vzAny Contracts

vzAny and Multi-Site

The `vzAny` managed object provides a convenient way of associating all endpoint groups (EPGs) in a Virtual Routing and Forwarding (VRF) instance to one or more contracts, instead of creating a separate contract relation for each EPG.

In the Cisco ACI fabric, EPGs can only communicate with other EPGs according to contract rules. A relationship between an EPG and a contract specifies whether the EPG provides the communications defined by the contract rules, consumes them, or both. By dynamically applying contract rules to all EPGs in a VRF, vzAny automates the process of configuring EPG contract relationships. Whenever a new EPG is added to a VRF, vzAny contract rules automatically apply. The vzAny one-to-all EPG relationship is the most efficient way of applying contract rules to all EPGs in a VRF.

Advantages

Policy information in Cisco ACI is programmed in the fabric switches' TCAM tables. TCAM entries are typically specific to each pair of EPGs that are allowed to communicate with each other via a Contract. This means that even if the same contract is re-used, multiple TCAM entries are created for every pair of EPGs.

The size of the policy TCAM table depends on the generation of the switches that you are using. In certain large scale environments it is important to take policy TCAM usage into account and ensure that the limits are not exceeded.

vzAny allows you to combine all EPGs within the same VRF into a single "group" and create a contract relationship with that group rather than individual EPGs within it, while consuming only a single TCAM entry. This saves the time you would otherwise spend creating multiple contract relationships for individual EPGs in the VRF as well as the TCAM space.

Use Cases

There are two typical use cases for vzAny:

- Free communication between EPGs within the same VRF, as described in Free Intra-VRF Communication, on page 6.
- Many-to-one communication allowing all EPGs within the same VRF to consume a shared service from a single EPG, as described in more detail in Many-to-One Communication, on page 11.

vzAny and Multi-Site Guidelines and Limitations

The following guidelines and limitations apply when using vzAny:

- If you plan to enable the vzAny object for a given VRF to provide or consume a contract, the following additional restrictions apply:
  - If vzAny for a given VRF is configured as consumer of a contract `C1`, the vzAny objects for other VRFs must not be configured as providers of `C1`.
  - If vzAny for a given VRF is configured as provider of a contract `C1`, the vzAny objects for other VRFs must not be configured as consumers of `C1`.
  - If an External EPG part of a given VRF is consuming a contract `C1`, the vzAny objects for other VRFs must not be configured as providers of `C1`.
  - If an EPG part of a given VRF is consuming a contract `C1`, the vzAny objects for other VRFs must not be configured as providers of `C1`.
  - If vzAny for a given VRF is configured as provider of a contract `C1`, then EPGs, External EPGs or vzAny objects for other VRFs must not be configured as consumers of `C1`.
- EPGs and External EPGs objects in a given VRF must not be configured as part of the Preferred Group if vzAny for that VRF is already consuming or providing a contract.
- If any EPG or External EPG objects in a given VRF are deployed in a cloud site, it is not possible to configure vzAny for that VRF to consume or provide a contract
- vzAny is supported with inter-VRF intersite L3Out configurations only when the fabrics are part of the Multi-Site domain running Cisco ACI 5.2(4) release or later.
- vzAny must not consume or provide a contract that is associated with a Service-Graph with PBR.
- vzAny can be configured as provider, consumer or both of a contract for establishing intra-VRF communication.
- vzAny is supported only as a consumer of a shared service but not as a provider.
- We recommend stretching the vzAny VRF to all sites where you plan to deploy EPGs and BDs that use it.
- You can import existing vzAny configurations from an APIC.

Note In certain cases due to an existing issue (CSCvt47568), if you make changes to the imported configuration before re-deploying it from the Multi-Site Orchestrator, some changes may not get correctly updated in the APICs. To avoid this, re-deploy the configuration immediately after importing but before making any changes to it. After you re-deploy the unchanged config, you will be able to update it as normal.

- vzAny providers and consumers include application EPGs, external EPGs associated to L3Outs, and endpoint groups for in-band or out-of-band access.
- vzAny implicitly creates a `0.0.0.0/0` classification for externally originating traffic, allowing all traffic originating from any external IP subnet. When vzAny is in use for a VRF, it also includes the external EPGs associated to the L3Outs part of that VRF, hence it is equivalent to having created a L3external classification that includes the subnets specified in the VRF itself.
- If an EPG within a VRF is consuming a shared service contract provided by an EPG in a different VRF, the traffic from the EPG of the provider VRF is filtered within the consumer VRF. vzAny is equivalent to a wildcard for the source or destination EPG.

  Be careful when you configure a shared service contract between vzAny in the consumer VRF and an EPG1 in a different provider VRF. Since the policy enforcement (filtering) is always performed in the consumer VRF, if the subnet associated to another EPG2 that is part of the provider VRF is leaked into the consumer VRF, then EPG2 will start communicating with consumer EPGs across VRFs even without explicitly providing a contract. Failure to observe this guideline could allow unintended traffic between EPGs across VRFs.
- Configuring a VRF with vzAny as both provider and consumer of a contract using an "allow all" filter, is the same as configuring an unenforced VRF. This implies that all EPGs within that VRF are free to communicate to each other without a contract.
- If the contract scope is application-profile, the vzAny configuration is ignored and filter rules are expanded; CAM utilization is the same as if specific contracts were deployed between each pair of consumer and provider EPGs. In this case, there is no benefit in terms of TCAM space usage.
- In the case of shared services, you must define the provider EPG shared subnet under the EPG in order to properly derive the classification (`pcTag`) of the destination on the consumer (vzAny) side. If you are migrating from a BD-to-BD shared services configuration, where both the consumer and provider subnets are defined under bridge domains, to vzAny acting as a shared service consumer, you must take an extra configuration step where you add the provider subnet to the EPG with the shared flags at minimum. However, since the subnet under the EPG is not needed for connectivity, it is always recommended to check the `No default SVI gateway` flag.

  If you add the EPG subnet as a duplicate of the defined BD subnet, ensure that both definitions of the subnet always have the same flags defined. Failure to do so can result in an error.

Create Contract and Filters

When using vzAny, you are essentially creating a single point for a contract relationship, as such you must have a typical contract you will use for any such relationship as well as the filter for the contract.

This section describes how to create a new contract specifically for this purpose. Alternatively, you can choose to import any existing vzAny contracts you have configured on each APIC site.

Procedure

Step 1 Log in to the Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation pane, select **Schemas**.

Step 3 Select the Schema where you want to create your Contract.

If you have an existing Schema you want to update, simply click the Schema's name in the main window pane. Otherwise, if you want to create a new Schema, click the **Add Schema** button and provide the schema information, such as the name and tenant, as you typically would.

Step 4 Create a filter.

a) Scrolls down to the **Filter** area and click the **+** sign to add a new filter.
b) Provide the name for the Contract.
c) Click **+Entry** to add a filter entry.
d) In the **Add Entry** window, provide filter details.

Provide the filter details as you typically would to define the kind of traffic you want to allow.

e) Click **SAVE** to add the entry.
f) (Optional) If required, create additional filter entries.

Step 5 Create the contract.

a) Scrolls down to the **Contract** area and click the **+** sign to add a new contract.
b) Provide the name for the Contract.

For example, `contract-vzany`.

c) Choose the scope for the contract

Choose the scope appropriate for your use-case. For example, if you want to enable cross-tenant shared services, you must set the scope to `Global`.

d) Choose whether the contract will apply in both directions
e) Click **+Filter** to add one or more contract filters.
f) In the **Add Filter Chain** window, choose the filter you created in the previous step.
g) Click **SAVE** to add the filter.
h) (Optional) If required, repeat the procedure to provide additional filters.
i) (Optional) If you disabled the **Apply Both Directions** option, provide filters for both, consumer and provider directions.

You have now created the contract you will use with vzAny in the next section.

Configure vzAny to Consume/Provide a Contract

This section describes how to create a vzAny VRF or enable an existing VRF for vzAny.

Before you begin

You must have:

- Created a Contract and one or more Filters to use with vzAny as described in Create Contract and Filters, on page 3.

Procedure

Step 1 Log in to the Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigration pane, select **Schemas**.

Step 3 Select the Schema for the vzAny VRF.

If you have an existing Schema you want to update, simply click the Schema's name in the main window pane. Otherwise, if you want to create a new Schema, click the **Add Schema** button and provide the schema information, such as the name and tenant, as you typically would.

Step 4 Create or select a VRF.

If you have an existing VRF for which you want to configure vzAny to provide/consume a contract, simply click the VRF in the main window pane. Otherwise, if you want to create a new VRF, scroll down to the **VRF** area and click the + sign.

Step 5 Select vzAny.

In the right sidebar, check the **vzAny** checkbox.

Step 6 Select the vzAny contract.

The **+Contract** option becomes available after you enable the **vzAny** checkbox.

a) Click **+Contract** to add the contract

b) Select the contract.

Select the contract you created in Create Contract and Filters, on page 3.

c) Select the Contract type.

You can choose either `consumer` or `provider` for the contract based on your use case.

Create EPGs to Be Part of the vzAny VRF

You can choose to create new or use existing EPGs for your vzAny use cases. There are no explicit vzAny settings on the EPGs and free communication is allowed by default for any EPG that is part of the vzAny VRF. If you simply enabled vzAny for an existing VRF with all its EPGs already created and configured, you can skip this section.

Before you begin

You must have:

- Created a Contract and one or more Filters to use with vzAny as described in Create Contract and Filters, on page 3.
- Created the vzAny VRF and assigned the Contract to it as described niConfigure vzAny to Consume/Provide a Contract, on page 4.

Procedure

Step 1 If you want to create an EPG to be part of the vzAny VRF

a) Create a BD you will use for your EPG.
b) In the BD configuration sidebar's **Virtual Routing & Forwarding** dropdown, select the vzAny VRF you created.
c) Create an EPG.
d) In the EPG configuration sidebar's **Bridge Domain** dropdown, select the BD you created.

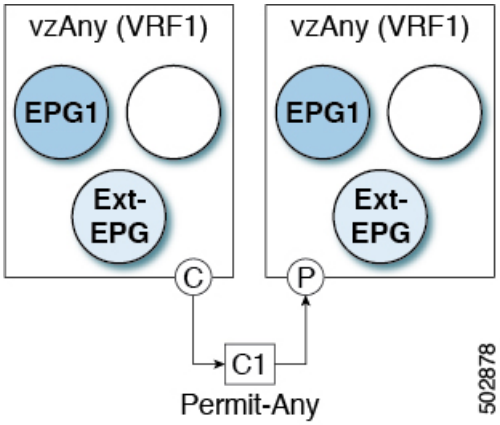
Step 2 If you want to create an External EPG to be part of the vzAny VRF...

a) Create an external EPG.
b) In the External EPG configuration sidebar's **Virtual Routing & Forwarding** dropdown, select the vzAny VRF you created.

Free Intra-VRF Communication

This section shows a number of schema examples for unrestricted intra-VRF communication. In all shown scenarios vzAny provides and consumes a contract with a `permit-any` filter. This essentially uses the ACI fabrics for network connectivity only without any policy enforcement and is equivalent to the "VRF Unenforced" option.

Figure 1:



For all the following use cases, you will need to create the same objects and policies summarized below. However, the schema and template design will depend on the number of sites as well as which objects are going to be stretched. Specific sections contain recommendation on template layout.

Procedure

Step 1 Create a Schema.

Step 2 Create a single common Template.

Step 3 Create any additional templates for every combination of sites where EPGs will be deployed .

If you will deploy a single template to all sites, you can skip this step. The use-case diagrams in the following sections provide template examples.

Step 4 Within the common Template, create the contract and filters to be consumed/provided by vzAny.

In this specific use case, the contract should have a single "permit-any" filter rule.

For specific steps, see Create Contract and Filters, on page 3.

Step 5 Within the common Template, create a VRF and configure vzAny to consume and provide the previously defined contract with the "permit-any" rule.

This ensures that free intra-VRF communication can be established.

For specific steps, see Configure vzAny to Consume/Provide a Contract, on page 4.

Step 6 Within each site's template, create and configure the EPGs that will be deployed to that site only.

If you will deploy a single template to all sites, create the EPGs within the same template as the VRF instead. The use-case diagrams in the following sections provide template examples.

This is described in Create EPGs to Be Part of the vzAny VRF, on page 5.

Step 7 Assign the common Template to every site.

Step 8 Assign each template to the appropriate sites.

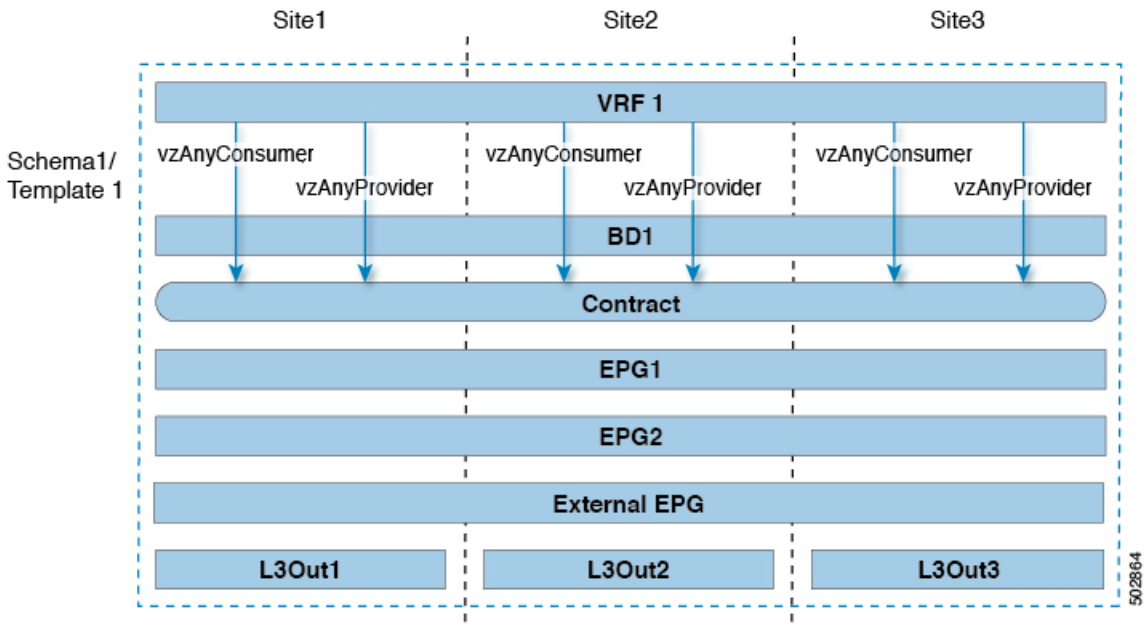
Step 9 Deploy the templates.

Stretched EPGs

The following example shows intra-VRF communication between EPGs or External EPGs all of which are stretched between sites. In this example EPG1 and EPG2 are mapped to the same BD1, but they could each be part of different BDs as long as both BDs are part of VRF1.

In this case you can create all objects within the same template and then deploy the template to all sites.

Figure 2:



Site-Local EPGs

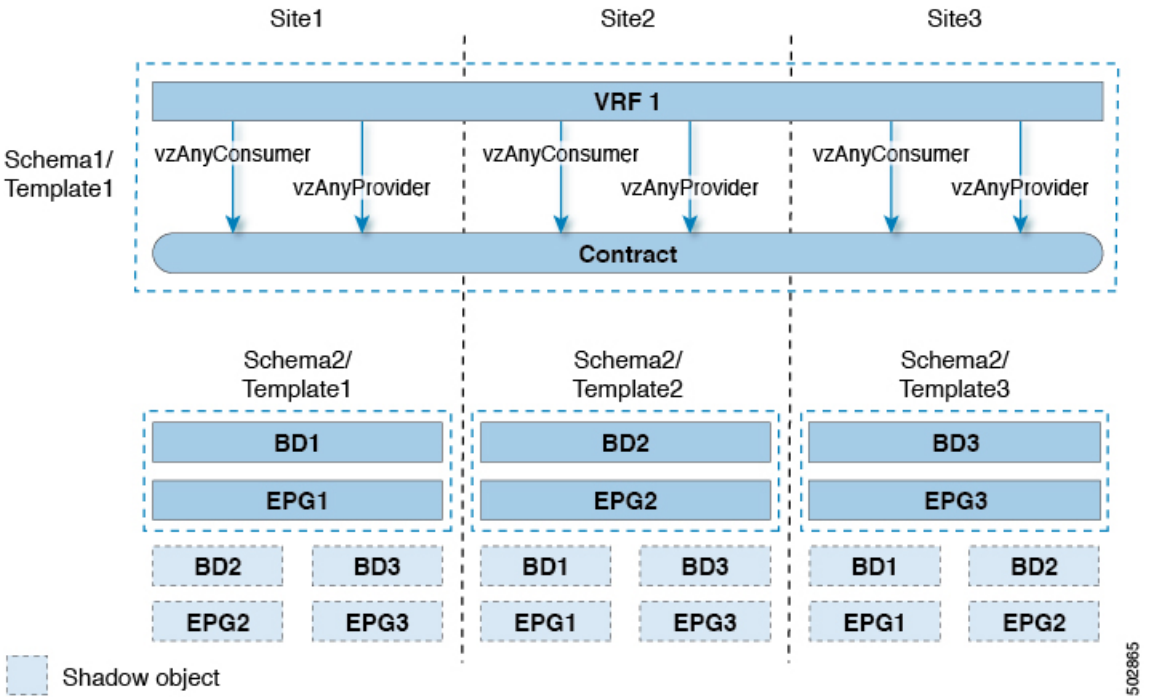
The following example shows intra-VRF communication between EPGs or External EPGs where none of the EPGs are stretched but can still freely communicate with each other since vzAny consumes and provides the "permit-any" contract.

In this case you will need to create multiple templates:

- A single template for the shared objects (VRF, Contract) deployed to every site.
- And a separate template for every site containing the EPG and BD deployed that site.

For the objects that are not stretched, shadow objects are created in other sites.

Figure 3:



Combination of Site-Local and Stretched EPGs

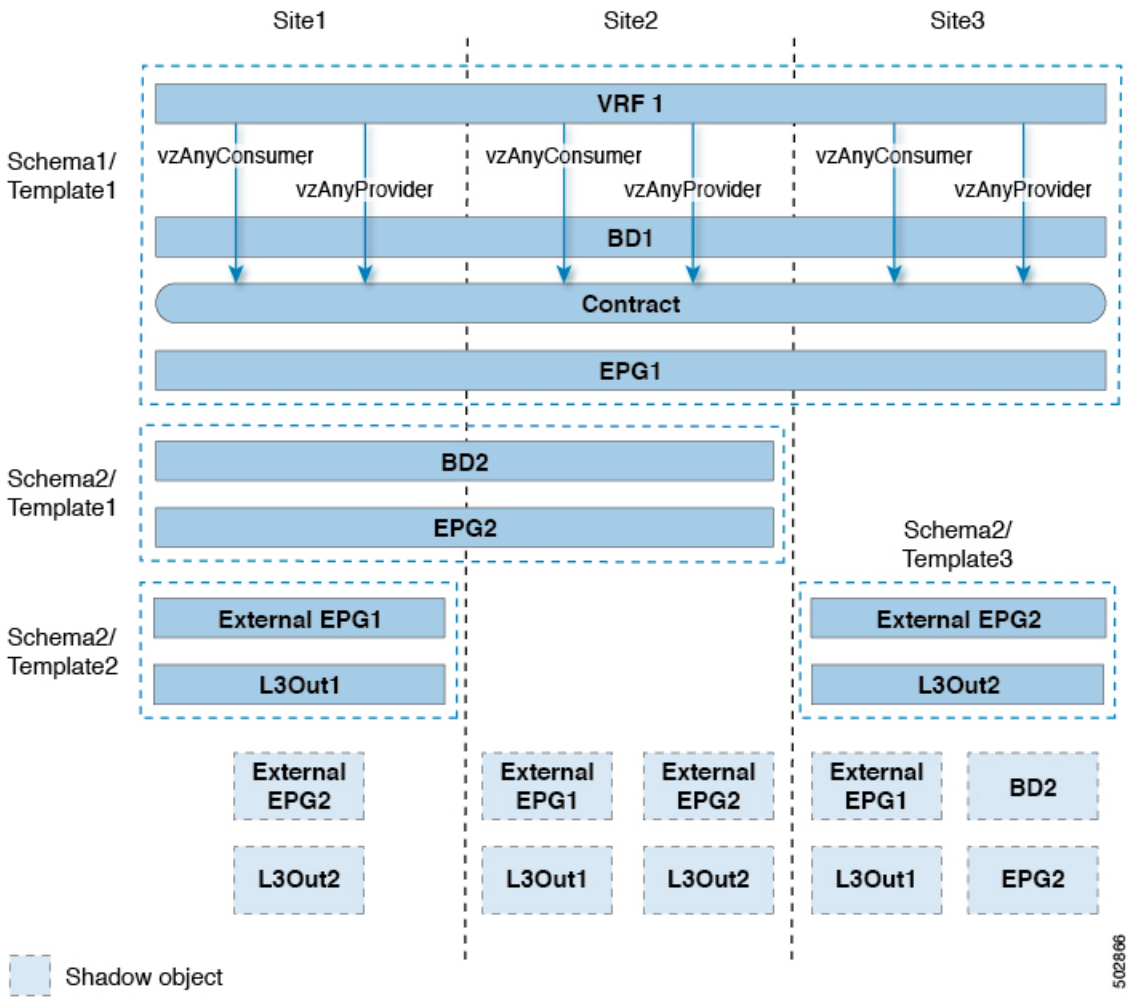
The following example shows intra-VRF communication between EPGs or External EPGs where some EPGs are stretched while others are deployed to a single site only. All EPGs can still freely communicate with each other since vzAny consumes and provides the "permit-any" contract.

In this case you will need to create multiple templates:

- A single template for the shared objects (VRF, Contract, BDs) deployed to every site.
- And a separate template for every site combination containing the objects deployed only to those sites.

For the objects that are not stretched, shadow objects are created in other sites.

Figure 4:



Intra-VRF Intersite L3Out

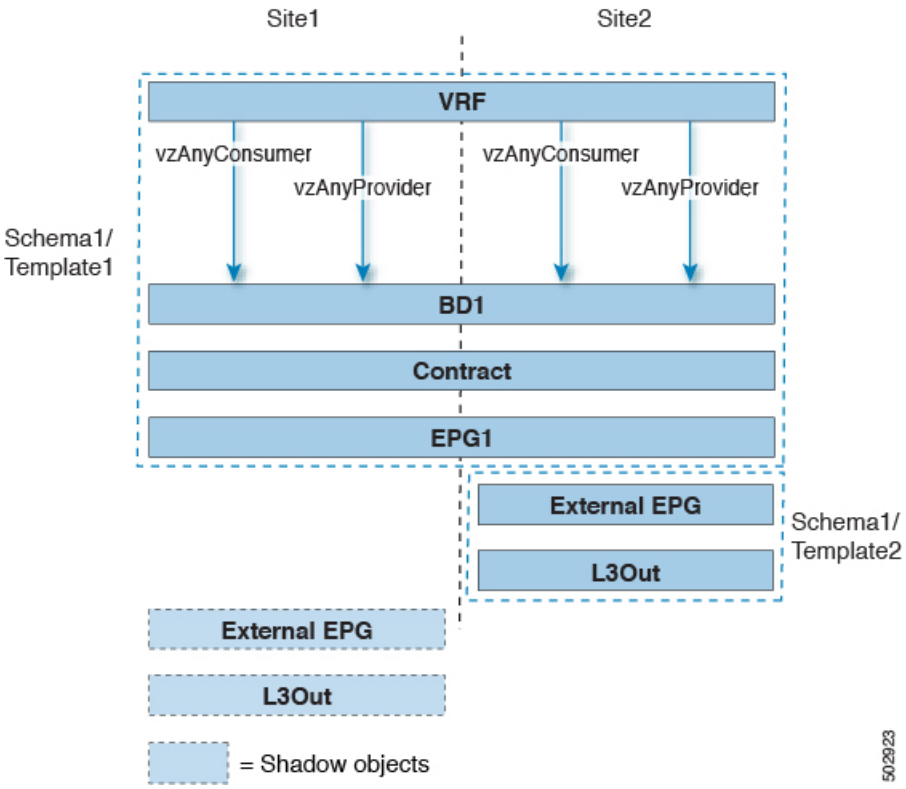
This use case allows you to configure an intersite L3Out for multiple EPGs within a vzAny VRF. When the L3Out's external EPG is in the same VRF, you do not need to explicitly add the provider contract to the external EPG.

Keep in mind, when configuring an intersite L3Oout, you must configure a routable TEP pool for each Pod. Additional intersite L3Out details and requirements are described in the Intersite L3Out Overview section.

In this case you will need to create multiple templates:

- A single template for the shared vzAny objects (VRF, Contract, BD) deployed to one or more sites.
- And a separate template for every site combination containing the objects deployed only to those sites.

Figure 5:

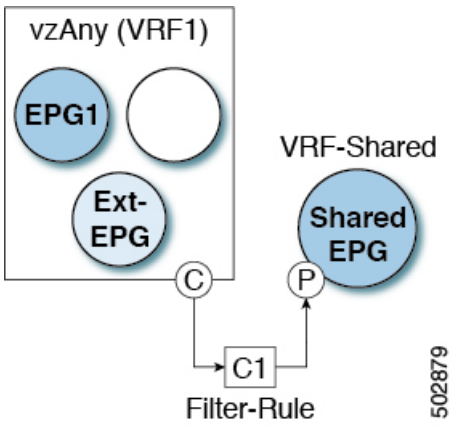


Many-to-One Communication

The following three sections provide schema examples of multiple EPGs that are part of the same vzAny VRF communicating with a single EPG that is providing a shared service. In this case, the contract can specify one or more filter rules.

The EPG providing shared services can be in a separate VRF (as shown in the figure below) or it can be part of the vzAny VRF.

Figure 6:



For all the following use cases, you will need to create the same objects and policies summarized below. However, the schema and template design will depend on the number of sites as well as which objects are going to be stretched. Specific sections contain recommendation on template layout.

Procedure

Step 1 Create a Schema.

Step 2 Create a single common Template.

Step 3 Create any additional templates for every combination of sites where EPGs will be deployed .

Step 4 Within the common Template, create the contract and filters to be consumed by vzAny and provided by the EPG offering shared services.

This is described in Create Contract and Filters, on page 3.

Step 5 Within the common Template, create a VRF and configure vzAny to consume the previously defined contract.

This is described in Configure vzAny to Consume/Provide a Contract, on page 4.

Step 6 Within each site's template, create and configure the EPGs that are part of the vzAny VRF.

This is described in Create EPGs to Be Part of the vzAny VRF, on page 5.

Step 7 Create new or configure existing provider EPG or external EPG.

You create and configure the provider EPG or external EPG as you typically would.

Step 8 Assign the Contract to the provider EPG.

In addition to assigning the contract to be consumed by vzAny, you will also need to assign the same contract to the provider EPG.

Provider EPG Within vzAny VRF

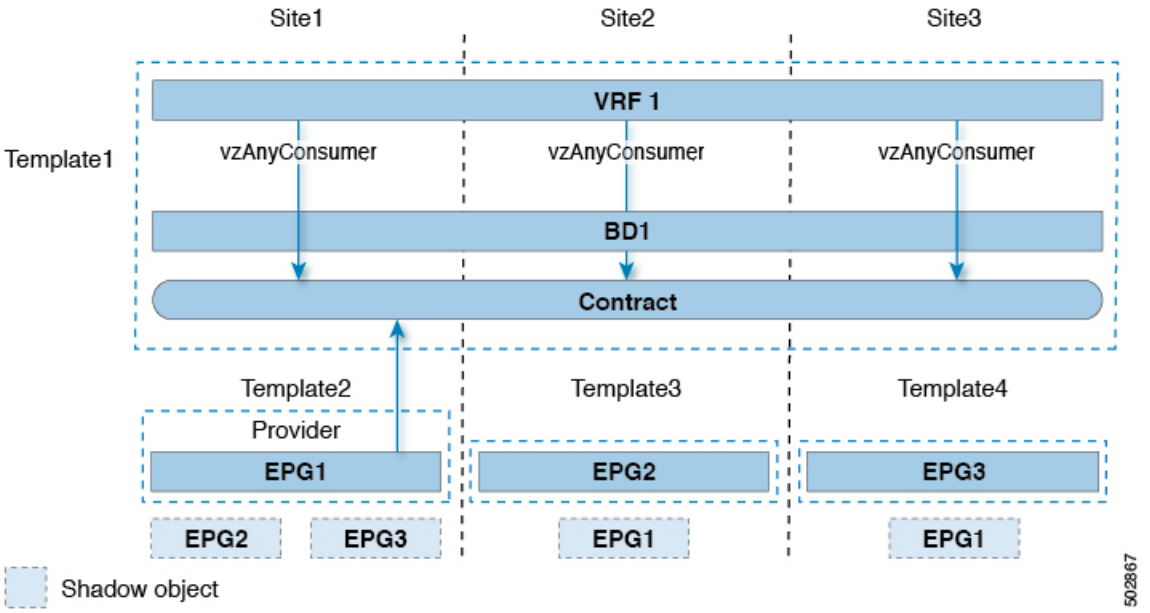
The following example shows intra-VRF communication between a single provider EPG (for example, shared service) and all other EPGs within the same VRF consuming the service.

In this case you will need to create multiple templates:

- A single template for the shared objects (VRF, Contract, BDs) deployed to every site.
- And a separate template for every site combination containing the objects deployed only to those sites.

The following figure shows a single stretched VRF/BD configuration. Alternatively, you can also configure and map a dedicated BD for each EPG, in which case shadow BDs would be deployed in the remote sites.

Figure 7:



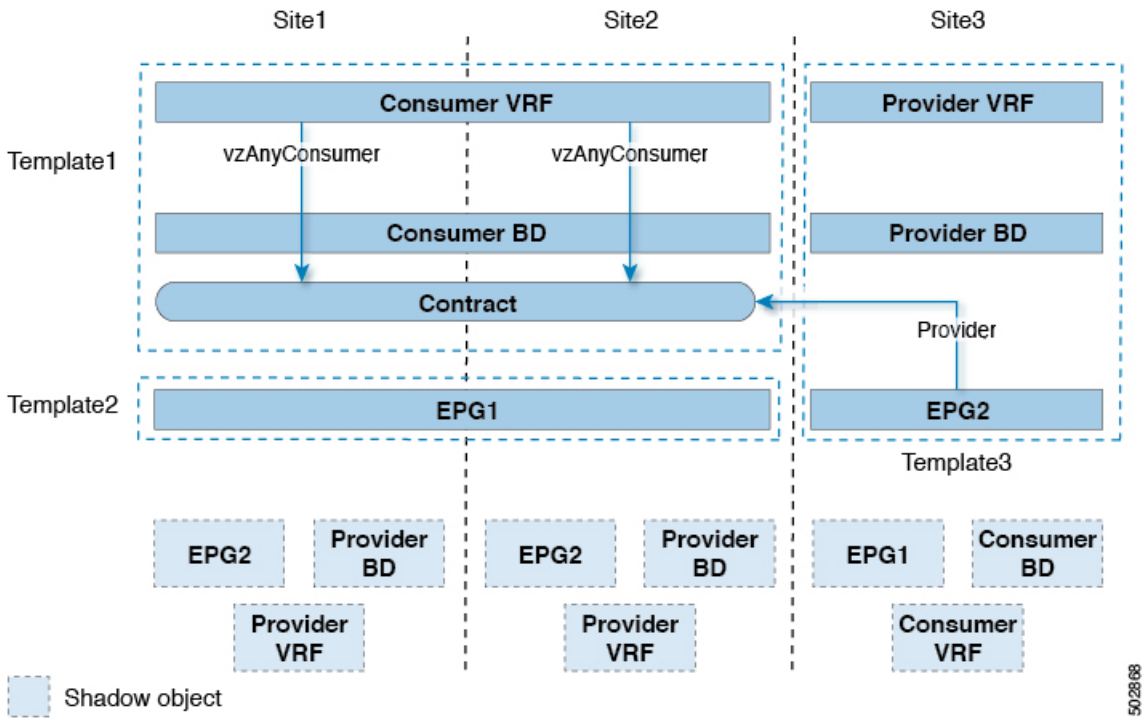
Provider EPG In Its Own VRF

The following example shows communication between a single EPG (for example, shared service provider) in its own VRF and all EPGs within a different, vzAny VRF. The provider EPG can be deployed to the same or a different site as the consumer EPGs in the vzAny VRF.

In this case you will need to create multiple templates:

- A single template for the shared vzAny objects (VRF, Contract, BD) deployed to one or more sites.
- And a separate template for every site combination containing the objects deployed only to those sites.

Figure 8:


Site1
Site2
Site3
Consumer VRF
Provider VRF
Template1
vzAnyConsumer
vzAnyConsumer
Consumer BD
Provider BD
Contract
Provider
Template2
EPG1
EPG2
Template3
EPG2
Provider BD
EPG2
Provider BD
EPG1
Consumer BD
Provider VRF
Provider VRF
Consumer VRF
Shadow object
502868