Controlling the Cloud: Requirements for Cloud Computing

F5 Networks' perspective on cloud computing: definition, architecture, and development.

White Paper
by Lori MacVittie

Introduction

You can hardly open any technology or business journal, website, or newspaper today without hearing some commentary on cloud computing-what it is and how it will change IT and business. Cloud computing's impact will continue to be felt for many years regardless of how it all comes together in the end. At the same time no single definition of cloud computing exists or is being talked about, planned, and even implemented in today's enterprise networks. Defining cloud computing and notions of "the cloud" are extremely ambiguous and difficult to nail down. Few vendors are willing to step beyond the marketing hype and "cloud washing" to present a perspective of what true cloud computing represents, what currently exists, what is missing, and the characteristics required for enterprise adoption of this dynamic and powerful change in computing ideology.

Cloud computing represents not a revolution but an evolution of existing enterprise computing architectures, dating back to the first instance of networked computing. The difference is that today there are vast advances in virtualization in nearly every aspect of the data center. There has also been an emergence of a dynamic understanding and need to control what, how, and when the cloud provides services to the consumers of those services. This new dynamic paradigm must be able to intercept application and data traffic, interpret the current context, and instruct the cloud infrastructure on how to most efficiently deliver the request.

The cloud is a holistic ecosystem of components, not a point product or single vendor solution, and has basic, specific requirements to meet the needs of enterprise organizations. These requirements include scalability, adaptability, extensibility, and manageability. In addition, the cloud must exhibit additional capabilities that address the best-in-class requirements of the enterprise-such as providing for security, real-time availability, and performance.

The question that remains, however, is what does this new dynamic computing architecture look like and what is required-above and beyond the standard tools we have today-to qualify as a "cloud"?

The Big Picture

An important strategic consideration is the integration of all the pieces of the infrastructure to create the cloud. This includes everything from the bare metal to the users to all of the elements in between. In addition, there are different ways to view the interaction of various operations within the architecture, depending on your role.
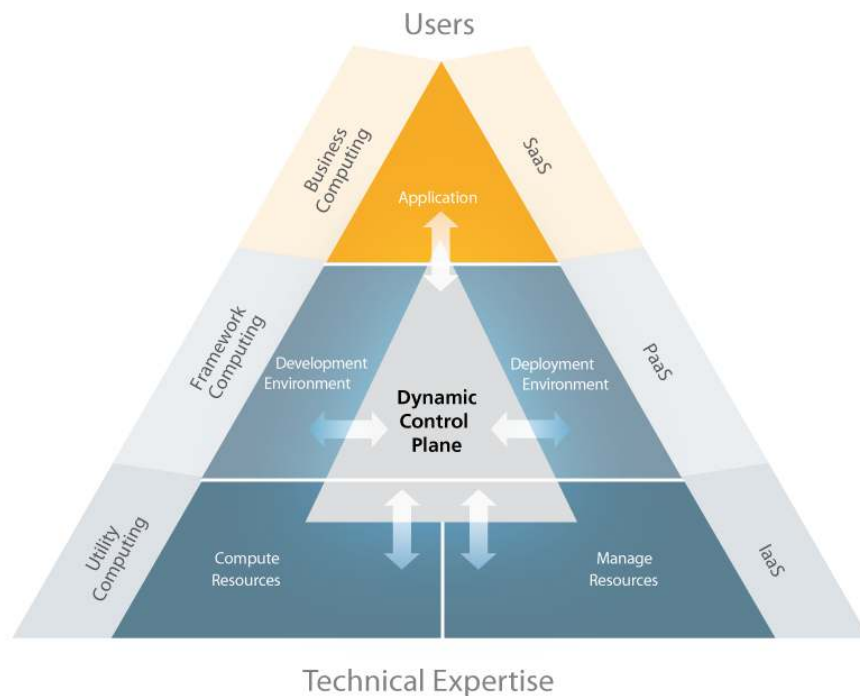
Figure 1. Cloud architecture

The cloud computing architecture is built upon several functional component blocks (for example, compute resources or deployment environments), which are organized into specific layers of a pyramid. The width of these layers represents the depth of technical expertise required to build and/or deploy that layer. The pyramid layers are roughly synonymous with the notions of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). At the apex of the pyramid are users accessing the applications; in the center is a dynamic control plane that traverses all others and provides real-time connectivity, information coordination, and flow control between the layers.

In order to maximize the value of cloud architecture, each component must exist in some state or another. For example, dynamic control plane elements are a requirement at every layer of the cloud architecture in order for cloud environments to be operationally efficient and on demand. This requires a level of automation and orchestration that can only be achieved by integrating components across the architecture. Without this capability, organizations cannot realize the benefits associated with the model. If any of the core components is not implemented, such collaboration will fail and true cloud architecture cannot be achieved.

Designing the Cloud

The components required to build the cloud are similar to the components required to build a traditional architecture; the difference is how they integrate, communicate, and act.

Certainly, a fairly strong argument can be made that web-hosting services from a decade ago represent the first implementations of software as a service (SaaS). Some will even argue that it represented the first platform as a service (PaaS) solution as well-providing an HTML platform to build custom applications. Many of the same people will suggest that the rack-n-power providers of the dot-com boom provided infrastructure as a service (IaaS) solutions.

Cloud computing has evolved from a single server being provisioned for a single customer, to a hosting provider and then to a business continuity and disaster recovery provider. Along the way, technology overcame physical limitations with devices like load balancers, WAN optimization, compression, caching, and content delivery networks (CDNs). We learned to integrate devices: built APIs, consolidated racks of servers in racks of blades and learned to provide automated provisioning. Cloud architecture is simply the logical conclusion of this decade's long evolution. We now have unprecedented levels of virtualization of hardware, software, network, and storage and are on the verge of putting it all together.

So, what is that final threshold, what is the difference between a cloud and falling short of the cloud?

Dynamic Control Plane

Traditional traffic and computing systems often break processing into two discrete components: the data plane and the control plane. The data plane is concerned with the basic process of getting data-be it input from a system or, in our case, requests from users-and returning data (output, files, or responses). The data plane is the basic connectivity that handles traffic flow to and from destinations. The control plane, on the other hand, is more concerned with managing that data in response to context and policy; it changes the "how" of the data plane.

The core idea of cloud architecture is to connect users-who might be mobile and moving between LANs, WLANs, and Internet connections-and services to the applications they consume, which can also move between cloud centers based on different needs of the business. As hardware resources and servers are spun-up or decommissioned, as applications are moved from development to production, or as entire applications are moved from the internal data center to a cloud provider, the cloud architecture requires a dynamic control plane that monitors the data and ensures that it is constantly connected in the best possible manner. The dynamic control pane must be able to Intercept traffic as it traverses the cloud, Interpret the data, and Instruct the cloud architecture on how to efficiently connect the user to the appropriate application instance.

Intercept

The dynamic control plane, must be in a position to have visibility to all traffic between the user and the application and across the entire cloud platform. Without the ability to intercept traffic and data requests, the dynamic control plane cannot appropriately do any of the other things it needs to do. Not only must it be able to see the actual flow, it must also be able to intercept the metadata or context of the traffic. The dynamic control plane must have the visibility into the data plane and all components that operate within the data plane.

Interpret

Having all the information about data and application flow is not enough. The dynamic control plane must have the ability to understand the elements of context in relation to the individual request, business policy, and other application and cloud traffic. The dynamic control plane must constantly evaluate the context and policy to make intelligent decisions at any given moment.

Instruct

Once the dynamic control plane has all the available information and analyzed the context, it must instruct the architecture on how best to connect the two endpoints. The dynamic control plane must also communicate with the infrastructure-the data plane-to change the current delivery model to meet the needs identified. This might require sending requests to the a new instance of the application or to a new data center, changing compression and encryption settings, or even instructing other components in the architecture to create or destroy resources necessary to delivering that application or data. It might also be necessary for the dynamic control plane to simply deny access based on the policies and context at any given moment.

These three things necessitate the integration that we alluded to before and underscore the necessity for the inclusion of the dynamic control plane interface at each level and within each component of the cloud architecture. Whether it is a native, purpose-built integration or simply an open standard that can be used by the consumer, **the dynamic control plane must be integrated in order to fully intercept, interpret, and instruct or it is not a cloud.** The greater the capability to provide these services, the greater the ability for the dynamic control plane to operate intelligently and entirely on its own without manual, human intervention.

Cloud Architecture

Utility Computing

Built from core components that include compute resources and management resources, the base layer of the cloud architecture requires the most technical competence to build and deploy. This is the very foundation upon which a cloud is built and, as suggested, is made up of the components most often supplied by vendors who provide IaaS solutions to their customers.

IaaS

Infrastructure as a Service (IaaS) is a cloud computing model based on the premise that the entire infrastructure is deployed in an on-demand model. This almost always takes the form of a virtualized infrastructure and infrastructure services that enables the customer to deploy virtual machines as components that are managed through a console. The physical resources-servers, storage, and network-are maintained by the cloud provider while the infrastructure deployed on top of those components is managed by the user. Note that the user of IaaS is almost always a team comprised of IT experts in the required infrastructure components.

IaaS leverages the dynamic control plane to enable on-demand scalability through the rapid and automatic provisioning of compute resources. In the case of a virtualized architecture-the most common form of IaaS architecture-this involves the automatic deployment and launch of new instances of a virtual machine. The amount of instances launched will match the amount needed to meet capacity, with the expectation that these instances will be decommissioned as demand decreases.

In this layer of the architecture, each component is responsible for providing actionable data to the other components and performing specific tasks to successfully execute an auto-provisioning or decommissioning scenario.

IaaS is often considered utility computing because it treats compute resources much like utilities (such as electricity) are treated. When the demand for capacity increases, more computing resources are provided by the provider. As demand for capacity decreases, the amount of computing resources available decreases appropriately. This enables the "on-demand" as well as the "pay-per-use" properties of cloud architecture.
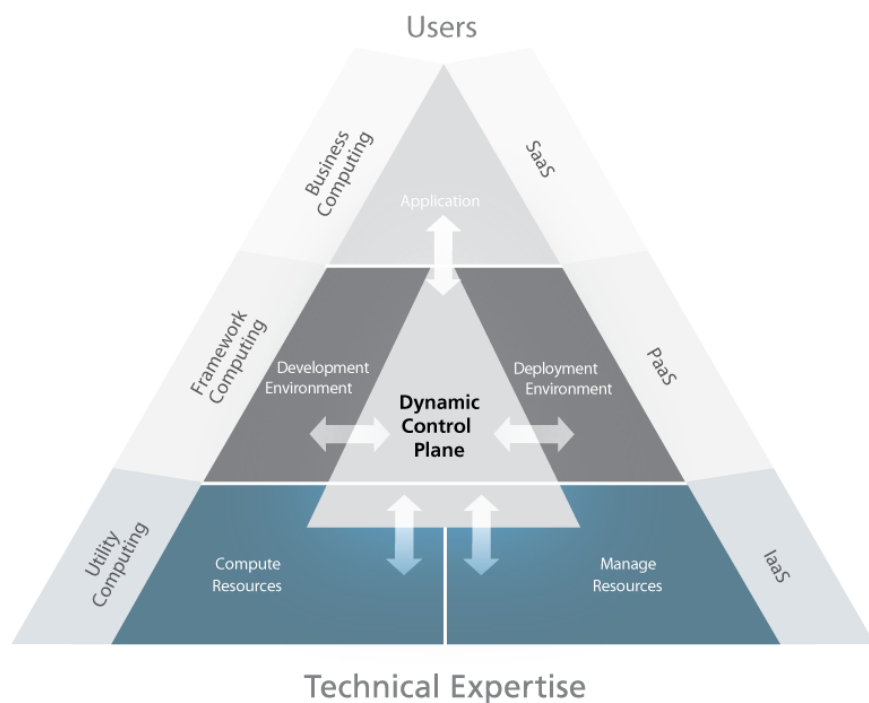

Figure 2. IaaS and cloud architecture

Compute Resources

Compute resources are one of the most basic components of the cloud-bare-metal resources such as CPU, memory, and disk-that ultimately power applications built within the cloud. This might be a hosting service provider with hundreds or thousands of installed server systems waiting to be used by subscribers or it could be a single blade-chassis with extremely dense resources designed for virtual segmentation.

This layer typically conjures the image of a traditional server. Of course, today's systems are much more complicated and versatile. They can have massive numbers of processing cores and memory that can be carved into virtual systems; auto-provisioning network interface cards that can dynamically be configured from 10MB to multi-gigabit; and both direct attached and network-attached storage systems to support the needs of the application software that will eventually reside on top of it.

Manage Resources

Manage resources are the components required to turn bare metal into usable server platforms with the appropriate CPU, memory, and disk resources necessary to support the applications that will be built upon them. Manage resources are also responsible for continuing to monitor the resource needs and ensuring that the application receives all the compute resources it needs-and moving the application or finding additional resources.

This component is most often synonymous with virtual machine management or software provisioning systems which can take the bare metal and apply operating systems, patches, and application logic and apply higher-level network connectivity (IP addressing and more).

It is necessary to share information between the layer of compute resources and manage resources so as not to waste compute resources that could be better used by another application.

Framework Computing

Many applications are built on software platforms that run on top of infrastructure services. These platforms can be environments such as Oracle or ASP.NET and provide a convenient way for businesses to build custom applications without worrying about the details that lay beneath the platforms. While many platforms are based on standards-for example Java EE-others are proprietary in nature, including Google AppEngine and architecture frameworks developed and deployed by enterprise architects.

PaaS

Platform as a Service (PaaS) is a cloud computing model in which a specific development and deployment platform-for example, Java EE, IBM WebSphere, Oracle, Google Apps, .NET, BizTalk-is the basis for deployment. These clouds are proprietary in the sense that only applications developed for the specific platform can be deployed in the cloud.

PaaS is a kind of framework computing in that the platform provided is the core framework in which applications are specifically developed. These applications will not run on any other platform, and often include platform-specific extensions or services, such as Amazon's SimpleDB, that cannot be ported to other environments. The concept of framework computing comes from architectures in which a layer of capabilities and services are provided that abstract (and insulate) developers from the underlying details. This approach leads to more rapid development and deployment of applications.
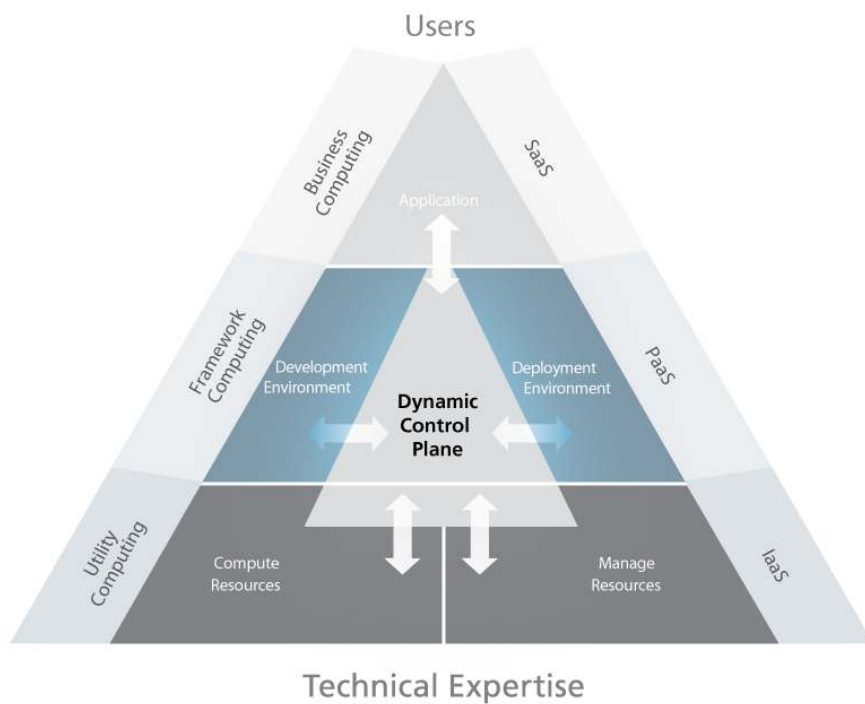

Figure 3. PaaS and cloud architecture

Development Environment

All platforms require a development environment in which the applications are designed, built, tested, and validated-outside of the production environment. These development environments can be traditional integrated development environments (IDE) that are configured to deploy to resources within a PaaS environment or they can be integrated directly as part of the PaaS offering. Microsoft Visual Studio-derived environments are as capable of connecting to internal as external instances of Microsoft-specific platform, thus enabling "offline" development of applications to be deployed in a PaaS environment. Increasingly, less standard and more proprietary offerings-those that are wholly dependent on resources that exist only in the PaaS environment such as Salesforce.com's Force. com-provide a PaaS-hosted development environment through which developers can build, test, and deploy their solutions.

Deployment Environment

A second component required is to deploy the application into production once it is ready to be consumed by the end users. This ability is essentially the run-time environment in which the applications are deployed. The difference between the PaaS run-time environment and that of hosted or even traditional enterprise-deployed platforms is the expectation of on-demand scalability associated with PaaS that does not exist in other incarnations. This on-demand scalability can result from deploying the environment on a generic IaaS, or from specifically building out and connecting the required development, deployment, and dynamic control plane. The latter results in the creation of a platform-specific IaaS, with the required components arranged specifically to provision and decommission resources based on the unique needs of the deployment environment.

Business Computing

At the top of the pyramid is general business computing. This is where many organizations-especially business organizations-find themselves; with the ability to identify a business need, but without the ability to build an application or the infrastructure upon which it runs. Instead of relying on an internal IT organization to build and/or deploy infrastructure and platforms, business stakeholders simply select an application and run it. Most organizations choose this option because the capital, operating expenses and hours required to implement standardized applications are not financially feasible, not an efficient use of IT resources, or simply beyond the capabilities of the organization.

SaaS

Software as a Service (SaaS) is a cloud computing model in which pre-built applications (such as CRM, SFA, word processing, spreadsheets, and HRM) are offered to customers via a web browser or other local interface such as a mobile device application. These applications are generally customizable, though the customer need not be concerned with the underlying infrastructure or the development platform or the actual implementation.

Despite the appearance that applications deployed in a SaaS cloud architecture are merely hosted applications-for example, the ASP model of the "dot com/dot bomb" era-these business computing-focused architectures are often built on a PaaS deployed upon an IaaS. While the platform upon which the application is deployed might be hidden from the consumer, it is the source of the multi-tenant properties of SaaS and the customization capabilities offered to its consumers. The underlying platform for Salesforce.com, for example, was exposed in recent years as a separate PaaS offering called Force.com, providing customers with additional options for building custom solutions.
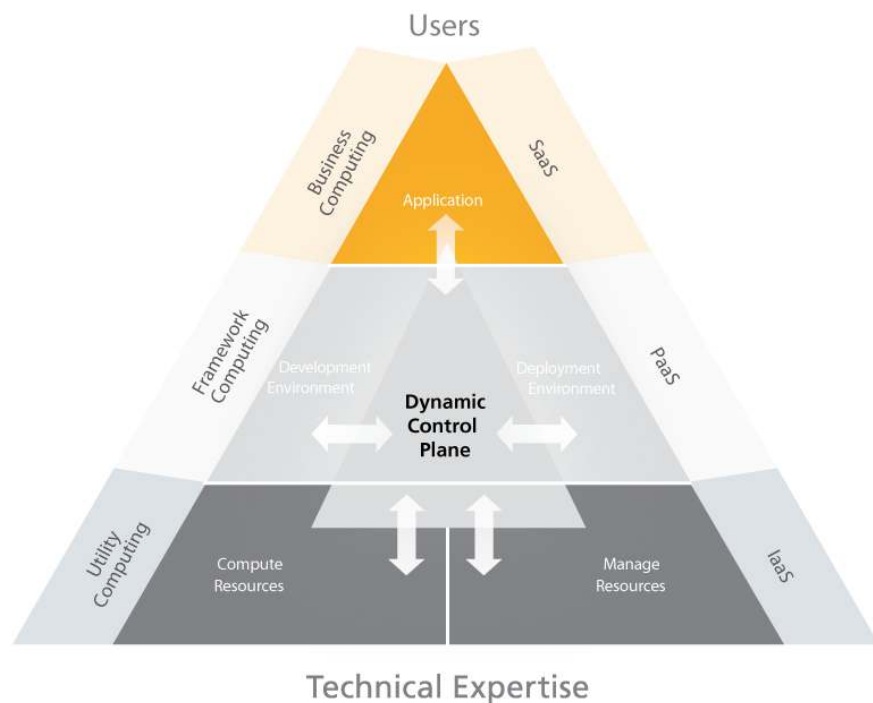

Figure 4. SaaS and cloud architecture

Application

Applications are the only component here. Whether the application is the result of building utility computing, followed by a platform, or it is simply an application deployed on a server, this is what users interact with. Users do not care how it was built, where it resides, or the compute resources required to deliver it. They simply expect it to be available when they want it, responsive and well-performing enough to be useful, and expect it to be secure regardless of where, when, and how they access it.

Putting it all Together

As we move up through the pyramid from the building blocks of infrastructure to the pinnacle of the application, the skill and knowledge necessary to build the components decreases. This is simply because each layer can be built on top of the previous without having to fully understand the underlying layer. An organization with limited infrastructure skills can readily purchase IaaS from a vendor and build their own platform (or several) upon that infrastructure without needing the expertise to completely build the infrastructure from scratch. This has been happening for years in the managed hosting business. The organization does not have to be hardware or networking experts and therefore, as an organization, they require less technical expertise.

Ideally, an IT organization will build a set of services that fit the needs of the business organization. They can build an IaaS solution within their own data center, a PaaS (or several) on top of that and even deploy ready-made applications in a cloud context, satisfying the needs of the business in an agile way. This enables them to react quickly to the ever-changing needs of the business.

In a similar manner, business units can now deploy solutions based on their needs and level of technical competency. They can deploy SaaS solutions via an external cloud provider or rely on internally available solutions; or they can build apps upon platforms or deploy their own IaaS solution.

The cloud architecture enables organizations to deploy solutions that naturally meet at the intersection of IT and business. Given the dynamic and non-static mappings between the applications and the resources-regardless of whether the IT organization built the application from the ground-up to meet the needs of the business or the business simply deployed their own solutions-cloud architecture enables them to seamlessly integrate at the most appropriate point for the organization. The organization, therefore, is able to maintain various elements of control (for example, security and compliance) while still providing the maximum level of agility to the business.

Making the Cloud IT-Ready

This newer dynamic cloud architecture may bring a wealth of new benefits, but it must still satisfy the basic mandates of enterprise IT and, if the internal systems are going to mesh with external ones, it must provide for consistent and reusable methods of interconnecting the divergent implementations at the cloud providers as well. In order to achieve mass adoption within by IT, the cloud and the dynamic control plane must be an enterprise-class solution and not just theoretical or a hodge-podge of workarounds and unique implementations.

When originally discussing the concepts of Infrastructure 2.0 (now an ongoing working group looking at formalizing the requirements of cloud infrastructure), there were several key components identified as being necessary to create and maintain this dynamic control plane, a critical element to Infrastructure 2.0. Amongst these where an infrastructure that was scalable, adaptable, extensible, and manageable. However, there are additional, basic fundamental needs of enterprise IT that must be taken into account. Any component that provides such comprehensive involvement in the applications and data must also be secure and be able to operate in real-time; it cannot degrade security or impede performance.

The dynamic control plane must always be available to connect users to the appropriate resources and it must do so in real-time without impacting performance. Despite the fact that the dynamic control plane needs to mediate and account for every user session and the movement of each application connection and each data access in order to be enterprise-ready, it must do so with little to no additional latency. As was previously stated, the application and the user experience of that application is the final, ultimate goal of any IT architecture. Correctly balancing that user experience with the controls and policies required by the business is the ultimate goal of cloud architecture.

Conclusion

Cloud computing is not a revolution. It is an evolution that has been ongoing for well over a decade, if not since the very beginning of electronic computing. The cloud is simply an architectural model that employs many of the same components used in datacenters around the world today in a more flexible, responsive, and efficient way. The primary difference is in how these components are tied together with a dynamic control plane which helps enlighten and inform the architecture about the rapidly changing requirements of today's applications, data and clients.

The dynamic control plane must be able to Intercept traffic as it traverses the cloud, Interpret the data and Instruct the cloud architecture on how to efficiently connect the user to the appropriate application instance. However, in order to be truly ready for enterprise deployment, it must also be scalable, adaptable, extensible, manageable, and secure with real-time performance. And in order to support this dynamic environment the cloud must be built with these ideals in mind, with each component-such as IaaS, PaaS, SaaS, users, and applications-designed to work together and as part of the dynamic control plane.

There is no doubt that some concept of cloud computing will become the primary method of delivering business critical applications in the coming years. There is little doubt that a move to cloud architecture will continue to provide the tools needed to better align business and IT. About the only thing that remains to be seen is whether the vendors and manufactures can deliver an enterprise ready dynamic control plane to bring the entire picture together to provide those benefits.