Cisco Cloud Network Controller Policy Model

- About the CCNC Policy Model, on page 1
- Policy Model Key Characteristics, on page 1
- Logical Constructs, on page 2
- The CCNC Policy Management Information Model, on page 3
- Tenants, on page 5
- Cloud Context Profile, on page 8
- VRFs, on page 8
- Cloud Application Profiles, on page 9
- Cloud Endpoint Groups, on page 10
- Contracts, on page 12
- About the Cloud Template, on page 14
- Managed Object Relations and Policy Resolution, on page 16
- Default Policies, on page 17

About the CCNC Policy Model

The Cisco Cloud Network Controller (CCNC) policy model enables the specification of application requirements policies. The Cisco Cloud Network Controller automatically renders policies in the cloud infrastructure. When you or a process initiates an administrative change to an object in the cloud infrastructure, the Cisco Cloud Network Controller first applies that change to the policy model. This policy model change then triggers a change to the actual managed item. This approach is called a model-driven framework.

Policy Model Key Characteristics

Key characteristics of the policy model include the following:

- As a model-driven architecture, the software maintains a complete representation of the administrative and operational state of the system (the model). The model applies uniformly to cloud infrastructure, services, system behaviors, and virtual devices attached to the network.
- The logical and concrete domains are separated; the logical configurations are rendered into concrete configurations by applying the policies in relation to the available resources. No configuration is carried out against concrete entities. Concrete entities are configured implicitly as a side effect of the changes to the Cisco Cloud policy model.

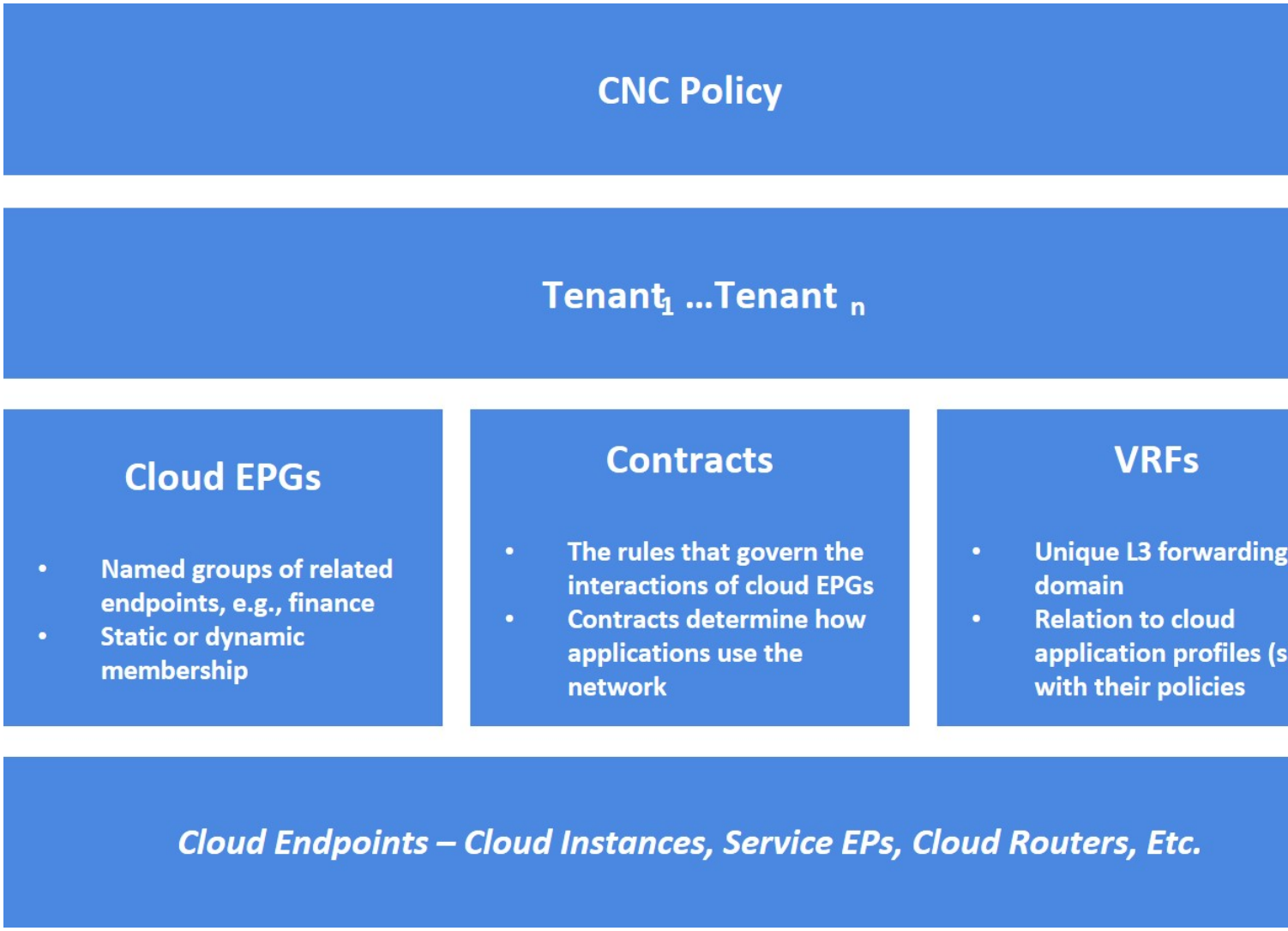
- The system prohibits communications with newly connected endpoints until the policy model is updated to include the new endpoint.
- Network administrators do not configure logical system resources directly. Instead, they define logical (hardware-independent) configurations and the Cisco Cloud Network Controller policies that control different aspects of the system behavior.

Managed object manipulation in the model relieves engineers from the task of administering isolated, individual component configurations. These characteristics enable automation and flexible workload provisioning that can locate any workload anywhere in the infrastructure. Network-attached services can be easily deployed, and the Cisco Cloud Network Controller provides an automation framework to manage the lifecycle of those network-attached services.

Logical Constructs

The policy model manages the entire cloud infrastructure, including the infrastructure, authentication, security, services, applications, cloud infrastructure, and diagnostics. Logical constructs in the policy model define how the cloud infrastructure meets the needs of any of the functions of the cloud infrastructure. The following figure provides an overview of the CCNC policy model logical constructs.

Figure 1: CCNC Policy Model Logical Constructs Overview



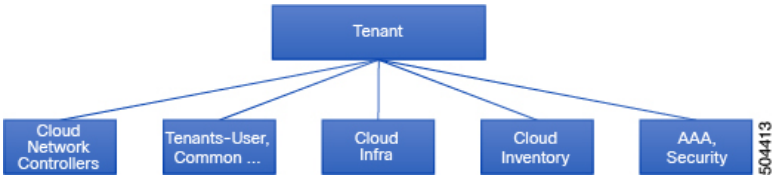
Certain administrators (tenant or cloud infrastructure-wide) create predefined policies that contain application or shared resource requirements. These policies automate the provisioning of applications, network-attached services, security policies, and tenant subnets, which puts administrators in the position of approaching the resource pool in terms of applications rather than infrastructure building blocks. The application needs to drive the networking behavior, not the other way around.

The CCNC Policy Management Information Model

The cloud infrastructure comprises the logical components as recorded in the Management Information Model (MIM), which can be represented in a hierarchical management information tree (MIT). The Cisco Cloud Network Controller runs processes that store and manage the information model. Similar to the OSI Common Management Information Protocol (CMIP) and other X.500 variants, the Cisco Cloud Network Controller enables the control of managed resources by presenting their manageable characteristics as object properties that can be inherited according to the location of the object within the hierarchical structure of the MIT.

Each node in the tree represents a managed object (MO) or group of objects. MOs are abstractions of cloud infrastructure resources. An MO can represent a concrete object, such as a cloud router, adapter, or a logical object, such as an application profile, cloud endpoint group, or fault. The following figure provides an overview of the MIT.

Figure 2: CCNC Policy Management Information Model Overview



The hierarchical structure starts with the policy universe at the top (Root) and contains parent and child nodes. Each node in the tree is an MO and each object in the cloud infrastructure has a unique distinguished name (DN) that describes the object and locates its place in the tree.

The following managed objects contain the policies that govern the operation of the system:

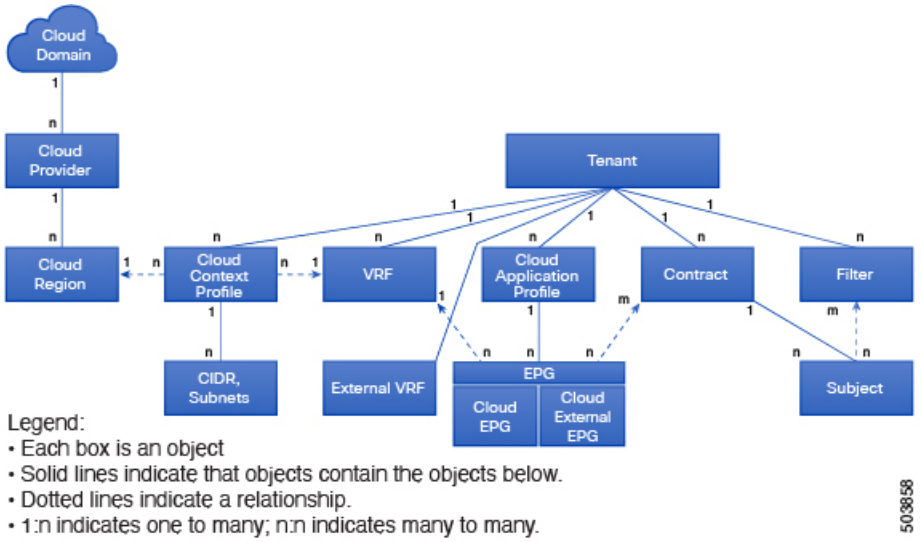
- A tenant is a container for policies that enable an administrator to exercise role-based access control. The system provides the following four kinds of tenants:
  - The administrator defines user tenants according to the needs of users. They contain policies that govern the operation of resources such as applications, databases, web servers, network-attached storage, virtual machines, and so on.
  - Although the system provides the common tenant, it can be configured by the cloud infrastructure administrator. It contains policies that govern the operation of resources accessible to all tenants, such as firewalls, load balancers, intrusion detection appliances, and so on.
  - The infrastructure tenant is provided by the system but can be configured by the cloud infrastructure administrator. It contains policies that govern the operation of infrastructure resources. It also enables a cloud infrastructure provider to selectively deploy resources to one or more user tenants. Infrastructure tenant policies are configurable by the cloud infrastructure administrator.
- The cloud infra policies enable you to manage on-premises and inter-region connectivity when setting up the Cisco Cloud Network Controller. For more information, see the *Cisco Cloud Network Controller Installation Guide*.
- Cloud inventory is a service that enables you to view different aspects of the system using the GUI. For example, you can view the regions that are deployed from the aspect of an application or the applications that are deployed from the aspect of a region. You can use this information for cloud resource planning and troubleshooting.
- Access, authentication, and accounting (AAA) policies govern user privileges, roles, and security domains of the Cisco Cloud ACI cloud infrastructure. For more information, see Cisco Cloud Network Controller Security

The hierarchical policy model fits well with the REST API interface. When invoked, the API reads from or writes to objects in the MIT. URLs map directly into distinguished names that identify objects in the MIT. Any data in the MIT can be described as a self-contained structured tree text document encoded in XML or JSON.

Tenants

A tenant (`fvTenant`) is a logical container for application policies that enable an administrator to exercise domain-based access control. A tenant represents a unit of isolation from a policy perspective, but it does not represent a private network. Tenants can represent a customer in a service provider setting, an organization or domain in an enterprise setting, or just a convenient grouping of policies. The following figure provides an overview of the tenant portion of the management information tree (MIT).

Figure 3: Tenants



Tenants can be isolated from one another or can share resources. The primary elements that the tenant contains are filters, contracts, Virtual Routing and Forwarding (VRF) instances, cloud context profiles, Google Cloud provider configurations, and cloud application profiles that contain cloud endpoint groups (cloud EPGs). Entities in the tenant inherit its policies. VRFs are also known as contexts; each VRF can be associated with multiple cloud context profiles. A cloud context profile, in conjunction with a VRF, tenant and region, represents a resource group in Google Cloud. A VPC is created inside the resource group based on the VRF name.

Tenants are logical containers for application policies. The cloud infrastructure can contain multiple tenants. The CCNC cloud infrastructure supports only IPv4 configurations for tenant networking.

Support for Multiple Cloud Accounts Under a Single Tenant

Beginning with 26.0(2), multiple cloud projects can be associated to a given tenant and deploy different cloud resources in multiple google cloud projects. Different VPCs can also be deployed in different projects under the same VRF for a given tenant.

For example, if you have only cloud deployments where cloud resources have to be deployed in different cloud projects, you can now create a tenant that has multiple projects and then have VPCs point to the respective cloud projects.

Note Multi-Account tenant is only suppported on cloud deployments. This is not supported on configurations deployed in Nexus Dashboard Orchestrator.

Support for Inter-Tenant Shared Services in Hybrid Cloud Environments

In Cisco APIC, a pre-defined tenant (the tenant `common`) is available to provide common services to all tenants, such as shared L3Out, private networks, DNS, DHCP, and Active directory. Prior to release 26.0(3), endpoints on an on-premises ACI tenant and endpoints in a user tenant using networking resources from the on-premises tenant `common` cannot communicate with endpoints on the cloud user tenant. Beginning with release 26.0(3), support is now available for inter-tenant shared services between the on-premises tenant `common` and cloud user tenants.

Cisco Cloud Network Controller, used in conjunction with Nexus Dashboard Orchestrator, supports inter-tenant shared services in a hybrid cloud environment, allowing you to deploy resources in on-premises tenants and cloud tenants, where contracts are deployed in tenant `common`. The tenant `common` still exists on the Cloud Network Controller; however, it is not associated with any cloud account. It is just used for storing filters and contracts that later can be used for a shared service policy. Beginning with release 26.0(3), support is available for having resources in the on-premises Cisco APIC tenant `common` for both Application EPGs and external EPGs, as well as having inter-tenant shared services in a hybrid cloud environment.

For example, assume that you already have an on-premises Cisco APIC tenant `common` deployed with a VRF. You can have bridge domain or EPG in the tenant `common` as you normally would, or you can now create a new user tenant to leverage the VRF and bridge domain in the tenant `common`.

Prior to release 26.0(3), the following variants of standard tenant are supported:

- Regular EPG in a user tenant to a cloud tenant
- External EPG in a user tenant to a cloud tenant

With this update in release 26.0(3), the following variants of the on-premises ACI tenant `common` are also supported:

- Regular EPG in the tenant `common` to a cloud tenant
- External EPG in the tenant `common` to a cloud tenant
- Regular EPG in a user tenant with a bridge domain and VRF in the tenant `common` to a cloud tenant
- External EPG in a user tenant with a VRF in the tenant `common` to a cloud tenant

Use Cases

This section describes several use case examples related to the support for inter-tenant shared services in hybrid cloud environments in release 26.0(3).

On-Premises Cisco APIC Tenant Common Use Case

In this use case, an on-premises Cisco APIC tenant `common` is deployed with either or both of these configurations:

- Application EPGs in the bridge domain or subnet

- External EPG subnet in the L3Out

There is also a contract configured with a user tenant in a cloud site.

The user tenant in the cloud site can be stretched to all the sites, including the on-premises and other cloud sites, and traffic will still flow between the on-premises tenant `common` and the user tenant across all sites.

Site1: On-Premises Site	Site2: Cloud Site
VRF in tenant `common` in Site1: VRF1	VRF in tenant in Site2: VRF2
EPG in Site1: EPG1	EPG in Site2: EPG2
Tenant in Site2 stretched to Site1	Tenant `common` in Site1 available in Site2
External EPG available in VRF1 in tenant `common`	External EPG can be created on Site1

Site User Tenants Use Case

In this use case, a tenant (Tenant1) is deployed only in Site1, which is either an on-premises site or a cloud site, and another tenant (Tenant2) is deployed only in Site2, which is a cloud site, and a contract is shared across tenants.

Site1: On-Premises or Cloud Site	Site2: Cloud Site
VRF in tenant (Tenant1) in Site1: VRF1	VRF in tenant (Tenant2) in Site2: VRF2
EPG in Site1: EPG1	EPG in Site2: EPG2
Tenant2 in Site2 stretched to Site1	Tenant1 in Site1 stretched to Site2
External EPG available in VRF1 in Tenant1	External EPG can be created on Site1

Example Configuration Process

The following general steps provide an example for configuring inter-tenant shared services in hybrid cloud environments. See the Nexus Dashboard Orchestrator documentation for more details.

1. Define the tenants, if necessary.

   In this example scenario, two tenants need to be defined:

   - Cloud only tenant that is associated with a cloud account
   - On-premises `common` tenant, which is already defined through APIC and exists in both the on-premises ACI and the cloud by default

2. Define the tenant templates in Nexus Dashboard Orchestrator (NDO) that are associated with the two tenants.

   In this example scenario, you will define two tenant templates in NDO:

   - `cloud-tenant-template`: Tenant template that is associated with the cloud only tenant
   - `common-tenant-template`: Tenant template that is associated with the on-premises `common` tenant

3. Create a schema (for example, `common-schema`) with the necessary templates.

You can have multiple templates within a schema. For example, you could create two templates within this schema:

- `common-policy`: In this example scenario, we will make the following configurations in this template:
  - We will associate this template with the `common` tenant in the cloud site. This template is to deploy the contracts and filter to the `common` tenant on the cloud (though the `common` tenant is not associated with any cloud account) and the `common` tenant on the on-premises ACI site.
  - We will also create two contracts in this template:
    - One for the external EPG from the on-premises site to the cloud site
    - One for a regular EPG from the on-premises site to the cloud site
  - We will also configure the necessary policy contract and filters within this template.
- `common-app`: In this example scenario, we will associate this template only with the tenant `common` in the on-premises site, and we will make the necessary configurations with this on-premises site, such as configurations related to an application profile, VRF, bridge domain, L3Out, external EPG, and so on.

4. Create a second schema (for example, `cloud-schema`) with a single template (`cloud-only`), where we will associate this template only with the cloud only tenant, and we will make the necessary configurations with this cloud site, such as configuruations related to an application profile, VNet/vPC, and so on.
5. Configure contracts using the contracts that you defined when you created the schemas.
6. Deploy the configurations in NDO.

Cloud Context Profile

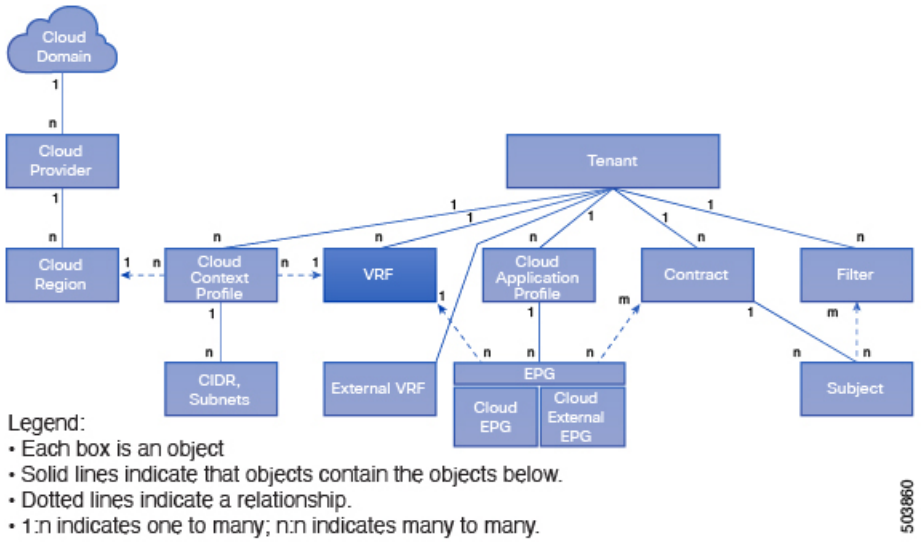
The cloud context profile contains information on the following Cisco Cloud Network Controller components:

- CIDRs
- VRFs
- EPGs
- Regions
- VPCs
- Endpoints

VRFs

A Virtual Routing and Forwarding (VRF) object (`fvCtx`) or context is a tenant network (called a VRF in the Cisco Cloud Network Controller GUI). A tenant can have multiple VRFs. A VRF is a unique Layer 3 forwarding and application policy domain. The following figure shows the location of VRFs in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 4: VRFs



A VRF defines a Layer 3 address domain. One or more cloud context profiles are associated with a VRF. You can only associate one cloud context profile with a VRF in a given region. All the endpoints within the Layer 3 domain must have unique IP addresses because it is possible to forward packets directly between these devices if the policy allows it. A tenant can contain multiple VRFs. After an administrator creates a logical device, the administrator can create a VRF for the logical device, which provides a selection criteria policy for a device cluster. A logical device can be selected based on a contract name, a graph name, or the function node name inside the graph.

External VRF

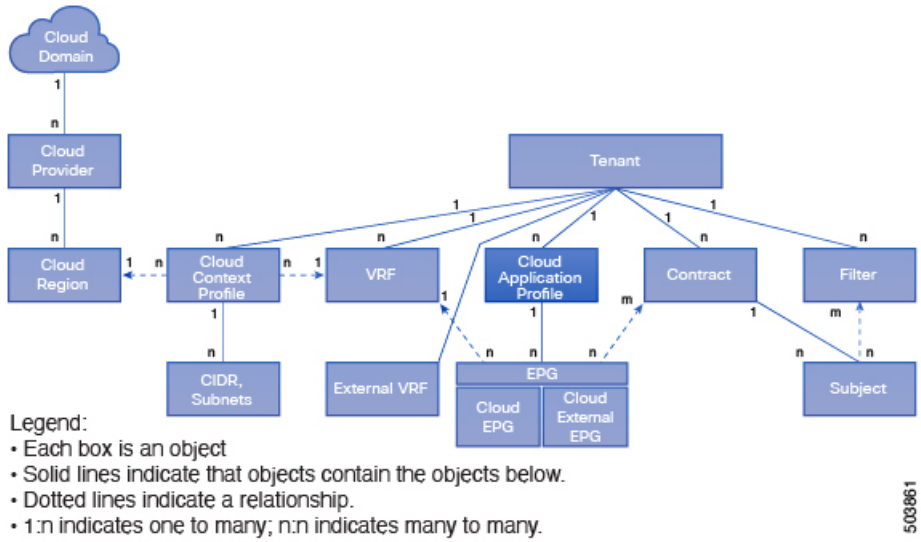
An **external VRF** is a type of VRF available for Cisco Cloud Network Controller. An external VRF is a unique VRF that does not have any presence in the cloud. This VRF is not referred to in any cloud context profile used by Cisco Cloud Network Controller.

An external VRF represents an external network that is connected to other cloud sites or to on-premises sites. Multiple cloud VRFs can leak routes to an external VRF or can get the routes from an external VRF. When an external network is created on an external VRF, inter-VRF routing is set up so that routes received and advertised on the external network are received or advertised on the external VRF.

Cloud Application Profiles

A cloud application profile (`cloudAp`) defines the policies, services and relationships between cloud EPGs. The following figure shows the location of cloud application profiles in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 5: Cloud Application Profiles



Cloud application profiles contain one or more cloud EPGs. Modern applications contain multiple components. For example, an e-commerce application could require a web server, a database server, data located in a storage service, and access to outside resources that enable financial transactions. The cloud application profile contains as many (or as few) cloud EPGs as necessary that are logically related to providing the capabilities of an application.

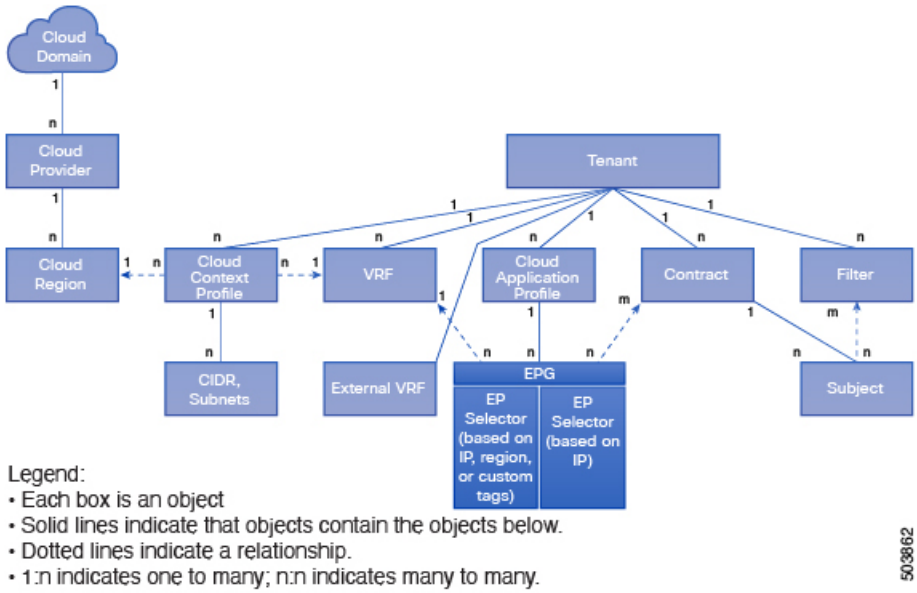
Cloud EPGs can be organized according to one of the following:

- The application they provide, such as a DNS server or SAP application (see *Tenant Policy Example* in *Cisco APIC REST API Configuration Guide*).
- The function they provide (such as infrastructure)
- Where they are in the structure of the data center (such as DMZ)
- Whatever organizing principle that a cloud infrastructure or tenant administrator chooses to use

Cloud Endpoint Groups

The cloud endpoint group (cloud EPG) is the most important object in the policy model. The following figure shows where application cloud EPGs are located in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 6: Cloud Endpoint Groups



A cloud EPG is a managed object that is a named logical entity that contains a collection of endpoints. Endpoints are devices that are connected to the network. They have an address (identity), a location, attributes (such as version or patch level), and are virtual. Knowing the address of an endpoint also enables access to all its other identity details. Cloud EPGs are fully decoupled from the physical and logical topology. Endpoint examples include servers, virtual machines, storage services, or clients on the Internet. Endpoint membership in a cloud EPG can be dynamic or static.

The CCNC cloud infrastructure can contain the following types of cloud EPGs:

- Cloud endpoint group (`cloudEPg`)
- Cloud external endpoint group (`cloudExtEPg`)

Cloud EPGs contain endpoints that have common policy requirements such as security services. Rather than configure and manage endpoints individually, they are placed in a cloud EPG and are managed as a group.

Policies apply to cloud EPGs, never to individual endpoints.

Regardless of how a cloud EPG is configured, cloud EPG policies are applied to the endpoints they contain.

WAN router connectivity to the cloud infrastructure is an example of a configuration that uses a static cloud EPG. To configure WAN router connectivity to the cloud infrastructure, an administrator configures a `cloudExtEPg` cloud EPG that includes any endpoints within an associated WAN subnet. The cloud infrastructure learns of the cloud EPG endpoints through a discovery process as the endpoints progress through their connectivity life cycle. Upon learning of the endpoint, the cloud infrastructure applies the `cloudExtEPg` cloud EPG policies accordingly. For example, when a WAN connected client initiates a TCP session with a server within an application (`cloudEPg`) cloud EPG, the `cloudExtEPg` cloud EPG applies its policies to that client endpoint before the communication with the (`cloudEPg`) cloud EPG web server begins. When the client server TCP session ends, and communication between the client and server terminates, the WAN endpoint no longer exists in the cloud infrastructure.

The Cisco Cloud Network Controller uses endpoint selectors to assign endpoints to Cloud EPGs. The endpoint selector is essentially a set of rules that are run against the cloud instances that are assigned to the Google

Cloud VPC managed by Cisco CCNC. Any endpoint selector rules that match endpoint instances assign that endpoint to the Cloud EPG. The endpoint selector is similar to the attribute-based microsegmentation available in Cisco ACI.

Contracts

In addition to cloud EPGs, contracts (`vzBrCP`) are key objects in the policy model. Cloud EPGs can only communicate with other cloud EPGs according to contract rules. The following figure shows the location of contracts in the management information tree (MIT) and their relation to other objects in the tenant.

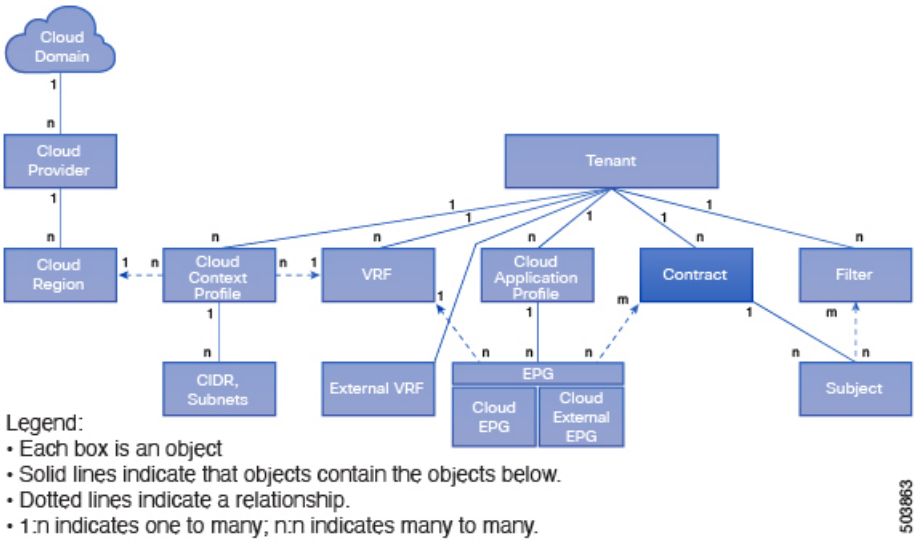


Figure 7: Contracts

An administrator uses a contract to select one or more types of traffic that can pass between cloud EPGs, including the protocols and ports allowed. If there is no contract, inter-EPG communication is disabled by default. There is no contract required for intra-EPG communication; intra-EPG communication is always implicitly allowed.

Contracts govern the following types of cloud EPG communications:

- Between cloud EPGs (`cloudEPg`), both intra-tenant and inter-tenant

Note In the case of a shared service mode, a contract is required for inter-tenant communication. A contract is used to specify static routes across VRFs, although the tenant VRF does not enforce a policy.

- Between cloud EPGs and cloud external EPGs (`cloudExtEPg`)

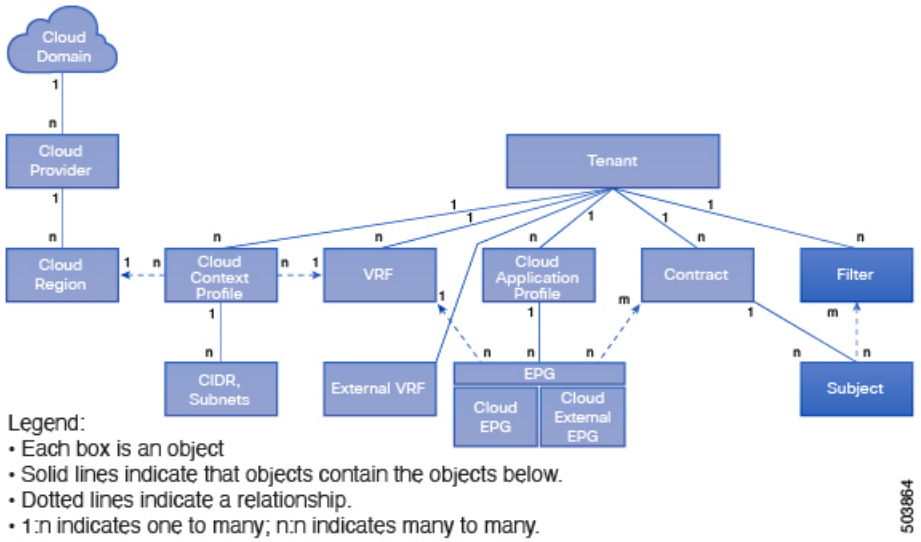
Contracts govern the communication between cloud EPGs that are labeled providers, consumers, or both. The relationship between a cloud EPG and a contract can be either a provider or consumer. When a cloud EPG provides a contract, communication with the cloud endpoints in that cloud EPG can be initiated from cloud endpoints in other cloud EPGs as long as the communication complies with the provided contract. When a cloud EPG consumes a contract, the cloud endpoints in the consuming cloud EPG may initiate communication with any cloud endpoint in a cloud EPG that is providing that contract.

Note A cloud EPG can both provide and consume the same contract. A cloud EPG can also provide and consume multiple contracts simultaneously.

Filters and Subjects Govern Cloud EPG Communications

Subject and filter managed-objects enable mixing and matching among cloud EPGs and contracts so as to satisfy various applications or service delivery requirements. The following figure shows the location of application subjects and filters in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 8: Subjects and Filters



Contracts can contain multiple communication rules and multiple cloud EPGs can both consume and provide multiple contracts. A policy designer can compactly represent complex communication policies and re-use these policies across multiple instances of an application.

Note Subjects are hidden in Cisco Cloud Network Controller and not configurable. For rules installed in Google Cloud, source port provided in the filter entry is not taken into account.

Subjects and filters define cloud EPG communications according to the following options:

- Filters are Layer 3 to Layer 4 fields, TCP/IP header fields such as Layer 3 protocol type, Layer 4 ports, and so forth. According to its related contract, a cloud EPG provider dictates the protocols and ports in both the in and out directions. Contract subjects contain associations to the filters (and their directions) that are applied between cloud EPGs that produce and consume the contract.
- Subjects are contained in contracts. A subject within a contract uses filters to specify the type of traffic that can be communicated and how it occurs. For example, for HTTPS messages, the subject specifies the direction and the filters that specify the IP address type (for example, IPv4), the HTTP protocol, and

the ports allowed. Subjects determine if filters are unidirectional or bidirectional. A unidirectional filter is used in one direction. Unidirectional filters define in or out communications but not the same for both. Bidirectional filters are the same for both; they define both in and out communications.

- CCNC contracts rendered in Google Cloud constructs are always stateful, allowing return traffic.

About the Cloud Template

The cloud template provides a template that configures and manages the Cisco Cloud Network Controller infra network. The template requires only the most essential elements for the configuration. From these elements, the cloud template generates a detailed configuration necessary for setting up the Cisco Cloud Network Controller infra network. However, it is not a one-time configuration generation—it is possible to add, modify, or remove elements of the template input. The cloud template updates the resulting configuration accordingly.

One of the central things in the Google Cloud network configuration is the Virtual Private Cloud (VPC). Google Cloud supports many regions worldwide and one VPC is specific to one region.

The cloud template accepts one or more region names and generates the entire configuration for the infra VPCs in those regions. They are the infra VPCs. The Cisco Cloud Network Controller-managed object (MO) corresponding to the Google Cloud VPC is `cloudCtxProfile`. For every region specified in the cloud template, it generates the `cloudCtxProfile` configuration. A `cloudCtxProfile` is the topmost MO for all the configuration corresponding to a region. Underneath, it has many of other MOs organized as a tree to capture a specific configuration. The `cloudCtxProfile` MO for the infra VPC is generated by the cloud template. It carries `ctxProfileOwner == SYSTEM`, which means that this MO is generated by the system. For the non-infra network, it is possible to configure `cloudCtxProfile` directly; in this case, `cloudCtxProfile` carries `ctxProfileOwner == USER`.

A primary property of a Google Cloud VPC is the CIDR. In Cisco Cloud Network Controller, you can choose and deploy CIDRs in the user VPCs. The CIDRs for the infra VPC are provided by users to the cloud template during the initial setup of the cloud site, and are deployed to the Google Cloud by the cloud template.

A property called `createdBy` is also available for the CIDR. The default value for this `createdBy` property is `USER`.

- For all user-created CIDRs, the value for the `createdBy` property is set to `USER`.
- For cloud template-created CIDRs, the value for the `createdBy` property is set to `SYSTEM`.

Multiple CIDR and subnet blocks can be configured on the infra VPC. You can create CIDRs and associate subnets in the infra VPC. The cloud template subnets will be mapped to the overlay-1 VRF. All subnets in the respective VRFs will have separate route tables in the cloud for VRF segregation.

For more information, see Creating an Application EPG Using the Cisco Cloud Network Controller GUI.

The cloud template generates and manages a huge number of MOs in the `cloudCtxProfile` subtree including, but not limited to, the following:

- Subnets
- Cloud routers
- IP address allocation for the cloud router interfaces
- IP address allocation and configuration for tunnels

- IP address allocation and configuration for loopbacks

Without the cloud template, you would be responsible for configuring and managing these.

The *Cisco Cloud Template MO* table contains a brief summary of the inputs (MOs) to the cloud template.

Table 1: Cloud Template MOs

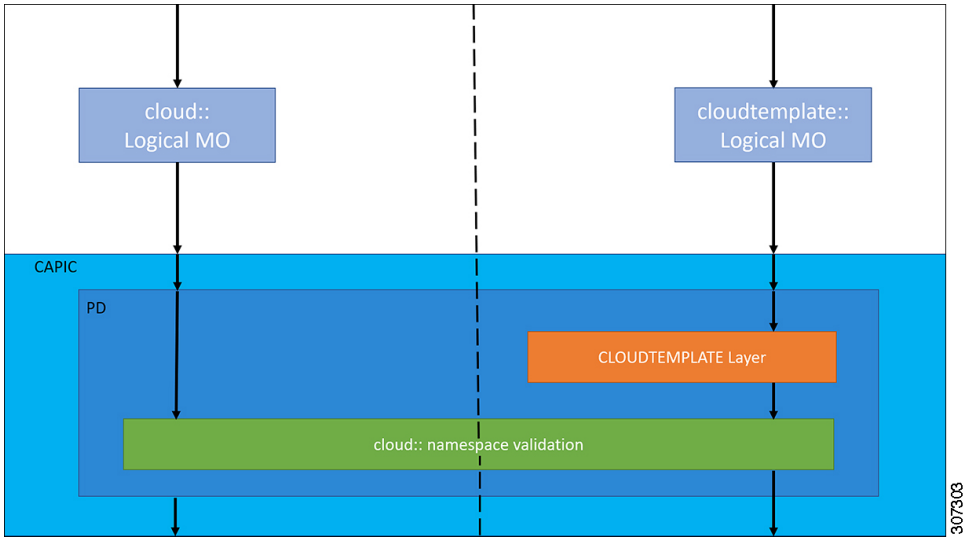
MO	Purpose
`cloudtemplateInfraNetwork`	The root of the cloud template configuration. Attributes include: `numRoutersPerRegion`—The number of cloud routers for each `cloudRegionName` specified under `cloudtemplateIntNetwork`.
`cloudtemplateIntNetwork`	Contains a list of regions, which specify where you deploy the cloud routers. Each region is captured through a `cloudRegionName` child MO
`cloudtemplateExtNetwork`	Contains infra network configuration input that is external of the cloud. Contains a list of regions where cloud routers are configured for external networking. Each region is captured through a `cloudRegionName` child MO
`cloudtemplateIpSecTunnel`	Captures the IP address of the IPSec peer in the ACI on-premises site.

In Cisco Cloud Network Controller, the layering of MOs is slightly different from a regular Cisco APIC due to the cloud template. In a regular Cisco APIC, you post logical MOs that go through two layers of translation:

1. Logical MO to resolved MO
2. Resolved MO to concrete MO

In Cisco Cloud Network Controller, there is an additional layer of translation for the infra network. This additional layer is where the cloud template translates logical MOs in the `cloudtemplate` namespace to logical MOs in the cloud namespace. For configurations outside of the infra network, you post logical MOs in the cloud namespace. In this case, the MOs go through the usual two-layer translation as in the regular Cisco APIC.

Figure 9: Cloud and Cloud Template MO Conversion



Note For information about configuring the cloud template, see Configuring Cisco Cloud Network Controller Components

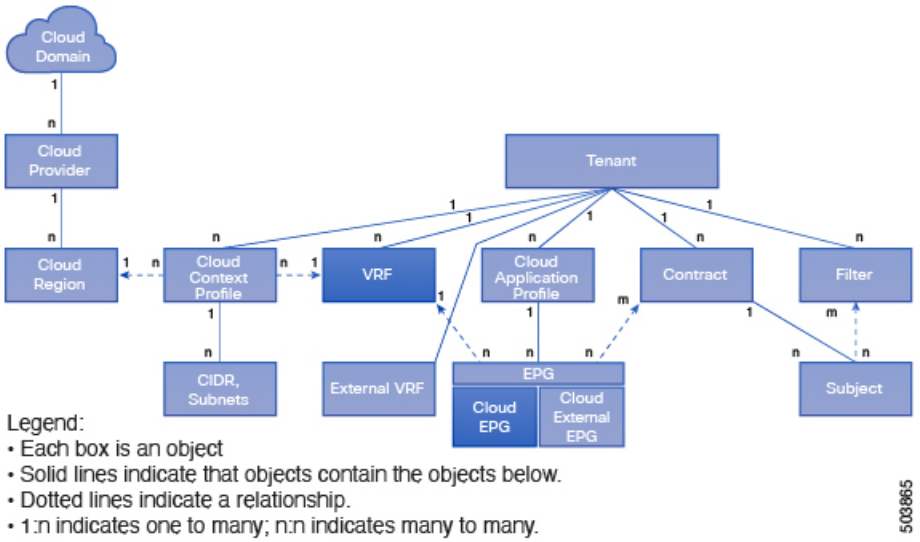
Managed Object Relations and Policy Resolution

Relationship-managed objects express the relation between managed object instances that do not share containment (parent-child) relations. MO relations are established between the source MO and a target MO in one of the following two ways:

- An explicit relation, such as with `cloudRsCloudEPgCtx`, defines a relationship that is based on the target MO distinguished name (DN).
- A named relation defines a relationship that is based on the target MO name.

The dotted lines in the following figure show several common MO relations.

Figure 10: MO Relations



For example, the dotted line between the cloud EPG and the VRF defines the relation between those two MOs. In this figure, the cloud EPG (`cloudEPg`) contains a relationship MO (`cloudRsCloudEPgCtx`) that is named with the name of the target VRF MO (`fvCtx`). For example, if production is the VRF name (`fvCtx.name=production`), then the relation name is production (`cloudRsCloudEPgCtx.tnFvCtxName=production`).

In the case of policy resolution based on named relations, if a target MO with a matching name is not found in the current tenant, the CCNC cloud infrastructure tries to resolve in the common tenant. For example, if the user tenant cloud EPG contained a relationship MO targeted to a VRF that did not exist in the tenant, the system tries to resolve the relationship in the common tenant. If a named relation cannot be resolved in either the current tenant or the common tenant, the CCNC cloud infrastructure attempts to resolve to a default policy. If a default policy exists in the current tenant, it is used. If it does not exist, the CCNC cloud infrastructure looks for a default policy in the common tenant. Cloud context profile, VRF, and contract (security policy) named relations do not resolve to a default.

Default Policies

Warning Default policies can be modified or deleted. Deleting a default policy can result in a policy resolution process to complete abnormally.

The CCNC cloud infrastructure includes default policies for many of its core functions. Examples of default policies include the following:

- Google Cloud provider (for the infra tenant)
- Monitoring

Note To avoid confusion when implementing configurations that use default policies, document changes made to default policies. Be sure that there are no current or future configurations that rely on a default policy before deleting a default policy. For example, deleting a default firmware update policy could result in a problematic future firmware update.

A default policy serves multiple purposes:

- Allows a cloud infrastructure administrator to override the default values in the model.
- If an administrator does not provide an explicit policy, the Cisco Cloud Network Controller applies the default policy. An administrator can create a default policy and the Cisco Cloud Network Controller uses that unless the administrator provides any explicit policy.

The policy model specifies that an object is using another policy by having a relation-managed object (MO) under that object and that relation MO refers to the target policy by name. If this relation does not explicitly refer to a policy by name, then the system tries to resolve a policy that is called default. Cloud context profiles and VRFs are exceptions to this rule.