OPEN ALLIANCE for CLOUD ADOPTION

Topic: Hybrid IT Responsibility

Contributors:

Brian Wilson – The Walt Disney Company
Mark Williams – RigD
Matt Estes—The Walt Disney Company
Pankaj Fichadia – NAB
Ryan Skipp—T-Systems
Shamir Charania – Keep Secure
Tom Scott—The Walt Disney Company
William Dupley — Liam Associates Inc
Krishna Jadhav – Tech Mahindra Ltd

Contents

LEGAL NOTICE

This "**OACA Hybrid IT Responsibility - White Paper** " is proprietary to the Open Alliance for Cloud Adoption - A Linux Foundation Project (the "**Alliance**") and/or its successors and assigns.

EXECUTIVE SUMMARY

Hybrid IT is a term that refers to the combined and simultaneous consumption of cloud and traditional IT models to deliver IT services in production.

This may include different public cloud platforms simultaneously, traditional internal IT and Cloud (private or public), use of IaaS, PaaS & SaaS, or FaaS together, and various other combinations that an enterprise may find effective.

Most importantly though, and this is what this paper addresses, is how a business must shape its IT Organization and modify the culture, processes, and systems to work between the different operational models that each of these IT service frameworks brings. This can be very difficult for a more traditional IT shop to deal with, and if not properly handled, may lead to loss of control, key staff loss, service failure, and potentially brand impact due to data breaches or service "pauses."

Leaders must therefore understand the different elements of Hybrid IT, what they imply regarding operational models, and then shape their IT organization and processes to leverage the different technology frameworks consistently, safely, and efficiently.

HYBRID IT RESPONSIBILITY MODEL

Dual or split responsibility is a term used to describe the relationship between a Cloud Service Provider (CSP) and the business/customer (often called the consumer or subscriber of cloud services). This relationship can be modeled in various ways to capture the roles and responsibilities of both parties. Creating a clear end-to-end chain of roles and responsibilities enables cloud partners and the enterprise IT groups to work together as a coordinated single team, with each individual focusing on their special part. This coordinated effort also creates much more efficiency and speed for the IT organization, much like a highly efficient production line.

If responsibility for services located in the cloud are not clearly assigned and managed between the enterprise and the provider, the cloud-based IT Services provided to the business may end up being incorrectly configured, inconsistently managed, or non-compliant. This can lead to significant business disruption and negative impact on the enterprise. For example:

1. The service may not be deployed in a known/legal/compliant geographic location, fault domain or jurisdiction, matched to the company's requirements.
2. Service element events may not be covered by the "assumed" support entities, and existing assumptions expose real gaps (and miscommunication) when a "day 2" outage or breach occurs, leading to commercial impacts on the enterprise.

A broader range of elements needs to be considered from a responsibility perspective in the new model, which differs extensively from traditional IT models. These include elements on two axes as illustrated in the graphic with varying levels of participation and responsibility illustrated in the intersect blocks:

1. Viewpoints (where the shared roles & responsibilities are assigned)
2. Domains & Services (where the technology and processing are bundled)

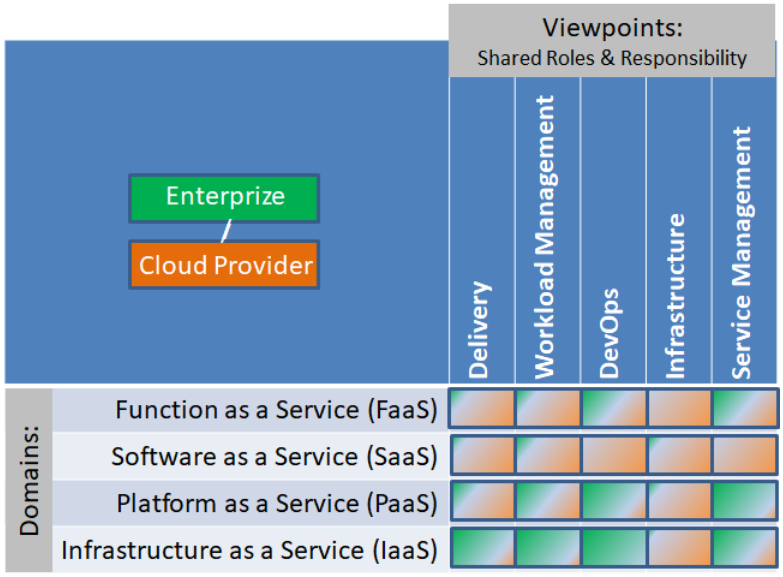

FIG 1: HYBRID IT RESPONSIBILITY MODEL

This paper describes the Hybrid IT Responsibility Model (and provides views and best practice recommendations that can advantageously improve cloud services for both the business and for the technology teams who support the business.

INTRODUCTION

As business functions continue to migrate into the realm of cloud service providers, businesses are being forced to redefine the level of ownership and control they have over their business functions and technology footprint.

With "traditional" on-premiseises environments there was an assumed level of control, i.e. you had exclusive ownership over every facet of the environment. From the air-conditioning, power, and fire suppression systems to the flavor of hardware and operating system selection, software, and programming languages - you controlled it all. In today's cloud model, the resource ownership and decision ownership has changed.

BUT your business is still dependent on the level of knowledge, support, and control, albeit "shared", the technology teams have over their environments.

It is said that "Cloud is just another data center somewhere that is now also being shared by others". Sharing resources will not impact your business as much as the blueprint for how you have architected your service organization and IT solution. If architected properly, sharing resources should not impact your business.

PROBLEM: RESOURCE SHARING

What will impact your business is how to continue to have control over an environment that you have 100% accountability for but effectively 0% 'real' ownership. This is a critical concept to grasp in maintaining jurisdiction over the technology that supports your business.

In order to retain control and ensure sustainability over your technology environments, no matter where they are, it will be critical to identify roles and responsibilities between the technology teams and the cloud service providers, not just from a data and security perspective, but for the whole technology stack. The lines that divide the roles and responsibilities can quickly become blurred, or worse, assumed. Often this will lead to significant challenges, especially when changes or incidents occur, leading to confusion in the roles and responsibilities of the support teams and how they respond / interact with each other.

PROBLEM: TEAM ALIGNMENT

Having misaligned teams often lead to significant challenges in not only how to respond to off-premises incidents but can also be disruptive for the technology teams that still need to operate, maintain, and support the vested on-premises, traditional environments. If roles are not clearly defined you could end up having on-premises teams being split, supporting the off-premises cloud deployments while still maintaining operational support of the on-premises environments leading to insufficient resources to support both.

Alternately you may end up with a deprecating support staff who have not been given the opportunity to train and support the off-premises cloud environments, leading to potential disruption of the business that is dependent on cloud.

In either case, this will usually lead to a gap in both knowledge and resources needed to sustain the business.

PROBLEM: PRODUCT MANAGEMENT VS. DEVELOPMENT PROCESS

The development space is crowded by another entity who is entitled to be developing, but their development and releases may have impact on your evolving IT services. This entity is the service provider and their service offerings across the entire cloud stack.

For example: you are developing a SaaS solution that is dependent on a PaaS build which provides certain functionality needed to support your implementation. The provider's PaaS lacks some needed functionality so you develop a build to address the gaps. As you are about to go live, your provider releases an update that addresses 85% of your gap build addressing the needed functionality….and now conflicts potentially with what your team has just developed!

This speaks to the need for coordination between the service provider's development and service releases and the consumer's build and deploy process. In the example above, if better communication was established up front, then the development team could have diverted resources to other builds and abstracted the missing functionality in order to address, for the interim, the PaaS shortcomings.

It is important for leadership to recognize that another entity, i.e. the cloud provider, can introduce changes into the operating environment which can potentially cause an impact. It is equally as important that the product management teams need to understand and align with the providers' build-and-deploy roadmaps and timelines. This will enable dev teams to align resources with deployments, enabling synchronization with changes, as they occur.

PROBLEM: EVERYONE NEEDS TO KNOW WHAT IS GOING ON

Keeping in mind that there are multiple teams across the company, one must also recognize that there are multiple teams across the provider, not to mention SaaS teams, vendors, etc. who are all working to deploy and maintain the solution environment. This means that where enterprise software builds were locked down and protected from change previously, that this is no longer true.

Data management tools are also needed where they were not needed before, and there is a need to re-examine the architecture to support this new model. An oversight role is needed to ensure that all parts of the system environment remain coordinated.

As discussed later in this paper as a best practice, there should be a Service Owner who is supported by a Technical Lead. There should also be continuous re-evaluation of the architecture (as part of a regular process) as to whether the existing deployment is still technically current or not. Performance of this role, often called the “tech lead”, is someone on the tech team, usually an architect, who coordinates with the product manager/owner, strictly follows the development process, and is accountable across the hybrid service chain.

HYBRID IT

Hybrid IT represents a model of enterprise computing in which organizations deliver business value using a combination of services and information technology (IT) resources hosted both on-premises and in the cloud. Hybrid IT can be thought of as using different delivery models to host the various components of an overall solution.

For example, a business may make use of public cloud-based web interfaces, on-premises service buses, cloud-based storage solutions for long term retention, and an on-premises application which leverages PaaS or SaaS offerings from “outside”.

A hybrid model enables an enterprise to maintain a centralized approach to IT governance, while leveraging cloud computing and services that enable them to avoid their own development/maintenance, and accelerate their overall service evolution delivery to their client base.

The term *hybrid IT* is regularly interchanged with the term *hybrid cloud*. Hybrid cloud can also refer to a cloud architecture where a vendor who has a private cloud, forms a partnership with a

public cloud provider - or a public cloud provider forms a partnership with a vendor that provides private cloud platforms. Three primary cloud types can all be part of such a Hybrid operating environment:

1. **Private Cloud**: Typically on-premises of an enterprise and dedicated to that one enterprise
2. **Public Cloud:** Typically a shared environment, shared between many enterprises, usually accessed over the internet, amongst other routes, with no one individual enterprise aware of who the other consumers are or what infrastructure or service endpoints they share with them.
3. **Community Cloud:** Similar to a Public Cloud, usually with a specific function range, servicing a defined community, with controlled access and use – e.g., government services, university campus environment, industry specific groups, etc.

SHARED RESPONSIBILITY

A MODEL FOR MANAGING SHARED RESPONSIBILITY

In order to structure the relationships and responsibilities in a shared consumer/provider environment, a model is helpful. There are multiple approaches, and the following illustrative model can be considered as a good base to start evaluating your organization and clarifying responsibility. Within this model are two distinct roles:

1. Cloud Provider
2. Cloud Consumer

For the purposes of this paper we use the NIST Cloud Computing Reference Architecture (NIST SP 500-292) scope: "The Cloud Provider and Cloud Consumer share the control of resources in a cloud system." The grey blocks in the middle may be considered as "Domains," and the Provider and Consumer may be considered as "Viewpoints"

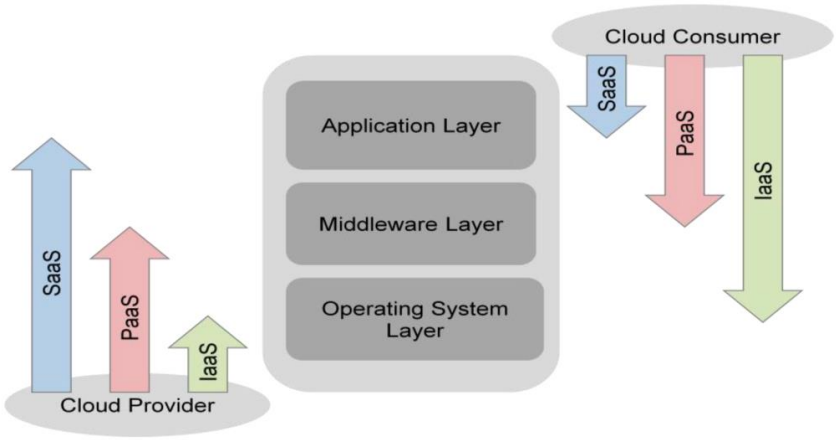

FIG 2: POSITIONING RESPONSIBILITY

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-292.pdf

It should be recognized that there must be a delineation of control based on the cloud service model being leveraged in each case. The service model in use (e.g., IaaS/PaaS/SaaS/FaaS/etc) defines the responsibilities of parties and the service start and stop points for those participating in managing the cloud service. This is illustrated in the model below (Cloud Service Provider=CSP) :

Hybrid IT Resource Model

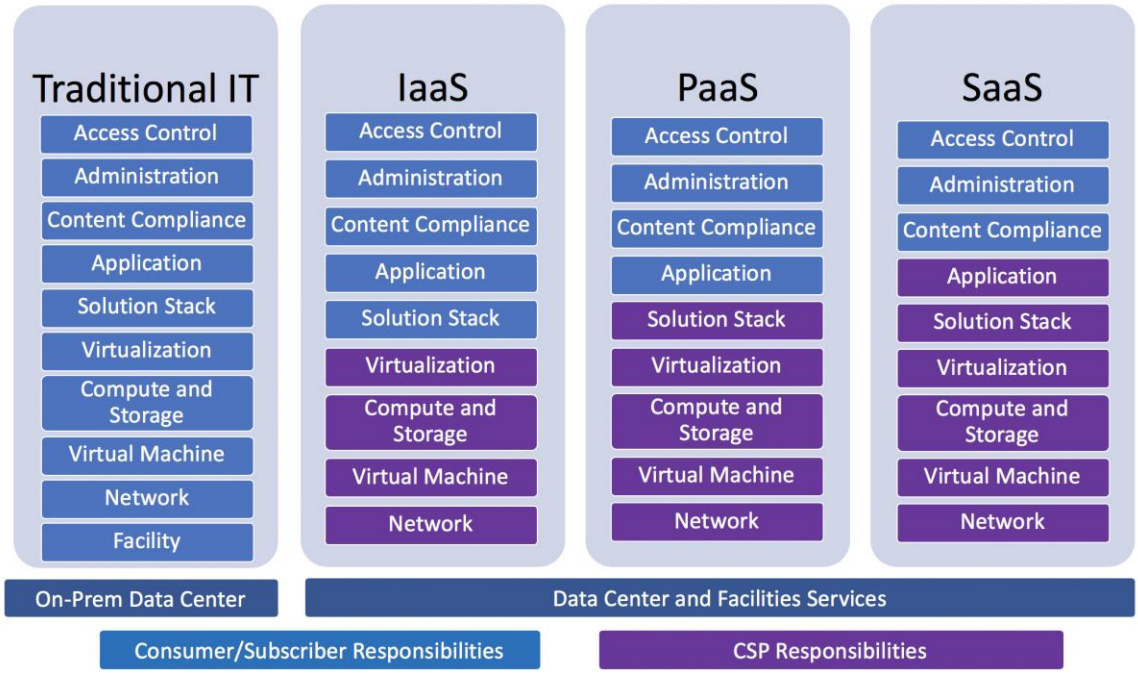

FIG 3: RESPONSIBILITY WITHIN CLOUD SERVICE MODELS

In order to ensure sustainability of operations for each service model, both the provider and the consumer need to identify their actors (i.e., roles & responsibilities) and clearly assign service and communication interface points. These defined interfaces specify the point where the roles and responsibilities change, and each is called a "**service transfer point**".

The service transfer points between the Provider and Consumer are used to identify who has what responsibility at what time. In order to align technology teams with service transfer points, businesses will need to change the way they map responsibilities for the different aspects of the business function.

This process, also called the "Service Chain", is used to coordinate the service at the transfer points between the various parties. These parties can include various IT departments, application support and operation teams, the cloud service providers' as well as involved SaaS providers.

Layers of functions can range from network provisioning, through platform services, containerized services, micro-services, security, and data protection.

VIEWPOINTS & DOMAINS

It is easier to define roles and responsibilities if we consider them from a 'viewpoint' perspective. A number of viewpoints must be considered which carry responsibility amongst them, and they can differ enormously between the Consumer and Provider roles in a Hybrid IT/ Hybrid Cloud environment:

1. Hybrid Delivery
2. Hybrid Application Workload
3. Hybrid DevOps
4. Hybrid Infrastructure
5. Hybrid Service Management

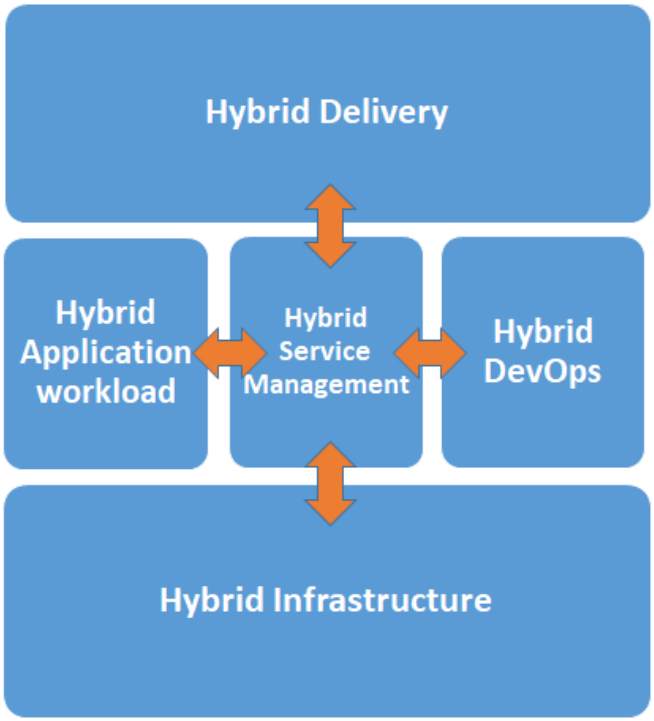

FIG 4: COMPONENTS OF HYBRID IT RESPONSIBILITY

Based on the OACA model, we will next consider these viewpoints in detail and how they relate to each other.

1 - HYBRID DELIVERY

IT is reshaping the Global Economy. New forms of knowledge resulting from in-depth Big Data capture and analysis is rocking the foundations of many enterprises worldwide. Conventional service portfolio management and Enterprise Architecture are no longer meeting the needs of the business. Transformation and Innovation now go hand in hand as the creators of this new world order. It is not IT developing these new solutions. It is the business users themselves. The new business innovators exploit the myriad of new cloud services, such as SaaS, FaaS, and PaaS, to rapidly create new service offerings. These citizen integrators are creating completely new IT solutions in just days.

The historical approach to Enterprise Architecture and Service Portfolio management can be defined as the development of standards that businesses had to use. There was little freedom. It did not support creativity to enable a business to change quickly. To address today's business reality, rapid change is critical, so IT needs to change IT to accomplish this. IT needs to enable a business to assemble its own applications and systems and rapidly bring those new IT solutions to production very quickly while enabling them to be repeatedly changed under their own control and approval. All these changes are in complete opposition to conventional ITIL control processes.

The new Hybrid delivery model is at the heart of this transformation. Enterprise architects must focus on creating an engine for change. Enterprise Architecture is no longer a puzzle made of parts that have to fit together to form a predefined outcome. Rather, it needs to be viewed as a game where there are predefined rules and resources that business players can utilize to create different products and functions. Hybrid delivery delivers those services.

This does not mean that we abandon standards. Even a chess game has to have rules for the players, or there will just be chaos. An enterprise architecture still needs rules to ensure systems are interoperable and adhere to governance requirements such as record retention and security. Enterprise Architecture needs to focus on enabling citizen integrators to build their systems - this is called DIY (Do It Yourself). Hybrid Delivery must deliver this capability.

Business appreciates the ease of being able to order cloud services from providers (such as Amazon). They want the same ease of provisioning across all of their IT services. Hybrid Delivery must operate in the same way. A Hybrid IT Service Delivery model acknowledges that all IT services are no longer delivered from within the walls of an IT organization. Services now can come from the public cloud, SaaS providers, private clouds, virtual private clouds, and traditional services. In a Hybrid Delivery Model, IT must become a one-stop shopping source for all IT services.

IT needs to deliver an Integrated Hybrid Service Portfolio of pre-approved cloud providers and SaaS suppliers that meet the service level expectations of their Business customers. They need to ensure those services adhere to the governance policies of their enterprise. The graphic below illustrates the Hybrid Delivery operational model.

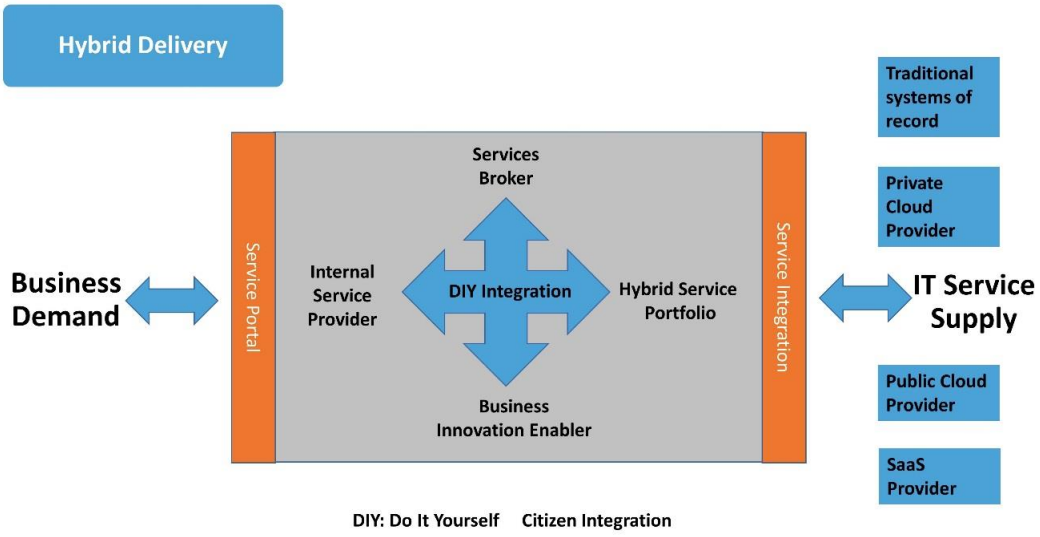

FIG 5: HYBRID IT DELIVERY MODEL

AREAS WHERE A CHANGE IN HYBRID DELIVERY IS REQUIRED

Historically, IT services were delivered individually. Development and integration of IT systems was the responsibility of IT and businesses rarely developed any systems other than a few complicated spreadsheets or Microsoft Access databases.

The Hybrid Delivery operating model supports citizen Integrators and reduces the need for business users to have to involve IT when they need to build or integrate systems together. To accomplish this IT must transform their operational delivery model in the following areas:

1. Hybrid Delivery processes and roles.
2. Hybrid Delivery architecture.
3. Hybrid Delivery service portfolio.

1. HYBRID DELIVERY PROCESSES AND ROLES

The graphic in Fig 6 further below outlines the work, roles, and OACA domains that Hybrid Delivery impacts. The overall responsibilities of the shared responsibility model are as follows.

- **Deliver Service broker services**: This is the ability to deliver Traditional systems of record, SaaS, Private cloud, and Public Cloud services from one portal.
- **Deliver Service Provider services**: Ensure that all services provided adhere to Enterprise governance policies (e.g., Record retention, Country of Origin) and meet business service level expectations.
- **Facilitate chargeback for all services:** This ensures that departments that order the services are charged for the services.
- **Facilitate the purchase approval process:** This ensures that those who order the services have the budget for them and that standard purchasing approval processes and authorization are enforced.
- **Provision Systems integration services:** This provides a catalog of APIs to existing services (often with documentation).
- **Provision patterns of use:** These patterns explain how applications should be integrated together to ensure good operability and security.
- **Deliver Business innovation enabler services:** These services provide consulting advice on how to use the new citizen integration services and which cloud service should be considered to meet a client's needs.

The OACA domains impacted include:

- Financial Management
- Enterprise Strategy (Business)
- Enterprise Structure (Business organization)
- Culture transformation
- Skill development
- Compliance management
- Governance and control
- Business process design
- Procurement management
- Commercial documentation
- Portfolio management

- Project Management

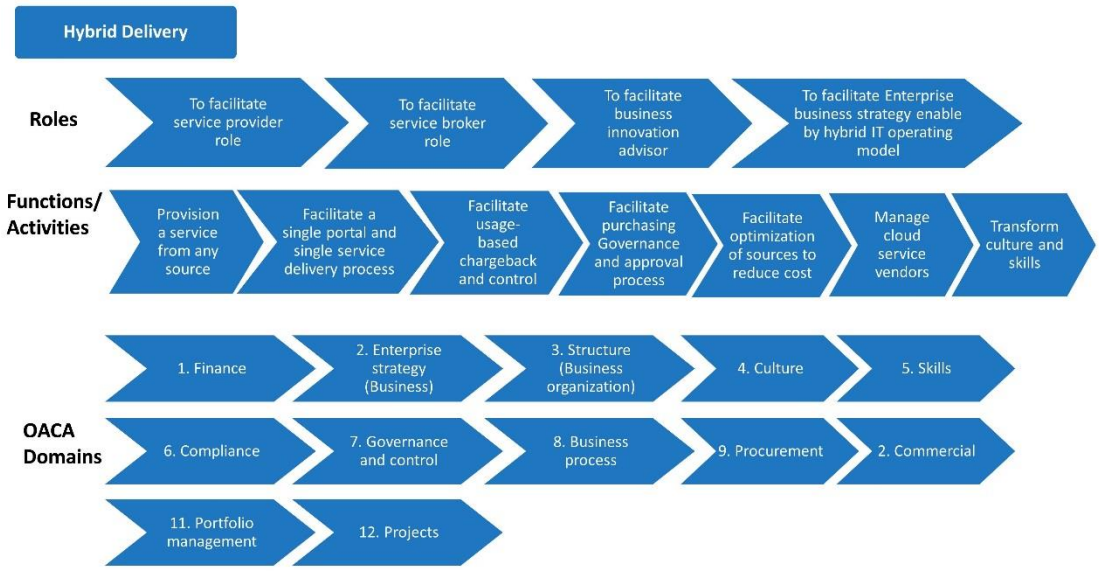

FIG 6: HYBRID DELIVERY COMPONENTS

Change #1: IT must become a Service Broker.

Service broker: IT is responsible for facilitating the provisioning of services from any source to the customer, facilitating the chargebacks for those services, and providing one-stop shopping through a common portal for their customers. There are several processes and systems that need to be changed or developed to make this possible. These changes are summarized as:

- Development of a Hybrid Delivery Portal (Service Catalog) that is integrated to cloud providers.
- Integration of Cloud provider's billing systems to the Hybrid Delivery chargeback system.
- Implementation of an approval process and system integration that ensures that when a customer orders a service from the Hybrid Delivery portal, they have the authority and budget to buy the service.

Change #2: IT must become a Service Provider

Service provider: IT has the responsibility to ensure that the services that they provide through their portal meet the required service-level agreements and the services offered meet the governance policies of the enterprise. The Hybrid Delivery Service catalog must document the service levels of each service offered and monitor the actual services as they are used.

Some SaaS and cloud service providers may have service levels that do not meet the needs of an enterprise. The Hybrid Delivery portal must clearly state these limitations or limit the use of such services for production. IT must also manage the relationship with cloud providers and address any shortfalls in service levels.

Change #3: IT must deliver Business Innovation Enabler Services

Business innovation enabler: IT has a deeper or perhaps new responsibility and role to help customers identify the best methods of delivering workloads. They need to be able to recommend SaaS services, PaaS, or other delivery models to their customers and provide integration advice.

2. HYBRID DELIVERY DOMAINS & ARCHITECTURE

Change #4: Common Hybrid Delivery Portal

Hybrid delivery requires the implementation of a universal service delivery portal. This portal enables a customer to understand the approved cloud services available, the service level expectations, and the costs while facilitating robust, on-demand ordering and fulfillment capabilities. The portal facilitates all approval cycles and provisioning activities with providers. The portal also needs to integrate all internal delivery systems and approved cloud providers.

Below is an example of a Hybrid Delivery technical architecture.

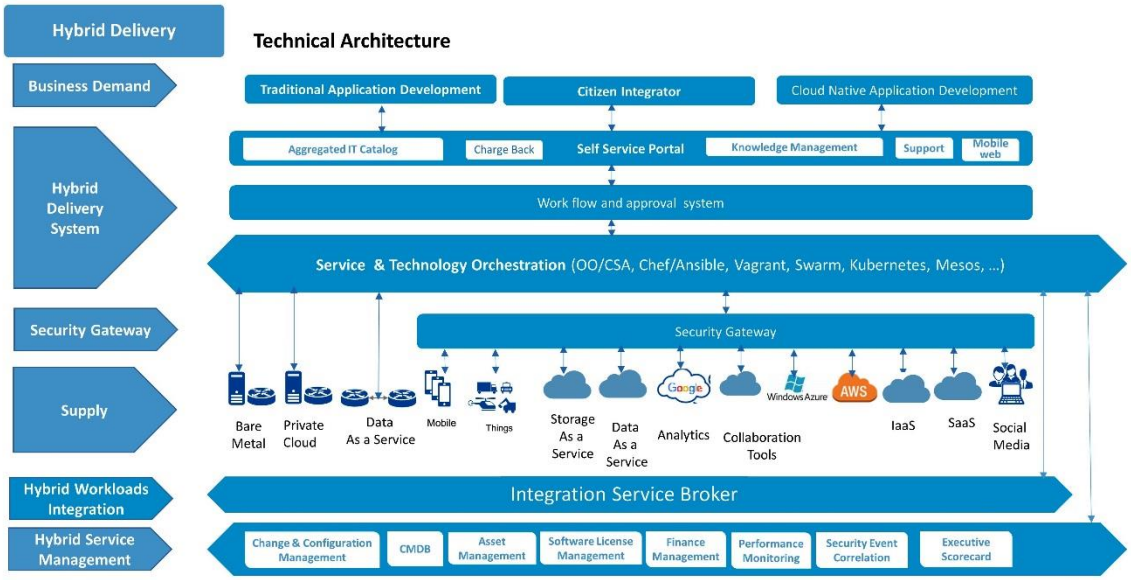

FIG 7: HYBRID DELIVERY ARCHITECTURE

In most firms, such an implementation will require a significant overhaul of the provisioning process, supporting tools, and service catalog to accomplish the desired result.

3. HYBRID DELIVERY SERVICE PORTFOLIO

Change #5: Service Portfolio Transformation

A conventional Service Portfolio / Service Catalog portal consists of services in the following categories.

- **Accounts and Access:** includes WebLogin and Two-Factor Authentication, ID Card, Authority Manager, and more.
- **Backup and Storage:** includes Encryption, Data Storage, and Backup services.
- **Business Intelligence and Reporting**: Business Intelligence, providing analyzed data and report summaries, to support decision making at both human and automated levels.
- **Communications:** includes Telephones, Cell Phones, Cable TV, and more.
- **Consulting and Development:** includes Business Solutions Consulting and Development.
- **Document and Digital Asset Management:** Includes file sharing and document management, electronic document storage and imaging, and multiple-use tools such as Google Apps.
- **Email and Calendar:** includes Email & Calendar, Mailing Lists, and more.
- **Networks and Connectivity**: includes Network, Wi-Fi, VPN, and more.
- **Productivity and Collaboration:** includes video conferencing, Content Management, and Instant Messaging.
- **Security:** includes desktop and mobile device management, antivirus, disk encryption, SEIM, and more.
- **Servers, Storage, and Data:** includes centralized server hosting, system administration, virtual servers, Storage, and databases.
- **Software and Business Applications:** includes software distribution and licensing, service ordering, and application implementation and/or hosting.
- **Support and Training**: includes an online help system, Incident Management, Knowledge Management, Technology Training programs, and more.
- **Web Development and Hosting:** includes Web Services, WWW hosting, content management systems, survey and form tools, and more.

A Hybrid Delivery Service portal must facilitate the provisioning of these services, plus the following new categories and enhancements to existing catalog items.

Enhancements:

- **Consulting and Development**: Business innovation enablement services need to be added to the Professional service portfolio.
- **Software & Business Applications**: SaaS must be added.
- **Servers, Storage, and Data**: Infrastructure as a service, private and public cloud services must be added.

New Services

A number of new services must also become part of the catalog:

Platform as a Service or Application Platform as a Service (aPaaS): is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.

Functions as a Service (FaaS): is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app. It includes that ability to build an application using "serverless" architecture and is typically used when building micro-services applications.

Data as a Service (DaaS): includes the ability to access data sets using web service APIs that adhere to the REST architectural constraints

Change #6: Publish Application Architecture Patterns

IT needs to publish Application Architecture patterns of use that describe how systems should be assembled to ensure interoperability and adherence to security policies. An architectural pattern is a general reusable solution to a commonly occurring problem in software architecture within a given context [Ref 2.1]. Architectural patterns are similar to software design patterns but have a broader scope. The architectural patterns address various issues in software engineering, such as computer hardware performance limitations, high availability, and minimization of a business risk. Some architectural patterns have been implemented within software frameworks. Below is an example of an Application Architectural pattern of use.

Example: Composite application Architecture (multiple patterns)
Partner Services Application to dispatch and reimburse partners for service

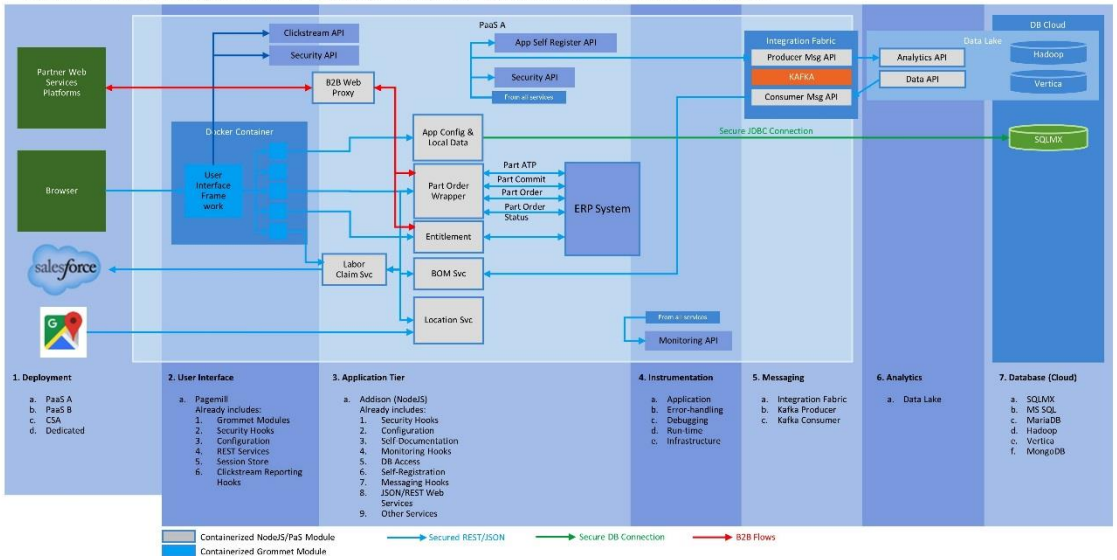

FIG 8: HYBRID DELIVERY APPLICATION EXAMPLE

2 - HYBRID APPLICATION WORKLOAD

Businesses in every sector today have a myriad of specialized FaaS and SaaS offerings and services available to them which are tailor made for their industry and compliance requirements. This makes it really easy for them to be able to bundle functions together as they create new products and services and to then retire those that they don't find effective. However, in order to stay competitive, they also usually need to develop their own special elements of intellectual property in their processing chain, or else their competitors would quickly duplicate and overtake them. In addition, there may be very sensitive data that they do

not feel comfortable placing in the public cloud space and so rather retain it in the Private cloud with secure connections to public cloud functionality. These proprietary elements of development, sensitive data, and controls are usually located within the company network and its data centers.

The business applications that leverage functionality from the public environment and resources within the private environment are what we term “hybrid application workloads”.

It is often difficult for traditional IT teams to deal with these – as they do not have control over the public cloud functions, but they are part of the critical business process chain. Monitoring, governance, and administration need to be set up to support the matrixed teams, who operate these hybrid applications.

In addition, it has to be understood that the cloud provider will continue developing their offerings in order to remain attractive to specific industries, and to stay ahead of the competition. This forces the business consumer to stay up to date as well and drives a need for DevOps and IT Product Management. No longer can one siloed team just build an application and expect it to “run forever” as released initially. Another challenge is that if some of the cloud providers’ offerings don’t sell well, the provider may want to stop development of them or retire them, even although they may have become critical to your business!

From the above, it is clear that the cloud provider and consumer need to have working communication and interfaces between them in order to manage these hybrid workloads, understand each other’s needs, and coordinate associated activities. In addition, dealing with processing peaks is important – and both sides need to be able to deal with scaling needs interactively. These include data and system access between the publicly located application portions, and the privately located portions, so as to maintain end-to-end service quality for the end consumer.

HYBRID APPLICATION WORKLOAD PROCESSES AND ROLES

Processes should be considered as follows for this dimension of the model (which follows below):

- **Application Delivery:** Facilitate the delivery of all components of the application no matter where they are located (between the provider and the enterprise)

- **Integration Platform Enablement**: Deliver the Integration Platform as a Service to equip "citizen integrators" with the necessary tools, functions, and capabilities
- **Data & Information Services:** provision the required data and supporting resources, as well as the rules and classifications around the applications environments, for citizen integrators to work within.
- **Provide the Integration Fabric:** create, manage, and provide the necessary elements that citizen integrators will need to find, connect, and integrate business functions, business data, and message bus interfaces so as to turn them into operational services.

The OACA Domains impacted include:

- IPaaS
- IT Architecture
- Applications
- SaaS
- Data
- AI
- API's

A model representing Hybrid Application Workload and some of its dimensions follows below:

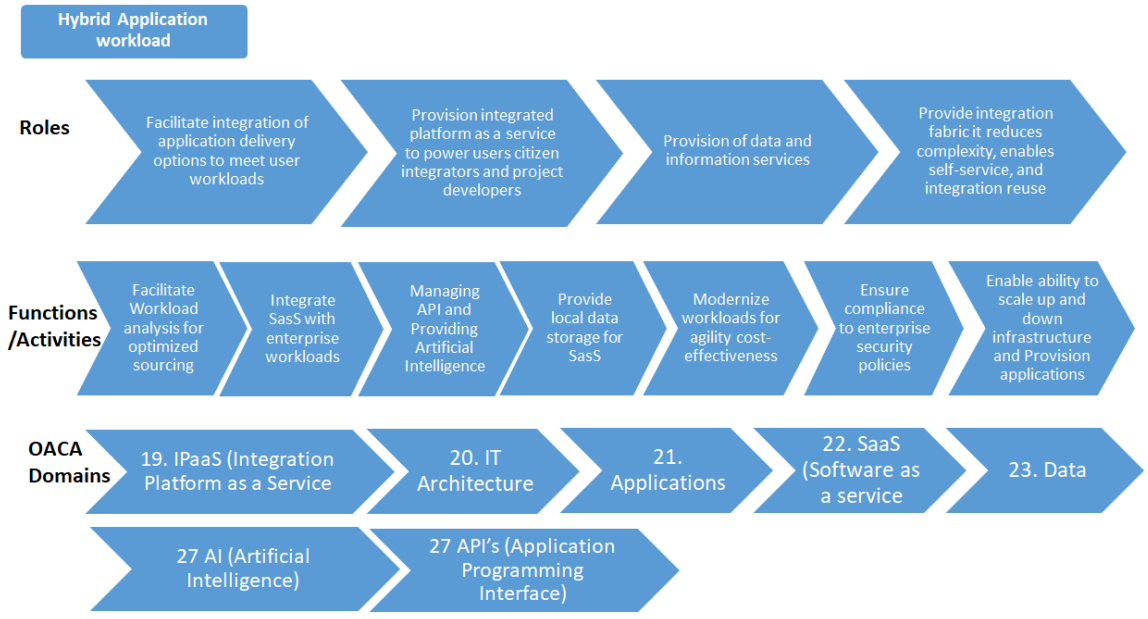

FIG 9: HYBRID APPLICATION WORKLOAD COMPONENTS

HYBRID APPLICATION WORKLOAD DOMAINS & ARCHITECTURE

IDC (International Data Corporation) has described that there are three application development platforms in regular use in today's enterprise application landscapes [Ref 1.1].

These platforms are:

- 1st Platform: Centralized IT and/or Mainframe.
- 2nd Platform: Decentralized IT, LAN/Internet Client.
- 3rd Platform: Democratized IT. Democratized IT is an environment that enables anyone to do the work that historically only IT did.

A Hybrid Application leverages the following capabilities:

- Ability to integrate SaaS with back-office systems.
- Ability to facilitate citizen integration.
- Ability to migrate production workloads from a private, public, or community cloud to a separate private, public, or community cloud provider on demand.
- Ability to merge traditional deployments with software as a service (SaaS), across various cloud types, public, private, etc., using managed cloud resources.

Almost all applications today are accessible via mobile devices and most back office systems are now integrated in some way to cloud services.

Following is an example of a Hybrid Application stack. There are thirty applications in this system. Seven are in the cloud, the remainder are in the customer's data center. There are four different communication protocols.

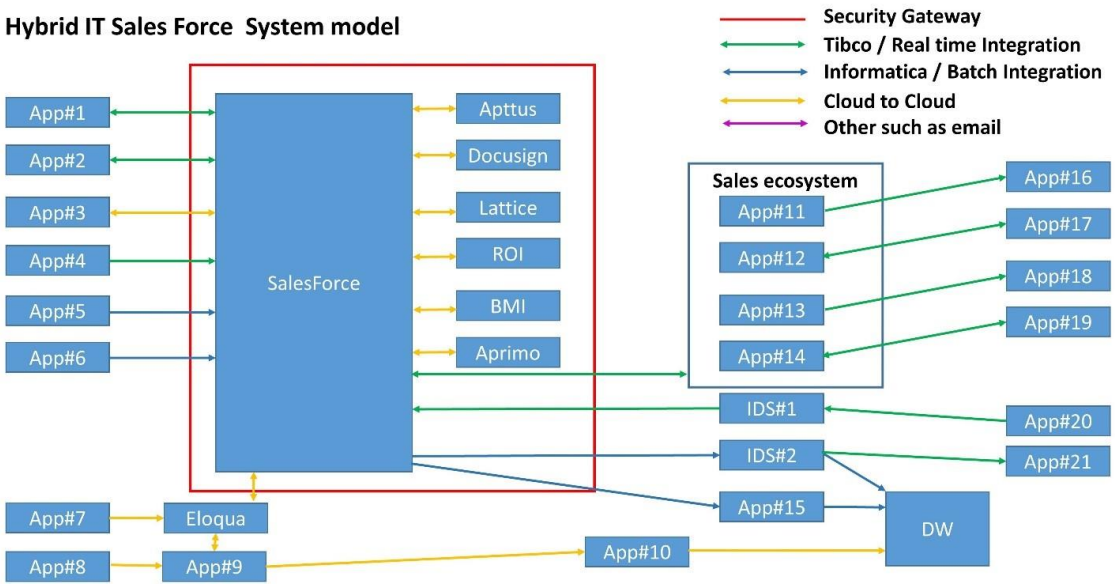

FIG 10: HYBRID WORKLOAD EXAMPLE

It is important to determine where each workload in the Hybrid Application Architecture is best facilitated in order to address location and integration challenges. Determining where each workload in a Hybrid Application Architecture is best hosted requires considering several challenges.

Challenge 1: How do you decide where an application workload should be placed?

The first challenge in developing a Hybrid Application workload architecture is deciding where the component workloads should be placed. The first step is to classify the workload - for example: IT resources with a utilization that grows or shrinks constantly over time experience "Continuously Changing Workload". By identifying this pattern, one can apply the correct architecture to the deployment model. Determining which workloads should remain in a traditional data center and which workloads should be migrated to a Cloud Service requires some in-depth analysis.

Several key areas of analysis must be considered when making these decisions. Here are a few:

- Location, quantity, and capacity of data movement.

- Workload Patterns describe the regular workload experienced by applications over a defined period [Ref1.2].
- Workloads Static as a process describes fixed profiles of workload experienced by applications under specific scenarios. The service can be static, periodic, once in a lifetime, unpredictable, continuously changing, or high growth. For example – what does the workload look like at start of month logon.
- Deployment models employed by cloud providers and the different deployment options for clouds need to be understood in order to architect the right solution.
- WAN demands.
- Seasonal or project-oriented capacity demands.
- Security.
- Country and jurisdictional regulations.
- The speed of new service development requirements.

A great many voices are calling out saying "place your services on our SaaS service" or "develop your application on our platform services." Others cry out "utilize our Infrastructure as a service, and we will dynamically increase the supply of computing power as your application service requires it."

All these options are viable, and many may be an excellent solution to your enterprises demand. However, there are many areas of impact on an application architecture that the placement of a workload can dramatically impact. This impact could be mitigated by identifying the "Application Workload Pattern". For example: the workload describes what the application is experiencing. It then uses this to measure the form of application utilization (e.g., the number of seasonal requests, server load, etc.) This is a good example of how an application experiencing varying workload can benefit from cloud computing.

Change #1: Implement three key KPI's to determine the impact of workload placement

We believe there are three key performance indicators to help determine the best location for workload placement. They are:

- **Application performance**: The performance of the application service from the user's perspective. If an application is slow from a user's point of view, it is now considered down.
- **Response time to real-time decision making**: The ability of the application to make a real-time decision based on the data being collected. For example: the sun is out, and it's a holiday - bring on the ice-cream vendors.
- **Volume of data**: All data must move over networks, the more data, the slower the performance, the higher the networking costs requiring higher capacity to meet demands.

Using these key KPI's will simplify the decision process of application placement.

Challenge 2: How do you bring consistency to the application development process?

With so many ways to build an application stack, it is now possible for each development group to develop the same functions using entirely different approaches, tools, and SaaS services. The result will be extra costs and support services. There needs to be a solution to standardize the method and develop a standardized set of building blocks that the developers can utilize. Consistency in development applies to all aspects of development (e.g., code style, comments, tools, onboarding, creation of new services, monitoring and management etc.) It also extends to product management, defining and tracking tasks, and using the same method that should produce similar results. This is one of the most important aspects to be taken into account in the measurement methods of the software. For more information on how to bring consistency to the development process read more on the Joint Information Environment (JIE) [Ref1.3].

Challenge 3: Don't start thinking about application workload placement when you are in the Technical view.

The requirements of workload placement must be first defined in business and functional views. The technical view further defines requirements such as the required response time (mobile access) and the decision timeframe. Unfortunately, all too often a technical decision such as: "We are going to use this SaaS supplier" is made before any requirements are actually defined. This can often lead to user dissatisfaction with a new system.

The historical approach of Application Workload Management is for each component in the architecture to be managed independently. Unfortunately, this approach will not allow us to achieve the new service levels we now must deliver. We need to control/monitor the service as

a whole and integrate the Application Workload Management processes, responsibilities, and architecture into one integrated system. To accomplish this, we need to make changes in the following three specific areas:

1. Hybrid Application Workload Management processes and roles
2. Application architecture patterns of use
3. Hybrid Application Workload Management architecture

1. Hybrid Application Workload Management processes and roles

The following graphic outlines the work, roles, and OACA domains that Application Workload Management involves. The overall responsibilities of the Shared responsibly model are as follows:

- Facilitate integration of application delivery options to meet user workloads
- Provision integrated platform as a service to power users, citizen integrators, and project developers
- Provision of data and information services
- Provide integration fabric that reduces complexity, enables self-service, and integration reuse

The Functions/activities required to accomplish this include:

- Facilitate workload analysis for optimized sourcing
- Integrate SaaS with enterprise workloads
- Managing API and providing Artificial Intelligence
- Provide local data storage for SaaS
- Modernize workloads for agility and cost-effectiveness
- Ensure compliance to enterprise security policies
- Enable the ability to scale up and down infrastructure and provision applications

Change #2: Implement a four view approach to application workload

Each enterprise is unique; constructed of unique organizational structures, products, services, operating environment, and technology infrastructure. Decisions for moving business functions, services, and applications to the cloud are unique to each individual enterprise. Yet, as more and more enterprises move a business to the cloud, a general pattern for progressing through this change has emerged.

We do this by employing four viewpoints, delivered as a series of stages, in forming a strategy for hosting a function or a component thereof in the cloud:

- Business View
- Functional View
- Technical View
- Implementation View

Historically, these strategies were developed independently. However, they now need to be integrated into one unified force that looks at the service as a whole. It starts by all agreeing that service experience needs to be measured at the user's screen and not on the individual application parts. The overall Application Service Manager is responsible for coordinating all parties in this ecosystem.

Appendix A of the "ODCA Digital Transformation to Cloud adoption considerations and methodology" Ref 5.1 contains a set of questions that need to be considered in each viewpoint. These questions have been compiled based on the experience of many enterprises that have implemented cloud solutions. They were asked, "what had they wished they had known before they committed to a cloud solution or provider." We recommended that you answer each of these questions as you work through your transition to a Cloud Operating model.

METHOD

There are a series of steps which flesh out these viewpoints, possibly as a multi-pass process, represented generically as follows:

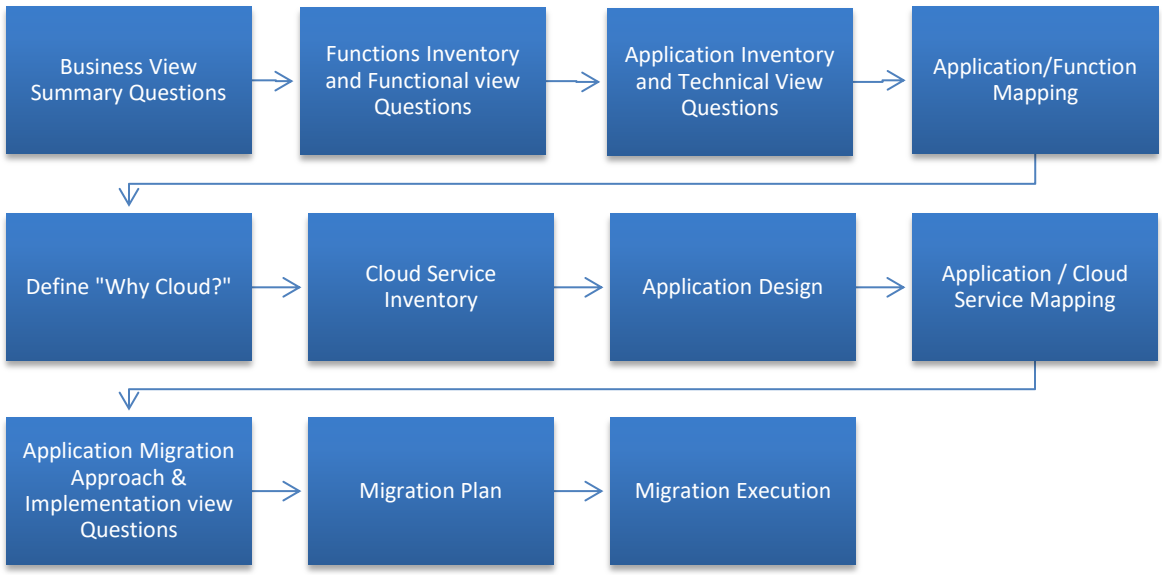

FIG 11: HYBRID WORKLOAD PLACEMENT PROCESS

BUSINESS VIEW

Business View Summary Questions

- The key to a successful move starts with understanding the business of an enterprise. Identify the **vision of the company, its business model, strategy, and objectives** to guide the teams' decision making. Without a foundation in these objectives, it will be difficult to move to the second step. Complete the Appendix A Business view questions in the "ODCA Digital Transformation to Cloud adoption considerations and CMM methodology" [Ref 5.1]

FUNCTIONAL VIEW

Functions Inventory and Functional view questions

- The second step is a documented **understanding of the functions within the enterprise** that deliver the business capability. Leverage the OACA Cloud Maturity Model (CMM) in

this step to consider the Enterprise capabilities per Domain. Complete the Appendix A Functional view questions

TECHNICAL VIEW

Application Inventory & Technical view questions

- The third step is to **inventory the applications and technology services** across the enterprise. The CMDB is the best place to start. It should contain all the applications and relationships. Starting with an application inventory, applications should be grouped into classes based on size, complexity, dependencies, and technology platforms. The OACA CMM is a useful tool for analyzing the current state of (for example) the Data, IT Architecture, and Infrastructure environments. Complete the Appendix A Technical view questions

Application/Function Mapping

- Once all applications have been assessed, **applications should be mapped to the functions** they support.

Why Cloud

- Now ask "Why Cloud?"
  1. Assess why a cloud deployment would be beneficial for that application / asset group / asset portfolio, what considerations to list (such as speed to value, IT simplicity, or currency, primarily leveraging the innovation or scale in the marketplace, for availability requirements, or financial drivers/ business scaling elements).
  2. Not all apps are suitable for cloud deployment or can leverage the benefits offered by a service-based model. Based on the characteristics of the application, use those to complete a rough estimate of whether the application should move to the cloud or remain on-premises, whether there is a clear-cut provider choice, and whether the application would likely need to be refactored or not (e.g., old technology, does adequate technical documentation exist, are there SME's still at the company, etc.). Once "why cloud" is answered, we can assess the "how to adopt cloud" question – worrying about architecture, design for failure, cloud-aware architecture, licensing, etc.

Cloud Service Inventory

- Having completed the business, functional, and application assessments, the analysis turns to the cloud provider(s) and whether the provider(s) is/are external to the enterprise or not. An **understanding of the cloud provider(s) products and services** is critical for mapping applications and services to the optimal cloud provider services. It is important to keep in mind that there are now two development teams working: the provider team and the consumer team. It is important that they are kept in sync, else one can build over the other.

Application Design

- The sixth step is to work through each application in the inventory to **document its design**; ensuring that each application's architecture, service dependencies, technology platforms, risk and compliance posture, and operational support requirements are documented.

Application / Cloud Service Mapping

- With this detail in hand, decisions for matching a given application to a set of cloud provider services follows.

IMPLEMENTATION VIEW

Application Migration Approach

- The next step in the process is to **determine the migration approach** for each application. An application's value to the company should be weighed against the effort, risk, and complexity of rewriting the app to take full advantage of the cloud. In some cases, a lift and shift approach, merely moving the application as-is to the cloud, is the best balance of risk/reward (even if this approach minimizes the value of hosting the application in the cloud). In other cases, if the application's value to the company is sufficient and the risk posture low enough, it may be viable to support refactoring or re-writing the app to take full advantage of the cloud platform. The OACA Cloud Maturity Model (CMM) should be used in this step, to identify the impact on all the domains of implementing the new application ecosystem and to identify the additional work that will be required in those domains to successfully implement the operating model to support the cloud application ecosystem. Complete the Appendix A Implementation view questions

Migration Plan

- This analysis should be completed for each application in the application inventory. Having classified each application for its potential move to the cloud, appropriate project management processes may then be applied to develop a migration plan and ultimately deliver the migration of selected applications to selected clouds.

Migration execution

- According to the migration plan, migration from selected applications to selected cloud services should be executed step by step. Eventually, the business function, which is used to be supported by a selected application, will be hosted by the new cloud services. This transfers the ongoing business function transformation from on-premises to the cloud.

2. Application Architecture Patterns of Use

There are many ways to deliver services today. Two issues need to be addressed to ensure that the best application architecture is implemented.

Issue #1: Favorite way: Developers become comfortable developing with specific tools. Sometimes developers do not want to try to use new tools or approaches and prefer to build applications using their favorite way. This is typical response; however, it is an impediment to adopting new technology and preventing accumulation of technical debt.

Change #3: Implement a compensation system that rewards new technology adoption.

There are two ways to overcome this resistance to change. The first is to implement a compensation program that rewards those will adopt and recommend new technologies and deployment approaches.

Change #4: Implement Design Review Panel.

The second way is to implement a design review process step that requires all new applications be reviewed by a panel of developers to determine the best way to design the new app. This will minimize the "favorite way" pitfall.

Change #5: Implement Standardized Patterns of use.

Issue#2: Lack of knowledge or multiple variations to the design approach. Lack of knowledge is merely a description for when a developer doesn't know how to implement an application using new technologies. Multiple variations are caused by each developer developing an application using a different set of tools. Both of these issues are addressable using standardized patterns of use.

Below is an example of a standardized pattern of use that explains how an application should be designed for a specific business category of applications

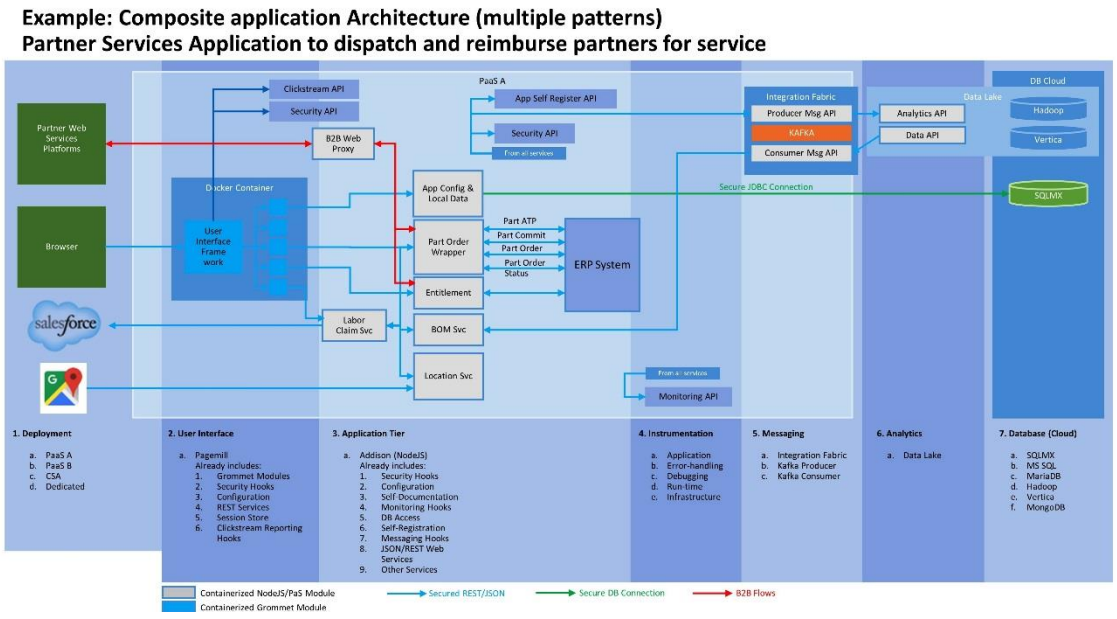

FIG 12: HYBRID WORLKLOAD EXAMPLE

3. Hybrid Application Workload Management architecture

The next area of change is the technical architecture of the application workload.

Below is an example of a technical architecture that reinforces a common Application Workload Management Architecture. This architecture looks at an application service as a whole, not just the individual components.

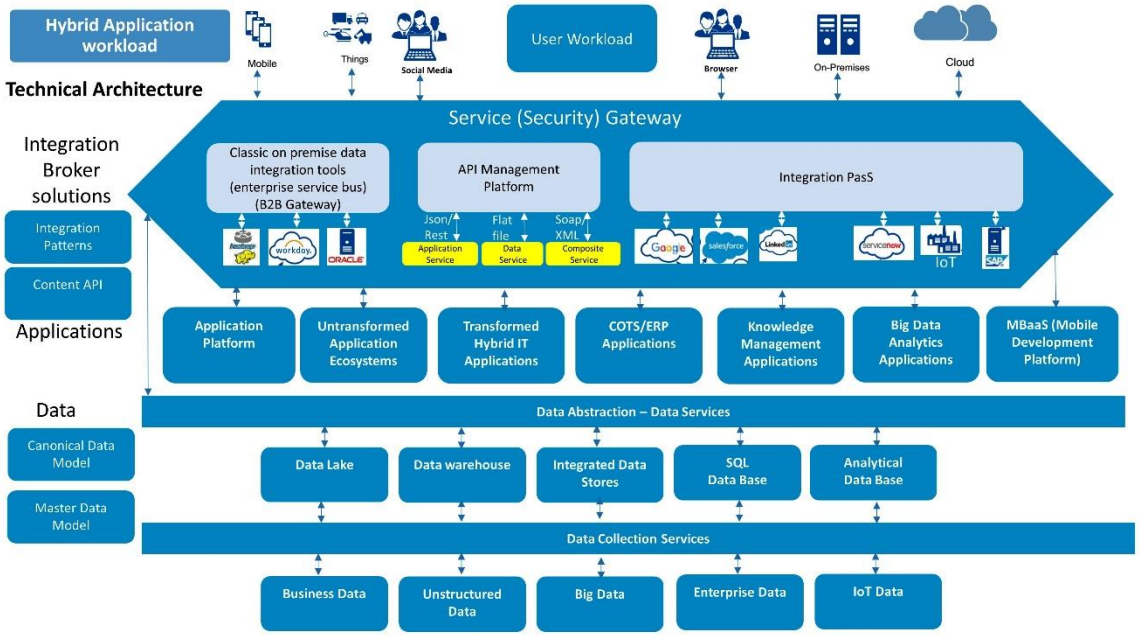

FIG 13: HYBRID WORKLOAD ARCHITECTURE

Change #6: Publish a Standardized catalog of Cloud Services.

This architecture defines the standardized technologies that a developer can choose from to develop their new applications. In the manufacturing industry, the cost is driven down by publishing the standardized parts catalog that a designer can build a product from. This approach helps eliminate unnecessary variation in the complexity of the new product and helps purchasing increase their buying power. This same discipline needs to be implemented in IT. Using this approach, the overall application architecture will be more straightforward, and purchasing will have to ability to negotiate a better deal with Cloud providers.

Close

The transformation to a Hybrid shared resource Application Workload Management model requires new skills, new methods, new process steps, and artifacts to be produced by an IT organization. If done well, then an organization can rapidly exploit new technologies and bring new capability to market much faster than their competitors.

3 - HYBRID DEVOPS

DevOps is a framework that enables development, quality assurance, and operations to meet customer needs. A key dimension of DevOps is the move away from a project-based focus, towards a product- or service-based focus and culture, with the DevOps team taking an overall view and responsibility from end to end. It contains capabilities related to:

- Integrating Development and Operations teams to facilitate communication, collaboration, and integration to manage today's rapidly changing business demands.
- Enabling developers to provision, change, and manage their development environments without IT operations involvement.
- Allowing developers to promote cloud-native applications to production without IT Operations involvement.
- Enableing both conventional application development acceleration and cloud-native application development techniques.

Benefits of DevOps

- Speed of release of new applications.
- Higher quality applications released to production.
- Having a defined (common) cloud architecture that aligns both operations and development teams with business requirements and objectives, is key to leveraging the native capabilities and opportunities that are inherent in cloud technologies.
- When DevOps Teams have defined process goals that align with business requirements and demands, then the realization of business agility and performance can be achieved.

DevOps is about streamlining development and optimizing operations to enhance service delivery while decreasing the time it takes to design, build, deploy, and support products and services. In the traditional "on-premise" environment, the roles and responsibilities of development teams and operations personnel did not so much change as combine to achieve the realization of DevOps. When this is applied to "Cloud," it becomes more challenging due to the variation of consumption models, i.e., SaaS, PaaS, IaaS, and FaaS. Each type will represent a

different development and support model. Throw into this the Hybrid environment, and roles/responsibilities could become blurred. As businesses struggle to find the right fit for their workloads, IT is also struggling to align with this new paradigm. Best practices are still being defined, and configurations are usually being optimized manually in response to overages or (worse) security breaches.

The Hybrid model is unique in that it can be applied to both the traditional on-premise environment as well as the off-premise on-demand environment. Using this model will not only enable better alignment of roles but will ensure that both environments are included in the IT ecosystem. When implementing DevOps in a hybrid scenario, actors and roles must also be considered closely, and parties need to be aligned to the business objectives, timelines, and concepts of Hybrid IT to fully leverage the capabilities of DevOps.

Example: The on-premise DevOps team traditionally deployed a logging solution with "debugging" turned on. This typically resulted in a lot of logging data and was historically parsed and distributed across the on-premise logging storage environment which was closely monitored by the hardware vendor's proprietary tools. The "business" had decided to use storage in the cloud as part of its logging solution and received a bill 10 times the expected amount. This was due to the additional data the enabled debugging produced. It is important to define roles and responsibilities regarding how resources are consumed and utilized across both the cloud and the traditional workloads. Cross-organizational transformation needs to occur from being a basic "IT operations department" into a "cloud consultation team."

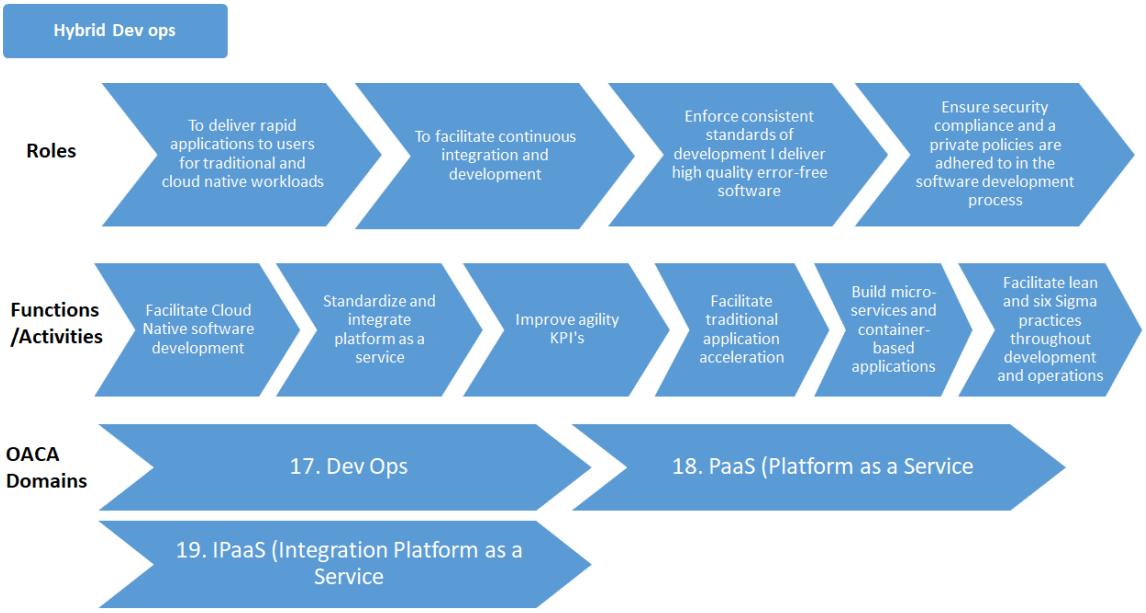

FIG 14: HYBRID DEVEOPS COMPONENTS

The historical approach of Software Development is that Development, Quality assurance, and Operations operated separately. Development would develop a solution and throw it over the wall to Quality Assurance, who would test it and then throw it back to Development. Eventually, Quality Assurance would authorize it for release and throw over the wall to Operations. The result of this model was that Operations needed to learn the application and often found out things in production that impacted the availability, reliability, and performance of the application, resulting in extensive change requests and reliability issues with the first releases of the applications. All of this resulted in long delays to release new applications and reliability issues in production.

SaaS and Cloud services challenged this entire model by enabling a business developer to order services directly, and not involve operations or Quality Assurance. The result was much faster release to production of new capability, (but often without governance and security oversight). With the introduction of cloud-native application services, release times could also be radically changed. New code could be deployed in seconds and a developer could make multiple releases in one day.

HYBRID DEVOPS PROCESSES AND ROLES

The Hybrid Devops process is a key enabler (if done properly) for dealing with both the on-premise traditional environment as well as the off-premise on-demand environment. Leveraging this approach will not only enable better alignment of roles, but also ensure that the relevant environments are properly included in the IT ecosystem. When implementing DevOps in a Hybrid environment, all relevant actors and roles must be considered closely. Leveraging smaller parallel DevOps teams helps to align the specific components more directly to the business objectives. Small well-functioning DevOps teams also enhance timelines for development, and enable the concepts of Hybrid IT, in order to fully leverage the capabilities of Cloud.

HYBRID DEVOPS DOMAINS & ARCHITECTURE

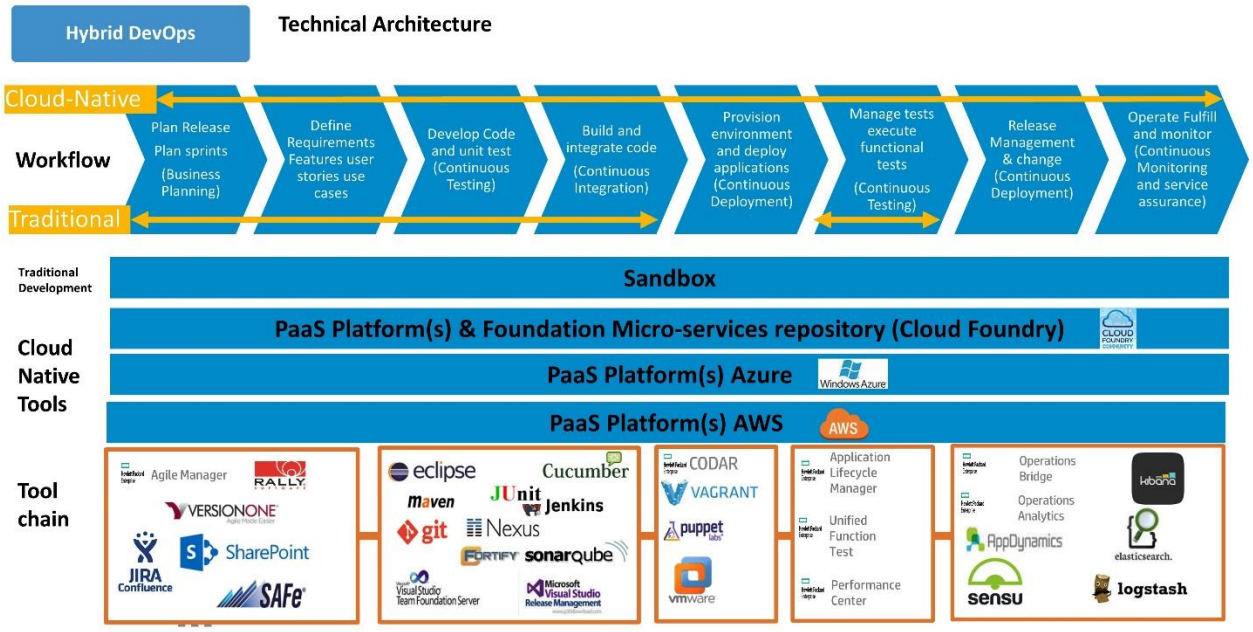

FIG 15: HYBRID DEVOPS ARCHITECTURE

Traditional application development versus cloud-native.

The above graphic describes the eight major processes in a software development process. Traditional application development has four groups of individuals who are involved in the release to production of an application. The Development group is responsible for the following activities:

- Plan and release sprints
- Define requirements features user stories and use cases
- Develop code and unit testing
- Build and integrate code

Operations is responsible for moving the code to the Quality assurance systems. The Quality assurance group manages the testing process and executes the functional tests. If the tests pass, then the Release management group schedules the release. Operations moves the application to production at the required time and assumes responsibility to monitor and operate the code. In some companies, if databases are necessary, there is a fifth group (database administrators) involved in the provisioning of databases to the development group.

All of these different groups and handoffs add more and more time to the provisioning of the service.

In contrast, in a cloud-native application development approach, a developer is responsible for conducting all eight processes and promotes the release to production under their own approval. The result is a much faster release to production process.

DevOps is designed to equip IT to release new capability rapidly. There are three key approaches to accomplishing this goal.

- Traditional application acceleration
- Native cloud applications
- Cloud native application design

Each of these strategies requires a significant change to the current development and operations relationship and operating model.

Change #1: Traditional Application Acceleration

In most large companies the majority of applications are still implemented using the waterfall software development methodology as described below.

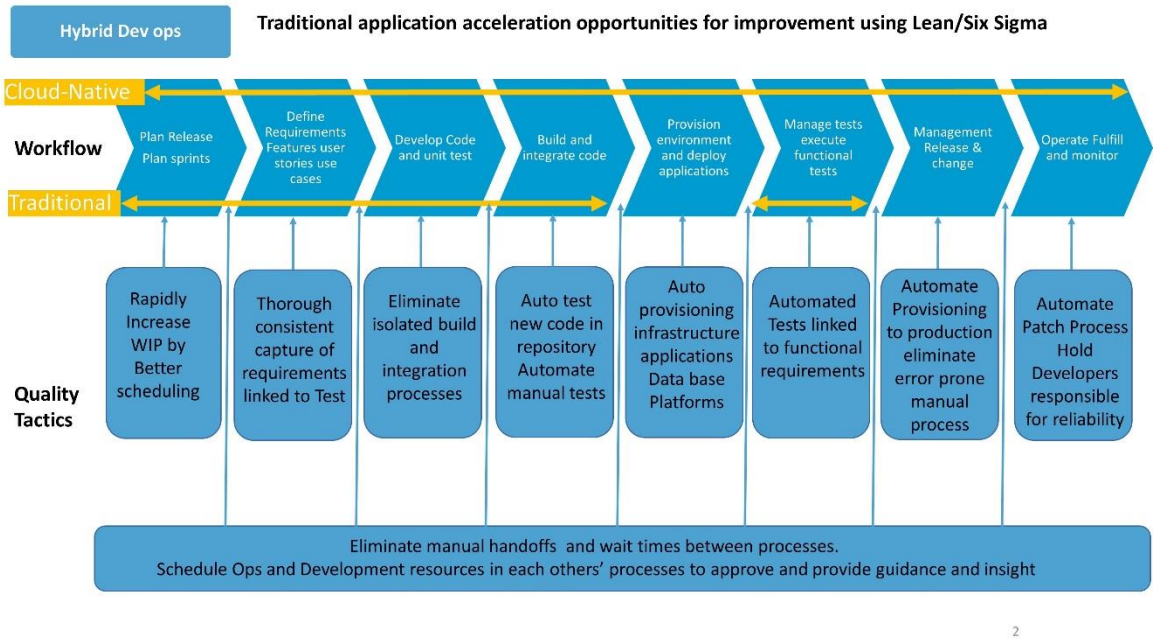

FIG 16: HYBRID DEVOPS PROCESS

Using total quality control techniques companies are now applying Six Sigma methods to their software development process. This approach is enabling these companies to eliminate the excess time and handoff issues between the different groups. It requires Application development, DBA's, Quality Assurance, and operations groups to work closely together to identify all non-value-added activities, eliminate delays, and compress cycle time of events. The above graphic describes the quality improvement tactics that can be used.

Example Company # 1

Using the quality improvement approach Company #1 identified that the number one delay in provisioning new systems to production was the provisioning of databases. DBAs were taking up to nine weeks to provision databases to the development groups. They developed a private cloud the delivered standardized databases for Microsoft SQL, Oracle, Mongo, and Cassandra and reduced the time to provide a new Database from nine weeks to two minutes.

Example Company #2

Company #2 wanted to solve application release to production problems. Of the 150 Applications deployed in 2014, none went right in the first run. There were a lot of manual steps in the deployment process. There was a lack of quality in the deployment process, every deployment had a different end result, and the duration of the implementation was taking too long. To solve this problem, they focused on the two most critical business-critical applications of the company and implemented Continuous Delivery using a continuous deployment solution that provided automation and release management of complex multi-tier applications across the application lifecycle. The results of this approach were:

- The three-month manual preparation process was compressed to twenty minutes.
- Deployment time of an application dropped from five hours to two and a half minutes.
- Multiple deployments were done per day. Deployment results are always the same, and there were no incidents in production anymore.
- Overall results: Deployment time increased 228x faster, improved quality, better integration development, and Quality Assurance.

Change #2: Native Cloud Application Development

"A Native Cloud Application (NCA) is a program that is designed specifically for a cloud computing architecture. Native Cloud applications are developed to take advantage of cloud computing frameworks, which are composed of loosely-coupled cloud services. That means that developers must break down tasks into separate services that can run on several servers in different locations. Because the infrastructure that supports a Native Cloud app does not run locally, NCAs must be planned with redundancy in mind so the application can withstand equipment failure and be able to remap service endpoints automatically should hardware fail.

The design paradigm is cost-effective; however, because services and resources for computation and storage can be scaled out horizontally as needed, which negates the need for overprovisioning hardware and having to plan for load balancing. Virtual servers can quickly be added for testing and, in theory, a Native Cloud App (NCA) can be brought to market on the

same day it's created. In general, a native app is an application program that has been developed for use on a particular platform or device [Ref3.1].

Example Company #3

Company #3 had a new product that was to be released, and the marketing team felt the web experience was not compelling enough to reflect the image of the new product. It was decided that an entirely new web presence was needed to reflect the creativity of the new product.

A new website platform was built on their private cloud. Apache and MariaDB were chosen to deliver as the application stack. The Private cloud delivered the underlying infrastructure and platforms for development, production, and test environments. The time to develop the new system and provision the infrastructure and application stack was 48 hours.

Example Company #4

After the Madoff scandal, the contact volume into the Office of Investor Education and Advocacy increased to over 90,000 contacts annually, and the decade-old system for managing contacts was incapable of keeping up. They decided to move to a cloud solution that allowed representatives around the country access to all documents in real time. The results of this change were

- Reduced a 30-day response to investors to less than 7 business days.
- Created a completely paperless system for handling investor inquiries.
- Provided staff with contact history to better serve consumers.
- Reduced timeline for system configuration from months to minutes [Ref 3.2].

Change #3: Cloud Native Application Development

Cloud-native computing takes advantage of many modern techniques, including PaaS, multi-cloud, microservices, agile methodology, containers, CI/CD, and DevOps. This approach requires a significant change in Application design disciplines. This approach allows a developer to promote directly to production. There are no gateways to ensure that what they release will not cause production problems. As a result, the number one change required to successfully adopt this approach is to make developers hold equal accountability for production reliability.

Cloud native design requires a developer to follow the 12 factor application development disciplines. "The twelve-factor app is a methodology for building software-as-a-service apps that:

- Use declarative formats for setup automation, to minimize time and cost for new developers joining the project;
- Have a clean contract with the underlying operating system, offering maximum portability between execution environments;
- Are suitable for deployment on modern cloud platforms, obviating the need for dedicated servers and systems administration;
- Minimize divergence between development and production, enabling continuous deployment for maximum agility; and
- Can scale up without significant changes to tooling, architecture, or development practices" [Ref 3.3].

The 12 factors are:

1. Codebase: One codebase tracked in revision control, many deploys
2. Dependencies: Explicitly declare and isolate dependencies
3. Config: Store config in the environment
4. Backing services: Treat backing services as attached resources
5. Build, release, run: Strictly separate build and run stages
6. Processes: Execute the app as one or more stateless processes
7. Port binding: Export services via port binding
8. Concurrency: Scale-out via the process model
9. Disposability: Maximize robustness with fast startup and graceful shutdown
10. Dev/prod parity: Keep development, staging, and production as similar as possible
11. Logs: Treat logs as event streams

Admin processes: Run admin/managemehttps://searchitoperations.techtarget.com/definition/native-cloud-application-NCA

[ii] https://cio.gov/public-sector-cloud-computing-case-study-securities-and-exchange-commission/

[iii] https://12factor.net/

[iv] https://12factor.net/

12. nt tasks as one-off processes [Ref 3.4].

Company #5 example:

Company #5 had a .NET application that was over ten years old. It handled 300k transactions/day. It was not reliable and did not meet the company's current needs. They built the replacement application using cloud-native disciplines. Here is a summary of their approach:

- Developed a Cloud Native application on two Stackato development platforms using Docker containers.
- Developed three micro-services,
- Implemented weekly releases,
- Implemented auto scaling,
- Implemented a Continuous Delivery Pipeline development process that supports the entire system, not just the application source code, and
- Implemented Infrastructure as code on their private cloud

The results of this initiative were:

- Demo app ready in 11 days! Minimum viable product (MVP) in 7 weeks,
- New UX in 4 days, Alpha in 4 sprints, Beta in 5 sprints
- 454 builds and deploys to Stackato
- 250,000 users worldwide

Close

The transformation to a shared responsibility DevOps model is one of the most challenging activities a technology organization has to achieve due to the magnitude of the changes in process, people, and technology. It requires the adoption of new technologies, new methods, and new people measures. There are however two significant issues in DevOps implementations. The first is people related, the second is the current change and release process.

Resistance to change and fear from Operations that developers will release poor code is the number one problem. To address this reality, two changes have to be implemented

They are:

- The measurement system on the development community has to change from release to production time, to quality of software in production.
- The change and release management process needs to be modified to allow a developer to self-authorize release to production.

DevOps is a critical discipline for all IT environments to adopt. All enterprises require IT to operate much faster. DevOps enables this requirement.

4 - HYBRID SERVICE MANAGEMENT

A Cloud service is rarely implemented as one end-to-end service. It usually needs to be integrated into the existing technology architecture and operating model. This results in different organizations being responsible to manage the individual component services, each to a specific service level. In addition to this change, overall service level expectations have also changed.

For example, the only acceptable service level today for an Application service is:

- 100% available
- 100% reliable
- Always fast, measured at the screen of the user, no matter the device.
- Always secure

The challenge we face today is how to manage people, processes, and technologies to achieve this service level in a Cloud/Hybrid IT environment when an application service is delivered through multiple partners, providers, and technologies. The compound impact on the reliability

of a service as a whole increases with each additional component or service provider added to the service. For example, assume an application service consists of the following elements:

- Mobile application
- Mobile device
- Mobile carrier
- Mobile network
- Data Center network
- Application Server
- Database Server

There are seven components in this system. Let's assume the reliability of each element is 99.9%. 99.9% to the seventh power is 99.3% or 61 hr. & 21 min. of downtime per year.

Furthermore, application performance has now become even more critical to manage. Systems are considered **down** if they get **slow**. Now again consider a Hybrid IT/Cloud application service with multiple service providers that are all contributing to variations in performance. Figuring out why service is slow is virtually impossible in the way IT is traditionally managed.

Areas where change in Service management is required

The historical approach of Service Management is each component in the architecture is managed independently. Unfortunately, this approach will not allow us to achieve the new service levels we now must deliver. We need to control/monitor the service as a whole and integrate the Service Management processes, Service Management responsibilities, and Service Monitoring technologies into one integrated system. To accomplish this, we need to make changes in the following three specific areas:

1. Service Management processes and roles
2. People performance measurement and compensation systems
3. Service monitoring tools

1. HYBRID SERVICE MANAGEMENT PROCESSES AND ROLES

The following graphic outlines the work, roles, and OACA domains that service management involves. The overall responsibilities of the Hybrid IT Shared responsibly model are as follows:

- Ensure security of applications infrastructure data facilities and networks
- Facilitate unified demand management
- Facilitate unified service level management
- Facilitate information lifecycle management
- Facilitate efficient use of assets

The work areas required to accomplish this include:

- Security management
- Demand management
- Service management
- Operations management
- Information lifecycle management

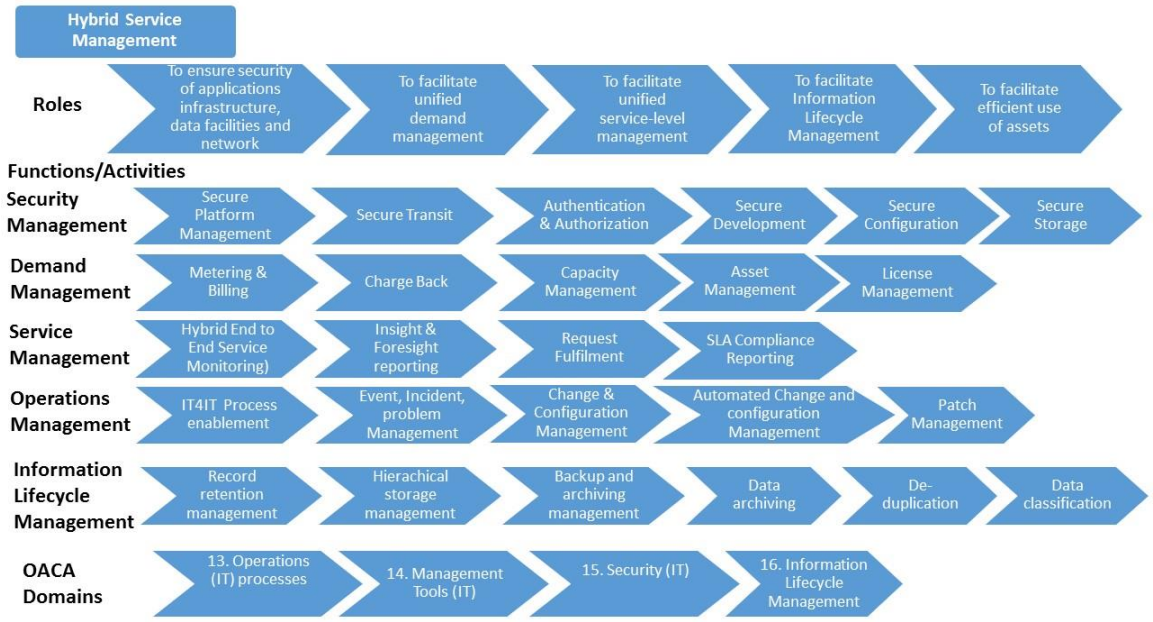

FIG 17: HYBRID SERVICE MANAGEMENT COMPONENTS

Change #1: Service Manager responsible for application service as a whole

Historically these groups operated independently. However, they now need to be integrated into one unified force that looks at the service as a whole. It starts by all agreeing that services need to be measured at the user's screen, and not at the individual service elements.

In the new model, a Service Manager needs to be responsible for the service measured at the user's point of view. The SLA should also be developed from the customer's point of view. Generally, an SLA will be related to the application's performance, or the response time of resolving an incident on that application, even though the service is dependent on several cloud or underlying service providers.

To appreciate the magnitude of this change, consider our previous application example. A Service Manager now needs to negotiate and understand the service level requirements for all of these components of the service.

- Mobile application
- Mobile device
- Mobile carrier
- Mobile network
- Data Center network
- Application Server
- Database Server

The Service Manager also needs to forecast the demand of the new application service and the impact on all underlying components to ensure that sufficient supply is proactively in place as the overall service demand increases.

Change #2: Service Level Agreements based on application performance

Next, the service level agreements need to be rewritten to reflect the new measurement criteria. Each application needs to be measured by the performance, availability, and reliability expectations measured at the screen of a user.

It is recommended that a set of key transactions that a user needs be the basis of the SLA, and that the Response Time performance, Service Availability, and Service Reliability be defined and included in the SLA. This will enable the service manager to have a common measurement test to understand the impact on each of the underlying components in the application's architecture. This may seem simple enough except for the fact that some cloud providers do not guarantee the performance of their services. They often only provide an SLA on uptime, so it may not be possible for a Service Manager to develop a performance service level measure in the SLA.

This Cloud Service provider fact may require the Service manager to recommend that a Cloud Provider not be used and that the service be delivered using an alternative method, such as a dedicated server in a company's data center or rebuilt using a scale out architecture. A service cannot achieve a service level if it is not initially designed to meet that service level. A Service Manager needs to detect issues with the proposed service architecture and identify where changes need to be implemented, before it goes into production.

Service performance is an Achilles heel for many cloud providers. A Service Manager must deeply understand the limitations of all cloud providers used in their service.

It is also the responsibility of the Service Manager to set the expectations of the service as a whole for a customer in their SLA.

2. PEOPLE PERFORMANCE MEASUREMENT AND COMPENSATION SYSTEMS

A primary challenge in many IT organizations is that many IT people work in silos, and they have never been responsible for delivering an end-to-end service. Technology personnel need to start thinking like a cell phone provider or electricity provider. Everyone knows that electricity's

service level is measured at the user's "switch", and it is either working or not. When the power goes out, the transmission line department doesn't say it's not their problem because the transmission lines are performing at 100%. Instead, the electricity provider knows the blackout area immediately and how many people are impacted and can triage/troubleshoot the incident down to the component level.

Once the failed component is repaired, the electricity is not turned on for everyone at once because that would merely cause another outage due to power surges. Instead, the electricity provider knows their customer base and first restores power to the portions of the grid that contain critical customers such as hospitals.

Change #3: Transformation of the mindset of Technology Staff from technology centric to service centric and become one unified service management force

In most Technology organizations the measurement and compensations systems reward each group in the area of their responsibility. In the Cloud/Hybrid IT Paradigm, Technology staff should be paid on the performance of the service as a whole and not on their own individual component's performance. All Technology staff also need to be transformed into an integrated unified force that is focused on managing the service as a whole.

The level of collaboration needs to be much more than merely transactional using email or a ticketing system. It needs to be an integrated partnership between groups wholly dependent on one another. Here are some examples of this type of operating model:

- Collaboration between internal/external groups to develop joint or integrated monitoring systems
- Facilitating operational reviews for performance & SLA metrics
- Joint efforts for troubleshooting incidents and analyzing the cause

The service levels need to become the overall measure of all the individuals responsible for the architecture. No longer will it be okay to achieve a 99.9% reliability metric on a network or a database or some component in the architecture and still be deemed as doing the job well. In

the new Shared Responsibility model, if the overall service level measured at the user screen fails, everyone fails.

David Packard of Hewlett Packard fame one time said, “Tell me how a person is measured, and I will tell how they will behave.” This statement is very accurate in the case of service management. If employees are measured only on the performance of their part of the architecture, then they will not be concerned about service management on the service as a whole.

Change #4: Compensation/Reward/Measurement system

The measurement/compensation systems need to be changed to reward the required service levels. This one change alone will bring the highest amount of sensitivity to ensuring the service level measured at the user’s point of view is achieved. All staff (including managers) need to be compensated according to the Application service level achieved, measured at the user screen. Bonuses and compensation should be directly connected to reaching or exceeding the target application service levels.

3.HYBRID SERVICE MANAGEMENT DOMAINS & ARCHITECTURE

The next area of change is to the technical architecture of the service monitoring and management tools. Historically tools have been chosen to enable personnel within a specific functional domain. For example, the networking team chose their tools; the database team chose their tools. The data that was collected from these tools was kept within that domain and not shared widely. In fact, in most cases, if the information is needed in another area, it has to be requested. There is no access to that data outside that group.

Also, in most cases, the data is not used to do predictive analytics. To be able to achieve a 100% reliability service level the service management team needs to have the ability to foresee incidents before they occur. This capability change will require a change in the way the data is collected, integrated, and analyzed to be able to achieve service levels.

These demands will also require a significant change in the willingness of the different groups to move to:

- One common service management architecture,
- One common service management database, and
- One common set of service management analytic tools

Change #5: Common System Management Architecture

Below is an example of a technical architecture that reinforces a common Service Management Architecture. The tools are designed to monitor and manage the service as a whole, not just the individual components.

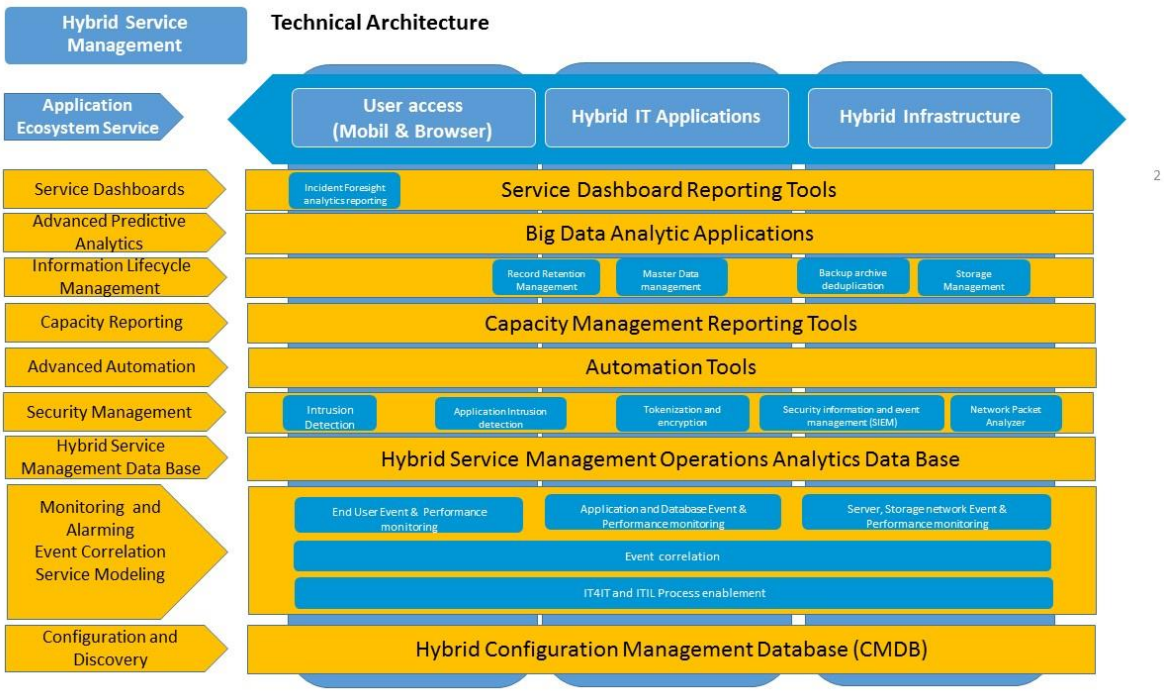

FIG 18: HYBRID SERVICE MANAGEMENT ARCHITECTURE

In most firms, this will require a significant overhaul of the current system management architecture to accomplish this result. Technology will also need to integrate internal service management systems with cloud providers.

Most cloud providers will not provide service management information on their internal systems, so the new service management architecture needs to be able to detect performance, availability, and reliability of cloud providers by treating them as black boxes. This will require the addition of monitoring points and tools aimed at each cloud provider's services, measuring the transactions delivered by that provider.

Change #6: Common Data Model and Predictive Analytics

Historically a System Management tool stores data within the Tools repository. In the new model, all System Management tool data is integrated into one database and analyzed by one analytic toolset. An essential addition to this new architecture is the data analytics applications that will allow the service management team to be able to do advanced predictive analytics and foresee incidents before they occur. This is often largely based on log file monitoring and analysis, for application level transactions. In hybrid clouds, this aspect of monitoring data increases extensively!

5 - HYBRID INFRASTRUCTURE

When people think of infrastructure, they generally think about the fundamental facilities that serve the country such as roads, bridges, water systems, electrical grids, etc. Without a well developed and managed infrastructure, a modern society cannot flourish. Even the Roman Empire understood the criticality of developing a solid, well-managed infrastructure. There is evidence of their roads and water systems still in operation in Europe today.

IT infrastructure has now moved into the role of being a critical facility for any modern society. Without a well-managed IT infrastructure, a modern society cannot operate. Commerce cannot happen, communication cannot be facilitated, though services can be delivered.

There are significant challenges to delivering an IT infrastructure, however. The first and foremost is security. One global corporation stated that they need to deal with 23 billion security events monthly and employ 5,000 security professionals [Ref 4.1]. It is estimated that cybercrime generated a profit of $1.5 trillion in 2018 [Ref 4.2].

To further complicate the challenge, IT infrastructure is no longer delivered from individual data centers but is now a compilation of services delivered from multiple cloud providers and traditional data centers. Furthermore, IT infrastructure is no longer controlled solely by operations personnel who follow strict ITIL disciplines. It can be provisioned directly by software developers using infrastructure is a code.

Traditionally, IT/Engineering has always been responsible for the deployment of services to meet the needs (both in functionality and in scalability) of the business. In most cases, requests for infrastructure came from other groups within IT/Engineering who acted as a representative of the business units needing the services. These sanitized requests would feed in to long-term planning strategies, allowing IT/Engineering to make strategic investments in a handful of technologies. Managing this technology was made somewhat less complex by the bounded options and features of the selected toolsets. Requests outside the current toolset were fed in to program management activities to be actioned at a later date.

Today, designing a solution infrastructure is more involved. From a feature perspective, toolsets between and within large cloud providers offer similar outcomes with nuanced differences. Many of the public cloud provider offerings are built on top of existing open-source software, further increasing the delivery options. This toolbox approach allows citizen integrators to solve the same problem multiple different ways, balancing the tradeoffs as they see fit. The challenge on IT/Engineering is how to enable similar services from different providers while providing a unified/familiar management and control plane. All this, of course, needs to be handled at cloud speed.

From a scalability perspective, features of cloud providers allow IT to perform better at matching provisioned resources to business demand curves. Choice in the marketplace allows for distributed computing deployments, taking advantages of platform benefits and/or cost benefits as required. Managing how and where resources are deployed is paramount to ensuring appropriate business services are deployed at the best price possible.

As the Hybrid DevOps section discussed, delivering hybrid infrastructure solutions has also changed. With a change in focus from stability to agility, new tools and techniques have emerged that increase coordination within IT/Engineering teams and increase the speed at which deployments can be made. The end goal from a delivery perspective is to provide a

unified toolkit/platform that citizen integrators can use to create desired cloud services that are controlled/secured/scaled appropriately.

These changes have demanded a significant change in the way that a hybrid IT infrastructure is managed and provisioned. The following graphic outlines the roles, functions, and OACA domains that describe Hybrid Infrastructure.

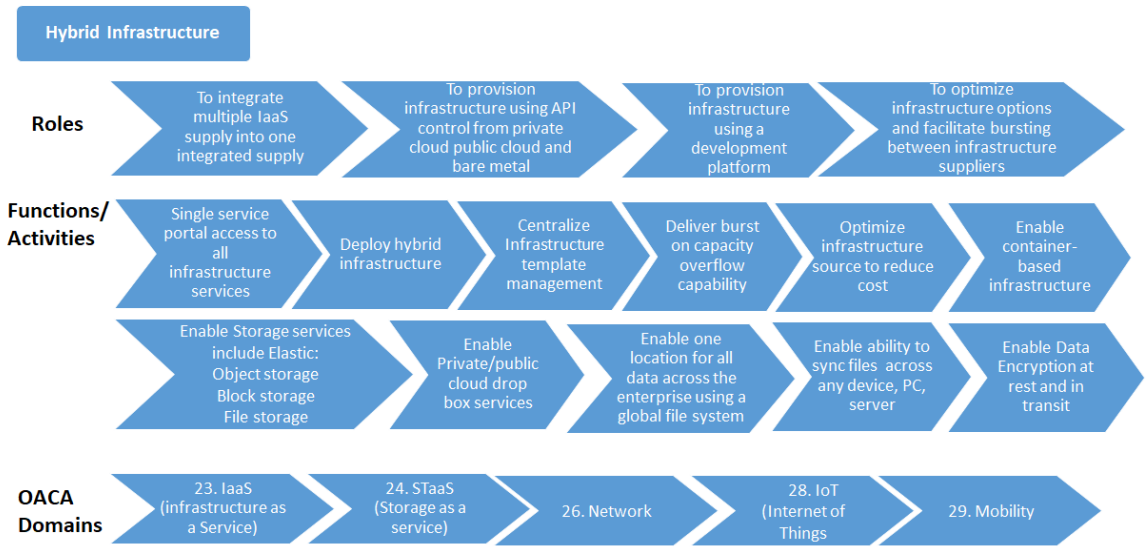

FIG 19: HYBRID INFRASTRUCTURE COMPONENTS

New roles for Hybrid Infrastructure are:

- **Integrated Supply**: To integrate multiple IaaS supply into one integrated supply. IT teams need to integrate services from multiple cloud vendors so that citizen integrators have access to the best tools to solve any given problem.
- **Unified Management:** To manage multiple clouds using a unified view across all management functions/portals.
- **Infrastructure As Code:** To provision infrastructure using API control from private cloud public cloud and bare metal from a development platform using development-like tools and processes allow for various benefits to infrastructure deployments including activities like policy review and testing.
- **Optimized for value/cost:** To meet business demand curves, making use of bursting and cloud cost controls to optimize workload deployments.

Hybrid IT Infrastructure delivers the following functions

- Single service portal access to all infrastructure services
- Deploy hybrid infrastructure
- Centralize Infrastructure template management
- Deliver burstable capacity overflow capability
- Optimize infrastructure sources to reduce cost
- Enable container-based infrastructure
- Enable Storage services including elastic: Object storage, Block storage, File storage
- Enable Private/public cloud drop box services
- Enable one location for all data across the enterprise using a global file system
- Enable ability to sync files across any device, PC, server
- Enable Data Encryption at rest and in transit

Hybrid IT Infrastructure Maturity is defined in the following OACA domains

- 23. IaaS (infrastructure as a Service)
- 24. STaaS (Storage as a service)
- 26. Network
- 28. IoT (Internet of Things
- 29. Mobility

Hybrid IT Infrastructure challenges

There are many challenges facing a hybrid IT infrastructure operating model. We will examine each one of the hybrid IT infrastructure domains and discuss the issues that need to be addressed in the recommendations to the shared services model to address these.

There are however common interface points on all the domains of a Hybrid IT Infrastructure Operating model. These are: User Demand; User Delivery; IT Budget management.

User Demand involves forecasting of the resources that will be required from the hybrid IT infrastructure model. User demand needs to be estimated on all infrastructure resources regardless of their source.

User delivery involves the provisioning of the infrastructure to meet the users' requests. Often users will not request the infrastructure directly but will request an application stack which in and turn requests infrastructure.

IT Budget Management involves capturing the actual cost of Hybrid Infrastructure. This has become very difficult because many Business Units are now just buying Infrastructure services from a cloud provider directly and paying for it out of their own budget.

Change #1: A focus on application infrastructure

Historically, IT was focused on the raw components that would make up a solution. This included elements such as networking, storage, and compute. While some infrastructure related to applications was managed by IT (for example, Exchange and MS SQL) the majority of those components were managed/provisioned by development teams.

With cloud providers providing a toolbox of services (and abstracting away the traditional core components such as compute, storage, etc.), IT teams need to push "up the stack" to understand more about the application infrastructure in aggregate to support the business solutions. This includes services such as

- Managed web applications
- Managed service bus
- Managed data services (SQL, NoSQL, etc.)

Depending on the target cloud platform (IaaS/PaaS/FaaS), the amount that an IT/Engineering team may need to push up the stack will vary. For example, it would not be uncommon in a FaaS deployment for the team to know not only about networking, but also about API keys, their deployment, and relevant security concerns.

Change #2: Adopt a guardrail approach to service delivery and enable Citizen Integration

One of the main advantages with using cloud provider-based services is the reduction of the time it takes to provision and de-provision resources. This agility in service delivery allows for the following:

- More people in the organization can deploy resources
- Proof-of-concept type approaches are easier to execute

Because cloud users want to move at cloud speed, IT needs to move from a roadblock approach (IT must approve all requests) to a guardrail approach (IT must allow all reasonable requests). Examples of guardrails could include:

- Security (configuration and usage)
- Cost
- Region deployment

Utilizing built-in cloud tools and/or custom management solutions, IT teams must aim to use governance mechanisms to set the boundaries for what citizen integrators can and cannot do. Governance mechanisms should favor those with automated controls and/or automated reporting capabilities.

IT must become an enabler of citizen integration.

Guardrail: automated enforcement of pre-determined rules/policies (auto configured in all deployments)

Change #3: Predictive demand management

IT/Engineering teams already spend a great deal of time monitoring their production environments. Current goals include:

- Analyzing long-term trends
- Comparisons over time
- Alerting and Dashboards
- Conducting ad-hoc analysis

The artifacts of these demand management activities are utilized in long-term planning processes. Resources deployed in cloud environments typically have better built-in telemetry options, along with faster resource scaling. This added capability allows of IT/Engineering teams to create complete, granular demand usage models, using them to make real-time scaling decisions.

Cloud provisioning models are API-driven by nature. Because of this, there are a variety of ways IT/Engineering teams can interact with cloud providers to provision required resources. Infrastructure can now be specifically deployed that is fit-for-purpose and conforms to the exact specifications of a given solution. Furthermore, there is no downside to provisioning infrastructure to support a single application.

However budgets still need to be prepared and IT/Engineering must now predict the aggregate demand of all Hybrid IT infrastructure services, so that an accurate understanding of future IT operating costs can be predicted.

Change #4: Incorporate development processes

- Blue/Green Testing

With new deployment methods, IT/Engineering teams need to incorporate more development type processes into how infrastructure is provisioned. For example: Cloud resources can be deployed, and traffic can be shifted to test new infrastructure deployments with a subset of production traffic.

- Source code management

Cloud resources can be defined in templating/scripting languages which can be versioned and controlled in a similar manner to application code. In many cases, the infrastructure definitions for a given solution can live alongside the application code and evolve at the same rate.

- Infrastructure validation / linting

Cloud resources can be defined in templating/scripting languages which allow for the execution of tests and linters against the resulting definitions [Ref4.3]. Security issues, best practice deviations, and corporate policy violations can be caught prior to any resources being deployed.

- Multiple environments

Templating and scripting can ensure that all environments that host an application (for example development and test environments) are provisioned exactly the same. This allows for appropriate testing at appropriate scale.

Change #5: IT Service management portal

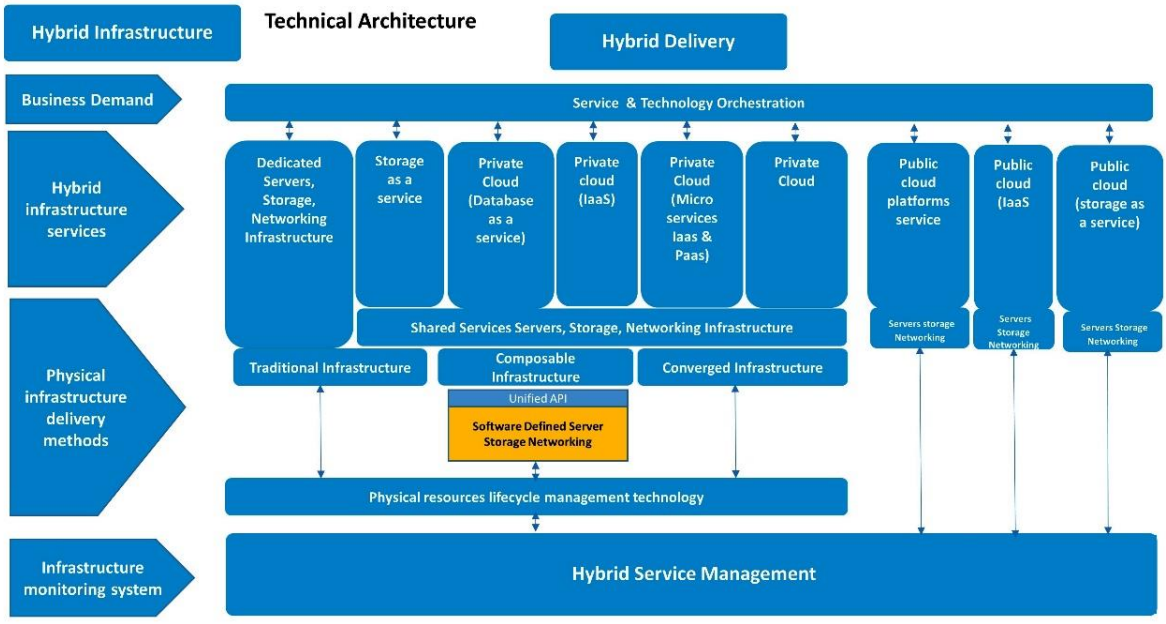

FIG 20: HYBRID INFRASTRUCTURE ARCHITECTURE

Deploying resources in the cloud can be quite involved. The multitude of options leaves the potential for resources to be incorrectly configured leading to issues such as security lapses and cost overruns. While trained cloud professionals can work with the complexities at hand, IT/Engineering teams need to enable citizen integrators to deploy required resources at cloud speed.

A resource provisioning portal would allow citizen integrators to deploy approved/vetted resources to approved cloud providers. This portal would allow for:

- Rapid deployment of required resources
- Ensuring resources adhere to security/governance policies
- Ensuring deployed resources are cost efficient

A virtual IT Budget is the control process for how much infrastructure can be purchased.

Change #6: Consolidate all infrastructure purchases via one multi-cloud Portal and virtual IT budget

Hybrid IT infrastructure has introduced significant budgetary problems for many companies because cloud services can be purchased by anyone on their own budgets. Often cloud services are purchased by individual managers and simply process through the expense report system. This leads to the IT budget being very poorly managed, and in many cases, companies have no idea now how much they're actually spending on IT.

A significant change is eliminating the ability for individual business units to purchase their own IT services and to require all IT services to be managed by one budget or at least be visible in one budget. Although this would solve the problem, it rarely works.

An alternative is to manage a virtual budget that consolidates the overall IT spend across a Business but allow Business units to allocate that budget to individual projects as they see fit. This would at least help the CFO to have visibility of the total IT cost. By providing this centralized view, it is possible to identify waste and where infrastructure is being bought and not used.

Change #7: Create Infrastructure teams to manage infrastructure demand and supply

The last recommendation is to create infrastructure teams that are responsible for significant infrastructure components such as server team, storage team, networking team, mobile, and desktop team. Most companies have had silos within their infrastructure group which have had responsibility for similar functionalities, we're suggesting that this not be abandoned but expanded so that the server group is now responsible for all server infrastructure regardless if it's within the data center or delivered by a cloud provider, the same for storage, and networking.

These teams will be first responsible for forecasting the demands on the infrastructure within their area and determining the most cost-effective method to deliver that infrastructure. For example, the server group may estimate the number of servers that need to be purchased for the following year and the additional service on demand that they may purchase from cloud functions to meet seasonal or event driven requirements.

By leveraging a central multi-cloud ordering / orchestration portal, one is able to track and forecast actual demand / use of resources and identify trends. Based on these trends, one can pre-negotiate better agreements with external providers, leveraging volume-based commitments.

Deeper analysis enabled by such a centralized control system may also help compare performance and public and private cloud costs, yielding surprising results.

Let's consider some of the unique management challenges for each of the domains:

Server Management Challenges

- Physical and virtual environments are owned by different people from different companies, with different processes being applied.
- Each group does not have sight of the other groups' operations or management, and do not share log or management data with each other.

Storage management challenges

- Storage environments are designed to be multi-functional – for one client it may host web services, and for another a transactional DB.
- Performance is not guaranteed although certain IOPS levels may be provided as a minimum.

Network management challenges

- Hybrid environments introduce additional network connections between DB, Web, Application, and clients – each under control of the different providers of that service, and managed differently, with different constraints.
- Wherever data transitions a network, it usually generates extra "egress" costs from the provider – extra costs that may not have shown before, in addition to the additional network costs.

Mobile and desktop management challenges

- Every end-user uses a different browser of their preference – version updates and plugins may not always be at most recent levels, impacting performance and functionality. The teams need to accommodate these differences for every supported browser, and monitor the transaction performance at each endpoint, and intervene in real time where problems occur.

Use of a central management portal (with dashboards and reports that are integrated into the services) that provides an end-to-end view of a cloud service and its components will ease troubleshooting and dealing with these challenges.

CONCLUSION

The transformation to a Hybrid IT Infrastructure model requires new skills, new methods, further process steps, and artifacts to be produced by an IT organization. If done well, an organization can rapidly exploit new technologies and bring new capability to the market much faster than their competitors.

REFERENCES

References and other material addressing this topic:

ID	Name	Location
1	Hybrid Workload	(1) https://en.wikipedia.org/wiki/Third_platform (2) http://www.cloudcomputingpatterns.org/cloud_computing_fundamentals/ (3) https://en.wikipedia.org/wiki/Joint_Information_Environment
2	Hybrid Delivery	(1) https://en.wikipedia.org/wiki/Architectural_pattern#cite_note-TMD-1
3	Hybrid DevOps	(1) https://searchitoperations.techtarget.com/definition/native-cloud-application-NCA (2) https://cio.gov/public-sector-cloud-computing-case-study-securities-and-exchange-commission/ (3) https://12factor.net/ (4) https://12factor.net/
4	Hybrid Infrastructure	(1) https://business.financialpost.com/technology/cio/hewlett-packard-co-opens-canadian-security-operations-centre (2) https://www.thesslstore.com/blog/2018-cybercrime-statistics/ (3) https://jslint.com/ https://www.pylint.org/
5	CMM	(1) https://www.oaca-project.org/cmm40/

https://blogs.msdn.microsoft.com/azuresecurity/2016/04/18/what-does-shared-responsibility-in-the-cloud-mean/

https://aws.amazon.com/compliance/shared-responsibility-model/

https://www.alertlogic.com/solutions/platform/google-cloud-security/