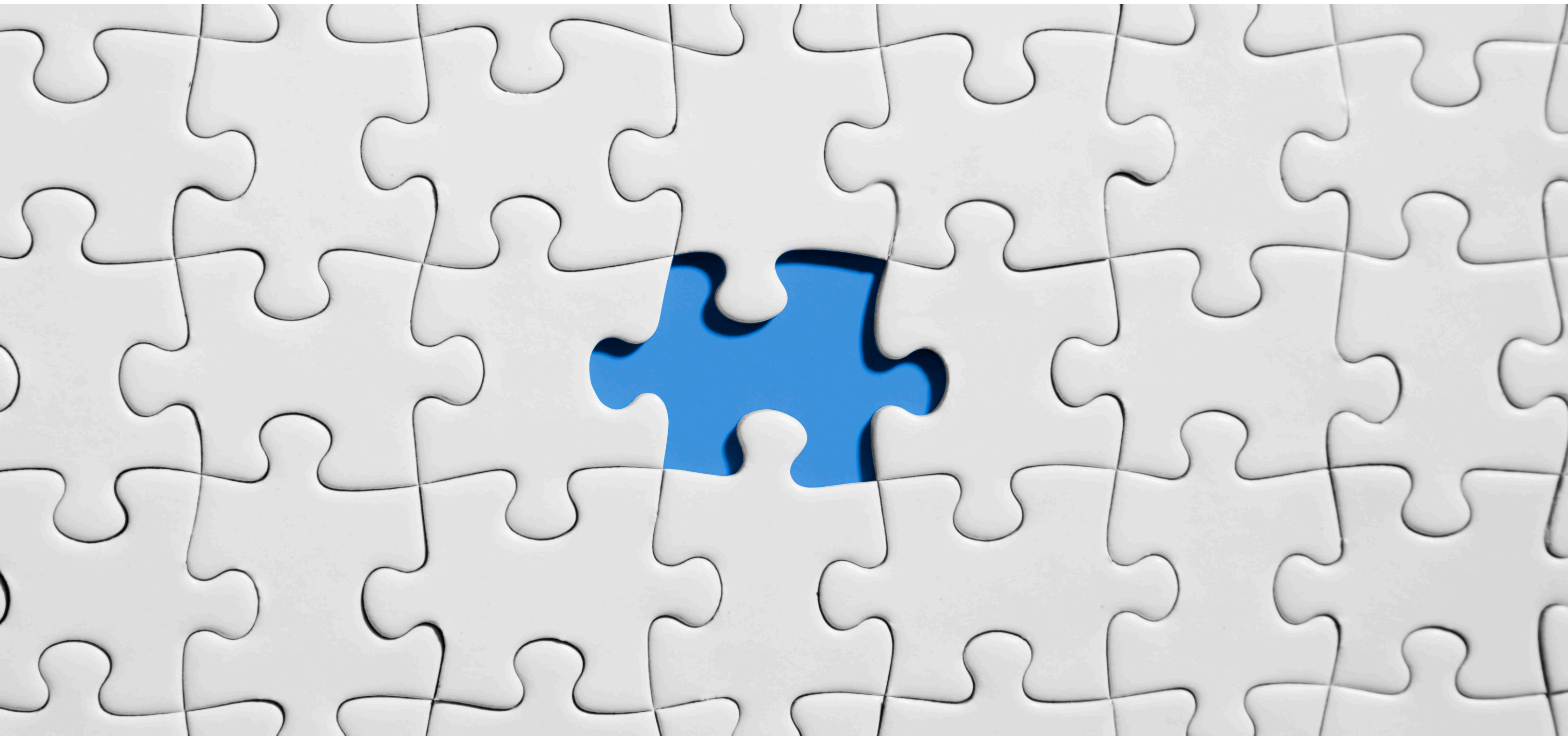
SERVERLESS BY DESIGN: THE HYBRID CLOUD



Anthony Dutra

Systems Engineer

Dell Technologies

Anthony.dutra@dell.com

Varshith Anilkumar

Senior Developer

Wayfair

Varshith@bu.edu

William Lupo

Senior Developer

Thomson Reuters

Williamdelmontlupo@gmail.com

Knowledge Sharing Article

Table of Contents

Abstract ... 3
Let's Talk Digital Transformation ... 4
  Redefining the Data Center with Software ... 5
Evolving into the Cloud ... 6
  Key Components of Cloud ... 6
  Not Everything is Cloud-Ready ... 8
It's All About the Applications ... 9
  Cloud Extensibility ... 10
    Long-Term Retention in the Cloud with Dell EMC Unity ... 10
    Data Protection with VMware on AWS ... 11
A Clean Slate: Greenfield Deployments ... 12
  All-in-One with Converged Infrastructures (CI) ... 12
    CI with Dell EMC ... 12
  Foundation for Hybrid-Cloud Agility-Hyperconverged Infrastructures (HCI) ... 13
    The Foundation of the Hybrid Cloud- VxRail ... 13
    HCI to Hybrid Cloud with Dell EMC ... 14
  Integration with Multi-Cloud Services ... 15
    Orchestration with APIs- Dell Boomi ... 15
    Replication Across Clouds- Dell EMC Cloud Snapshot Manager (CSM) ... 16
New Age of Enterprise Hardware ... 17
  Faster for Flash-NVMe & SCM ... 17
  Software is becoming the Bottleneck ... 17
  "Cloud-Like": Composable Infrastructures ... 18
  What Does This All Mean? ... 19
May We Suggest "Serverless" ... 19
  What is Serverless Technology? ... 19
  Limitations of Serverless ... 20
Our Vision: A Serverless Hybrid Cloud Methodology ... 21
  A Decoupled Design ... 21
  The Publish-Subscribe Pattern ... 22
Future Technologies & Serverless ... 26
Appendix ... 27

Abstract

Gone are the days of the server…sort of. Virtualization of the data center has single-handedly revolutionized the way we utilize and perceive IT infrastructure. In breaking the constraints of physical hardware, resource abstraction evolved into an idea that transformed the IT ideology of cloud computing.

Virtualization created the cloud and by association, also created what we believe to be the next evolution of the data center. Functions as a Service (FaaS) or known by most as "serverless" is streamlining the way we look at events in both our data centers and our applications. As everything is based on HTTP, serverless can integrate with any service or application. Where this becomes powerful is simplifying operations and security automation through simpler policy enforcement. IT organizations can expand serverless compute scenarios to take advantage of the resource savings by only launching programmatic actions.

This technology may provide an enormous benefit for many businesses, whose first step in their journey to the cloud either hasn't been taken or may have initially tripped over its complexities. At the outset, most look to the public cloud to begin save money within their IT infrastructure. They want to remove the burdens associated with management and orchestration of on premises hardware. What they soon recognize is that the lift and shift associated with re-architecting applications to mold to the form of the cloud quickly becomes costly and cumbersome. Not to mention the development expenses, such as education, required to do so. This is where serverless could save the day. If properly packaged into an orchestration and management platform, a serverless-based program could provide an organization a true path to the cloud.

This is exactly what we look to uncover in this article by first explaining the challenges of the journey to the cloud. Next, we explore how technology is evolving in both the hardware and software spaces – further complicating the IT infrastructure domain. We touch on non-volatile memory express (NVMe), Storage Class Memory (SCM) drives, and composable architectures for hardware. In software, we provide an overview of containers, microservices, and infrastructure as code. We seek to prove that together, these technologies are further driving the need for hybrid cloud infrastructures to keep a business cost-effective and competitive. We share our knowledge on how cloud technology is going through changes as well as challenges developed from the clash between monolithic and cloud-native applications.

Then we dig into serverless technology. We provide an overview of the technology, examine the available serverless public cloud technologies, and describe the skills required to take advantage of it. We then turn our attention to proposing how serverless is poised to enable the true hybrid cloud. We describe where and how serverless can be used in existing infrastructures to enable hybrid cloud-like functionalities.

Next, we showcase with a couple examples how we are driving the serverless evolution with public cloud offerings and Dell technologies. These basic starting points provide a glimpse into how organizations can digitally transform their current IT infrastructure today into a hybrid cloud solution.

Finally, we touch on the future of how technologies such as artificial intelligence (AI) and Blockchain can be empowered further by serverless.

If virtualization created value through the abstraction of resources, then serverless is crafting value through decentralization. It says, let the events of the application dictate what my business needs for IT infrastructure. This article hopes to capture the importance of this technology to the critical future of our IT infrastructure.

Let's Talk Digital Transformation

In many ways, technology has always been an effective medium for us to tell our story. Through the modernization of video streaming, at our fingertips we have access to thousands of stories – something as old as civilization itself. The art of storytelling has evolved with our relationship to technology. It started millennia ago by listening to the village elder retell his crazy wise tales.

Then we leapt forward to the invention of paper in ancient China. This meant people no longer had to wait for a story, they could write one down themselves and share it. Those once storytellers became scribes and those scribes became writers, and journalists, and musicians. As technology propelled us further we became radio DJs, and television personalities, and YouTube stars. We create, upload, download, archive, and stream our stories where and whenever we prefer.

That's because technology is in every aspect of our lives. From the time we wake up to the time we go to bed, we are constantly connected through our mobile devices. While our Facebook pages may be filled with awkward memes from our more-estranged relatives, it's the technology and the massive amounts of data being processed and analyzed to help us better understand ourselves as well as each other, while service technologies available through web portals provide a platform for us to create almost anything as a service. Consider how easy it is to get around now with ride-sharing services such as Lyft or Uber.

Applications are at the center of this digitalization of a business. The information they generate lives inside of some data center somewhere on this planet. processing the bits and bytes that represent our day-to-day interactions with these applications. As one can imagine, this demand for applications is not slowing down. In a 2018 report written by Artyom Dogtiev from the blog site *Business Of Apps*, it is projected by 2021 the number of mobile applications downloads will reach a stunning 352 billion(1) across all major app stores, a significant jump from the 197 billion(1) downloaded four years earlier, a staggering increase in demand. Note too that this statistic only represents mobile applications. When web-based or proprietary applications used to manage the daily operations of a business are taken into consideration, the count becomes innumerable.

It isn't just the volume associated with the growth in applications that is a cause for concern, it is also their complexity. As users or business demands in application functionality evolve, it is the job of the Information Technology (IT) team to respond to that demand. This reactionary form of IT is an outdated mindset derived prior to a technology that was a quintessential breakthrough in initializing the evolution of the data center.

How Virtualization Changed EverythingIn the not so distant past, hosting even the most basic applications was operationally arduous and financially taxing for many IT organizations. A new application release would need to be stable enough in design and functionality to justify the purchase of physical server hardware to host the application. The server would then need to be racked into a chassis on the data center floor, stacked with the necessary software, and integrated with the rest of the data center. This whole process is inefficient from start to finish, slowing the time to market of a new application. This process was revolutionized when virtualization swept the IT industry, cleaning up these inefficient processes through software.

Virtualization is the abstraction of an operating environment from the physical hardware to be used by other services. In server virtualization, for example, the technology that provides the abstracted or logical resources is called a hypervisor, the base component of a virtualized compute environment. This software layer is installed on the physical server to abstract each of its components into a pool of resources. This abstracted pool of resources such as CPU, memory, network interface cards, storage interface cards, and disks are shared amongst virtual machines (VM). This resource abstraction utilizes only a share of the actual physical resources, which allows multiple virtual machines to share the resources of a single computer system. Once a VM is created, consumers can install applications on the operating system. Every piece of the data center today, from server, to networking, to storage array, can be virtualized. It is common in data centers across many enterprises, commercial businesses, and government entities.

Another way to virtualize IT resources is through containers. Containers are logical compute constructs that still reside on a physical server with a base operating system but there is no hypervisor and therefore no hardware virtualization. Containers share the same physical hardware and base operating system. To better manage these abstracted resources, a management and virtualization layer is added to the host OS which is responsible for deployment, resource provisioning, and logical separation of the containers. Applications run in their own confined area just like in a virtual machine, but they share the same kernel with the host and there is no physical device abstraction. Containers can scale faster than virtual machines and consume less memory, CPU, and storage as they do not incur a resource overhead common in virtual machines. In a clustered format, mainly through programs such as Kubernetes, multiple isolated systems are run on a single control host and access a single kernel – the central component for managing resources between the hardware and the software. The containers hold the components such as files, environment variables, and libraries necessary to run the desired software and run only the required resources. In short, a container can be associated to being “pieces of an application in a box”. These components include the runtime components – such as files, environment variables and libraries – necessary to run the desired software consuming fewer resources.

What makes virtualization so powerful is that it is abstraction that drives its efficiency. Its benefit is beyond the dollars and cents of being able to effectively utilize a compute resource. Virtualization enabled something unique – the ability to consolidate IT resources while at the same time making them completely malleable. Virtualization gave our data centers the ability to do more with less, allowing us to abstract not only what we managed, but how we managed our IT infrastructure in relation to our business model.

Redefining the Data Center with Software

Deep integration with virtualization technology allows a business’s IT organization to then take the next step in digitally transforming their data center into a software-defined, service-driven architecture. The adoption of this phase not only further integrates the data center into the unique value proposition of the business. Technology becomes a proactive force for the business, instead of a reactive one. This can be an expensive undertaking for many businesses

Definitions of the term “software defined data center” (SDDC) may vary across these domains and vendors. Generally, the term can be described as a method of delivering consumable IT resources using software, standards, policies and protocols that enable enterprises to deliver services across multiple hardware platforms in a scalable and automated method. This is the next step from virtualization as software defined environments enable programmatic provisioning, control, and configuration of these abstracted resources which are then presented to services as virtualized resources. The purpose of software-defining the resources inside the data center allows the administrators to employ orchestration tools that can facilitate provisioning of IT resources to the business in an on-demand fashion.

In gaining this capability, the data center digitally transforms from the once rigid and expensive system to one that allows business to consume IT in a manner that we are all familiar with. The ability to purchase, deploy, and then de-provision a "digital segment" of a data center all through an easy to use portal makes application development as easy as ever. In moving to a software-defined approach, each business unit can now leverage this "catalog-like" capability inside of the on-premises data center. The goal for organizations at this point is to fully develop services around the abstraction of the data center.

Evolving into the Cloud

The term cloud computing has various meanings to different organizations. Many experts view the cloud computing definition by the National Institute of Standards and Technology (NIST), a US Government standards authority, as a fair working definition. According to NIST, cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources, (e.g. servers, storage, networks, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction(2). In short, cloud computing refers to the provisioning of on-demand computational resources (i.e. a collection of servers, client computers, storage units, and applications) through a virtualized network.

Key Components of Cloud

Evolution of the software-defined data center into a cloud-driven architecture is done through a business-layer abstraction of the IT infrastructure, providing end-users the ability to allocate and de-provision resources as the business demands. This is facilitated through development of a service catalog which is backed by orchestration and automation software. This orchestration layer defines rules which form workflows, provide communication of components through Application Programmable Interfaces (APIs) to facilitate actionable requests automatically for end users. As such, there is usually a requirement for a centralized platform in which the deployment of these tools can be initiated by various components within the IT stack.

This platform is referred to as the cloud management platform (CMP), a combination of infrastructure applications – regularly a collection of virtual machines managed through a centralized interface of a virtualized server. The foundation of a CMP generally contains a portal for administrators, authentication mechanisms (such as DNS services) for users, the service catalog, automation and orchestration engines, and metering capabilities for business chargeback. These components are made up of many different applications including the integration of software-defined controllers, load balancer appliances, element managers, and other cloud platform technologies. The ability for each of the components to communicate with one another is critical. This is done through the applications via an intermediary service such as a message queue. Information from the various components are stored temporarily in a queuing service for others to process. In addition, it is also important that the components of a CMP store information such as application state, configuration, user data, permissions, and roles in virtual database servers.

In one form or another, these components are integrated into cloud services which can be deployed in four separate models. The following is a glimpse into each cloud deployment as defined by NIST(2):

- *Public cloud:* The cloud infrastructure is provisioned for open use by the public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the Cloud provider. Examples of Public cloud offerings are Amazon Web Services (AWS), Microsoft Azure, Alibaba Cloud, and Google Cloud Platform (GCP).

- *Private cloud:* The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (i.e. business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

- *Community cloud*: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (i.e. mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

- *Hybrid cloud*: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability.

These four cloud models can be consumed in three basic types of services models. These models are based on the level of shared responsibility between the cloud service consumer and the cloud service provider.

1) **Infrastructure as a Service (IaaS)** The end user is purchasing the ability to provision processing, storage, networks, and other essential computing resources to deploy their software, operating systems (OS), some networking capabilities, and applications of which they have full control.
    a. *Provider Responsibility*: Securing the data center and maintaining the hardware
    b. *Customer Responsibility*: Settings, installation, patching, management of the resources they provision

2) **Platform as a Service (PaaS)** The end user is obtaining the space to deploy their created or procured applications utilizing the programming languages, libraries, services, and tools that are supported by the cloud provider only. The end user has no management insight or control to the underlying cloud infrastructure yet has total control over the deployed applications.
    a. *Provider Responsibility*: Securing the data center and maintaining the hardware. Also responsible for the settings, installation, patching, and maintenance of the platform.
    b. *Customer Responsibility*: Installing, patching, and maintaining the applications

3) **Software as a Service (SaaS)** The end user has no line of sight into any of the cloud infrastructure (including network, servers, operating systems, storage) and has limited access to the application capabilities itself. In this service model, the end user is given the right to run the cloud provider's application(s) on the cloud infrastructure. The application(s) are easily accessible from multiple end user devices.’
    a. *Provider Responsibility*: Performing everything from securing the data center and maintaining the hardware to installing, patching, and maintaining the application.
    b. *Customer Responsibility*: Utilizing the application

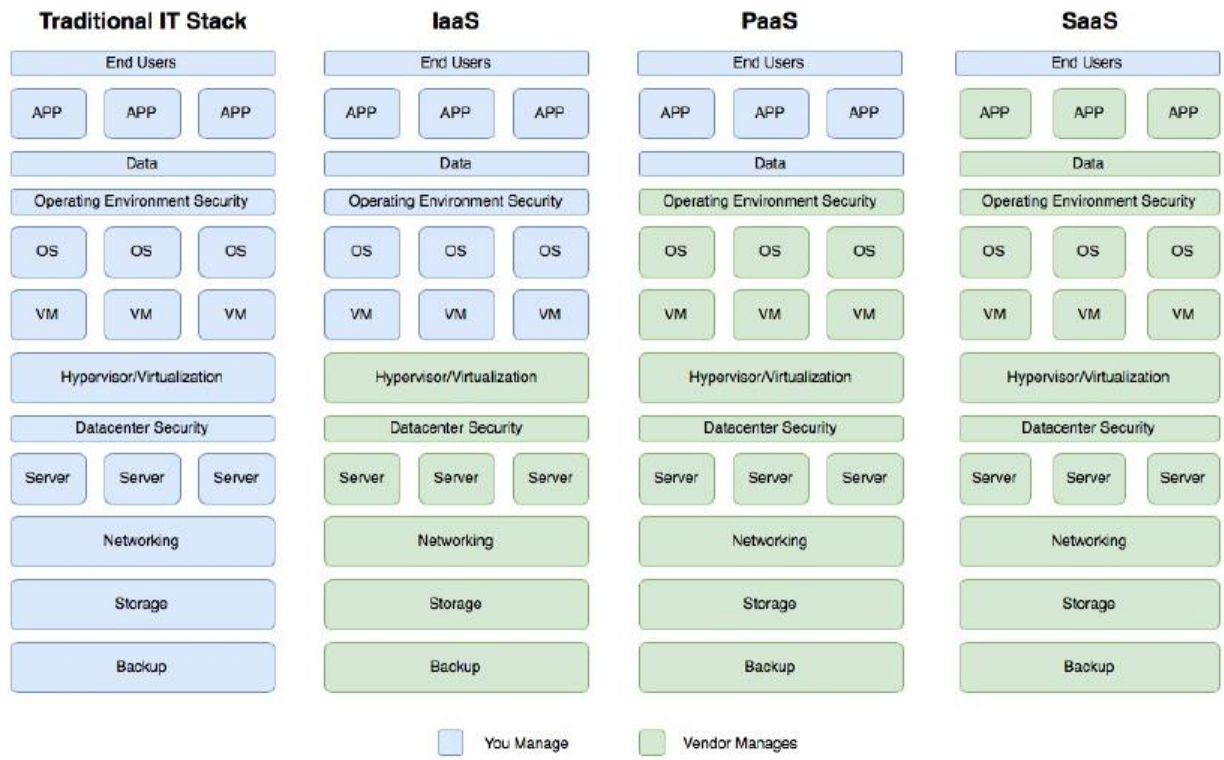


Figure credit: Terence Fowler-Dell EMC (2018)

Not Everything is Cloud-Ready

It is important to remember that the cloud is not a "one-size fits all" model. Instead, the design of the application in relation to its business functionality is what dictates whether an application can be set up in a cloud infrastructure. If there are any roadblocks, stakeholders must decide whether they need to redesign the application or the infrastructure it is running on. Applications plagued by specific hardware, operating system, or driver dependencies cannot operate in a cloud infrastructure. Many applications also require connectivity to another application that is run in an environment unavailable to the cloud infrastructure. In this case, migrating this application to a cloud may necessitate bulky design additions. Finally, consider the type of applications that just don't make sense to move. If an organization uses an email system such as Microsoft Exchange, it would not be cost effective to migrate Exchange servers into a private cloud since the users will never request these instances from a service catalog. However, it may make sense to use a public cloud provider to house the Exchange servers if the organization wishes to maintain a minimal footprint in their data center.

In this case, the choice for an organization becomes a consumer of instances or email-as-a service from a public cloud catalog.

Understanding how to incorporate multiple cloud service models into a future cloud design of both on-premises and cloud resources is important as plans develop to support them. For example, an organization may state that it wishes to implement a PaaS solution that allows the organization to develop and deliver web applications. Would this be best served in-house or through a public cloud provider? A public cloud PaaS service may provide the flexibility required to develop and test the application, but if the deployed application is resource intensive or latency sensitive, the organization may not have the ability to tweak infrastructure components to maintain performance. Another approach may be to use a public cloud IaaS service and layer their on-premises or private cloud PaaS infrastructure on top so that the organization has more control of the entire stack.

The goal for an IT organization who has been able to develop their data center framework up to this point should be to create an IT as a Service (ITaaS) model. This model is a business-centric, transformational approach to providing IT services to an organization. It focuses on business outcomes such as operational effectiveness, affordability, and rapid response which may improve costs for a potentially quicker time-to-market goals. ITaaS is the final transformation of the role of IT from a cost center to an agent of strategic business value. ITaaS defines and provides many services to the consumers of IT and uses the cloud infrastructure to help deliver these services. ITaaS changes the way in which IT consumers demand and utilize IT resources.

In our professional experience, this is where we find many IT organizations are at a crossroad. Attaining an ITaaS status is difficult as the combinations of cloud infrastructure and service models are vast with no formalized process to get there. Developing a private cloud is an intensive process that requires a well devised strategy around component choice of the hardware to software layers for both the production data and the CMP. Meanwhile, any attempt to “lift and shift” to the public cloud is usually foiled by the rigid nature of a business’s monolithic applications.

Therefore, we believe the best methodology for a business seeking to digitally transform must first understand their applications before they can craft a cloud-ready infrastructure.

It’s All About the Applications

Not all applications are developed equally. Application assessment is a critical step when designing a data center to be “cloud-ready” as it impacts both the business and function of critical components such as the CMP. Businesses must work with IT to identify which applications will be made available in the cloud. This collaboration helps determine the performance characteristics, availability requirements, security requirements, governance policies, and so on. However, unlike its the traditional counterpart, a cloud introduces functionality that will require further considerations for applications.

Traditionally, workloads are persistent, stateful, and monolithic. This means there is rigidness in the relationship between the application and underlying data center infrastructure supporting it. Although this workload runs in a virtual machine, it still maintains a client or application execution state in local memory or disk. These virtual machines must remain in an always-on state for the application to function appropriately. If a host running this virtual machine fails, the application must be repaired and may take time to recover. When traditional style workloads require better performance or capacity, the only option tends to be added resources to the existing virtual machine. This eventually means that the infrastructure supporting this type of workload must also be able to scale up to support additional resources. These applications can only scale well by running on larger hardware or must have a low tolerance threshold for component outages. Scalability requires not only the addition of physical hardware, but a seamless integration of that hardware and software components into the IT environment.

Due to rapid adoption of cloud technologies, there is now an emergence of what is known as “cloud-native” applications. Cloud-native applications are designed with the assumption resources will fail. Many public cloud providers build logical resource redundancy into their commodity hardware to tolerate these failures. Instances of an application that fail can be released and replaced with a new one. Applications are also designed to horizontally scale-out and scale-back as needed. Helping the organization understand these concepts is part of the architect's job. It also means that the architect needs to understand that some of the older ways to build infrastructure are no longer required when creating a cloud design. These applications introduce complexity into the software defined data center design. The architecture is disposable, stateless, and modular in design, meaning these workloads do not maintain any local state. If a host housing an application fails, it may be easier to turn on another copy or redeploy the workload on a different virtual machine. When cloud-native workloads require resources, they are usually scaled out horizontally by adding additional instances and include a load-balancing solution at the front-end. This means that the underlying infrastructure must also support a scale-out capability to easily add physical resources (i.e. compute and storage) to an IT infrastructure with minimal integration downtime.

The ability to successfully deploy an infrastructure for both monolithic and cloud-native applications depends on the structural layout and limitations of the data center.

Repurposing the Data Center for Cloud: Brownfield Deployments

Organizations that seek to repurpose their traditional data center into any cloud model is referred to as starting with a "brownfield" environment. In this type of environment, the goal is to design a cloud-based solution using existing infrastructure because the business is not buying as much equipment to support the new initiative. The benefits of starting in this position is that existing staff already has the required expertise to support the environment. While this saves valuable business dollars, it comes with a few shortcomings. The existing infrastructure or processes may place extra constraints on the design, which can undesirably impact performance or functionality and as mentioned previously with many traditional infrastructures, each component operates in silos. Another drawback to designing a cloud model through a brownfield environment is the extensive effort required to upgrade existing infrastructure or migrate existing workloads so that infrastructure can be repurposed. It is these hinderances that become the death of cloud-focused IT projects. As such, it is important for those starting from a brownfield deployment to consider a more "refactored" approach to evolving the data center with cloud technologies.

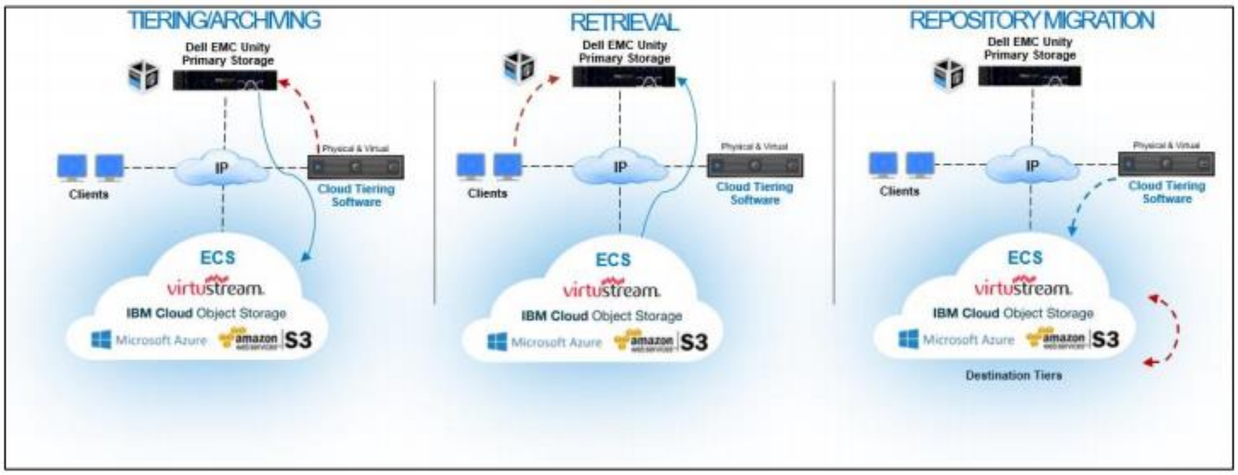
Cloud Extensibility

The easiest way to take advantage of cloud technologies is to incorporate cloud extensibility into the cloud. In delivering IT services to the business using a recipe of internal resources and external (usually public cloud) application services. While extensibility functionalities do not make sense for all aspects of the data center, where they do make the most sense are in improvements to business continuity processes. This includes functionalities such disaster recovery as single VM or long-term retention of files into an object storage service. These processes are orchestrated by a cluster of virtual appliances which enable further features such as data reduction and encryption during data transmission. The following are example solutions from Dell Technologies with cloud extensibility functionalities.

Long-Term Retention in the Cloud with Dell EMC Unity

Dell EMC Unity is a midrange storage array that consolidate block and file workloads into one platform. The array can extend into the cloud through a pair of virtual appliances called the Cloud Tiering Appliance (CTA). CTA enables administrators to facilitate cloud file migration, tiering, and archiving from the array into a target cloud repository.

An example of this functionality is exemplified best in a November 2017 whitepaper on the solution. As the whitepaper discusses, through a web-based interface, storage administrators can interact with CTA to create policies which interact with the storage system by identifying files that fit a predefined criterion. For these files, CTA initiates movement to a target cloud repository and places an 8 KB stub file in the original file location, to save space on the storage system instead of having to move the full size of the file[3]. When an end user reads the stub file, CTA will orchestrate the recalls or pass operations through the original file located in the cloud. This is completely transparent to the end user whose view of the file would still be the original location of the file Unity storage system. This solution makes file tiering seamless for the user who reads the archived file data and is supported on target cloud repositories, including Virtustream, Dell EMC Elastic Cloud Storage (ECS), Microsoft Azure, AWS S3, and IBM Cloud Object Storage (COS) [3].



Adapted from Dell EMC Whitepaper[3]

Data Protection with VMware on AWS

Taking advantage of extensibility into a public cloud provider is of immense benefit to the overall dynamic of the data center. Best practices must remain standard across both public and private cloud patforms. Dell EMC Data Protection Software provides exactly this by providing protected VM-level backup and recovery for workloads running VMware Cloud on AWS. Incorporating Dell EMC Cloud Disaster Recovery (Cloud DR), into their on-premises infrastructures, they can copy data into AWS S3, taking advantage of the cost-effective nature of cloud object storage. Recovery of VMs to VMware Cloud on AWS is another way to leverage this technology. Virtualization administrators can provision VMware Cloud on AWS's SDDC only when recovery is required, reducing overall cost in a flexible business continuity strategy.

Once the VMware Cloud on AWS environment is available and the user has deployed and connected the Cloud DR add-on (CDRA), users can recover from the S3 copies in 2 clicks simply by selecting the desired VM copy and it will be recovered as a new VM running within the VMware Cloud on AWS environment[4]. This orchestrated recovery process of the VMs protected by VMware Cloud on AWS requires no data conversion, providing recovery of 150 GB VM in 30 minutes on average. All of this consumes minimal compute cycles, enabling a disaster recovery solution with minimal cost. In the event of a disaster, the workloads can be run directly in AWS, or recovered to VMware Cloud on AWS.

Adapted from Dell EMC Whitepaper(4)

Whether on-premises or part of the business continuity strategy, cloud extensibility can be the first step in the journey to the cloud. However, another type of deployment may prove to be a more efficient path to the cloud.

A Clean Slate: Greenfield Deployments

An alternative approach to designing a cloud infrastructure is the introduction of a "greenfield" environment. Designed with exactly what is required to meet the business needs, greenfield environments use new infrastructure on pre-engineered hardware and software licensing suites. Though spending a little more to include aspects of training the staff on the newer technology, organizations may find the benefits outweigh the costs. In utilizing these prepackaged solutions, a cloud infrastructure starting here would avoid misconfigurations, constraints, and bottlenecks that exist in the current environment by starting with a clean slate. Greenfield environments have the added benefit of allowing a business to migrate infrastructure to integrate diverse technologies to avoid future lock-in.

Two IT infrastructure models whose adoption is growing rapidly in the industry today best fit this greenfield approach. These are known as converged and hyperconverged infrastructures.

All-in-One with Converged Infrastructures (CI)

Unlike its traditional infrastructure counterpart, a converged infrastructure can be described as one whose hardware components are tightly bundled with an underlying proprietary software. This single entity is a pre-configured bundle of compute, storage, networking, and virtualization software typically confined to a small number of racks, usually from a single or small set of vendors. While the design of these all-in-one solutions are rigid, meaning the hardware and software components in the solution are restricted to a compatibility matrix, this does allow for a streamlined and inclusive refresh cycle. This impacts scalability of the solution, requiring the addition of modular units of converged infrastructure.

CI with Dell EMC

The VxBlock 1000 enables a choice of multiple storage arrays, compute servers, and network stacks through customization of Cisco Systems and AMP management infrastructures. AMP management infrastructures are responsible for management and orchestration of the components in the smart chassis. The compute layer includes both Cisco UCS B-Series and C-Series Servers, while the storage layer includes multi-array support from VMAX All-Flash, Dell EMC Unity, PowerMax, XtremIO, and Isilon storage to fulfill both block and file storage needs[5].

Foundation for Hybrid-Cloud Agility-Hyperconverged Infrastructures (HCI)

Traditionally, data centers and, to some extent converged infrastructures, operate in silos. These pieces are expensive, requiring large upfront capital expenditure for hardware, software, and space. While vendor choice does allow for some flexibility in the environment, the technology behind each may differ, making synergy and growth of an infrastructure cumbersome. In the end, this results in high total cost of ownership (TCO) from the costly monitoring, management, backup, and support of the environment. To avoid these issues – particularly for remote office locations and mid-sized enterprises – IT leaders need to re-assess how they undertake every facet of what they do, how they do it, and how their IT can help drive business objectives. The data center today needs to be as effective as the vision of the business. In response to this, the industry is shifting from the traditional “build” model to a “buy” model in implementation of their data centers. What this means is that in one virtual cluster of servers and network switches, the data center has been condensed into a pre-engineered, pre-validated solution made up of nodes whose hardware and software is supported by a single vendor. Nodes are combined to present a distributed pool of compute, network, and storage and are operated as one entity. To add capacity in any one resource type, a node is added providing resources in all types but at smaller increments than in converged. The power of this solution is in the software-abstraction of management and traffic through virtualization technologies.

The Foundation of the Hybrid Cloud- VxRail

This is where hyperconverged solutions such as Dell Technolgoies’ VxRail can help change the very dynamics of your business This node-based architecture delivers virtualization, compute, storage and data protection in a scalable, easy to manage package. This plug-in appliance integrates with existing VMware management software, licenses, and standard tools, thus eliminating IT silos by keeping everything in one environment that supports replication, backup and recovery, and is cloud-enabled. VxRail eliminates the physical SAN by using software to turn commodity servers with local disk into software-defined storage, enabling storage and applications to run on the same physical server. It does so with kernel layer integration between VMware vSAN and the vSphere hypervisor vSAN, making VxRail the most powerful hyperconverged solution on the market as you have storage, compute, and memory in a virtual pool. VxRail uses vSphere to manage the fully virtualized environment. This, in effect, produces the power of a whole SAN in a four-rack unit appliance, providing a hyper-converged solution for a wide variety of applications and workloads.

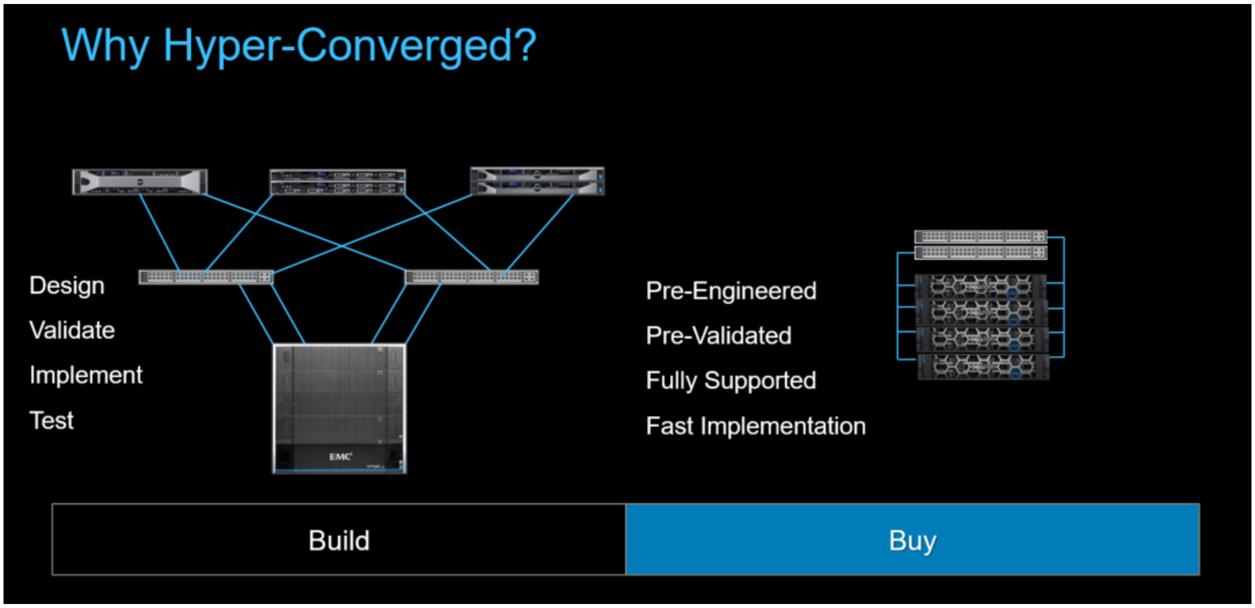


Figure credit: Dell EMC (2018)

While the complete integration of VMware into VxRail makes the automated lifecycle management of your infrastructure seamless, of even greater value is its flexibility in updates, licensing, and support.

When an update is required to the software, Dell Technologies ensures that all infrastructure pieces still work within the system, then package and update the code in the infrastructure through a one-button push, leaving more time for IT administrators to continue to drive business value. As a distributed system of modular building blocks**,** VxRail scales linearly from one to sixteen appliances with the ability to scale in node increments starting from the recommended 4 to 64 nodes in one cluster or add capacity drives to expand storage capacity of the cluster. Going greenfield with the VxRail, infrastructure can start as small as one appliance and grow out the environment safely while reducing overall power and cooling costs, eliminating the need for forklift upgrades and decreasing migration risks as additions to the infrastructure simply become part of the cluster.

HCI to Hybrid Cloud with Dell EMC

We believe the optimal route for an organization accelerating IT transformation to a Hybrid/Multi-Cloud deployment would be to start with a greenfield VxRail deployment. Support in both hardware and software components eliminates the need for IT administrators to worry about the day-to-day management of the cluster. Instead, they will be able to focus more of their time on “bridging the gap” between IT and business requirements by beginning to experiment further with automation, orchestration, and public cloud technologies.

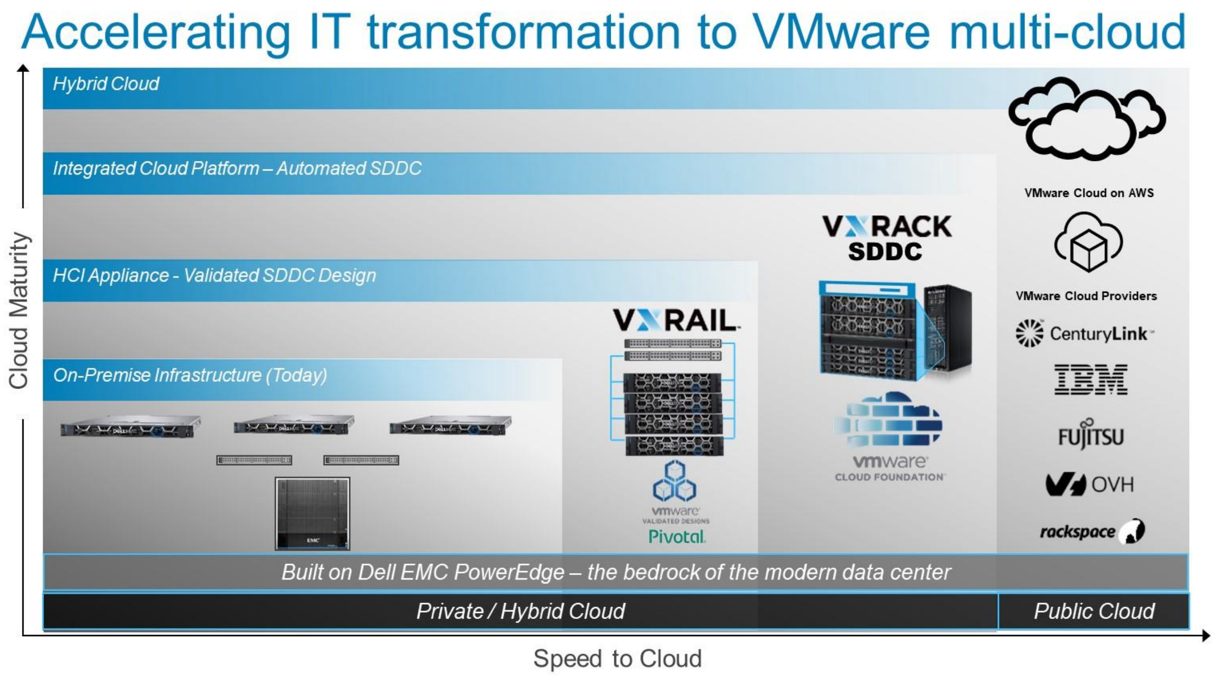


Figure credit: Dell EMC (2018)

The roadmap depicted above provides the baseline for an IT team to focus on strategic transformation. For those teams looking to build a complete Software-Defined Data Center using VxRail as a building block, they can now leverage guidance of the latest version of VMware Validated Designs (VVD) to reduce risk and simplify operations and maintenance by building a validated, standardized SDDC design that is based on VMware's best practices. This is done by adding NSX and vRealize Suite licensing into the HCI software layer. For those seeking to employ a service-oriented IT model with the fastest and easiest path to hybrid cloud, VxRack SDDC is the most advanced VMware Cloud Foundation engineered system. VxRack SDDC leverages the full suite of VMware software licensing to create virtual workload domains to virtually segregate workloads, delivering a complete, turnkey, rack-scale HCI system for VMware SDDC – from virtualized compute, storage and networking to the complete set of cloud management capabilities.

Realizing the VMware multi-cloud vision means to build a single, consistent hybrid cloud experience – one that is built on a simple, powerful, secure platform that can span multiple clouds. It means application extensibility and compatibility across private and public clouds. This creates a platform that can support both traditional VMs and newer cloud-native applications and containers.

Integration with Multi-Cloud Services

The ability to integrate both traditional and newer cloud native applications is mainly facilitated using APIs. This allows IT administrators to develop their own automation and orchestration strategies to enable extensibility and integration of traditional and cloud native applications. Adding these tools to the repertoire would make cloud migration and orchestration an easier process.

Orchestration with APIs- Dell Boomi

As businesses add new business units, merge other IT units through acquisitions, or experiment with public cloud solution offerings, application incompatibility is a problem that they will inevitably confront. In a 2018 whitepaper discussing the idea of the "connected business", identifying poor integration and business connectivity is a serious threat to the future existence of an organization as a competitive entity. There is a price to pay for poor integration – lost opportunities, lost efficiency, and lost time. For over four in ten (41%), ultimately it is costing them money in lost efficiency[6]. Solving this problem of poor integration and connectivity can be addressed with Dell Boomi.

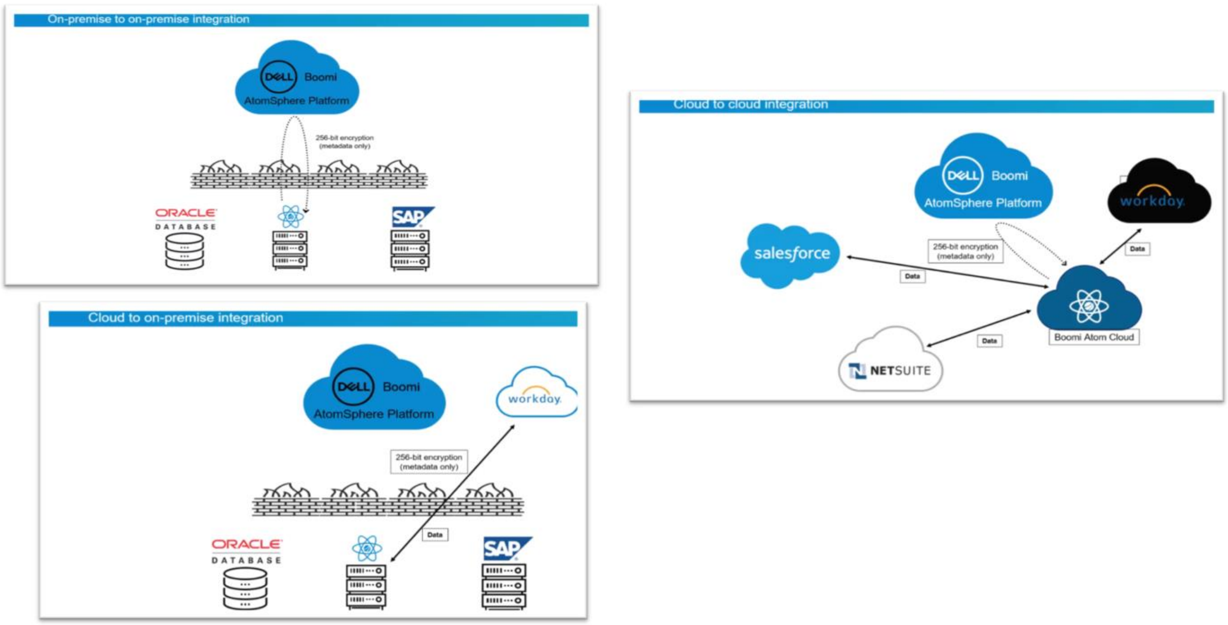


Figure credit: Dell Boomi (2018)

Whether on-premises or in the cloud, Dell Boomi offers an easy-to-use interface to create a workflow orchestration between different components. Delivered as an Integration Platform as a Service (iPaaS), Dell Boomi is the easiest way to give those experimenting with cloud integration a safe and effective way to do so.

Replication Across Clouds- Dell EMC Cloud Snapshot Manager (CSM)

Integration of a disaster recovery strategy through multiple public clouds is another great way to begin cloud integration. Dell EMC provides this freedom of cloud choice without requiring installation or infrastructure through the Dell EMC Cloud Snapshot Manager (CSM).

This SaaS solution offers a web-based policy engine which facilitates automatic assignment of resources to protection policies, enabling snapshots of AWS and Azure to be copied from one region to another in case of a disaster.

An independent study by the Enterprise Strategy Group (ESG) concluded that by implementing Dell EMC Data Protection Software such as this provides key benefits, i.e. reduced IT infrastructure costs, faster resource provisioning, and increased “time to value” for new applications and IT services. This, combined with ongoing dissipation of concerns over public cloud security, reliability, and data availability, will give early adopters the confidence to entrust more critical applications and processes to cloud services over time[7]:

• Up to 67.96% lower total monthly cost for in-cloud data protection infrastructure.

• Up to a 70.8% reduction in the amount of S3 storage required for the backup repository.

• Up to 87.8% lower monthly cost for the required EC2 and EBS data protection application resources.

No matter the plan to move into cloud or leverage solutions that span multiple cloud providers, Dell Technologies has the portfolio to start that journey. In the next section, we briefly explore emerging technologies which will change the dynamics of data center hardware, further muddling its evolution.

New Age of Enterprise Hardware

Digital transformation of a data center is an arduous journey, complicated further by the explosive growth in complexity of the applications it supports. Ever since the abstraction of physical IT resources through virtualization technologies, while data centers have become more efficient, they have grown equally in complexity. Hardware innovation has not slowed down either. Consider the rapid adoption of solid state disks (i.e. flash drives) as a storage media. The price of a 128 GB SATA-Drive in 2013 started at $91.98. The price of a drive almost three times that capacity in 2018; around $60[(8)]. Mass adoption and incorporation of this sophisticated storage media into the data center today is driving the performance bottleneck of the data center in relation to the applications it hosts. This is a trend that is continuing even further with the adoption of hardware technologies such as non-volatile memory express (NVMe), Storage Class Memory (SCM), and composable architectures.

Faster for Flash-NVMe & SCM

At a high level, the need to evolve Flash drives lies in the transfer limit of the standard system protocols. Flash drives can outperform storage interfaces and bus architectures intended for hard disk drives. Keeping servers with powerful, multicore processors and heaps of RAM waiting for data isn't exactly a prudent use of one's IT investments. NVMe is a host controller interface protocol that uses the PCI Express (PCIe) bus to connect flash drives to a server. NVMe was architected to take advantage of the parallelism of modern CPUs and flash drives and next-gen media and removes the bottleneck from storage (SAS), which maximizes the power of flash drives, and most importantly opens the door to the next media disruption; storage-class memory (SCM). Though not yet offered in the marketplace, SCM will be provide orders of magnitude better drive performance, bridging the gap between flash drives and volatile memory (DRAM).

Software is becoming the Bottleneck

The pair of storage media and protocol coupled with NVMe over Fabrics (NVMe-oF) network protocol will enable radically low latency storage services from an NVMe flash drive outside of a server, over Ethernet, Fibre Channel and InfiniBand networks, pushing the bottleneck of bridging the gap between internal and external storage.

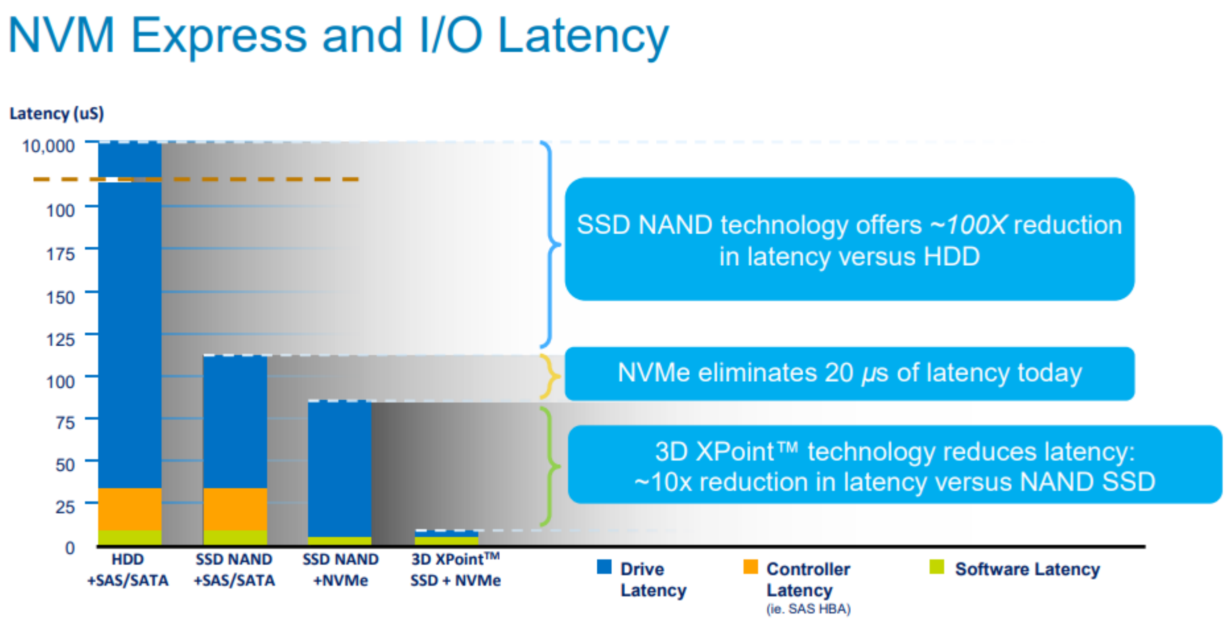


Figure credit: Storage Technologies Group, Intel. Comparisons between memory technologies based on in-market product specifications and internal Intel specifications.

NVMe, NVMe-oF, and SCM are designed to take advantage of low-latency transport and media. As such, once these technologies begin to find their place in data center refreshes, the combination of high-speed media formats coupled with minimal latency access mechanisms will foster new decentralized data center architectures which we discuss in the next section.

“Cloud-Like”: Composable Infrastructures

Imagine turning the physical data center into pools of modular building blocks for the applications to leverage as services. Picture CI & HCI infrastructures but the underlying hardware is just as flexible as the software allowing for heterogenous components of the data center to meet the demand for various workloads. No longer would any application or workload be siloed to the virtual domain it inhabits. This creates the idea of a decentralized on-premises data center architecture, which Robert Hormuth describes in a 2017 blog post titled, *Composable Infrastructure: Today and Tomorrow*.

In an effort by a consortium of OEM technology providers, the vision of a composable infrastructure, as Hormuth writes, means [9]:

- Customers can finally capitalize on the promise of “pay-as-you-go” from the end-to-end consumption model of IT.
- End users will be able to dynamically adjust IT resource consumption as business needs fluctuate.
- IT providers will be able to efficiently orchestrate business demands across their infrastructure without the need to physically set up or reconfigure hardware resources to maintain competitiveness.

Much of this sounds like enabling the flexibility and elasticity of cloud-based technology onto the on-premises data center. In obtaining such infrastructure fluidity, Hormuth provides insight to the vision from a hardware perspective. He explains that on the hardware side, accelerators, IO and memory inside the server are the least composable resources today because we don’t have technology available for dis-aggregation down into the memory semantic world. The difficultly lies in the elements we are trying to dis-aggregate – it's all about bandwidth and latency[9]. He then goes on to illustrate a rack-scale solution; the DSS 9000 and its benefits[9]:

- Ensure interoperability with heterogeneous systems
- Enable compute, storage and networking resources to be grouped together, provisioned and managed as one
- Allow customers to compose systems from the rack and across the datacenter
- All while working in the cloud environment of choice – whether its OpenStack, VMware, Microsoft or a custom orchestration platform

The below image helps illustrate this point.

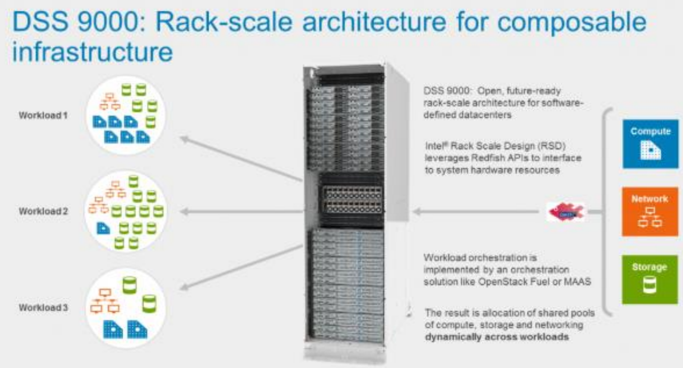


Figure credit: Composable Infrastructure: Today and Tomorrow[9]

Note the similarities between this architecture and the VxRack SDDC. They both utilize the abstract of the data center into "workload domains" to better manage applications that support the business. Where they differ is in this "cloud-like" elasticity which is a lot easier to obtain with a composable infrastructure design. Imagine being able to spin-up cloud-native applications on-premises. The flexibility this enables for the adoption of the hybrid cloud would be seamless.

While exciting, much of progression of composable infrastructures has been slow, as Hormuth shares: "OEM architects know that composable isn't a full-fledged set of capabilities today…That nirvana has not been reached and your warning alarms should go off for any vendor who tells you otherwise[9]".

What Does This All Mean?

While these technologies may be As NVMe, SCM, and composable infrastructures adapt to become industry standard. This explosive evolution in hardware is further driving the complexities of the data center back into its application layer. As the concept of composable infrastructures show, on-premises hardware is allowing a flexibility only seen in public cloud solutions today. However, as we know, this too is soon to be the main design focus of many technology vendors.

We encourage IT professionals to start taking small steps in digital transformation, rather than jumping in all at once. One way to start-we suggest-is looking "serverless".

May We Suggest "Serverless"

What is Serverless Technology?

Mainly found as a public cloud solution, serverless architectures are the missing piece between the on-premises infrastructure today and the cloud. Serverless computing is defined as the "abstraction of servers, infrastructure and operating systems from application development[10]. Serverless architectures eliminate the complexity and maintenance burdens of managing cloud resources as they do not require the provision or direct management of server technology. Rather, they abstract and automate those services to enable businesses to allocate additional resource and capital towards application design and development. Metrics such as memory consumption and time spent operating the request are tracked for cost purposes, requiring that business need only pay for the time and resources used when serverless functions are running.

This ability to scale to zero running instances is a key differentiator of a serverless platform. Serverless deployments offer a way to invoke one serverless function from another (whether it is a scheduled time event or a trigger), but some platforms provide higher level mechanisms for composing these functions and may make it easier to construct more complex serverless applications. This promotes an increase in compute resource utilization, and greatly reduces wasted operational cost of maintaining idle applications when not in active use or performing meaningful work. This resource optimized pricing model provides a definitive economic benefit for infrequent process execution and processes that run on a timed schedule or other processes that simply do not require uninterrupted availability. In this regard, serverless computing introduces a deeper level of abstraction in cloud infrastructure design from a Platform as a Service (PaaS) to a Function as a Service (FaaS).

Serverless architectures support a wide variety of programming languages including JavaScript, Java, Python, Go, C#, and Swift. Most platforms support more than one programming language. Thus, teams seeking to develop serverless applications that would help orchestrate migrating or replicating on-premises virtualized resources to a public cloud infrastructure would require expert-level knowledge in at least one of these programming languages.

We recommend that organizations seek to properly integrate public cloud technologies to handle cloud-native applications with an on-premises implementation to handle monolithic applications, Additionally, we believe all organizations should seek to develop a hybrid cloud through serverless technologies. In a hybrid cloud format, IT organizations maintain the best of public and private clouds. While maintaining full control over the on-premises portion of an application, an IT organization can scale in case of an unexpected spike in resource requirements for their monolithic applications while being able to develop cloud-native applications that may exist both on-premise or in a public cloud. Coupling best-of-breed technology with a service model solution to deploy services in both a public and private cloud provides the data center with hybrid-cloud agility.

The following sections explore a simple implementation of “serverless” technologies and explain how IT administrators can use it to bridge the gap between on-premises and the public cloud to enable synergy with their monolithic and cloud-native applications.

Limitations of Serverless

Weaknesses in FaaS designs include; applications that require higher computational processing power for durations over a minute; security; and complexities in software design. Concerning compute performance, serverless architecture is predisposed to producing higher levels of latency, a result of the additional overhead required to provision server resources code execution. FaaS applications benefit from automatic resource scaling and on-demand resource generation. FaaS functions generally require more time to execute as opposed to a monolithic application that may require the constant use of compute resources. In counteracting the initial added latency, FaaS resources must remain temporarily active for any consequent events.

Application and infrastructure security adds another dimension of complexity to serverless architectures. These issues are presented by Vince Power in his 2018 sponsored blog post *Serverless Security Risks Laid Bare.* The key issue relevant to the purpose of this document is centered around input validation. Proper input validation does more than merely ensure that area codes are properly formatted. Input validation stops some of the most common types of attacks that applications face. In the case of serverless applications, the most common type of injection attack is SQL Injection – an attack which involves inserting code into a request that is designed to either return too much data or destroy data [11]. This is concerning to serverless architectures as the purpose of the technology is to facilitate clean data across applications and infrastructures. This vulnerability is explored later in this document.

Finally, while FaaS provide unmatched elasticity and is an enticing cost structure for businesses, the limitations and added software complexity of this cloud architecture better suit certain use cases such as chaining together established applications and data sources in new ways (i.e. the example of storing database events in a separate data store), or new applications that can benefit from outsourcing sporadic compute resources away from the primary compute resource. This is where we take the best use case of serverless technology and apply it an on-premises and public cloud infrastructure to develop the concept of a serverless orchestration tool.

Our Vision: A Serverless Hybrid Cloud Methodology

To gain insight to bridging the gap between their current on-premises infrastructure and the public cloud, this article explores ideas and processes that enable an organization to go beyond cloud extensibility. In essence, having one piece of technology in the data center that can leverage the cloud in a limited fashion. This can be done through many different processes and with many different data center technology vendors, usually through what we would consider “server-driven” frameworks.

There is a demand in the industry today to be able to migrate virtual resources (i.e. VMs, snapshots, etc.) to public cloud providers. As authors of this article, we intend on sharing our knowledge of alternative “serverless” frameworks to assist enterprise IT environments in developing. This format would be different in the design of its hybrid or community cloud counterparts by incorporating serverless applications that facilitate migration and business continuity to form a hybrid cloud with an existing on-premises infrastructure.

By sharing this knowledge we believe that any organization can begin their journey to the deployment of a hybrid cloud infrastructure in an economically efficient manner. Leveraging serverless technologies to create a flexible cloud-based configuration eliminates the need to maintain multiple application programming interfaces (APIs) or manage different versions of code. By blending these enterprise technologies alongside serverless technologies, we believe the movement towards this new hybrid cloud infrastructure can be accelerated with minimal upfront cost. Developing migration and orchestration technologies through a serverless framework will offset the limitations of virtualization technology, ultimately reducing the overall cost in doing so.

The remainder of this article will cover what we believe to be the most efficient way for IT organization’s in any business, across any industry, to turn their data centers into a hybrid cloud. We plan to further develop on the ideas shared into a service to assist customers seeking to migrate to the cloud.

A Decoupled Design

Serverless architecture by design is flexible, scalable and responsive to unpredictable computational demands. This adaptability coupled with an exceptionally granular design structure allows engineering teams to produce products at an enterprise production state very quickly. One such use case to be examined is the potential of leveraging serverless architecture to extend software applications and data stores hosted on private servers as well as exploring the opportunity to leverage serverless technology to bridge on premises infrastructure to any cloud service provider.

To reiterate, a serverless application is a collection of loosely coupled functions that reside within a cloud service. Each function independently executes in response to one or more triggers or events. Example events could be scheduled triggers, webhook triggers, IoT device events, database-triggered events, or HTTP requests initialized from any source (including other serverless functions). Due to the loosely coupled nature of serverless architecture and the myriad means of connecting to a serverless system, FaaS architecture is well suited for quickly and efficiently bridging different systems, applications and data stores.

With the rise of microservices and distributed systems, traditional monolithic application architecture has given way to small, single responsibility applications that can be deployed, scaled, and tested independently[12]. While each microservice is independent in terms of application code, data storage and infrastructure, it is common for services to require data that is generated and stored within another service. As a result, mechanisms must be built to allow transfer of data between services without one service relying upon (or being aware of the existence of) another service to maintain the strict decoupling that characterizes a distributed system.

In this regard, there is strong similarity between decoupled microservices and that of bridging an on premises server to a cloud service. The cloud service may rely upon the state of the on premises server (or vice versa) but by design, neither should be tightly integrated with the other or aware of the others existence. This strict decoupling has two clear strengths – decoupled systems allow for smaller and more specialized engineering teams and the flexibility of a company being cloud agnostic in the services it chooses to utilize.

This decoupling allows specialized engineering teams to focus exclusively on the system architecture they are most familiar with. Just like that of a distributed system, this leads to rapid design to deployment workflows without fear of interrupting the processes of any other existing infrastructure. This decoupling also provides the company utilizing this hybrid structure the flexibility to intermingle cloud services as their budget and the skills of their cloud engineering teams allow without fear of integrating too heavily with a single cloud service provider.

While there are similarities between distributed system inter-service communications and hybrid cloud information passing, a clear distinction is the lack of an orchestration environment to assist simplifying the communication between services. This void of an overarching orchestration system can be supplemented with a combination of software design patterns and a flexible FaaS architecture.

A software design pattern often utilized in disseminating events in a loosely coupled manner is the Publish/Subscribe Pattern. When utilized in conjunction with a serverless backend architecture, the result is a dynamic pipeline that not only replicates and syncs data between systems but can be utilized to replicate and sync cloud infrastructure with modern DevOps and Infrastructure as Code practices.

The Publish-Subscribe Pattern

At a high level, software design patterns are generally agreed upon best practices, templates and solutions to common software architecture and design use cases. As Edward Crookshanks explains, a software design pattern is defined as “a known arrangement of software components that solves a particular problem[13].” It is also understood that no one design pattern is a “silver bullet” in that while each pattern attempts to solve a problem, pros and cons in doing so inevitably arise. This section will examine and detail the pros and cons of implementing a Publish/Subscribe Pattern within a serverless software system designed as a hybrid cloud solution.

The Publish-Subscribe software pattern facilitates communication between loosely coupled objects by means of a messaging-oriented middleware, or an event broker. In this design system, when a change in state has occurred an "event" is created and is communicated from the "publisher" application to the messaging-oriented middleware application. An event, as defined by Microsoft is "…the smallest amount of information that fully describes something that happened in the system. Every event has common information such as the source of the event, time the event took place, and a unique identifier. Every event also has specific information that is only relevant to the specific type of event(14)." Subscriber applications (as the title implies) "subscribe" to predetermined event classifications and are notified by the event broker when that class of event has been received. In this manner, the publishing applications are only concerned with passing event information to the messaging middleware without explicit knowledge of any subscribing recipients. Conversely, subscriber applications are only notified by the event broker for those events that the subscriber has an interest in with no knowledge of the publisher. This design system structure is illustrated in the image below(15):

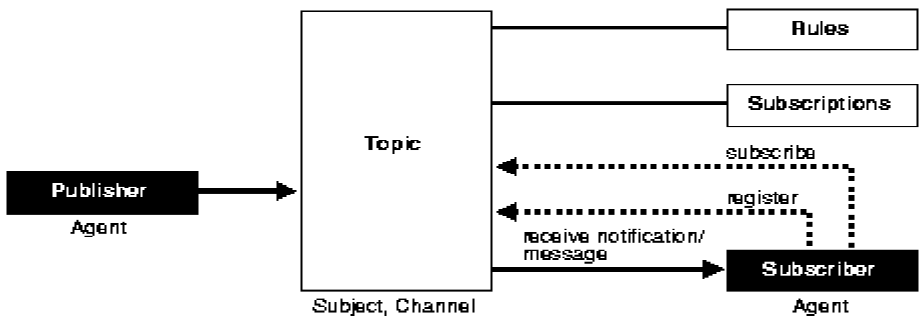


Figure credit: Oracle Application Developer's Guide - Fundamentals

The Publish-Subscribe design pattern provides software systems with two clear advantages – a loose coupling of applications and system scalability. In terms of coupling, both the publisher and the subscribers are ignorant of the existence of the other. This loose coupling allows for publishers and subscribers to not only be developed in different languages and frameworks, but in entirely separate hosting locations (cloud or otherwise). System scalability interestingly is both a positive aspect of this design but also the cause for potential concerns within this type of system. "…scalability remains a sensitive issue, because publish/subscribe interaction can be built on top of various communication substrates and can easily be hampered by an inappropriate architecture, when publish/subscribe systems are built on top of infrastructures that were not designed with scalability in mind(16)." As described by Eugster et al, a publisher-subscriber system is scalable in the abstract form but can be a hindrance when incorrectly implemented in an ecosystem that is not designed to scale as the demand fluctuates.

Two distinct disadvantages of the Publish-Subscribe pattern are the added complexity to the overall system design and the added requirement to maintain the integrity of the data messaging system. With regard to the added complexity, while the flexibility of building, testing and deploying many potential decoupled infrastructures and services allows for greater engineering autonomy, it also requires that specialized teams work and communicate cohesively. From a technical standpoint, each system is independent in infrastructure, application and data storage but require significant coordination and interpersonal communication to ensure the overarching processes are working in concert. The second clear weakness of this design model is the required development and maintenance of the systems that facilitate the messaging pipeline to ensure it is functioning correctly. For example, if a message that is produced fails to be delivered to the appropriate subscriber node, a system must be in place to log the chain of events that lead to the discrepancy so that the root error can be located and corrected. This added infrastructure requires time and resources to build, maintain and monitor and can result in an unstable environment if not properly established.

Serverless architecture effectively fits into this design system for two reasons – the dynamic allocation of serverless resources are capable of scaling to meet any event load (or turn off entirely in the absence of events to be processed) and the ability to connect from and into any existing system with micro facility. Serverless architecture is built upon the concept of scalability – each function dynamically allocates and deallocates resources to keep pace with the computational load that is required. This dynamic infrastructure fits nicely into the Publish-Subscribe design as a means of acting as a dynamic receptor that scales to the size of the event queue and disseminates the data or instructions accordingly. Autonomous, adaptive resource scaling further provides engineering teams the flexibility to focus on the function and design of what the events are meant to achieve without needing to spend added resources on infrastructure management. Conversely, in the instance that the event queue is empty, serverless resources "turn off" and do not accrue any costs.

The second advantage of utilizing serverless technology in conjunction with a Publish-Subscribe infrastructure design is the adaptive connectivity inherent to how serverless applications function with external sources. Serverless technology has been described as a "glue" that helps bridge and connect separated systems and services. Because serverless applications are entirely stateless and event-based, they can act as universal piping between any on premises or cloud application, data store and hardware if the publisher and subscriber are open to send or receive data from a trusted external source.

For example, a company may decide to use a cloud-based marketing application for programmatic communication and client interaction tracking, but their application stack (and customer usage analytics) resides within a on premises server. In this scenario, the marketing team would greatly benefit from customer usage analytics being available within the cloud-based software to facilitate better product targeting, customer service communication and to identify trends in communication strategic campaigns. In this instance, a Publish-Subscribe workflow with a serverless connector would be an ideal solution. As events occur on the on premises application (customers add a product to a cart for example, or a potential customer is viewing a type of product often) the event message is created and sent to a cloud-based messaging middleware. The middleware identifies the event as a marketing message and notifies the appropriate serverless application that an event has been received. The serverless application allocates the appropriate resources on demand, retrieves the message, transforms the data as needed and communicates with the marketing software via API. The event could then be logged in a cloud storage location for future analysis and auditing. The marketing software is now aware of the event that occurred on the server and the appropriate communication workflows are set in motion. Although this example examines a single event, in theory there could be a massive number of events being created from any number of sources each moment and the serverless applications would scale accordingly to meet the demand and transform and disseminate the events to any source required as needed.

In this example workflow, the on premises application and server are completely unaware of the marketing software and vice versa. If the example company makes the decision to migrate to a different marketing software, it can do so without any changes needing to be made to the on premises server. The serverless application code is updated accordingly and the of data between systems continues uninterrupted.

Serverless Cloud Foundation-- VxRail & Pivotal Cloud Foundry (PCF)

A great solution to start experimenting with this stateless infrastructure would be to combine the best of HCI with Pivotal Cloud Foundry (PCF).

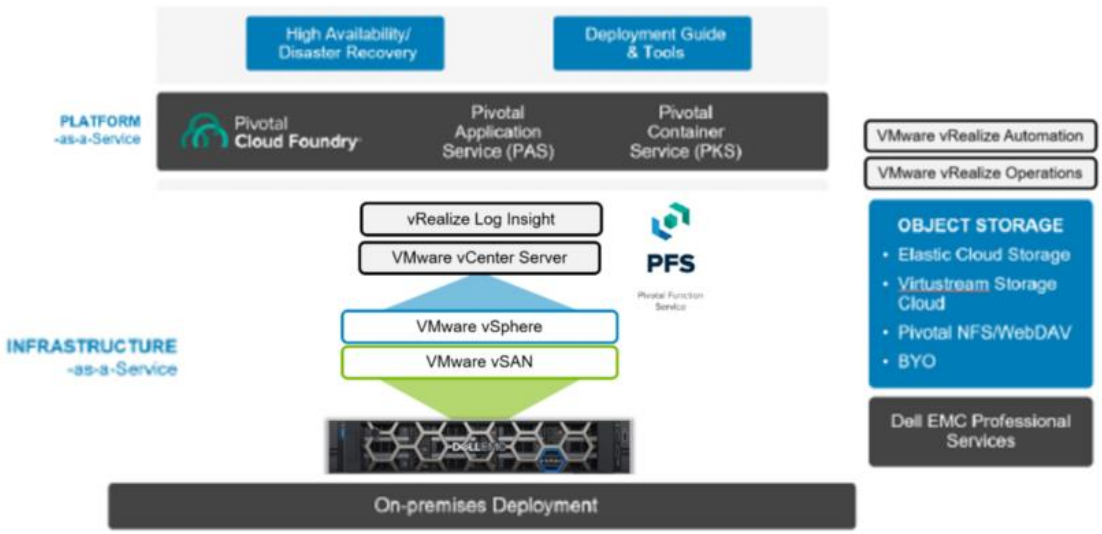


Figure credit: Dell EMC & Pivotal Cloud Foundry

This enterprise-grade developer platform on hyper-converged infrastructure helps organizations innovate faster using cloud-native applications in one pre-packaged solution, optimizing the self-service, cloud-native development capabilities of PCF. This offers the best path for rapidly deploying Pivotal Cloud Foundry on-premises, using reliable, resilient infrastructure for simple, seamless lifecycle management that keeps the environment up-to-date with nearly no downtime. In tandem, implementing the serverless computing product Pivotal Function Service (PFS) will add the ability to respond automatically to events with pre-configured, container-based workflows. It supports event-stream processing, connecting to message topics via a language-neutral, function container interface. Couple this on-premises, hybrid-cloud enabled infrastructure further with basic cloud FaaS offerings (i.e. AWS Lambda, Azure Functions, Alibaba Functions, etc.) and the vision of a “serverless” hybrid cloud is not far from reach.

In summary, gaining the ability to separate workflows via serverless technology will better facilitate the move to a hybrid-cloud infrastructure. The dynamic nature of the Publish-Subscribe pattern and how it uses events to trigger infrastructure as code affords users the ability to make serverless the focal technology around the orchestration and management of their monolithic applications on-premises to blend with cloud-native ones.

Future Technologies & Serverless

We conclude by taking a brief peek into what future technologies, such as blockchain and AI, can better support serverless methodologies. As many companies face the advent of strong data privacy laws, we believe through the blend of serverless and these technologies, we can suggest an architecture for prediction mechanisms and recommendation engines based on secure and self-learning AI models in the blockchain. This allows creation of user personas which are private and completely secured to the specific user. The configured “oracles” (services like Dell EMC Boomi) help communicate all relevant changes required to update the relevant self-learning models and generate prediction metrics based on user profiling. The abstractions to formulate the AI models are configured in the form of smart contracts in the blockchain which allows for updating as well as creation of new machine learning and deep learning models. The AI formulation on blockchain can also be abstracted based on specific use cases (i.e. Banking, Retail, Social media, etc.). A blockchain is a distributed digital ledger which inherently offers immutability, trust and security by design. The architecture of a blockchain is based on the principle of all nodes in a network working in an existing state of consensus always.

There are numerous use cases for a blockchain, the most prominent one being to solve the problem of double spending in digital currency transactions without the need for a centralized system. But what is not as known is the ability to add blockchain to create a distributed storage service. Decentralized cloud storage is more difficult to attack than traditional centralized data. On a decentralized network, files are broken apart and spread across multiple nodes, in a process called sharding[(17)]. These files are encrypted with a private key which makes it impossible for any other node participating in the network to look at your file. Plus, due to sharding, the files are just a fraction of their original self, which makes reading their entire content impossible[(17)]. Imagine data moving across serverless platforms, from on-premises to a public cloud provider completely secured. This would avoid issues such as SQL injection in serverless code, a huge benefit toward adoption of the technology. This is our vision of where serverless will go, but the possibilities are truly endless.

Appendix

(1) Dogtiev, Artyom (2018), "App Download and Usage Statistics (2018)", Business of Apps, http://www.businessofapps.com/data/app-statistics/

(2) Mell, Peter & Grance, Timothy (2011), "National Institute of Standards and Technology Special Publication 800-145", http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

(3) Dell EMC (2017), "Dell EMC Unity: Cloud Tiering Appliance-A Detailed Review", https://www.emc.com/collateral/white-papers/h16376-dell-emc-unity-cloud-tiering-appliance.pdf

(4) Dell EMC (2018), "Dell EMC Data Protection for VMware Cloud on AWS", https://www.dellemc.com/collateral/handout/h16519-data-protection-vmware-cloud-aws.pdf

(5) Dell EMC (2018), "VxBlock System 1000 Architecture Overview", https://www.emc.com/collateral/technicaldocumentation/vxblock-1000-architecture-overview.pdf

(6) Dell Boomi (2018), "The Connected Business", https://boomi.com/wp-content/uploads/Dell-Boomi-and-Vanson-Bourne-The-Connected-Business-whitepaper.pdf

(7) ESG (2017), "Whitepaper- Economic Value of In-cloud Data Protection with Dell EMC Data Protection Software", https://www.emc.com/collateral/analyst-reports/esg-in-cloud-data-protection-wp.pdf

(8) JCMIT (2018), "Flash Memory and SSD Prices (2003-2018) from NewEgg.com", https://www.jcmit.net/flashprice.htm

(9) Hormuth, Robert (2017), "Composable Infrastructure: Today and Tomorrow", Direct2DellEMC, https://blog.dellemc.com/en-us/composable-infrastructure-today-and-tomorrow/

(10) Owen Rogers, 451 Research (2017), "Economics of Serverless Computing", Page IV, Retrieved January 2019

(11)Power, Vince (2018), "Serverless Security Risks Laid Bare", The New Stack Blog. https://thenewstack.io/serverless-security-risks-laid-bare/

(12) Larrucea, X., Santamaria, I., Colomo-Palacios, R., & Ebert, C. (2018, May). Microservices. IEEE Software.

(13) Crookshanks, E. (2015). Practical Enterprise Software Development Techniques. Associated Press.

(14) Concepts in Azure Event Grid. (2019, January). Retrieved from Microsoft Documentation: https://docs.microsoft.com/en-us/azure/event-grid/concepts#event-delivery

(15) Using the Publish-Subscribe Model for Applications. (2019). Retrieved from Oracle Application Developer's Guide - Fundamentals: https://docs.oracle.com/cd/B10501_01/appdev.920/a96590/adg15pub.htm

(16) Eugster, P., Felber, P., Guerraoui, R., & Kermarrec, A.-M. (2003). The Many Faces of Publish/Subscribe. ACM Computing Surveys, 114-131

(17)BlockApps (2018), "How Blockchain Will Disrupt Data Storage", https://blockapps.net/blockchain-disrupt-data-storage/