WHITE PAPER–JUNE 2019

# Service-defined Firewall Solution Architecture

vmware®

## Table of Contents

"Application deployment with virtual machines (VMs) made it easier, cheaper, and quicker to deploy in scale. Separating services within an application became a design principle adding stability, scale, and ease of development. This facilitated a new multi-tier application framework."

## Application Evolution

If the history of enterprise applications has taught us anything it is that application architectures are going to continue to change as software continues to drive innovation. This is only further supported by a little bit of forward exploration of what is considered cutting edge in software development. By and large the changes we have seen and will see in application architecture have happened over a relatively short period of time. As a whole, applications have gone from singular individual systems that combine hardware and software into much smaller, highly distributed components of services, functions, and abstractions. Figure 1 shows a conceptual history of application architecture evolution.

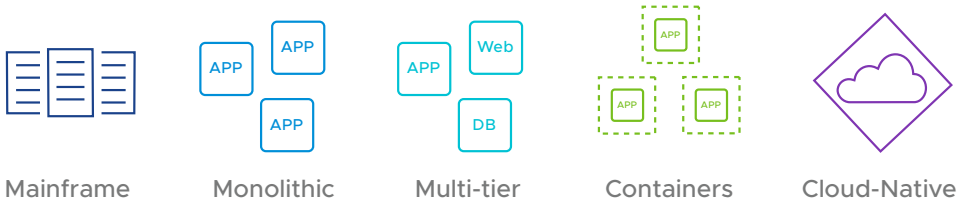


**Figure 1.** Application Architecture Evolution

Starting in 1950, mainframes performed repetitious tasks with limited interfaces that included paper tape, magnetic tape, and punch cards. Mainframes continued to evolve and in the 1970s included interactive user terminals and support for multiple users. But at the same time, the microprocessor became more widespread and started the revolution of personal computers. In the beginning, security was done through user access and not by the network. There was not a concern for external access. Access to the console provided access to the system. Role-based access control enforced security. Fast-forward to today, and mainframes are connected to the network and remain an integral part of many enterprise environments. The security strategy now is to protect mainframes by a DMZ strategy with physical firewalls.

In the 1980s, the rise of the personal computer sparked a new application architecture. Monolithic applications were developed for personal productivity (word processing, spreadsheets) and entertainment in the home. Perhaps two of the most famous of the monolithic applications to hit the enterprise space are Microsoft SQL Server (1989) and Microsoft Exchange Server (1996). Microsoft SQL Server ran on OS/2 until 1993, when it moved to Microsoft Windows NT. For almost a decade the monolithic app was a constant.

At this point, networking that connected these monolithic applications to users introduced a need for network security. Security strategies protected north-south traffic flows between users and applications with a physical firewall at the perimeter. A perimeter-only strategy became the primary source of network security for almost two decades.

This began the rise of the data center. Companies consolidated their compute and storage into a centralized location. Physical servers were dedicated to specific services. Application architecture remained largely client-server.

In the early 2000s, server virtualization introduced a new enterprise platform for server consolidation using virtual machines. Application deployment with virtual machines (VMs) made it easier, cheaper, and quicker to deploy at scale. Separating services within an application became a design principle adding stability, scale, and ease of development. This facilitated a new multi-tier application framework. For example, Microsoft Exchange Server expanded to include multiple tiers: presentation, logic, and data tiers.

As with Microsoft Exchange, many more vendor applications and even custom-developed applications adopted an n-tier architecture. With a growing number of n-tier applications across a shared infrastructure, east-west traffic between the tiers and applications

"Charles Darwin busted many of the notions people had about evolution when he published *Origin of the Species* in 1859. Darwin's findings proved the importance of adaptation to the survival of a species. His work showed that over time, as organisms encounter environmental conditions that jeopardize their survival they inherently build protection mechanisms to overcome the threat."

became a security concern. VLANs and PVLANs provided base-level segmentation for the shared infrastructure. Unfortunately, these strategies are more about networking than they are about security and didn't offer the applications the type of security they needed. Network security was still managed and enforced at the perimeter.

Even today, n-tier applications are a large part of enterprise application deployment. Adding to the complexity of these deployments is the ability to span data centers and clouds.

Further, a new application architecture sprang into action with the introduction of containers and cloud-native applications. With containers came the microservice architecture, which drastically increased the interconnectivity requirements of any given application and added to the need for east-west security.

Application development with containers is thriving in Amazon Web Services, Azure, Google Cloud Platform, VMware, and more.

These changes are not just evolutions in application architecture, but also adaptations to new technologies, business requirements, industry shifts, and consumer demands.

## Wearing Our Security

Why don't polar bears wear fluffy white winter coats and snow boots while carrying backpacks of protein bars?

Polar bears have adapted to their arctic conditions. They have changed physically and behaviorally to ensure their survival. Polar bears have thick white fur that allows them to blend in with snow and ice, along with a layer of fat to keep them warm. They use their fat as energy reserves during times when food is scarce. Polar bears also have large wide paws that make it easier to walk in snow.

Charles Darwin busted many of the notions people had about evolution when he published *Origin of the Species* in 1859. Darwin's findings proved the importance of adaptation to the survival of a species. His work showed that over time, as organisms encounter environmental conditions that jeopardize their survival, they inherently build protection mechanisms to overcome the threat. These mechanisms are not temporary or add-on solutions, but changes to the physical makeup of the organism.


**Figure 2.** The ecosystem of security companies has grown exponentially as part of the bolt-on security mindset.

The implicit story of figure 2 proves that the enterprise security response to application evolution has been to add a winter coat to a polar bear. As an industry, we have added on and retrofit security into the enterprise rather than inherently change the DNA of the enterprise infrastructure.

"Enterprise security design should revolve around a defense in depth strategy that meets the organization's needs, not just for security, but for availability, manageability, performance, and recoverability."

## Security Strategies

The collective complexity of application architectures within enterprises led to the creation of multiple defense-in-depth strategies.

These strategies range from deep architecture integration to surface-level inspections. These strategies support various use cases for which they offer a valid solution either on their own or in combination with another. They each have advantages and disadvantages when it comes to the amount of security they offer and their ease (or difficulty) of implementation and manageability. Enterprise security design should revolve around a defense-in-depth strategy that meets the organization's needs, not just for security, but for availability, manageability, performance, and recoverability.

It should be noted that some of the following strategies discussed are redefined by vendors to fit their own security products and initiatives. The descriptions listed below are a broad view of each security option as they relate to one another in a larger picture of enterprise security options. Enterprise architects must be able to look at every level of enterprise design and understand the options available for protection.

### Network Segmentation

As mentioned, VLANs and PVLANs provide a simple form of network segmentation, but are often misconstrued as security constructs. A VLAN represents an IP address space. By separating workloads onto separate VLANs (IP networks), a router and a defined IP gateway is then required to reach a different VLAN. If a workload doesn't need to communicate with any device on a different network, VLANs are an easy way to provide separation. They are limited to a maximum of 4094 VLANs, however, posing a problem too large for many network architectures.

PVLANs, or private virtual local area networks, provide a subdivision of a VLAN to allow or deny communication of devices on the same VLAN. PVLANs introduce a layer of complexity and challenges to large network design, as each port in the VLAN needs to be configured for a PVLAN setting. PVLANs can be configured as isolated, community or promiscuous. This configuration is performed per switch port and therefore is not ideal for dynamic environments. Isolated PVLAN ports can communicate only with ports configured as promiscuous. Community-configured PVLAN ports can communicate with other ports of the same community and promiscuous ports. Promiscuous PVLAN ports can communicate with any other ports.

VLANs and PVLANs, shown in Figure 3 and Figure 4, can solve simple network problems when it comes to allowing or denying communication at a network level. However, they lack any type of intelligence about the communication and can easily be manipulated.

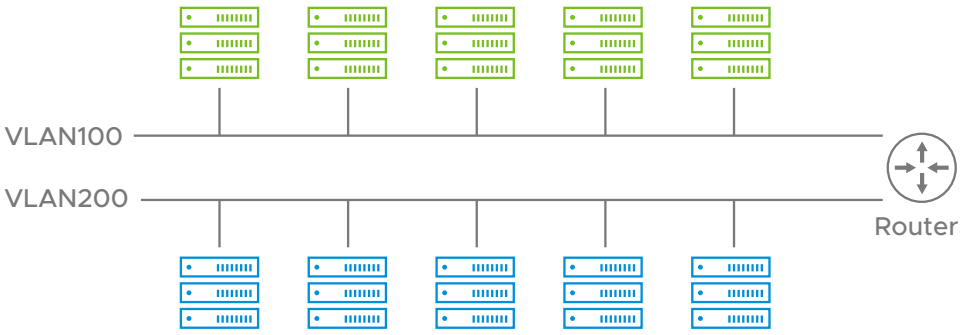


**Figure 3.** VLANs are a simple solution for placing devices on separate IP networks. Communication between VLANs requires a router.

"Some security vendors provide solutions to manage and control the native operating system firewall. However, these types of solutions lack isolation leaving the system vulnerable to attacks that disable the enforcement services."

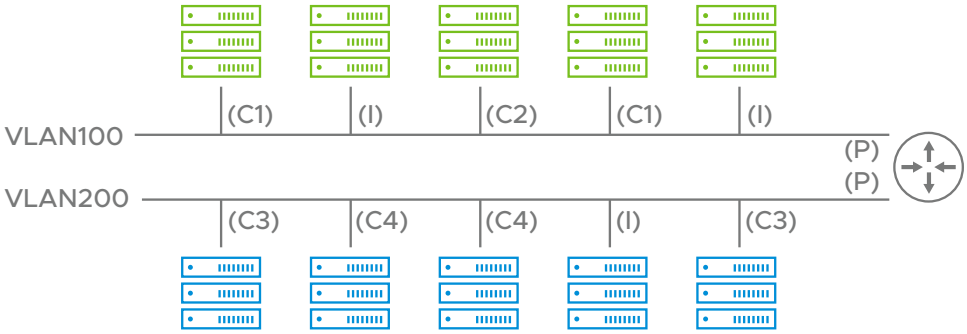


**Figure 4.** PVLANs allow devices on an IP network to be broken into subdomains. The subdomain configuration determines if devices are permitted to communicate.

VLANs and PVLANs are network constructs, not security constructs. The complexity of trying to achieve security goals with network constructs is why VLANs and PVLANs have not solved application security concerns for the evolving application.

### Macro-segmentation

Macro-segmentation is a high-level separation of objects, including but not limited to applications, clusters, and networks. Some examples include

- Segmenting production clusters, development clusters, and DMZ clusters
- Segmenting applications from one another with routers and/or firewalls

### Application Isolation

Application isolation, a form of macro-segmentation, is the construction of a security boundary around the workloads of an entire application. Application isolation can use both firewall features and network address translation to maintain control of traffic to any of the application workloads. Figure 5 shows the virtual machines of an application stack grouped logically and protected by a security boundary.

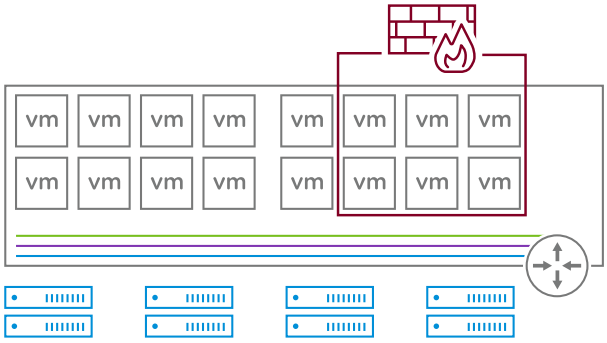


**Figure 5.** Application isolation builds a secure perimeter around a collection of systems that make up an application stack.

### Guest OS Firewall

The guest OS firewall, as the name suggests, is a firewall that resides inside the guest operating system. Although host-based firewalls have good contextual awareness of what is being protected, they are often difficult to manage and lack isolation. The Windows Firewall and Linux IPTables have been features of these two operating systems for more than a decade, though they are rarely used on a wide scale mainly because of operational complexity and lack of isolation. The enforcement of security inside the guest operating system is managed by internal services that can be disabled if the system is compromised. Some security vendors provide solutions to manage and control the native operating

"The extent of complexity for learning, enforcement and management hinges on the ability to use a cohesive set of tools that are integrated into the infrastructure. Using disparate tools to correlate data and implement security is too complex and error prone."

system firewall. However, these types of solutions lack isolation, leaving the system vulnerable to attacks that disable the enforcement services. These issues introduce a need for additional tools for monitoring, automation, and remediation. This added complexity is a continuation of the bolted-on security paradigm. Figure 6 shows guest operating system firewalls.

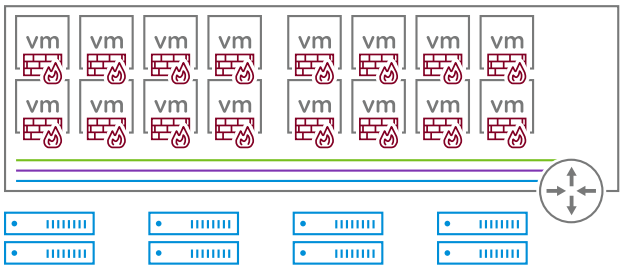


**Figure 6.** Guest operating system firewalls can be difficult to manage at scale and lack isolation from the system it is protecting. Attacks that disable the enforcement services leave the system vulnerable.

## Micro-segmentation

Micro-segmentation takes the concept of network macro-segmentation and applies enforcement of security (firewall) at a more granular level. Like the guest operating system firewall, each system has an individual security enforcement point, as shown in Figure 7. Micro-segmentation secures traffic within and between applications for a more detailed security strategy. This requires a deeper understanding of an application's communication patterns. To understand these patterns might require additional tools for learning and visibility. The extent of complexity for learning, enforcement, and management hinges on the ability to use a cohesive set of tools that are integrated into the infrastructure. Using disparate tools to correlate data and implement security is too complex and error prone.

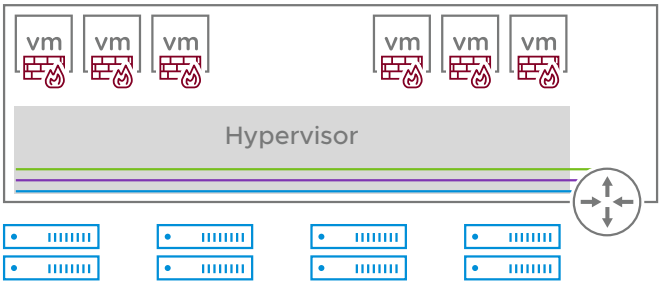


**Figure 7.** Micro-segmentation offers a more granular level of security policy enforcement within and between applications.

## Introspection

Introspection offers deeper insights into the purpose of network traffic. By integrating network introspection or deep packet inspection, network architects can verify that communication between devices is legitimate in its intention. Introspection, while offering a stronger security strategy, incurs additional overhead and could impact performance. Introspection is, therefore, not commonly used as a carte blanche security strategy across all workloads and devices. It is often reserved for a limited subset of workloads for which the impact is justified because of the highly sensitive nature of the data. For example, in a multi-tier application, introspection policies might be defined to perform a deeper packet inspection of traffic coming into the web tier from unknown, external sources. This additional overhead adds a stronger security posture to an external-facing web application, at the expense of performance. Scaling out the web tier helps overcome the performance impact that can hinder client experience.

"Using internal firewalls that enforce security in the hypervisor provides an ability to integrate with user identity mechanisms. This integration then allows user group memberships to be part of the characteristics for determining access to network resources."

### Whitelisting

Protecting a system from all known or potential threats is impossible, mostly because of potential threats that haven't even been created by those malicious individuals looking to attack. As much as the security landscape has evolved, the hacking landscape has also evolved, and will continue to evolve, even faster. Unattended existing vulnerabilities, new attach methods, and most of all, social engineering are always going to expose enterprise systems to threats. Rather than trying to understand, identify and prevent all of the known bad, security should focus on protecting the known good. The known good of an application is an exponentially smaller set of services, events, characteristics, and so on, than a stated list of known bad, and thus is easier to defend. Application owners working in alignment can identify the known-good function of an application and then leverage infrastructure security tools to ensure that the application operates only under those conditions. Applications running under a defined known good can then be configured with a predetermined response policy to any variation from the known good. This policy can range from blocking any further communication to simply notifying a responsible individual or team.

### Identity Firewall

The ability to leverage a user's identity to determine access is not new. At an enterprise level, this concept goes back to Novell's eDirectory and the more dominant Microsoft Active Directory. For a long time, we have used identity to determine access to data at the file and folder level. Identity then became part of a broader security strategy around authorization, authentication, and accounting. Application access was granted or denied based on identity, and that information could be logged for tracking purposes. Adding the ability to use identity as part of a firewall policy is where the novelty of identity comes through. Using internal firewalls that enforce security in the hypervisor provides an ability to integrate with user identity mechanisms. This integration then allows user group memberships to be part of the characteristics for determining access to network resources. Therefore, access and control to a network's resources can now be evaluated by interrogating the source system and the user initiating communications from that source. This is extremely useful for shared-client infrastructure technologies such as VDI and RDSH.

## Service-defined Firewall Design

Every enterprise design must account for five key qualities: availability, manageability, performance, recoverability, and security. Architects must understand the relative importance of each design quality against the other qualities. In addition, architects must know how stakeholders rank these design qualities. It is impossible to maximize all of the design qualities given that some of the qualities have inverse relationships; the more you get of one, the less you get of another. Take, for example, security and manageability. Security and manageability have an inverse relationship. The more secure you make something the more difficult it becomes to manage. Nothing that is more secure will ever be easier to manage than if you didn't have to secure it. Knowing these tradeoffs is critical to being able to design the correct defense in depth strategy for the enterprise.

Service-defined Firewall is a different paradigm in enterprise security. A Service-defined Firewall is built on the premise that security is an embedded part of an application or workload regardless of its location. A Service-defined Firewall is not a reinvention of security constructs, but instead is a better positioning of proactive security enforcement to ensure consistent security posture. This is a contradiction to the way security has been approached for decades. A 2018 report from Cyber Defenders noted that 80 percent of enterprise IT investments in security were put toward security tools that were reactive in nature. Along the same lines, the report also noted that 72 percent of venture capital

"VMware NSX provides a secure infrastructure platform that protects all workloads, from single system applications to virtual desktop environments, no matter where they reside."

investment in security start-ups went to reactionary product development. Clearly, the idea of chasing threats has somehow seemed more logical than the idea of reducing attack surface.

Building a reliable, scalable, and secure foundation for enterprise IT infrastructure must be a top priority. Whether it is private cloud, public cloud, or hybrid cloud, architects must focus on business goals and applications that are fundamental to the business. The days of designing infrastructure to support geographic needs have ended. Individualized security products that are bolted on to perform reactionary measures must be replaced with platforms that offer ubiquitous security across clouds. VMware NSX® has been developed from the ground up to offer a single-solutions architecture that creates secure infrastructure no matter the type of workload. VMware NSX provides a secure infrastructure platform that protects all workloads, from single system applications to virtual desktop environments, no matter where they reside.

At the beginning of this white paper, Figure 1 showed the evolution of application architecture from mainframe to cloud-native applications. In many cases, new applications frameworks are deployed alongside the legacy application architectures, forcing companies to manage different security models. The blend of application architectures makes it even more critical to find a security platform that offers proactive, core protection strategies rather than disparate, bolted-on security for each application type. As already noted, the design qualities of security and manageability have an inverse relationship. Using individualized tools pushes those qualities even further apart. Using a centralized security platform that is built into the infrastructure allows you to maximize both security and manageability.

The VMware Service-Defined Firewall isn't a product, or a feature; rather, it's a solution architecture designed specifically for mitigating threats across all types of application architectures. It all begins by changing your mindset toward networking and security. As the term suggests, Service-defined Firewall is a strategy for protecting each service at all possible layers or levels in the infrastructure. Using a multilayer, defense-in-depth strategy minimizes the attack vector. The first step in designing a Service-defined Firewall architecture is to create a conceptual design of the infrastructure. A conceptual design remains generic in nature and does not call out specific products for solutions. The conceptual design is a high-level look at the overall environment and where security enforcement is available across the infrastructure.

A conceptual design looks at the environment from the perspective of high-level objects and components including

- Environment – Identifying supported infrastructure (private cloud, public cloud, and so on)
- Workload – Identifying workload types (mainframe, monolithic, multi-tier, virtual machines, bare metal, containers, and so on)
- Policy management and orchestration
- Security policy enforcement – A layer for workload security enforcement

Figure 8 is a conceptual design diagram for an enterprise with two private cloud data center environments and two public cloud instances.

"The logical design for a Service-defined Firewall solution includes design decisions specific to the security policy management and orchestration and the security policy enforcement."

Enterprise Hybrid Cloud Service-defined Firewall Conceptual Design

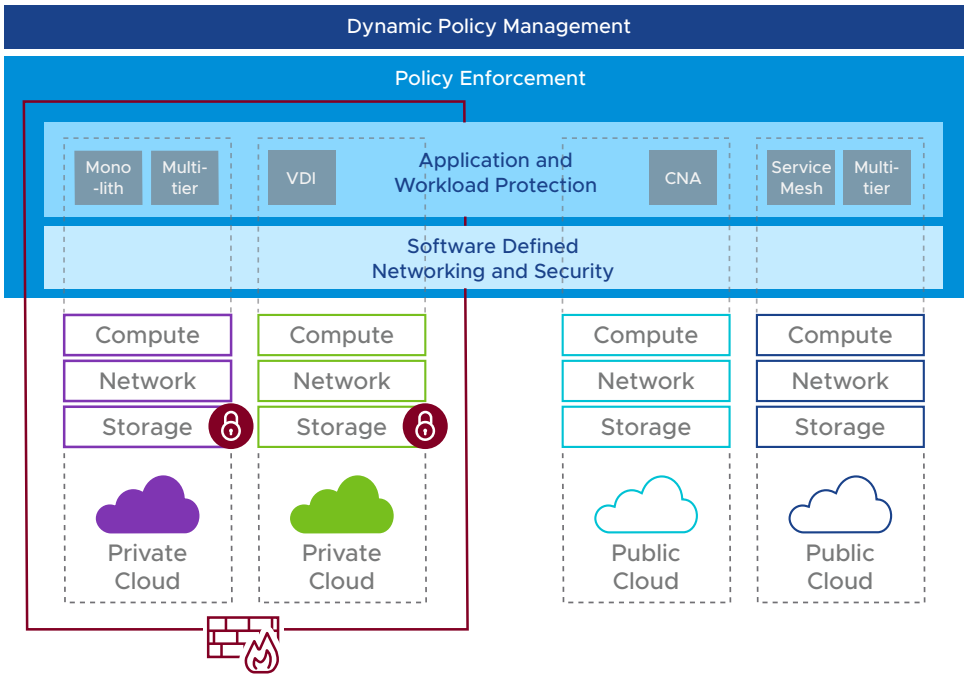


**Figure 8.** A conceptual design for a Service-defined Firewall solution architecture.

The next level of a solution architecture, the logical design, adds more detail to the conceptual design. The logical design brings to light infrastructure-specific architectural components, design decisions, and vendor products for providing solutions. The logical design for a VMware Service-Defined Firewall solution includes design decisions specific to the security policy management and orchestration and the security policy enforcement. The following table shows the design decisions specific for a VMware Service-defined Firewall architecture.

| Component | Design Decision |
|---|---|
| Security Policy Management and Orchestration | • Centralized policy manager with common language; VMware NSX Manager |
| Security Policy Enforcement | • Consistent enforcement attached to the workload independent of location; VMware NSX |

Figure 9 shows a logical design diagram for security with VMware NSX.

"A Service-defined Firewall architecture with VMware NSX includes a centralized policy manager that can be accessed through a simplified user interface, advanced user interface or a declarative API."

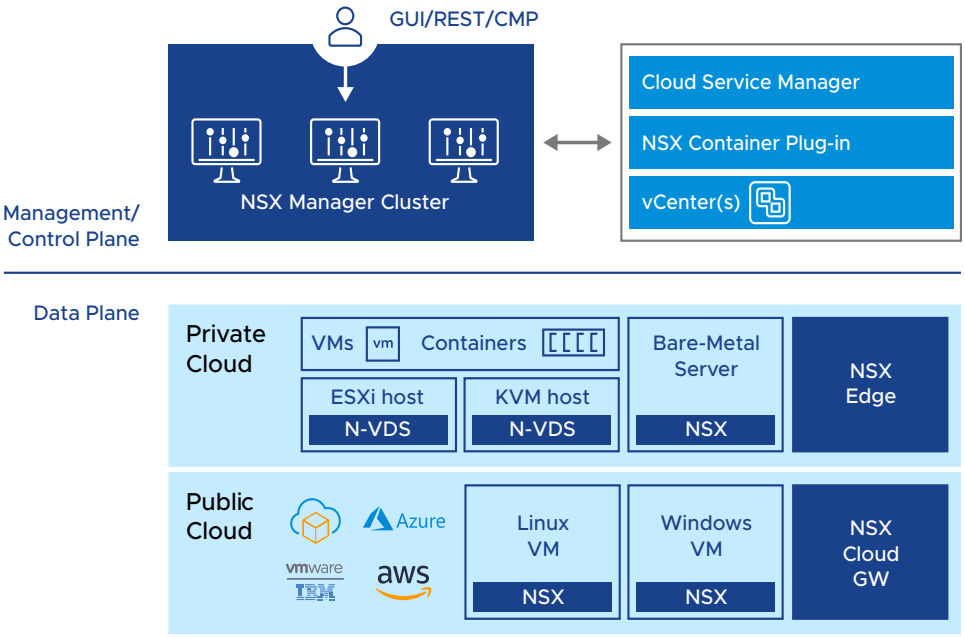


**Figure 9.** The logical design diagram adds details about the solution architecture and introduces vendor products.

## Security Policy Management and Orchestration

Centralized policy management is key to helping with manageability. A security solution that requires decentralized control can often be enough of a deterrent that alternative, less secure designs are implemented. A Service-defined Firewall architecture with VMware NSX includes a centralized policy manager that can be accessed through a simplified user interface, an advanced user interface, or a declarative API. As shown in Figure 10, the VMware NSX policy manager extends across VMware and non-VMware environments.

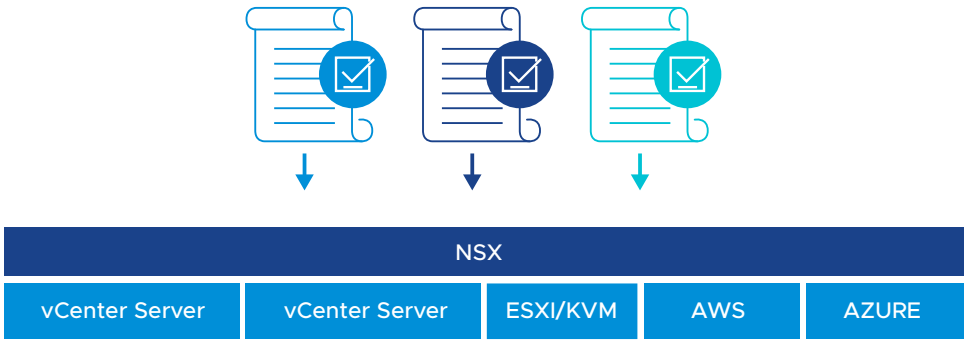


**Figure 10.** VMware NSX policy manager allows for defining a policy once while enforcement of that policy can happen in VMware and non-VMware environments.

The VMware NSX simplified user interface uses the following constructs for defining a security policy

- Domain – A logical construct representing a security zone and all rules and groups. The default domain represents the entire NSX environment.
- Security Policy – A structure to encompass various security elements, including firewall rules and service configurations. Security Policies include Distributed Firewall Policies, Gateway Policies, Network Introspection Policies, and Endpoint Policies.

"The basis of a logical grouping can be common characteristics such as machine name, operating system, network name, IP address set, MAC set, and more. Being able to create custom groupings with tags creates a more flexible and dynamic environment for security policy application."

- Rule – Set of parameters that flows are evaluated against and that define which action should be taken upon match. Rules include parameters such as Source/Destination, Service, Context Profile, Logging, and Tag.
- Group – Grouping construct to group the different objects statically and dynamically. Used in Source/Destination of Rules. Group inclusion includes virtual machines, logical ports, IP/MAC sets, AD User Groups, and other nested groups. Dynamic inclusion for VMs can be based on tag, virtual machine name, operating system type, or computer name.
- Service – Defines a combination of port and protocol. Used to classify traffic based on port/protocol. Predefined services or user-created services can be used in Rules.
- Context Profile – Defines one context-aware attribute, including APP-ID and/or Domain name, as well as subattributes such as application version or cipher set. Rules can optionally include a context profile to enable a Layer 7 firewall.

In addition to simplifying policy creation and management, a good security design will minimize the work required to apply security policy to objects. The best way to do this is through logical groupings of objects that share characteristics. The basis of a logical grouping can be common characteristics such as machine name, operating system, network name, IP address set, MAC set, and more. Being able to create custom groupings with tags creates a more flexible and dynamic environment for security policy application. Figure 11 shows a logical design of security groups and policies for a multi-tier application architecture. In the example, all web servers are grouped using a wildcard definition in which the virtual machine name contains the text "web." The application servers and database, however, do not have a consistent naming policy, so instead of using the virtual machine name, a custom tag is applied to each system.

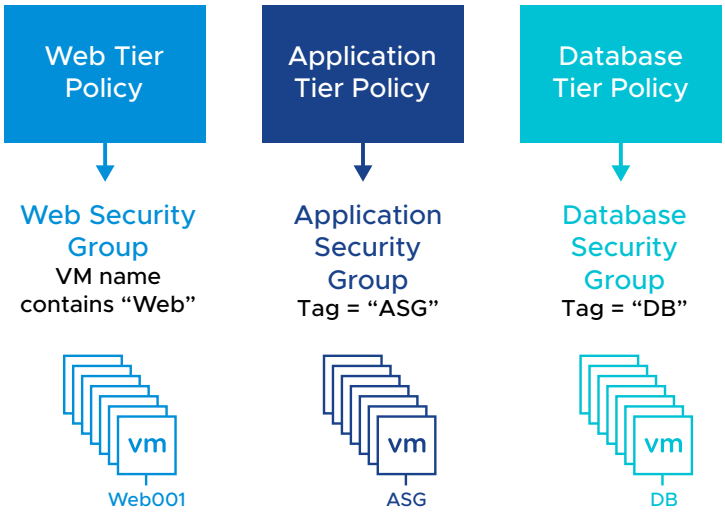


**Figure 11.** Security policies attached to security groups. Security groups are logical groupings of systems that have one or more shared characteristics.

The formation of security groups on the basis of shared function is a powerful way to design security. In some cases, the shared trait among systems in a group might be their function or security level. For companies with sensitive data or regulatory compliance standards, such as HIPPA or PCI in the health and retail industries, systems can be grouped together for the purpose of securing communication between all the systems where sensitive data is transmitted or stored. In this case all of the systems can be tagged as "Sensitive" and then grouped based on that tag.

"Common policy refers to the global nature with which a set of policy characteristics are used throughout an enterprise."

"Consistent policy refers to the frequency with which a single policy is applied across an entire environment to apply a single set of policy characteristics."

## Common vs. Consistent Policy

Policy must be common and consistent.

The terms common policy and consistent policy are often used but rarely defined. Their distinction, however, is important to understanding how security can be made more manageable using new approaches with software-defined infrastructure.

Common policy refers to the global nature with which a set of policy characteristics are used throughout an enterprise.

For example, suppose that the security team has a documented corporate policy that all web servers should allow communication only through port 443. This is a single documented policy. Without the right infrastructure design, the application and enforcement of this documented policy could require multiple instances of the same policy. If an enterprise has two data centers that include a mix of virtual and physical machines along with two public cloud instances, as shown in Figure 12, there would need to be five policies created and applied to meet the documented company policy.

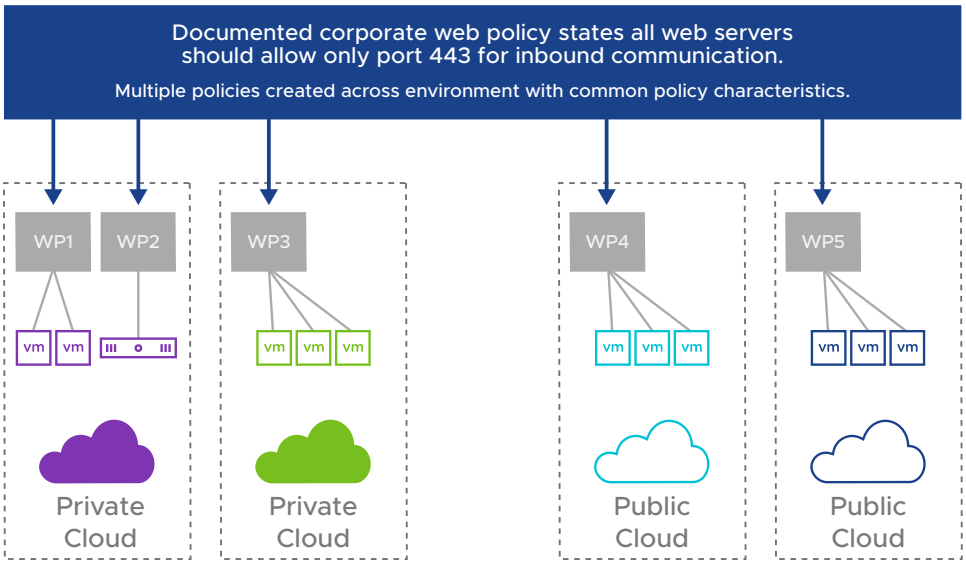


**Figure 12.** A common policy refers to characteristics that are common among multiple instances of a policy.

In this case, if a change is made to the documented company policy, five subsequent changes will be required across all environments. Similarly, if the security team comes up with a new policy, five new policies will need to be created. Using this approach with hundreds of applications, each with multiple tiers, renders policy management almost impossible. Not to mention the opportunity for human error of missing a policy for update or not creating a new policy in one of the environments. This type of enterprise security design lacks consistency.

Enter consistent policy. Consistent policy refers to the frequency with which a single policy is applied across an entire environment in order to apply a single set of policy characteristics. Building off the previous example, the security team has a documented policy for web servers. With a consistent policy strategy, rather than building five policies to fit the environment, a single policy is created that spans the entire enterprise architecture. This approach, shown in Figure 13, has tremendous advantages. Managing policies is much easier, because there are exponentially fewer policies to update and create as application requirements change over time. In addition, security teams can be confident that the policy definition does not vary from data center to data center, location to location, or web VM to web VM whether physical or virtual.

"Consistent policy is better with consistent ownership of the enforcement. A policy that is translated to something outside of the policies enforcement realm introduces opportunities for security posture to be compromised."

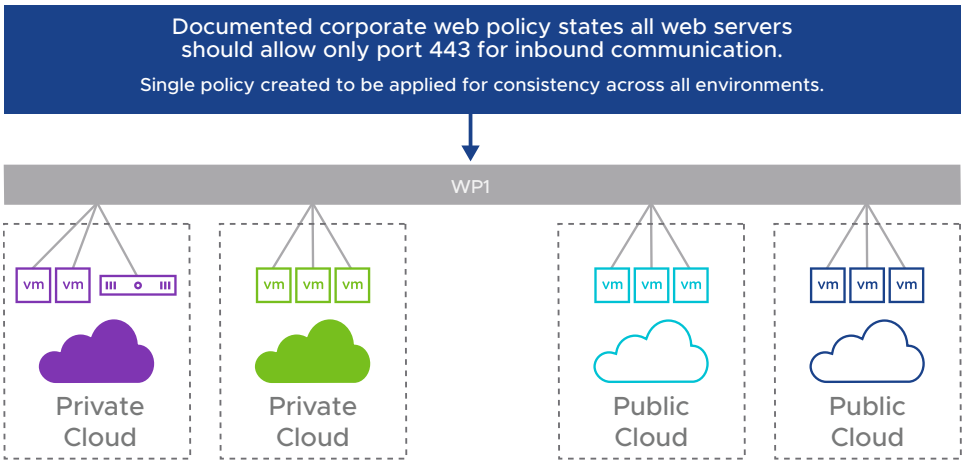


**Figure 13.** A consistent policy applies a single policy of security characteristics across an entire enterprise architecture.

Consistent policy is better with consistent ownership of the enforcement. A policy that is translated to something outside of the policies enforcement realm introduces opportunities for the security posture to be compromised. Only under unique circumstances should it be acceptable to accept disparate control and enforcement. While enforcement methods are sure to vary between virtual machines, physical machines, containers, and so on, it is advantageous to security engineers to minimize the number of tools, technologies, and enforcement required to be compliant with documented company policy. Figure 14 compares consistent policy and enforcement with consistent policy and translated enforcement.

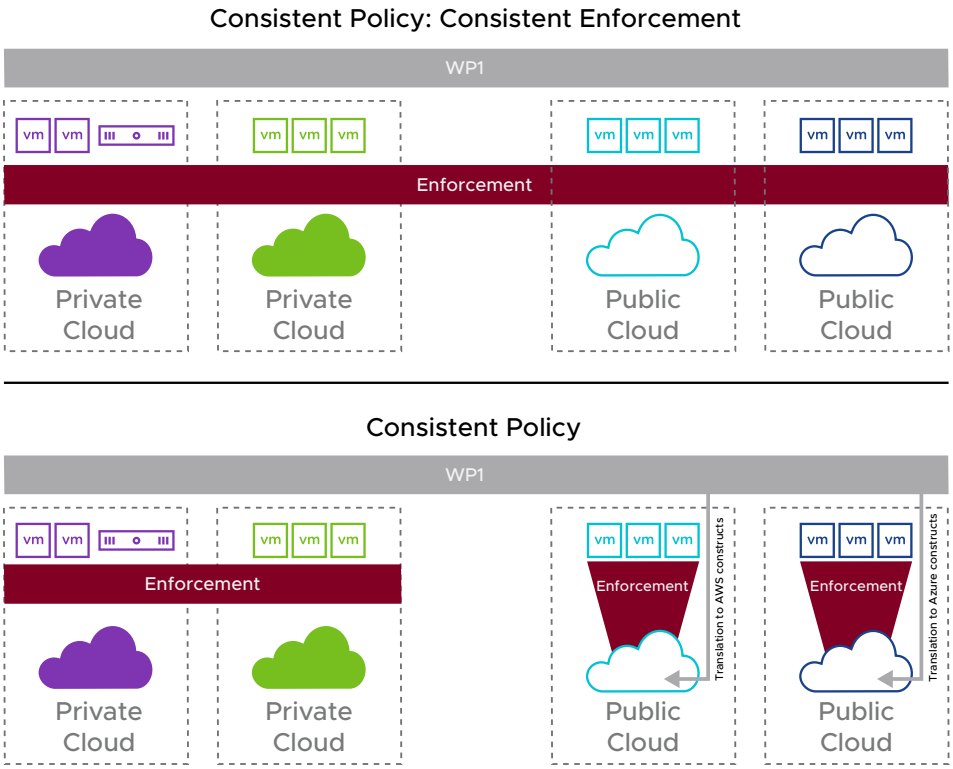


**Figure 14.** A comparison of security models. Consistent policy and consistent enforcement vs consistent policy. Eliminating translation avoids problems that arise from lack of a consistent data plane enforcement.

"A good Service-defined Firewall solution architecture design will also provide details about the security policy enforcement tier."

VMware NSX offers common policy and maintains the ownership of enforcement in the data plane. There is no translation to local elements. Figure 15 shows the VMware NSX architecture for cloud security.

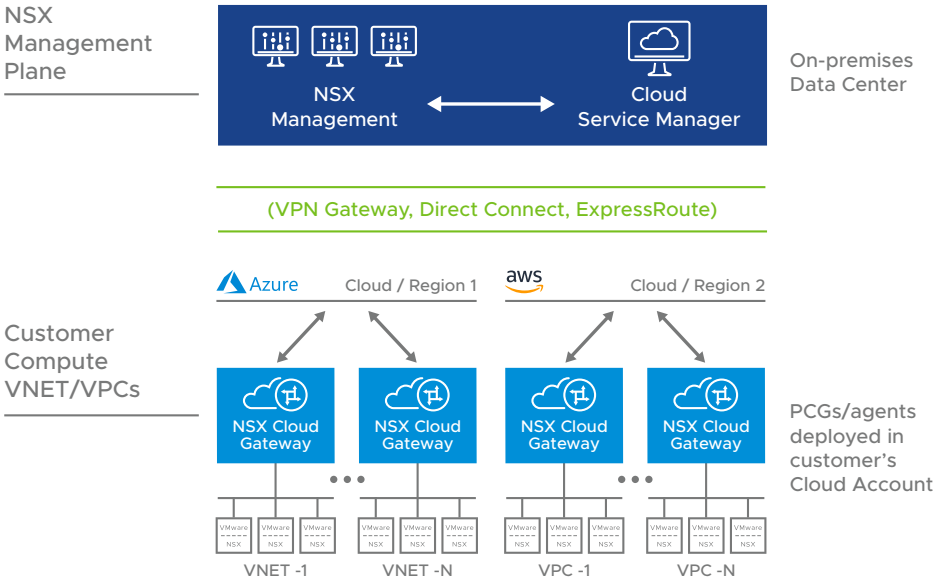


**Figure 15.** VMware NSX owns all aspects of policy creation and enforcement in for private and public cloud workloads. Translation is avoided to eliminate inconsistent enforcement methods.

## Security Policy Enforcement

A good Service-defined Firewall solution architecture design will also provide details about the security policy enforcement tier. This includes defining which elements of security are statically bound to the host upon which a workload runs and which elements are bound to the workload. Those elements bound to the workload will remain attached to the workload as it moves throughout and between data centers and clouds. Figure 16 shows a detailed look at the security stack for an application. Security constructs bound to the host are those that rely on a Trusted Platform Module (TPM) and encryption at the storage level. This isn't to suggest that these constructs are not available on destination hosts as the workload migrates across the environment, but that the TPM and encryption details will change because they are host specific.

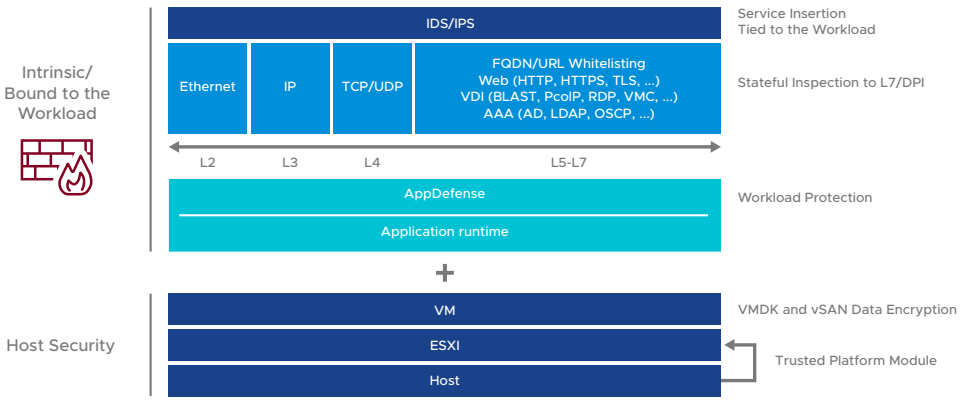


**Figure 16.** A more detailed look at the security stack shows that some elements of security are host specific, while the majority are bound to the workload. Those bound to the workload remain with the workload as it migrates through the environment.

"VMware AppDefense leverages its position in the hypervisor to learn and monitor the intended state of an application."

## Host Security

Security at the host level is easy to control in the private cloud model but becomes impossible in the public cloud. In a private cloud, architects have access to, and control over, the hardware configuration. In a public cloud, this access and control is forbidden. Therefore, application security must be designed to leverage the software components of the architecture. The security stack in Figure 13 provides a conceptual look at the architectural levels of security.

For more information on host security, see the VMware Security Hardening Guides at https://www.vmware.com/security/hardening-guides.html.

## Intended State and Behavior

The application runtime is made up of the actual application processes, executables, functions, or services that make the application run. Applying a whitelisting approach to the application runtime ensures that the application maintains an intended state. It is easier from a security and application protection standpoint to define a known good behavior than it is to try to identify and prevent all unknown bad behaviors. This level of application visibility results in more accurate security policies and faster remediation, simplifying the prevention of malicious behavior. This result is a common source of truth for IT and security teams, making it easy for them to collaborate around compliance, security incident investigation, and incident response.

VMware AppDefense™ is a new approach to application security. In contrast to traditional security solutions, which focus on chasing threats, VMware AppDefense leverages its position in the hypervisor to learn and monitor the intended state of an application. VMware AppDefense responds quickly to any deviations from the intended state. Figure 17 highlights the VMware AppDefense security strategy.

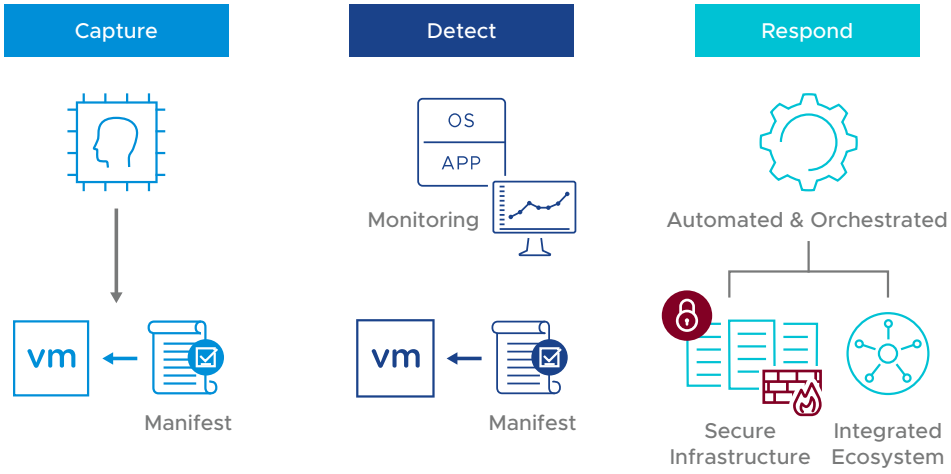


**Figure 17.** VMware AppDefense learns the intended state of an application, then monitors, detects, and automatically responds to any deviations from that state.

VMware AppDefense builds context by gathering the inventory of virtual machines and the application details from automation and provisioning tools (examples include vCenter, Puppet, and Ansible) to understand the intent of a particular machine and application. It then monitors the behavior of the virtual machine, operating system, processes, and application, and correlates this information along with the intent defined during provisioning. Using machine learning, VMware AppDefense creates a blueprint based on known good behavior—how the machine and application should be functioning and communicating.

"Hypervisor-enabled security enforcement ensures isolation from the workloads being protected. The security is managed and enforced outside of the operating system, making it difficult for attackers to circumvent or disable the policy enforcement."

Once the blueprint is established, it is stored in a secure partition of the hypervisor. VMware AppDefense monitors for any changes, detecting and preventing any deviations from the intended, established state, and ensuring the integrity of applications, infrastructure, and the operating system. When a threat is detected, it can immediately respond through a variety of capabilities natively and through VMware NSX Data Center for enforcement and containment.

### Stateful Inspection to L7/DPI

Security in Layers 2 through 7 is where we find a combination of traditional methods and new, more efficient methods. An in-kernel distributed firewall allows rules to be created based upon the traditional networking constructs of MAC address, IP address, port, and protocol. While the rule components are no different, the enforcement of them is very different. Historically a physical firewall would enforce this type of rule. However, with in-kernel firewall capabilities, the enforcement capability is brought closer to each and every workload component. Figure 18 shows VMware NSX distributed firewall architecture as an in-kernel, internal firewall.

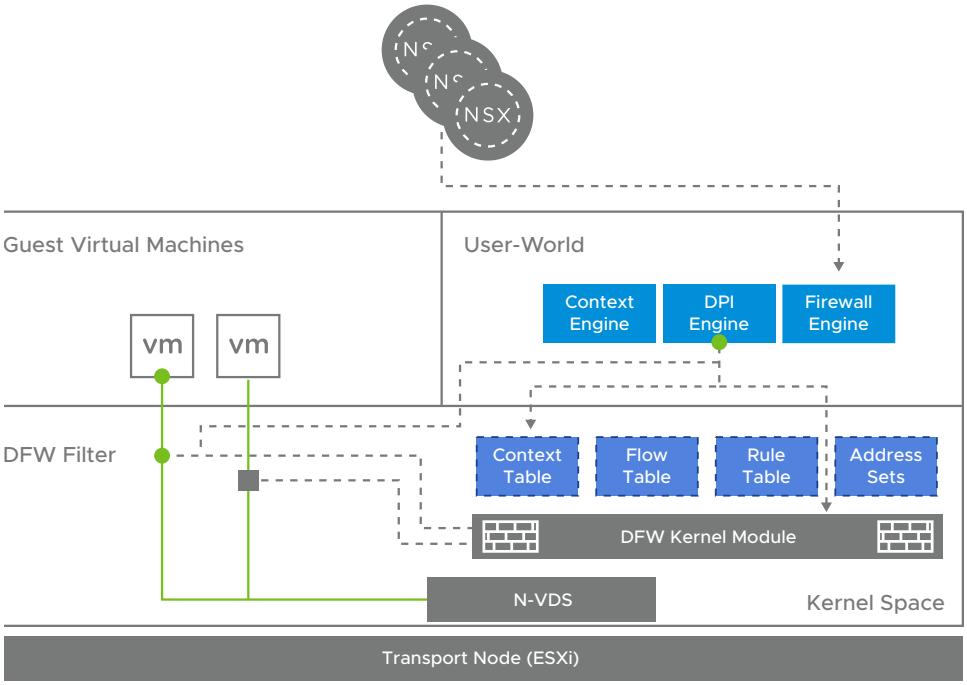


**Figure 18.** VMware NSX provides in-kernel firewall protection for virtual machines running on ESXi.

Using an in-kernel firewall as part of a defense-in-depth strategy—in particular, a zero-trust policy model—dramatically enhances the physical network efficiency. Security policy enforcement in the hypervisor prevents any illegitimate traffic from reaching the physical network, thereby allowing the physical network to process only legitimate traffic in an efficient manner. Hypervisor-enabled security enforcement ensures isolation from the workloads being protected. The security is managed and enforced outside of the operating system, making it difficult for attackers to circumvent or disable the policy enforcement. VMware NSX, with its integration into the rest of the VMware architecture, has better context than external hardware-centric security tools. This context allows rules to be created that leverage VMware vSphere® constructs such clusters, resource pools, virtual machine names, port group names, logical switch names, and custom tags. These characteristics are in addition to the aforementioned traditional data of MAC, IP, port, and protocol.

"By having a service virtual machine on each host, the deep inspection can happen prior to the egress of the host onto the physical network."

### Service Insertion Tied to the Workload

For added security to specific workloads, service insertion can be added as another element of security. Service insertion policies for intrusion detection systems (IDS) and intrusion prevention systems (IPS) are bound to the workload. These services do not happen in kernel but are maintained within the same host through a service virtual machine that resides on each host. By having a service virtual machine on each host, the deep inspection can happen prior to the egress of the host onto the physical network. While this type of distributed model does consume more compute and storage resources, it provides a much greater advantage to the efficiency of the security architecture. Figure 19 shows the architecture for service insertion.

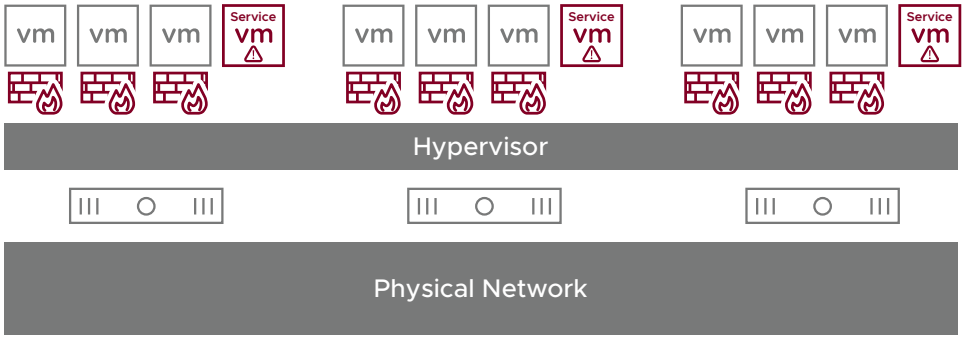


**Figure 19.** Service insertion is best when using a distributed model in which a service virtual machine resides on each hypervisor. This allows the deep security inspection to happen prior to egress of the physical host onto the physical network.

## Call to Action

The most immediate call to action is to change the way you think about enterprise security. Your mindset about protecting the infrastructure will determine your ability to solve new and future business problems. Looking at security through the lens of the traditional, hardware-centric model will prevent you from seeing the new and exciting security developments that are breaking that mold.

VMware NSX represents a best-of-breed solution that helps create secure, manageable, and efficient infrastructure through a common control plane and data plane enforcement approach. Figure 24 shows the VMware NSX architecture protecting the enterprise multi-cloud infrastructure.

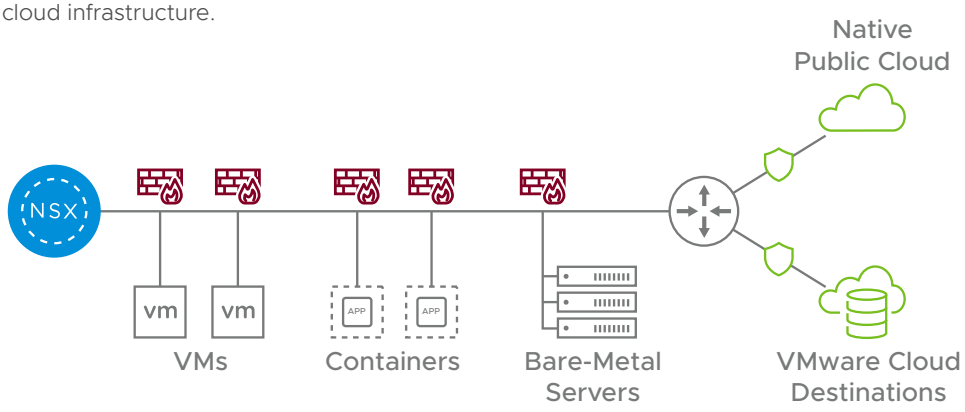


**Figure 24.** VMware NSX offers a centralized control plane and consistent data plane across private and public clouds.

The configuration of the security groups and policies through VMware NSX Manager™ can be performed via graphical interface or automated through REST API or a cloud

"New platforms of security such as VMware NSX make it possible to define security and enforce security as needed for each application, workload, and service that supports the enterprise."

management platform. The centralized management and control plane offers consistent policy throughout the private and public cloud. The consistent presence of VMware NSX in the data plane eliminates issues with translation to public clouds, thereby maintaining a single software-driven security platform for management, enforcement, monitoring, and operations.

Designing a secure defense-in-depth strategy for an enterprise architecture on-premise or across clouds will never be easier than not prioritizing security. However, the days of security being an afterthought are long gone. Security must be at the forefront of any enterprise design. The challenge for architects is to identify intrinsic, manageable, and ubiquitous solutions that provide deep, broad, and specific security without being overly complex. The security posture of workloads must be minimized and never jeopardized by the demands of workload scale or portability. New platforms of security such as VMware NSX make it possible to define security and enforce security as needed for each application, workload, and service that supports the enterprise. A Service-defined Firewall is critical to the secure enterprise, because it is the adaptation of security design that increases the chances of survival.

**VMware, Inc.** 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com  Item No: SDF_Solution_ArchitectureJR7_062619 2/19