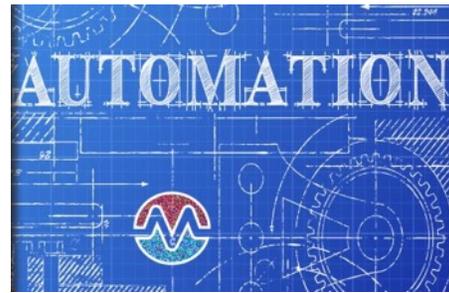




The Inevitable Rise of Platform Operations

An Intellyx Analyst Guide - By Jason Bloomberg and Jason English



We continue to evolve toward delivering both traditional and cloud-native applications atop a hybrid cloud future that is truly distributed and heterogeneous. This requires a new way of thinking about service delivery and developer enablement.

A **hybrid cloud platform operations** approach can pave the way for product teams and developers to use their favored tooling and pipelines, while at the same time dealing with environment inconsistencies and making sure there are guardrails in place so the enterprise can safely and cost effectively scale.

This Intellyx Analyst Guide prepared for Morpheus Data will educate IT leaders on the benefits of platform operations as well as the requirements to support an agile self-service provisioning framework which spans private and public clouds, VMs and containers, plus automation and IaC tools while unifying IT service management from day-0 through day-2.



Table of Contents

| | |
|--|-----------|
| Bringing DevOps and CloudOps together | 3 |
| Taming wild containers on the Hybrid Cloud & DevOps Frontier | 8 |
| Scaling Automation & IaC with a Self-Service Platform Framework | 14 |
| Modernizing the IT service management and operations dream | 19 |
| About the Authors | 24 |
| About Intellyx & Morpheus Data | 25 |



Part 1: Bringing DevOps and CloudOps Together

By Jason Bloomberg Managing Partner & Analyst, Intellyx



The common theme across the first generation of cloud management platforms (CMPs) was command and control. The cloud operations (CloudOps) team used CMPs to keep the lights on while sticking to the budget, turning various knobs as needed to ensure their organization's cloud infrastructure kept humming along.

What's good for CloudOps, however, isn't necessarily best for developers.

The old days when dev would throw code over the wall to ops are long gone. Today, developers require tooling that enables them to work hand-in-hand with ops as they develop and deploy the software so essential to the business.

First-generation CMPs, however, fall short. Such platforms provide a central control point and source of truth for CloudOps but lack the features that support the developer experience necessary for modern software. Additionally, the increasingly heterogeneous nature of IT has meant hypervisor or cloud-specific tools have driven the need for a more agnostic approach than these early CMPs provided.



From Command and Control to Cacophony and Chaos

The opposite extreme from CMP's ops-focused command and control are the loose assemblages of development and CI/CD tools we call the DevOps toolchain.

DevOps toolchains vary from one development shop to another, and often within large shops as well from project to project. Take some Git-based tool, add one or more infrastructure-as-code products, mix in some value stream management, and top it off with deployment tools. Voila! You have a DevOps toolchain.

These toolchains have one primary benefit: flexibility. Members of the development team can pick and choose the tools they want to use, ideally empowering them to achieve greater levels of productivity and job satisfaction. However, such flexibility comes at a cost. Toolchains can be brittle, expensive to maintain, and lack the governance policy that enterprises demand.

Organizations thus face a dilemma. Legacy CMPs offer command and control without developer flexibility, while DevOps toolchains bring flexibility to the table at the expense of increased complexity and risk. The solution is somewhere in the middle: **platform engineering**.



Platform Engineering: Paving the Road to Hybrid Cloud and DevOps Best Practices

Platform engineering seeks to balance two conflicting priorities: the self-service capabilities that give developers the flexibility they require while supporting the operations rigor necessary to meet the organization's hybrid cloud management needs.

The goal of a platform engineering team is to enable product and development groups to move faster via an internal developer platform (IDP) that provides a curated set of tools, capabilities, policies, and processes as a single, coordinated framework architected for self-service.

Platform engineering is the discipline of building such a framework that supports the DevOps toolchain requirements while also enabling the enforcement of policy controls in a way that is sustainable for the organization. Such practices, however, should never be written in stone. Instead, one of the most important goals of the platform engineering effort is to make the best approach the easiest and most intuitive approach, without actually mandating it.

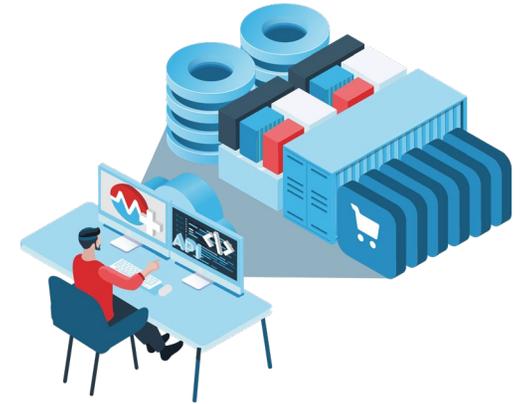
A common metaphor for describing this approach is a 'golden path' or a 'paved road.' There will always be other paths to the goal of deployed, quality software, but one road is the paved one, and hence the easiest and best-used path.



Extending Platform Engineering to Platform Operations

IDPs are still a relatively new innovation, and as a result, the organizations that have been implementing them have largely built them in-house. Given the complexities involved, such a challenge is not for the faint of heart. Only the most mature software development shops are up to the task.

Fortunately, there are unified platforms like [Morpheus](#) that provide the basis for an IDP by offering next-generation CMP capabilities while also supporting DevOps toolchains that meet various development team requirements.



Leveraging an orchestration framework to support platform engineering efforts addresses do-it-yourself IDP challenges while also supporting day-2 operations and service management needs of IT. In other words, this approach extends platform engineering to encompass platform operations.

Morpheus offers all the features of a modern CMP, but at its core, it is a self-service developer platform that was designed by (and for) [platform engineering and operations teams](#).

In particular, it offers self-service provisioning of VMs, containers, clusters, and apps into any public or private cloud with all the guardrails that enterprise ops, security, and finance teams typically require.

Morpheus also supports a fully customizable instance catalog, facilitating the on-demand deployment of operating systems, databases, and other common infrastructure components into any environment, from bare metal to virtualization stacks, to Kubernetes and even cloud-managed services. Catalog items can also represent triggers for automation and infrastructure-as-code execution.

Perhaps most importantly, Morpheus offers over 90 built-in integrations with numerous CloudOps and DevOps tools, enabling organizations to quickly implement their own DevOps toolchains while simultaneously supporting self-service CloudOps in a way that is less brittle than hand-crafted home-grown platforms.

While people have called Morpheus a CMP, it's actually much more. It is a next-generation framework for supporting platform engineering and operations teams, enabling organizations to build IDPs and unify cross-cloud CloudOps without having to start from scratch.

The Intellyx Take

The techie's most important tool for dealing with overweening complexity is abstraction – and building full-lifecycle abstractions is at the core of Morpheus.

The platform supports each step in the lifecycle – control, provision, connect, configure, deploy, operate, and scale – in spite of the fact that each step involves many different pieces of technology and various orchestrations and workflows.

Such abstraction-driven simplicity is inherently opinionated, favoring particular ways of doing things that form the 'paved road' so essential to the operation of an IDP.

The flexibility to veer from the road is always available – abstractions only hide complexity rather than eliminating it – but with the help of Morpheus, the development and operations teams are better able to achieve the proper balance between the control of a CMP and the flexibility of a DevOps toolchain.

Part 2: Taming wild containers on the hybrid cloud DevOps frontier

By Jason English Partner & Principal Analyst, Intellyx



The wild west days of operating physical IT infrastructure for use by application development teams—before the cloud abstracted everything—really aren't that far in the past. Even as recently as a decade ago, most executives probably thought of the cloud as a place for hosting SaaS apps, rather than an operating model for application environments.

Today, we must support DevSecOps patterns across both private and public clouds, each with its own complexities. Operations teams are being asked to provide management and observability front ends, databases, and other services, many of which are still delivered as virtual machines while more and more are being delivered within customized Docker containers and ephemeral Kubernetes-orchestrated pods.

Now hold on a minute—it sure sounds like we are still living in the wild west! How can platform operations teams corral so many moving parts for application delivery teams, and bring this heterogeneous hybrid cloud herd into the future?



Drawing ops into the dev container wrangling business

The early stages of cloud-native development happened in fits and starts, as developers sought ways to run applications within highly portable containers that could run practically anywhere, encapsulating dependencies including code, OS, and libraries.

For the first time, developers could get past some of the operational constraints of legacy IT approaches. They could realistically provision their own target applications—just download a container image from a library that looks close enough, install code, and spin it up on their own test server, in the data center, or on AWS or Azure—in less than a second.

Kubernetes (or K8s) came along and offered even more promise for development teams to control their own destiny, by letting them deploy and orchestrate complete cloud-native environments with internal networking, security, and data handling features, within the release pipeline.

However, anyone who has ever had to install and setup a K8s cluster much less maintain their clusters post-release in a real production environment will tell you that wrangling containers across multiple releases and distributions, in a way that supports an entire enterprise just isn't that simple. Time for ops teams to ride to the rescue...



The changing role of Operations on the hybrid cloud frontier

Cloud-native development and CI/CD approaches encouraged us to govern the organization's herd of K8s clusters and containers as 'cattle not pets' — so that developers favor ops-approved packages and configurations over highly bespoke, unique distributions.

The concept of *platform engineering* arose from the DevOps tenet of ops teams providing a platform for self-service environments, so devs could grab a self-service environment that represents the organization's current target architecture. Devs want to continue building new functionality, rather than requesting ops involvement for provisioning environments for each release.

Unfortunately, even the best-laid provisioning plans didn't account for the widespread developer use of Git-style repos of readily downloadable K8s cluster definitions and container configurations that would be shared among developers in the wild over time.

Furthermore, no application stands alone, even if developers started building it in the 'globally approved' environment. Modern applications have many dependencies on third-party services and API calls, while still needing to bring forward the existing enterprise fleet of VMs and core systems that never really went away.

To address this moving architectural target, the discipline of *platform operations* picks up where the self-service concept of platform engineering leaves off on Day 1 and operationalizes the continuing management of their environments on Day 2 and beyond, when devs are equipping themselves to deliver the next set of features and rethink how they are deployed.



Normalizing the inconsistencies of operating dev environments

Over time, large companies will acquire many unique and heterogeneous technology stacks as they grow and adapt their hybrid cloud application and data estate to the bespoke needs of customers, partners and their own delivery groups serving different parts of the business.

This leaves platform operations teams with the difficult—but not insurmountable—task of normalizing a highly inconsistent development and delivery environment.

We'd love to be able to agree on standard definitions, but whose standard will everyone trust? VMware? Red Hat? Rancher? AWS? Azure? In any decent-sized enterprise, there's a massive number of permutations of IaC configurations, charts, and delivery mechanisms that have already been selected in the past that may be hard to swap out or replace without serious impact on the business.

Rather than forcing dev teams to accept a single rigid enterprise standard, the platform operations approach offers developers a self-service 'paved road' approach of offering a library of tested and well-maintained packages proven most likely to work for different groups of constituents and application types.

This paved road can extend to the underpinning K8s clusters themselves. While there are historical business and technical reasons a given distribution may have been selected, there is an opportunity to streamline how those clusters are requested, deployed, and managed. It's also important that the road does not exist on an island, given that enterprise application libraries are still made up of more than just containers. The goal is to minimize the bumps and bruises along the way as the herd makes its way across the bare metal, virtualized, and hyperscale landscape.



Platform operations teams maintain awareness of all available hybrid IT infrastructure for the region and application use case, so developers and product teams can auto-pilot provisioning to continue down the 'paved road' approach or specify which type of infrastructure would work best for a given project.

Keeping the paved road clear of potholes

Some production environments and networked services trend toward peak demand levels and become costly bottlenecks to successful outcomes, whereas others become as abandoned as ghost towns with tumbleweeds rolling by. Platform operations work on behalf of finance and other key stakeholders to continuously optimize the utilization of clusters, containers, VMs, and more across the hybrid cloud estate. Frequently used environments need high availability and autoscaling capabilities on highly performant infrastructure, while underutilized environments that are in less demand should be deprioritized or decommissioned to save costs.

Who owns what in this hybrid cloud, and who pays for what?

Answering authorization and FinOps questions in a highly interdependent microservices world requires more than simply assigning account ownership rights to individuals or groups, or arbitrarily rate-limiting services to meet cost-cutting goals.

Platform operations teams also offer a central point of policy for governing access and chargebacks, allowing companies to continue using their preferred service management tools, in order to realize efficiencies of scale and management effort over the long term.

The Intellyx Take

Platform operations is a practice that advances the goals of the enterprise for maturing the use of hybrid cloud infrastructure. In an organization, platform operations teams fulfill the -Ops requirement of DevSecOps, where usage patterns are variable and the demand for ready environments that meet the application form factor is highest.

Platform operations practices don't generally dictate a particular technology stack, but instead can best support development teams through a well-oiled self-service portal that eliminates friction while still providing operational guardrails.

[Morpheus Data](#) offers a solution designed for hybrid cloud platform operations teams that can be represented to the organization as a self-service platform operations portal, managing most known flavors of containers, K8s clusters, VMs, cloud-native PaaS workloads, and more, so development teams can use their preferred CI/CD, GitOps, service management, and collaboration tooling, atop almost any private or public cloud infrastructure. Beyond that, their platform includes significant day-2 capabilities to tie into centralized monitoring, logging, backup, and other technologies to complete the picture.

Even as the DevOps frontier expands outward over the cloud-native deployment horizon, platform engineering, and platform operations make it easier for developers to benefit from well-governed, tested, and defined images and cluster layouts that help the org meet performance, cost, and compliance goals—rather than riding out like rogue cowboys and rustling their own.



Part 3: Scaling Automation & IaC with a Self-Service Framework

By Jason Bloomberg Managing Partner & Analyst, Intellyx



IT automation has a turbulent history... Sometimes code is not enough

First, there were runbooks: simple, step-by-step instructions for how to handle certain IT processes for on-premises physical equipment. Early runbooks consisted entirely of manual steps, but over time, engineers would automate individual tasks within each runbook using configuration management tools.

Virtualization and then cloud computing brought powerful abstractions to bear. Provisioning an instance in the cloud, for example, was a simple matter of entering a few configuration parameters and clicking a few buttons – a far cry from the weeks-long process of provisioning physical servers back in the day.

More recently, cloud-based simplicity has empowered engineers to up their automation game with infrastructure as code (IaC). IaC technologies like Terraform by HashiCorp let engineers create complex, dynamic IT infrastructure automation scenarios – not simply for operations, but across the entire software development lifecycle.

However, IaC is not a panacea, as IT environments consist of hybrid, cloud-native combinations which are too dynamic and diverse for IaC to provide for everything that Ops needs. Furthermore, every organization that is implementing dynamic and diverse environments also struggles with staffing. There are simply not enough qualified technical people to go around.

As a result, today's platform operations teams with limited resources need help with context and collaboration across multiple tools and clouds to support enterprise scale.

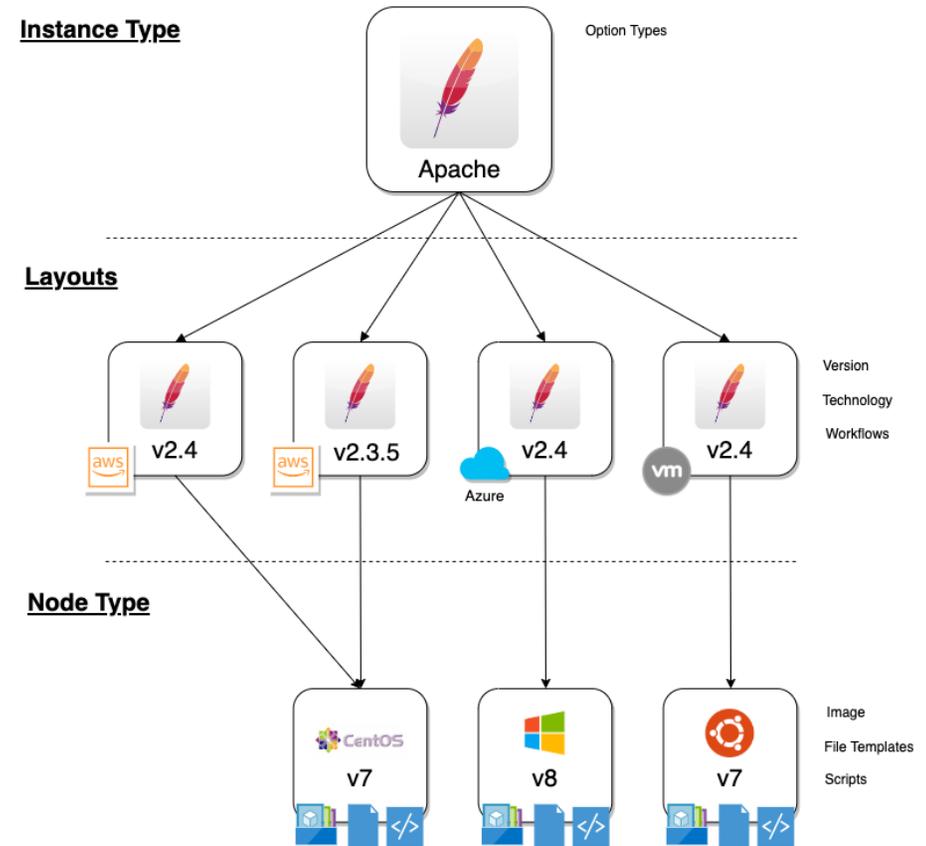
A Blueprint for Infrastructure-as-Code and Automation Success

IaC and configuration management in general have been important enablers of DevOps, as they bring automation to DevOps toolchains, enabling continuous deployment within CI/CD pipelines.

In practice, however, the more complex scripts become, the more they struggle with the same issues that code faces: versioning, testing, and increased complexity that limits scalability. These limitations are particularly pronounced in heterogeneous cloud native and multi-cloud environments.

At scale, pure script-based solutions with tools like IBM's Red Hat, Ansible, or Terraform can become brittle. The solution is to add a layer of abstraction that enables the right tool for the right job without adding complexity. This diagram from [Morpheus Data](#) illustrates how these abstraction layers work.

In the diagram, the platform operations team wants to support the ability of engineers to provision different instances of the Apache web server in different clouds on different operating systems based on specific project requirements.

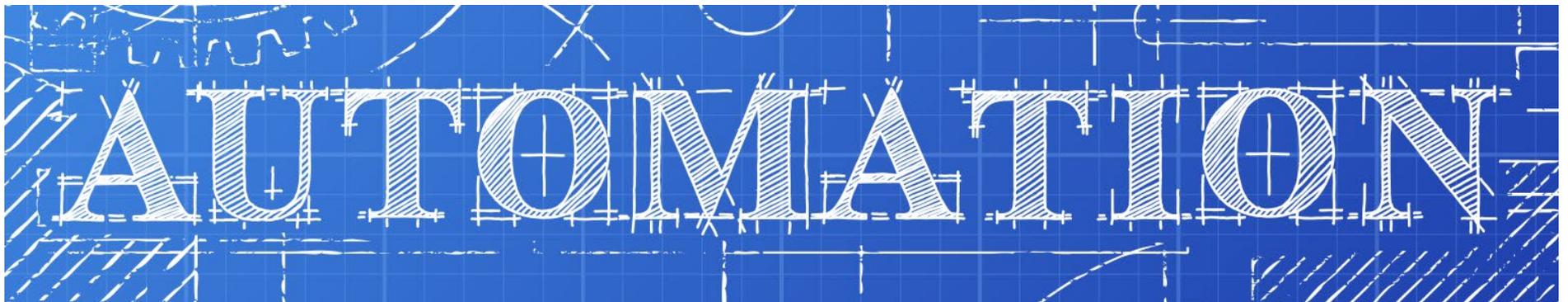


On the Morpheus platform, the diagram above represents a blueprint. Provisioning a web server is but one of many examples of the power of blueprints and their constituent layouts to provide a seamless self-service experience. The same experience could apply to Databases, OS's, Container Images, Kubernetes Clusters, etc.

Morpheus breaks down such blueprints into multiple layouts (the middle tier above), each of which corresponds in this case to a particular version of Apache on a particular cloud, along with the workflows necessary to set them up. Each layout, in turn, runs on a node type. The platform team can embed the right scripts in context at this lower level to properly provision each instance of a node type from the appropriate machine image and file templates. The blueprint can also include scripts mapped to the natural lifecycle phases of service – start, stop, reconfigure, teardown, etc.

From the developers' perspective, they simply have a 'provision Apache' button in a unified portal as well as a single API set or Terraform provider should they want to stand everything up programmatically.

They do not need to worry about the nuance of specific clouds or configuration minutiae; all they do is indicate the right cloud and layout, and the platform (in this case, Morpheus) does the rest.



Extending Self-Service Beyond Workload Provisioning

Platform operations is a modern approach that seeks to balance two conflicting priorities: the self-service provisioning capabilities that give developers the agility and flexibility they require while supporting the automated operations and rigor necessary to meet the organization's cloud management needs.

Take, for example, common scenarios like managing service accounts within a cloud, provisioning new landing zones, patching distributed systems, or simply cleaning up infrastructure. Such mundane activities are not as sexy as developer enablement, but they are in fact critical to maintaining the production operations of a global enterprise.

Individual automation engineers often perform these routine tasks using their unique home-grown scripts based on their tool of choice. Such manual work leads to siloed automation and one-off responses to issues that slow down the effort and prevent teams from scaling.

Morpheus aims to simplify life for platform teams by expanding the ability of operations to take advantage of automation using the same self-service capabilities that benefit developers. By bringing tasks and workflows into a centralized [self-service automation](#) platform, operators can benefit from automation without having to be an expert in the underpinning task type. This can be particularly important in heterogeneous environments that often use a mix of Terraform, Ansible, Puppet, Bash, PowerShell, etc.

Exposing automation tasks within a self-service catalog can take advantage of application development practices such as version control and collaborative development, without falling into the trap of custom responses to individual issues.



The Intellyx Take

One of the primary constraints to the deployment of cloud-native hybrid IT at scale is the shortage of qualified technical talent. The Morpheus approach to hybrid cloud platform operations helps to mitigate the challenge, as blueprints supplant complicated, time-consuming tasks with simple self-service functionality.

Morpheus also lessens organizations' dependency on expensive add-ons to open-source IaC projects – tooling whose capabilities are now too narrow to address the gamut of hybrid infrastructure automation requirements. IaC still plays a role, but it is now a piece of the automation puzzle rather than the puzzle itself.

Service catalogs and marketplaces have been around for years, but typically host simple, preconfigured services. Morpheus Data's blueprints avoid this problem by adding additional layers of abstraction and a unified programmatic interface that enable developer and operator self-service, even in dynamic hybrid IT environments at scale.



Part 4: Modernizing the ITSM and ITOM dream

By Jason English, Partner & Principal Analyst Intellyx

How hybrid cloud platform operations can provide an evolutionary jump in how the enterprise thinks about IT service delivery



***“To dream the impossible dream. To fight the unbeatable foe
To bear with unbearable sorrow. To run where the brave dare not go...”***

– “The Impossible Dream (The Quest)” from Man of La Mancha

Enterprise IT leaders have long dreamt of delivering self-service application environments that are easy to maintain and monitor within budget constraints yet are as agile as the DevOps teams they support.

Perhaps this quest is no longer as quixotic as it used to be, thanks to the advance of elastic computing resources atop private and hybrid cloud platforms and hyperscale providers, driven by cloud native technology, infrastructure-as-code, and delivery automation.

Even if dev and ops teams sally forth with the bravest of automated cloud provisioning efforts, without a good connection to the business and organizational realities represented within service management, observability and FinOps platforms, they might as well be tilting at windmills.

There's a lot of excitement today around *platform engineering*, and moving away from classic, sequential delivery processes by setting up self-service provisioning atop hybrid cloud infrastructure.

By comparison, *hybrid cloud platform operations* is an equally if not more important superset of tooling and processes which surround modern IT. We want to offer developer self-service, yes, but we also need to monitor and unify the ongoing operations of the cloud resources we spin up today into the future, in a manner that is embedded into the everyday process engineers participate in.

Embedding service management in modern practices

In the classic model of provisioning, a centralized IT Ops team took in dev team requests from an IT Service Management (ITSM) system like ServiceNow or Jira as a front door.

Each request ticket—hopefully accompanied by thorough documentation—would then go through a number of resource, budgeting, quality control, and security approval processes and system checks from there, until the Ops team finally built and deployed the requested system a few weeks later.

Modern development teams can ill afford to wait for such cycles within tight timelines. Self-service automation is a core tenet of the DevOps movement, and without it, provisioning and maintaining environments will become the primary bottleneck to productivity.

Unless a course correction is made quickly, teams will 'go rogue' and do their own thing without the blessing of IT. In fact, many frustrated teams did just that, which resulted in a rash outbreak of 'shadow IT' over the last several years. Ungoverned cloud usage can



cause even worse problems when cloud costs spin out of control and deployed applications are breached or fall down in front of customers.

Internal developer platforms (IDPs) are a more modern way to address the need for speed with built-in guardrails, though we must deal with the underlying reality of populating and updating configuration management databases and tools, and interfacing with ITSM systems that have become critical for servicing customer needs.

ITSM tools and platforms like ServiceNow still have an important role to play in a hybrid cloud platform operations framework. Now they are simply a layer of the cake, with processes like CMDB CI updates baked into the system rather than bogging it down.

Stretching the limitations of discovery and observability

IT operations management (ITOM) has traditionally represented the discovery, planning, and management of IT resources on the heels of a service management delivery approach, but just like with ITSM, the tooling has evolved.

New DevOps approaches ask us to move quickly, but before we run off and start stamping out new automated environments using platform engineering, it is vital to recognize that there is no such thing as a true greenfield estate.

Being able to conduct a thorough discovery process, and wrap your arms around the enterprise's resources, relationships, and costs that are already in play is the best starting point for enabling hybrid cloud platform operations. More importantly, that information can guide the guardrails and policies that get put in place to manage costs going forward.

Once we've properly discovered the resources and automated how new environments are provisioned, organizations should unify how they leverage modern monitoring and observability platforms such as Datadog, ScienceLogic, or OpsRamp.



In an ideal world, the hybrid cloud platform operations process would populate and unify the enterprise's ITSM, change management, monitoring, financial and compliance data in parallel, rather than asking a case manager to shepherd each change ticket through separate tools and teams.

Collapsing weeks into minutes at AstraZeneca

[AstraZeneca](#), one of the world's leading bio-pharmaceutical companies, depends upon continuous innovation within its highly regulated international application estate to support the design and production of life-saving treatments.

The company's global information services (GIS) organization oversees more than 16,000 virtual and physical servers and 2,700 applications across 15 global locations spanning on-prem VMware infrastructure and multiple public cloud providers including AWS, Microsoft Azure, and Google Cloud Platform.

With sensitive IP and clinical data on the line, compliance comes first. So the process for requesting new deployments passed through several departmental and system review checks, adding costly delays to each build and release cycle.

Using [Morpheus](#) to enable hybrid cloud platform operations, they went from an end-to-end service fulfillment cycle that took 80 hours, to automation that provisioned as many as 30 servers at a time in just minutes.

With the right guardrails in place, they could let development and product teams move forward on their own, while still pushing cost metrics to financial models, config updates to change management records, anomalies to monitoring dashboards, and incidents into ServiceNow.

The Intellyx Take

In this series, we've traversed the rise of hybrid cloud platform operations by looking at the divide between [DevOps and CloudOps](#), exploring the frontier of [wrangling containers and clusters](#) in cloud native environments, and challenging the limits of [infrastructure-as-code with a self-service platform](#).

As it turns out, the lesson of our hero's journey is one of simplicity. Doing more work with less tools and less friction between teams is the end game and it's closer to reality with the help of hybrid cloud platform operations.

Making self-service environments sustainable over the long haul is not just about installing another provisioning tool in the cloud. It's about transforming the way the business thinks about its operating model and the tooling that it uses to realize that vision.

Companies don't need a tight command and control doctrine. With visibility and the right management approach, development teams get the agility they want on demand, while the operations organization can still set the policies and get the oversight needed to continuously sustain the business.

Copyright ©2023 Intellyx LLC. Intellyx is solely responsible for the content of this e-book. As of the time of writing, [Morpheus Data](#) is an Intellyx customer. No AI chatbots were used to write this article.

About Jason Bloomberg



Jason Bloomberg is founder and managing partner of enterprise IT industry analysis firm Intellyx. He is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

He is #13 on the [Top 50 Global Thought Leaders on Cloud Computing 2023](#) and #10 on the [Top 50 Global Thought Leaders on Mobility 2023](#), both by Thinkers 360. He is a leading social amplifier in Analytica's [Who's Who in Cloud?](#) for 2022 and a [Top 50 Agile Leaders of 2022](#) by Team leadersHum.

Mr. Bloomberg is the author or coauthor of five books, including *Low-Code for Dummies*, published in October 2019.

About Jason 'JE' English



Jason "JE" English is Partner & Principal Analyst at Intellyx. Drawing on expertise in designing, marketing and selling enterprise software and services, he is focused on covering how agile collaboration between customers, partners and employees accelerates innovation.

A writer and community builder with more than 25 years of experience in software dev/test, cloud computing, security, blockchain and supply chain companies, JE led marketing efforts for the development testing and virtualization software company ITKO from its bootstrap startup days, through a successful acquisition by CA in 2011. He co-authored the book *Service Virtualization: Reality is Overrated* to capture the then-novel practice of test environment simulation for Agile development. Follow him on [Twitter at @bluefug](#).



About Intellyx



Intellyx is the first and only industry analysis, advisory, and training firm focused on customer-driven, technology-empowered digital transformation for the enterprise. Covering every angle of enterprise IT from mainframes to cloud, process automation to artificial intelligence, our broad focus across technologies allows business executives and IT professionals to connect the dots on disruptive trends. Read and learn more at <https://intellyx.com> or follow them on Twitter at [@intellyx](https://twitter.com/intellyx).

About Morpheus Data



Morpheus Data is the market leader in hybrid cloud platform operations software, helping hundreds of enterprises, agencies, and service providers grow revenue by enabling development teams to move faster while staying within operational and financial guardrails.

The Morpheus software platform gives developers and product teams a customizable self-service application provisioning catalog spanning dozens of on-premises hypervisors, public clouds, and automation tools. This agnostic approach reduces the impact of skills gaps so teams can focus on innovation. With more 3rd-party integrations and functionality than any other platform, Morpheus streamlines workflows, reduces tool sprawl, and unifies processes.

Learn more and request a personalized platform demonstration at www.morpheusdata.com/demo



