



## 付録

---

- クラス (1 ページ)
- パッケージデコーダ (13 ページ)
- 頭字語とその定義 (19 ページ)
- 参考資料 (40 ページ)

## クラス

Application Policy Infrastructure Controller (APIC) のクラスは、システムイベントおよびフォールトがオブジェクトモデル内のオブジェクトとどのように関連しているかを理解するために操作面で重要です。システムの各イベントやフォールトは、設定、正常性、フォールト、および/または統計情報にアクセスできる固有のオブジェクトです。

アプリケーションセントリック インフラストラクチャ ファブリックを構成するすべての物理および論理コンポーネントは、階層管理情報ツリー (MIT) で表されます。このツリー内の各ノードは、管理ステータスと動作ステータスを含む、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。

プログラムの REST API は、REST アーキテクチャを使用します。API は、JSON または XML 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。プログラミング言語を使用して、API メソッドまたは管理対象オブジェクト (MO) の説明を含むメッセージおよび JSON または XML ドキュメントを生成できます。

HTTP/1.1 または HTTPS POST、GET、または DELETE メッセージを APIC に送信することで、API 機能呼び出すことができます。POST メッセージの HTML 本文には、MO または API メソッドを記述する JSON または XML データ構造が含まれます。応答メッセージの HTML 本文は要求されたアクションが要求されたデータ、確認、エラー情報が含まれる JSON または XML 構造が含まれます。

クラスの完全なリストにアクセスするには、APIC を指定し、URL の最後で **doc/html** ディレクトリを参照します。

```
https://apic_ip_address/doc/html/
```

## ファブリックのモニタリング

### topSystem

名前 : top:System

説明 : コントローラ、リーフおよびスパインなど、ファブリック内のすべてのデバイスのリストを提供します。

使用 : topSystem クラスは、inb/oob 管理の詳細、現在時刻、システムの稼働時間と現在の状態を含むオブジェクトプロパティを取得するために使用できます。

topSystem REST :: <https://172.16.96.2/api/node/class/topSystem.json>

### fabricNode

名前 : fabric:Node

説明 : コントローラ、リーフおよびスパインなど、ファブリックの一部であるすべてのノードのリストを提供します。

使用 : fabricNode クラスは、ノードのシリアル番号、割り当てられたノードID、ノードのモデル番号とデバイス ロールを含むオブジェクトプロパティを取得するために使用できます。

fabricNode REST :: <https://172.16.96.2/api/node/class/fabricNode.json>

### faultInst

名前 : fault:Inst

説明 : フォールトの詳細情報が含まれます。このオブジェクトは、フォールト状態が発生したオブジェクトの子としてアタッチされます。親オブジェクトの各フォールト状態に対し1つのインスタンス オブジェクトが作成されます。フォールト インスタンス オブジェクトはフォールト コードで識別されます。

使用 : faultInst クラスは、APIC内のファブリック、テナントまたは個々の管理対象オブジェクトに関連付けられたすべてのフォールトを取得するために使用できます。

faultInst REST :: <https://172.16.96.2/api/node/class/faultInst.json>

### fabricHealthTotal

名前 : fabric:HealthTotal

説明 : ファブリックの総ヘルス スコア インスタンス。

使用 : fabricHealthTotal クラスは、システム全体の状態を取得するために使用できます。

fabricHealthTotal REST :: <https://172.16.96.2/api/node/class/fabricHealthTotal.json>

### fvCEp

名前 : fv:CEp

説明 : ネットワークにアタッチしているクライアントエンドポイント。

使用：fvCEp クラスは、ファブリックにアタッチされたエンドポイントおよび関連付けられている IP/MAC アドレス、および各オブジェクトのカプセル化のリストを取得するために使用できます。

fvCEp REST :: <https://172.16.96.2/api/node/class/fvCEp.json>

### fvRsCEpToPathEp

名前：fv:RsCEpToPathEp

説明：これは、パスのエンドポイントとの関係を提供する内部オブジェクトです。

使用：fvRsCEpToPathEp クラスは、ノードおよびポートなどのパスのファブリック詳細と、テナント名、アプリケーションプロファイルおよびエンドポイントグループなどのテナントの詳細を取得するために使用できます。

fvRsCEpToPathEp REST :: <https://172.16.96.2/api/node/class/fvRsCEpToPathEp.json>

### eqptFabP

名前：eqpt:FabP

説明：ファブリックポート、外部 IO ポートに向いているファブリック。

使用：eqptFabP クラスは、ファブリックポートと、ラインカードやシャーシの位置などの関連情報のリストを取得するために使用できます。

eqptFabP REST :: <https://172.16.96.2/api/node/class/eqptFabP.json>

### eqptLeafP

名前：eqpt:LeafP

説明：ファブリックポート、外部リーフ IO ポートに向いている非ファブリック。

使用：eqptFabP クラスは、非ファブリックポートと、ラインカードやシャーシの位置などの関連情報のリストを取得するために使用できます。

eqptLeafP REST :: <https://172.16.96.2/api/node/class/eqptLeafP.json>

### eqptCh

名前：eqpt:ChA

説明：ハードウェアのシャーシコンテナ。

使用：eqptCh クラスは、シャーシのリストと、動作状態、シリアル番号、モデル番号などの関連情報を取得するために使用できます。

eqptCh REST :: <https://172.16.96.2/api/node/class/eqptCh.json>

### eqptLC

名前：eqpt:LCA

説明：IO ポートを含むラインカード (IO カード)。

使用：eqptLC クラスは、ファブリック内に展開されたラインカードと、冗長化の状態、モデル、シリアル番号、ポート数などの関連情報のリストを取得するために使用できます。

```
eqptLC REST :: https://172.16.96.2/api/node/class/eqptLC.json
```

### eqptFt

名前：eqpt:Ft

説明：インベントリ化されているファントレイ。

使用：eqptFt クラスは、ファントレイと、動作状態、モデル番号、シリアル番号、ハードウェアバージョンなどの関連情報のリストを取得するために使用できます。

```
eqptFt REST :: https://172.16.96.2/api/node/class/eqptFt.json
```

### eqptPsu

名前：eqpt:Psu

説明：電源装置。

使用：eqptPsu クラスは、ファブリック内の電源装置と、モデル番号、シリアル番号、動作ステータス、電圧源などの関連情報のリストを取得するために使用できます。

```
eqptPsu REST :: https://172.16.96.2/api/node/class/eqptPsu.json
```

### eqptSupC

名前：eqpt:SupC

説明：コントロールプレーンを実行中の CPU を含む、スーパーバイザカード。

使用：eqptSupC クラスは、ファブリック内に展開されたスーパーバイザカードと、モデル番号、シリアル番号、動作ステータス、冗長化の状態などの関連情報のリストを取得するために使用できます。

```
eqptSupC REST :: https://172.16.96.2/api/node/class/eqptSupC.json
```

### ethpmPhysIf

名前：ethpm:PhysIf

説明：物理インターフェイスの情報ホルダー。

使用：ethpmPhysIf クラスは、ファブリック内の物理インターフェイスと、速度、デュプレックス、動作ステータス、使用状態などの関連情報のリストを取得するために使用できます。

```
ethpmPhysIf REST :: https://172.16.96.2/api/node/class/ethpmPhysIf.json
```

### dbgAcTrail

名前：dbg:AcTrail

説明：アトミックカウンタ証跡。

使用：dbgAcTrail クラスは、ファブリック内に展開されたアトミックカウンタと、ドロップされたパケット統計情報やパケット数などの関連情報のリストを取得するために使用できます。

```
dbgAcTrail REST :: https://172.16.96.2/api/node/class/dbgAcTrail.json
```

### dbgEpgToEpgRsIt

名前 : dbg:EpgToEpgRsIt

説明 : エンドポイント グループからエンドポイント グループへのアトミック カウンタ、オンデマンド、エントリ。

使用 : dbgEpgToEpgRsIt クラスは、ファブリック内に展開された EPG から EPG へのアトミックカウンタと、ドロップされたパケット統計情報やパケット数などの関連情報のリストを取得するために使用できます。

```
dbgEpgToEpgRsIt REST :: https://172.16.96.2/api/node/class/dbgEpgToEpgRsIt.json
```

### dbgEpToEpRsIt

名前 : dbg:EpToEpRsIt

説明 : エンドポイントからエンドポイントへのアトミック カウンタ、オンデマンド、エントリ。

使用 : dbgEpToEpRsIt クラスは、ファブリック内に展開されたエンドポイントからエンドポイントへのアトミックカウンタと、ドロップされたパケット統計情報やパケット数などの関連情報のリストを取得するために使用できます。

```
dbgEpToEpRsIt REST :: https://172.16.96.2/api/node/class/dbgEpToEpRsIt.json
```

## VM\_Monitoring

### compVm

名前 : comp:Vm

説明 : 仮想マシン オブジェクト。

使用 : compVm クラスは、ファブリック内に展開された仮想マシンと、名前や状態などの関連情報のリストを取得するために使用できます。

```
compVm REST :: https://172.16.96.2/api/node/class/compVm.json
```

### compHv

名前 : comp:Hv

説明 : コンピューティング ハイパーバイザを表すオブジェクト。

使用 : compVm クラスは、ファブリック内に展開されたコンピューティング ハイパーバイザと、名前やステータスなどの関連情報のリストを取得するために使用できます。

```
compHv REST :: https://172.16.96.2/api/node/class/compHv.json
```

### fvRsVm

名前 : fv:RsVm

説明：ハイパーバイザに接続された仮想マシンとの関係。これは内部オブジェクトです。

使用：fvRsVm クラスは、ハイパーバイザに接続された仮想マシンの関係を取得するために使用できます。

fvRsVm REST :: <https://172.16.96.2/api/node/class/fvRsVm.json>

### fvRsHyper

名前：fv:RsHyper

説明：APIC VM を制御およびモニタするハイパーバイザとの関係。これは内部オブジェクトです。

使用：fvRsHyper クラスは、APIC VM を制御およびモニタするハイパーバイザの関係を取得するために使用できます。

fvRsHyper REST :: <https://172.16.96.2/api/node/class/fvRsHyper.json>

### vmmCtrlrP

名前：vmm:CtrlrP

説明：VMM コントローラプロファイル。ポリシー適用のドメインを含めることの一部である単一の VM 管理コントローラへの接続方法を指定します。たとえば、VMM コントローラプロファイルは、VMM ドメインの一部である VMware vCenter に接続するためのポリシーにできます。

使用：vmmCtrlrP クラスは、接続された VM ドメインの IP アドレスとデータセンター名を取得するために使用できます。

vmmCtrlrP REST :: <https://172.16.96.2/api/node/class/vmmCtrlrP.json>

## レイヤ4～レイヤ7のモニタリング

### vnsAbsGraph

名前：vnsAbsGraph

説明：抽象グラフは、抽象的なノードから構成され、ロードバランシング、SSL オフロード、またはファイアウォールなどのサービス機能によるトラフィックフローを定義するために使用されます。抽象的なノードは、サービス ノード バランサ (SLB) またはファイアウォール (FW) などのサービスノード、抽象的な条件ノード (エンドポイントグループに接続されているノード)、および接続から構成されます。

使用：クラス vnsAbsGraph は、APIC で設定されたサービス グラフ テンプレートとそのプロパティのリストを取得するために使用できます。

vnsAbsGraph REST :: <https://172.16.96.2/api/node/class/vnsAbsGraph.json>

### vnsLDevVip

名前：vnsLDevVip

説明：L4-L7 デバイス クラスタ。単一の仮想 IP (VIP) によって表されます。設定は、VIP アドレスにプッシュダウンされます。

使用：クラス `vnsLDevVip` は、ファブリック内の論理デバイス クラスタ用に設定されたすべての VIP を取得するために使用できます。

```
vnsLDevVip REST :: https://172.16.96.2/api/node/class/vnsLDevVip.json
```

### **vnsCDev**

名前：vnsCDev

説明：個々のサービス デバイス。具体的な L4-L7 サービス デバイスを定義するために使用されます。

使用：クラス `vnsCDev` は、L4-L7 サービス統合の一部として設定された具体的なデバイスのリストを取得するために使用できます。

```
vnsCDev REST :: https://172.16.96.2/api/node/class/vnsCDev.json
```

### **vnsLif**

名前：vnsLif

説明：L4-L7 デバイス クラスタからの具体的なインターフェイスのセットに関連付けられている論理インターフェイス。

使用：クラス `vnsLif` は、サービス グラフとデバイス インターフェイス間の接続を取得するために使用できます。

```
vnsLif REST :: https://172.16.96.2/api/node/class/vnsLif.json
```

### **vnsLDevCtx**

名前：vnsLDevCtx

説明：コントラクト、サブジェクト、機能ラベルまたは名前に基づいて特定のデバイスを選択するのに使用されるデバイス クラスタを指すデバイス クラスタ コンテキスト。ワイルドカードを指定するには、名前を [Any] に設定します。

使用：クラス `vnsLDevCtx` は、ノードおよびコントラクト名を取得するために使用できます。

```
vnsLDevCtx REST :: https://172.16.96.2/api/node/class/vnsLDevCtx.json
```

### **vnsRsLDevCtxToLDev**

名前：vnsRsLDevCtxToLDev

説明：サービス デバイス クラスタまたはテナント内の論理デバイス クラスタ用のプロキシオブジェクトの抽象化に対する送信元の関係。

使用：クラス `vnsRsLDevCtxToLDev` は、`vnsLDevCtx` と `vnsLDev` 間の関係を取得するために使用できます。

```
vnsRsLDevCtxToLDev REST :: https://172.16.96.2/api/node/class/vnsRsLDevCtxToLDev.json
```

## 統計情報

### compHostStats1h

名前 : comp:HostStats1h

説明 : 1 時間のサンプリング間隔におけるホストの最新の統計情報を表すクラス。このクラスは 15 分ごとに更新されます。

使用 : compHostStats1h クラスは、コンピューティング ハイパーバイザに関連付けられた統計情報を取得するために使用できます。

```
compHostStats1h REST :: https://172.16.96.2/api/node/class/compHostStats1h.json
```

### compRcvdErrPkts1h

名前 : comp:RcvdErrPkts1h

説明 : 1 時間のサンプリング間隔における受信したエラーパケットの最新の統計情報を表すクラス。このクラスは 15 分ごとに更新されます。

使用 : compRcvdErrPkts1h クラスは、受信したエラーパケットの最新の統計情報を取得するために使用できます。

```
compRcvdErrPkts1h REST :: https://172.16.96.2/api/node/class/compRcvdErrPkts1h.json
```

### compTrnsmtErrPkts1h

名前 : comp:TrnsmtErrPkts1h

説明 : 1 時間のサンプリング間隔における送信されたエラーパケットの最新の統計情報を表すクラス。このクラスは 15 分ごとに更新されます。

使用 : compTrnsmtErrPkts1h クラスは、送信されたエラーパケットの最新の統計情報を取得するために使用できます。

```
compTrnsmtErrPkts1h REST :: https://172.16.96.2/api/node/class/compTrnsmtErrPkts1h.json
```

## Authentication, Authorization, and Accounting (認証、許可、およびアカウントリング)

### aaaModLR

名前 : aaa:ModLR

説明 : AAA の監査ログレコード。ログレコードは、ユーザがオブジェクトを変更するたびに自動的に生成されます。

使用 : aaaModLR クラスは、すべての変更およびイベントに関するファブリックベースの監査ログを取得するために使用できます。

```
aaaModLR REST :: https://172.16.96.2/api/node/class/aaaModLR.json
```



**aaaUser**

名前：aaa:User

説明：ローカル認証されたユーザ アカウント。

使用：aaaUser クラスは、ファブリック内に展開されたユーザ アカウントのリストを取得するために使用できます。

```
aaaUser REST :: https://172.16.96.2/api/node/class/aaaUser.json
```

**aaaRemoteUser**

名前：aaa:RemoteUser

説明：リモート ユーザのログイン アカウント。

使用：aaaUser クラスは、ファブリック内に展開されたリモート ユーザ アカウントのリストを取得するために使用できます。

```
aaaRemoteUser REST :: https://172.16.96.2/api/node/class/aaaRemoteUser.json
```

## ファブリックの容量

**ポリシー TCAM**

名前：eqptcapacityPolEntry5min

説明：ポリシー CAM エントリの統計情報。5 分間のサンプリング間隔におけるポリシー エントリの最新の統計情報を表すクラス。このクラスは 10 秒ごとに更新されます。

使用：eqptcapacityPolEntry5min クラスは、ポリシー TCAM の使用に関連付けられた現在の値を取得するために使用できます。

```
eqptcapacityPolEntry5min REST ::  
https://172.16.96.2/api/class/eqptcapacityPolEntry5min.json
```

**プレフィックス TCAM**

名前：eqptcapacityL3Entry5min

説明：Layer3 エントリの統計情報。5 分間のサンプリング間隔における layer3 エントリの最新の統計情報を表すクラス。このクラスは 10 秒ごとに更新されます。

使用：eqptcapacityL3Entry5min クラスは、プレフィックス TCAM の使用に関連付けられた現在の値を取得するために使用できます。

```
eqptcapacityL3Entry5min REST ::  
https://172.16.96.2/api/class/eqptcapacityL3Entry5min.json
```

## SNMP および syslog

### SNMP トラップの宛先

名前：snmpTrapDest

説明："background-color:rgb(255, 255, 255); color:rgb(51, 51, 51)"> トラップおよび通知が送信される宛先。

使用：snmpTrapDest クラスは、ファブリック内で実装される SNMP トラップ宛先の現在のリストを取得するために使用できます。

```
snmpTrapDest REST :: https://172.16.96.2/api/node/class/snmpTrapDest.json
```

### syslog リモート宛先ホスト

名前：syslogRemoteDest

説明：syslog リモート宛先ホストによって、APIC およびファブリック ノードからのメッセージの転送先となる syslog サーバを指定できます。

使用：syslogRemoteDest クラスは、ファブリック内で実装される syslog リモート宛先の現在のリストを取得するために使用できます。

```
syslogRemoteDest REST :: https://172.16.96.2/api/node/class/syslogRemoteDest.json
```

## 使用例

次の使用例 #1 および #2 で使用されているクラス **faultInst** は、前述の、または APIC ドキュメント内で指定された、管理対象オブジェクトクラスのいずれかに置き換えることができます。『Cisco APIC Command-Line Interface User Guide』も、次の項を理解する上で役立ちます。

**ケース 1**：ファブリック内のフォールトの現在のリストを取得するためのアプリケーションスクリプトの作成。

この使用例は一般的に、ACI 管理者がファブリック内の現在のフォールトのリストを取得する環境向けです。ユーザには、CLI、Visore、POSTMAN および/または Cobra によって結果を収集するオプションがあります。アプリケーション固有のアクセスと説明については上記の項を参照してください。

CLI の観点からは、次のコマンドを使用してクエリを実行します。

```
admin@apic1:~> moquery -c faultInst
```

Visore の観点からは、次のパラメータを使用してクエリを実行します。

```
Class or DN      :: faultInst
Property         :: n/a
Op               :: n/a
Value            :: n/a
```

POSTMAN の観点からは、次の REST GET を使用してクエリを実行します。

```
GET http://<your apic ip address>/api/node/class/faultInst.xml
```

Cobra の観点からは、次のクラス クエリを使用します。

```
# Class Query
classQuery= ClassQuery(' faultInst')
for fault in md.query(classQuery):
    print fault.name
```

次は、ファブリック内のフォールトをキャプチャするための Cobra スクリプトのサンプルです。

```
#!/usr/bin/env python
import cobra.mit.access
import cobra.mit.session
from cobra.mit.session import LoginSession
from cobra.mit.request import ClassQuery
ls = cobra.mit.session.LoginSession('https://<your apic ip address>, <username>,
<password>, secure=False)
md = cobra.mit.access.MoDirectory(ls)
md.login()
# Class Query
classQuery= ClassQuery(' faultInst')
for fault in md.query(classQuery):
    print fault.name
```

**ケース 2：設定の失敗によるファブリック内のフォールトの現在のリストを取得するためのアプリケーションスクリプトの作成。**

この使用例は一般的に、ACI管理者がファブリック内の現在のフォールトのリストを取得する環境向けです。ユーザには、CLI、Visore、POSTMAN および/または Cobra によって結果を収集するオプションがあります。アプリケーション固有のアクセスと説明については上記の項を参照してください。

CLI の観点からは、次のコマンドを使用してクエリを実行します。

```
admin@apic1:~> moquery -c faultInst -f 'fv.faultInst.cause=="config-failure"
```

Visore の観点からは、次のパラメータを使用してクエリを実行します。

```
Class or DN    :: faultInst
Property      :: cause
Op            :: ==
Value        :: config-failure
```

POSTMAN の観点からは、次の REST GET を使用してクエリを実行します。

```
GET http://<your apic ip address>/api/node/class/faultInst.xml?
query-target-filter=and(eq(faultInst.cause,"config-failure"))
```

Cobra の観点からは、次のクラス クエリを使用します。

```
# Class Query
classQuery= ClassQuery(' faultInst')
classQuery.propFilter = 'wcard(faultInst. cause, "{0}")'.format(' config-failure')
for fault in md.query(classQuery):
    print fault.name
```

設定不良によるフォールトをキャプチャするための Cobra スクリプト。

```
#!/usr/bin/env python
import cobra.mit.access
import cobra.mit.session
from cobra.mit.session import LoginSession
from cobra.mit.request import ClassQuery
ls = cobra.mit.session.LoginSession('https://<your apic ip address>, <username>,
<password>, secure=False)
md = cobra.mit.access.MoDirectory(ls)
md.login()
# Class Query
classQuery= ClassQuery(' faultInst')
for fault in md.query(classQuery):
    print fault.name
```

### ケース 3：特定の管理対象オブジェクト DN のプロパティを取得するためのアプリケーションスクリプトの作成

この使用例は一般的に、ACI 管理者がテナント名 **Common** のプロパティを取得する環境向けです。ユーザには、CLI、Visore、POSTMAN および/または Cobra によって結果を収集するオプションがあります。アプリケーション固有のアクセスと説明については上記の項を参照してください。

CLI の観点からは、次のコマンドを使用してクエリを実行します。

```
admin@apic1:~> moquery -d uni/tn-common
```

Visore の観点からは、次のパラメータを使用してクエリを実行します。

```
Class or DN    :: uni/tn-common
Property       :: n/a
Op             :: n/a
Value         :: n/a
```

POSTMAN の観点からは、次の REST GET を使用してクエリを実行します。

```
GET http://<your apic ip address>/api/node/mo/uni/tn-common.xml?query-target=self
```

Cobra の観点からは、次のクラス クエリを使用します。

```
# DN Query
dnQuery= DnQuery(' uni/tn-common')
for results in md.query(dnQuery):
    print results.dn
```

設定不良によるフォールトをキャプチャするための Cobra スクリプト。

```
#!/usr/bin/env python
import cobra.mit.access
import cobra.mit.session
from cobra.mit.session import LoginSession
from cobra.mit.request import DnQuery
ls = cobra.mit.session.LoginSession('https://<your apic ip address>, <username>,
<password>, secure=False)
md = cobra.mit.access.MoDirectory(ls)
md.login()
# DN Query
```

```
dnQuery= DnQuery('uni/tn-common')
for results in md.query(dnQuery):
    print results.dn
```

#### ケース 4 : ファブリックにアタッチされているエンドポイント (MAC アドレス) の現在のリストを取得するためのアプリケーションスクリプトの作成

この使用例は一般的に、ファブリックにアタッチされている現在のエンドポイントのリストと、各エンドポイントに関連するノードの詳細をキャプチャするアプリケーションスクリプトを ACI 管理者が作成する環境向けです。

設定不良によるフォールトをキャプチャするための Cobra スクリプト。

```
#!/usr/bin/env python
from cobra.mit.access import MoDirectory
from cobra.mit.session import LoginSession
from cobra.mit.request import ClassQuery
lls = cobra.mit.session.LoginSession('https://<your apic ip address>, <username>,
<password>', secure=False)
md = MoDirectory(lls)
md.login()
q = ClassQuery('fvCEP')
q.subtree = 'children'
q.subtreeClassFilter = 'fvRsCEPToPathEp'
mos = md.query(q)
for mo in mos:
    for child in mo.rscEpToPathEp:
        print child.dn
```

## パッケージ デコーダ

ACI オブジェクトモデルのクラス名に使用される省略形がいくつかあります。次に、一般的に使用される省略形について説明します。これらは、クラスオブジェクトを REST コールで使用する際にクラス オブジェクトが何であるかを解読するときに役立ちます。

- aaa** : 認証、許可、アカウントインテグレーション
- ac** : アトミック カウンタ
- actrl** : アクセス制御
- actrlcap** : アクセス コントロール機能
- adcom** : アプライアンス ディレクタ通信
- aib** : 隣接関係情報ベース
- arp** : Address Resolution Protocol
- bgp** : Border Gateway Protocol
- callhome** : Cisco Smart Call Home サービス
- cap** : 機能
- cdp** : Cisco Discovery Protocol

**cnw** : ノード クラスタ

**comm** : 通信ポリシー

**comp** : コンピューティング

**compat** : 互換性

**condition** : 正常性ポリシー

**config** : 設定ポリシー

**coop** : Council of Oracles プロトコル

**copp** : コントロールプレーン ポリシング ポリシー。ポリサー レートを説明する一連のルールが含まれます。

**ctrlr** : コントローラ

**ctx** : コンテキスト

**datetime** : 日付/時刻ポリシー

**dbg** : デバッグ

**dbgac** : デバッグのアトミック カウンタ

**dbgexp** : デバッグのエクスポート ポリシー

**dhcp** : Dynamic Host Configuration Protocol

**dhcptlv** : Dynamic Host Configuration Protocol のタイプ、長さ、値

**dhcptlvpol** : Dynamic Host Configuration Protocol のタイプ、長さ、値ポリシー

**dns** : ドメイン名サービス

**draw** : GUI のグラフの可視化

**epm** : エンドポイント マネージャ

**eqpt** : 機器

**eqptcap** : 機器の機能

**eqptcapacity** : 機器のキャパシティ

**eqptdiag** : 機器の診断

**eqptdiagp** : 機器の診断ポリシー

**ethpm** : イーサネット ポリシー マネージャ

**event** : イベント ポリシー

**extnw** : 外部ネットワーク

**fabric** : ファブリック

**fault** : フォールトポリシー、カウンタ

**file** : ファイルパス、設定のインポート/エクスポート ポリシー

**firmware** : ファームウェア

**fmcaster** : ファブリック マルチキャスト

**fsm** : 有限状態マシン

**fv** : ファブリックの仮想化

**fvns** : ファブリックの仮想化の名前空間

**fvtopo** : ファブリックの仮想化トポロジ

**geo** : 位置情報

**glean** : グリーニング隣接関係

**ha** : ハイ アベイラビリティ

**health** : ヘルス スコア

**hvs** : ハイパーバイザ仮想スイッチ

**icmp** : インターネット制御プロトコル

**icmpv4** : インターネット制御プロトコルバージョン4

**icmpv6** : インターネット制御プロトコルバージョン6

**ident** : アイデンティティ

**igmp** : インターネット グループ管理プロトコル

**igmpsnop** : インターネット グループ管理プロトコル スヌーピング

**im** : インターフェイス マネージャ モジュール

**imginstall** : イメージ インストール

**infra** : インフラストラクチャ

**ip** : インターネットプロトコル

**ipv4** : インターネットプロトコルバージョン4

**ipv6** : インターネットプロトコルバージョン6

**isis** : Intermediate System to Intermediate System

**isistlv** : Intermediate System to Intermediate System のタイプ、長さ、値

**I1** : レイヤ1

**I1cap** : レイヤ1 機能

**I2** : レイヤ2

**I2cap** : レイヤ2 機能

**I2ext** : レイヤ2 の外部

**I3** : レイヤ3

**l3cap** : レイヤ 3 機能

**l3ext** : レイヤ 3 の外部

**l3vm** : レイヤ 3 仮想マシン

**lacp** : リンク集約プロトコル

**lbp** : ロード バランシング ポリシー

**leqpt** : ルーズ機器 (ファブリック外の管理対象外ノード)

**lldp** : Link Layer Discovery Protocol

**lldptlv** : Link Layer Discovery Protocol のタイプ、長さ、値

**lldptlvpol** : Link Layer Discovery Protocol のタイプ、長さ、値のポリシー

**maint** : メンテナンス

**mcast** : マルチキャスト

**mcp** : マスター制御プロセッサ

**memory** : メモリ統計情報

**mgmt** : 管理

**mo** : 管理対象オブジェクト

**mock** : モック (統計情報/フォールトなどの表示のために主にシミュレータで使用されるオブジェクト)

**mon** : モニタリング

**monitor** : モニタ(PAN)

**naming** : 名前付きオブジェクトの概要

**nd** : ネイバー探索

**nw** : ネットワーク

**oam** : イーサネットの操作、管理および管理

**observer** : 統計情報、フォールト、状態、ヘルス、ログ/履歴のオブザーバ

**opflex** : OpFlex

**os** : オペレーティング システム

**ospf** : Open Shortest Path First

**pc** : ポート チャネル

**pcons** : \*\*内部プロセスによって生成され、使用されます\*\*

**phys** : 物理的なドメイン プロファイル

**ping** : ping の実行と結果

**pki** : 公開キー インフラストラクチャ



**pol** : ポリシー定義

**policer** : トラフィック ポリシング (レート制限)

**pool** : オブジェクトプール

**pres** : \*\*内部プロセスによって生成され、使用されます\*\*

**proc** : システム負荷、CPU およびメモリ使用率の統計情報

**psu** : 電源装置ポリシー

**qos** : Quality of Service ポリシー

**qosm** : QoS の統計情報

**qosp** : qos/ 802.1p

**rbqm** : デバッグ

**regress** : 回帰

**reln** : \*\*内部プロセスによって生成され、使用されます\*\*

**repl** : \*\*内部プロセスによって生成され、使用されます\*\*

**res** : \*\*内部プロセスによって生成され、使用されます\*\*

**rib** : ルーティング情報ベース

**rmon** : リモート ネットワーク モニタリング/インターフェイス統計情報/カウンタ

**rpm** : ルート ポリシー マップ

**rtcom** : ルート制御コミュニティ リスト

**rtctrl** : ルート制御

**rttextcom** : ルータの拡張コミュニティ

**rtflt** : ルート フィルタ

**rtleak** : ルート リーク

**rtmap** : RPM ルート マップ

**rtpfx** : ルート プレフィックス リスト

**rtregcom** : ルートの通常のコミュニティ リスト

**rtsum** : ルート集約のアドレス/ポリシー

**satm** : サテライト マネージャ

**snmp** : 簡易ネットワーク管理プロトコル

**span** : スイッチド ポート アナライザ

**stats** : 統計情報収集ポリシー

**statstore** : 統計データ ホルダー

**stormctrl** : ストーム制御 (トラフィック抑制) ポリシー

**stp** : スパニング ツリー プロトコルの定義とポリシー

**sts** : サービス タグ スイッチング (サービス挿入に使用)

**svccore** : コア ポリシー

**svi** : スイッチド仮想インターフェイス/ルーテッド VLAN インターフェイス

**synthetic** : 模擬オブジェクト (テスト用)

**sysdebug** : システム デバッグ

**sysfile** : システム ファイル

**syshist** : システム カードのリセット記録/履歴

**syslog** : syslog ポリシー

**sysmgr** : システム マネージャ (ファームウェア、スーパーバイザ、システム状態など)

**sysmgrp** : すべての QoS ポリシー定義のコア ポリシーおよび抽象クラスのコンテナ

**tag** : エイリアス (dn の記述名に使用)、タグ (複数のオブジェクトを記述名でグループ化)

**task** : タスクの実行、インスタンスおよび結果

**test** : テストルール、サブジェクトおよび結果の抽象クラス

**testinfralab** : テスト インフラストラクチャ

**tlv** : タイプ、長さ、値のシステム構造

**top** : プロセッサ アクティビティのシステム タスク マネージャ

**topoctrl** : トポロジ制御ポリシー (sharding、ファブリック LB、ファブリック VxLan など)

**traceroute** : traceroute の実行と結果

**traceroutep** : traceroute のエンド ポイント

**trig** : トリガー ポリシー

**tunnel** : トンネリング

**uribv4** : ipv4 ユニキャスト ルーティング情報の基本エンティティ

**vlan** : VLAN インスタンス

**vlanmgr** : VLAN マネージャ コントロール プレーン

**vmm** : 仮想マシン マネージャ (コントローラ、vmm ポリシーと定義)

**vns** : 仮想ネットワーク サービス (L4-L7 ポリシーと定義)

**vpc** : 仮想ポート チャンネル (vpc ポリシーと定義)

**vsvc** : サービス ラベル (プロバイダー/コンシューマ)

**vtap** : 外部ノードの変換アドレス (サービス ノードの NATed IP)

**vxlان** : 仮想拡張 LAN の定義

**vz** : 仮想ゾーン (ポリシー制御の以前の名前)、すなわちコントラクト

**モデル命名方式**

**Rs** : 関係のソース

**Rt** : 関係のターゲット

**Ag** : 集約統計

**BrCP** : バイナリコントラクトプロフィール

## 頭字語とその定義

この項は、このマニュアルで使用される用語および概念の概要説明を提供するように設計されています。ACIはパケットを回線に送信する方法を変更していませんが、いくつかの用語と概念が新しく採用されました。これらの新しい用語と概念を理解することは、ACIで機能するそれらが、それらのビットを送信するためにACIで使用される要素について、相互に通信する際に役立ちます。関連する新しい頭字語についても説明します。

これは、すべての用語および概念の完全なリストや詳細なディクショナリを意図するものではなく、共通の専門語の一部でない場合がある主要な用語および概念、または説明済みのトラブルシューティングシナリオでカバーされたトラブルシューティング演習に関連するもののみを示します。

### AAA

Authentication (認証)、Authorization (許可)、および Accounting (アカウントिंग) の頭字語。

### アクセスのカプセル化

EPGは「EPGとしてのVLAN」モデルに従うため、リーフスイッチ上の単一ポートからの複数のEPGがある場合、問題が生じます。ポートで動作させるカプセル化のタイプを選択する場合は、EPGが存在するポート経由で他のEPGがアクセスできるかを確認する必要があります。ポートには、次の3種類のモードがあります。

- **Tagged** (デフォルト) : ホストからのトラフィックにVLAN IDがタグ付けされている場合は、このモードを選択します。このモードでは、ポートから出るすべてのパケット (VLAN 0を含む) が802.1Qヘッダーでタグ付けされます。これにより、ユーザは、どのEPGでネイティブ (タグなし) VLANを使用するかを決定せずに、複数のEPGを追加してポートに接続することができます。
- **Untagged** : ホストからのトラフィックがタグ付けされていない (VLAN IDも802.1Qヘッダーもない) 場合、このモードを選択します。このモードは、EPG内エンドポイントがポートで着信される際、他のEPGからの他のエンドポイントを伴わないことを条件とします。

- 802.1P：ホストからのトラフィックが 802.1P タグでタグ付けされている場合、このモードを選択します。これにより、単一の EPG でそのトラフィックに 802.1Q をタグ付けせずに、単一のポートで複数の EPG を着信することが可能になります。

### ACI 外部接続

エンドポイントがファブリックの管理対象範囲から外れる場合に、外部のルーテッドまたはスイッチド中継システムを使用するファブリックとの間の接続。

### ACID トランザクション

ACID は、Atomicity（アトミック性）、Consistency（整合性）、Isolation（分離）、Durability（耐久性）の頭文字語です。これらは、データベーストランザクションの整合性を確保するトランザクションのプロパティです。ACI クラスタ内の APIC デバイスへのトランザクションでは、データベースの整合性が維持されるように ACID が考慮されます。これは、トランザクションの一部が失敗した場合、トランザクション全体が失敗することを意味します。

### アクション

フィルタされたトラフィックに対して行うアクション。たとえば、許可、拒否、ロギング、マークなどがあります。次のアクションがサポートされます。

- トラフィックの許可（通常のコントラクトのみ）
- トラフィックのマーク（DSCP/CoS）（通常のコントラクトのみ）
- トラフィックのリダイレクト（サービス グラフによる通常のコントラクトのみ）
- トラフィックのコピー（サービス グラフ、SPAN による通常のコントラクトのみ）
- トラフィックのブロック（タブー コントラクトのみ）

トラフィックのロギング

（タブー コントラクトのみ）

### ALE

ACI リーフ エンジン。

### ユニキャスト ゲートウェイ

「ユニキャストルーティング」がオンになっている場合にエンドポイントが存在するリーフスイッチのそれぞれに、同じゲートウェイ IP アドレスを提供します。パーベイシブ SVI は、すべてのリーフ スイッチに分散されたデフォルト ゲートウェイを提供します。すべてのオフのサブネット パケットを、元のリーフ スイッチから直接ルーティングできるようにします。

### APIC

Application Policy Infrastructure Controller は、集約ポリシー管理コントローラのクラスタです。APIC は、ファブリックにポリシーのインテンド状態を設定します。元々は、Insieme Fabric Controller（IFC）と呼ばれていました。

## API

プログラム可能な拡張性に使用するアプリケーションプログラミングインターフェイス。

## アプリケーション セントリック仮想スイッチ (AVS)

元々は、Nexus 1000 が ACI および VXLAN と連動するよう再プログラムされたもの。

## アプリケーション プロファイル

アプリケーションの論理コンポーネントと、そのコンポーネントの通信方法をモデル化するアプリケーションプロファイル管理対象オブジェクト参照を参照するために使用される用語。アプリケーションプロファイルは、アプリケーションを表すのに使用されるキーオブジェクトで、ACIファブリック内の自動インフラストラクチャ管理のアンカーポイントでもあります。

簡潔に言えば、アプリケーションプロファイルとは、さまざまなエンドポイントグループの集合と、それらの間で通信するために必要なポリシーを意味します。各アプリケーションプロファイルには、1つ以上のアプリケーションエンドポイントグループが含まれる場合があります。アプリケーションプロファイルレベルで、その下で定義されているアプリケーションエンドポイントグループの QoS 分類を設定します。アプリケーションエンドポイントグループレベルで、ブリッジドメインを定義します。ネットワーク下でブリッジドメインは VRF にリンクします。

## ASE

ACI スパイン エンジン。

## 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。

## 接続可能エンティティ プロファイル (AEP)

これは、エンティティがファブリックにアタッチする場合に適用されるインターフェイスの設定プロファイルです。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。AEP は、物理ポートをスイッチ ポリシーへのドメイン (物理または仮想) に結合するメカニズムでもあります。AEP は、ドメイン (ドメインは、VLAN プールから VLAN をリンクする) を、MIT 内のスイッチ ポリシーグループ (設定) にリンクします。AEP の 2 つ目の用途として、AEP は、直接的なアクセスなしで管理者により設定された VMM ポリシーにテナントがアクセスできるようにします。

## AV

アプライアンス ベクトル。

## BD-VLAN

BD-VLAN はブリッジドメインを表すために使用され、複数のハードウェア VLAN および内部 VLAN に複数の FD-VLAN (カプセル化 VLAN) をリンクすることができます。こ

これは、Broadcom ASICにより使用される1つの転送方法であり、処理のためにトラフィックをローカルでスイッチするか、またはNorthstar ASICに転送するかを決定するのに使用されます。BD-VLANは、さまざまなローカルFD-VLANを単一のブリッジドメインに接続します。また、Broadcom ASICで、複数のサブネットまたはACCESS\_ENCを含む場合があるレイヤ2ブロードキャストドメインを決定するために使用されます。

### バウンス エントリ

エンドポイントが別のリーフスイッチに移動する際、そのエンドポイントが以前に置かれていたリーフスイッチには、エンドポイントの移動時に送信中のトラフィックのすべてが新しいロケーションのエンドポイントに向かって確実に送信され続けるように、バウンスエントリが設置されます。これは、リーフが他のリーフにパケットを直接転送するものの、そのEPが他のリーフに移動している場合に発生します。中央のリーフは、エンドポイントが実際に存在する新しいリーフにパケットをバウンスします。

### ブリッジドメイン (BD)

各一意のレイヤ2フォワーディングドメイン(フラッドドメイン)のレイヤ2転送動作(ブロードキャスト、ARPフラディングなど)を定義するACI構造。ブリッジドメインは、IPサブネットのコンテナでもあり、ファブリックレイヤ3ゲートウェイの機能が設定される場所です。ブリッジドメインは、従来のVLANの動作をエミュレートできますが、スケール制限の転送では制限されません。ACIオブジェクトモデルでは、ブリッジドメインはプライベートレイヤ3またはコンテキストの子です。エンドポイントグループは、1つのブリッジドメインのメンバーにのみなることができます。MACアドレスは、ブリッジドメインごとに(サブネット間で)一意である必要があります。

### CLOS ファブリック

多階層ノンブロッキングリーフ-スパインアーキテクチャネットワーク。

### クラスタ

同じまたは類似の一連の機能を提供するために、単一システムとして連携する一連のデバイス。通信によりスケラブルで分散可能なコントローラを提供する一連のAPICはクラスタにあたります。MITデータベースの共有にはクラスタサイズが使用されます。クラスタは、コントローラアプリケーションを実行し、相互に通信して、ファブリックの単一の論理ビューを形成する一連のアプライアンスです。

### 物理モデル

物理モデルは、APICで実行されているロジック、およびスイッチで実行されているポリシー要素によりレンダリングされます。スイッチのソフトウェアでは、物理モデルを使用して、サービスに対してスイッチデータプレーンのプログラミングを調整します。これには、設定オブジェクトと管理対象動作オブジェクトが含まれています。ユーザはこのモデルを見ることはできますが設定はできません。

### コンシューマ

コンシューマは、コントラクトにおいて、トラフィックの「送信元」です。EPGがコントラクトを消費すると、消費するEPGのすべてのエンドポイントが、そのコントラクトを提供している任意の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。

## コンテキスト

VRF と同等のレイヤ 3 フォワーディング ドメイン、ACI の用語ではプライベート レイヤ 3。

## コントラクト

エンドポイントグループ間の通信のルールを制御するフィルタに関連するサブジェクトの論理コンテナ。ACI はホワイトリスト ポリシー モデルで動作します。コントラクトがないと、デフォルトの転送ポリシーはエンドポイントグループ間での通信を許可しませんが、エンドポイントグループ内の通信は許可されます。

## コントラクト範囲

コントラクト範囲は、2つ以上 EPG 間でのコントラクトの適用のレベルを意味します。以下の状態があります。

- 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は、同じアプリケーションプロファイル (AEP) 内にある他のエンドポイントグループとのみ通信できます。
- プライベートネットワーク (デフォルト)：エンドポイントグループは、同じプライベートネットワーク (VRF) 内にある他のエンドポイントグループとのみ通信できます。
- テナント：エンドポイントグループは、同じテナント内にある他のエンドポイントグループとのみ通信できます。
- グローバル：エンドポイントグループは、ファブリック全体の中に存在する他のエンドポイントグループと通信できます。

コントラクト範囲は、基本的に、EPG と通信できるホストの範囲を、同じアプリケーションプロファイル内、同じ VRF (プライベートネットワーク) 内、同じテナント内、またはファブリック内であれば場所を問わない、に制限します。

## Council of Oracles プロトコル (COOP)

マッピング情報 (ロケーションおよび ID) をスパインプロキシに伝達するために、COOP を使用します。iLeaf は、ZeroMQ (ゼロメッセージキュー) を使用して、エンドポイントのアドレス情報をスパイン「Oracle」に転送します。スパインノードで実行している COOP によって、すべてのスパインノードが一貫性のあるエンドポイントアドレスとロケーション情報のコピーを維持することができ、さらに、ロケーションマッピングデータベースに対するエンドポイント ID の DHT レポジトリを維持することができます。

## データ管理エンジン (DME)

データモデルのデータを制御する、APIC で実行されるサービス。DME は、通常、要求と応答を含む「stimulus」と呼ばれるメッセージエンティティを使用して通信します。APIC またはスイッチの上の各サービスは、DME と呼ばれるライブラリまたはフレームワークで構築されます。

## DLB

ダイナミック ロード バランシング : Flowlet スイッチングに基づく ACI ファブリックのネットワーク トラフィック ロードバランシング メカニズム。

## dMIT

分散管理情報ツリー。上部にツリーのルートがあり下部にツリーのリーフがある ACI オブジェクト モデルを表します。ツリーには、ACI ファブリックを表すオブジェクト モデルのすべての側面が含まれます。

## Dn

識別名。ACI 管理情報ツリー (MIT) 内の特定の管理対象オブジェクトやツリー内の特定のロケーション情報を表す完全修飾名。これは、それ自体からツリーのルートに戻るすべての相対名の連結で構成されます。例として、タイプがアプリケーションプロファイルのポリシー オブジェクトが Prod という名前のテナント内で commerce workspace という名前で作成された場合、dn は uni/tn-Prod/ap-commerceworkspace と表されます。

## Encap

関連付けられているエンドポイントグループの仮想マシンマネージャ (VMM) がカプセル化されたもの (VLAN または VXLAN)。これは、接続されたエンドポイントを含むエンドポイント グループに割り当てられる VLAN です。

## エンドポイント (EP)

ホストまたはレイヤ 2/レイヤ 3 のアドレス指定可能エンティティとも呼ばれています。ポートに向けたファブリックではないリーフスイッチ上のポートに直接または間接的に接続された論理または物理デバイス。エンドポイントには、アドレス、ロケーションなどの固有のプロパティがあり、エンドポイントの識別に使用されます。アドレス指定可能な MAC エンティティ (仮想 NIC インターフェイス、物理 NIC インターフェイス、スイッチ CPU インターフェイスなど) はすべて、エンドポイントと見なされます。エンドポイントは、エンドポイント グループにまとめられます。

## エンドポイント グループ (EPG)

共通ポリシーに対する共通の要件、または同じポリシー処理を必要とする共通の要件に基づいてグループ化することができるエンドポイントの集合体。エンドポイント グループは、ダイナミックまたはスタティックにできます。ACI ファブリックは通常、802.1Q VLAN タグを使用してエンドポイント グループを表します。これは、多くの場合「EPG としての VLAN」モデルと呼ばれます。次のタイプのエンドポイント グループが定義されています。

- 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

エンドポイント グループのメンバーシップはファブリックで次によって定義されます。



- 入力物理ポート（リーフまたは FEX）
- 入力論理ポート（VM ポート グループ）
- VLAN ID
- VXLAN（VNID）
- IP アドレス（FCS での外部/境界リーフ接続にのみ適用）
- IP プレフィックス/サブネット（FCS での外部/境界リーフ接続にのみ適用）

ポリシーが適用されるのはエンドポイントグループであり、個々のエンドポイントには適用されません。エンドポイントグループは、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。

エンドポイントグループには以下のものが関連付けられます。

- 単一のレイヤ 2 仮想ネットワーク（ブリッジグループ）または単一のレイヤ 3 仮想ネットワーク（VRF またはプライベート ネットワーク）
- 単一のセキュリティグループ（コントラクト）

#### [Error]

エラーは APIC でのみ発生します。エラーには、重複 MO があるイベントや、RADIUS サーバに到達するエラーが記述されます。

#### Ethertype

フィルタ エントリの EtherType。現在、次の EtherType があります。

- 未指定（デフォルト）（すべてのプロトコル）
- ipv4
- ipv6
- lldp
- 8021ad
- 8021q
- arp
- fcoe
- flow\_control
- mac\_ または VMMrity
- mpls\_mcast
- mpls\_ucast
- ptp
- qinq

- rarp
- slow\_protocols
- trill
- wake\_on\_lan

## イベント

イベントは、エラーでもフォールトでもありません。何かが発生した場合に、ステップごとの情報を示します。たとえば、インターフェイスが起動すると、インターフェイスが起動している間に何が起きているかを示すイベントが提供されます。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。

## Fault

フォールトが発生したり、アラームが鳴ると、システムによってフォールトに対するフォールト管理対象オブジェクトが作成されます。フォールトには、影響を受けるオブジェクトの動作状態に関する条件、情報、および問題の潜在的な解決策が含まれています。フォールトは、APIC、ファブリック、およびホストで生成されるエラーです。

## ファブリック

ACI ソリューションと関連付けられているエンドポイントの集合（リーフ、スパインおよび仮想スイッチ + APIC）。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 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。ファブリック内では、転送は VTEP（ACI VXLAN トンネル エンドポイント）間で行われ、ACI VXLAN ポリシー ヘッダーと呼ばれるエクステンダ VXLAN ヘッダー フォーマットが使用されます。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。ファブリックの概要は次のとおりです。

- インフラストラクチャ トポロジのために IS-IS 使用します。IS-IS は、TEP を識別して、各リーフ ノードからファブリック内の他のすべてのノードにトンネルの作成をア ナウンスする役を担います。
- ループバック アドレスおよび VTEP アドレスをアドバタイズします。
- ベンダーの TLV を使用して、ファブリック内でマルチキャスト FTAG ツリーを生成する役を担います。

ファブリック内のすべてのテナントトラフィックは、ファブリック内のアプリケーション エンドポイントのポリシー属性を識別する ACI VXLAN (VXLAN) ヘッダーでタグ付けされます。

- ポリシー グループ（送信元グループ）
- 転送グループ（テナント、VRF、ブリッジ ドメイン）
- ロード バランシング ポリシー

- テレメトリ ポリシー

入力ポートで、ファブリックは内部 VXLAN タギング形式を使用して、異なるアプリケーション エンドポイントを区別するのに使用できる外部識別子を変換します。

ファブリックではカプセル化が使用され、そこでレイヤ 2 イーサネット パケットはカプセル化されてファブリックを通過します。ファブリック全体のデフォルト MTU は 9150 バイトです。

## FCAPS

ISO モデルは、ネットワーク管理タスクを定義します。FCAPS は、管理カテゴリであるフォールト (fault)、構成 (configuration)、アカウンティング (accounting)、パフォーマンス (performance)、セキュリティ (security) の頭字語です

## FD\_VLAN

フラッドドメイン VLAN。FD-VLAN は、Broadcom ASIC 上でトラフィックを転送するのに使用される VLAN です。FD\_VLAN は ACCESS\_ENC に直接リンクされ、内部 VLAN とも呼ばれます。FD\_VLAN は、ACCESS\_ENC を BD\_VLAN に直接リンクするのではなく、ACCESS\_ENC を表すために使用されます。FD\_VLAN を使用すると、BD\_VLAN は、さまざまな ACCESS\_ENC にリンクして、それらが NX-OS スイッチ上の同じ 802.1Q VLAN にあるかのように、それらすべてを処理することができます。ブロードキャスト パケットが ACI ファブリックからリーフ スイッチに着信すると、BD\_VLAN は、パケットが別の ACCESS\_ENC を使用して別のポートから転送されるよう、複数の FD\_VLAN にマッピングすることができます。FD\_VLAN は、レイヤ 2 MAC アドレスを学習するのに使用されます。

## Filters

フィルタは、コントラクトに一致するレイヤ 2 ~ レイヤ 4 フィールドを説明するルールを定義します。すべてのコントラクトは 1 つ以上のサブジェクトから構成され、各サブジェクトには 1 つ以上のフィルタが含まれ、各フィルタには 1 つ以上のエントリが含まれます。各エントリは、ACL 内の 1 行に相当し、EPG 内の EP が接続されているリーフ スイッチで適用されます。フィルタは、エントリがコントラクト内で PERMIT ステートメントなのか DENY ステートメントなのかを定義しません。これは、フィルタが関連付けられているコントラクトタイプによって決定されます。

## Flowlet スイッチング

2004 年の MIT による調査に基づく、最適化されたマルチパス ロードバランシング方法。Flowlet スイッチングは、フローをフローレットに動的に分割し、パケットの再順序付けを必要とすることなく複数の平行パスにわたりトラフィックを分割することで、TCP フローをより効率よく転送するために TCP のバースト特性を使用する方法です。

## FNV

ファブリック ノード ベクトル。

## GUI

グラフィカル ユーザ インターフェイス。

## ヘルス スコア

ヘルススコアは、さまざまなシステムビューに、エラーがあるかどうかを示します。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。

ヘルススコアは、パーセンテージの設定ポイントによって異なる色で表示されます。

- **66%** : この割合を下回ると、ヘルススコアは黄色で表示されます。
- **33%** : この割合を下回ると、ヘルススコアは赤色で表示されます。

## HTML

ハイパーテキストマークアップ言語。Web ページの書式設定に焦点を合せたマークアップ言語。

## ハイパーバイザ

ホストマシン上のハードウェアを抽象化し、ホストマシンが複数の仮想マシンで実行できるようにするソフトウェア。

## ハイパーバイザの統合

APIC に仮想マシンの可視性とポリシーの適用に関するメカニズムを提供するための、仮想マシンマネージャへの ACI ファブリック接続の拡張。

## 内部ファブリックメッセージ (IFM)

ACI ファブリック上のデバイス間の通信に使用されます。各エージェントのさまざまな DME アドレス間メッセージの提供を試行するソフトウェアレイヤは、「アイデンティティ」と呼ばれます。アイデンティティの形式は次のとおりです。

```
system-type:system-id:service:slot
```

たとえば、スイッチ 119 のポリシー要素は次のように表されます。

```
1:119:5:0
```

IFM はリモート接続に SSL over TCP を使用します。次のように、プロセスにより異なるポートが使用されます。

```

eventmgr 12119 Alerts, Faults, Health Scores
nginx    12151
policyelem 12183
policymgr 12215 processes policy addition/deletion/modification events and
the
converts the policy into ppf representation and distributes
policies to policy clients using PPF library. It uses 2-pass
verify-commit model to get the policy programmed in the
hardware.
reader 12247
ae 12279
topomgr 12311
observer 12343 Roll up various HW faults
dbgr 12375 Atomic Counters
observerelem 12407
dbgrelem 12439
vmmgr 12471 VM manager interactions
nxosmock 12503
bootmgr 12535
appliancedirector 12567 Clustering, Sharding, Syncing of replicas
dhcpd 12695 DHCP addressing
scripthandler 12727
idmgr 12759

```

## インバンド管理 (INB)

インバンド管理設定を使用した接続。これは、ファブリックと APIC の外部管理接続にリーフスイッチの前面パネル（データプレーン）ポートを使用します。管理接続にファブリックを使用します。スイッチに 2 番目の VTEP アドレスを適用します。

## iPing

ファブリック認定 ping です。iPing は以下と連動するよう設計されています。

- パーベイシブ ファブリック インターフェイス（各リーフスイッチ上のゲートウェイ IP アドレス）
- マルチテナントとオーバーレイ

通常の ping の問題は、ファブリック上のパーベイシブ IP ゲートウェイでは、ゲートウェイ IP アドレスが複数のリーフスイッチ上にあるため、発信元リーフが ping 応答を受信しない可能性があることです。iPing により、ファブリック情報をペイロードに含めることで、別のリーフスイッチに転送された場合でも、発信側リーフスイッチは ping 応答を受信できます。

## IS-IS

インフラストラクチャ トポロジのためにファブリックで利用されるリンク ローカルルーティングプロトコル。ループバックおよび VTEP アドレスは、IS-IS 上で内部的にアドバタイズされます。IS-IS は、リーフ ノードからファブリック内の他のすべてのノードへのトンネルの作成をアナウンスします。

## iTraceroute

ファブリック認定トレースルート。iTraceroute には次のような固有の機能があります。

- 複数パスを検出およびリポート

- パスあたり 1 つのプローブ パケットのみを転送
- 詳細なノード情報をレポート
- 適用されるポリシーでパスを確認し、テナント トラフィックをシミュレート

## JSON

JavaScript Object Notation。データ オブジェクトを属性と値のペアにカプセル化するために人間が読めるテキストを使用するデータ カプセル化形式。

### レイヤ 2 アウト (I2out)

ACI ファブリックの外部に存在する外部ネットワークへのレイヤ 2 接続。

### レイヤ 3 アウト (I3out)

ACI ファブリックの外部に存在する外部ネットワークへのレイヤ 3 接続。

## L4-L7 サービス挿入

トラフィックのフローへの仮想または物理サービスアプライアンス（ファイアウォール、IDS/IPS、ロードバランサ、DLP など）の VLAN/レイヤ 3 構成の挿入およびスティッチング。サービス ノードは、OSI モデルのレイヤ 4 とレイヤ 7 の間で動作するのに対し、ネットワーク要素（ファブリック）はレイヤ 1～3 で動作します。

## Label

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するために使用されます。

## Leaf

ホストとボーダーの接続を提供するファブリック内のネットワーク ノード。リーフは、ホストおよびスパインのみに接続します。リーフは、相互に接続しません。すべてのデバイスは、外部ルータまたはホストを含め、リーフ スイッチに接続されます。

## レガシー モード

コントラクトは、ブリッジ ドメイン下に作成されると、そのブリッジ ドメインに対して適用されず、このブリッジ ドメイン上のすべての EPG にカプセル化が適用されます。この場合、ブリッジ ドメイン、エンドポイント グループ、および VLAN はすべて、1 対 1 でマッピングされます。

## ライフ サイクル

ライフサイクルには、フォールトの始まりから終わりまでが記述されます。これには、次の状態に移行するまでの特定の経過期間が含まれます。ライフサイクルという概念の背景には、非常に長い期間継続する一時エラーを管理者が確認できるようにすることです。

ライフサイクルには、フォールト MO が、SOAKING、RAISED、または RETENTION の状態である期間が記述されます。

また、ライフサイクルには、重大度（INITIAL、TARGET、または CLEAR）が記述されず。

## リンク層検出プロトコル (LLDP)

LLDP は、APIC によりリーフ スイッチを検出するために使用され、リーフ スイッチによりファブリック検出のためにスパイン スイッチを検出するために使用されます。

## 論理モデル

APIC で、ユーザにより、GUI または XML/JSON API を使用して設定されるモデル。論理モデルは `/aci->/aci/viewfs` の下にあります。

## 管理対象オブジェクト (MO)

MIT で管理される ACI ポリシー モデルのすべての構成可能なコンポーネントは MO と呼ばれます。管理対象オブジェクトは設定や状態を表します。管理対象オブジェクトおよびプロパティには、読み取り/書き込み可能と、読み取り専用があります。名前付き関係におけるポリシー解決ベースの場合、次のように動作します。

- 一致する名前を持つターゲット管理対象オブジェクトが現在のテナントで検出されない場合、ACI ファブリックは、共通のテナントでその名前の解決を試行します。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性管理対象オブジェクトを含んでいた場合、システムは共通のテナントでその関係性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。
- 現在のテナントに存在しない場合、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。

注：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。

## 管理情報ツリー (MIT)

ファブリックの管理対象オブジェクトのすべてを含む階層型管理情報ツリー。管理情報モデル (MIM) と呼ばれます。

## Match Type

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。プロバイダー ラベルとコンシューマ ラベルの一致を確認する場合、設定はプロバイダー EPG によって決定されます。次のタイプがあります。

- **AtleastOne** (デフォルト) : 少なくとも 1 つのラベルが、プロバイダー EPG とコンシューマ EPG で一致する。空白のラベルは一致と見なされます。
- **AtmostOne** : EPG 上のすべてのラベルがまったく同じ場合にのみ一致する。空白のラベルは一致と見なされます。
- **None** : サブジェクト ラベルのいずれも一致しない。
- **All** : 両 EPG に空白のラベル以外のすべてのラベルがある場合にのみ一致する。

## モデル

モデルは、エンティティとエンティティ間に存在する関係性を表す概念です。

## 多階層アプリケーション

プレゼンテーション機能、アプリケーションロジック機能およびデータベース管理機能には物理的または論理的な分離が必要で、アプリケーション機能のために他の階層と通信するためにネットワーキング機能を必要とする、クライアント - サーバアーキテクチャ。

## NginX

NginX（「エンジン X」と読む）は、Igor Sysoev が最初に記述した、HTTP リバース プロキシサーバ、メールプロキシサーバ、および汎用 TCP プロキシサーバです。APIC のすべての入力方法（REST API、Web GUI、および CLI）で、入力情報は NginX に送信されます。なかでも GUI は、NginX からステータスコードを取得し、コマンドを受信するスイッチによるコマンド検証を行いません。コマンドが適用されない場合、フォールトはスイッチから送信し直されます。設定を適用した後は、[Faults] タブを必ず確認してください。

## Northstar ASIC

Cisco ACI の ASIC。24 x 40G ポートのワイヤスピード。VXLAN セグメント間ルーティング、大容量バッファ、ポリシー適用、高度なダイナミックロードバランシング、ブリッジドメイン内 EPG 間のマルチキャスト転送とブリッジング、QoS、アトミックカウンタ、および遅延測定を提供します。また、NorthStar は、このような付加価値機能の一部をローカルトラフィックおよびアップリンクに適用できるようにするループバックパスを提供します。前面ポートからファブリックを出力する際、NorthStar は次の機能を提供します。

- VxLAN 終端
- ステーション ルックアップ
- ポリシー ルックアップ
- 出力ポートの選択

前面ポートからファブリックに入力する際、NorthStar は次の機能を提供します。

- EPG の取得
- ステーション ルックアップ
- ポリシー ルックアップ
- カプセル化（プロキシおよび非プロキシ）
- Bounce

## オブジェクトモデル

オブジェクトとクラスのコレクションは、そのオブジェクトモデルを公開するシステムの設定および実行状態を確認し、処理するのに使用されます。ACI では、オブジェクトモデルは分散管理情報ツリー（dMIT）と呼ばれるツリーとして表されます。



## OpFlex

OpFlex は、Cisco APIC と任意のデバイス（ハイパーバイザ スイッチ、物理スイッチ、およびレイヤ 4～レイヤ 7 ネットワーク サービスなど）のネットワーク ポリシー コントローラ間で、XML または JavaScript オブジェクト表記（JSON）で抽象ポリシーを転送するためのオープンで拡張可能なポリシー プロトコルです。

## アウトオブバンド管理（OOB 管理）

すべてのスイッチおよび APIC で特定のアウトオブバンド管理インターフェイスを使用した外部接続。

## オーバーレイ - 1

ファブリック スイッチおよび APIC が使用する VRF（コンテキスト）。

## 物理的なドメイン

ベア メタル ホスト用に、または APIC に VMM が定義されずに接続されている場合に使用されるエンドポイント。

## ポリシー制御（PC）タグ

ポリシー制御タグは、名前の代わりに数値で識別するためにエンドポイントグループに割り当てられます。「送信元 ID」または「宛先 ID」と呼ばれています。

## ポート チャネル

複数の物理インターフェイスを単一の論理インターフェイスにバインドし、トポロジを変更することなく集約帯域幅とリンクフォールトのリカバリをさらに提供するポートリンク集約技術。

## Prefix

プレフィックスは、どの外部サブネットがファブリック内に許可されるかを制御します。ファブリックが外部ルータによりピアリングされている場合、サブネットプレフィックスはエンドポイントであると見なされます。

## プライベート ネットワーク

VRF と同様、ルーティング インスタンスを分離します。管理用の分離として使用できません。テナント内の非重複 IP アドレス空間を定義するために使用されます。エンドポイントグループは、1 つのプライベート ネットワークにのみ関連付けることができます。

## Provider

プロバイダーは、コントラクトにおいて、トラフィックの「受信元」です。

## 特権

権限は、ロールが実行できるタスクを決定します。

## ポリシー（Policy）

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

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。

### ポリシー グループ

ポリシーおよびプロファイルのグループ。

### Raised

フォールトの状態がソーキング間隔後にも続く場合、フォールト MO は [Raised] 状態になります。[Initial] 重大度は、[Target] 重大度に進みます。1 つの状態として [RAISED-CLEARING] があり、これは、フォールトがソーキング間隔中、そして発生期間になってもアクティブであったものの、現在はクリアされたことを意味します。

### 解決されたモデル

解決されたモデルは、論理モデルから物理モデルへのマッピングを定義します（論理モデルは [intended] 状態を含み、物理モデルは [actual] 状態を含みます）。これは、APIC 上で発生します。ユーザはこのモデルを見ることはできますが設定はできません。解決されたモデルは、`/mit ->/ .aci/mitfs` で APIC により暗黙的に設定されます。

### Retaining

[Raised-Clearing] または [Soaking-Clearing] のいずれかの状態のクリア期間にフォールトの状態がない場合、フォールト MO は重大度レベルが [cleared] の [Retaining] 状態になります。保持期間が開始され、その間フォールト MO はフォールトポリシーで指定した期間、保持されます。この期間は、障害が発生した状態が緩和された場合でも管理者が障害に気付くようにしたり、障害が早々に削除されないようにするためのものです。

### Role Based Access Control (RBAC)

ロールをユーザに割り当て、それらのロールをデバイス、オブジェクトおよび権限レベルへのアクセス権の付与または拒否のプロセスで使用することで、インフラストラクチャへのセキュアなアクセスを管理する方法。RBAC 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（テナント）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（テナント）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。

### Representational State Transfer (REST)

ステートレスなプロトコルは通常、API にアクセスするホスト用のローカルクライアントを書き込む必要なしに、サーバ側またはクラウドベースの API へのアクセスをクライアントに許可する HTTP 上で動作します。クライアントがアクセスする場所は通常、クライアントがサービスからアクセスをしようとするデータを定義します。データは通常、XML または JSON 形式でアクセスされ、返されます。ACI を設定する方法はどれも、REST API につながります。CLI、GUI、Python (cobra) API、および XML API では、設定に REST API を使用します。

### RESTful

REST、つまり Representational State Transfer を使用する API。

## Rn

相対名。完全修飾されていない ACI 管理情報ツリー内の特定のオブジェクトの名前。Rn は個々のオブジェクトに重要ですが、コンテキストなしでは、ナビゲーションにおいてあまり有用ではありません。Rn は、識別名を作成するためにそれ自体からルートまで戻るすべての相対名と連結する必要があり、そうすることでナビゲーションにおいて有用になります。例として、アプリケーションプロファイルのオブジェクトが「commercespace」という名前で作成されると、アプリケーションプロファイルの相対名はすべて「ap-」の文字で始まるので Rn は「ap-commercespace」となります。Dn の定義も参照してください。

## ロール

各ロールには特権が割り当てられています。特権を使用すると、管理者は、アクセス制限に関してテナント内の精度をより高めることができます。

## RV

レプリカ ベクトル。

## セキュリティ ドメイン

セキュリティ ドメインは、ユーザを、特定のテナントまたはそのテナントまたは VMM 下の MIT に制限します。

```
Security Domain (per tenant or VMM)
|-----|-----|
Role1          Role 2          role 3
priv1          priv3          priv1
priv2          priv4          priv4
```

## セグメント ID

プライベート ネットワーク名の数値表現。

## サービス グラフ

定義されたパラメータに基づいてトラフィックと VLAN スティッチングのリダイレクションを自動化する ACI 内のメカニズム。必要とされるすべてのサービスが、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。サービス グラフは、アプリケーションに必要なネットワーク機能またはサービス機能のセットを識別し、各機能をノードとして表します。

## モジュール

モジュールはデータベースの一部です。モジュールにより、複数の物理デバイス間でデータベースを複製できます。各モジュールには、3 つの APIC にわたる 3 つのレプリカがあります。各 APIC はデータベースの完全なコピーを持ちますが、APIC がアップデートできるのは、割り当てられたデータベースの一部（またはモジュール）のみです。APIC に格納されている他のモジュールは、モジュールを所有する APIC が更新通知するまでは読み取り専用です。APIC クラスタで APIC が損失した場合、そのクラスタでまだアクティブな別の APIC がデータベース内のそのモジュールを制御することが想定されます。

## Soaking

ソーキングは、フォールトの状態が検出されたときに最初に作成される時間間隔を記述します。その initial 重大度が設定されます。1つの状態として SOAKING-CLEARING があり、これは、ソーキング間隔の間にフォールトがクリアされたことを示します。

## スパイン

リーフから集約ホストトラフィックを伝送するファブリック内のネットワークノード。ファブリック内のリーフのみに接続し、他のデバイスタイプには接続しません。

## スパイン リーフ トポロジ

スパインノードがリーフノードに接続し、リーフノードがホストと外部ネットワークに接続する、clos ベースのファブリックトポロジ。

## スタティック バインディング パス

通常の場合では、ACI は自動的にリーフスイッチ上に EP (EPG 内) トラフィックを入れるよう VLAN を生成しています。スタティックパスバインディングでは、リーフスイッチ上のポートを EPG に対して設定されるよう VLAN を割り当てる必要があります。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 は定義済みのスタティック VLAN プールの一部である必要があります。これは「EPG としての VLAN」モデルに従うものです。EP をスイッチに導入したりポートを起動したりせずに、エニーキャストゲートウェイをリーフスイッチで表示するようするには、この方法が容易です。

## サブジェクト

コントラクトによって含まれており、フィルタとコントラクト間の関係を作成します。1つのコントラクトに複数のサブジェクトが含まれている場合があります。サブジェクトは、コントラクトに適用されるとテンプレートのように機能し、EPG に一方向的に渡す必要がある特定のサービスを提供します。サブジェクトは、フィルタ、アクション、および (任意で) ラベルの組み合わせです。

Filter	Action	Label
TCP Port 80	Permit	Web Access
Any	Permit	common/default

[Apply Both Directions] が選択されている場合、それぞれの EPG は別の EPG に 1 方向のトラフィックを送信できます。[Reverse Filter Ports] がオンになっていない場合、TCP 内の ACK を含むすべてのリターントラフィックはブロックされます。[Reverse Filter Ports] を有効にするには、[Apply Both Directions] を最初にチェックする必要があります。[Reverse Filter Ports] がオンになっていても、コントラクトのコンシューマだけがトラフィックを開始できます。

## サブネット

ブリッジドメインまたはエンドポイントグループによって含まれており、サブネットはブリッジドメイン内で使用できる IP アドレス範囲を定義します。サブネットはブリッジドメインで定義され、OpenStack の「サブネット」と同じです。サブネットのタイプは次のとおりです。

- **Shared** : エンドポイントグループ下のサブネットを、共有サービスを別の VRF 内のエンドポイントグループにルートリークするよう定義します。
- **Public** : エンドポイントグループ下のサブネットを、ファブリックの他のテナントにルートリークするか、またはファブリック外に外部からアドバタイズされるよう定義します。たとえば、ブリッジドメイン設定のもとでは、パブリックサブネットはルーティングプロトコル経由で外部ネットワークにアナウンスされます。
- **Private** : (デフォルト) ブリッジドメイン下のサブネットを、そのテナン内でのみ使用されるよう定義します。つまり、サブネットは他のテナントにリークされません。

## スーパーバイザ

95xx スイッチのコントロールプレーンを提供するスイッチモジュール。

## tDN

ターゲット DN (識別名)。明示的なリファレンスにより、送信元 MO (管理対象オブジェクト) とターゲット MO の特定のインスタンス間の関係性が定義されます。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

アプリケーションポリシーのすべてのポリシーをグループ化する論理コンテナ。各テナントには、個別のレイヤ 3 アドレス空間または VRF (プライベートネットワーク) があります。テナントは OpenStack の「プロジェクト/テナント」の概念と似ています。ACI には次の 3 種類の事前定義されたテナントがあります。

- **Common** : すべてのテナントで使用できる ACI 構造を定義します。
- **mgmt** : APIC、リーフ、およびスパイン型スイッチのインバンドおよびアウトオブバンド管理を設定します。
- **infra** : スパインとリーフ間のファブリックポリシーを設定します。

テナントは、管理情報ツリー (MIT) 内で最高レベルの管理対象オブジェクトです。テナントが含む主要な要素は、コントラクト、ブリッジドメイン、プライベートネットワーク、およびエンドポイントグループを含むアプリケーションプロファイルです。

## 仮想 eXtensible LAN (VXLAN)

レイヤ 3 ネットワーク上のレイヤ 2 オーバーレイ方式。トラフィックの分離/セグメント化のため、VLAN 単独で達成可能な 4K セグメントと異なり、24 ビット VXLAN セグメント ID または VXLAN ネットワーク ID (VNI) がカプセル化に含められて、最大 16M の VXLAN セグメントを提供します。これらのセグメントのそれぞれは固有のレイヤ 2 ブロードキャストドメインを表し、指定したテナントのアドレス空間またはサブネットを特定することが可能な方法で管理されます。VXLAN は、レイヤ 2 LISP プロトコル

(draft-smith-lisp-layer2-01) を拡張したもので、ポリシーグループ、負荷およびパスメトリック、カウンタと入力ポート、およびカプセル化の情報が追加されています。

## 仮想ネットワーク ID (VNID)

さまざまなケースで、様々な方法でパケットを転送するために使用されます。

- ケース 1 : レイヤ 2、VLAN dot1q、または VXLAN VNID が「ブリッジドメイン」を識別します。
- ケース 2 : VRF-Lite、VLAN dot1q タグが、VRF (プライベート ネットワーク) を識別します。
- ケース 3 : mBGP EVPN、VXLAN VNID が VRF を識別するか、またはレイヤ 2 の場合はブリッジドメインを識別します。

ACI の場合は、外部の VLAN/VXLAN が EPG を識別します。内部では、VXLAN ヘッダー内の S-Class が、EPG、VNID、および VRF/ブリッジドメインを識別します。

## 仮想ルーティングおよびフォワーディング (VRF)

複数のコンテキストを単一のデバイスまたはインフラストラクチャに展開できるようにするレイヤ 3 名前空間分離の方法論。

## 仮想ゾーン (vZ)

管理情報ツリー内のコントラクトは、たとえば vzAny コントラクトなどのように最初に vZ が付きます。vzAny は、VRF 内のすべての IP アドレスがこの EPG に到達でき、しかも不明のプレフィックスを含んでいることを意味します。

## 仮想化

ハードウェアリソースを仮想表現に抽象化するために使用され、ソフトウェアの設定可能性を可能にする技術。

## Visore

解決されたモデルを表示するために APIC に組み込まれたアプリケーション。MIT の設定を確認するために使用されます。Visore にアクセスするには、次の URL を使用します。

[https://your\\_apic/visore.html](https://your_apic/visore.html)

## VNI

VXLAN ネットワーク ID

## vPC

仮想ポート チャンネル。このチャンネルでは、ポート チャンネルがリンク集約のために作成されますが、2 つの物理スイッチ (それ以上でもそれ以下でもない) にわたって展開されません。

## VXLAN トンネル エンドポイント (VTEP)

各ファブリックスイッチと APIC には、スタートアップスクリプトで指定されたサブネットの VTEP が割り当てられます。

## XML

eXtensible Markup Language。ドキュメントのデータの書式設定よりもドキュメントのエンコーディング データに焦点を置いたマークアップ言語。

### ゾーンおよびゾーン分割ルール

EPGのコントラクトは、ゾーンまたはゾーン分割ルール内のルールの形式で、スイッチ対してプッシュされます。各ゾーン分割ルールは、NX-OS ACLの1行に相当します。DME/PE (Policy Elements) は、リーフスイッチに接続された EP が関連付けられている EPG によって提供されるかまたは消費されるコントラクトに基づき、リーフスイッチにゾーン分割設定をプッシュします。相互に通信する2つの EPG は、1つのゾーン内にグループ化されます。EPG が複数の EPG と通信している場合、その EPG を複数のゾーンと関連付けることができます。コントラクトとルールに対するポリシーの連携およびフローは次のとおりです。

1. APIC のポリシー マネージャは、スイッチのポリシー要素マネージャと通信します
2. スイッチのポリシー要素マネージャは、スイッチのオブジェクトストアをプログラムします。
3. スイッチのポリシーマネージャは、スイッチの ACLQOS クライアントと通信します。
4. ACLQOS クライアントはハードウェア (TCAM) をプログラミングします。

ゾーニングルールはコントラクトに関し、次のように指定されます。

- 範囲 (アプリケーション、プライベート ネットワーク、テナント、グローバル)
- 送信元の クラス ID または SRC EPG
- 宛先のクラス ID または DST EPG
- Filter

ルールは、EPGに適用されるコントラクトから取得されます。次の2種類のルールがあります。

- **actrlRules** : EPG/ブリッジドメインとして PE によってプログラミングされたルール。リーフ スイッチに展開されます。
- **MgmtRules** : スイッチのスーパーバイザにヒットするよう指定された、トラフィックのルール。

リーフ スイッチでは、次のコマンドを含むゾーン分割ルールがあります。

```
show zoning-rule [src-epg sclassID] [dst-epg dclassID]
```

クラス ID 中の「0」は、「any」を示すために使用されています。

次のコマンドは、ゾーン中のどのルールがパケットによりヒットされているかを示します。

```
show logging ip access-list internal packet-log
```

## 参考資料

このオペレーションガイドの範囲外のトピックについては、他の場所でドキュメント化されている場合があります。ここでは、有益なその他の参照マニュアルへのリンクを提示します。このリンクからマニュアルを表示して読むことができます。

### ACI のインストールおよびアップグレードのガイド

<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-installation-guides-list.html>

### ACI 入門 - ファブリックの初期化

<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-installation-and-configuration-guides-list.html>

### ACI 設計ガイド

[http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731960.html#\\_Toc405844675](http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731960.html#_Toc405844675)

### ACI トラブルシューティング ガイド

<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-troubleshooting-guides-list.html>

<https://datacenter.github.io/aci-troubleshooting-book/>

### ACI ホワイト ペーパー

<http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/white-paper-listing.html>

### ACI 導入事例

<http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/customer-case-study-listing.html>

### ACI のデモ、プレゼンテーションおよびトレーニング

<http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/sales-resources-list.html>

### ACI エコシステム互換性リスト

<http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/solution-overview-c22-732445.html>

### ACI パートナーおよびカスタマーの提示

<http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/presentations-listings.html>

### ACI と Microsoft SCVMM の統合ワークフロー

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/virtualization/workflow/cisco\\_aci\\_microsoft\\_scvmm\\_workflow.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/virtualization/workflow/cisco_aci_microsoft_scvmm_workflow.html)



### ACI ソリューションの概要

<http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/solution-overview-listing.html>

### ACI ツールキット

<http://datacenter.github.io/acitoolkit/>

### ACI 互換性ツール

<http://www.cisco.com/web/techdoc/aci/acimatrix/matrix.html>

### AVS 構成および拡張性ガイド

<http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/products-installation-and-configuration-guides-list.html>

### AVS トポロジおよびソリューション ガイド

<http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/products-technical-reference-list.html>

### APIC コマンドライン インターフェイスのユーザ ガイド

<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-command-reference-list.html>

### APIC Layer 4 to Layer 7 Services Deployment Guide

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/L4-L7\\_Services\\_Deployment/guide/b\\_L4L7\\_Deploy.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/L4-L7_Services_Deployment/guide/b_L4L7_Deploy.html)

### Cobra のドキュメント

<http://cobra.readthedocs.org/en/latest/>

### Cobra GitHub

<http://github.com/datacenter/cobra>

### Connecting ACI to Outside Layer 2 and 3 Networks

<http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c07-732033.html>

### ファブリック接続のビデオ

[https://www.youtube.com/watch?v=\\_iQvoC9zQ\\_A](https://www.youtube.com/watch?v=_iQvoC9zQ_A)

### Nexus CLI から Cisco APIC へのマッピング

<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-configuration-examples-list.html>

### POSTman

<http://www.getpostman.com>

### サポートされる SNMP MIB のリスト

<http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html>

