CREATING AGILITY & EFFICIENCY AT SCALE: THE ECONOMIC ADVANTAGES OF OPEN ARCHITECTURE PLATFORMS IN NFV DEPLOYMENTS

SUMMARY

Burgeoning demand for new applications and a wide range of use cases has created a powerful motivation for communication service providers to become more agile in how they deliver their services. A dominant model for enabling this innovation is adoption of cloud-native designs using general-purpose hardware and flexible, openly architected software in which functionality can be enhanced more rapidly than in legacy designs.

One model for achieving this that has been broadly evaluated over the past several years is running operators' network functions as software in virtual computing systems of the type used in cloud computing services that web-scale providers have used and which have enhanced their popularity and innovation. Under the umbrella of network functions virtualization (NFV) operators have researched whether their networks can become as agile as the cloud and have performed extensive specification, development and proof-of-concept testing to evaluate whether NFV will support their services with the flexibility they need.

Having proven the essential functionality of numerous types of network appliances and applications in lab tests and trials, it is now time for operators to put the platforms to use in progressively larger deployments and determine how well they can meet their goals. Given the range of approaches that developers have taken toward implementing the virtualized model, it is also timely for operators to compare the effectiveness of alternative designs they have considered.

REPORT HIGHLIGHTS

- **5 year cumulative TCO** of an open architecture POD (OAP) in this operator's design **is 53% of the tightly bundled POD** (TBP)
- **Capex** of the OAP **is 47% less** than that of the TBP
- **5 Year opex** of the OAP **is 57%** of the TBP
- **Creating a new application or service** in the OAP **requires one-third the time** on average as the time to create the same service in the TBP
- **Underlying reasons** for the advantage **are openness** in components' hardware + software designs, and extensive **automation** based on open, standard **APIs** + information models

One prominent Tier 1 service provider did such a comparison based on extensive analysis and testing to deploy NFV at scale in its national services infrastructure. The model it chose as the foundation for its deployment is based on open, modular and extensible designs for the key elements in its service delivery PODs. The deployment is made efficient and dynamic by extensive software integration using well-known APIs, by adhering to the reference models the operator embraced in its research into NFV, as well as on other open architectures it employs in its network, application, and OSS/BSS deployments.

The team of suppliers supporting its new virtual infrastructure platform determined with the operator that it would be informative to compare the relative efficiency and advantages of the open architecture platform to the more tightly bundled alternative it had considered. The analysis would consider the main architectural differences between the approaches, the operator's goals in deciding its course of action, and the resulting economic advantages the operator obtained by moving forward on its design. Based on ACG Research's experience analyzing the nature of architectural transitions such as NFV in service providers' environments and its strengths in determining the economic advantages on both cost and revenue-generating aspects of designs, the team engaged ACG to do this analysis.

This report is the outcome of that work. It is organized into the following sections:

- The Rise of NFV in Service Providers' Environments
- Beyond Theory & POCs: Realizing the NFV Vision
- The Case of a Tier 1 Operator's Deployment of NFV at Scale based on Open Architecture Platforms
- The Operator's Perspective on Its Deployment
- Dimensions of the Analysis
- Architectural, Operational, and Economic Attributes Compared
- Economic Measures Employed in Our Analysis
- Results of the Economic Comparison
- Conclusion

The Rise of NFV in Service Providers' Deployments

The benefits of incorporating applications and network connections pervasively into our lives have accelerated at a rapid pace over the past several years. Spurred by innovations in mobile communications and connected things, along with incorporating smart devices into our buildings and homes, users in every domain are motivated to reach for the convenience and efficiency the innovations provide, continuously, and everywhere.

Although these innovations have delivered enhancements to mobile devices and introduced smarter things into our workplaces and homes, they have also stimulated a forward-looking focus on the part of network service providers to transform themselves, their business models, and the platforms they deploy for generating revenue. They have analyzed these factors using agile, software-driven designs like those that have powered web-scale application providers. Accomplishing this transformation has become a competitive imperative for those operators as the pace of innovation in all the domains adjacent to them continues.

An important component of this transformation is using cloud computing technologies to power network functions within the new model of NFV. Adopting NFV brings the promise of eliminating rigidity in operations based on legacy, purpose-built platforms that are too slow to evolve and too costly to run. Moving to more elastic, software-driven, openly architected and economically designed platforms will be a key element in their transformation.

Beyond Theory & POCs: Realizing the NFV Vision

With this as the inspiration to transform, the job cannot be achieved overnight or simply by describing it. It requires broad consideration of the ways the technologies will be deployed and the way teams will work together to get it done. And it requires a steady cross-functional effort in each of the lines of business and operations that have a stake in making the transformation real.

Extensive analysis, design, development and testing have occurred over the past three to four years as views on how to make NFV part of the solution have been formed. During this time, progress has been made in many areas: detailed models of virtual infrastructure and its use in NFV have been developed; control software has been developed for managing the virtual functions; and innovation in underlying protocols and interfaces for integrating NFV into production offerings has occurred.

Subsequent testing in pockets of design has been productive, but now capabilities in many of the elements have moved far enough along to allow them to be used and evaluated at greater scale.

During these early stages of broader NFV adoption, two design patterns are most common for operators to consider: using open, best-of-breed platform combinations and considering use of more tightly bundled, vendor-specific platforms for a certain range of functionality. Each has its pros and cons, which need to be evaluated by operators when making their choices. Each of them is in focus with the analysis in this report.

In general, open, best-of-breed combinations emphasize freedom of choice among suppliers to address requirements in each portion of a deployment, so long as the chosen offering meets the operator’s specific functional and interoperability objectives. This approach allows an operator to consider advances within each portion or domain independently over time and to make step function improvements in performance, capacity or cost at a time of its choosing. In this mode, the operator and its suppliers are bearing the burden of certifying the integration of the solution for its purposes.

The alternative model, which we have described as a tightly bundled approach, is based on a stricter coupling of elements in a design, often bound by a proprietary implementation. In such offerings, the range of options between which an operator can choose can be artificially constrained, and the pace at which innovations can be adopted is limited by the pace at which the supplier of the tightly bundled platform (TBP) can achieve them. A TBP is often chosen when an operator wants to limit the amount of responsibility it is accepting for the interoperability among its solution components, and they are willing to pay the price in relatively reduced flexibility and often higher base price that a TBP brings with it.

A good example of tight bundling relevant to NFV is support of a limited, specialized storage or networking mode by a supplier to make a given function possible at the expense of leaving the interface open for support of other modes at the customer's choice. More on these considerations is described in the platform comparisons in the report.

The Case of a Tier 1 Operator's Deployment of NFV at Scale Based on Open Architecture Platforms

One prominent Tier 1 operator has chosen to deploy virtual network infrastructure and cloud-native applications using a robust open architecture platform, leveraging best-of-breed solutions in each infrastructure function. Unlike the direction often chosen in many early of the early trials of NFV where solutions of modest scope have been tested in tightly constrained implementations, this deployment is intended to be a general-purpose platform for use by many tenants in the operator's business. It is planned as a distributed, open, software-driven platform that scales efficiently and is quickly responsive to the operator's need to innovate and evolve.

The team of companies engaged with the operator in creating this platform thought it would be useful to analyze its economic advantages in comparison with the more tightly bundled solution the operator considered as an alternative. Shedding light on these advantages and the attributes that enable them would be of value to many parties working on NFV in their own companies and in the industry at large.

The team of companies engaged ACG Research to analyze the economics in this light. ACG has analyzed deployments of NFV, SDN and related service provider infrastructures in a broad range of use cases since their inception and brings deep experience to the task.

The Operator's Perspective on Its Deployment

This platform, which the operator refers to as its cloud service delivery platform, is the foundation for its transformation to a continuous innovation, continuous delivery mode of operation. It is open for evolution and automated in both daily, ongoing operations and in new service introductions. It is focused on leveraging the economics of virtual system infrastructures as widely as possible, while meeting technical and application delivery requirements.

A key element of the operator's design is supporting innovation within well-defined domains that improve the efficiency, agility, and elasticity of its platforms but which do not necessarily require modification of functions in other domains. This aligns generally with the tenets of NFV as envisioned by the industry and elaborated in the ETSI NFV Industry Specifications Groups' work. The philosophy of this approach is shown in the Reference Architecture for NFV published by the ETSI working groups, in Figure 1.[1]

[1] *Network Functions Virtualization: Architectural Framework*, European Telecommunications Standards Institute, 2014.

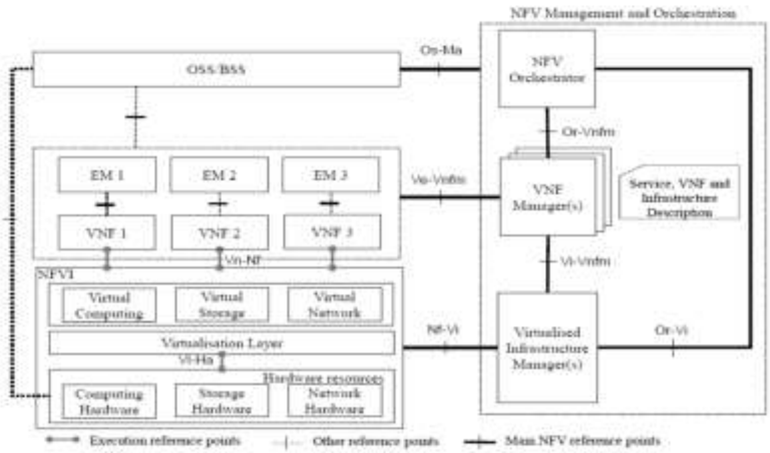

Figure 1. ETSI Reference Architecture for NFV

The operator expects the platform to be an integral part of its development and operations (DevOps), Platform-as-a-Service approach to operations. Its integration with application development and service orchestration systems is based on well-defined information models and reliable, extensible workflows to allow for smooth integration of services and clear operational controls.

In this context, the operator understood that deploying an open and efficient **foundation** for supporting these approaches was one of the most important initial steps it could take. Without an agile, efficient, scalable foundation, the services seeking to operate in such an environment would ultimately fall short of their goal. Thus, the operator's initial deployments and this analysis are focused on the agility and efficiency of the NFVI in its overall deployment.

Dimensions of the Analysis

The operator is deploying its NFV infrastructure at a significant number of locations within its national service delivery infrastructure. The sites vary in function and size and include large, centralized installations focused on core network services, management of operations, and aggregation of subscriber and service management data to support its offerings. They also include smaller, distributed installations that perform local network aggregation plus management and application functions best executed at the local level.

The initial deployments are at **seven locations** in its national footprint. Compute power is based on nearly 2,000 servers (dual-socket, multi-core x86 servers with total cores approaching 40,000). The deployment includes 350 local network switches, 28 petabytes of HDD and SSD storage in nearly 100 arrays, and 1,000s of workload instances. Ultimately, the scale of the installations will increase, and extension of the blueprint to additional sites will occur.

In the analysis, we focused on a **standard configuration** of the POD as deployed in one of the core operations sites. The aim was to keep the scope manageable while focusing on the elements consistent in the designs employed throughout the national footprint. In this way, the essential economic conclusions drawn could be defined and applied to larger deployments, if desired. The POD in the analysis is sized to support **a range of virtual network functions** and closely-related cloud-native applications. It is inherently designed for multi-tenancy and dynamic workload management.

We focused on the elements in the virtual systems infrastructure the operator deployed and as generally defined in the ETSI working groups' NFV reference architecture previously mentioned. These are outlined in orange and highlighted as NFVI in Figure 2.

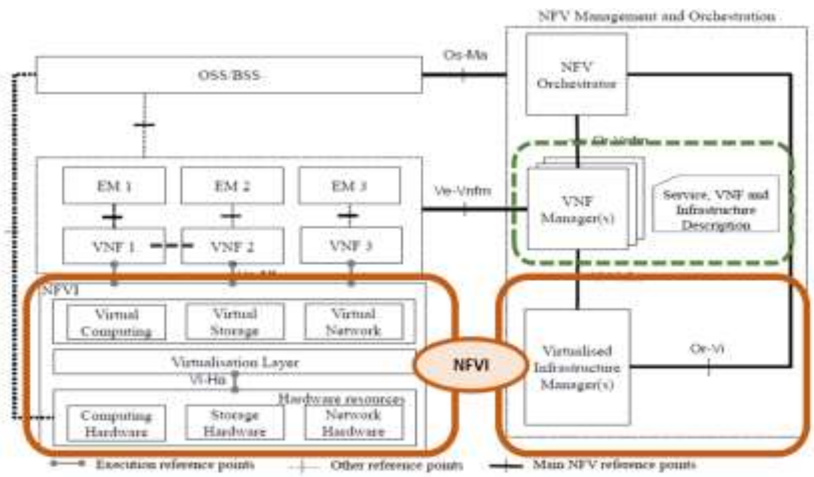

Figure 2. ETSI/NFV Architecture, Highlighting NFVI & VIM for Economic Analysis Project's Focus

The designs the operator considered are both **broadly** aligned with the ETSI model. Because we are highlighting the results achieved in the open architecture design in this report, the individual products that comprise each of the elements are highlighted in the list and referenced in more detail when discussing their contribution to results.

The products included in the open architecture design, and their suppliers, are:

- Physical servers: **Dell EMC PowerEdge**
- Server OS and Hypervisors/Virtual Machines: **Red Hat Enterprise Linux (RHEL) and KVM**
- Server-resident and network-attached physical storage: **Dell EMC Modular Disk Arrays**
- Virtual storage software and management: **Red Hat Ceph Software-Defined Storage**
- Physical and virtual network fabric: **Big Switch Networks Big Cloud Fabric (BCF), delivered via Dell S6000 and S4000 switches**
- Virtual Infrastructure Manager (VIM) Software: **Red Hat OpenStack Platform (OSP)**
- Firewall/IPS Security: **F5 Networks Viprion Application Delivery Controller**[2]

The open architecture platform (OAP) was compared with generally equivalent functionality in each NFVI category in the tightly bundled platform (TBP) design. The TBP included a mix of blade and rack servers for its compute requirements. Blade server chassis were used for POD control and management

[2] In the initial deployment security for the POD was enabled with a physical firewall/IPS platform. As time progresses and the use of security protections evolves within the virtual infrastructure evolves, virtual form factors will be progressively incorporated.

functions; rack servers were used for general-purpose workloads. Its network fabric was tightly coupled with its server infrastructure, using a unique and relatively rigid approach for server-to-server and server-to-network communication at both Layer 2 and Layer 3. Although the design enables these elements to work successfully with each other in a limited way, over time the rigidity restrains innovation in each category and introduces inefficiencies in integration and deployment with other elements compared to its open architecture counterpart. Similarly, the storage solution of the TBP uses a relatively more fragmented underlying approach to supporting block and object storage operations for NFV compared with the open platform's solution, again extending the time for integration and deployment and introducing extra costs. The tightly bundled platform, although superficially appearing closely integrated on several of its dimensions, introduced constraints in the efficiency and flexibility of operation and over time in the freedom to innovate within categories.

The specifics of the POD in our analysis, the number and type of each infrastructure element are shown in Table 1.

Element	Quantity in POD	Notes
Racks	12	10 compute and storage racks, plus 2 service, control and connectivity racks. Compute + storage racks are the where VNFs + applications run. Service, control + connectivity racks house management + control servers, security appliances, and nodes connecting this NFV POD with other networks and systems
Physical servers	225	These are the compute nodes that run both management and VNF/application workloads
Server OS + hypervisor software	225	Underlying compute node software for POD, OpenStack and virtual storage management + for running VNF/application workloads
Physical storage	10	4 PB of hybrid HDD + SDD storage for server/POD, OpenStack, and VNF/application use
Virtual storage	40	Ceph OSD (Object Storage Daemon) + monitor nodes managing distributed storage
Physical network	39	Leaf + spine fabric for communications within the POD, and between the POD and other networks
Virtual network	225	Virtual network nodes supporting OpenStack services + VNF + cloud app communications within + between servers, racks, remote applications + PODs, + other networks
Firewall/IPS nodes	2	POD boundary firewalls initially, to secure POD resources internally and externally

Table 1. Core Network Site Elements

A high-level view of the hardware configuration of this POD is shown in Figure 3. Several elements in the configuration deployed are not shown (including patch panels, cables, and out-of-band management servers). These are essentially identical in the two configurations (other elements of each solution are unique). Thus, the identical and less distinguishing components are not shown.

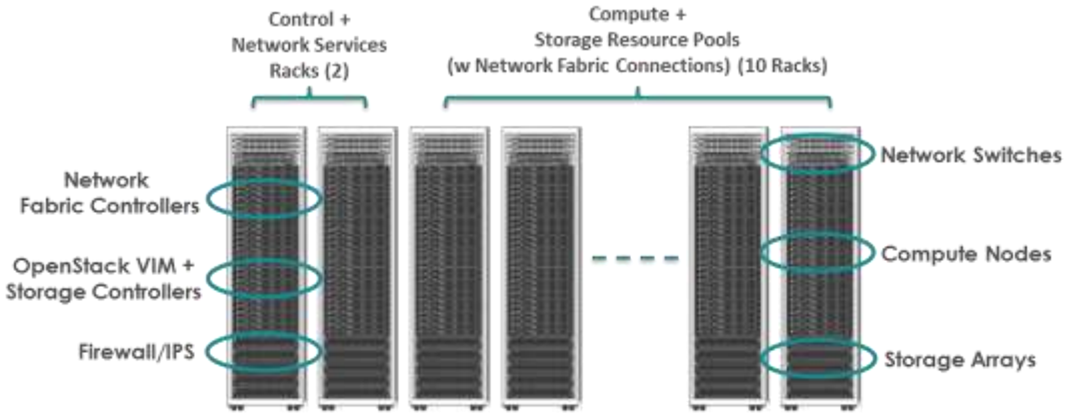

Figure 3. High-Level View of the Open Architecture POD

Architectural and Economic Attributes Compared

In this section, we highlight the architectural and the economic attributes of the PODs we considered in our analysis.

Detailing all the architectural attributes the operator has considered for its deployments is beyond the scope of this report, though it is worth highlighting several that have a major influence on how efficiently either design can meet the operator's goals for continuous innovation, elasticity and scale as it is deployed into the operator's existing service infrastructures. Each is a dimension we considered to determine the effectiveness of the two designs.

Briefly, the operator is focused on ensuring its platforms:

- **Integrate well with** its existing, large-scale, customer-facing and **revenue-generating networks**. The NFV solution is integrated into the operator's public Layer 2 and Layer 3 networks. It is also integrated into internal operator's private Layer 2 and Layer 3 networks.
- **Integrate well with** the operator's **business and operational support systems** from billing and customer support to traffic analysis and end-to-end orchestration. This is connected to how well the platforms support higher level data modeling and service automation, including integration with its Ansible and OpenStack HEAT orchestration systems.
- Are **simple and efficient to install, operate and upgrade** from the beginning to the end of their useful life.
- Are **open** between elements of the NFVI and to other systems with which they work.
- Are **versatile** to support VNFs with different functional requirements. Some VNFs are throughput-intensive, others require multiple virtual machines to operate, and many need service chains with multiple attributes and configurations. Incorporating a diverse range of VNFs and enabling them dynamically and resiliently is a critical success factor for the operator's NFV deployment.
- Are **resilient** in overcoming faults and failures as automatically and transparently as possible.

- Support **distribution of management** across a widely-distributed set of installations, especially as new services such as IoT, 3D video and virtual/augmented reality gain favor with customers, and the scale of the NFV infrastructures continues to expand.
- Demonstrate **best-in-class economics** in the product categories they support.

Economic Measures Included in the Analysis

For each of the PODs (the POD based on open architecture design and the POD based on tightly bundled design) we concentrated on **two major economic** metrics**: lowest total cost of ownership (TCO) and greatest agility in new service creation** (represented by greatest efficiency in new service creation and fastest time to new service revenues). We describe each of these measures at their basic level and describe the results of our analysis from applying them to the two alternative designs in the section that follows.

Cumulative Total Cost of Ownership

Total cost of ownership is the sum of capital expenditures (capex) and operational expenditures (opex) for a given solution. It is a crucial metric for whether the benefits it produces are worth the costs of delivering it. Both capex and opex have their own components and methods of calculation. **Cumulative** TCO sums TCO components over a period. In this analysis, we considered the TCO of the two designs over five years.

Capital Expenditures

Capex is the cost of purchasing the elements of a design. In this case, it is the cost of the servers, storage arrays, network switches, firewalls, and the software used to run them. We use **company public list-prices** in our capex costs. We assume the capital for the purchases is expended in total at the time the elements are purchased (in this analysis the bulk of the purchases are in Year 1). The configurations are based on the capacity and deployment requirements defined by the operator and summarized previously. Multi-year license pricing for options of an equivalent nature between the PODs is employed for the software.

Operational Expenditures

Opex includes the costs of running the environment for the deployment **as well as the personnel costs of managing it throughout its life cycle**. The environmental costs in our analysis are the costs of electrical power and cooling for the configurations over five years. Personnel costs are costs incurred at different stages of the life cycle, including design, deployment and operation across five years. To arrive at these, we:

- Evaluate the amount of work to do a defined set of tasks, based on our understanding of the types of personnel involved in the different tasks (system architects, operations managers, department heads, etc.) and the cost structures related to each of them in the service provider market.
- Apply an amount of time required for each person to accomplish the task within each of the required workflows.

- Sum the resulting costs to each of the infrastructures we are analyzing, based on the attributes of the systems included in the designs.
- Examine the frequency with which the tasks involved are required in a specific day, month, year, and in total across the period, allowing us to construct a complete and integrated view of the lifecycle personnel costs of the POD.

The life cycle model we employ is inherently multi-disciplinary. This is critical because of the depth and breadth of the architectural shift represented by NFV for its adopters. **NFV involves an integration of IT centric, network-centric; software-centric, hardware-centric; application-centric; business-centric and market-centric skills into a new constellation of skills that make its deployment possible**. We know the operator has adopted such a multi-disciplinary approach to staffing and enabling its NFV deployments. And we also know it is a model being pursued broadly among global operators working on NFV deployments.

Agility in New Service Creation

Although optimizing TCO is critical to harnessing key benefits from NFV, it is also crucial to consider how efficiently and creatively NFV solutions help operators develop and deliver new service offerings to respond to revenue opportunities. Although it takes a range of platforms and skills to accomplish these goals, including for example, customer and market analytics applications and software development platforms used by multiple groups in an operator's organization that are beyond the scope of this report, the NFVI solution the operator deploys also has a meaningful impact on how efficiently new services can be developed and deployed and how quickly they can be modified (or even taken down) when responding to customers' demands.

We considered how each platform's functionality contributes to achieving these goals, and the amount of work each person involved in our multi-disciplinary staffing profile contributes to developing and deploying a new service offering. We focused on **greatest efficiency in both dollars and elapsed calendar time in bringing new services to market and gaining access to the new revenue** generated by the service as the most significant measure by which the agility of the platform could be gauged.

Results of the Economic Comparison

The following sections convey the results of our analysis. They are reported at both a summary level (focused on the cumulative TCO, for example, of deploying each of the NFV PODs), as well as in the individual categories that make up each of the major dimensions of the analysis (elements of capex, opex, and agility in new service creation). They are presented in the order of cumulative TCO, followed by capex, opex, and agility in new service creation.

To gain some appreciation of the data, for each element in the analysis, we prepared a source data profile of the element that contains its details. The source data records for each element and POD are then aggregated to a summary description based on like comparison logic. For example, although there are multiple types of network switches (leafs and spines) in each design, and there are different types of storage arrays and controllers (densities and types of HDD/SSD drives), we make a comparison of similarities between the PODs by comparing the amount of network or storage cost incurred by each when meeting the operator's requirements.

As we describe each element of the results, we provide sufficient context as to the root causes of the result and the reasons they have the result to grasp the underpinnings of the analysis.

Cumulative TCO versus the Tightly Bundled POD

The five-year cumulative TCO of the open architecture POD in the operator's environment is $22.7 million, which is 52.7% (or roughly half) of the cumulative TCO of the tightly bundled POD, which is $43.0 million.

This is the aggregate of the capex and opex costs of each of the deployment configurations over five years of the analysis. As we will see in subsequent sections, both capex and opex make material contributions to this result. There are economies in design in significant elements of the solutions, as well as architectural differences that affect the degree of openness and flexibility available to the operator in achieving its objectives. There are also substantially greater levels of software integration applied to simplifying functionality at multiple levels of the open architecture operation of the POD that bring significant advantages to the teams running and enhancing it in support of the operator's services. **It is on virtually every measure a simpler and more efficient design.**

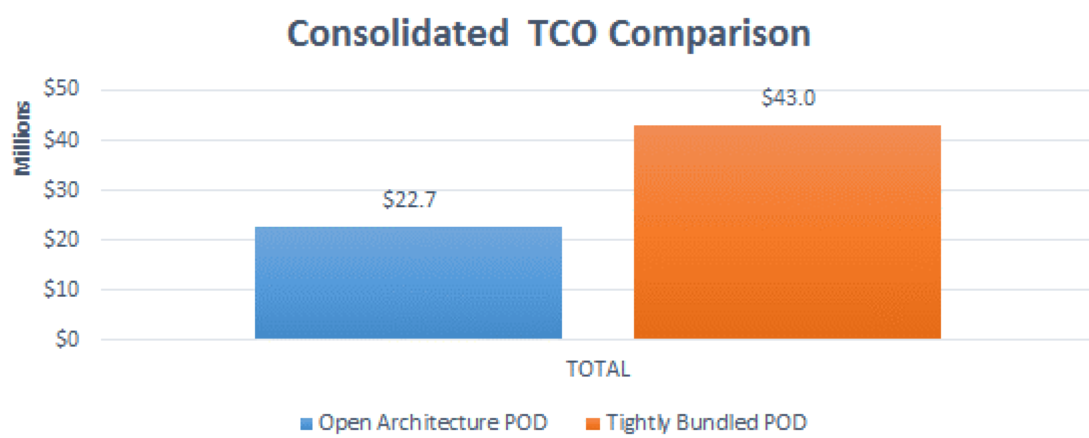

Figure 4. Cumulative TCO of the Open Architecture and the Tightly Bundled PODs

Capex Costs: Comparison and Analysis

The capital expense of the open architecture POD in this NFV deployment is $21.4 million; the capex of the tightly bundled POD for the same operating requirements is $40.7 million.

The composition of these totals is shown in the table and further illustrated in Figure 5.

	Category	Open Architecture POD	Tightly Bundled POD
CapEx	Network	$ 4,225,980	$ 12,979,009
	Storage	$ 3,333,104	$ 4,133,800
	Compute	$ 5,717,566	$ 11,833,095
	Security	$ 2,051,860	$ 4,650,600
	VIM Software	$ 6,080,000	$ 7,040,000
	Mgmt Software	$ -	$ 42,000
	TOTAL	$ 21,408,510	$ 40,678,504

Table 2. Capex Costs by Category of Configuration Element

The largest proportional differences in capex between the two designs are in the network, compute, and security areas. In each case **the reason is the elements in the open architecture POD employ an economy and simplicity of design not present in the tightly bundled POD solution**. One example is in the way the physical server and network infrastructures are integrated in each case. In the open architecture POD, an economy of implementation is achieved by integrating functions more fully and straightforwardly into embedded functions on either side of the server-network interface, in the Dell PowerEdge servers and the Big Switch fabric physical and virtual nodes. Substantially more flexibility in deployment options is built into the hardware bases of this configuration, and the software supporting them is significantly simpler and more efficient with respect to services and topologies it can support. In addition, in the tightly bundled design, an entire additional hardware element in each rack is built into its server-network interface, which consumes considerable additional space, power, and significantly, capex and opex dollars.

Similarly, in the security/firewall design of the PODs, the open architecture solution, based on F5's Viprion ADC, is more streamlined in its implementation of functions for the range of protections and the amount of bandwidth required in the operator's deployment. The result is a substantially lower cost of solution for security in the open architecture POD than in the TBP.

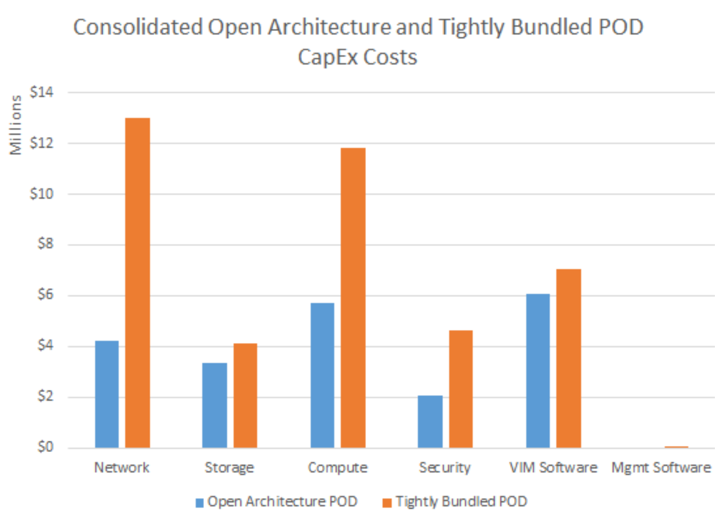

Figure 5. Comparison of Capex Costs by Element in the Open Architecture and the Tightly Bundled PODs

From the capex point of view, the operator can support an equivalent number of VNFs for approximately half the capital cost in the open architecture POD. Stated differently, the operator could deploy twice as much capacity and support twice as many customers in the OAP as in the TBP.

Opex Comparisons and Analysis

Although the capex costs of the two designs are markedly different, the opex costs incurred in each of the designs are also significantly different and contribute significantly to the results the operator could expect by employing either of them in its deployments.

The cumulative five-year opex of the open architecture POD in this site is $1,317,748, which is 56.7% of the cumulative five-year opex of the tightly bundled POD, at $2,320,711.

We analyzed opex costs keeping in mind that the operator in focus is beginning to adopt an agile operations model in its network engineering and operations teams, as well as in its application development groups interacting with the evolving service delivery infrastructures. This is in line with the continuous innovation, continuous delivery (CI|CD) paradigm of the DevOps model. The continuously evolving nature of this model is shown in the Figure 6.

Figure 6. Emerging DevOps Operating Model

At the same time the new solutions being created for deployment using NFV are being installed and integrated into a classic brownfield environment made up of large, nationwide, revenue generating networks the operator is already running. In many of those environments the NFV teams also need to accomplish their work using a more conventional flow based on a known cadence of releases, introducing functions only at specified calendar dates, constraining the pace of new feature deployments overall. This more stringently calendar-based model of deployments, often referred to as a waterfall approach, is illustrated in Figure 7.

Figure 7. Phased Calendar Release Model of Service Introductions

These two models coexist in the operator's environment, though they are trending incrementally toward the continuous innovation model, the NFV and cloud application infrastructures being at the forefront of that evolution.

We turn to our analysis of the opex costs involved in the deployment of the two alternative designs. The overall composition of the costs is shown in Table 3 and illustrated graphically in Figure 8. As the data show, opex costs are incurred across a continuum of tasks in the lifecycle of the deployments. They start with design and installation of the PODs, progress to occasional expansion of capacity as services grow, and to periodically upgrading the software of the multiple elements of the POD. In parallel with dimensioning these workflows, we examined the costs of powering and cooling the configurations over the period of the analysis. We analyzed the comparative costs of new service development and deployment, which are associated with efficiency in capturing new revenues. These are described in the report.

	Task	Open Architecture POD	Tightly Bundled POD
5 Year OpEx	POD Design	$ 199,053	$ 269,129
	Physical Installation	$ 42,547	$ 57,888
	Baseline Configuration	$ 3,116	$ 37,358
	VIM Installation	$ 97,443	$ 148,153
	Capacity Expansion	$ 62,491	$ 145,038
	Software Upgrade	$ 319,337	$ 534,432
	New Service Creation	$ 223,485	$ 668,182
	Power + Cooling	$ 370,276	$ 460,531
	TOTAL	**$ 1,317,748**	**$ 2,320,711**

Table 3. Five-Year Opex Costs by Category of Task

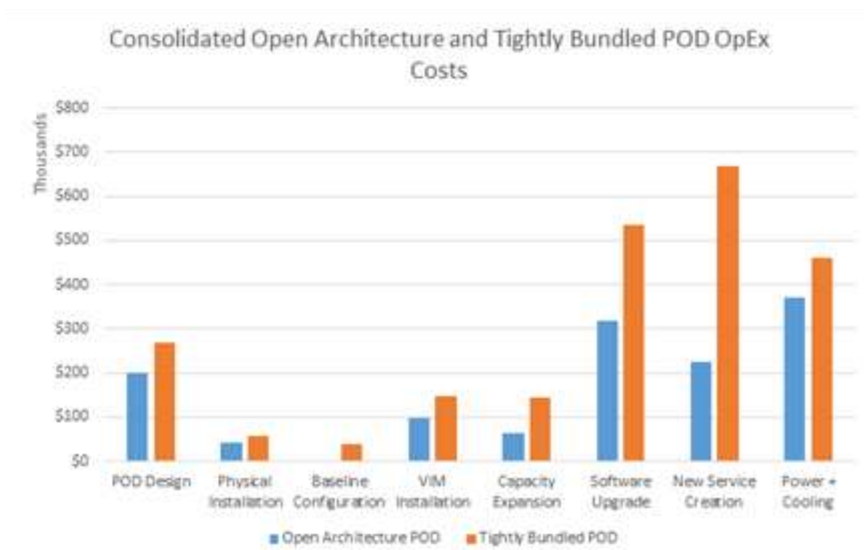

Figure 8. Five-Year Summary of Opex Costs by Category of Task in the Open Architecture and the Tightly Bundled PODs

We discuss key elements in the workflows in each deployment, highlighting the characteristics of the open architecture design that give it such a commanding advantage in operational efficiency and service creation agility compared with the tightly bundled solution. Certain elements of the workflow create a more dramatic difference than others. We concentrate on those, so we are clear about the basis of the advantages: designing the PODs; enabling their baseline configurations; installing a virtual infrastructure manager (VIM) solution in the PODs; upgrading the software of the PODs; and creating and deploying new services to be run in the PODs.

Designing the PODs

Because NFV involves such a new blend of skills in creating a deployable platform, determining what the NFVI will need to include to be successful involves combining insights across a range of disciplines, and blending them into a design that has the highest chances of success. We looked at the tasks from this perspective and analyzed the workload in each major step.

The design cycle included tasks for architects, engineers and managers in server, storage, network, security, cloud/NFV software infrastructure, VNF (such as mobile network and mobile application services), as well as developers and managers in revenue-generating application and OSS/BSS software. Contributions from these teams are blended into an approach that will work in both brownfield and emerging cloud-native environments.

Tasks we analyzed range from capacity and performance planning, to functional and integration test planning, to operating, upgrade and deployment evolution considerations. They span from research in specialists' own areas of responsibility, to review and agreement among teams on direction, to soliciting bids and evaluating offerings, to finally selecting the designs that would be used.

Because the operator's intent was to have its NFV platform serve as a blueprint for agility as it goes about enhancing its offerings, the emphasis on diligence and proof of value in the selection process was high.

The aggregate of personnel time to design the open architecture POD was 455 person-days across the specialties contributing to the result. The 455 person-days occurred over an elapsed calendar time of 4 months. In the context of major architectural transformation of this scale, this operator's teams were working at a very rapid pace. By comparison the tightly bundled POD required 625 person-days and an elapsed calendar time of 6 months (from specification through POCs to selections) to design. As shown in the data, the cost of design time was $199,053 for the OAP and $269,129 for the TBP.

> *Aggregate personnel time required to design the open architecture POD was 455 person-days ... equivalent tasks for the tightly bundled POD required 625 person-days.*

The primary sources of difference between them are the complexity of server-to-network interfaces throughout the POD in the TBP, the relative lack of integration between virtual and physical components of the NFVI in the TBP, and the relatively limited integration with the VIM software for the POD that is managing its workloads. Each of these makes the design more complicated to resolve, and makes planning operations in the TBP for its workloads more complex.

Baseline Configuration

Setting up configuration baselines for a POD (before the VIM and cloud management software are installed) is another area of meaningful efficiencies in the OAP versus the TBP. Although the aggregate amounts are smaller than some areas of opex in the analysis, they matter in the bigger picture in that the tasks recur in capacity expansion, as well as growth into new locations using the same underlying blueprints.

Creating baseline configurations for NFV PODs includes installing operating systems in many elements and control software in others. It also involves setting up underlying mechanisms needed to run the POD (node names, operator roles and permissions, network identities and functions, storage controller and drive relationships, and security alignment with the topologies and elements of the POD).

Baseline configuration is an area in which the degree of automation in the platforms and their architecture in terms of how they operate at scale has a direct bearing on the efficiency and speed with which each configuration can be created. The personnel time required for the team to set up the baseline configuration in the OAP was 55 hours over an elapsed time of 1 calendar week; the time required for setting up the TBP was 318 person-hours over an elapsed time of 3 calendar weeks. The OAP was 3 times faster and required slightly less than 20% of the personnel time to create its baseline configuration than the TBP.

> *Personnel time required to do baseline configuration for the OAP was 55 hours … personnel time required to do the same for the TBP was 318 hours … setting the baseline in the OAP was 80% faster than in the TBP.*

The primary sources of the difference in required time are in the simplicity of establishing the server baseline configurations, the pervasive automation employed in configuring the network fabric of the POD, and the relative simplicity of enabling the firewall/IPS platform in the OAP compared with the corresponding elements in the TBP.

Virtual Infrastructure Manager Installation

Although it is technically a part of the management and orchestration functionality in the NFV Reference Architecture (Figure 2) few elements of deployment affect the efficiency, agility and scalability of NFV infrastructures as pervasively as the VIM and the degree to which it is integrated with both southbound and northbound controls in a design. When the software integration is extensive and threads through each component of the POD and when these integrations directly control

> *Few elements affect the efficiency, agility and scalability of NFV infrastructures as pervasively as the virtual infrastructure manager.*

major operations and resource management levers that determine the efficiency and performance of the POD, the contribution of the VIM to achieving the operator's goals in deploying NFV is fundamental.

The VIM enables virtual computing, virtual storage, and virtual networking resources required for running VNFs. It also coordinates supporting functions that do important security, monitoring, and orchestration tasks in support of the operations of the POD. In this operator's case, the VIM for the OAP is Red Hat's OpenStack Platform (OSP) and the VIM in the TBP is transitioning to that same solution from a previously implemented choice (over time). One effect of this is that the extent of integration achieved among the elements of the OAP was substantially further along at the time of initial NFV deployments. The integration of surrounding processes with the VIM was more advanced.

To see more vividly the pervasive role of the VIM in the NFV deployment, consider the number of functions it is touching and controlling in an OpenStack deployment based on OSP, shown in Figures 9 and 10.[3]

In Figure 9 we see the relationship of the management dashboard at the top of the diagram and the platform's orchestration service, which drive functions controlling each portion of the POD (compute, networking, storage, telemetry, statistics, authentication, identity control, and others).

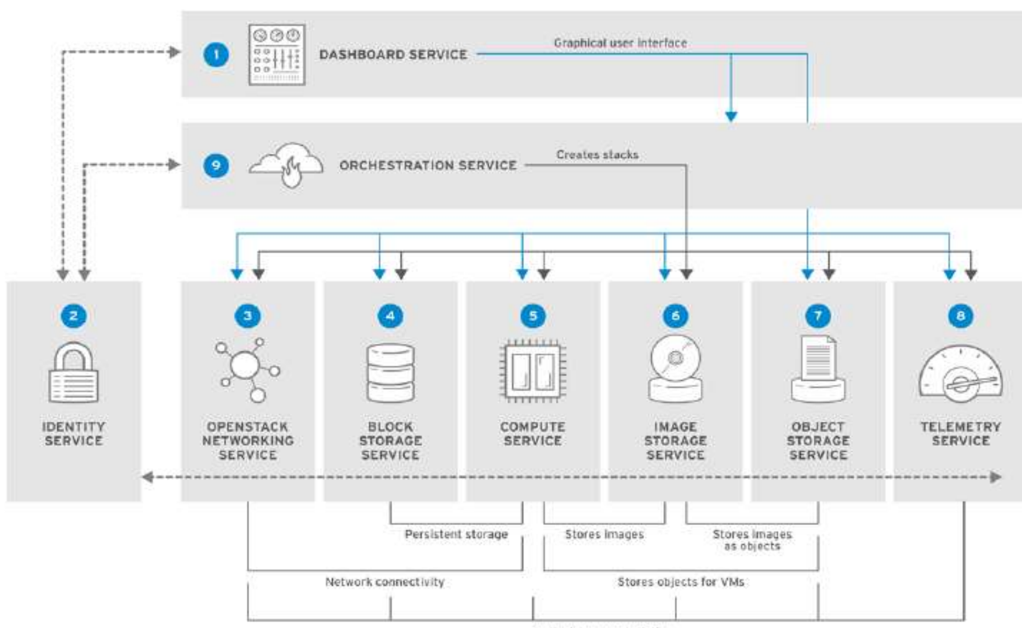

Figure 9. Representative Scope of an OpenStack Platform Deployment

Figure 10 clicks down one level to illustrate the operation of the VIM with the elements that are delivering the functions required by the VNFs of the POD and the applications. For example, we see a controller node (a server running OpenStack services that manage the other elements shown)

[3] Diagrams are from *Red Hat OpenStack Platform 10 Architecture Reference Guide* and *Red Hat OpenStack Platform 10 Director Installation and Usage*, Red Hat, Inc., 2016.

connected via multiple networks to other elements in the POD. These networks are virtual networks designated for the purposes described. For example, the green provisioning network connects all the nodes with which the controller is working to manage their basic attributes in the OpenStack configuration. The blue tenant network connects the controller to a compute node running workloads required by a user of the POD, such as the operator's mobile services business that needs to run one of its VNFs (a packet gateway node or a subscriber policy control module). We see an orange storage management network that manages resources for workloads running in compute nodes. We can appreciate the nuances and pervasiveness of the networking functions in the POD by noting that every line in the diagram represents either a physical network element (a network interface card or a physical fabric switch) or an overlay network virtual node, such as a virtual switch running in any of the compute or control nodes to connect the processes with each other.

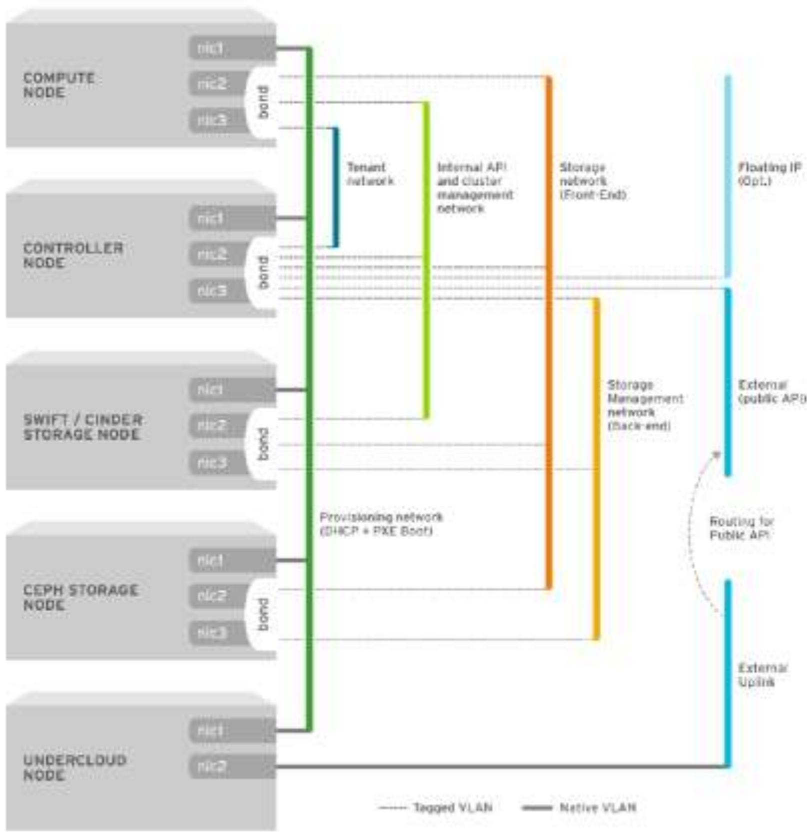

Figure 10. Representative Design of an OpenStack Platform Deployment

With this degree of interconnection and the relationships indicated by every element in the POD (which has 12 racks, over 200 servers, over 30 switches and over 600 individual storage drives) the degree of integration between the VIM and the nodes it is supporting has a major effect on the efficiency and agility of the deployment overall. Installing the VIM so it **works automatically** with the resources on which it is expecting to rely becomes a determinant of the opex costs the operator will incur and the life cycle required for getting the PODs to do what they will do.

VIM installation happens after baseline configurations. The team for VIM installation includes server, storage, network, and security architects and engineers. Tasks include installing control software in controllers and installing OpenStack modules in **all** nodes of the POD (they are all taking part in the OpenStack operation). In addition, integrating the underlying network and security platforms with the VIM needs to be done by activating and validating plug-ins from each of those elements with the relevant services in the VIM. Finally, installation involves not just the first-level steps but also testing and validating to verify the services are functioning and are ready to support loading and running production workloads from stakeholder users and businesses.

For the open architecture POD, we analyzed the total personnel time required to prep, install and validate the VIM as 78 person-days over an elapsed calendar time of 3 weeks. The corresponding time for the tightly bundled POD was 136 person-days over an elapsed calendar time of 1 month. VIM installation in the OAP required 57% of the personnel time as VIM installation in the TBP and was accomplished in 25% less calendar time.

The personnel time required to prep, install and validate the VIM in the OAP was 57% of the time required for the same tasks in the TBP.

The primary reasons for greater efficiency in the case of OAP are the extensive integration of the Big Switch fabric with the Red Hat OpenStack platform installer; a similarly broad integration of Red Hat's Ceph virtual storage software with the VIM; and the integration of F5's FW/IPS platform with Red Hat's OSP.

Upgrading the Software in the POD

After the NFVI is fully up and running over time there are various aspects of operation that contribute to its efficiency and flexibility. One of these is how easy or hard it is to upgrade software. When we reflect on the pervasive integration of the VIM with the other resources in the NFVI and we also note the extensive interaction of the VIM with the VNFs and the OSS/BSS software running northbound, it is evident how significant a software upgrade to the VIM can be.

Upgrading software in the NFVI and the VIM can be a significant gating factor to progress in realizing the goals in an NFV deployment.

In the industry, more broadly, the early days of deploying NFV showed that upgrading from one version of software to the next in server OS, in network fabric, in storage, in VNFs and in orchestration can be a significant stumbling block and gating factor in making progress toward using valuable new functionality. It can be a major consumer of time. Many efforts in vendors' and open source developments have focused on eliminating this stumbling block. One well-known open source initiative that has worked on this is OPNFV, focused on delivering an open source NFVI and VIM solution. The first three releases of the OPNFV distributions (Arno, Brahmaputra and Colorado) each included a strong focus on hardening installation and deployment functions for NFVIs and VIMs. Completing these releases took almost two calendar years with over 150 engineers and developers collaborating. Knowing about these complexities and working toward overcoming them are a major reason why the suppliers of the open architecture POD collaborated to such a great extent on a pervasive integration of their installation and operational functions to simplify installing and running the NFVI.

To analyze the costs of upgrading the software for the POD designs, we focused on the case where the driver for the upgrade is installing a new version of the VIM. This is normally triggered by adoption of a new OpenStack release such as Kilo, Mitaka or Newton integrated into a new version of Red Hat's OpenStack Platform (such as RHOSP Version 8, 9 or 10). The reason for focusing here is over time upgrading the VIM with new features typically requires an upgrade to the software of the other elements in the deployment that must also support the new functionality. Although it is **not true that every** element **always** requires a new software version in parallel with the new version of the VIM, it is more often the case that they do than that they do not. Thus, in addition to upgrading the software for the VIM, we included the work of installing new releases of software for the server, storage, network, and security elements in the PODs.

The team working on these tasks includes architects and engineers from each of the areas of functionality whose solutions are running in the PODs. Work includes analyzing changes coming in the upgrades, planning the deployment into the target configurations, testing the versions to validate their readiness, defining the steps for accomplishing the upgrades, and validating their operation once they are installed in the POD.

Personnel time required for upgrading to a new release in the open architecture POD was 92 person-days over 1 a month of elapsed time. Time for the equivalent steps in the tightly bundled POD was 229 person-days over 1.5 months of elapsed time. Staff time required to plan and execute the upgrade in the open POD design was 40% of staff time required in the tightly bundled alternative, and elapsed time was a third less in the OAP case. If we assume at least one software upgrade per year is likely across the next five years of operation, then for just the core site PODs we are analyzing the operator would save 137 person-days each year among the teams contributing to evolving this infrastructure on this task alone.

> *Personnel time required to upgrade to a new release of software in the OAP was 40% of the time required in the TBP.*

Power + Cooling

The cost of powering and cooling network and IT infrastructures is an important component of opex budgets. We compared costs of powering and cooling the two POD configurations using a consistent average cost per kilowatt hour of electricity for the operator's sites. We used the published and measured ratings of power consumption for the elements running in the PODs as published by their suppliers. We applied the same cost of electricity for providing power to the equipment initially, as well as for powering the cooling systems to dissipate the generated heat. We assumed the infrastructures operated at peak load for 16 hours each day, 365 days each year to simplify calculations. With this as the baseline, we determined the costs of powering and cooling the open architecture POD configuration to be $370,276 over five years, approximately 20% less than the cost of powering and cooling the tightly bundled POD, which was $460,531, as shown in Table 4.

	Open Architecture POD	Tightly Bundled POD
	5 Year Cost	5 Year Cost
Power + Cooling	$ 370,276	$ 460,531

Table 4. Five-Year Costs of Power and Cooling for OAP and TBP Configurations

Relative Agility of the Two Designs: Time to Create New Service Offerings

Having looked at several of the opex costs that are purely focused on making the infrastructures work, we turn to workflows directly related to creating new customer-facing services that will generate additional revenues. Since a primary motivation for making the transformation to virtual system infrastructures is to help providers become more agile and broadly appealing to their customers, this aspect of NFV deployments is especially important.

Given our focus on NFVI in this project, we concentrated on relative efficiency of the NFVI in the PODs in contributing to these goals. Both PODs were being implemented in support of the operator's higher level service templates and their application using OpenStack HEAT and Ansible playbooks to describe the operation of new service offerings. The main difference between them on this dimension was how much further along the OAP was in its application of OpenStack functions across all of its components.

The more profound differences in new service creation are in the degree to which the infrastructure of the two PODs are integrated with the VIM. In the open architecture POD, the scope of integration between the elements in its NFVI and the OpenStack functions in its VIM is significantly more extensive than in the TBP. Three areas of operation put a spotlight on the point. First, with respect to its network fabric, the unification of Big Switch Networks' virtual and physical network nodes into a single logical fabric, managed as a unified entity from a consistently implemented control point, dramatically simplifies deployment of new service functions compared to the relatively more segregated virtual and physical design of the network infrastructure in the tightly bundled POD. In addition to its operation as a single logical fabric, the BCF is also comprehensively integrated with the functionality of the Red Hat OpenStack Platform, which increases significantly the number of functions that can be done automatically by leveraging the respective APIs of the platforms. These functions span from installer operations linking new application functions with supporting network operations, to dynamic network processing using Neutron plug-in operations, to

In the open architecture POD the scope of integration with the OpenStack functions in the VIM is more extensive than in the tightly bundled design.

diagnostic and measurement integrations with the OpenStack console to automatically validate the operation of network connections in the POD.

In parallel, configuration and management of security in the OAP using F5's Viprion application delivery controller is accelerated via similar plug-in integration with the Red Hat VIM. Plug-ins adapt template declarations to POD FW and IPS protections to support both the NFVI and tenant workloads as VNFs are deployed into new production services. Finally, the Ceph-based software-defined storage of the OAP is extensively integrated with the Red Hat VIM, allowing storage resources and operations to be provisioned, verified, and dynamically managed with the performance, scaling, and functional attributes specified in the orchestration templates processed by the VIM.

Although these types of integration exist to some degree in various elements of the tightly bundled POD, it is the depth and breadth of integration across elements in the OAP that makes it more responsive and efficient in supporting new service creation than the TBP.

We analyzed the full cycle of tasks for bringing new services to deployment in the PODs, from concept to justification to experiment, development, testing, validation and deployment. In a model trending toward devops, it is fair to ask if the infrastructure is all programmable, and new services use well-defined templates and APIs, where is the incremental opex incurred? The answer is in the addage, trust but verify. With the scale of the operation in Tier 1 service providers, such as the operator in focus in this study, and the stakes of the deployments taking place, verifying successful operation of a newly offered service still needs to occur even though a more streamlined process is being used, and the offer itself may be taken down if it turns out to be unappealing or unworkable in some important way. When we analyzed the new service creation workflow, we considered evaluation of compatibility by developers and engineers at the architectural level for a new service ofering; validation of functionality and performance in stremalined testing; verifying securitry protections for the new service offering; development of OSS support; validation of VNFs in the new service configuration; along with validation in operations that the new functionality is operating successfully.

In a model trending toward DevOps it's fair to ask if the infrasructure is programmable, where is there incremental opex in creating new service offerings? The answer is in the addage, trust but verify.

The operator can deploy three times as many new revenue services per year in the OAP compared to the TBP.

We used a service creation cycle occurring on a twice-yearly cadence, and analyzed the personnel time required to deliver each new offering. In the open architecture POD, we determined the average service creation time per year using this pattern to be 100 person-days at an average annual cost of $44,697. The comparable average service creation time in the tightly bundled environment is 290 person-days per year at an average annual personnel cost of $133,636. On this cadence of new service deliveries, it is possible to deliver new services in the OAP with 35% of staff time at 33% of the cost. Stated differently, the operator could deploy three times as many new revenue services per year in the OAP compared to the TBP.

Although these amounts vary based on service and application and while they **do not include** other costs such as the original developer's time to code or costs in other parts of the organization to bring the new service to market (such as marketing and customer support), they **do show** clearly how a more agile service delivery platform can contribute to the operator's efficiency in bringing new services to market and gaining access to their revenues.

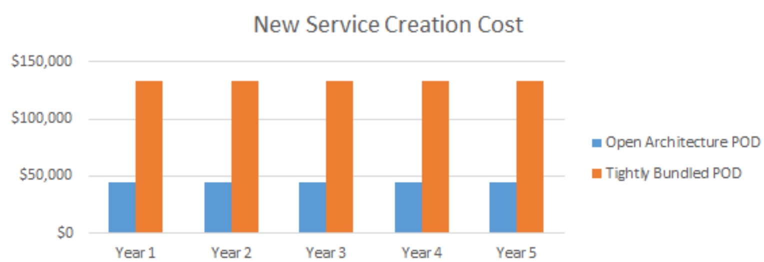

Figure 11. Costs of New Service Creation in the Open Architecture and the Tightly Bundled PODs

CONCLUSION

The imperative for communication service providers to embrace more agile service delivery platforms that help them become more continuously innovative is compelling and widespread. Based on extensive investigations of virtualized and software-driven service delivery architectures over the past several years, the feasibility of deploying more cloud-native designs is now exiting the proof-of-concept phase and entering early stages of more general-purpose production deployments.

When making this transition, operators have alternatives they can consider in the infrastructure platforms on which they rely for supporting the virtual service deployments. There are meaningful and material economic differences between the alternatives based on degrees of openness, simplicity of design, and degrees of automation built into the solutions.

In the case of the Tier 1 service provider's deployment analyzed in this study, the advantages of the open architecture platform design provide a compelling justification for using that approach as the foundation for the operator's transformation efforts. New services can be introduced in one-third the time as with the alternative the operator was considering. And the cumulative total cost of operation of the open architecture approach is roughly half or 53% of the more tightly bundled option. Each of these outcomes makes a material contribution in each POD that is deployed and at scale toward enabling the operator to become a more continuously innovative and valuable supplier of services to its customers.