Application Modernisation: The Essential Guide

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Introduction:

The idea of meeting customer demand means that the business must react quicker to change. This is a major driver in application modernisation.

IT has to support these changes with a stream of rapidly developed software. DevOps done well can lead to a continuous loop where teams plan, code, build, test, release, deploy, operate and monitor.

In the age of the customer – where you must tie together systems of record and systems of engagement to optimise customer experiences and business outcomes – there is a new role for the configuration management system (CMS). Analyst Forrester looks at how CMS can facilitate the evolution of systems of record and their integration with systems of engagement.

In the past, tight integration with core IT systems – the so-called systems of record – often led to a backlog in the

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

development and deployment of software. But software can be re-architected to support a more DevOps style of working.

One of the emerging trends is to engineer applications using microservices. Such an approach promotes developing and deploying applications that are composed of independent, autonomous, modular, self-contained units.

This is fundamentally different from the way traditional, monolithic applications are designed, developed, deployed and managed. A microservices architecture means software development teams can rapidly update individual microservices without disrupting other parts of the application.

In this guide, we look at how distributed systems have evolved to support more agile software development.

Cliff Saran, Managing Editor

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

The network is the application: Why APIs are agents of change

Cliff Saran, Managing Editor

Engineer, entrepreneur and inventor Bob Metcalfe once described how the value of a network grew as the number of Ethernet cards increased. The concept also works for applications.

While the cost of a network grows linearly based on the number of connections, the network effect, hypothesised by Ethernet inventor Metcalfe, describes how its value is proportional to the square of the number of users.

The network effect has now arrived in the application development space with the emergence of the application network, at least according to application programming interface (API) management specialist MuleSoft.

At the company's London Summit earlier in May 2016, Ross Mason, CEO at MuleSoft, warned delegates that their organisations need to get used to change.

"The world is changing and getting good at adapting to change, and ingesting new tech, is essential," he said. However, if change is the new

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

normal, the way IT has evolved to support new technological breakthroughs in the areas of social, mobile, cloud and big data is too slow.

Mason warned that all industry sectors face serious disruption. “You have to change quickly because you don’t know where the competition will be,” he said.

And according to Mason, making IT more efficient is not necessarily the answer: “You can go 30% faster, but how do you drive a much faster clock speed in IT?”

His answer is that CIOs needs to think differently and this is why IT should apply Metcalfe’s network effect to applications. “We almost always extend IT architectures too far. They break,” said Mason.

Creating building blocks

Previously, the recommended best practices in IT were that organisations invest in developing an all-encompassing platform, such as a service-oriented architecture (SOA).

Instead, Mason said: “Look at API-led connectivity.” Rather than build a large-scale reusable services infrastructure, Mason urged delegates to consider developing connections to applications as and when they are needed. These connections create building blocks.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

These building blocks form the basis of the application network, and can be made available to other parts of the business, which can use them to build their own applications.

If new connections to application functionality are needed, new APIs can be created as and when the request is made. The APIs are then developed with re-use in mind and become part of the application network, where they can be redeployed over and over again.

APIs for fast food

An example of where such a network could immediately add value is at fast food chain McDonalds.

MuleSoft customer, McDonalds, operates a franchise model. Given the nature of the fast food industry, it can be hard for the company to understand its customers, or its customers' needs.

One example of a successful franchise business that does appear to understand its customers is Uber, which recently extended its services to a parallel business opportunity in a way that meets customers' needs in certain geographies.

Mason said: "Uber shows us how a digital franchise can work and in New York it lets customers order food."

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

This ability to enable Uber drivers to offer takeaway deliveries on top of the core taxi booking service is something that McDonalds wants to adapt within its business.

Just as New York Uber drivers, who are effectively franchisees, offer takeaways, Mason said McDonalds is beginning to look at how its franchisees can develop additional value on top of core technology building blocks provided by the company.

Adaptive business integration

Another MuleSoft customer, Unilever, has taken the principles of an application network to transform IT from being a service provider to becoming a business enabler.

At the beginning of 2014, the consumer packaged goods company realised its technology platform could not achieve the speed and agility demanded by the business. Nor could it support business requirements for integrating a growing number of cloud services and mobile apps.

Despite its IT being assessed by analyst Gartner as world class, Unilever realised it needed to put in place an adaptive business integration capability to support the pace at which the company was driving digital business initiatives.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

"The only way was to make a step change and decentralise ownership of certain IT components and give business entities a level of autonomy," said Frank Brandes, director of global enterprise business at Unilever.

Brandes said the team was able to show the business how quickly adaptive integration could be delivered. "We set up an adaptive integration capability with an in-house DevOps team which could start developing applications from day one."

The adaptive integration team reports to the head of enterprise business integration. Working with Unilever's enterprise architecture department, the team used MuleSoft Anypoint integration platform as a service (iPaaS) to provide the cloud-service integration and API management required for its self-service API connectivity strategy.

A strategy for re-use

Large-scale IT architectures hark back to an era of command and control IT. A centralised IT function can no longer work effectively by providing an overarching middleware architecture for developers in the business to use.

Such architectures take far too long to build, and fail to keep up with the pace of change needed to support digital business initiatives.

What is interesting about Unilever is that IT did not have to open up every conceivable API the business might need. As Unilever's Brandes found, self-

service drives API development and for “every requirement we check if there is an API, and if there isn’t one we look at whether we should create one”.

This creates re-deployable API access, according to Brandes.

McDonalds is also looking to move beyond a centralised IT infrastructure. Giving franchises the ability to create value on top of a core set of APIs promises to make the business more adaptable.

Both McDonalds and Unilever are applying the principles of the network effect to make APIs more valuable to the business. In the so-called application network, the more APIs that exist, the greater potential value the business can derive. Surely, this is Metcalfe’s network effect in practice.

Next article

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

An application ecosystem for the all-digital era

Jean-Pierre Garbani, Guest Contributor

For the past 10 years, IT professionals have considered the configuration management database (CMDB) and configuration management system (CMS) the linchpins of service management.

CMS concerns mapping applications to infrastructure components. It is a management tool for business services and was created at a time when IT professionals built and ran infrastructure on-premises and a user device meant a laptop or desktop running Microsoft Windows.

This made sense in the context of key processes such as change, incident and capacity-management, and automated dependency discovery tools. But the CMS proved too slow for the complexity of applications and infrastructures in a dynamic and fast-moving environment. Numerous manual corrections and the very scale of the required reconciliation tasks led to very high resource consumption and high CMS project failure rates.

However, the effect of digital disruption on the business and technology is now at the core of technology management's renewed interest in CMS projects.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

To fulfil the business requirements of this digital disruption, IT managers have to pursue a dual agenda that includes systems of record and systems of engagement.

In the age of the customer – where you must tie together systems of record and systems of engagement to optimise customer experiences and business outcomes – the CMS has an expanded role. It needs to facilitate the evolution of systems of record and their integration with systems of engagement.

Traditional systems of record are typically a mix of in-house, enterprise-specific management software, often built from layers of legacy hardware and software. Systems of engagement, on the other hand, focus on interactions with people such as customers, partners or employees – first through websites, portals and e-commerce, and increasingly through mobile apps. While systems of record require care and support – consuming a large portion of the technology management budget – systems of engagement need innovation, flexibility and speed to keep the business ahead of the competition.

The need to acquire or develop more applications and the complexity of linking systems of engagement to systems of record means organisations must shift budget from maintaining systems of record to innovating in systems of engagement.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

The CMS plays a critical role in reducing the maintenance and operating costs of systems and equipment. Reviewing the application portfolio forces the business and technology management organisations to determine what they're actually using and which applications are redundant or replaceable by another technology, such as a cloud-based software-as-a-service package. It also prepares for application modernisation. The IT department needs to create a complete inventory of what was done and how it worked, if it is to prepare the whole technology management organisation for the all-digital era without major business disruption.

The CMS must change for application ecosystems

The CMS is the cornerstone of service management. But service management as we know it, and as defined by the information technology infrastructure library (ITIL), is fast becoming a thing of the past – or is at least in need of a serious revamping when you take into account infrastructure evolution and the focus on systems of engagement. Does this mean the CMS is obsolete? Actually, it just means that it has to expand to answer emerging problems posed by digital disruption, such as:

- The use of hybrid and cloud environments. If you create the CMS with a quasi real-time ability to detect changes, you still don't have much information when it comes to orchestrating applications into a different infrastructure, such as from a physical to virtual environment, or from a virtual to cloud environment.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

- The introduction of DevOps. Systems of engagement require rapid changes to applications delivered in small increments. Application-release automation requires data about items such as configurations and datasets that do not figure in today's CMS.
- The increasing use of microservices and containers. As modern applications evolve, re-usable services gradually shrink in size and scope from relatively large subsystems to smaller services that are easier to write, test and re-use. But microservices are too small to warrant a full virtual machine; and, without such isolation, they risk creating hidden dependencies. To solve this problem, containers such as those supported by Docker have exploded in popularity. Containers provide simple, lightweight infrastructure on which developers can deploy microservices.
- The rise of composite applications. Platform-as-a-service offerings from leading cloud providers give software development teams foundational capabilities and rich sets of services to rapidly build and deploy complex applications with a minimum of coding. These platforms enable companies to build rich applications and focus on creating differentiating value, without having to create complex supporting software. This results in an explosion of many-to-many relationships between applications and services.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

A tall order for the IT department

Creating a CMS – especially in large organisations – is not a walk in the park. You should approach it with a serious project plan, an understanding of the limitations of off-the-shelf systems and reasonable expectations of the time and resources it will take. Most dependency discovery solutions today are based on a bottom-up process – but a top-down approach may yield better results. Why? Because starting at the critical business service level may make it easier to collect data, and you can approach CMS creation incrementally rather than trying to boil the ocean.

Based on a modern CMS, a service information system (SIS) adds data from diverse sources to create a business service ecosystem that condenses information from all corners of the IT organisation. Building an application ecosystem through the SIS forces the IT organisation to work together toward a common goal. Because information-gathering touches many aspects of production, it exposes many of the processes IT uses, and validates or invalidates their effectiveness.

If building a CMS and SIS sounds like a tall order to impose on the IT department, that's because it is. Digital disruption is forcing IT to adopt new operating models in the near future. We need to move from a Jurassic period of IT – where each technology generation deposits layers upon layers of sediment – to a rational and competitive business technology that focuses on user value rather than technological prowess.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Microservices: How to prepare next-generation cloud applications

Janakiram MSV, Guest Contributor

In March 2014, Martin Fowler and James Lewis from ThoughtWorks published an article on microservices. Since then, the concept has gained prominence among web-scale startups and enterprises.

A microservice architecture promotes developing and deploying applications composed of independent, autonomous, modular, self-contained units.

This is fundamentally different from the way traditional, monolithic applications are designed, developed, deployed and managed.

Distributed computing has been constantly evolving in the past two decades. During the mid-90s, the industry evaluated component technology based on Corba, DCOM and J2EE. A component was regarded as a reusable unit of code with immutable interfaces that could be shared among disparate applications.

The component architecture represented a shift away from how applications were previously developed using dynamic-link libraries, among others.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

However, the communication protocol used by each component technology was proprietary – RMI for Java, IIOB for Corba and RPC for DCOM. This made interoperability and integration of applications built on different platforms using different languages a complex task.

Evolution of microservices

With the acceptance of XML and HTTP as standard protocols for cross-platform communication, service-oriented architecture (SOA) attempted to define a set of standards for interoperability.

Initially based on Simple Object Access Protocol, the standards for web services interoperability were handed over to a committee called Oasis. Suppliers like IBM, Tibco, Microsoft and Oracle started to ship enterprise application integration products based on SOA principles.

While these were gaining traction among the enterprises, young Web 2.0 companies started to adopt representational state transfer (Rest) as their preferred protocol for distributed computing.

With JavaScript gaining ground, JavaScript Object Notation (JSON) and Rest quickly became the de facto standards for the web.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Key attributes of microservices

Microservices are fine-grained units of execution. They are designed to do one thing very well. Each microservice has exactly one well-known entry point. While this may sound like an attribute of a component, the difference is in the way they are packaged.

Microservices are not just code modules or libraries – they contain everything from the operating system, platform, framework, runtime and dependencies, packaged as one unit of execution.

Each microservice is an independent, autonomous process with no dependency on other microservices. It doesn't even know or acknowledge the existence of other microservices.

Microservices communicate with each other through language and platform-agnostic application programming interfaces (APIs). These APIs are typically exposed as Rest endpoints or can be invoked via lightweight messaging protocols such as RabbitMQ. They are loosely coupled with each other avoiding synchronous and blocking-calls whenever possible.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Factors that influence and accelerate the move to microservices

Contemporary applications rely on continuous integration and continuous deployment pipelines for rapid iteration. To take advantage of this phenomenon, the application is split to form smaller, independent units based on the functionality.

Each unit is assigned to a team that owns the unit and is responsible for improving it. By adopting microservices, teams can rapidly ship newer versions of microservices without disrupting the other parts of the application.

The evolution of the internet of things and machine-to-machine communication demands new ways of structuring the application modules. Each module should be responsible for one task participating in the larger workflow.

Container technology such as Docker, Rocket and LXD offer portability of code across multiple environments. Developers are able to move code written on their development machines seamlessly across virtual machines, private cloud and public cloud. Each running container provides everything from an operating system to the code responsible for executing a task.

Infrastructure as code is a powerful concept enabling developers to programmatically deal with underlying infrastructure. They will be able to

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

dynamically provision, configure and orchestrate a few hundred virtual servers. This capability, when combined with containers, offers powerful tools such as Kubernetes to dynamically deploy clusters that run microservices.

Developers are choosing best-of-breed languages, frameworks and tools to write parts of applications. One large application might be composed of microservices written in Node.js, Ruby on Rails, Python, R and Java. Each microservice is written in a language that is best suited for the task.

This is also the case with the persistence layer. Web-scale applications are increasingly relying on object storage, semi-structured storage, structured storage and in-memory cache for persistence. Microservices make it easy to adopt a polyglot strategy for code and databases.

Benefits of microservices

With microservices, developers and operators can develop and deploy self-healing applications. Since each microservice is autonomous and independent, it is easy to monitor and replace a faulty service without impacting any other.

Unlike monolithic applications, microservice-based applications can be selectively scaled out.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Instead of launching multiple instances of the application server, it is possible to scale-out a specific microservice on-demand. When the load shifts to other parts of the application, an earlier microservice will be scaled-in while scaling-out a different service. This delivers better value from the underlying infrastructure as the need to provision new virtual machines shifts to provisioning new microservice instances on existing virtual machines.

Developers and administrators will be able to opt for best-of-breed technologies that work best with a specific microservice. They will be able to mix and match a variety of operating systems, languages, frameworks, runtimes, databases and monitoring tools.

Finally, by moving to microservices, organisations can invest in reusable building blocks that are composable. Each microservice acts like a Lego block that can be plugged into an application stack. By investing in a set of core microservices, organisations can assemble them to build applications catering to a variety of use cases.

Getting started with microservices

Docker is the easiest way to get started with microservices. The tools and ecosystem around Docker make it a compelling platform for both web-scale startups and enterprises.

Enterprises can sign up for hosted container services like Google Container Engine or Amazon EC2 Container Service to get a feel of deploying and managing containerised applications.

Based on that learning, enterprises can consider deploying container infrastructure on-premise.

Next article

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Legacy application modernization projects: Proceed with caution

Crystal Bedell, Guest Contributor

As IT organizations undergo legacy application modernization projects, they must understand what approach they want to take and what they are trying to achieve. Although plenty of tools are available, few of them result in a modernized application, according to Gartner analyst Dale Vecchio and other experts. In addition, today the lines are blurred between application modernization and migration to cloud applications.

Making sense of application modernization tools is no easy task. No single tool will work for every project and, experts warn, even the right tool won't get you across the application modernization finish line.

Choosing the right tool for legacy application modernization

Choosing the right tool "depends on where you're coming from, and where you're going to," Vecchio said. "There are a variety of tools depending on the approach you want to take." Vecchio defines three primary approaches to legacy application modernization:

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

- Non-invasive application modernization involves exposing an app to be consumed in new ways, such as through a service or Web front end.
- Invasive modernization involves "code transformation, from an old language to a new one; COBOL to a new modern language, usually Java or C#. It's the same app implemented on a completely new stack," he said.
- Migration is, at essence, a rip-and-replace approach. Today, that usually means that legacy, on-premises applications are replaced by cloud applications.

Eighty percent of legacy application modernization projects today are just migration, according to John David Head, director of enterprise collaboration, PSC Group, a technical and business consultancy largely focused on software modernization. He said PSC sees businesses that have a legacy platform that's problematic -- in costs, performance, etc. -- and they are not looking to vendor updates as solutions. For various reasons, he said, they want to go to another platform. "That's not modernization," he said. Vecchio agreed, noting a few differences between simple legacy application migration and full-on modernization projects. If a business wants only to move to a different package, for example, they're focused on how to map the existing functions to the package, identify the gaps and decide how to fill them.

True app modernization involves much more than migration. For example, Head said, "The migration of collaboration apps from one technology

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

platform to another offers a unique opportunity to modernize apps, update processes, improve security and prepare for a more mobile driven and dynamic future."

The perils of invasive software modernization

Unfortunately, using tools for what Vecchio calls invasive application modernization doesn't get IT organizations much closer to modernization than migrating from one app to another. "Transformation of the technology alone doesn't really cut it. It's an OK start, but the way you implement old legacy apps is different than the way you do it in the new world, so when you just do a transformation you don't get the app you wanted," Vecchio said.

Capgemini's Ron Tolido agreed. "Of course, you can use tools that pick up an existing application and immediately transform it to a next-generation application, but in practice that's not really what you want," said Tolido, senior vice president, group CTO office, and advisory and architecture lead of the global insights and data practice for Capgemini. "If it's already a sausage, you cannot turn it into a pig again. It's a little bit too late. If there's an old application, you don't want to automatically turn it into something new. There are tools that do it, but it's quite awkward."

That said, it is important that IT organizations understand what legacy application modernization tools can and can't do. "My opinion is that what the tools do best is migrate the data. What they attempt to do is move the

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

business functionality. We have found that most of that works very poorly," Head said.

Outsourcing legacy application modernization projects

Instead of using a tool themselves, Vecchio said most IT organizations he speaks to prefer to outsource legacy application modernization projects to service providers. "No one wants to spend money on a tool to get from point A to point B because when they get to point B, what are they going to do with it?" Vecchio asked. "They don't have the resources or skills, so they look to service providers to support them."

Still, it's important that IT organizations understand what it takes for true application modernization to take place. PSC Group works with most, if not all, of the application modernization tool vendors. "I'm not disparaging them in any way," Head said. Often, however, there's a huge gap between the customer's expectation of what they want and the reality of what tools can do in migration. For example, the customer may walk in and ask for a simple migration, but the customer adds that the new app has to fit with the existing platforms. "If the customer wants customization, I need to go through a modernization process to build a custom experience," Head said. That's why businesses and internal IT or third-party service providers must agree on requirements. "That level set is really, really important," Head said.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Digital transformation needs mainframe DevOps

Kurt Bittner and **Robert Stroud,** Guest Contributors

Mainframe systems of record are the beating heart of most large businesses. The mainframe is essential to 92 of the top 100 banks worldwide, 23 of the top 25 US retailers, all of the world's 10 largest insurers, and 23 of the world's 25 largest airlines.

Successful digital businesses unleash the data and business processes encoded in their mainframe-based applications.

Ad hoc integrations between systems of engagement and systems of record get them started, but they soon find their ability to define innovative products and services is limited by an inability to evolve and improve their mainframe applications.

Mobile and cloud applications garner the greatest attention, but nearly every aspect of what they do depends on data and processes locked in mainframe apps. A seemingly simple mobile insurance app is actually the gateway to a complex set of applications that must work seamlessly with the app and with each other.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Many companies speak of their mainframe-based applications as assets, but they are more often liabilities that impede their ability to innovate. Those old applications embody calcified processes that prevent companies from responding to market opportunities with new products and services.

IT leaders have legitimate fears about modifying applications they know little about. Combinations of code analysis tools and services help them better understand the structure of their applications, letting them make informed changes with greater confidence.

The various approaches to this have different advantages and disadvantages: service-based approaches help IT organisations quickly eliminate technical debt. Offerings such as those from Infosys and The Software Revolution use analytic tools plus expertise to help IT teams make sense of the mess.

Mainframe applications are complex. While tools provide insight, human judgement is often needed to make the right call about where and how to restructure and modernise.

These services make the most sense for organisations undertaking significant restructuring, re-platforming, or rewriting of specific mainframe applications.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Low risk insight into application code

Tools-only methods favour a gradual self-serve approach. For organisations keen to do their own analysis, tools like Cast, Compuware's Topaz and IBM's Rational Asset Analyzer provide insights into application code that increase understanding and reduce the risk of change. These tools help them understand the structure of the code and identify opportunities for improvement.

Static code analysis helps developers improve code quality. Integrating static analysis, using tools like RogueWave's Klocwork and SonarQube, into pre-commit processes helps to correct modularity and code quality problems before they affect the main code base.

Developers tend to reject static analysis when it is used to judge or evaluate them or when the rules are poorly defined and produce too many warnings on non-critical errors.

Reducing thechance of code change

Dynamic analysis and analytics help pinpoint operational issues. Instrumenting and monitoring mainframe applications, as well as moving production monitoring tools and practices into testing environments, also help developers reduce the risk of changing code. Identifying and resolving issues earlier reduces the chance that changes will lead to problems in

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

production; when they do occur, the issues can be found and resolved faster. Tools such as IBM Application Delivery Intelligence and Compuware Strobe are being integrated with widely used DevOps tools from companies like AppDynamics, Atlassian, Jenkins, SonarSource and Splunk to provide greater insight across the life cycle.

Antiquated coding tools such as ISPF present significant barriers to younger developers working on mainframe application code. Familiar tools, such as IDEs with their integrated debuggers, syntax-directed editors and familiar interfaces, make picking up Cobol or PL/1 relatively easy. Tools such as Compuware's Topaz Workbench, IBM Rational Developer for zSystems, and Micro Focus Enterprise Developer eliminate the need to learn old platform-specific tools and provide rich developer experiences that bring modern practices to bear on old code.

Realising the benefits

Mainframe applications must not prevent organisations from reaching digital transformation goals. The same modern DevOps practices that help IT organisations deliver mobile and cloud apps faster will help them deliver mainframe applications faster, too.

To start realising these benefits, IT leaders should only modernise mainframe applications that are essential to digital experience. Most organisations have vast portfolios of mainframe applications. They can't all

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

be modernised at once, nor do they need to be. Focus on the applications that provide information critical to customer experiences.

These applications provide essential data to customer journeys, define and manage the products customers buy, or manage the processes that guide the customer journey.

Forrester recommends funding mainframe application improvements through digital initiatives. Organisations struggle to fund modernisation for modernisation's sake, but they usually have plenty of money for customer-facing initiatives that create new revenue opportunities. Clearly delineate how improving mainframe application delivery is essential to the success of digital initiatives, and build funding into the digital initiative budget to make the necessary improvements.

Forcing developers to use antiquated tools and processes that only apply to mainframe development turns new developers off. Giving them modern and familiar tools makes mainframe development just another technology to master and reduces the amount they need to learn, so they can be productive much faster.

While languages are different, developers are generally multilingual and adopt new languages all the time. The fundamental processes are exactly the same, as are the benefits of automation and standardisation. Improve mainframe application delivery speed by using automation to speed, simplify and standardise the process.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

The mainframe is not going away, but the way it is used will change. Containers and microservices are coming to every platform, including the mainframe. Breaking large monolithic applications into smaller services will help the transition to a containerised future that promises faster application delivery, greater scalability and better manageability.

Next article

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

How to deploy microservices in a hybrid, multicloud environment

Tom Nolle, Guest Contributor

Most IT organizations have come to recognize the benefits of componentized software in development and deployment. In the cloud, componentization brings important advantages, such as increased resiliency and support for horizontal scaling.

Microservices -- or small, functional elements that are often shared among applications -- can magnify these benefits considerably. But first, you have to plan, develop and deploy microservices properly.

Understand what makes microservices tick

To begin microservices planning, IT teams need to understand what makes microservices different than application components or elements of a service-oriented architecture. Microservices are not complete application pieces; they are designed to be shared, as services, among applications -- meaning multiple apps can invoke a single instance of a microservice at the same time. Microservices are also designed to use web-like RESTful interfaces.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

If microservices don't fit the model above, they aren't likely to deliver as many benefits. When microservices do match the characteristics above, you need to sustain each of them in a hybrid or multicloud deployment.

Microservices' impact on multicloud networks

Because microservices are small pieces of functionality, they can divide applications into many successive requests for an external service. This service is accessed over a network that can introduce propagation delay and other network performance issues. It is critical that the network connection that links microservices to the applications that use them delivers the quality of service (QoS) needed to support users' experience. Before you deploy microservices, test their performance in all of the hosting variations across your hybrid or multicloud environment. If your QoS falls below acceptable levels, change your network connectivity to correct it. Alternatively, you could design your application deployment process so that services aren't moved to dead spots in your network.

Network performance issues in hybrid and multicloud applications are usually related to the way traffic passes through the multicloud -- or cloud and data center -- boundary points. Talk to your cloud providers, your VPN provider and your data center networking team to optimize connectivity. Be especially wary with multicloud applications, because many public cloud providers won't connect directly with other providers; they will expect to connect back through your VPN or data center network. If an application in

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

one cloud uses a microservice in another, there could be a long potential propagation delay. If you can't reduce it, avoid crossing cloud provider boundaries with microservice access. You may need to deploy a duplicate of the service in each cloud to avoid such network performance issues.

The need for multiple applications to access a microservice may also require network accommodations. The easiest way to approach microservice access is to assume that you have a flat private network that joins all your clouds and data centers. That way, you can deploy microservices anywhere, and applications can reach them using standard IP mechanisms, such as URLs and Domain Name Services (DNS), or other service cataloging methods.

Another challenge occurs when a microservice moves from one cloud provider to another, or between a cloud provider and a data center. Normally, this kind of movement requires a change in the IP address, which means the logical name of the service will have to be associated with a different address after the move. Make sure your tools and practices for replacing a failed component make the necessary change to DNS or service catalog entries so your applications can find the microservice in its new location.

The fact that multiple applications often share a single microservice can create two other challenges in hybrid and multicloud environments: security and compliance, and stateful vs. stateless behavior.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Deploy microservices securely

Any time applications share functionality, there's a risk that an application with rigorous compliance requirements will be contaminated. This is because a shared service might provide outsiders with a portal for entry. Since moving microservices, or duplicating them under load, requires fairly open addressing, you need to secure each microservice with respect to its access. Avoid microservices that mix features demanding security and compliance monitoring with other features open to a larger community -- make them two different microservices.

Explore the stateful vs. stateless issue

The stateful vs. stateless issue is complex, even for software architects and developers. Applications typically support transactional activity that involves multiple steps or states. For example, imagine we have a service called "add two numbers." If we present the first number on one request and the second on another, other users could inadvertently introduce their own number between our two, and we'd get the wrong answer.

If microservices cannot save data between requests made to it, then make the requests stateless or ensure they can somehow convey the state, if needed. In our example, providing both numbers to be added eliminates the need for multiple requests, as well as the stateful behavior risk. It's also possible to have the request include a user ID that the microservice would

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

associate, through a back-end database, with the state. When our first number is presented, the microservice would record that number in the database. Then, when the second is presented, it could add them and return the answer.

There's always a price to be paid for versatility, agility and flexibility -- and combining microservices with both hybrid and multiclouds represents the leading edge in our search for these three benefits. Plan carefully to minimize that price and deploy microservices that extend easily into a complex cloud future.

In this e-guide

- The network is the application: Why APIs are agents of change
- An application ecosystem for the all-digital era
- Microservices: How to prepare next-generation cloud applications
- Legacy application modernization projects: Proceed with caution
- Digital transformation needs mainframe DevOps
- How to deploy microservices in a hybrid, multicloud environment

Getting more CW+ exclusive content

As a CW+ member, you have access to TechTarget's entire portfolio of 120+ websites. CW+ access directs you to previously unavailable "platinum members-only resources" that are guaranteed to save you the time and effort of having to track such premium content down on your own, ultimately helping you to solve your toughest IT challenges more effectively – and faster – than ever before.

Take full advantage of your membership by visiting www.computerweekly.com/eproducts

Images; Fotalia