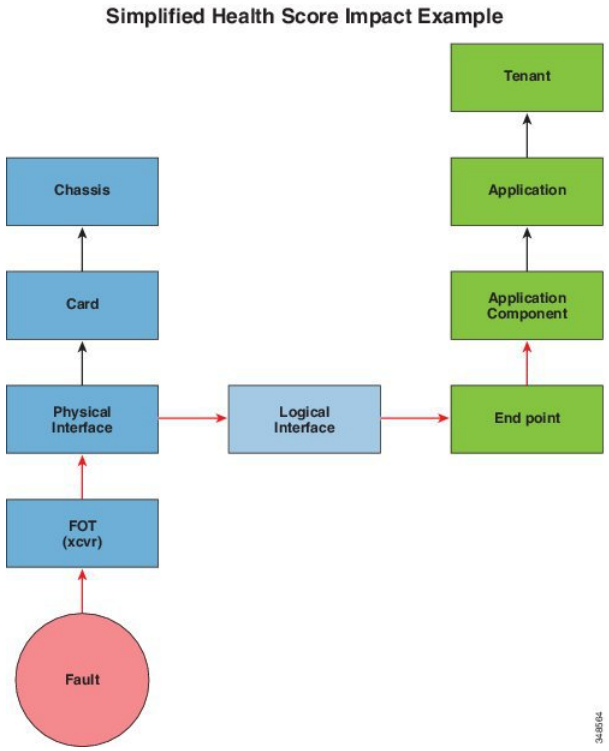
図 *95*：ヘルス スコアの集約



集約されたヘルス スコアは、APIC で計算されます。

次の図では、ハードウェアの障害が、アプリケーション コンポーネントのヘルス スコアに影響します。

図 *96*：簡略化したヘルス スコアの影響の例



アトミック カウンタ

アトミック カウンタは、ファブリック内のドロップとルーティングミスを検出し、迅速なデバッグとアプリケーションの接続性問題の分離が可能になります。アトミック カウンタには、アクティブなファブリック ネットワーク タイムプロトコル（NTP）ポリシーが必要です。アトミック カウンタは、IPv6 または IPv4 いずれかの送信元アドレスおよび宛先アドレスに対して動作しますが、異なるアドレス ファミリ間で動作することはできません。

たとえば、管理者はすべてのリーフ スイッチでアトミック カウンタを有効にして、エンドポイント 1 からエンドポイント 2 のパケットをトレースすることができます。送信元と宛先のリーフ以外のリーフでゼロ以外のカウンタがある場合は、管理者はそれらのリーフにドリル ダウンできます。

従来の設定では、baremetal NIC から特定の IP アドレス（エンドポイント）または任意の IP アドレスへのトラフィックの量をモニタすることはほぼ不可能です。アトミック カウンタでは、データ パスに干渉することなく、管理者が baremetal エンドポイントから受信されたパケットの数を数えることができます。さらに、アトミック カウンタはエンドポイントまたはアプリケーション グループで送受信されるプロトコルごとのトラフィックをモニタリングできます。

リーフ間（TEP 間）アトミック カウンタは次を提供できます。

- ドロップ、承認および超過パケットのカウント
- 最後の 30 秒などの短期間のデータ収集、5 分、15 分、またはそれ以上の長期間のデータ収集
- スパイン トラフィックごとの詳細は、TEP、リーフ、または VPC の数が 64 未満の場合のみ利用できます。
- 継続的なモニタリング

（注） リーフ間（TEP間）アトミック カウンタは累積であり、クリアできません。ただし、30 秒のアトミック カウンタは 30 秒間隔でリセットされるため、断続的な問題や、再発する問題の分離に使用できます。

テナントのアトミック カウンタは次を提供できます:

- ドロップ、承認および超過パケットを含む、ファブリック全体のトラフィックのアプリケーション固有カウンタ
- モードは次を含みます。
  - エンドポイント間 MAC アドレスまたはエンドポイント間 IP アドレス注：単一のターゲット エンドポイントに、それに関連付けられた複数の IP アドレスがある場合があります。
  - EPG ツー EPG
  - EPG ツー エンドポイント
  - EPG ツー *（任意）
  - エンドポイント ツー外部 IP アドレス

（注） アトミック カウンタの使用は、エンドポイントが異なるテナントまたは同じテナント内の異なるコンテキスト（VRF）にある場合はサポートされません。アトミック カウンタは IPv6 の送信元と宛先で動作しますが、IPv4 アドレスと IPv6 アドレスを混在させて送信元 IP アドレスと宛先 IP アドレスを設定することはできません。

マルチノード SPAN

APIC トラフィック モニタリング ポリシーは、適切な場所でポリシーを展開して、各アプリケーション グループのすべてのメンバーとそのメンバーが接続される場所を追跡することができます。メンバーが移動すると、APICは新しいリーフにポリシーを自動的にプッシュします。たとえば、エンドポイントが新しいリーフに VMotion すると、スパン設定が自動的に調整されます。

ERSPAN ヘッダーの詳細については、次の URL にある IETF インターネット ドラフトを参照してください。https://tools.ietf.org/html/draft-foschiano-erspan-00。

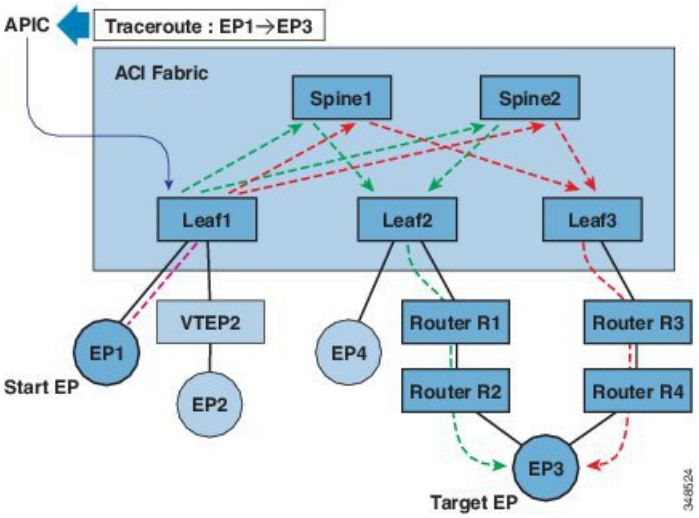
ACI ファブリックは、リモートの SPAN（ERSPAN）について以下の 2 つの拡張形式をサポートします。

- アクセスまたはテナント SPAN：フィルタとしての VLAN の使用の有無にかかわらず、リーフ スイッチ前面パネル ポートのために実行。リーフ スイッチの Broadcom Trident 2 ASIC は、ERSPAN タイプ 1 形式とはやや異なるバージョンをサポートしています。これは、GRE ヘッダーが 4 バイトのみでシーケンス フィールドがないという点で、前述のドキュメントで定義される ERSPAN タイプ 1 形式とは異なります。GRE ヘッダーは、常に 0x000088be で符号化されます。0X88be は ERSPAN タイプ 2 を表していますが、フィールドの残り 2 バイトは、これが 4 バイトの GRE ヘッダーを持つ ERSPAN タイプ 1 のパケットであることを示しています。
- ファブリック SPAN：Northstar ASIC によってリーフ スイッチ内で実行、またはスパイン スイッチ内で Alpine ASIC によって実行。これらの ASIC は ERSPAN タイプ 2 および 3 形式をサポートしていますが、上記のベースライン ドキュメントに記載されているように、現在 ACI ファブリックはファブリック SPAN について ERSPAN タイプ 2 しかサポートしていません。

ARP、ICMP ping および traceroute

デフォルトのゲートウェイ IP アドレスの ARP は入力リーフ スイッチでトラップされます。入力リーフ スイッチは ARP 要求を宛先にユニキャストし、宛先は ARP 応答を送信します。

図 *97*：*APIC* エンドポイント/エンドポイント *traceroute*



テナントのエンドポイントから開始される traceroute は、中間ホップとしてデフォルト ゲートウェイが入力リーフ スイッチに表示されることを示します。

tracerouteモードには、エンドポイント/エンドポイント、リーフ/リーフ（TEP/TEP）があります。tracerouteは、ファブリック全体のすべてのパスおよび外部エンドポイントの出口のポイントを検出し、パスがブロックされたかどうかを検出するのに役立ちます。

tracerouteはIPv6の送信元アドレスおよび宛先アドレスとともに動作しますが、IPv4アドレスとIPv6アドレスを混在させて送信元アドレスと宛先アドレスを設定することはできません。送信元（`RsTrEpIpSrc`）と宛先（`RsTrEpIpDst`）の関係は、タイプ`fvIp`の送信元と宛先をサポートします。時には複数のIPアドレスが同じエンドポイントから学習されることもあります。管理者は、必要な送信元アドレスと宛先アドレスを選択します。

付 録 A

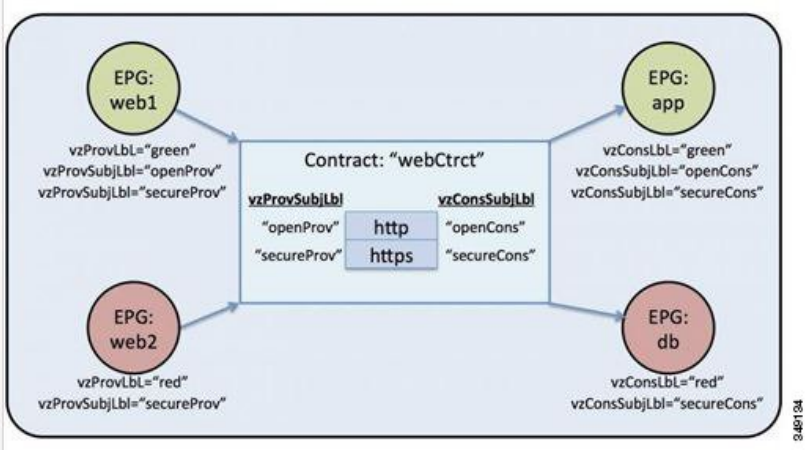
テナント ポリシーの例

この章の内容は、次のとおりです。

テナント ポリシー例の概要

この付録のテナント ポリシー例の説明では、XML 用語（http://en.wikipedia.org/wiki/XML#Key_terminology）を使用します。この例では、基本的な APIC ポリシー モデル構造が XML コードにどのようにレンダリングされるかを示します。次の図は、テナント ポリシー例の概要について説明します。

図 *98*：テナント *Solar* に含まれる *EPG* とコントラクト



この図では、webCtrct および EPG ラベルと呼ばれるコントラクトに従って、グリーンラベルの EPG:web1 が http と https の両方を使用してグリーンラベルの EPG:app と通信でき、レッドラベルの EPG:web2 は https のみを使用してレッドラベルの EPG:db と通信できます。

テナント ポリシー例の XML コード

```
<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>
```

テナント ポリシー例の説明

この項には、テナント ポリシー例の詳しい説明が含まれます。

ポリシー ユニバース

ポリシー ユニバースには、各テナントのポリシーが定義されているすべてのテナント管理対象オブジェクトが含まれます。

```
<polUni>
```

最初の行のこの開始タグ <polUni> は、ポリシー ユニバース要素の開始を示します。このタグは、ポリシーの最後にある </polUni> と一致します。間にあるのはすべて、ポリシー定義です。

テナント ポリシーの例

タグ <fvTenant> は、テナント要素の開始を識別します。

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

このテナントのポリシーはすべてこの要素で定義されます。この例でのテナントの名前はsolarです。テナントの名前はシステム内で一意である必要があります。テナントが含む主要な要素は、フィルタ、コントラクト、外部ネットワーク、ブリッジドメイン、およびEPGを含むアプリケーション プロファイルです。

フィルタ

フィルタ要素は、タグ <vzFilter> から始まり、タグ <vzEntry> で示される要素が含まれます。

次に、「Http」と「Https」フィルタを定義する例を示します。フィルタの最初の属性が名前で、name 属性の値はテナントに一意の文字列です。これらの名前は異なるテナントで再利用できます。これらのフィルタは、この例の後でコントラクト内のサブジェクト要素で使用されます。

```
<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>
```

この例では、2 つのフィルタ、Http および Https を定義します。フィルタの最初の属性はその名前で、name 属性の値はテナントに一意の文字列です。つまり、これらの名前は異なるテナントで再利用できます。これらのフィルタは、この例の後のほうでコントラクト内のサブジェクト要素で使用されます。

各フィルタには、各エントリがレイヤ 4 TCP または UDP ポート番号のセットを説明する 1 つ以上のエントリを含めることができます。<vzEntry> 要素の考えられる属性の一部は次のとおりです。

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

この例では、各エントリの name 属性が指定されます。名前はフィルタ内で一意でなければならない ASCII 文字列ですが、他のフィルタで再利用できます。なぜなら、この例では、後で特定のエントリを参照せず、「e1」という単純な名前が与えられるためです。

EtherType 属性 etherT が次です。ipv4 の値が割り当てられ、このフィルタが IPv4 パケット用であることを指定します。この属性には考えられる他の多くの値があります。一般的なものは、ARP、RARP、IPv6 などです。デフォルトは unspecified なので、値を割り当てることが重要です。

EtherType 属性の後は、prot 属性です。この属性は tcp に設定され、このフィルタが TCP トラフィック用であることを示します。代替プロトコル属性には、udp、icmp、および unspecified（デフォルト）があります。

プロトコルの後、宛先の TCP ポート番号は 80 ～ 80 の範囲（正確には TCP port 80）になるように dFromPort および dToPort 属性で割り当てられます。送信元と宛先が異なっている場合、それらはポート番号の範囲を指定します。

この例では、これらの宛先ポート番号は属性 dFromPort および dToPort で指定されます。ただし、コントラクトで使用されている場合は、TCP クライアントからサーバへの宛先ポートのためにリターン トラフィックの送信元ポートとして使用する必要があります。詳細については、この例の後に出てくる属性 revFltPorts を参照してください。

2 番目のフィルタは基本的に同じ機能がありますが、ポート 443 に対するものです。

フィルタは、ターゲットの識別名 tDn によってコントラクト内のサブジェクトによって参照されます。tDn 名は次のように構成されます。

```
uni/tn-<tenant name>/flt-<filter name>
```

たとえば、上記の最初のフィルタの tDn は uni/tn-coke/flt-Http です。2 番目のフィルタには名前 uni/tn-coke/flt-Https があります。いずれの場合も、solar がテナント名から取得されます。

コントラクト

コントラクト要素は、vzBrCP でタグ付けされ、name 属性があります。

```
<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>
```

コントラクトは EPG 間のポリシー要素です。コントラクトには、コントラクトを作成して消費する EPG 間で適用されるすべてのフィルタが含まれます。コントラクト要素は、vzBrCP でタグ付けされ、name 属性があります。コントラクト要素で使用できるその他の属性については、オブジェクト モデルの参照資料を参照してください。この例では、webCtrct という名前のコントラクトが 1 つあります。

コントラクトには、各サブジェクトが一連のフィルタを含む複数のサブジェクト要素が含まれます。この例では、2 つのサブジェクト、http と https があります。

コントラクトは、それを提供または消費する EPG によって後で参照されます。EPG は、以下の方法で名前によってそのコントラクトを参照します。

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

tenant-name はテナントの名前で、この例では「solar」となります。contract-name はコントラクトの名前です。この例では、コントラクトの tDn 名は uni/tn-solar/brc-webCtrct です。

サブジェクト

サブジェクト要素は、タグ vzSubj から始まり、3 つの属性、name、revFltPorts および matchT を持ちます。name は、単にサブジェクトの ASCII 名です。

revFltPorts は、このサブジェクトのフィルタ内のレイヤ 4 送信元および宛先ポートをフィルタの説明に示すとおりに転送方向（つまり、コンシューマからプロデューサ EPG の方向）に使用する必要があり、逆方向には逆の方法で使用する必要があることを示すフラグです。この例では、「http」サブジェクトには、TCP 宛先ポート 80 を定義し、送信元ポートを指定していない「Http」フィルタが含まれます。revFltPorts フラグが true に設定されているため、ポリシーは、TCP **宛先ポート 80** およびコンシューマからプロデューサへのトラフィック用の送信元ポートであり、また、TCP 宛先ポートおよびプロデューサからコンシューマへのトラフィック用の**送信元ポート 80** になります。コンシューマがプロデューサへの TCP 接続を開始することを前提としています（コンシューマがクライアントで、プロデューサがサーバ）。

指定しない場合、revFltPrts 属性のデフォルト値は false です。

ラベル

一致タイプ属性、provmatchT（プロバイダー一致の場合）および consmatchT（コンシューマ一致の場合）は、サブジェクトが所定のコンシューマとプロデューサのペアに対し適用されるかを判断するためにサブジェクト ラベルがどのように比較されるかを決定します。次の一致タイプの値が使用可能です。

- すべて（All）
- AtLeastOne（デフォルト）
- なし（None）
- ExactlyOne

サブジェクトがプロデューサとコンシューマ EPG 間のトラフィックに適用されるかどうかを決定する場合、一致属性は、これらの EPG で定義されている（または定義されていない）サブジェクト ラベルがサブジェクト内のラベルとどのように比較されるべきかを決定します。一致属性の値が All に設定されると、それはプロバイダー サブジェクト ラベル vzProvSubjLbl がサブジェクト内で定義されたすべての vzProvSubjLbl ラベルと一致するプロバイダーにのみ適用されます。2 つのラベルが定義されている場合、両方ともプロバイダー内にある必要があります。プロバイダー EPG に 10 個のラベルがある場合、サブジェクト内のすべてのプロバイダー ラベルが存在する限り、一致が確認されます。同様の基準が vzConsSubjLbl を使用するコンシューマに使用されます。matchT 属性値が AtLeastOne の場合、ラベルの 1 つだけが一致する必要があります。matchT 属性が None の場合、サブジェクト内のプロバイダー ラベルがプロバイダー EPG のプロバイダー ラベルと一致しない場合にのみ一致が発生します。コンシューマの場合も同様です。

プロデューサまたはコンシューマ EPG にサブジェクト ラベルがなく、サブジェクトがラベルを持たない場合、一致は All、AtLeastOne、および None の場合に発生します（ラベルを使用しない場合は、サブジェクトが使用され matchT 属性は問題になりません）。

この例には示されていないサブジェクトのオプション属性は prio で、フィルタに一致するトラフィックのプライオリティが指定されます。考えられる値は、gold、silver、bronze、または unspecified（デフォルト）です。

この例では、サブジェクト要素にフィルタ要素、サブジェクト ラベル要素およびグラフ要素への参照が含まれます。<vzRsSubjFiltAtt tDn="uni/tn-coke/flt-Http"/> は事前に定義されたフィルタへの参照です。この要素は vzRsSubjFiltAtt タグによって識別されます。

<vzRsSubjGraphAtt graphName="G1" termNodeName="TProv"/> は端末接続を定義します。

<vzProvSubjLbl name="openProv"/> は「openProv」という名前のプロバイダー ラベルを定義します。ラベルは、どのサブジェクトがどの EPG に適用されるかを認定したりフィルタリングするために使用されます。この特定のラベルがプロバイダー ラベルであり、対応するコンシューマ ラベルがタグ vzConsSubjLbl で識別されます。これらのラベルは、現在のコントラクトに関連付けられたプロバイダーまたはコンシューマ EPG の対応するラベルと一致します。前述の matchT 基準に従って一致が発生する場合は、特定のサブジェクトが EPG に適用されます。一致が発生しない場合、サブジェクトは無視されます。

複数のプロバイダーおよびコンシューマのサブジェクト ラベルをサブジェクトに追加して、より複雑な一致基準を可能にすることができます。この例では、各サブジェクトに各タイプのラベルが 1 個だけあります。ただし、最初のサブジェクトのラベルは 2 番目のサブジェクトのラベルとは異なり、これら 2 つのサブジェクトを対応する EPG のラベルに応じて、それぞれ処理できます。サブジェクト要素内の要素の順序は重要ではありません。

コンテキスト

コンテキストは fvCtx タグによって識別され、name 属性が含まれます。

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

テナントには、複数のコンテキストを含めることができます。この例では、テナントは「solartx1」という名前のコンテキストを 1 個使用します。名前は、テナント内で一意である必要があります。

コンテキストは、レイヤ 3 のアドレス ドメインを定義します。レイヤ 3 ドメイン内のすべてのエンドポイントが一意の IPv4 または IPv6 アドレスを持っている必要があります。なぜなら、ポリシーで許可されている場合にこれらのデバイス間でパケットを直接転送できるためです。コンテキストは、ネットワーキング ワールドの仮想ルーティングおよび転送（VRF）インスタンスに相当します。

コンテキストが一意の IP アドレス空間を定義する一方で、対応するサブネットがブリッジ ドメイン内で定義されます。各ブリッジ ドメインはその後コンテキストに関連付けられます。

ブリッジ ドメイン

ブリッジ ドメインの要素は fvBD タグで識別され、name 属性があります。

```
<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>
```

ブリッジ ドメインの要素内では、サブネットが定義され、対応するレイヤ 3 コンテキストへの参照が行われます。各ブリッジ ドメインは、コンテキストにリンクされ、少なくとも 1 個のサブネットを設定する必要があります。

この例では、「solarBD1」という名前のブリッジ ドメインを 1 個使用します。この例では、「solarctx1」というコンテキストが、`fvRsCtx` とタグ付けされた要素を使用して参照され、`tnFvCtxName` 属性に値「solarctx1」が与えられます。 この名前は、上記で定義したコンテキストから取得されます。

サブネットがブリッジドメイン内に含まれ、ブリッジドメインは複数のサブネットを含むことができます。この例では、2 個のサブネットを定義します。ブリッジ ドメイン内で使用されるすべてのアドレスは、サブネットで定義されるアドレス範囲のいずれかに分類される必要があります。ただし、サブネットは、使用されることがないであろう多数のアドレスを含む大規模なサブネットであるスーパーネットにすることもできます。現在および将来のアドレスすべてに対応する大規模なサブネットを 1 個指定すると、ブリッジ ドメインの仕様を簡素化できます。ただし、異なるサブネットがブリッジ ドメイン内で重複したり、または同じコンテキストに関連付けられている他のブリッジ ドメインで定義されたサブネットと重複してはなりません。サブネットは、他のコンテキストに関連付けられている他のサブネットと重複できます。

前述のサブネットは、11.22.22.xx/24 と 11.22.23.xx/24 です。ただし、24 だけが使用されることをマスクが示していても、アドレスの完全な 32 ビットが与えられます。それは、この IP 属性がそのサブネットに対するルータの完全な IP アドレスの役割も示しているためです。最初のケースでは、ルータの IP アドレス（デフォルト ゲートウェイ）は 11.22.22.20 で、2 番目のサブネットでは、11.22.23.211 です。

エントリ 11.22.22.20/24 は以下に相当しますが、コンパクト形式です。

- サブネット：11.22.22.00
- サブネット マスク：255.255.255.0
- デフォルト ゲートウェイ：11.22.22.20

アプリケーション プロファイル

アプリケーション プロファイルの開始はタグ `fvAp` で示され、name 属性があります。

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

この例では、アプリケーション ネットワーク プロファイルが 1 つあり、「sap」という名前になっています。

アプリケーション プロファイルは、EPG を保持するコンテナです。EPG は同じアプリケーション プロファイル内の他の EPG および他のアプリケーション プロファイル内の EPG と通信できます。

アプリケーションプロファイルは、互いに論理的に関連する複数のEPGを保持するために使用される簡易で便利なコンテナです。それらは、「sap」などの提供するアプリケーション、「インフラストラクチャ」などの提供する機能、「DMZ」などのデータセンターの構造内のそれらが存在する場所、または管理者が使用することを選択した組織化の原則によって組織化できます。

アプリケーション プロファイルに含まれるプライマリ オブジェクトは、エンドポイント グループ（EPG）です。この例では、「sap」アプリケーション プロファイルには 4 個の EPG、web1、web2、app および db が含まれます。

エンドポイントおよびエンドポイント グループ（EPG）

EPG は、タグ `fvAEPg` で始まり、name 属性があります。

```
<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>
```

EPGは、ポリシー モデルの最も重要な基本オブジェクトです。これは、ポリシーの観点から同じ方法で処理されるエンドポイントの集合を表します。 それらのエンドポイントは個別に設定および管理されるのではなく、EPG 内に配置され、集合またはグループとして管理されます。

EPG オブジェクトは、どのようなポリシーが適用されるのか、また他のどの EPG がこの EPG と通信できるかを規定するラベルが定義されている場所です。また、EPG 内のエンドポイントが関連付けられるブリッジ ドメイン、およびそれらが関連付けられる Virtual Machine Manager（VMM）のドメインへの参照が含まれています。VMM により、2 台の VM サーバ間の仮想マシンのモビリティがアプリケーションのダウンタイムなしで即座に可能になります。

この例の最初のEPGは「web1」という名前です。EPG内の `fvRsBd` 要素は、関連付けられるブリッジ ドメインを定義します。ブリッジ ドメインは `tnFxBDName` 属性の値によって識別されます。この EPG は、前述の「ブリッジ ドメイン」の項で名前を付けられた「solarBD1」というブリッジ ドメインに関連付けられます。ブリッジドメインへのバインディングは、デフォルトゲートウェイアドレスがこのEPGのエンドポイントのためにどうあるべきかをシステムが理解するために使用されます。エンドポイントがすべて同じサブネットにあることや、ブリッジングを介してのみ通信できることを意味しているのではありません。エンドポイントのパケットがブリッジングまたはルーティングされるかどうかは、送信元エンドポイントがパケットをデフォルト ゲートウェイまたは要求される最後の宛先に送信するかどうかで決定されます。デフォルト ゲートウェイにパケットを送信すると、パケットはルーティングされます。

このEPGで使用されるVMMドメインは `fvRsDomAtt` タグによって識別されます。この要素は、他の場所で定義された VMM ドメイン オブジェクトを参照します。VMM ドメイン オブジェクトは、その `tDn` name 属性によって識別されます。この例では、「uni/vmmp-VMware/dom-mininet」と呼ばれる VMM ドメイン 1 個のみを示します。

「web1」EPG の次の要素は、この EPG が提供するコントラクトを定義し、fvRsProv タグによって識別されます。「web1」が複数のコントラクトを提供すると、fvRsProv 要素が複数あります。同様に、それが 1 つ以上のコントラクトを消費すると、fvRsCons 要素があります。

fvRsProv 要素には、提供されているコントラクトの名前である必須属性があります。「web1」は、tDn="uni/tn-coke/brc-webCtrct" と呼ばれる以前に定義されたコントラクト「webCtrct」を提供しています。

次の属性は、matchT 属性です。その属性には、それがサブジェクト ラベルのコントラクト内にあったので、プロバイダーまたはコンシューマのラベルと一致するための同じセマンティックがあります（All、AtLeastOne または None の値を取ることができます）。この条件は、対応するコンシューマ ラベルと比較されるときにプロバイダーのラベルに適用されます。ラベルの一致は、コンシューマとプロバイダーがコントラクトで許可された場合に通信できることを意味します。つまり、コントラクトが通信を許可する必要があり、コンシューマとプロバイダーのラベルがプロバイダーで指定された一致基準を使用して一致する必要があります。

コンシューマには、対応する一致基準がありません。使用される一致タイプはプロバイダーによって常に定められます。

プロバイダー要素 fvRsProv の中で、管理者は使用するラベルを指定する必要があります。2 種類のラベル、プロバイダー ラベルとプロバイダー サブジェクト ラベルがあります。プロバイダー ラベル vzProvLbl は、前述の matchT 基準を使用する他の EPG 内のコンシューマ ラベルに一致させるために使用されます。プロバイダー サブジェクト ラベル vzProvSubjLbl は、コントラクトで指定されるサブジェクト ラベルに一致させるために使用されます。ラベルの唯一の属性は name 属性です。

「web1」EPG では、2 つのプロバイダー サブジェクト ラベル openProv および secureProv が「webCtrct」コントラクトのサブジェクト「http」および「https」と一致するように指定されます。1 つのプロバイダー ラベル「green」が、「App」EPG 内の同じラベルと一致する All の一致基準で指定されます。

この例の次の EPG「web2」は「web1」と似ていますが、vzProvSubjLbl が 1 つだけあり、ラベル自体は異なっています。

3 番目の EPG は「app」と呼ばれるもので、次のように定義されます。

```
<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>
```

最初の部分は「web1」EPG とほぼ同じです。主な違いは、この EPG は「webCtrct」のコンシューマで、対応するコンシューマ ラベルとコンシューマ サブジェクト ラベルがあることです。構文はほぼ同じですが、タグで「Prov」が「Cons」に置き換えられます。プロバイダーをコンシューマ ラベルに一致させるための一致タイプがプロバイダーで指定されるため、FvRsCons 要素に一致属性はありません。

最後の EPG では、純粋なコンシューマであるという点において「db」は「app」EPG と非常によく似ています。

この例では、EPG は単一コントラクトのコンシューマまたはプロデューサであり、EPG が即座に複数のコントラクトのプロデューサおよび複数のコントラクトのコンシューマになることが一般的です。

最後に

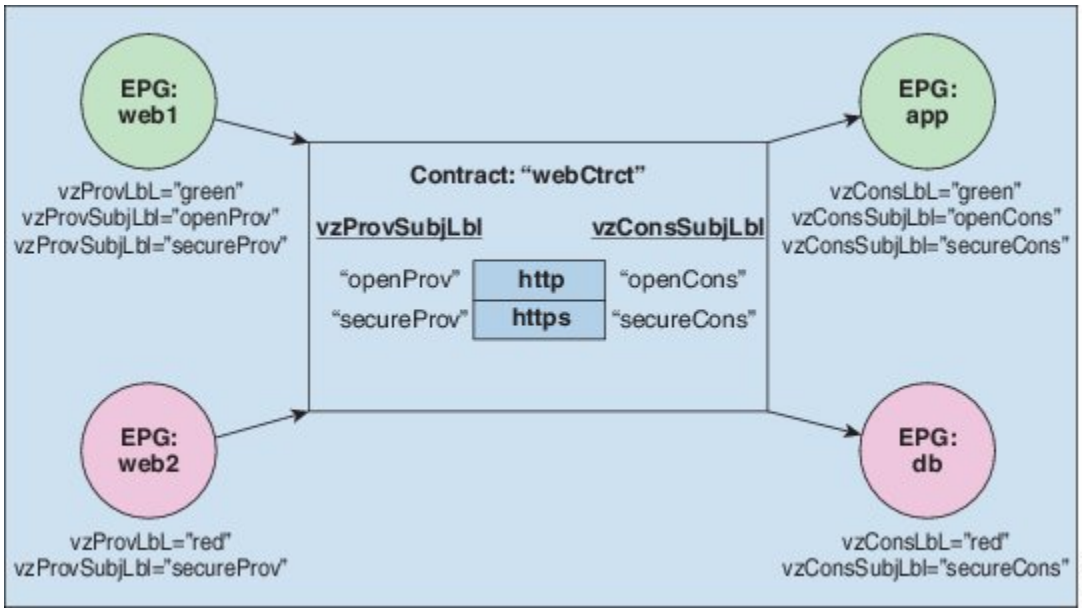
```
        </fvAp>
    </fvTenant>
</polUni>
```

最後の数行でポリシーが完了します。

この例のテナント ポリシーが行うこと

次の図は、コントラクトがエンドポイント グループ（EPG）の通信をどのように管理するかを示します。

図 *99*：*EPG/EPG* 通信を決定するラベルとコントラクト



4 つの EPG には、EPG:web1、EPG:web2、EPG:app および EPG:db という名前が付いています。EPG:web1 および EPG:web2 は webCtrct と呼ばれるコントラクトを提供します。EPG:app および EPG:db は、同じコントラクトを消費します。

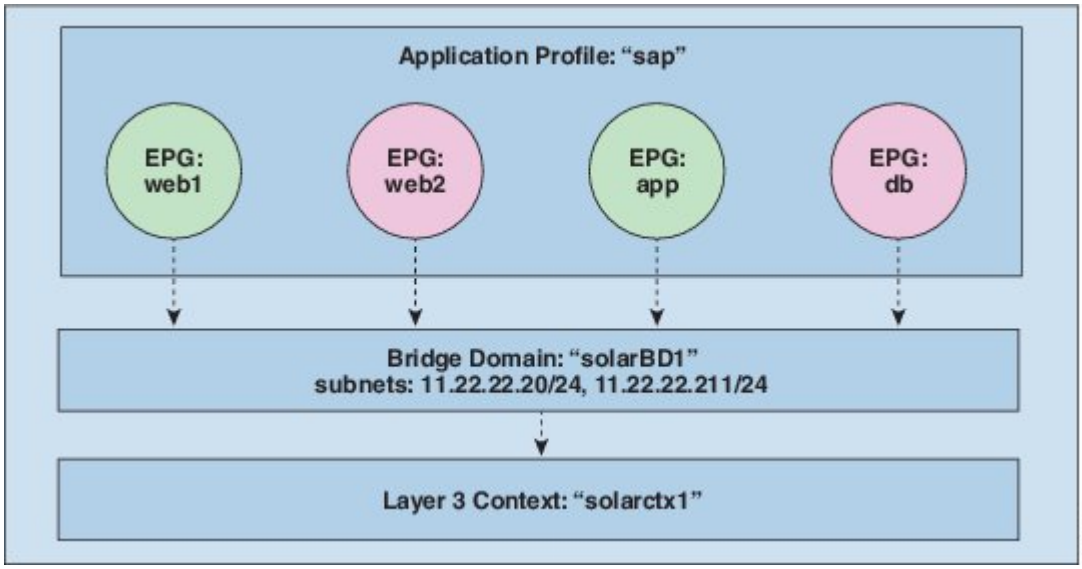
EPG:web1 は EPG:app のみと通信でき、EPG:web2 は EPG:db のみと通信できます。このインタラクションは、プロバイダーおよびコンシューマのラベル「green」と「red」によって制御されます。

EPG:web1 が EPG:app と通信するとき、webCtrct コントラクトが使用されます。EPG:app は、EPG:web1 が提供するコントラクトを消費するので、EPG:web1 への接続を開始できます。

EPG:web1 と EPG:app が通信を行うために使用できるサブジェクトは両方とも http および https です。なぜなら、EPG:web1 にはプロバイダー サブジェクト ラベル「openProv」があり、http サブジェクトにもそれがあるためです。EPG:web1 には、プロバイダー サブジェクト ラベル「secureProv」があり、サブジェクト https も同様です。同様に、EPG:app にはサブジェクト ラベル「openCons」および「secureCons」があり、サブジェクト http および https にもあります。

EPG:web2 が EPG:db と通信するとき、それらは https サブジェクトのみを使用できます。https サブジェクトのみがプロバイダーおよびコンシューマのサブジェクト ラベルを持っているためです。EPG:db は EPG:web2 への TCP 接続を開始できます。なぜなら、EPG:db が EPG:web2 により提供されるコントラクトを消費するからです。

図 *100*：ブリッジ ドメイン、サブネット、およびレイヤ *3* コンテキスト



この例のポリシーは、EPG、アプリケーション プロファイル、ブリッジ ドメインおよびレイヤ 3 コンテキスト間の関係を次のように指定します。EPG の EPG:web1、EPG:web2、EPG:app および EPG:db はすべて、「sap」と呼ばれるアプリケーション プロファイルのメンバーです。

これらの EPG はブリッジ ドメイン「solarBD1」にもリンクされています。solarBD1 には、2 つのサブネット 11.22.22.XX/24 と 11.22.23.XX/24 があります。4 つの EPG 内のエンドポイントは、これら 2 つのサブネット範囲内にある必要があります。これら 2 つのサブネット内のデフォルト ゲートウェイの IP アドレスは 11.22.22.20 と 11.22.23.211 です。solarBD1 ブリッジ ドメインは「solarctx1」レイヤ 3 コンテキストにリンクされます。

これらのポリシーの詳細はすべて、「solar」というテナントに含まれています。

付 録 B

ラベルの一致

この章の内容は、次のとおりです。

ラベルの一致

ラベルの一致は、通信可能なコンシューマ EPG およびプロバイダー EPG を判定するために使用します。コンシューマとプロバイダーが通信できるかどうかは、コントラクトの所定のプロデューサまたはコンシューマのコントラクト サブジェクトによって決まります。

一致タイプ アルゴリズムは、matchT 属性によって決定されます。matchT 属性は、次のいずれかの値を取ることができます。

- すべて（All）
- AtLeastOne（デフォルト）
- なし（None）
- AtmostOne

EPG とコントラクト サブジェクトのラベルが両方とも存在する場合、ラベルの一致の実行順序は、最初に EPG、次にコントラクト サブジェクトとなります。

プロバイダー ラベル vzProvLbl とコンシューマ ラベル vzConsLbl の一致を確認する場合、matchT はプロバイダー EPG によって決定されます。

サブジェクトを含むEPG内でプロバイダーまたはコンシューマのサブジェクト ラベル vzProvSubjLbl および vzConsSubjLbl の一致を確認する場合、matchT はサブジェクトによって決定されます。

matchT ロジックは、EPG ラベルの場合もコントラクト サブジェクト ラベルの場合も同じです。次の表は、すべての EPG/コントラクト サブジェクト プロバイダーとコンシューマの一致タイプおよびその結果の簡単な例を示します。この表で、[] エントリはラベルがないことを示します。

ラベルの一致

matchT	vzProvLbl vzProvSubLbl	vzConsLbl vzConsSubLbl	結果
すべて（All）	LabelX、LabelY	LabelX、LabelY	一致
すべて（All）	LabelX、LabelY	LabelX、LabelZ	一致しない
すべて（All）	LabelX、LabelY	LabelX	一致しない
すべて（All）	LabelX	LabelX、LabelY	一致
すべて（All）	[ ]	LabelX	一致しない
すべて（All）	LabelX	[ ]	一致しない
すべて（All）	[ ]	[ ]	一致しない
AtLeastOne	LabelX、LabelY	LabelX	一致
AtLeastOne	LabelX、LabelY	LabelZ	一致しない
AtLeastOne	LabelX	[ ]	一致しない
AtLeastOne	[ ]	LabelX	一致しない
AtLeastOne	[ ]	[ ]	一致
なし（None）	LabelX	LabelY	一致
なし（None）	LabelX	LabelX	一致しない
なし（None）	LabelX、LabelY	LabelY	一致しない
なし（None）	LabelX	LabelX、LabelY	一致しない
なし（None）	[ ]	LabelX	一致
なし（None）	LabelX	[ ]	一致
なし（None）	[ ]	[ ]	一致
AtmostOne	LabelX	LabelX	一致
AtmostOne	LabelX、LabelY	LabelX、LabelY	一致しない
AtmostOne	LabelX、LabelZ	LabelX、LabelY	一致

matchT	vzProvLbl vzProvSubLbl	vzConsLbl vzConsSubLbl	結果
AtmostOne	LabelX	LabelY	一致しない
AtmostOne	[ ]	LabelX	一致しない
AtmostOne	LabelX	[ ]	一致しない
AtmostOne	[ ]	[ ]	一致

ラベルの一致

付 録 C

アクセス ポリシーの例

この章の内容は、次のとおりです。

複数のスイッチに適用される単一のポートチャネルの設定

このサンプルの XML ポリシーは、リーフ スイッチ 17 でポート チャネルを 1 つ作成し、リーフ スイッチ 18 で別のポート チャネルを作成し、リーフ スイッチ 20 で 3 つ目のチャネルを作成します。各リーフスイッチで、同じインターフェイスがポートチャネルの一部になります（インターフェイス 1/10 ～ 1/15 および 1/20 ～ 1/25）。これらのポート チャネルはすべて同じ設定になります。

```
<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>
```

複数のスイッチに適用される 2 つのポート チャネルの設定

このサンプルの XML ポリシーは、リーフ スイッチ 17 でポート チャネルを 2 つ作成し、リーフ スイッチ 18 で別のポート チャネルを作成し、リーフ スイッチ 20 で 3 つ目のチャネルを作成します。各リーフ スイッチで、同じインターフェイスがポート チャネルの一部になります（ポート チャネル 1 の場合はインターフェイス 1/10 ～ 1/15、ポート チャネル 2 の場合は 1/20 ～ 1/25）。各スイッチブロックには連続するスイッチ ID のグループを 1 つしか含めることができないため、ポリシーは 2 つのスイッチ ブロックを使用します。これらのポート チャネルはすべて同じ設定になります。

（注） ポート チャネルの設定が同じであっても、この例では、2 つの異なるインターフェイス ポリシー グループを使用します。各インターフェイス ポリシー グループは、スイッチ上のポート チャネルを表します。所定のインターフェイス ポリシー グループに関連付けられているインターフェイスはすべて、同じポート チャネルの一部です。

```
<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>
```

2 つのスイッチ間での単一の仮想ポート チャネル

2 つのスイッチ間で仮想ポート チャネルを作成するための 2 つの手順は次のとおりです。

- `fabricExplicitGEp` を作成します。このポリシーは、仮想ポート チャネルを形成するためにペアになるリーフ スイッチを指定します。
- インフラ セレクタを使用してインターフェイス コンフィギュレーションを指定します。

APIC は、`fabricExplicitGEp` の複数の検証を実行し、これらの検証のいずれかが失敗すると、障害が発生します。1 つのリーフは、他の 1 つのリーフのみとペアにできます。APIC は、このルールに違反する設定を拒否します。`fabricExplicitGEp` を作成する際、管理者はペアにするリーフ スイッチの両方の ID を提供する必要があります。APIC は、このルールに違反する設定を拒否します。両方のスイッチを `fabricExplicitGEp` の作成時に起動する必要があります。片方のスイッチが起動していない場合、APIC は設定を受け入れますが、障害を発生させます。両方のスイッチをリーフ スイッチにする必要があります。片方または両方のスイッチ ID がスパインに一致すると、APIC は設定を受け入れますが、障害を発生させます。

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

2 つのスイッチの選択されたポート ブロックでの 1 個の仮想ポート チャネル

このポリシーは、リーフ 18 ではインターフェイス 1/10 ～ 1/15 を、リーフ 25 ではインターフェイス 1/20 ～ 1/25 を使用して、リーフ スイッチ 18 および 25 で単一の仮想ポート チャネルを作成します。

```
<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>
```

インターフェイス速度の設定

このポリシーは、一連のインターフェイスのポート速度を設定します。

```
<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>
```

付 録 D

FEX VPC ポリシーの例

この章の内容は、次のとおりです。

FEX VPC の例

この章では、FEX 仮想ポート チャネル XML ポリシーを記載しています。

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

<infraNodeP name="fexNodeP105">
    <infraLeafS name="leafs" type="range">
        <infraNodeBlk name="test" from_="105" to_="105"/>
    </infraLeafS>
    <infraRsAccPortP tDn="uni/infra/accportprof-fex116nif105" />
</infraNodeP>

<infraNodeP name="fexNodeP101">
    <infraLeafS name="leafs" type="range">
        <infraNodeBlk name="test" from_="101" to_="101"/>
    </infraLeafS>
    <infraRsAccPortP tDn="uni/infra/accportprof-fex113nif101" />
</infraNodeP>

<infraAccPortP name="fex116nif105">
   <infraHPortS name="pselc" type="range">
   <infraPortBlk name="blk1"
        fromCard="1" toCard="1" fromPort="45" toPort="48" >
   </infraPortBlk>
   <infraRsAccBaseGrp tDn="uni/infra/fexprof-fexHIF116/fexbundle-fex116" fexId="116" />
</infraHPortS>
</infraAccPortP>

<infraAccPortP name="fex113nif101">
   <infraHPortS name="pselc" type="range">
   <infraPortBlk name="blk1"
        fromCard="1" toCard="1" fromPort="45" toPort="48" >
   </infraPortBlk>
   <infraRsAccBaseGrp tDn="uni/infra/fexprof-fexHIF113/fexbundle-fex113" fexId="113" />
</infraHPortS>
</infraAccPortP>

<infraFexP name="fexHIF113">
    <infraFexBndlGrp name="fex113"/>
```

```
    <infraHPortS name="pselc-fexPC" type="range">
        <infraPortBlk name="blk"
            fromCard="1" toCard="1" fromPort="15" toPort="16" >
        </infraPortBlk>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-fexPCbundle" />
    </infraHPortS>
        <infraHPortS name="pselc-fexVPC" type="range">
        <infraPortBlk name="blk"
            fromCard="1" toCard="1" fromPort="1" toPort="8" >
        </infraPortBlk>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-fexvpcbundle" />
    </infraHPortS>
    <infraHPortS name="pselc-fexaccess" type="range">
        <infraPortBlk name="blk"
            fromCard="1" toCard="1" fromPort="47" toPort="47">
        </infraPortBlk>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-fexaccport" />
     </infraHPortS>

</infraFexP>

<infraFexP name="fexHIF116">
    <infraFexBndlGrp name="fex116"/>
    <infraHPortS name="pselc-fexPC" type="range">
        <infraPortBlk name="blk"
            fromCard="1" toCard="1" fromPort="17" toPort="18" >
        </infraPortBlk>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-fexPCbundle" />
    </infraHPortS>
        <infraHPortS name="pselc-fexVPC" type="range">
        <infraPortBlk name="blk"
            fromCard="1" toCard="1" fromPort="1" toPort="8" >
        </infraPortBlk>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-fexvpcbundle" />
      </infraHPortS>
      <infraHPortS name="pselc-fexaccess" type="range">
        <infraPortBlk name="blk"
            fromCard="1" toCard="1" fromPort="47" toPort="47">
        </infraPortBlk>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-fexaccport" />
     </infraHPortS>


</infraFexP>

<infraFuncP>
<infraAccBndlGrp name="fexPCbundle" lagT="link">
    <infraRsLacpPol tnLacpLagPolName='staticLag'/>
    <infraRsHIfPol tnFabricHIfPolName="1GHIfPol" />
    <infraRsAttEntP tDn="uni/infra/attentp-fexvpcAttEP"/>
</infraAccBndlGrp>

<infraAccBndlGrp name="fexvpcbundle" lagT="node">
    <infraRsLacpPol tnLacpLagPolName='staticLag'/>
    <infraRsHIfPol tnFabricHIfPolName="1GHIfPol" />
    <infraRsAttEntP tDn="uni/infra/attentp-fexvpcAttEP"/>
</infraAccBndlGrp>
</infraFuncP>


<fabricHIfPol name="1GHIfPol" speed="1G" />
<infraAttEntityP name="fexvpcAttEP">
    <infraProvAcc name="provfunc"/>
    <infraRsDomP tDn="uni/phys-fexvpcDOM"/>
</infraAttEntityP>

<lacpLagPol dn="uni/infra/lacplagp-staticLag"
    ctrl="susp-individual,graceful-conv"
    minLinks="2"
    maxLinks="16">
</lacpLagPol>
```

付 録 E

テナント レイヤ 3 の外部ネットワーク ポリシーの例

この章の内容は、次のとおりです。

テナントの外部ネットワーク ポリシーの例

次の XML コードは、テナント レイヤ 3 の外部ネットワーク ポリシーの例です。

```
<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">
        </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>
```

付 録 F

DHCP リレー ポリシーの例

この章の内容は、次のとおりです。

レイヤ 2 およびレイヤ 3 の DHCP リレーのサンプル ポリシー

このサンプル ポリシーでは、コンシューマ テナント L3extOut DHCP リレーの設定例を示します。

```
<polUni>
    <!-- Consumer Tenant 2 -->
 <fvTenant
  dn="uni/tn-tenant1"
  name="tenant1">
  <fvCtx name="dhcp"/>

        <!-- DHCP client bridge domain -->
  <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">
             <!-- Allowed routes to L3out to send traffic  -->
             <l3extSubnet ip="100.100.100.0/24" />
         </l3extInstP>
         <l3extLNodeP name="l3extLNodeP-pc">
          <!-- VRF External loopback interface on node -->
             <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>
  <!-- Static DHCP Client Configuration -->
  <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 Server Configuration -->
        <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>
```

このサンプル ポリシーでは、コンシューマ テナント L2extOut DHCP リレーの設定例を示します。

```
<fvTenant
  dn="uni/tn-dhcpl2Out"
  name="dhcpl2Out">
  <fvCtx name="dhcpl2Out"/>
             <!-- bridge domain -->

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

  <!-- Consumer bridge domain -->
  <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>
```

付 録 G

DNS ポリシーの例

この章の内容は、次のとおりです。

DNS ポリシーの例

Sample XML for dnsProfile:

```
<!-- /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>
```

Sample xml for dns label:

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

DNS ポリシーの例

付 録 H

サンプルの RBAC 規則

サンプルの RBAC 規則

次のサンプル JSON ファイル内の RBAC 規則は、VMM ドメイン リソースへのテナント アクセスとトランス テナント アクセスの両方を可能にします。コンシューマに必要なリソースは、`uni/tn-prov1/brc-webCtrct` と `vmmp-Vmware/dom-Datacenter` です。

次の 2 つの RBAC 規則では、コンシューマのテナントがコンシューマ postman のクエリを下記の JSON ファイルにポストすることができます。

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

次の JSON ファイルにこれら 2 つの RBAC 規則が含まれています。

```
{"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",
```

サンプルの RBAC 規則


```
"headers":"",
"data":
"<fvTenant name=\"cons1\">
     <aaaDomainRef name=\"cons1\"/>\n
</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",
```

サンプルの RBAC 規則


```
"description":"",
"url":"http://solar.local:8000/api/policymgr/mo/uni/tn-prov1.xml",
"method":"POST",
"headers":"",
"data":
"<fvTenant name=\"prov1\" status=\"modified\">\n
     <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}]}
```

サンプルの RBAC 規則

付 録

L4-L7ルートピアリング設定チュートリアル

この章の内容は、次のとおりです。

L4-L7 ルート ピアリングの設定

ルート ピアリングを設定するには、まず 1 つ以上の l3extOut を作成し、サービス デバイスを接続するファブリック リーフ ノードに導入します。これらの `l3extOut` ポリシーは、ファブリック リーフで OSPF を有効にするために必要な OSPF 設定を指定します。外部通信に使用される `l3extOut` ポリシーと非常によく似ています。

`l3extOut` ポリシーは、ファブリック内外にどのルートを配付するかを制御するプレフィックスベースの EPG も指定します。`Scope=import` 属性は、どのエンドポイント プレフィックスを学習するかを制御し、外部 L4-L7 デバイスにこのルートをアドバタイズするよう指示します。`Scope=export` 属性は、ファブリックによってこのルートを L4-L7 デバイスにアドバタイズする必要があると指定します。

`l3extOut` ポリシーの 2 つの例を次に示します。OspfInternal を eth1/23 で導入する例と、OspfExternal を eth1/25 で導入する例です。

図 *101*：*`OspfInternal`* という名前の *`l3extOut`* を *`eth1/23`* で導入

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- /api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="coke{{tenantId}}">
    {% if status is not defined %}
        {%set status = "created,modified" %}
    {% endif %}

    <l3extOut name="OspfInternal" status="{{status}}">

      <l3extLNodeP name="bLeaf-101">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.11"/>
        <l3extLIfP name='portIf'>
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"
            ifInstT='ext-svi' encap='vlan-3844' addr="30.30.30.100/28" mtu='1500'/>
            <!-- <ospfIfP authKey="tecom" authType="md5" authKeyId='1'> -->
            <ospfIfP>
              <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
            </ospfIfP>
          </l3extLIfP>
      </l3extLNodeP>

      <ospfExtP areaId='111' areaType='nssa' areaCtrl='redistribute'/>

      <l3extInstP name="OspfInternalInstP">
        <l3extSubnet ip="30.30.30.100/28" scope="import"/>
        <l3extSubnet ip="20.20.20.0/24" scope="import"/>
        <l3extSubnet ip="10.10.10.0/24" scope="export"/>
      </l3extInstP>

      <l3extRsEctx tnFvCtxName="cokectx1"/>

    </l3extOut>

    <ospfIfPol name="ospfIfPol" nwT='bcast' xmitDelay='1' helloIntvl='10' deadIntvl='40' status='created,modified'/>
    </fvTenant>
  </polUni>
```

349536

図 *102*：*OspfExternal* という名前の *l3extOut* を *eth1/25* で導入

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- /api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="common">
    <fvCtx name="commonctx"/>

    {% if status is not defined %}
        {% set status = "created,modified" %}
    {% endif %}

    <l3extOut name="OspfExternal" status="{{status}}">
      <l3extLNodeP name="bLeaf-101">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28"/>
        <l3extLIfP name='portIf'>
          {% if intfType is not defined %}
            {% set intfType = "ext-svi" %}
          {% endif %}
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="{{intfType}}" encap='vlan-3843' addr="40.40.40.100/28" mtu='1500'/>
            <!-- <ospfIfP authKey="tecom" authType="md5" authKeyId='1'> -->
            <ospfIfP>
              <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
            </ospfIfP>
          </l3extLIfP>
      </l3extLNodeP>

      <ospfExtP areaId='111' areaType='nssa' areaCtrl='redistribute'/>

      <l3extInstP name="OspfExternalInstP">
        <l3extSubnet ip="40.40.40.100/28" scope="import"/>
        <l3extSubnet ip="10.10.10.0/24" scope="import"/>
        <l3extSubnet ip="20.20.20.0/24" scope="export"/>
      </l3extInstP>

      <l3extRsEctx tnFvCtxName="commonctx"/>

    </l3extOut>

    <ospfIfPol name="ospfIfPol" nwT='bcast' xmitDelay='1' helloIntvl='10' deadIntvl='40' status='created,modified'/>
    </fvTenant>
  </polUni>
```

349537

次の l3extInstP では、プレフィクス 40.40.40.100/28 と 10.10.10.0/24 を、プレフィックス ベースのエンドポイント関連付けのため、およびこれらのルートをアドバタイズする L4-L7 デバイスへのヒントとして使用するよう指定されています。

図 *103*：対応する *l3extInstP*

```
<l3extInstP name="OspfExternalInstP">
  <l3extSubnet ip="40.40.40.100/28" scope="import"/>
  <l3extSubnet ip="10.10.10.0/24" scope="import"/>
  <l3extSubnet ip="20.20.20.0/24" scope="export"/>
</l3extInstP>
```

349538

次のポリシーでは、この L3extOut を導入する場所が制御されます。

（注） ルート ピアリングを機能させるには、l3extRsPathL3OutAtt と、L4-L7 論理デバイス クラスタが接続されている RsCIfPathAtt が一致する必要があります。この要件については次のトピックで説明します。

図 *104*：*l3extRsPathL3OutAtt*

```
<l3extLNodeP name="bLeaf-101">
  <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28"/>
  <l3extLIfP name='portIf'>
    {% if intfType is not defined %}
      {% set intfType = "ext-svi" %}
    {% endif %}
    <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
      ifInstT="{{intfType}}" encap='vlan-3843' addr="40.40.40.100/28" mtu='1500'/>
      <!-- <ospfIfP authKey="tecom" authType="md5" authKeyId='1'> -->
      <ospfIfP>
        <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
      </ospfIfP>
    </l3extLIfP>
</l3extLNodeP>
```

349539

L4-L7 クラスタの l3extOut ポリシーの指定

選択ポリシー vnsLIfCtx を使用する論理デバイス クラスタに、特定の l3extOut ポリシーを使用することができます。vnsRsLIfCtxToInstP は、LIfCtx を適切な OspfInternal および OspfExternal l3extInstP EPG に向かわせます。次の例は、その実行の方法を示しています。

```
<vnsLDevCtx ctrctNameOrLbl="webCtrct{{graphId}}" graphNameOrLbl="WebGraph" nodeNameOrLbl="FW">

    <vnsRsLDevCtxToLDev tDn="uni/tn-solar{{tenantId}}/lDevVip-Firewall"/>
    <vnsLIfCtx connNameOrLbl="internal">
        {% if L3ExtOutInternal is not defined %}
        <fvSubnet ip="10.10.10.10/24"/>
        {% endif %}
        <vnsRsLIfCtxToBD tDn="uni/tn-solar{{tenantId}}/BD-solarBD1"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-solar{{tenantId}}/lDevVip-Firewall/lIf-internal"/>
        {% if L3ExtOutInternal is defined %}
        <vnsRsLIfCtxToInstP
tDn="uni/tn-solar{{tenantId}}/out-OspfInternal/instP-OspfInternalInstP"
```

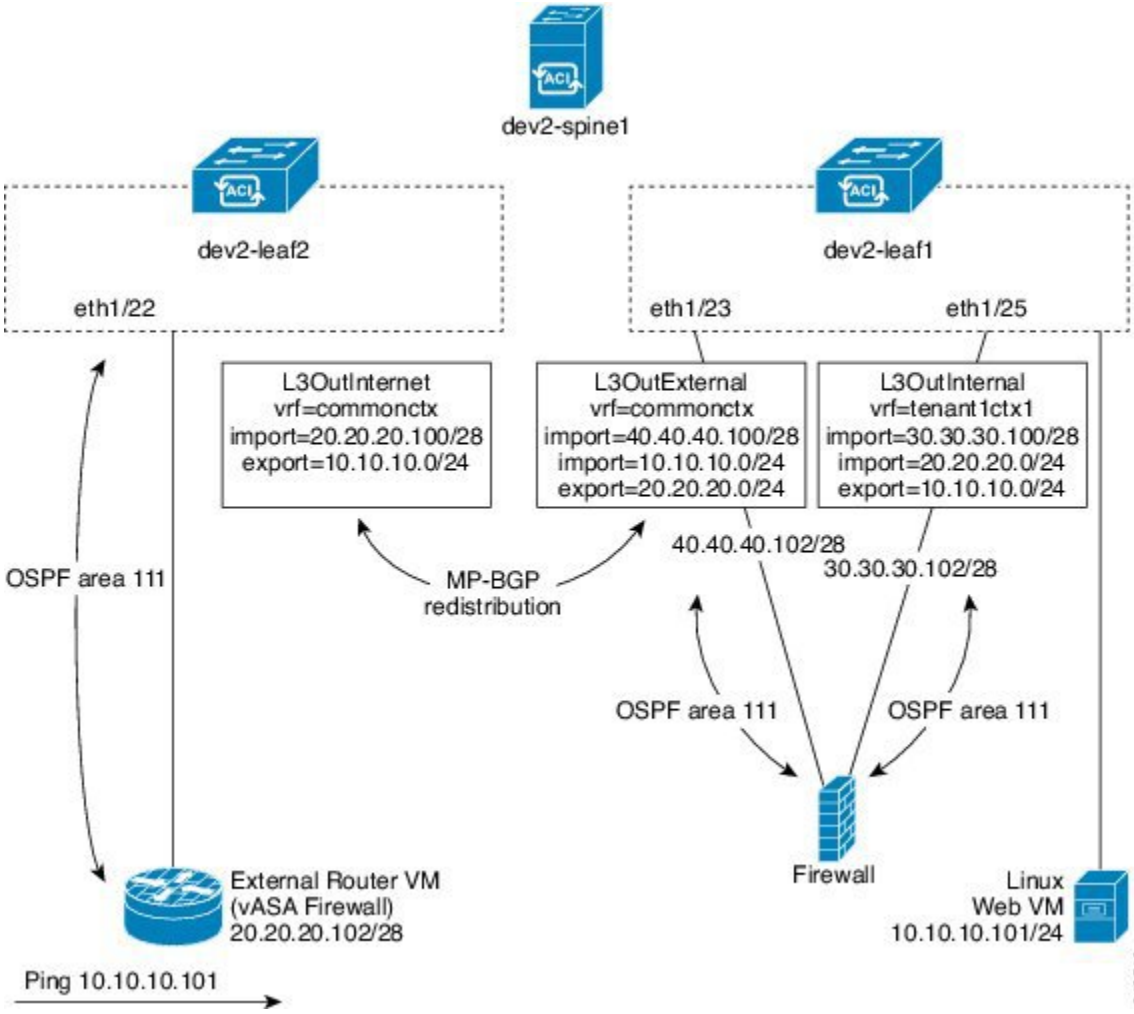
```
status={{L3ExtOutInternal}}"/>
        {% endif %}
    </vnsLIfCtx>
    <vnsLIfCtx connNameOrLbl="external">
        {% if L3ExtOutExternal is not defined %}
        <fvSubnet ip="40.40.40.40/24"/>
        {% endif %}
        <vnsRsLIfCtxToBD tDn="uni/tn-solar{{tenantId}}/BD-solarBD4"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-solar{{tenantId}}/lDevVip-Firewall/lIf-external"/>
        {% if L3ExtOutExternal is defined %}
        <vnsRsLIfCtxToInstP
tDn="uni/tn-solar{{tenantId}}/out-OspfExternal/instP-OspfExternalInstP"
status={{L3ExtOutExternal}}"/>
        {% endif %}
    </vnsLIfCtx>
</vnsLDevCtx>
```

次の例に示すように、関連付けられた具象デバイスには、それを同じファブリック リーフに導入する `vnsRsCIfPathAtt` を設定する必要があります。

```
<vnsCDev name="ASA">
    <vnsRsLDevCtxToLDev tDn="uni/tn-solar{{tenantId}}/lDevVip-Firewall"/>
    <vnsCIf name="Gig0/0">
        <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eht1/23]"/>
    </vnsCIf>
    <vnsCIf name="Gig0/1">
        <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eht1/25]"/>
    </vnsCIf>
    <vnsCMgmt name="devMgmt" host="{{asaIp}}" port="443" />
    <vnsCCred name="username" value="admin" />
    <vnsCCredSecret name="password" value="insieme" />
</vnsCDev>
```

次の図は、ルート ピアリングがエンドツーエンドに機能する様子を示しています。

図 *105*：導入例

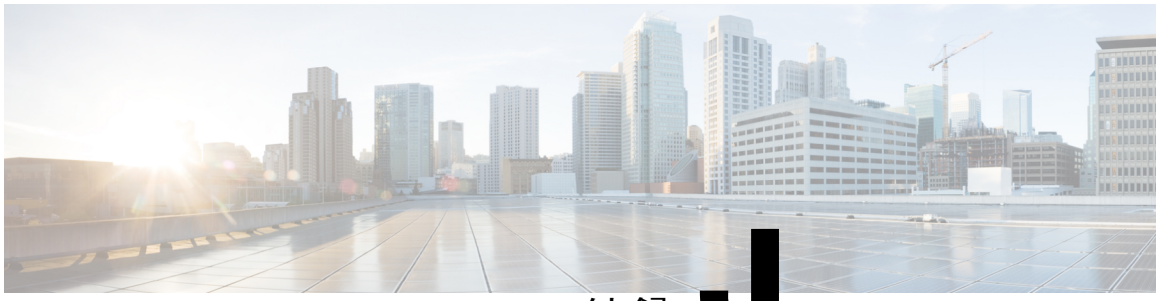


2 リーフ/1 スパインのこのトポロジでは、Linux Web サーバが IP 10.10.10.101/24 にあり、dev2-leaf1 に接続された ESX サーバでホストされます。サービス グラフは、同様に dev2-leaf1 に接続されたツーアーム ファイアウォールから構成された状態で導入されます。サービス グラフは、外部の `l3extOut` L3OutInternet をプロバイダー EPG（Web VM）と結び付けるコントラクトに関連付けられます。L3OutExternal と L3OutInteral の 2 つの内部 `l3extOut` ポリシーも、サービス デバイスが接続されているリーフ ポートに導入されます。

注意事項と制約事項

L4-L7 デバイスとのルート ピアリングを設定するときは、次のガイドラインに従ってください。

- ルート ピアリングングは仮想サービス デバイスと一緒に導入できます。この場合、vnsCIf との l3extRsPathL3OutAtt 検証は行われません。データ パスを機能させるためには、仮想サービス デバイスが接続されている正しいリーフに l3extOut を導入する必要があります。
- 現在、OSPF v2 のみテスト済みです。OSPF v3 も、若干の変更を伴いますが、機能すると予想されます。



付 録 J

コントラクト範囲の例

この章の内容は、次のとおりです。

コントラクト範囲の例

Ctx1 に EPG1 と EPG2 があり、Ctx2 に EPG3 と Epg4 があり、C1 というコントラクトを使用しており、`scope = context` であるとしましょう。

- EPG1 はコントラクト C1 を提供し、EPG2 はそれを消費します。
- EPG3 はコントラクト C1 を提供し、EPG4 はそれを消費します。

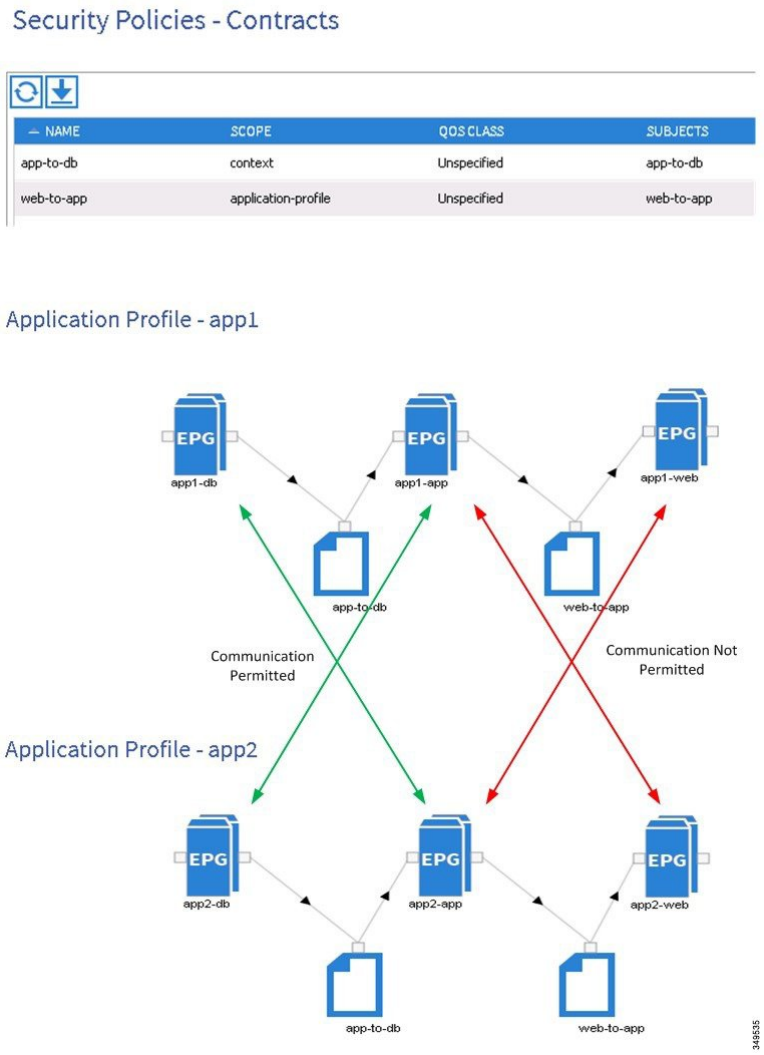
この例では、4 つの EPG すべてが同じコントラクトを共有していますが、そのうち 2 つは 1 つのコンテキスト内にあり、あと 2 つは別のコンテキスト内にあります。コントラクトは、EPG1 と EPG2 の間にのみ適用され、その後個別に EPG3 と EPG4 の間に適用されます。スコープの内容がどんなものであれ、コントラクトはスコープによる限定を受けます。この例では、コンテキストがスコープとなっています。

`scope = application profile` である場合も、同じことが適用されます。2 つのアプリケーションプロファイルに EPG があり、`scope = application profile` である場合、コントラクトはプリケーション プロファイルの EPG 上で適用されます。

コントラクト範囲の例

以下に、2 つのコントラクトの APIC GUI のスクリーンショットを示します。

図 *106*：セキュリティ ポリシー コントラクトの例



1 つのコントラクトは Web/アプリケーション間（web-to-app）通信用で、スコープはアプリケーション プロファイルとなっています。アプリケーション/データベース間（app-to-db）コントラクトのスコープはコンテキストとなっています。app1 と app2 のアプリケーション プロファイルは同じコンテキスト内にあります。各アプリケーション プロファイルには EPG が含まれます。

app-to-db コントラクトのスコープがコンテキスト レベルで適用され、どちらのアプリケーション プロファイルも同じコンテキストに属しているため、app-to-db コントラクトのすべてのコンシューマに、そのプロバイダーである EPG との通信が許可されます。

- EPG-app1-db は EPG-app1-app と双方向通信可能
- EPG-app2-db は EPG-app2-app と双方向通信可能

コントラクト範囲の例

- EPG-app1-db は EPG-app2-app と双方向通信可能
- EPG-app2-db は EPG-app1-app と双方向通信可能

次のエンドポイントペアは、スコープをアプリケーションプロファイルとして web-to-app コントラクトを使用しており、コントラクトのプロバイダーとコンシューマのみに、そのアプリケーションプロファイル内の通信を許可します。

- EPG app1 アプリケーションは EPG-app1-web と通信可能
- EPG app2 アプリケーションは EPG-app2-web と通信可能

上に示したものとは異なり、app および db EPG は、アプリケーションプロファイルの外部とは通信できません。

コントラクト範囲の例

付 録 K

セキュア プロパティ

この章の内容は、次のとおりです。

セキュア プロパティ

次の表に、プロパティ タイプがパスワード フィールドである管理対象オブジェクトのセキュア プロパティを示します。

プロパティタイプ	管理対象オブジェクトのクラス	プロパティ
パスワードフィールド	*pki:KeyRing*	*key*
	pki:WebTokenData	*hashSecret*
	pki:WebTokenData	*initializationVector*
	pki:WebTokenData	*key*
	pki:CsyncSharedKey	*key*
	pki:CertReq	*pwd*
	*mcp:Ins*t	*key*
	mcp:InstPol	*key*
	sysdebug:BackupBehavior	*pwd*
	stats:Dest	*userPasswd*
	firmware:CcoSource	*password*
	firmware:InternalSource	*password*
	f firmware:OSource	*password*
	firmware:Source	*password*
	bgp:PeerDef	*password*
	bgp:Peer	*password*
	bgp:APeerP	*password*
	bgp:PeerP	*password*
	bfd:AuthP	*key*
	comp:UsrAccP	*pwd*
	comp:Ctrlr	*pwd*
	aaa:LdapProvider	*key*
	aaa:LdapProvider	*monitoringPassword*
	aaa:UserData	*pwdHistory*

プロパティ タイプ	管理対象オブジェクトのクラス	プロパティ
	aaa:TacacsPlusProvider	*key*
	aaa:TacacsPlusProvidermonitoring	*password*
	aaa:AProvider	*key*
	aaa:AProvider	*monitoringPassword*
	aaa:RadiusProvider	*key*
	aaa:RadiusProvider	*monitoringPassword*
	aaa:User	*pwd*
	aaa:ChangePassword	*newPassword*
	aaa:ChangePassword	*oldPassword*
	ospf:AuthP	*key*
	ospf:IfPauth	*Key*
	ospf:AIfPauth	*Key*
	ospf:IfDef	*authKey*
	file:RemotePath	*userPasswd*
	file:ARemotePath	*userPasswd*
	vmm:UsrAccP	*pwd*
	snmp:UserSecP	*authKey*
	snmp:UserSecP	*privKey*
	snmp:UserP	*authKey*
	snmp:UserP	*privKey*
	snmp:AUserP	*authKey*
	snmp:AUserP	*privKey*
	vns:VOspfVEncapAsc	*authKey*
	vns:SvcPkgSource	*password*

プロパティ タイプ	管理対象オブジェクトのクラス	プロパティ
	vns:SvcPkgSource	*webtoken*
	vns:CCredSecret	*value*

付 録 L

設定ゾーンのサポート対象ポリシー

この章の内容は、次のとおりです。

設定ゾーンのサポート対象ポリシー

設定ゾーンでは次のポリシーがサポートされています。

```
cdp:IfPol
cdl:IfPolDef
cdp:InstPol
dhcp:NodeGrp
dhcp:PodGrp
edr:ErrDisRecoverPol
ep:ControlP
ep:LoopProtectP
fabric:AutoGEp
fabric:ExplicitGEp
fabric:HIfPol
fabric:ProtPol
fvns:McastAddrInstP
fvns:VlanInstDef
fvns:VlanInstP
fvns:VxlanInstDef
fvns:VxlanInstP
infra:AccBndlGrp
infra:AccBndlPolGrp
infra:AccBndlSubgrp
infra:AccCardP
infra:AccCardPGrp
infra:AccNodePGrp
infra:AccPortGrp
infra:AccPortP
infra:AttEntityP
infra:CardS
infra:ConnFexBlk
infra:ConnFexS
infra:ConnNodeS
infra:FexBndlGrp
infra:FexP
infra:FuncP
infra:HConnPortS
infra:HPortS
infra:LeafS
infra:NodeBlk
```

設定ゾーンのサポート対象ポリシー

```
infra:NodeP
infra:PodBlk
infra:PodP
infra:PodS
infra:PortBlk
infrazone:NodeGrp
infrazone:PodGrp
l2ext:DomDef
l2ext:DomP
l2:IfPol
l2:InstPol
l2:InstPolDef
l3ext:DomDef
l3ext:DomP
lacp:IfPol
lacp:LagPol
lacp:LagPolDef
lldp:IfPol
lldp:IfPolDef
lldp:InstPol
mcp:IfPol
mcp:InstPol
mgmt:NodeGrp
mgmt:PodGrp
phys:DomP
qos:DppPol
qos:DppPolDef
span:Dest
span:DestGrp
span:SpanProv
span:SrcGrp
span:SrcGrpDef
span:SrcTargetShadow
span:SrcTargetShadowBD
span:SrcTargetShadowCtx
span:TaskParam
span:VDest
span:VDestDef
span:VDestGrp
span:VDestGrpDef
span:VSpanProv
span:VSrcGrp
span:VSrcGrpDef
stormctrl:IfPol
stp:EncapInstDef
stp:IfPol
stp:IfPolDef
stp:InstPol
stp:MstDomPol
stp:MstRegionPol
vmm:DomP
vmm:DomPDef
vpc:InstPol
vpc:KAPol
```

付 録 M

用語集

この章の内容は、次のとおりです。

用語集

アプリケーション セントリック インフラストラクチャ（ACI）：ACI は、一元化された自動化とポリシーに基づくアプリケーションプロファイルを備えた、総合的なデータセンターアーキテクチャです。

Application Policy Infrastructure Controller（APIC）：スケーラブル マルチテナント ファブリックを管理する、ACIアーキテクチャの主要コンポーネント。APICコントローラは、マルチテナント ファブリックの管理、ポリシー プログラミング、アプリケーション展開、およびヘルス モニタリングを提供する複製同期されたクラスタ化コントローラを構成します。

コンシューマ：サービスを消費するエンドポイント グループ（EPG）。

コンテキスト：レイヤ 3 アドレス ドメインを定義します。

コントラクト：EPG 間でどのような通信がどのように行われるかを指定する規則。

識別名（DN）：MO について説明し、MIT 内の場所を検索する一意の名前。

エンドポイント グループ（EPG）：エンドポイントの集合を含む名前付き論理エンティティである MO。エンドポイントは、ネットワークに直接的または間接的に接続されたデバイスです。エンドポイントには、アドレス（ID）、ロケーション、属性（バージョンやパッチ レベルなど）があり、物理または仮想にできます。エンドポイントの例には、インターネット上のサーバ、仮想マシン、ストレージ、またはクライアントが含まれます。

フィルタ：EPG 間のインバウンドまたはアウトバウンド通信を定義するコントラクト内で使用される、レイヤ 3 プロトコル タイプ、レイヤ 4 ポートなどの TCP/IP ヘッダー フィールド。

ラベル：1 つのプロパティ、つまり名前を持つ管理対象オブジェクト。ラベルにより、互いに通信できるオブジェクトとできないオブジェクトを分類できます。

管理対象オブジェクト（MO）：ファブリック リソースの抽象化。

管理情報ツリー（MIT）：ファブリックのすべての MO を含む階層型管理情報ツリー。

外部ネットワーク：ファブリックの外部のネットワークへの接続を定義する MO。

ポリシー：システム挙動の一定の側面を制御するための一般的な仕様を含む名前付きエンティティ。たとえば、レイヤ 3 外部ネットワーク ポリシーには BGP プロトコルが含まれ、ファブリックを外部レイヤ 3 ネットワークに接続する場合に BGP ルーティング機能をイネーブルにできます。

プロファイル：ポリシーの 1 つ以上のインスタンスを実行するのに必要な詳細設定を含む名前付きエンティティ。たとえば、ルーティング ポリシーのスイッチ ノード プロファイルには、BGP を実行するのに必要なすべてのスイッチの具体的な設定詳細が含まれます。

プロバイダー：サービスを提供する EPG。

サブジェクト：どの情報をどのように伝えるかを指定するコントラクトに含まれる MO。

ターゲット DN（tDn）：ソース MO とターゲット MO の特定のインスタンス間の関係を定義する明示的なリファレンス。ターゲット インスタンスは、関係ソース（Rs）MO で明示的に設定されたターゲット DN（tDn）のプロパティによって識別されます。

テナント：テナントは、アプリケーション ポリシーの論理コンテナで、管理者はドメインベースのアクセス コントロールを実行できます。テナントはポリシーの観点から分離の単位を表しますが、プライベート ネットワークは表しません。テナントは、サービス プロバイダーの環境ではお客様を、企業の環境では組織またはドメインを、または単にポリシーの便利なグループ化を表すことができます。テナントが含む主要な要素は、フィルタ、コントラクト、外部、ブリッジ ドメイン、および EPG を含むアプリケーション プロファイルです。