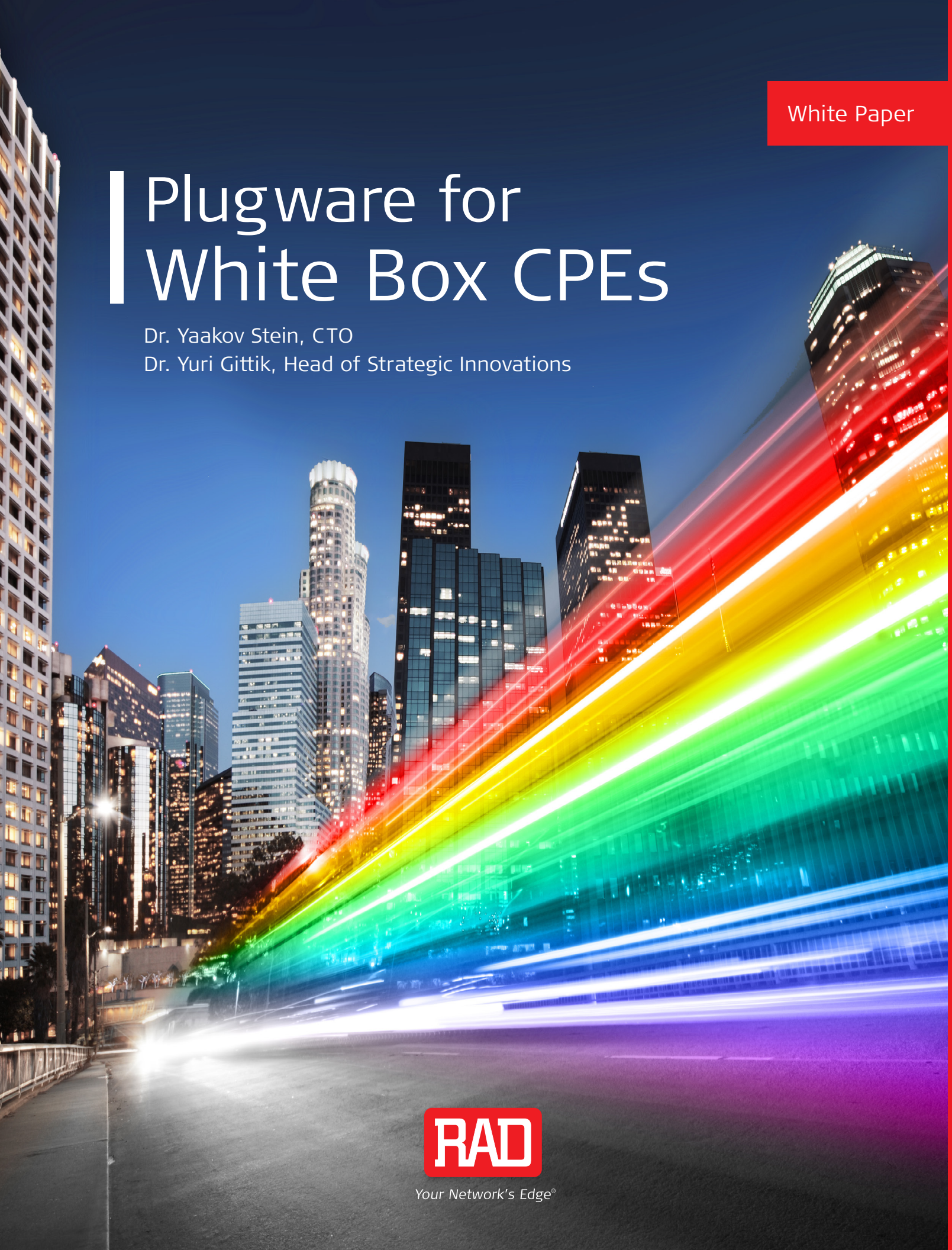
White Paper

Plugware for White Box CPEs

Dr. Yaakov Stein, CTO
Dr. Yuri Gittik, Head of Strategic Innovations



RAD

Your Network's Edge®

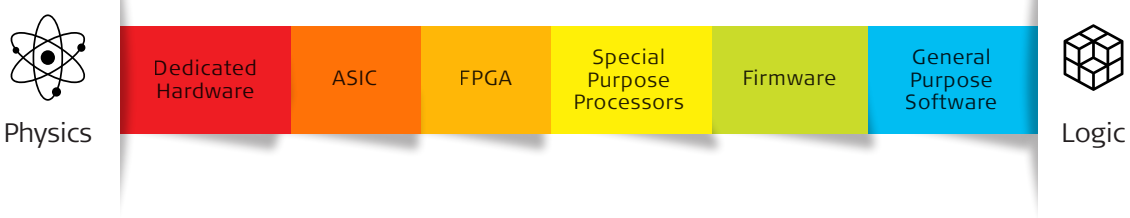
The concept of the white box CPE has taken the industry by storm. Recent analyst reports maintain that the vast majority of service providers intend to deploy them within the next few years, and that the market size is larger than that of any other NFV market segment[1]. This is quite astounding for a completely new apparatus, and for a network locale that until recently was considered ill-suited for NFV.

This mass market aspect of white box CPEs drives costs down, but comes with a catch. It has recently become clear that white boxes cannot, by themselves, rise up to all the challenges set before them. In particular, they cannot economically include all the plethora of different interfaces provided by physical CPEs before them, and cannot economically achieve high data rates for all the user plane functionalities furnished by physical CPEs.

In this whitepaper we propose a solution to this conundrum that maintains compelling white box principles while allowing the pendulum to swing back towards hardware implementations.

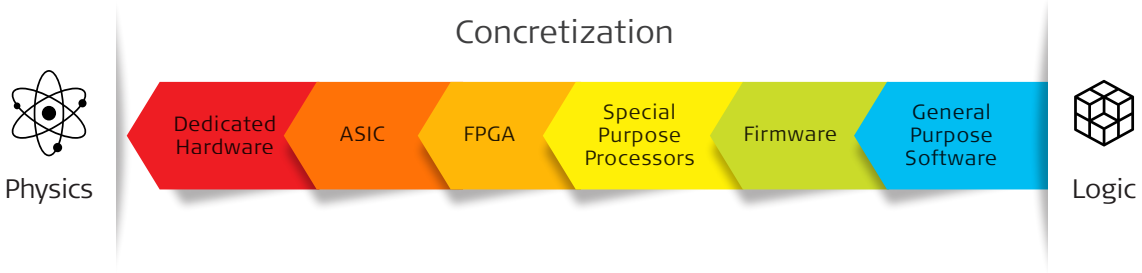
Virtualization

It is useful to think of implementation of any desired functionality on a scale extending from the most concrete (i.e., the most corporeal) to the most abstract (i.e., the most conceptual). Traditionally, R&D commences with an abstract idea and a list of definitions of what needs to be accomplished (i.e., at the far right of the figure), followed by a software simulation or prototype. If the product is deemed to have a good chance at success in the marketplace, it is usually initially implemented closer to the center of the spectrum, and migrates over time towards the left.



[1]Source: IHS Markit, 2017, NFV Strategies, Global Service Provider Survey

It is convenient to use the term concretization for the transitioning of an implementation towards the left of the implementation spectrum.

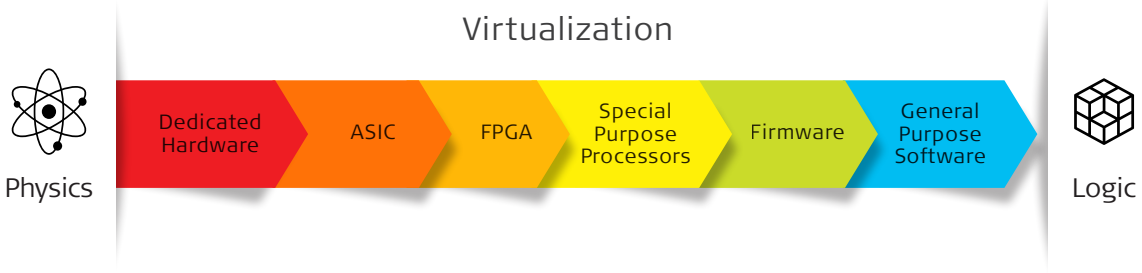


For many years, concretization was the major focus of networking R&D. This trend was understandable since it results in more efficient product design benefiting from:

- Cost savings
- Higher processing rates
- Product miniaturization
- Energy savings/lower heat dissipation

These characteristics were universally seen to be what service providers desired and what the industry needed.

A few years ago, something unexpected happened. It was called *virtualization* and took everyone by surprise. By virtualization we mean the transitioning of an implementation towards the right of the implementation spectrum, most frequently to the extreme of general purpose software.



The rationale behind virtualization is harder to grasp. Why would one want to take a highly efficient and inexpensive hardware implementation and replace it with a less efficient expensive one? The answer lies in flexibility, scalability, and reduced operational complexity.

Say a data center needs to allocate 100 CPUs to handle some amount of bursty application processing traffic, it can place these on 100 virtual machines (VMs) running on, say, 10 host processors, thus benefitting from statistical multiplexing

of compute resources. And if for some period of peak activity 500 VMs are required, these can be flexibly added to the workload, and freed up when no longer needed. Also, applications designed for different environments (available libraries, operating system version) or even different CPUs, can run on a single standardized hardware platform, negating the need to acquire and maintain all these platforms.

Network Functions Virtualization

But not only CPUs can be virtualized. In recent years the trend towards virtualization has spread from CPUs in data centers to network functions virtualization (NFV), wherein network functionalities (such as routers, switches, firewalls) that were efficient but implemented as vendor-proprietary hardware, have been re-implemented in software (as virtualized network functions, or VNFs) that can run on commercial off-the-shelf servers (COTS).

The reasoning behind this move was twofold: First, using COTS servers enables the CapEx to scale over time in concert with user demand. Second, the softwarization and concentration of functionalities enables OpEx savings due to simplifications in management and network orchestration (MANO).

It is not immediately clear that it makes sense to supplant hardware appliances, especially those implementing computationally intensive user-plane functions, with pure software VNFs. This substitution understandably results in some reduction in efficiency, and may therefore trigger a one-time CapEx increase, and over the longer term may result in higher energy budget and real-estate OpEx costs. The plan is for these cost increases to be more than offset by revenue increases resulting from shorter time to market, and more importantly OpEx reductions resulting from more efficient operations.

How can operations become so much more efficient? One of the incentives originally underpinning NFV was the ability to relocate many network functionalities from deep in the network to data centers – the so called *cloudification*. As previously explained, the use of virtualization in the data center resulted in significant savings, and re-orchestrating the network intelligence into the data center was expected to result in similar savings.

And indeed there are many functionalities that can profit from this move. Chief amongst these are functions that have traditionally been implemented only at the heart of the network, such as LTE's Evolved Packet Core. The vEPC was demonstrated with much fanfare, but in actuality is simply taking a function that was always implemented as software running in the core on proprietary boxes,

and porting it to software running in the core on COTS boxes. This may definitely lead to cost reduction due to opening the market to new vendors (software houses without previous networking experience), but is far from the revolution expected from NFV. Another class of functionalities are those conventionally implemented throughout the network, but that can be migrated to the center. For example, distributed routing requires a high degree of intelligence and thus substantial processing power in each and every router. SDN advocates leaving only dumb white box switches in the network, migrating all the (routing and control plane) intelligence to a centralized controller. This too can definitely be advantageous, for example by enabling optimizations that can only be carried out by an omniscient centralized controller; but that revolution has already been claimed by SDN!

And we need to realize that there are functionalities that must remain in the conventional locations. Here are two examples: It makes no sense to carry out encryption solely near one side, leaving the access link unprotected. Proper estimation of a user's quality of experience requires performance measurements to be made as close as possible to that user.

For these reasons, back in 2013 RAD proposed Distributed NFV (D-NFV) which advocates virtualizing network functions and locating them wherever it is most beneficial to do so. The placement decision needs to take into account service requirements (e.g., maximum latency), QoE, economies of scale, availability of real-estate and computational resources, transport costs, security, regulatory requirements, and so on.

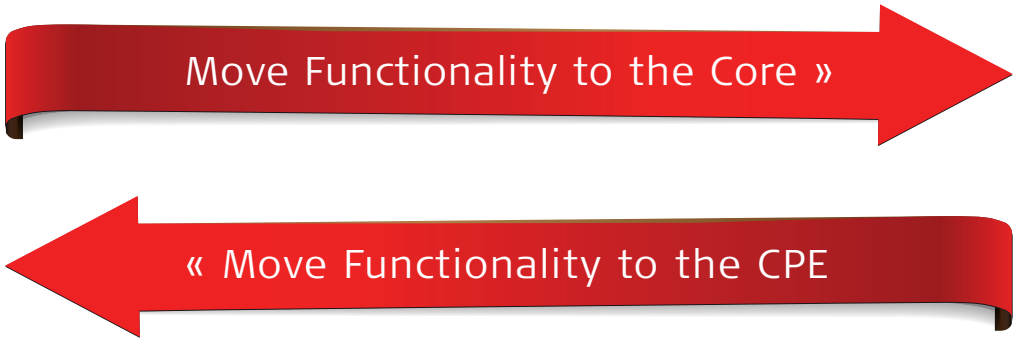
Virtualizing the CPE

An extreme case of D-NFV is Edge NFV, namely placing the VNF right on top of end-user, or at least at the customer premises. While originally ridiculed by those who saw the main advantage of NFV as enabling moving functions from the network to the core, the vCPE solution for business customers has emerged as a priority NFV use case. According to an IHS Markit survey, 100% of responding operators plan to immediately deploy business vE-CPE (also known as enterprise vCPE) for managed services using software VNFs, e.g., firewalls, IPS/IDS, SD-WAN, WAN optimization[2].

In particular a CPE comprising a COTS server that can host multiple VNFs, precludes the regular practice of deploying multiple physical appliances at the customer site. This capacity to reduce OpEx caused by clutter, space requirements, mis-wiring, and truck rolls, has been widely acknowledged by vendors and service providers alike.

[2]Source: IHS Markit, 2017, NFV Strategies, Global Service Provider Survey

As a matter of fact we have recently witnessed a complete reversal of judgement regarding CPEs. The initial understanding was that "virtualizing the CPE" meant leaving only a *thin* CPE at the customer premises and moving all virtualized functionality to the center in the "cloud". The more modern concept admits the diametrically opposed alternative of a pure software CPE situated at the customer premises. This so called universal CPE (uCPE) is based on a standard compute platform upon which all requisite networking functionalities are implemented as VNFs.



CPE White Boxes

Analogously to SDN's white box switch, a standard compute platform capable of acting as a VNF-hosting network element has been called a white box server. Since the white box server is simply a standard COTS (usually x86-based) platform, its adoption as a uCPE may definitely lead to cost reduction due to opening the market to new vendors (for example, server vendors without previous networking experience).

The white box server CPE provides a rather compelling story (and hopefully, business case), enabling extensive application functionalities (such as firewalling or PBX), along with inexpensive basic demarcation (such as connectivity and performance monitoring, rate limiting, end-to-end encryption) to be flexibly deployed and upgraded. The deal is clinched when we do not know a priori precisely which functionalities will be needed. A white box CPE can be programmed to perform whatever networking tasks are needed, and endlessly reconfigured as required.

However, the basic uCPE white box does suffer from two drawbacks.

First, in order to keep costs down, modern white box server platforms are outfitted with relatively few external interfaces, usually only USB and Ethernet ports (RJ45 and/or SFP cages). The conventional solution for those cases where other interfaces are required (for example, DSL, GPON or TDM) is to provide an external converter that adapts the needed interface to USB or a network port.

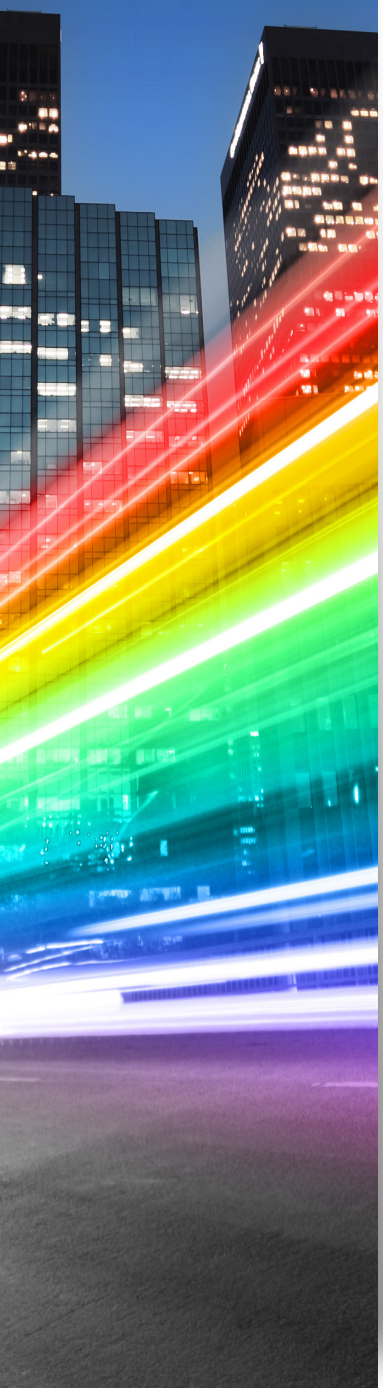
Second, and more significantly, white box CPEs seem to be mostly suitable for relatively low data-rates solutions, since implementation of some resource-intensive data-plane functionalities (such as encryption) at higher rates is not cost competitive with hardware implementations. For example, high-rate synthetic OAM generation, deep packet inspection, traffic policing/shaping, encryption algorithms, video transcoding/transrating, are all functions better performed by hardware.

Empower the White Box with Hardware

The white box CPE approach can be effectively maintained by augmenting the basic white box with minimal hardware providing interfaces when needed and accelerators for computationally intensive data-plane functionalities. This method leaves the lion's share of CPE functionality in software, including control and management, and data-plane functions that are not computationally intensive.

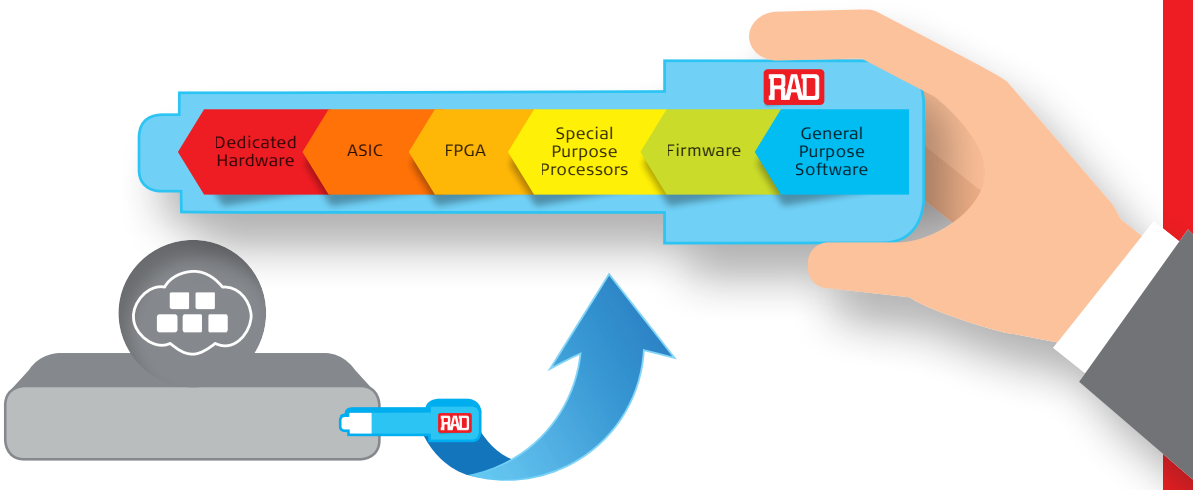
Running VNFs on a proprietary platform or embedding interfaces or accelerators into COTS hardware clearly violates the openness which is so fundamental to the white box approach. Preserving this approach necessitates decoupling the hardware and software, while maintaining management simplicity. There are two ways to achieve these challenging goals.

The initial approach, introduced by RAD in 2014, was that of the *gray box* or Whitebox+. In this approach an existing efficient hardware CPE was retrofitted with a COTS computational (NFV Infrastructure – NFVI) module. This module may be pluggable in the sense that the CPE could function without it, and upgraded only when NFV support is required, transforming it into uCPE. This approach enables flexibility in adding arbitrary VNFs, and can interface to modern NFV orchestration systems, but still feels more in the traditional hardware camp than in the revolutionary white box one.



A new approach, solidly in the white box camp, exploits augmentation via pluggable hardware. In this approach a COTS white box server is the basis for the uCPE. Pluggable hardware modules provide needed interfaces, or hardware acceleration, or both. If this software platform is sufficient to carry out all the required functionality, then nothing further is required. If interfaces are missing or the data rate or the functionality complexity are above what the white box can handle, specially designed hardware is plugged into the whitebox.

2nd Wave of Concretization - Plugware



The pluggable hardware takes advantage of standard interfaces available in existing white box servers, such as SFP designed for network ports, M.2 or U.2 designed for solid state storage, AGP designed for graphics cards, USB or PCI express designed as more generic interfaces. Once the hardware vendor provides the necessary device drivers, the pluggable hardware is recognized by the white box operating system, and becomes available for use. The drivers provide the software interfaces needed to detect, configure, and maintain the pluggable using standard means. In either case – interface or accelerator – the driver software converts the pluggable hardware into a standard device. This device may be accessed transparently by the white box operating system, or may be explicitly integrated into software components implemented as bare-metal processes, in containers, or in virtual machines.

The pluggable interface or accelerator may only realize a portion of the network function to be employed. For example, if the network function has both user plane and control plane portions, it is often more sensible to perform the latter in software.

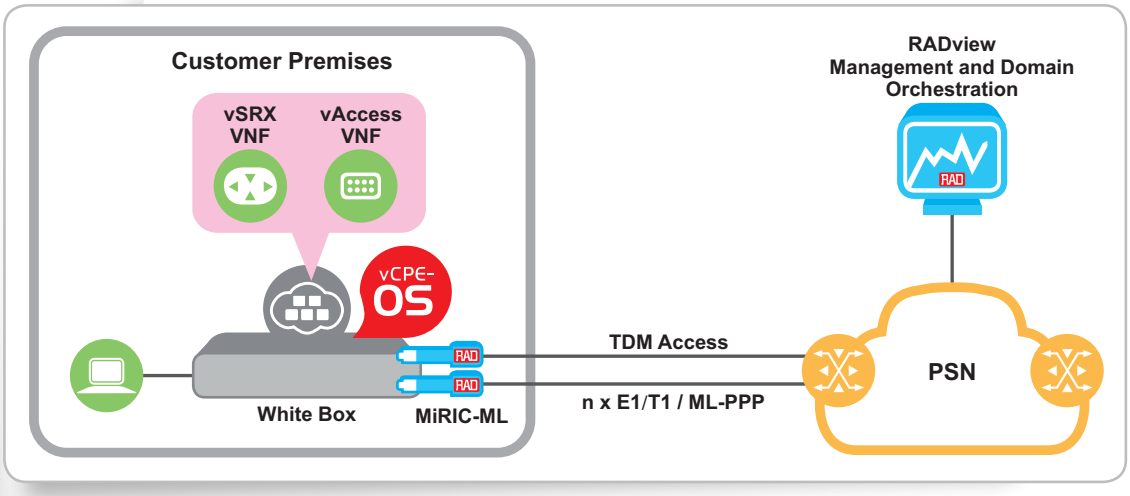
In many cases, even the user plane may be decomposed into VNF components (VNFCs), only one of which interfaces the outside world (requiring a pluggable interface) or only one of which is so computationally intensive as to necessitate its implementation in hardware. When using a pluggable accelerator, it is recommended that its functionality be also available in software, so that the VNF main software can fall back to a pure software implementation (with reduced performance) if the pluggable hardware is not detected. In some cases it may be operationally simpler to implement the entire network functionality (not merely a VNFC) as pluggable hardware (with perhaps some manual configuration). If a software implementation is also available, it is up to the orchestration system to choose whether software or pluggable hardware is merited.

vAccess

RAD's vAccess is a flexible product that comprises an assortment of software and pluggable hardware components. On the one hand, it can run as a conventional VNF and can optionally be service chained with other VNFs from RAD or from third parties. On the other, it can exploit RAD pluggable hardware, supporting interfaces and attaining performance hitherto unobtainable in the white box environment.

An illustrative use case is *vAccess for TDM access* links.

Consider a generic white box uCPE outfitted with network facing GbE SFP cages. Such a white box is suited for a large percentage of modern installations, but does not provide the physical interfaces required for those locations served by bonded TDM links (e.g., 4 Mbps over 2*E1 or 6 Mbps over 4*T1).



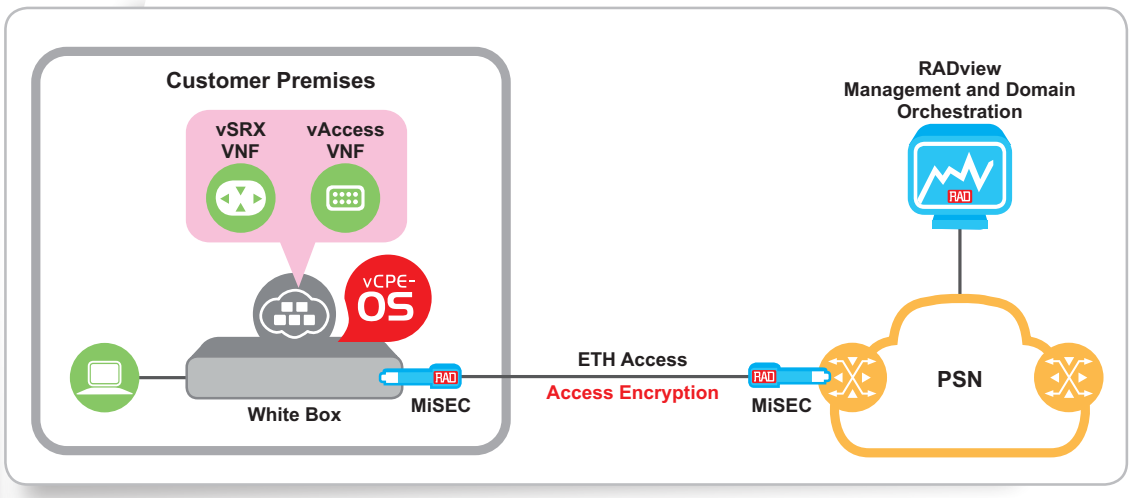
RAD's vAccess solution to this problem comprises several MiRIC-ML pluggable interfaces (transporting Ethernet or IP over TDM), the required device drivers, and a RAD-provided VNF to process the legacy MLPPP bonding protocol. Third-party VNFs, such as commercially available router software, can be chained after the vAccess VNF. The solution is depicted in the following figure.
This vAccess solution has been successfully deployed by a tier-1 service provider.

The same vAccess VNF may be equipped with additional functionality, such as collecting performance data by monitoring both networking and NFV segments.

A second example is *vAccess for communications security*.

Consider a generic white box uCPE with GbE uplink, which is required to protect all its traffic using IPsec. Performing the encryption in pure software would require either an expensive white box with a powerful CPU, or an intermediately priced white box of which no computational resources are left for needed functionality, e.g., firewall.

The vAccess solution to this problem comprises a MiSEC pluggable IPsec/MACsec accelerator. This SFP-based device performs AES/GCM wirespeed encryption, authentication, and integrity-check functions, leaving the key distribution to software of the vAccess VNF running on the white box. The solution is depicted in the following figure.



A prototype of this vAccess solution was publicly demonstrated at the MPLS+SDN+NFV World (Paris, April 2018) and BCE (Austin, May 2018).

Conclusions and Directions

While the white box uCPE story is compelling, it has become clear that white boxes cannot, by themselves, rise up to all the challenges set before them. This conclusion is a consequence of cost savings measures, which mandate that white boxes:

- Be equipped with only the most prevalent physical interfaces
- Be furnished with CPUs of the lowest possible cost
- Are expected to execute multiple network functionalities

We believe that the natural next stage in the ebb and flow of software vs. hardware is the return of hardware in the form of open pluggable modules. Being pluggable implies that standard white boxes can be field upgraded to obtain new capabilities and/or cost-performance levels. Being open means that the hardware can be readily integrated into any existing management and orchestration system.

By coupling pluggable hardware with software (device drivers and VNFs) RAD's vAccess facilitates adoption of plugware and tweaks the white box uCPE business case.

Initial vAccess use cases demonstrate its ability to assist service providers in maintaining the white box uCPE strategic direction, while minimizing cost and inventory.

vAccess is an evolving product line, and RAD invites service providers and vendors to help co-define capabilities and functionalities.