

vzAny Contracts for ACI Fabrics, 4.2.1

Table of Contents

New and changed information . . . 1
vzAny and Multi-Fabric . . . 2
  Advantages . . . 2
  Use Cases . . . 2
vzAny and Multi-Fabric Guidelines and Limitations . . . 3
Create Contract and Filters . . . 5
Configure vzAny to Consume/Provide a Contract . . . 7
Create EPGs to Be Part of the vzAny VRF . . . 8
Free Intra-VRF Communication . . . 9
  Stretched EPGs . . . 10
  Fabric-Local EPGs . . . 10
  Combination of Fabric-Local and Stretched EPGs . . . 11
  Intra-VRF Inter-Fabric L3Out . . . 12
  Inter-VRF Inter-Fabric L3Out . . . 13
Many-to-One Communication . . . 14
  Provider EPG Within vzAny VRF . . . 15
  Provider EPG In Its Own VRF . . . 15
Copyright . . . 17

New and changed information

The following table provides an overview of the significant changes up to this current release. The table does not provide an exhaustive list of all changes or of the new features up to this release.

Release Version	Feature	Description
Nexus Dashboard 4.2.1	There were no major changes from the previous release.	

vzAny and Multi-Fabric

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.



External EPGs that are associated with L3Outs and are part of a VRF are also included in the vzAny logical group.

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.
- 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.

vzAny and Multi-Fabric 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.
- vzAny is supported with inter-VRF inter-fabric L3Out configurations only when the fabrics are part of the Multi-Fabric 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.
- When configuring vzAny as a provider, the contract scope is restricted to context (VRF) only. Global scope contracts are not supported for vzAny providers, and Nexus Dashboard enforces this restriction.
- vzAny is supported only as a consumer of a shared service but not as a provider.
- We recommend stretching the vzAny VRF to all fabrics where you plan to deploy EPGs and BDs that use it.
- You can import existing vzAny configurations from an APIC.



In certain cases due to an existing issue (CSCvt47568), if you make changes to the imported configuration before re-deploying it from the Nexus Dashboard, 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 contracts you have configured on each APIC fabric.

1. Navigate to the **Orchestration** page.

   Manage > Orchestration

2. Choose **Tenant Templates > Applications**.
3. Choose 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 page. 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.

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** page, 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.
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** page, 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

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.

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

1. Navigate to the **Orchestration** page.

   Manage > Orchestration

2. Choose **Tenant Templates > Applications**.
3. Select the Schema containing the specific template with the definition of the VRF.

   For new configuration, you can create a new schema with the **Add Schema** button and then define a new template (associated to the tenant of interest) where you can configure VRF.

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 page. Otherwise, if you want to create a new VRF, scroll down to the **VRF** area and click the **+** sign.

5. Select vzAny.

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

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.

   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

Before you begin:

You must have:

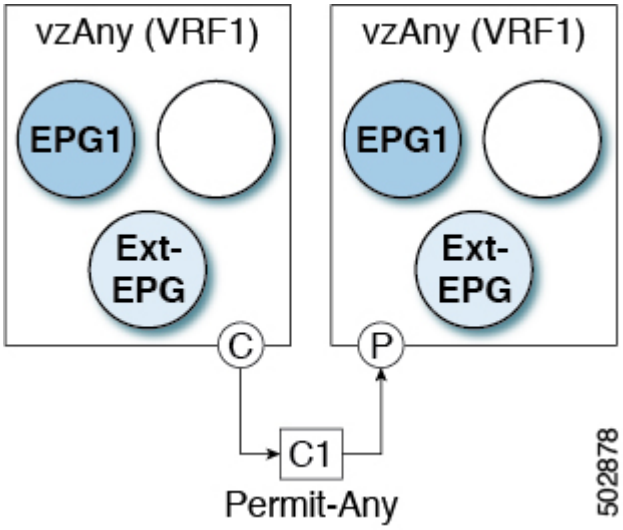
- Created a Contract and one or more Filters to use with vzAny as described in Create Contract and Filters.
- Created the vzAny VRF and assigned the Contract to it as described in Configure vzAny to Consume/Provide a Contract.

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 as soon as an EPG is associated to a BD in a VRF, the EPG becomes part of the vzAny logical group for that VRF (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.

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.
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.



Free Intra-VRF Communication

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 fabrics as well as which objects are going to be stretched. The specific sections below contain recommendation on template layout.

1. Create a Schema.
2. Create a common Template used to deploy configuration objects in all the fabrics (that is *stretched objects*).
3. Create any additional templates for every combination of fabrics where EPGs will be deployed .

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

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.

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.

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

If you will deploy a single template to all fabrics, 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.

7. Assign the common Template to every fabric.
8. Assign each template to the appropriate fabrics.
9. Deploy the templates.

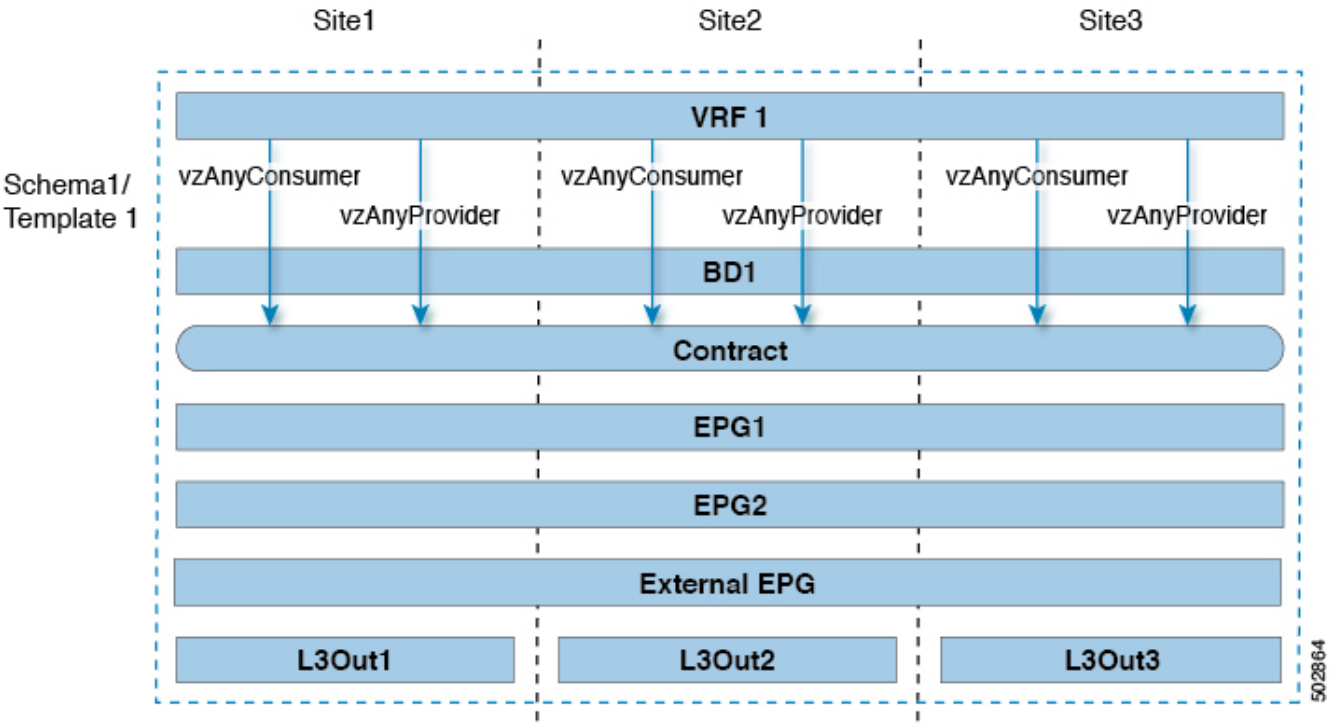
Stretched EPGs

The following example shows intra-VRF communication between EPGs or External EPGs all of which are stretched between fabrics. 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 fabrics.



As a best practice, it is recommended that the L3Out objects should instead remain defined only on Cisco APIC or configured on-fabric local templates on Nexus Dashboard.



Stretched EPGs

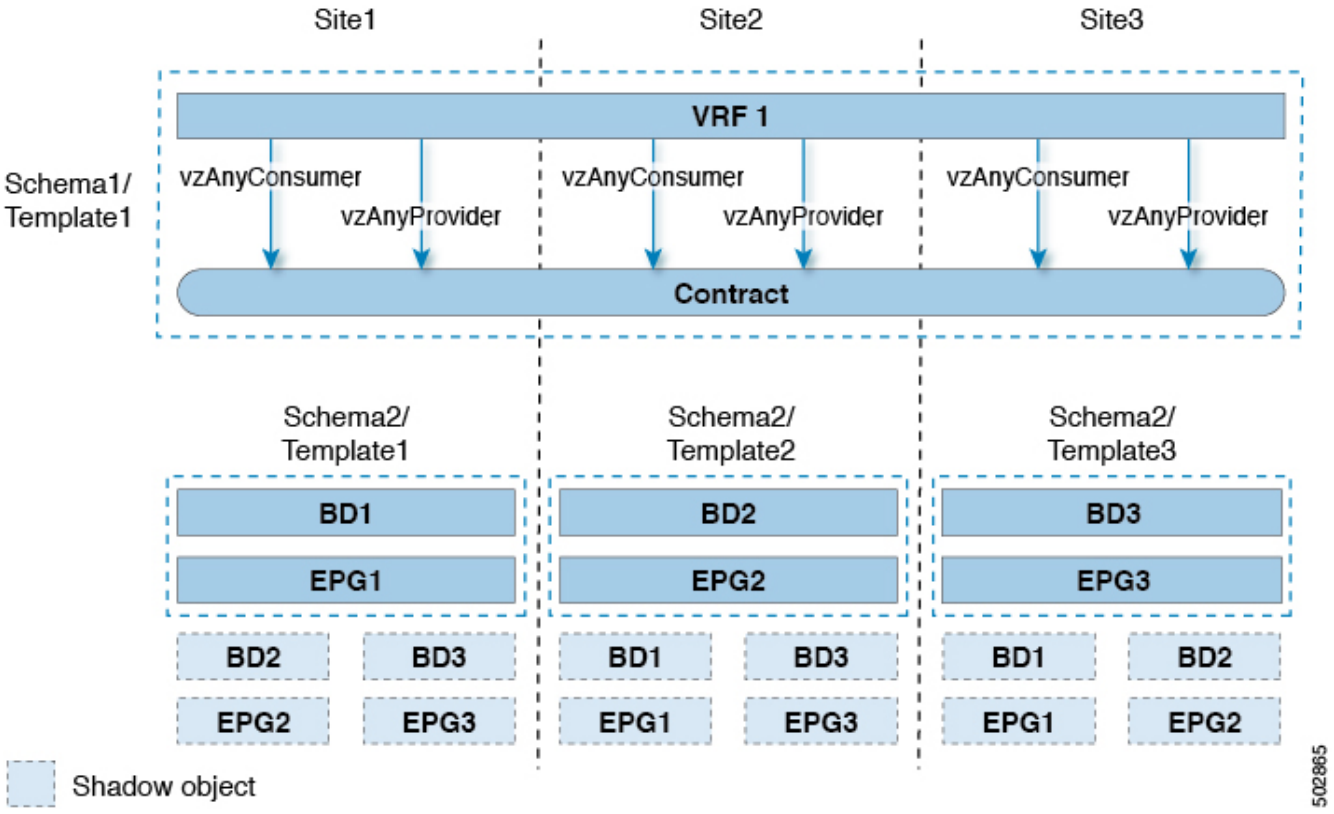
Fabric-Local EPGs

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 fabric.
- And a separate template for every fabric containing the EPG and BD deployed that fabric.

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



Fabric-Local EPGs

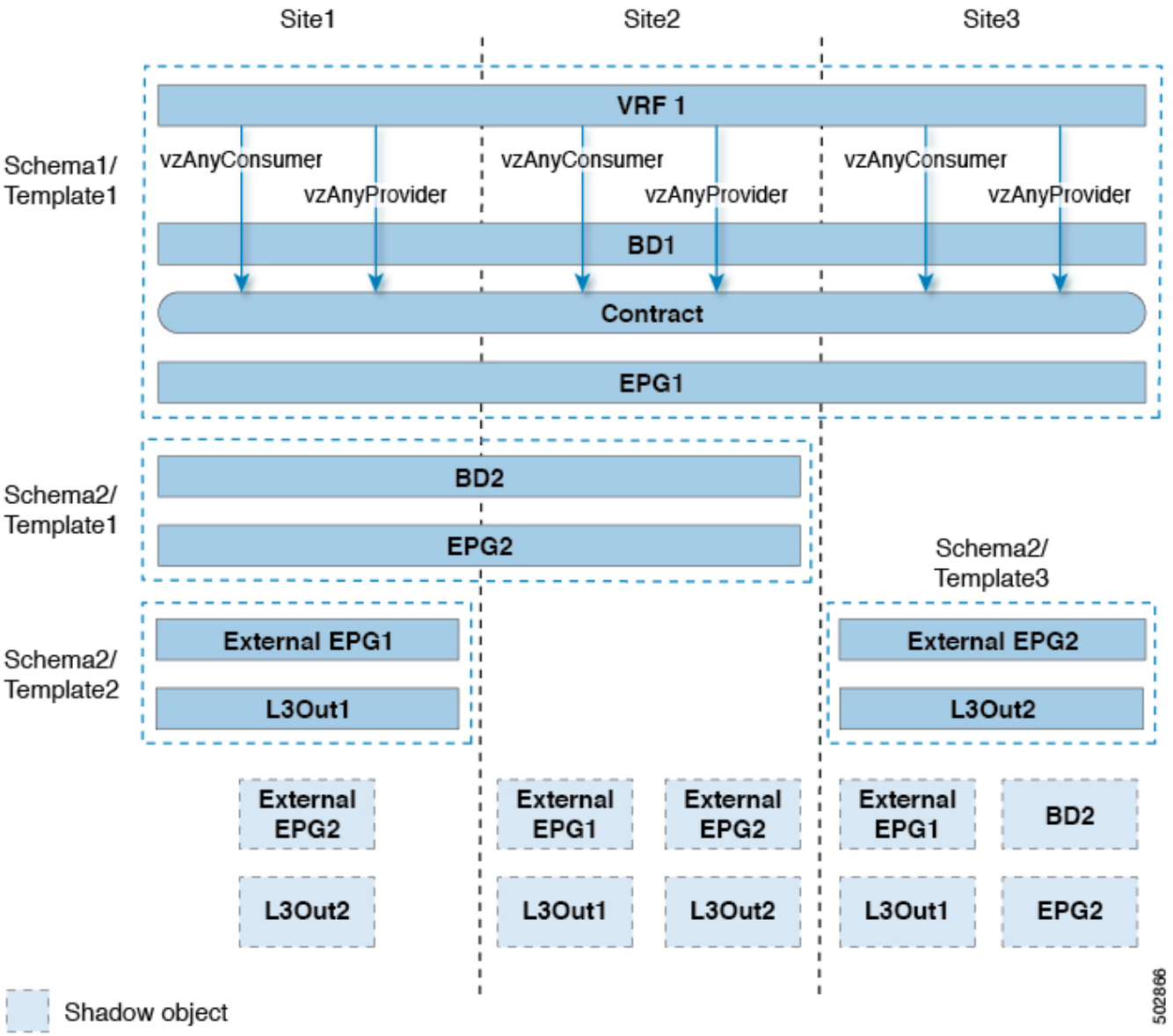
Combination of Fabric-Local and Stretched EPGs

The following example shows intra-VRF communication between EPGs or External EPGs where some EPGs are stretched while others are deployed to a single fabric 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 fabric.
- And a separate template for every fabric combination containing the objects deployed only to those fabrics.

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



Combination of Fabric-Local and Stretched EPGs

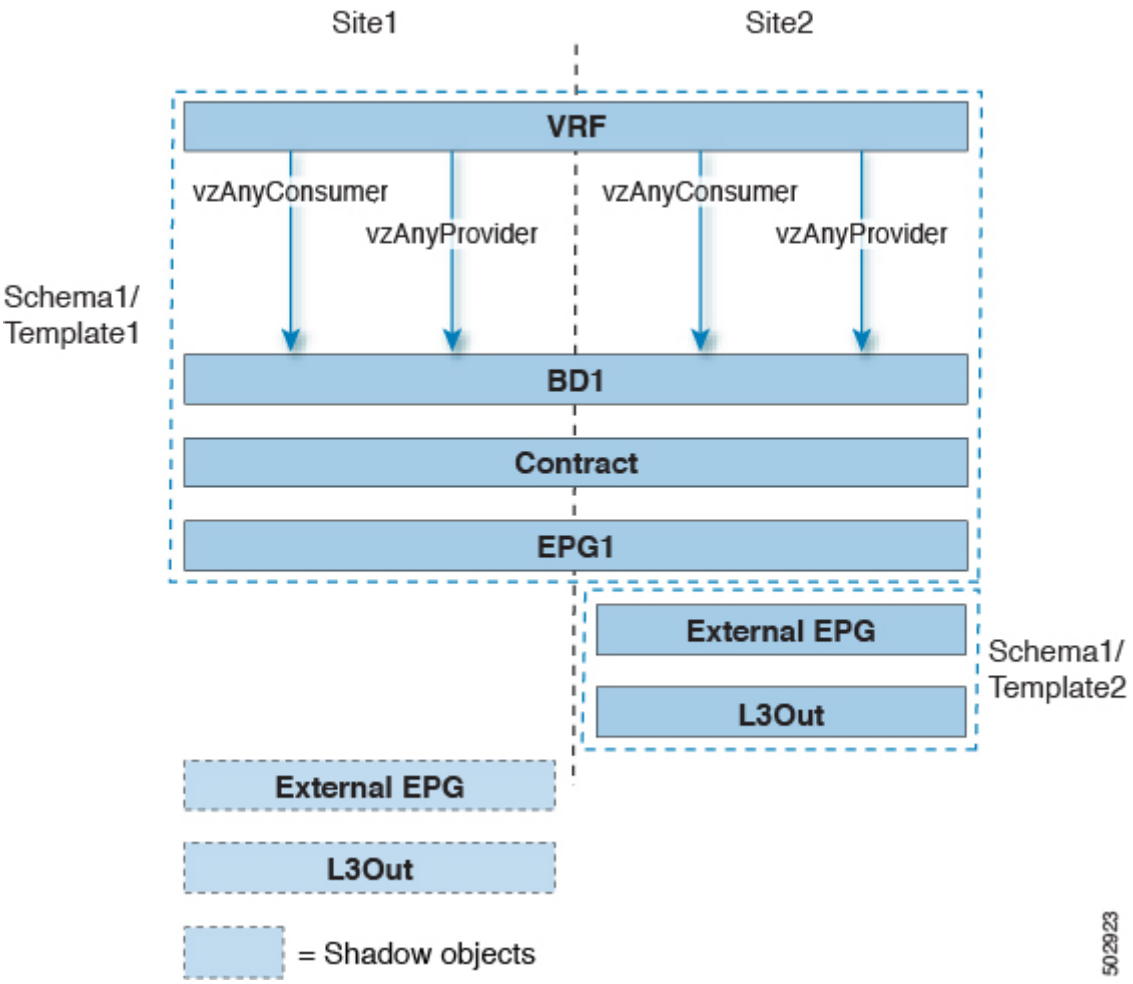
Intra-VRF Inter-Fabric L3Out

This use case allows you to configure an inter-fabric 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 inter-fabric L3Out, you must configure a routable TEP pool for each Pod. Additional inter-fabric L3Out details and requirements are described in the Inter-Fabric 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 fabrics.
- And a separate template for every fabric combination containing the objects deployed only to those fabrics.



Intra-VRF Inter-Fabric L3Out

Based on the configuration shown in the above figure, endpoints that are part of the stretched EPG1 and connected to Fabric1 will be able to communicate with the external network domain via the L3Out connection deployed in Fabric2. The same would apply for endpoints that are part of fabric-local EPGs deployed in Fabric1.

Inter-VRF Inter-Fabric L3Out

This use case allows you to enable vzAny contracts between a consumer VRF and L3Out external EPGs in a different provider VRF. Multiple EPGs that are part of the vzAny consumer VRF can communicate with a single EPG that is providing a shared service in a provide VRF. vzAny contract acts as a contract for all EPGs in a VRF. Each of the participating VRFs and L3Out external EPG can be stretched across fabrics.

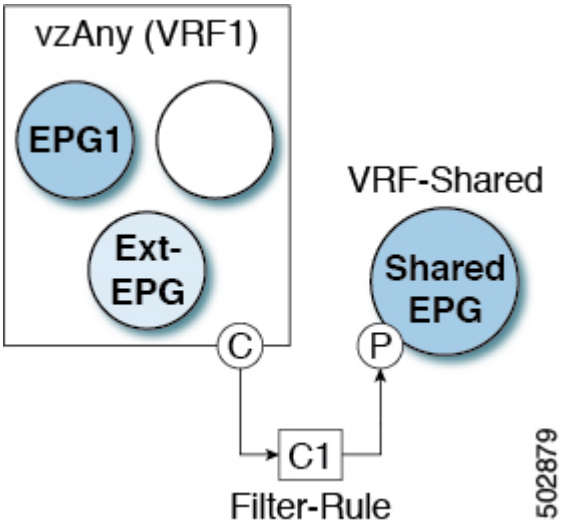


A VRF cannot be a vzAny provider.

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.



Many-to-One Communication

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 fabrics as well as which objects are going to be stretched. The specific sections below contain recommendation on template layout.

1. Create a Schema.
2. Create a common Template used to deploy configuration objects in all the fabrics (that is *stretched objects*).
3. Create any additional templates for every combination of fabrics where EPGs will be deployed .
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.

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.

6. Within each fabric'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.

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.

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.

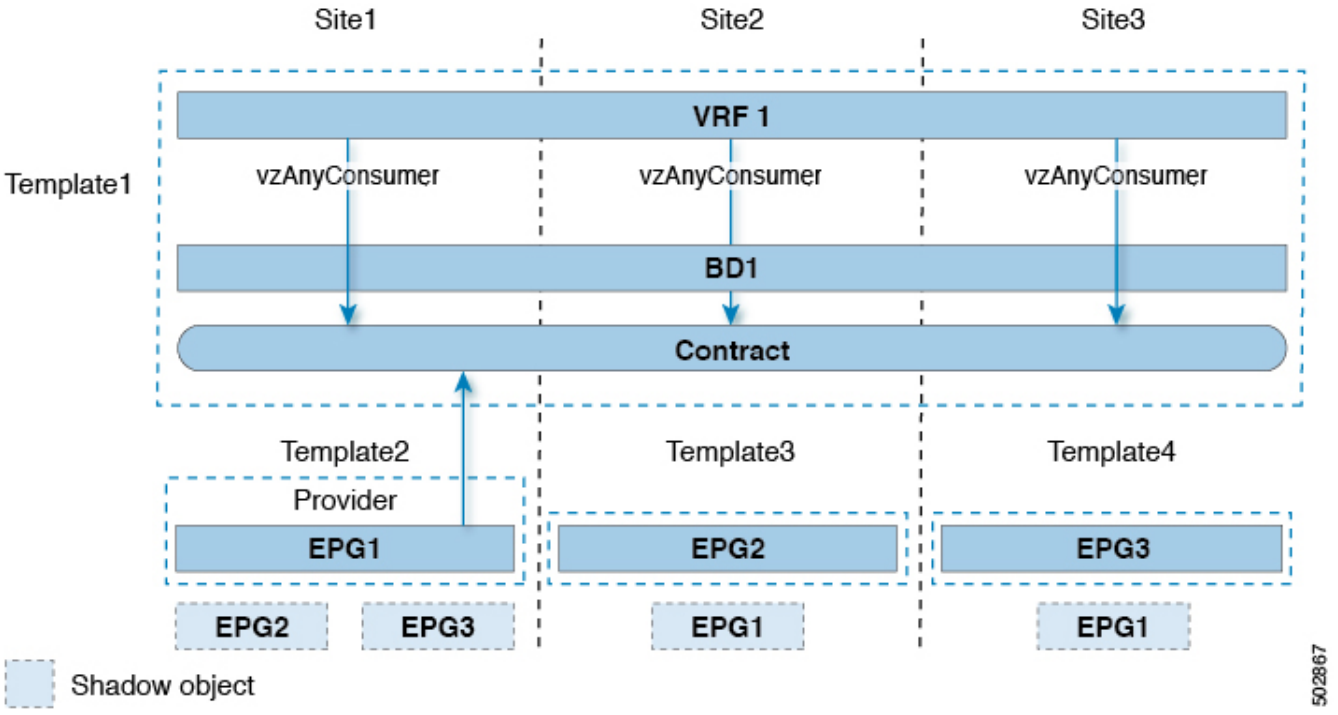
Provider EPG Within vzAny VRF

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 fabric.
- And a separate template for every fabric combination containing the objects deployed only to those fabrics.

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 fabrics.



Provider EPG Within vzAny VRF

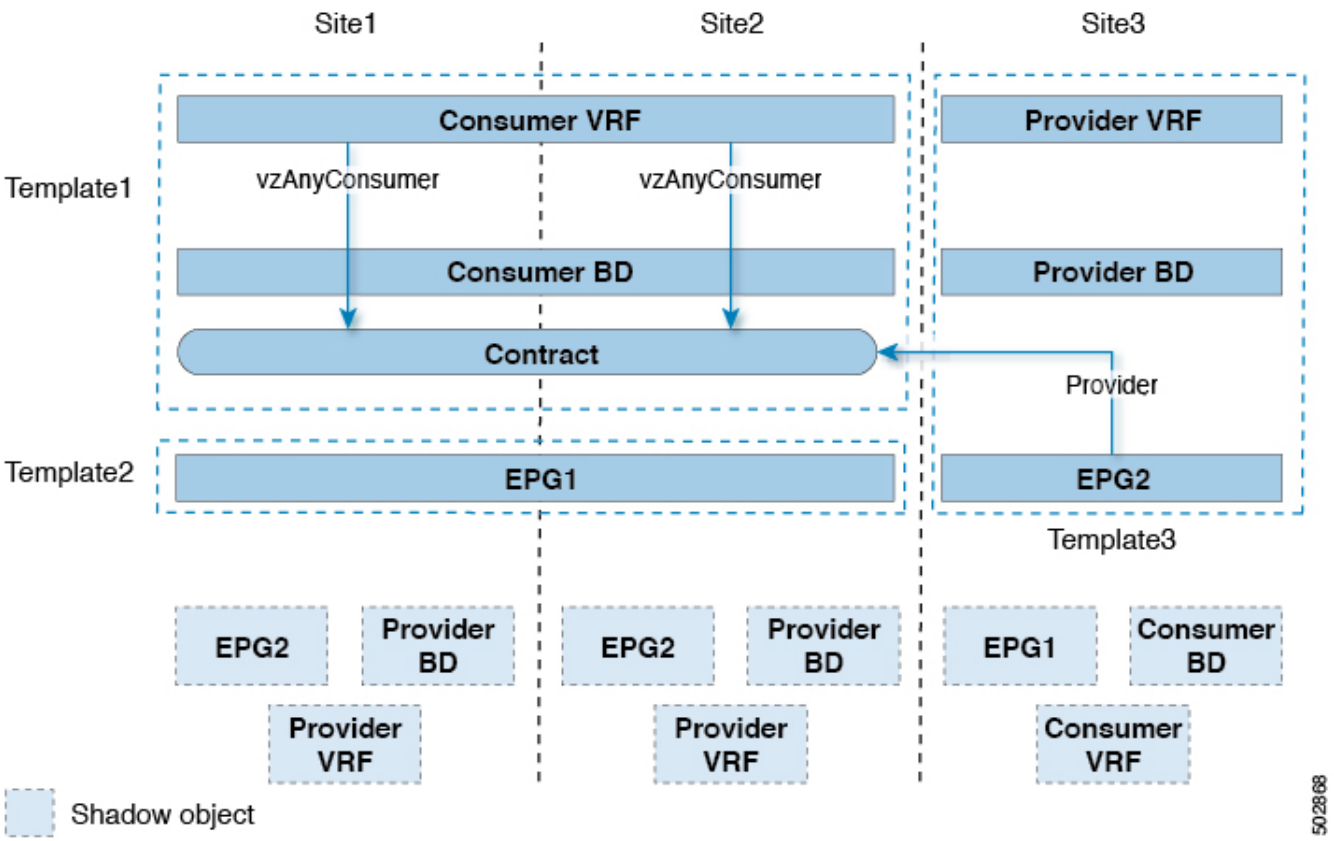
Provider EPG In Its Own VRF

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 fabric 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 fabrics.

- And a separate template for every fabric combination containing the objects deployed only to those fabrics.



Provider EPG In Its Own VRF

Copyright

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2017-2026 Cisco Systems, Inc. All rights reserved.

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706

USA
https://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883