White Paper

A Comparison of OpFlex and OVSDB: The Benefits of an Application Policy Language in Cisco ACI

What You Will Learn

IT departments and lines of business are looking at cloud automation tools and software-defined networking (SDN) architecture to accelerate application delivery, reduce operating costs, and increase business agility. The success of an IT or cloud automation solution depends largely on the business policies that can be carried out by the infrastructure through the SDN architecture.

Through a detailed comparison of critical architectural components, this document shows how the Cisco® Application-Centric Infrastructure (ACI) architecture supports a more business-relevant application policy language, greater scalability through a distributed enforcement system rather than centralized control, and greater network visibility than alternative software overlay solutions or traditional SDN designs.

Introduction: Trends and Requirements for Cloud Automation

Historically, IT departments have been seeking greater automation as device proliferation has accelerated to overcome the challenges of applying manual processes for critical tasks. About 20 years ago the automation of desktop and PC management was the focus, and about 10 years ago server automation became important as applications migrated to larger numbers of modular x86 and RISC-based systems. Today, with the consolidation of data centers, IT must address not only application and data proliferation but also the emergence of cloud deployment models, focusing IT now on cloud and network automation.

The emergence of SDN promised a new era of centrally managed, software-based automation tools that could accelerate network management, optimization, and remediation. Gartner has defined SDN as "a new approach to designing, building and operating networks that focuses on delivering business agility while lowering capital and operational costs[1]."

Furthermore, Gartner, in an early 2014 report, notes that "SDN is a radical new way of networking and requires senior infrastructure leaders to rethink traditional networking practices and paradigms[2]." In this same report, Gartner makes an initial comparison of mainstream SDN solutions that are emerging, including VMware NSX, and Cisco ACI. There has been some discussion whether Cisco ACI is an SDN solution or something more, but most agree that, in a broad sense, the IT automation objectives of SDN and Cisco ACI are basically the same, and some of the baseline architectural features, including a central policy controller, programmable devices, and use of overlay networks, lead to a useful comparison.

This document focuses on the way that Cisco ACI expands traditional SDN methodology with a new application-centric policy model. It specifically compares critical protocols and components in Cisco ACI with VMware NSX to show the advantages of Cisco ACI over purely software overlay networks and the advantages of the Cisco application policy model over what has been offered by any prior SDN solution. It also discusses what the Cisco solution means for customers, the industry, and the larger SDN community.

[1] "Ending the Confusion About Software-Defined Networking: A Taxonomy," Gartner, March 2013
[2] "Mainstream Organizations Should Prepare for SDN Now," Gartner, March 2014

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

Requirements for a Policy-Based Infrastructure Automation Solution

Businesses and IT want to deploy, scale, and optimize applications on demand, in minutes, not weeks, to increase business agility and lower operating costs. SDN and related solutions propose to accomplish this through programmatic extensions to the network, so that IT and business policies can for the first time be implemented in software, and the software can automate network administration tasks, reducing the need for manual operations. Programmatic automation of network administration also appeared initially to be the only way to achieve cloud-level scale, supporting tens of thousands of servers and millions of workloads and migrating them on demand to optimize resource utilization.

Although traditional SDN overlay technology and the more recent Cisco ACI have important differences, both address fundamental architectural requirements for policy-based IT infrastructure automation:

- Centralized policy store and infrastructure controller: In SDN and Cisco ACI, this feature is generally known as the controller (Cisco Application Policy Infrastructure Controller [APIC] for Cisco ACI).
- Programmable, or automated, network devices: All infrastructure devices, such as servers and network nodes, must be able to respond to and implement policies according to commands from the controller. This feature may involve agents running on the device, APIs in the devices themselves, or management hooks to the devices that are implemented in the controller.
- Controller southbound protocol to communicate with the managed or controlled devices and to communicate policy information: Initially, the OpenFlow protocol was used in SDN architectures, and vendors released OpenFlow-compliant switches. In Cisco ACI, OpFlex is the primary protocol used, although other mechanisms for integrating devices into the Cisco ACI policy model are supported. For virtual overlay networks, OpenFlow has also been complemented with the Open vSwitch Database (OVSDB) management protocol to control virtual switches. This document explores the differences between OpFlex and OVSDB with OpenFlow to show the advantages of the Cisco ACI policy capabilities.
- Northbound controller interfaces for integrating higher-level automation solutions on top of the policy and controller framework, including workflow automation tools and analytics: Modern SDN controllers include northbound APIs allowing for the integration of OpenStack or other vendor-specific cloud automation tools. This document compares the capabilities allowed in the Cisco ACI policy model for these orchestration stacks to those of existing virtual overlay solutions.

Cisco ACI OpFlex Protocol: Advantages over Existing Alternatives

This section takes a closer look at some of these architectural components of Cisco ACI and the VMware NSX software overlay solution to quantify the advantages of Cisco's application-centric policies and demonstrates how the ACI architecture supports greater scale and greater IT automation.

As called for in the requirements listed in the previous section, Cisco ACI is an open architecture that includes the policy controller and policy repository (Cisco APIC), infrastructure nodes (network devices, virtual switches, network services, etc.) under Cisco APIC control, and a protocol communication between Cisco APIC and the infrastructure. For Cisco ACI, that protocol is OpFlex.

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

OpFlex was designed with the Cisco ACI policy model and cloud automation objectives in mind, including important features that other SDN protocols could not deliver. OpFlex supports the Cisco ACI approach of separating the application policy from the network and infrastructure, but not the control plane itself[3]. This approach provides the desired centralization of policy management, allowing automation of the entire infrastructure without limiting scalability through a centralized control point or creating a single point of catastrophic failure. Through Cisco ACI and OpFlex, the control engines are distributed, essentially staying with the infrastructure nodes that enforce the policies.

This design makes sense when you consider that the ways that devices are configured and policies are enforced in network nodes vary considerably across vendors and models. These details would introduce complexity into the Cisco ACI policy model or result in a policy that does not take full advantage of each device's full capabilities. The likely result would be a proprietary system with controlled devices modeled on a small number of device types or a single vendor's products.

OpFlex, in contrast, uses high-level application-oriented policy statements sent to network nodes, and it relies on those devices to determine how to implement the policies. After device configurations are completed, such as configuration of an application delivery controller (ADC) to support a particular service or configuration of a firewall with the proper policies between two tiers (virtual machines) of an application, communication between Cisco APIC and the infrastructure may be required only in the event of policy updates or fault scenarios.

OpFlex also is a bidirectional communication protocol. In addition to communication in the form of commands from the controller to the managed devices, useful state, performance, and health metrics can be sent from network nodes to either Cisco APIC or an external analytics engine (also called an observer in the OpFlex specification). Within Cisco ACI, this capability is integral to the 360 degree network visibility across physical and virtual infrastructure and to the reporting of health scores on an application-by-application basis.

For a more complete understanding of the OpFlex protocol, please refer to http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731302.html.

Compare the Cisco ACI model to the VMware NSX architecture, which uses OpenFlow as the communication protocol to an open virtual switch (OVS), along with the OVSDB management protocol. OVS is critical for forwarding traffic between virtual machines in the VMware NSX environment, and it is open to programmatic extension through OVSDB. OVS instances can then be managed from an SDN controller that supports OpenFlow and the OVSDB protocol.

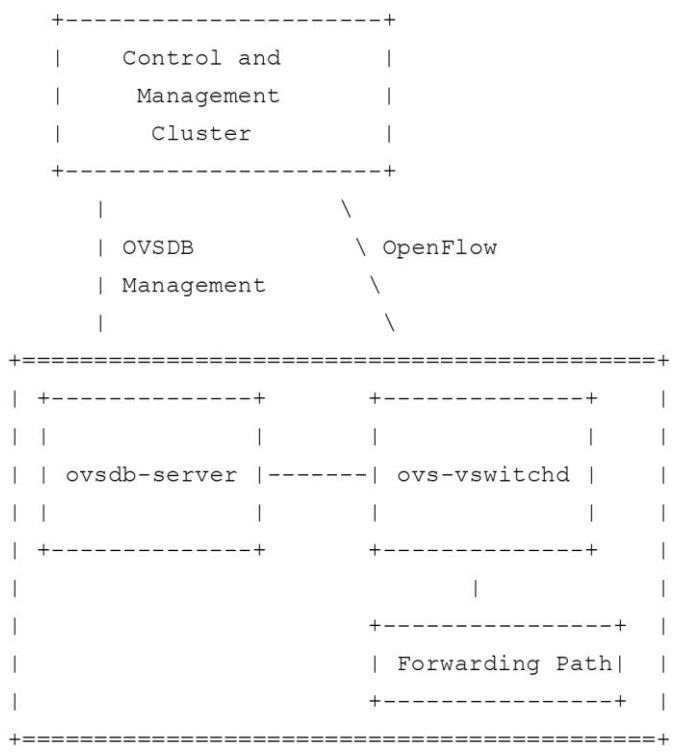
According to the OVSDB management protocol specification[4], the protocol is used to perform management and configuration operations on the OVS instance, and such operations occur "on a relatively long timescale" (taken to mean relatively infrequently). As a result, OVSDB does not perform management for individual data flows, leaving those instead to OpenFlow (Figure 1).

[3] Original SDN definitions included separation of control and data planes, although that architectural detail seems to have been relaxed as SDN solutions have evolved. Yet, as a result, some pragmatists point to this detail as ACI being a different class of solution that is not SDN, despite the fundamental similarities discussed earlier.

[4] "The Open vSwitch Database Management Protocol," IETF RFC 7047, B. Pfaff and B. Davie, VMware, ISSN 2070-1721 http://tools.ietf.org/html/draft-pfaff-ovsdb-proto-02

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

Figure 1. OVSDB Management Protocol and OpenFlow Combine to Support Virtual Switch Control (Source: OVSDB Management Protocol IETF Submission)



This design brings up a number of obvious comparisons to OpFlex and the Cisco ACI policy model. OVSDB management protocol and OpFlex are both used to communicate from a centralized policy controller to individual network nodes, and both are primarily used to configure devices, create connections, etc., and not to manage individual applications or traffic flows.

But whereas OpFlex and Cisco ACI rely on individual devices to carry out the policies based on the capabilities of the device, whether it is a switch, ADC, or firewall, VMware NSX and OVSDB require continuous management commands from the controller to the device through OpenFlow. As noted earlier, this requirement can seriously hinder scalability, and the whole network can fail if the controller and management cluster fails. The controller design also becomes more complicated because the controller must manage a lot of low-level device- and flow-specific activities. And because OVSDB is based on the OVS architecture, every device must have essentially the same characteristics or internal interfaces as a virtual switch, eliminating the capability of VMware NSX to apply policies to other classes of devices.

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

In a blog from June 12, 2014[5], Ivan Pepelnjak points out some of the limitations of OpenFlow for managing overlay virtual networks in this manner:

> **The first glitches: layer-2 gateways**
> Adding layer-2 gateways to overlay virtual networks reveals the first shortcomings of OpenFlow. Once the layer-2 environment stops being completely deterministic (layer-2 gateways introduce the need for dynamic MAC learning), the solution architects have only a few choices:
>
> - Perform dynamic MAC learning in the OpenFlow controller - all frames with unknown source MAC addresses are punted to the controller, which builds the dynamic MAC address table and downloads the modified forwarding information to all switches participating in a layer-2 segment. This is the approach used by NEC's ProgrammableFlow solution.
>
> Drawback: controller gets involved in the data plane, which limits the scalability of the solution.
> - Offload dynamic MAC learning to specialized service nodes, which serve as an intermediary between the predictive static world of virtual switching, and the dynamic world of VLANs. It seems NVP used this approach in one of the early releases.
>
> Drawback: The service nodes become an obvious chokepoint; an additional hop through a service node increases latency.
> - Give up, half-ditch OpenFlow, and implement either dynamic MAC learning in virtual switches in parallel with OpenFlow, or reporting of dynamic MAC addresses to the controller using a non-OpenFlow protocol (to avoid data path punting to the controller). It seems recent versions of VMware NSX use this approach.
>
> **The killer: distributed layer-3 forwarding**
> Every layer-2 overlay virtual networking solution must eventually support distributed layer-3 forwarding (the customers that matter usually want that for one reason or another). Regardless of how you implement the distributed forwarding, hypervisor switches need ARP entries (see this blog post for more details), and have to reply to ARP queries from the virtual machines.
>
> > Even without the ARP proxy functionality, someone has to reply to the ARP queries for the default gateway IP address.
>
> ARP is a nasty beast in an OpenFlow world - it's a control-plane protocol and thus not implementable in the pure OpenFlow switches. The implementers have (yet again) two choices:
>
> - Punt the ARP packets to the controller, which yet again places the OpenFlow controller in the forwarding path (and limits its scalability)
> - Solve layer-3 forwarding with a different tool (approach used by VMware NSX and distributed layer-3 forwarding in OpenStack Icehouse)
>
> **Do We Really Need OpenFlow?**
> With all the challenges listed above, does it make sense to use OpenFlow to control overlay virtual networks? Not really. OpenFlow is like a Swiss Army knife… - it can solve many problems, but is not ideal for any one of them.

[5] "Is OpenFlow the Best Tool for Overlay Virtual Networks?", Ivan Pepelnjak, http://blog.ipspace.net/2014/06/is-openflow-best-tool-for-overlay.html

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

It's clear that OpFlex and the Cisco ACI policy model provide more flexibility, greater resiliency, and the possibility of much more robust cloud and application automation policy across a wider range of network infrastructure than the overlay network approach used by VMware NSX.

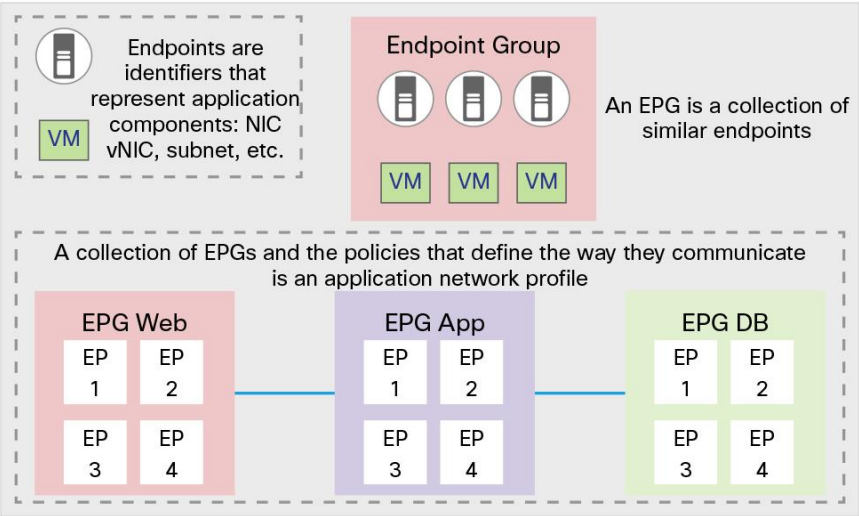
Benefits of an Application Policy Model

The Cisco ACI fabric is designed as an application-centric intelligent network. The Cisco APIC policy model is defined from the top down as a policy enforcement engine focused on the application itself and abstracting the networking functions underneath. The policy model unites with the advanced hardware capabilities of the Cisco ACI fabric underlying the business-application-focused control system.

The Cisco APIC policy object-oriented model is built on the distributed policy enforcement concepts for intelligent devices enabled by OpFlex and characterized by modern development and operations (DevOps) applications such as Puppet and Chef.

At the top level, the Cisco APIC policy model is built on a series of one or more tenants, which allows the network infrastructure administration and data flows to be segregated. Tenants can be customers, business units, or groups, depending on organization needs. Below tenants, the model provides a series of objects that define the application itself. These objects are endpoints and endpoint groups (EPGs) and the policies that define their relationships (Figure 2). The relationship between two endpoints, which might be two virtual machines connected in a three-tier web application, can be implemented by routing traffic between the endpoints to firewalls and ADCs that enforce the appropriate security and quality of service (QoS) policies for the application and those endpoints.

Figure 2. Endpoints and Application Workloads Along with Tenants and Application Network Profiles Are the Foundation of the Cisco ACI Policy Model



For a more thorough description of the Cisco ACI application policy model, please refer to http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731310.html.

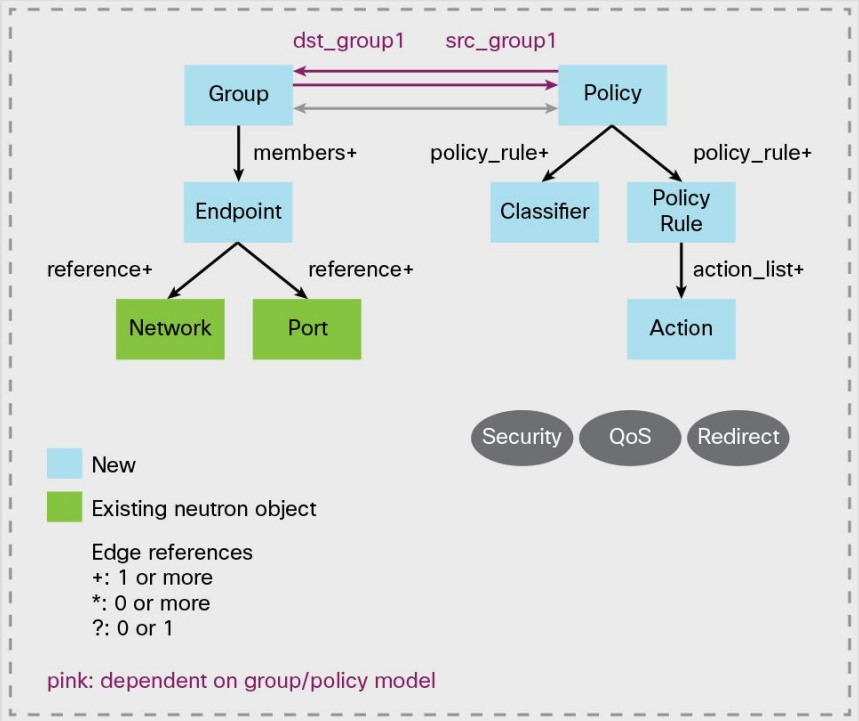
© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

For this discussion, the important feature to notice is the way that Cisco ACI policies are applied to application endpoints (physical and virtual workloads) and to EPGs. Configuration of individual network devices is ancillary to the requirements of the application and workloads. Individual devices do not require programmatic control as in prior SDN models, but are orchestrated according to the centrally defined and managed policies and according to application policies.

This model is catching hold in the industry and in the open source community. The OpenStack organization has begun work on including group-based policies to extend the OpenStack Neutron API for network orchestration with a declarative[6] policy-based model based closely on EPG policies from Cisco ACI[7].

Figure 3 (from the OpenStack website) shows how the group policy model extends the existing OpenStack networking concepts of networks and ports by applying the network objects to a new classifier, called the endpoint, with endpoints then further classified into groups. Policies are applied to endpoint groups, with policies consisting of classifiers, rules, and actions.

Figure 3. Group Policy Taxonomy in OpenStack
(Source: OpenStack)



[6] "Declarative" refers to the orchestration model in which control is distributed to intelligent devices based on centralized policies, in contrast to retaining per-flow management control within the controller itself.
[7] https://wiki.openstack.org/wiki/Neutron/GroupPolicy

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

As stated in the original group policy proposal[8]:

> The main advantage of the extensions described in this blueprint is that they allow for an interface to Neutron which is more application-centric than the existing Neutron APIs. For example, the current Neutron API is focused on very network-centric constructs: ports, networks, subnets, routers, and security groups. In the context of networking, these make complete sense. But in the context of cloud applications, these are more cumbersome than needed. Application developers think in different terms - the policy and group abstractions are designed to allow for the flexibility that an application developer may want when programming something like Neutron.
>
> The goal of these API extensions is that they become the main interface to Neutron for those deploying applications by providing a simpler interface in which to consume Neutron resources.

Similarly, the OpenDaylight open source controller group is working on specifying a group policy plug-in for the OpenDaylight controller. In this case, in addition to adopting the same policy model focused on groups of endpoints, the controller working group is specifying northbound APIs from the controller to accept abstract policy based on application requirements from orchestration tools and systems such as OpenStack, and to offer numerous southbound interfaces to allow network elements to be programmed and managed based on the application policies.

The OpenDaylight project webpage[9] lists the advantages of the group policy approach for SDN controllers:

- Easier, application-focused way to express policy: By creating policies that mirror application semantics, this framework provides a simpler, self-documenting mechanism for capturing policy requirements without requiring detailed knowledge of networking.
- Improved automation: Grouping constructs allow higher-level automation tools to easily manipulate groups of network endpoints simultaneously.
- Consistency: By grouping endpoints and applying policy to groups, the framework offers a consistent and concise way to handle policy changes.
- Extensible policy model: Because the policy model is abstract and not tied to specific network implementations, it can easily capture connectivity, security, Layer 4 through 7, QoS, etc.

As you can see, the open source SDN community is moving toward the group-based policy model inherent in Cisco ACI, which is enabled by the OpFlex protocol, as well as the policy language itself. These efforts are supported by a diverse community of vendors and solution providers, including IBM, Red Hat, Plexxi, Midokura, and Big Switch as well as the OpenDaylight participants. What is coming together is not just another set of SDN protocols or use cases, but an entire architecture for sharing policies and actionable data.

Extending Application Policies to DevOps

As noted earlier, modern DevOps applications such as Puppet, Chef, and CFEngine have already moved toward the declarative model of IT automation, so there is already some obvious synergy between DevOps and the Cisco ACI policy model. DevOps automation products are also optimizing application delivery processes and are designed to automate critical IT tasks to make the organization more agile and efficient.

[8] https://docs.google.com/document/d/1ZbOFxAoibZbJmDWx1oOrOsDcov6Cuom5aaBIrupCD9E/edit
[9] https://wiki.opendaylight.org/view/Project_Proposals:Application_Policy_Plugin

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

In an early 2014 blog post[10], Andi Mann, vice president of strategic solutions at CA Technologies, wrote about the evolution to DevOps and the synergy with the Cisco ACI policy model:

> Though the DevOps approach of today - with its notable improvements to culture, process, and tools - certainly delivers many efficiencies, automation and orchestration of hardware infrastructure has still been limited by traditional data center devices, such as servers, network switches and storage devices. Adding a virtualization layer to server, network, and storage, IT was able to divide some of these infrastructure devices, and enable a bit more fluidity in compute resourcing, but this still comes with manual steps or custom scripting to prepare the end-to-end application infrastructure and its networking needs used in a DevOps approach.
>
> The drag created by these traditional application infrastructures has been somewhat reduced by giving that problem to cloud providers, but in reality this drag never really went away until Cisco innovated application-centric programmability with Cisco ACI. This innovative new solution is now poised to greatly benefit the whole application economy, especially management of the DevOps application environment.
>
> **Cisco ACI and DevOps are the match made in heaven**
> Cisco ACI enables the automation needed for the many hardware infrastructure changes and application endpoint configurations that are essential in an application-centric, DevOps-style development environment. With Cisco ACI policy-driven network configuration, DevOps application environments are more closely controlled and aligned to application release automation needs.
>
> Cisco ACI multi-tenancy, along with context configurations, also creates a powerful partitioning and security mechanism to manage independent application environments. This enables rapid release for applications that require their own private environments and secure endpoint management, as part of the enterprise application DevOps lifecycle.
>
> Lastly, a key facet of DevOps is a closed-loop approach to iteration and quality improvement. It is therefore also very important to have real-time telemetry on application endpoints. Cisco ACI provides the ability to probe these application components for their network performance, and to accelerate virtualization of services with consistency and accuracy.

Conclusion

IT automation and cloud orchestration solutions are evolving from initial SDN programmability techniques to application-oriented group-based policy infrastructure. The Cisco ACI infrastructure uniquely supports the requirements of a group-based policy model through the:

- Declarative control model inherent in the OpFlex protocol
- Abstract policy language itself
- Focus on application of policy to endpoint groups
- Flexibility of incorporating a wide-range of network devices as policy endpoints to implement policy directives

[10] http://blogs.ca.com/devops/2014/01/26/enterprise-devops-and-cisco-application-centric-infrastructure-a-match-made-in-heaven/

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

Prior SDN solutions that focused on network protocols rather than application requirements and that incorporated centralized control on top of centralized policy management cannot match the scalability, flexibility, or integration of non-network devices into the policy model that Cisco's approach enables. OpenFlow and OVSDB are particularly limited in this regard, with their primary focus on managing the virtual switch and overlay environment only, and the requirement that the network control plane be centralized in the controller and management cluster.

With Cisco ACI leading the way as an initial implementation, Cisco is contributing the foundational elements of this new policy-based approach to other vendors and the open source community to guide the next-generation policy.

Orchestration tools, the open source community, and DevOps products are rapidly incorporating this new policy-based approach. This incorporation will enable development of value-added cloud automation suites and compatible alternative controller and networking devices that support the new application policy model.

For More Information

http://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.

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: 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)

Printed in USA C11-732292-00 08/14

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.