White Paper

Principles of Application Centric Infrastructure

What You Will Learn

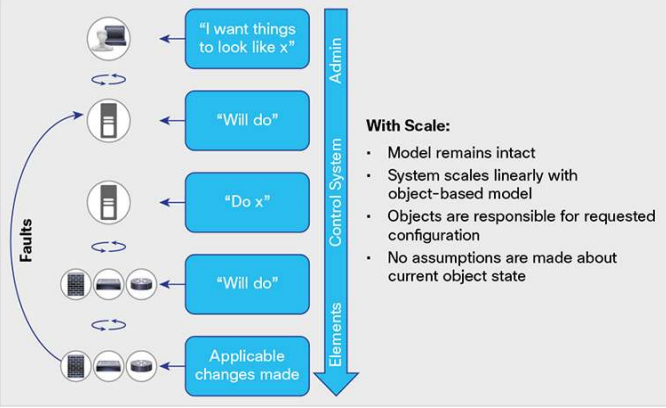
One of the main innovations in application centric infrastructure (ACI) is the introduction of a highly abstracted interface to express the connectivity of application components along with high-level policies governing that connectivity. The model was designed to be simple for application developers to use while simultaneously improving automation and security.

ACI Policy Theory

The ACI policy model is an object-oriented model based on promise theory. Promise theory is based on scalable control of intelligent objects rather than more traditional imperative models, which can be thought of as a top-down management system. In this system, the central manager must be aware of both the configuration commands of underlying objects and the current state of those objects.

Promise theory, in contrast, relies on the underlying objects to handle configuration state changes initiated by the control system itself as “desired state changes.” The objects are then responsible for passing exceptions or faults back to the control system. This approach reduces the burden and complexity of the control system and allows greater scale. This system scales further by allowing the methods of underlying objects to request state changes from one another and from lower-level objects (Figure 1).

Figure 1. Promise Theory Approach to Large-Scale System Control



Within this theoretical model, ACI builds an object model for the deployment of applications, with the applications as the central focus. Traditionally, applications have been restricted by the capabilities of the network and by requirements to prevent misuse of the constructs to implement policy. Concepts such as addressing, VLAN, and security have been tied together, limiting the scale and mobility of the application. As applications are being redesigned for mobility and web scale, this traditional approach hinders rapid and consistent deployment.

The ACI policy model does not dictate anything about the structure of the underlying network. However, as dictated by promise theory, it requires some edge element, called an iLeaf, to manage connections to various devices.

Object Model

At the top level, the ACI object model is built on a group of one or more tenants, allowing the network infrastructure administration and data flows to be segregated. Tenants can be used for customers, business units, or groups, depending on organizational needs. For instance, an enterprise may use one tenant for the entire organization, and a cloud provider may have customers that use one or more tenants to represent their organizations.

Tenants can be further divided into contexts, which directly relate to Virtual Routing and Forwarding (VRF) instances, or separate IP spaces. Each tenant can have one or more contexts, depending on the business needs of that tenant. Contexts provide a way to further separate the organizational and forwarding requirements for a given tenant. Because contexts use separate forwarding instances, IP addressing can be duplicated in separate contexts for multitenancy.

Within the context, the model provides a series of objects that define the application. These objects are endpoints (EP) and endpoint groups (EPGs) and the policies that define their relationship (Figure 2). Note that policies in this case are more than just a set of access control lists (ACLs) and include a collection of: inbound and outbound filters, traffic quality settings, marking rules, and redirection rules.

Figure 2. Logical Object Model

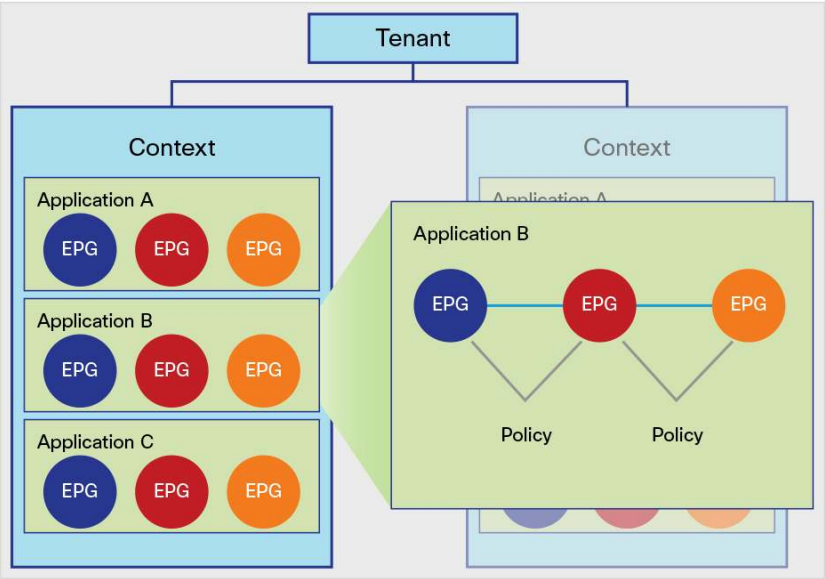


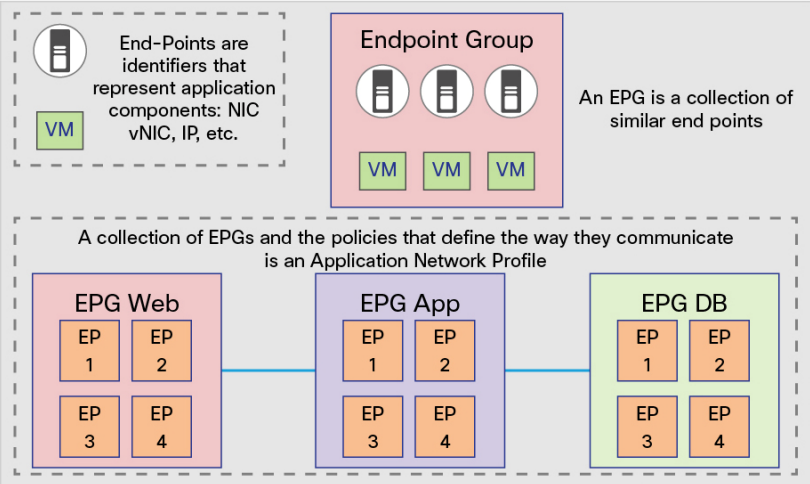
Figure 2 shows a tenant with two contexts and the applications that make up those contexts. The EPGs shown are groups of endpoints that make up an application tier or other logical application grouping. For example, Application B, shown expanded on the right side of the figure, may consist of a web tier (blue), application tier (red), and database tier (orange). The combination of EPGs and the policies that define their interaction is an Application Network Profile in the ACI model.

Endpoint Groups

EPGs are a collection of similar endpoints representing an application tier or set of services. They provide a logical grouping of objects that require similar policy. For example, an EPG could be the group of components that make up an application's web tier. Endpoints are defined using the network interface card (NIC), virtual NIC (vNIC), IP address, or Domain Name System (DNS) name, with extensibility to support future methods of identifying application components.

EPGs are also used to represent entities such as outside networks, network services, security devices, and network storage. EPGs are collections of one or more endpoints that provide a similar function. They are a logical grouping with a variety of use options, depending on the application deployment model in use (Figure 3).

Figure 3. Endpoint Group Relationships



EPGs are designed for flexibility, allowing their use to be tailored to one or more deployment models that the customer can choose. The EPGs are then used to define the elements to which policy is applied. Within the network fabric, policy is applied between EPGs, therefore defining the way that EPGs communicate with one another. This approach is designed to be extensible in the future to policy application within the EPGs.

Here are some examples of EPG use:

- EPG defined by traditional network VLANs: All endpoints connected to a given VLAN placed in an EPG
- EPG defined by Virtual Extensible LAN (VXLAN): Same as for VLANs except using VXLAN
- EPG mapped to a VMware port group
- EPG defined by IP or subnet: for example, 172.168.10.10 or 172.168.10
- EPG defined by DNS names or DNS ranges: for instance, example.foo.com or *.web.foo.com

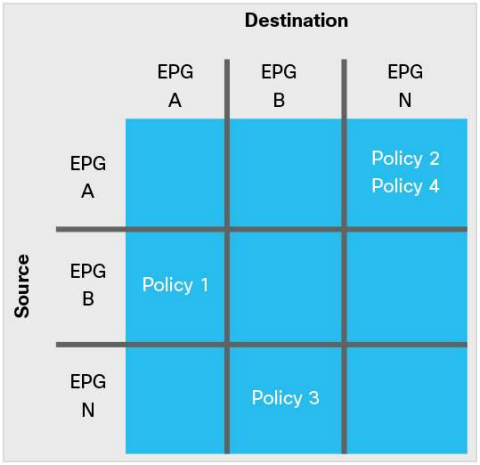
The use of EPGs is both flexible and extensible. The model is intended to provide tools to build an application network model that maps to the actual environment's deployment model. The definition of endpoints also is extensible, providing support for future product enhancements and industry requirements.

The EPG model offers a number of management advantages. It offers a single object with uniform policy to higher-level automation and orchestration tools. Tools need not operate on individual endpoints to modify policies. Additionally, it helps ensure consistency across endpoints in the same group regardless of their placement in the network.

Policy Enforcement

The relationship between EPGs and policies can be thought of as a matrix with one axis representing the source EPG (sEPG) and the other representing the destination EPG (dEPG.) One or more policies will be placed at the intersection of the appropriate sEPGs and dEPGs. The matrix will be sparsely populated in most cases because many EPGs have no need to communicate with one another Figure 4).

Figure 4. Policy Enforcement Matrix



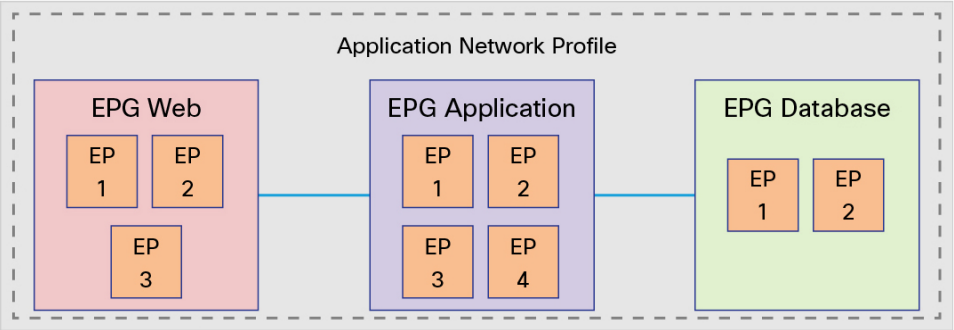
Policies are divided by filters for quality of service (QoS), access control, service insertion, etc. Filters are specific rules for the policy between two EPGs. Filters consist of inbound and outbound rules: permit, deny, redirect, log, copy, and mark.

Application Network Profiles

An Application Network Profile is a collection of EPGs, their connections, and the policies that define those connections. Application Network Profiles are the logical representation of an application and its interdependencies in the network fabric.

Application Network Profiles are designed to be modeled in a logical way that matches the way that applications are designed and deployed. The configuration and enforcement of policies and connectivity is handled by the system rather than manually by an administrator. Figure 6 shows an example of an access profile.

Figure 5. Application Network Profiles



These general steps are required to create an Application Network Profile:

1. Create EPGs (as discussed earlier).
2. Create policies that define connectivity with these rules:
   - Permit
   - Deny
   - Log
   - Mark
   - Redirect
   - Copy
3. Create connection points between EPGs using policy constructs known as contracts.

Contracts

Contracts define inbound and outbound permit, deny, and QoS rules and policies such as redirect. Contracts allow both simple and complex definition of the way that an EPG communicates with other EPGs, depending on the requirements of the environment. Although contracts are enforced between EPGs, they are connected to EPGs using provider-consumer relationships. Essentially, one EPG provides a contract, and other EPGs consume that contract.

The provider-consumer model is useful for a number of purposes. It offers a natural way to attach a “shield” or “membrane” to an application tier that dictates the way that the tier interacts with other parts of an application. For example, a web server may offer HTTP and HTTPS, so the web server can be wrapped in a contract that allows only these services. Additionally, the contract provider-consumer model promotes security by allowing simple, consistent policy updates to a single policy object rather than to multiple links that a contract may represent. Contracts also offer simplicity by allowing policies to be defined once and reused many times (Figure 6).

Figure 6. Contracts

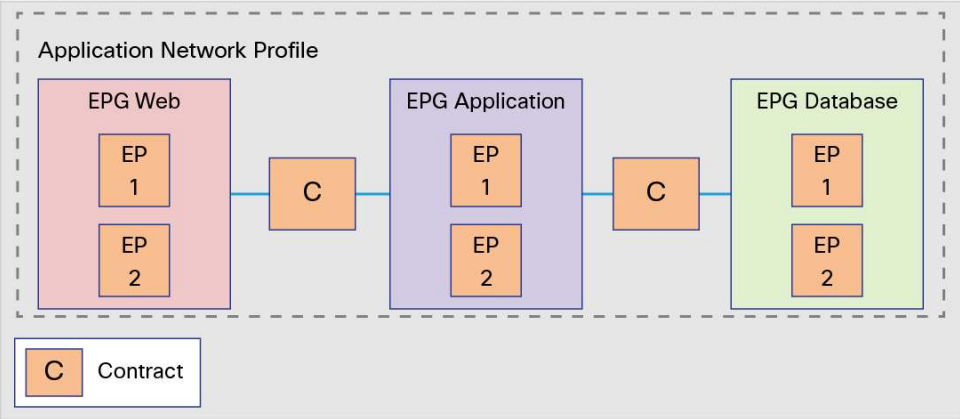
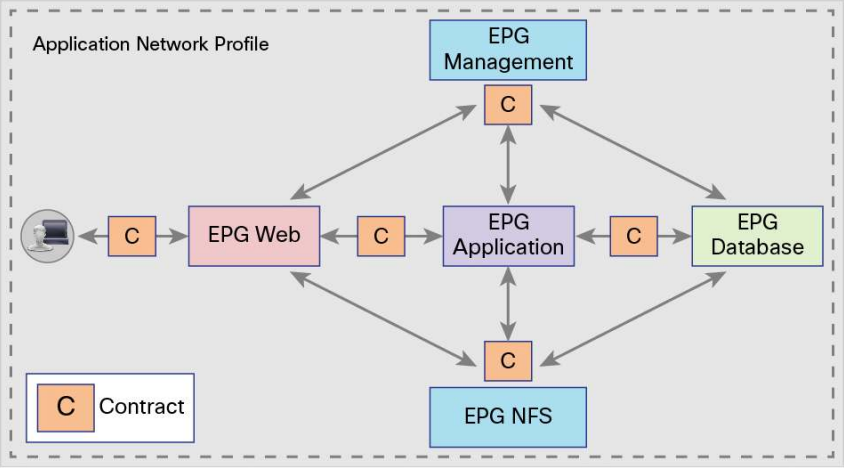


Figure 7 shows the relationship between the three tiers of a web application defined by EPG connectivity and the contracts that define their communication. The sum of these parts constitutes an Application Network Profile. Contracts also provide reusability and policy consistency for services that typically communicate with multiple EPGs.

Figure 7. Complete Application Network Profile



Conclusion

This document offers only an introduction to the ACI policy model: discussing what ACI is and how its policy model can be used. This model includes a number of other constructs and objects that, for simplicity, are not covered here.

For More Information

Please see http://www.cisco.com/go/aci.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA C11-729906-00 11/13