

## *"Software Defines Everything Mission"*

Every 5 years, you seem to hear a technology that is believed to change the industry and evolve in a manner that no-one has been preview to until it becomes just a buzzword that few implement in production to be truly called "Adopters".

With the needs of rising giants like facebook, amazon, google and others, the waves of cloud computing created a ripple effect over a decade ago with a need for a new way of thinking in the rigid networking world. A change to an existing problem can be addressed by several brilliant minds in many different mechanisms. The only thing constant in this world is change and unless you are onboard, you are extinct.

"OpenFlow" was one of the early methods to truly bring innovation with a disruptive means of a new open standard protocol compared to the traditional alphabet soups of large dominant players. The advantage for OpenFlow being open standard and driven by several large organizations and academia came as a breath of fresh air to the user community from being hostage to the narrow solutions defined by unilaterally focused large vendors. Unfortunately as much as the community increased and adoption flourished through several countries, the reality of OpenFlow replacing legacy deployments towards installations in a production deployment became scarce within a span of 5 years.

The basic principle was around unlocking the traditional approach of dependency on high priced smart HW and shifting the intelligence to a stack of software.

The solution was 3-staged :

1. An underlying hardware that could become OpenFlow compliant and be commoditized for all generic hardware vendors.
2. A centralized intelligent layer(also known as controller) that would distribute and be the network to communicate and process all data up/down the stack
3. A premium software layer above the controller platform to drive several applications focused on specific functions.

This solution ultimately did get adopted as you'll see below in several forms and shapes also giving rise to a new industry and entire category of startups.

As a start to the scheme of things came an array of open source openflow controllers and a community of app developers. Floodlight was one of the early ones to hit the road with several others like Ryu, Trema etc. Most of the later ones from startups and large organizations was a fork from these open source versions to build a more customized and feature rich option for customer deployment vs. a proof of concept lab.

Building a purposeful network function app wasn't as easy as it sounded. Several tried but to no avail...until there were network monitoring apps such as Big Tap from Big Switch Networks and traffic monitoring apps created by freelance developers.

As the realization of OpenFlow being a production ready protocol started dying down, Nicira made a pivot into establishing more of a Network virtualization focus bringing scalability to the agile demanding web players versus focusing on a net new protocol and a drastic approach to the stack change. This shift into Network Virtualization for Nicira brought about a \$1bn acquisition by VMware shaking the market and causing vendors to see the hype of possible monetization dying down with OpenFlow. Rethinking the strategy, we saw this emergence of network virtualization players and the terminology of Software Defined Networking later on becoming Software defined data center marketed by VMware.

The 2nd category of startups are those who followed in line with Nicira to redefine how Network Virtualization is the answer for the agility, business resiliency and reliability of the network in large cloud deployments. With not much focus on the hardware but building their solution heavily on the centralized controller and robust functional applications. A few of them in this space have been Midokura, Nuage Networks and PLUMgrid to lead with. As large organizations realized the need to play in this space and have an answer to this customer question of "What's your SDN story? ", the threat created several spin ups like Alcatel (Nuage Networks) , Juniper (OpenContrail) with a twist of having an open source and commercial solution to get customers intrigued enough to try the free version.

Huawei launched Sunlight as its answer while HP came up with its forked version as well. Dell believed in the OEM play and signed up Big Switch Networks, Cumulus Networks and few others in discussions.. Cisco and IBM not wanting to be left behind created a OpenSource SDN play with OpenDaylight Project and directed many of the discontented players from the ONF (OpenFlow Network Foundation) towards their efforts with a very high fee for the premium players club to enter. In parallel, Cisco also launched its commercial efforts of ONEpK ( its SDN controller) to the market and a Spin Up ( Insieme ) as a true answer to the Application delivery problem that customers are trying to solve with making infrastructure being simple.

Cisco, by its DNA cannot take the hardware commoditization route nor could it take the network virtualization route in isolation, hence it brought ACI(Application Centric Infrastructure) to the market that creates a very comprehensive story for why you pay the premium to maintain flexible infrastructure. The key was bringing the focus back to applications but creating the relevance for hardware and an end-end integrated story difference that Enterprise customers gravitate towards.

Within a matter of 2 years, Companies like Big Switch (the Other original OpenFlow Startup with roots from Stanford) were forced to market their mantra around SDN as well. This started a huge wave of companies calling themselves Software Defined Networking solutions to Software Defined storage to anything “SDX” with X claiming whatever market you were in.

OpenFlow now remains in the academic world recategorized as a SDN play with very minimal deployments found in Japan and some Asian countries. Vendors however have recreated their solutions to fit this curious “SDN” category as every CIO’s priority for the year seems to have “Exploring new technologies and understanding the role of SDN for my Org”.

Now, from its original 3 stack approach, the positive outcome has been the validation of original hardware vendors from China, Taiwan and others at the forefront like Quanta, Accton, Delta and more. Findings being hardware can be commoditized for several data center functions and the cost can be reduced 20X as a multiple while eliminating the large middle-layer companies like Cisco, Juniper, Dell and others.

With the hardware commoditization as a focus, a ray of hope emerged and companies like Cumulus Networks came into existence who built their software over bare metal hardware and presented a SDN-like solution with their approach of making networking simple, flexible and affordable. These solutions however don’t have the need for a controller or application. They don’t have a need for net-new protocols such as OpenFlow either. The focus is more about leveraging the proven success from the Server world of automation and bringing DevOps into the networker’s life.

You can chose to treat your automation platform as a centralized controller with analytics and other functions as apps running on an programmatic base. The fundamental difference here to the original OpenFlow/SDN 3 stack approach is you no longer need to have all 3 components dependent on each other for the full solution to work( Minimally intelligent hardware + Centralized Platform (brains of the operation) + Functional Applications).

Building this bare-metal ecosystem against the incumbent hasn’t been easy but the need for competition and alternative solutions from customers has driven this to even facebook open sourcing some of their installed deployments and proving scale is possible. This is now no longer a proof of concept lab solution but something that can be architected and designed to scale.

Amongst all these innovation waves through the enterprise world, service providers have had their own field of exploration with NFV (Network Function Virtualization). In a few words, it can be broken into 2 parts :- VNFs and NFVI. Virtualized network functions (VNFs) are software implementations of the different network functions that can be deployed on a

specialized infrastructure called (NFVI-Network function virtualization infrastructure). This brief doesn't dive into the details of NFV implementations but you probably won't have to dive into the details most likely unless you're running a carrier grade network.

Finally, automation and devOps is here to stay for the networking era that has been so hesitant to adopt it for decades. Whether you implement the basic layers of network automation for user interface, operations and infrastructure with Puppet, Ansible, Chef, SaltStack and CFEngine or leverage OpenStack. We will dive into Openstack as a follow-up session but if you're trying to build a Do-It-Yourself model, then having a Linux distributed OS for Network OS with OpenStack would be a great start to build your network stack for a smart operational framework.

The growth potential lies in many startups becoming consultants and service oriented companies where they provide "Enterprise Ready model for OpenStack" to be planned, designed and implemented in a customized manner for all customers ready to migrate from legacy environments. The 2nd growth space is for cloud solution providers who can design the best of the solutions across all vendors and customize some of the DIY solutions by vertical and offer a packaged offering by vertical whether it's Healthcare, Financial or Retail with all considerations embedded including Security. So, customers can focus more on size and personalized business demand versus every vendor criteria and cloud performance concern.

As a summary, it's essential that customers don't get perplexed and lost in the technology buzzwords and solutions proposed by different vendors, rather truly determine what's the problem they're trying to solve for ? What are their business needs ? How flexible does their underlying infrastructure need to be in order to support their business? What is the rate of business growth that their IT needs to adapt ? How sustainable and automated can their IT as a whole be in order to improve efficiency?

Step # 1 : Understand your skillset in-house and how they can be trained and augmented for the next-gen technology implementations

Step # 2 : Assess your infrastructure and have a better understanding of the usage vs. deployment and the type of functions driving the growth and demand.

Step # 3 : Plan for commoditizing your network hardware similar to your server infrastructure for as many functions as possible.

Step # 4 : Depending on your business vertical and segment ( Service Provider or an Enterprise Financial/Healthcare), create your list of critical business requirements that are of concern to you and how they map to your technical inefficiencies in-house today.

Step # 5 : See whether the Network Virtualization Story or whether a Distributed bare-metal solution with automation or a combination of both is the right answer for you.

Step # 6 : After you've mapped and created a skeleton of your solution, only then try to analyze the different vendors and see the best one that maps to your requirements at the right costs.

This approach is essential, else you land up being vendor focused and implementing a solution because it's new but not necessarily the right one for your environment and business. The key is understanding both short term benefits with the intent of achieving all the long term goals to help both your IT organization and the overall company's priorities as a CIO.

**Authored by :**

Meena Sankaran, Founder & CEO, Ketos Inc. (A Data Science Startup: currently in stealth mode). You can reach her at [meena.cohen@gmail.com](mailto:meena.cohen@gmail.com) for follow-up queries, and additional feedback as there are several detailed follow-up blogs intended in the upcoming pipeline. You can follow her on twitter [@meena\\_sankaran](https://twitter.com/meena_sankaran) and also on linkedin ([www.linkedin.com/meenas](http://www.linkedin.com/meenas))