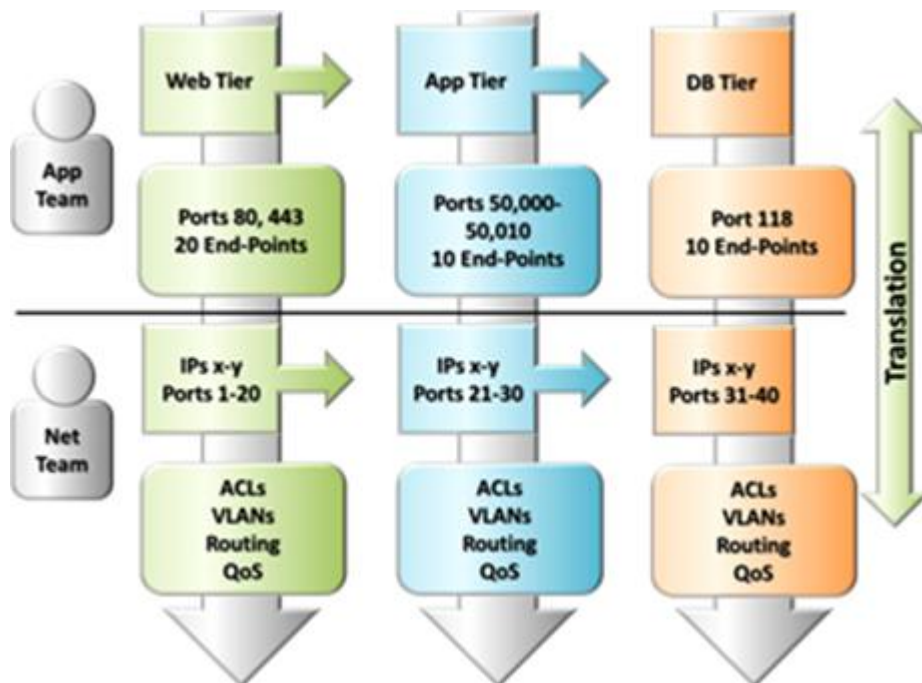


FOCUS ON THE BALL: THE APPLICATION

With the industry talking about Software Defined Networking (SDN) at full hype levels, there is one thing missing from many discussions: the application. SDN promises to reign in the complexity of network infrastructure and provide better tools for deploying services at scale. What often seems to be forgotten are the applications, which are the reason those networks exist. While application focus in itself is not a new concept it seems lost in the noise around SDN as a whole, with a few exceptions such as Plexxi being which focuses on Application Affinity.

Current SDN approaches provide tools to solve issues in one portion or the other of network infrastructure. Flow control mechanisms look to centralize the distribution and configuration of routing and forwarding. Overlays look to build virtual networks on existing IP infrastructure. Virtualized L4-7 services provide solutions to configure, stitch-in and control network services more closely to virtual machines themselves. None of these current approaches looks to tackle the whole picture from an application centric point of view. These solutions also take a myopic view that the VM is the network, this is far from the case. The closest models fall into dev-ops categories or orchestration but these require a deep understanding of the details and intricacies of the network.

In traditional networking environments there is a disconnect in communication between application and network teams. The languages and concepts are disparate enough that they don't translate, there is no logical continuation from application developer or owner to network designer. Application teams speak in OS instances, application tiers and components, tooling, language, end-user demands, etc. while network teams speak in switch-ports, VLANs, QoS, IP addressing and Access Control Lists (ACLs). The lack of common understanding and vocabulary causes architectures and implementations to suffer. The graphic below illustrates this relationship:



Building the flexible, scalable, manageable and programmable networks of the future requires a change in focus. The application needs to take center stage; it's the apps that solve business problems. From this focus, logical and physical

topology become secondary and are only designed once application requirements have been mapped out. Application centric policies must be designed first. Policies such as: security, load-balancing, QoS can all be designed based on application requirements, rather than network restrictions. Application developers define these requirements without the need to speak a network language.

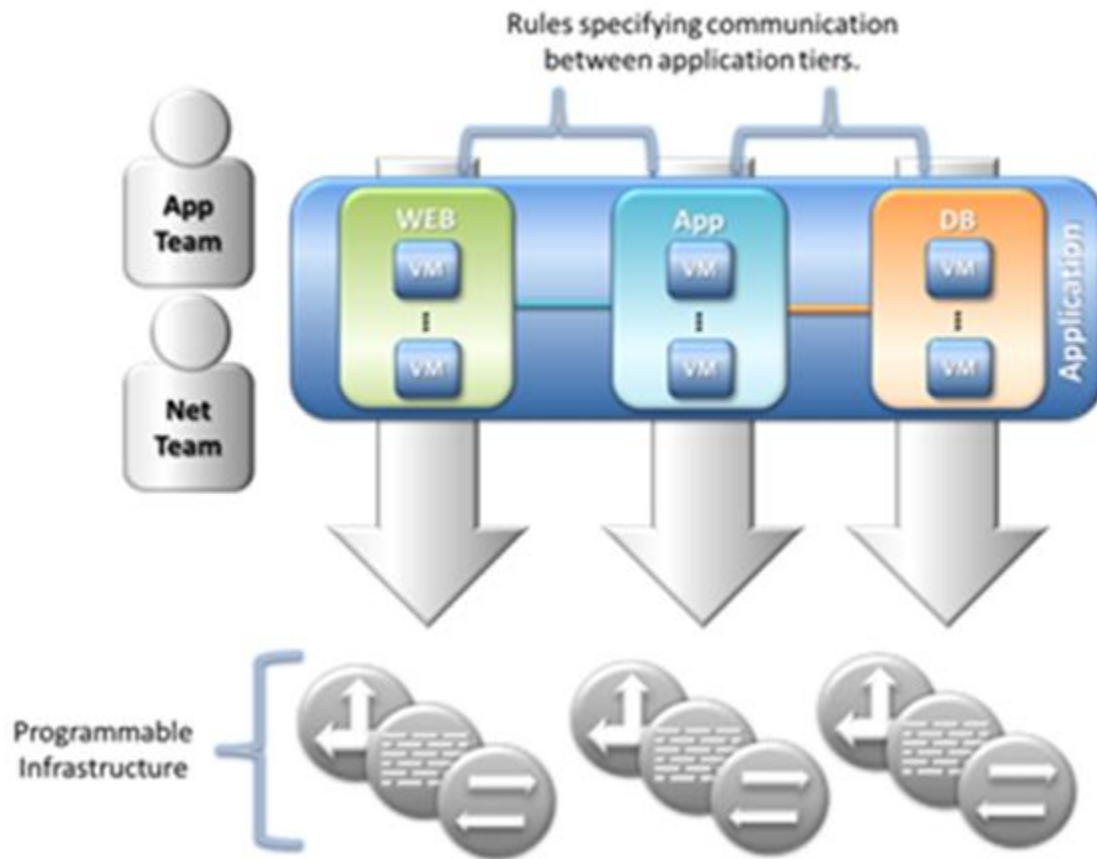
Traditional networks begin with a physical topology that is layered with L2 and L3 logical topologies and assumed application mobility and service domains such as a services tier in the aggregation level. Once these topologies are architected and implemented applications are built and deployed on them. This method limits the capabilities available to the application and the services deployed on them.

Application security is an excellent example of a system that suffers from traditional architectures. Network security constructs are implemented in the form of ACLs on switches, routers and firewalls. These entries suffer from two major drawbacks: complexity of design/implementation and scale of the TCAM that stores the entries. This means that application policies must be communicated effectively to network engineers who must translate those requirements into implementable ACLs across multiple devices in the network. This is then defined manually device-by-device. This is a system ripe for PEBKAC errors (Problem Exists Between Keyboard and Chair.)

The complexity and room for error in this system increases exponentially as networks scale, applications move and new services are needed. Additionally this leads to bad practice based on design limitations. Far too often outdated policy entries are left in place due to the complexity and risk of removing entries. This leads to residual entries in place consuming space long after an application is gone. Just as often policies are written more loosely than would be optimal in order to reduce required entries, and optimize space, through wild card summarization. To break this cycle networking systems need to take an application centric approach which models actual application requirements onto the network in a top down fashion. Systems need to take into account the structure of the application, its components, and how those components interact then provide tools for designing logical policy maps of these relationships. From there these policy maps can be programmatically applied to the networking infrastructure.

An application is not a single software instance running on a server. Applications are made up of the end-points required in a given tier, the tiers required for the service delivered and the policies that define how those tiers communicate, and their unique requirements. The application as a whole must be taken into account in order to provide robust, scalable service delivery.

The illustration below shows this relationship in contrast to the diagram above:



In this model network and application teams develop the systems of policies that define application behavior and push them to the network. Taking the application as a whole into focus instead of the myopic view of VMs, switch ports or IP addresses allows cohesive deployment and manageability at scale. The application is the purpose of having a network; therefore the application should define the network.

This definition of the network by the application should be done in a language that the developers understand, and the network can interpret and implement.

For example an app owner labels application traffic as ‘video’ and the network implements policies for bandwidth, QoS, etc. that video requires. These policies are predefined by the network engineers.

An application is more than an IP address and a set of rules; it is an ecosystem of interconnected devices and the policies that define their relationship. Traditional networking techniques anchor application deployment by defining applications in networking terms. In order to accelerate the application deployment (and re-deployment throughout its lifecycle) networks need to provide an application centric view and deployment model.

Source: <http://www.definethecloud.net/focus-on-the-ball-the-application/>