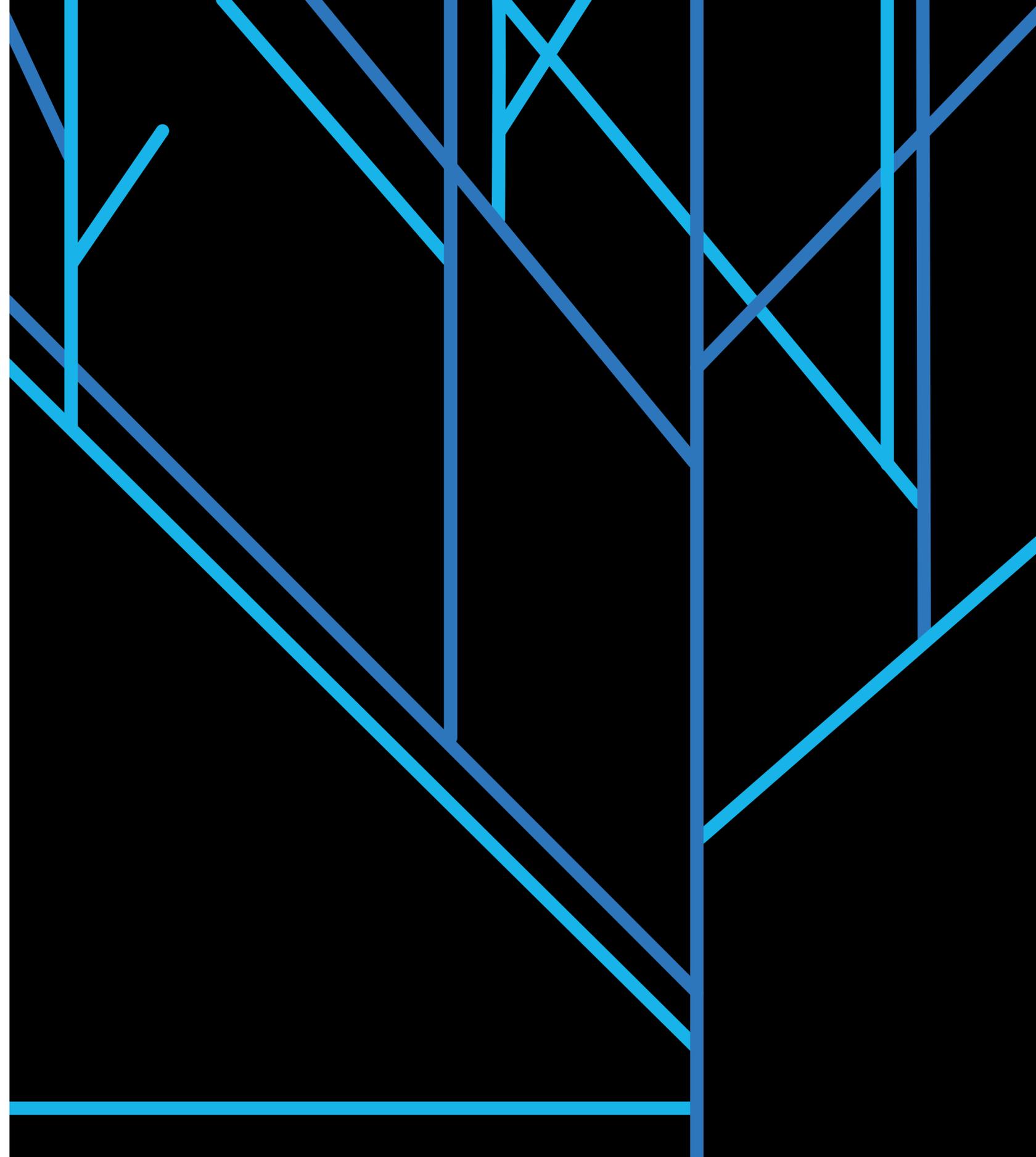
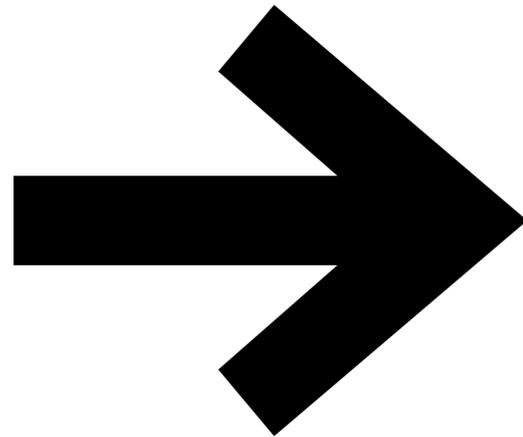


Microsoft Azure e-book

# Expedition Cloud

## Inside Microsoft's cloud migration and application development journey





<b>03</b>	<b>Digital transformation fuels today's enterprise growth</b>	<b>21</b>	<b>Refactor:</b> make minimal changes
<b>06</b>	<b>Cloud migration and application development accelerate your speed of innovation</b>	<b>22</b>	Case study 2 / Refactoring in action: achieving distributed deployment for remote assessment apps
<b>08</b>	<b>Choosing your cloud migration approach</b>	<b>25</b>	<b>Rearchitect:</b> get modular
<b>11</b>	<b>Expedition Cloud: Microsoft's cloud strategy</b>	<b>26</b>	Case Study 3 / Rearchitecting in action: redesigning a revenue reporting system for cloud architecture
<b>15</b>	<b>Expedition Cloud: inside Microsoft's cloud migration and application development journey</b>	<b>29</b>	<b>Rebuild:</b> go cloud native
<b>17</b>	<b>Rehost:</b> lift and shift quickly	<b>30</b>	Case Study 4 / Rebuilding in action: creating an integrated plug-and-play supply chain with serverless computing
<b>18</b>	Case study 1 / Rehosting in action: migrating business-critical Treasury applications to Azure	<b>33</b>	<b>Replace:</b> use business software as a service
		<b>34</b>	Case Study 5 / Replacing in action: simplifying the sales platform and deploying Dynamics CRM Online
		<b>37</b>	<b>Summary</b>

# Digital transformation fuels today's enterprise growth

**In the era of digital transformation, business applications are the backbone of any organization, driving essential business processes and helping to engage customers.**

Many of these applications were custom-built over time to support specific processes and enhance productivity. However, they often were designed using monolithic architecture and older technologies that are complex, expensive to support, and unable to meet the changing business demands of higher performance, visualization, omnichannel customer experiences, faster decision making, and easier integration with modern apps. With the cloud's leading-edge technologies, you can modernize your on-premises and legacy applications to enhance their usability and make them future-ready.

When migrating to the cloud, you want to focus on development and DevOps improvements that increase productivity, agility, flexibility, and operational efficiency. This does not necessarily mean completely redesigning applications to move to the cloud. The right migration strategy will be based on your organization's needs and priorities, along with the kind of applications being migrated.

The first step is to evaluate your applications to define an **application portfolio** strategy. You create an application portfolio by assessing the technical complexity and business impact of each application and classifying it to determine when it should be targeted for migration. The application portfolio must consider both the needs of your business and the requirements of your applications to ensure your cloud migration strategy is seamless and cost effective.

# Start with these questions

May  
2018

Digital transformation fuels today's enterprise growth

5

01 /

Which applications can be retired or left on-premises?

02 /

Which applications can be "lifted and shifted" directly to the cloud?

03 /

Which applications need to be transformed or rearchitected leveraging cloud technologies?

04 /

Which applications can be built from the ground up using cloud-native technologies?

05 /

Which applications can be outsourced by using commercial solutions like software as a service?

# Cloud migration and application development accelerate your speed of innovation

Your current application portfolio may include critical applications that your organization built over time and still uses for core business functions. The cloud provides a comprehensive set of services with the flexibility and security to build, deploy, and manage applications through a global network of datacenters. Integrated tools, DevOps practices, and a marketplace that provides support as you build anything from simple mobile applications to internet-scale solutions—or as you deploy your legacy apps.

**A good cloud provider should offer you the following:**

**Productivity**

Use the cloud to help developers deliver apps faster with end-to-end development and management tools. Modernize or build applications using the language of your choice and streamline the application lifecycle across teams with integrated source control, DevOps practices with continuous integration and delivery (CI/CD), and built-in support for application monitoring, log analytics, patching, backup, and site recovery.

**Hybrid capability**

Build and deploy apps consistently across public or private clouds with an integrated technology stack. Extend on-premises apps, databases, and identities across the cloud to maintain a consistent set of user identities and apps. Gain immediate insight into your workloads and infrastructure on-premises and in the cloud with powerful search, queries, and analytics

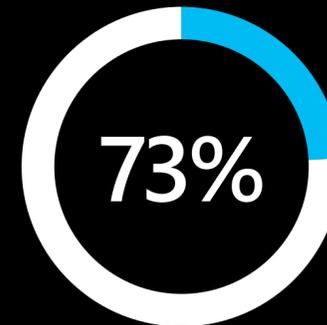
**Intelligent services**

Create the next generation of intelligent applications by using powerful data and artificial intelligence (AI) services. Easily infuse intelligence into your business applications without spinning up a team of data scientists.

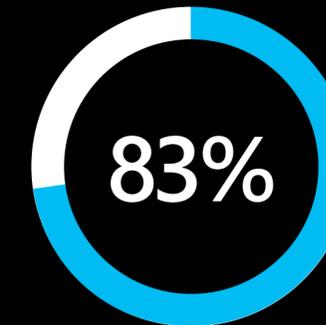
**Trust**

Use the cloud to help ensure security, privacy, transparency, and compliance for your organization. The cloud adheres to a broad set of international and industry-specific compliance standards like ISO 27001, HIPAA, FedRAMP, SOC 1 and SOC 2, as well as country-specific standards like Australia IRAP, UK G-Cloud, and Singapore MTCS.

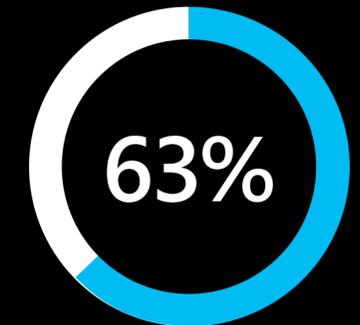
**Transforming your workloads<sup>1</sup> from On-premises to IaaS**



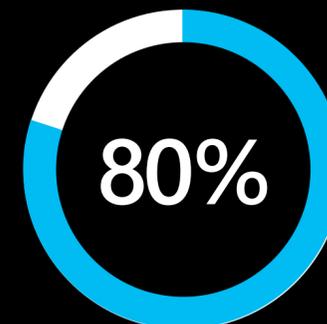
reduction in data center footprint and costs



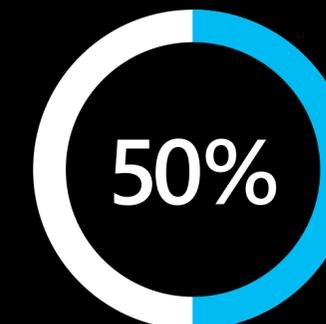
reduction in IT outsourcing cost



increase in annual customer web sales



improvement in IT application management



improvement in time per customer solution deployment



improvement in project or transaction revenue

1. Microsoft, The Total Economic Impact of Microsoft Azure PaaS, July 2016.

# Choosing your cloud migration approach

**Because every application is unique, there's no single set of steps to follow for modernization.**

Some applications are ready to modernize; some may require minor changes in code; and still others may need a complete redesign to make them cloud-ready. Choosing the right migration approach for different applications in your portfolio can help you extract maximum value from cloud-enabled and innovative technologies. Traditional enterprises, including Microsoft, align and execute application portfolios using the migration framework below.<sup>2</sup>

#### **Rehost**

Rehosting is also referred to as a "lift-and-shift" migration. In essence, the application is moved into the infrastructure as a service (IaaS) environment of a cloud provider while avoiding as many modifications to the system as possible.

#### **Refactor**

Refactoring, often referred to as "repackage", is to minimally alter application code or configuration changes necessary to connect the application to a provider's PaaS (Platform as a Service) and take better advantage of cloud.

#### **Rearchitect**

Rearchitecting is to modify or extend the existing application's code base to optimize it for a cloud platform and for better scalability. For example: you can decompose a monolithic application into microservices that work together and readily scale on Azure. Moreover, you can rearchitect your SQL Server database to a full managed Azure SQL Database.

#### **Rebuild**

Redevelop the application from scratch on a provider's PaaS platform. With rebuilding, you completely build your application to make it cloud-ready. Although rebuilding requires losing the familiarity of existing code and frameworks, the advantage is that you gain access to innovative features in the provider's PaaS platform.

#### **Replace**

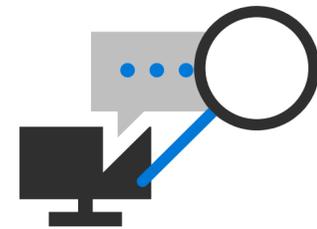
With replacing, you move your existing application directly to a software as a service (SaaS) offering. With this approach, you can avoid mobilizing a development team when requirements for a business function change quickly.

2. Gartner, [Devise an Effective Cloud Computing Strategy by Answering Five Key Questions](#) June 2017.

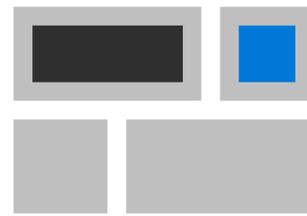
# Cloud migration & application development



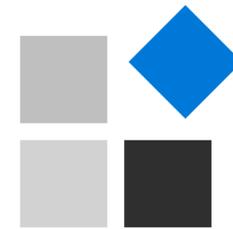
Rehost



Refactor



Rearchitect



Rebuild



Replace

# Expedition Cloud: Microsoft's cloud strategy

## Core Services Engineering and Operations (CSEO), formerly Microsoft IT, is the organization within Microsoft driving modernization and transformation.

When organizations consider application modernization, a common thought is often that it sounds good on paper but seems overwhelming to implement. At first, it was no different with Microsoft, despite its large scale and multinational resources. But since 2011, teams at Microsoft have actively—and successfully—pursued cloud adoption to benefit from platform efficiencies, development agility, and rapid deployment capabilities.

CSEO builds and manages the critical products, processes, and services that run Microsoft. They boldly pursue big ideas that power transformational advances for their customers, while helping internal Microsoft teams work smarter, faster, and more securely every day. To shorten feature and service development cycles and respond to internal customer needs faster, CSEO started on a journey to modern engineering. To become more agile as an organization, they are adopting a DevOps culture of close cooperation and joint ownership of services.

At the start of CSEO's cloud migration journey, their mission was "everything runs in the cloud"—and, indeed, Microsoft is on a rapid journey to adopt cloud computing throughout the entire company. As a part of its cloud adoption strategy, Microsoft plans to move approximately 2,100 line-of-business (LOB) applications to the cloud. These applications are spread across eight datacenters worldwide and comprise more than 40,000 distinct operating system instances. With a deep understanding of the application portfolio, the CSEO team is identifying and optimizing applications to better align to business functions and reduce waste.

From the start, the CSEO team understood that Azure is an excellent cloud platform for many workloads; however, some applications are not yet optimized for the cloud. Old application complexity and regulatory requirements made it challenging to immediately move a small fraction of internal apps. Despite this, the vast majority of LOB applications are targeted for

migration as soon as possible. The decision to retain a small number of applications in traditional datacenters for the near future meant that some components would be run in Azure and some in the datacenters. This resulted in a hybrid cloud configuration—a blend of on-premises (private) cloud and off-premises (public) cloud. Orchestration between the two allows mobility of workloads between locations, depending on need, cost, and flexibility.

CSEO always chooses the public cloud first because it offers maximum agility and scalability. In addition, they rely on SaaS for commoditized services, such as Microsoft Dynamics CRM Online, SharePoint, and email. Existing applications with planned investments use Azure PaaS, reducing lead time between strategy and service. If no further investment will be made in an existing, sustaining application but it still fulfills a service, the application will move to Azure IaaS. In addition, all preproduction environments will be moved to Azure.

When CSEO began selecting applications to move to the cloud, they made simple classifications to determine when an application should be targeted for migration. They created an application portfolio based on two major factors—technical complexity and business impact—and started with the least technically complex applications that had the least impact on business. This approach let them build new architecture models and enhance the skills of engineering teams to fully take advantage of new capabilities without great risk.

**To drive cloud adoption, CSEO provided critical functions, including:**

**Cloud adoption strategy**

Developed a strategy for a cloud adoption journey that meets the vision of “everything runs in the cloud” and takes advantage of cloud-enabled cost savings as well as agile development and deployment capabilities.

**Cloud adoption analysis**

Analyzed business-critical applications and their underlying platform requirements and dependencies; also assessed the cloud capabilities required to enable the adoption of cloud technologies.

**Cloud readiness assessment**

Assessed LOB applications for cloud compatibility and developed a decision framework for where—or if—an application would live in the cloud.

**Guidance and training**

Consulted with Business Process Units (BPUs) to provide guidance and training for enabling a cloud-first IT organization.

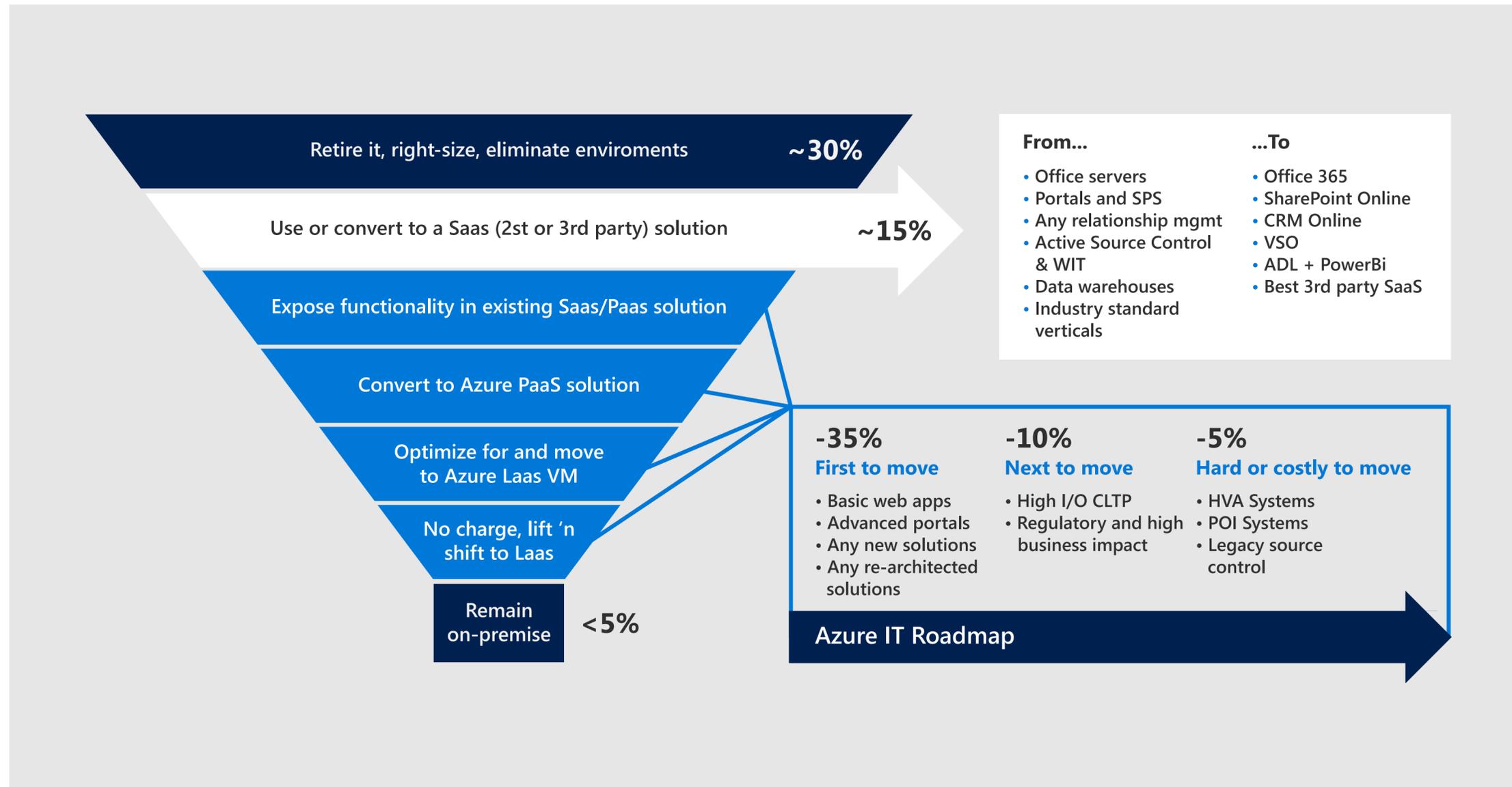
**Reporting**

Developed reporting to share accountability across BPUs.

These functions provided a deep and thorough evaluation, revealing how the vast portfolio is categorized and slated for adoption. To start, nearly 30 percent of the application portfolio could be retired, right-sized, or eliminated. Various functionalities could be consolidated into a single application or service line, resulting in the elimination of thousands of physical servers and virtual machines (VMs). Approximately 15 percent of the application portfolio was replaced by SaaS solutions, such as Microsoft Office 365, SharePoint Online, and some third-party offerings, as they became available.

Complex and customized LOB applications make up 50 percent of the application portfolio. Within that, most have been identified as “first to move.” These are basic web applications or rearchitected solutions. The remainder are identified as candidates that are “next to move” aggressively to IaaS. A small portion is identified as “hard or costly to move,” and less than 5 percent of applications will remain on-premises.

With this cloud adoption journey, CSEO is realizing multiple positive outcomes, including a simplified application portfolio, more time to focus on high-value activities, greater emphasis on delivering customer capabilities, and the ability to respond more quickly to business changes.



**Learn more**

Microsoft's cloud adoption strategy use cases:

[Our cloud journey with Brad Wright](#)

[Microsoft IT cloud computing strategies continue to evolve](#)

**For more information on the journey, take a look at the following Microsoft use cases:**

[Cloud strategy with Mel Lowe](#)

[Driving cloud adoption in an enterprise IT organization](#)

# Expedition Cloud: inside Microsoft's cloud migration and application development journey

Microsoft is using a well-planned application portfolio to guide its journey to the cloud. To modernize this large, complex, and diverse portfolio, Microsoft relied on five models like those suggested above: **Rehost, Refactor, Rearchitect, Rebuild, and Replace.**

Let's take a closer look at each model, along with how Microsoft used it.

# Rehost: lift and shift quickly

## Take the first step in your cloud migration journey with rehosting to the cloud.

Rehosting your legacy application by migrating to the cloud can help to overcome on-premises challenges like application downtime due to hardware failure, isolated management of IT infrastructure, expensive and time-intensive procurement processes, inefficient disaster recovery, and inconsistent archiving. Rehosting can be considered a first step toward cloud adoption and is the fastest way to migrate because it doesn't require any code changes to your app. Moving to the cloud enables better scalability, reliability, and security of the underlying application infrastructure; it also significantly reduces capital investments and overall operational cost.

Azure provides a comprehensive cloud migration path—from assessing your existing application to optimizing it post-migration. Rehost your legacy application to Azure by using services like Azure Migrate and Data Migration Assistant to assess application workloads and databases. Then use Azure Site Recovery and Azure Database Migration Service to perform actual migration to the cloud. From there, you can deploy workloads using Azure Virtual Machines, Azure Storage, and Azure Network. Plus, you can use Microsoft Operations Management Suite (OMS) to monitor application performance and forecast, track, and optimize your spend once the workload is migrated to Azure. In addition, you can adopt PaaS capabilities to modernize your application to be future-ready.

## Case study 1 /

# Rehosting in action: migrating business-critical Treasury applications to Azure

### Situations & challenges

Microsoft Treasury is responsible for managing the financial assets of the company. The high-level functional teams include Capital Markets, Capital Management, and Risk and Credit Services. These teams were facing critical challenges with the existing application solution: Data volumes were progressively growing, nearly doubling every year, which increased the cost of maintaining storage infrastructure. In addition, numerous data sources were disconnected and isolated from one another. Plus, because financial business experiences seasonal peaks in application and data usage, applications that support periodically high volumes were running on architecture that wasn't fully utilized for the entire year. Finally, a complex, manual process

was required for disaster recovery and high availability.

### Solution

The Microsoft CSEO team assessed the current Treasury environment and established high-level goals to create a specific set of functionalities in an Azure-based solution. The primary consideration for the applications was suitability for migration to PaaS to take advantage of built-in resiliency, scalability, and manageability. It was known that the PaaS adoption would stretch out over several months, as engineering and development needed to redesign the user experience, evaluate the Treasury experience, and assess applications for migration.



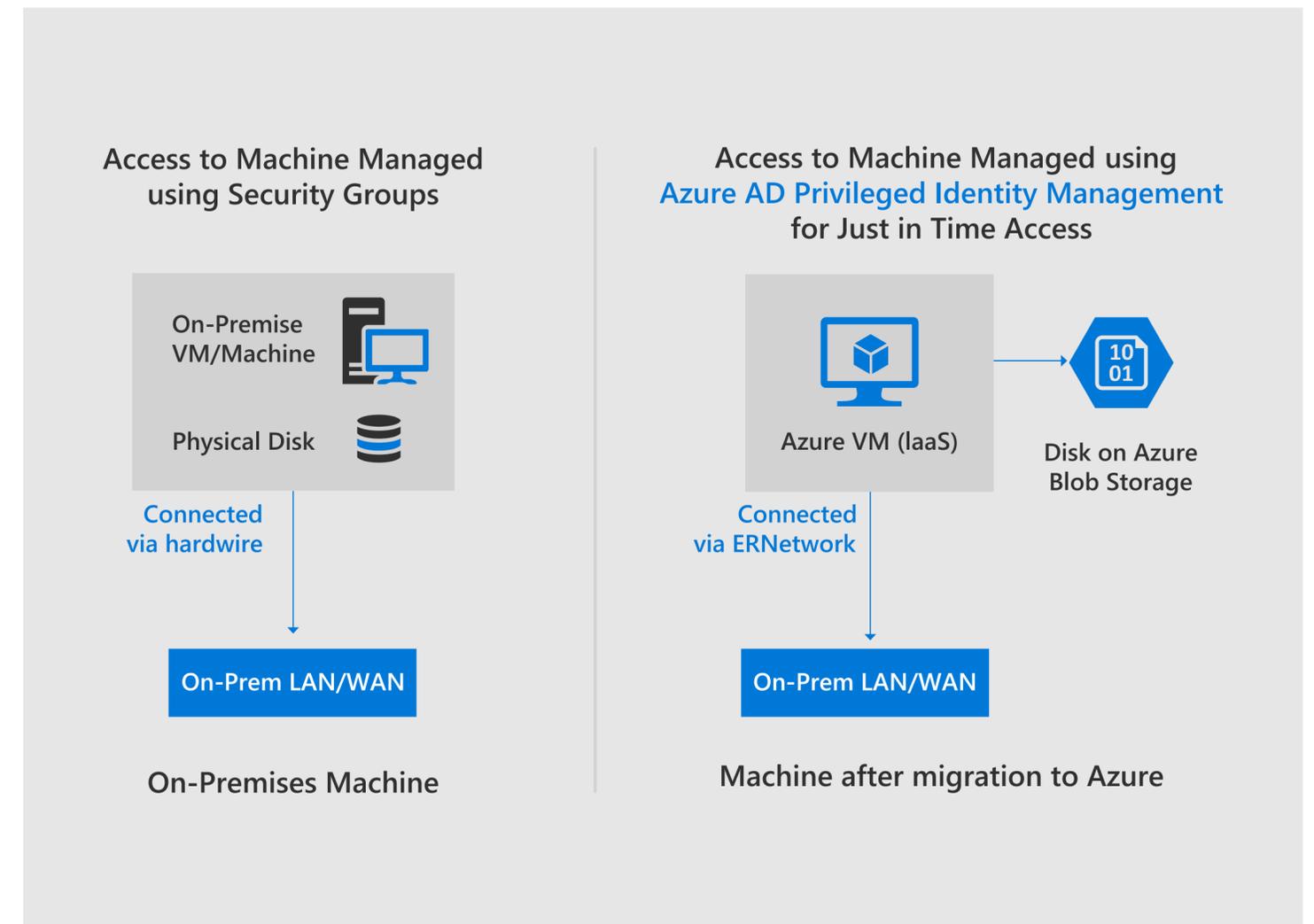
Migration to Azure IaaS was an ideal solution to immediately benefit from hosting the Treasury environment in Azure and provide time to properly plan for migrating into Azure PaaS. Lift and shift provided a simple migration method to get Treasury infrastructure onto the Azure platform quickly, which immediately reduced datacenter needs.

Once applications were transitioned into Azure IaaS, the team began analyzing them to determine which were suitable for moving to Azure PaaS. The migration approach was business-focused, so the team first established business objectives for the PaaS migration. Proceeding to assess first-party applications for migration, they realized that most applications identified as PaaS candidates would require some level of redesign. Development teams saw this as a chance to recreate the same functionality set in each application and to reconsider how the application could be more functional, more streamlined, or even combined with another app. This analysis provided

significant opportunities to build a better, more functional, and more unified Treasury application environment.

**Business impact**

Azure has changed the way the Microsoft Treasury IT infrastructure operates. Using a combination of Azure IaaS and PaaS components, CSEO was able to quickly migrate all on-premises applications to Azure. By doing so, they have created a single infrastructure cost—Azure subscription. Now they can succinctly quantify infrastructure spending and use those numbers to further reduce costs and gain efficiency. Scalability for application resources is now just matter of minutes or hours, compared to days, weeks, or potentially months. By reengineering the application portfolio for Azure PaaS, CSEO could revisit the Treasury experience and approach it as an entire solution, rather than a combination of individual apps. The overall Treasury experience has become more reliable and business-oriented as a result of migration to Azure.



The infrastructure footprint was reduced by 61% from 54 servers to 21, along with the storage space and network infrastructure required by those servers.

#### Lessons learned

While moving toward modernization, the Microsoft Treasury team identified several "Lessons learned" that can help shape their solution going forward:

#### Using a phased approach to achieve a smooth transition

Although cloud adoption was a high-level goal, Microsoft Treasury wanted to take advantage of the built-in resiliency, scalability, and manageability of PaaS solutions for their business apps. To move forward, they worked with CSEO to first identify and migrate applications best suited to the lift and shift process. They then analyzed the applications to determine their fit for moving to Azure PaaS. For non-suitable apps, the team defined what types of redesign were required to move them to the modern application stack.

#### Keeping security always in mind

Protecting business data is a top priority at Microsoft Treasury. When migrating Treasury applications to Azure, the team considered all compliance and data security aspects of hosting data in the public cloud.

#### Understanding suitability and migration efforts

Before moving to IaaS migration, the team focused on understanding the complexity of the application infrastructure, the size of each application and related databases (in terms of velocity, volume, and variety), and the requirements for seasonal elasticity.

#### Capturing all application and data legal requirements

Because complying with legal requirements for data safety and security can be a complex issue, the team worked with the stakeholders and data owners for each application to capture all corporate and legal compliance needs.

Learn more about application rehosting by exploring these resources:

[Migrating business-critical Treasury apps to Microsoft Azure](#)

#### Commonly used services

[Azure Site Recovery](#)

[Azure Cost Management](#)

[Azure Migrate](#)

[Azure Database Migration Service](#) [Azure](#)

[Virtual Machines](#)

# Refactor: make minimal changes

**Add cloud-enabled capabilities to your application while retaining your existing business logic and application code.**

Transforming your legacy application by modernizing your application deployment architecture lets you retain your existing application code and business logic. With this approach, you can add cloud-enabled and innovation capabilities to your application with minimal code changes and leverage the benefits of cloud PaaS services. For example, you can move your existing apps to App Service (Platform as a Service) or Azure Kubernetes Service (AKS). You can also refactor your SQL database into Azure SQL Database Managed Instance.

Container is one of the popular ways to refactor thanks to its portability, scalability and compatibility with PaaS. Package your legacy application using container services that combine the required configuration and dependencies as a single container image, which is abstracted from the environment and underlying infrastructure.

With Azure, you can build, deploy, and run container-based applications in the cloud or on-premises with a choice of leading open source tools and technologies, including Kubernetes, Docker Swarm, and DC/OS—all supported by Azure. Azure provides multiple options for your containerization strategy, including Azure Kubernetes Services (for containers as a service) and Web Application for Containers (for deploying containerized web apps). You also can directly deploy Azure Container Instances without having to provision any VMs. To build, debug, and deploy your containerized apps, you can rely on Microsoft Visual Studio tools. Plus, you can use Azure Container Registry services to store and manage images for all types of container deployments.

## Case study 2 /

# Refactoring in action: achieving distributed deployment for remote assessment apps

### Situations & challenges

The Microsoft Global Enterprise Services team provides quick responses to customer requests for assessing their business problems and mapping them to cloud solutions. With 1,100 architects and consultants, the team wanted to more efficiently scale their highly skilled experts to global customers. To achieve this, they needed a high degree of automation to augment their existing Microsoft Teams document-based assessment solution.

### Solution

With the help of open source technologies, the Global Enterprise Services team migrated their existing .NET Framework document automation applications to a container-based solution. This approach allowed the team to deploy the

solution to a local Docker environment, Azure Kubernetes Service, or Azure App Services. It also gave them the freedom to have personal and team deployments that can be scaled to the wider organization.

In moving to the cloud, the team built a distributed application architecture that delivers real-time data updates and a deployment model that allows features to be enabled at runtime. As the first step to application modernization, they migrated their existing .NET Framework applications to .NET Core. This step allowed them to standardize on a mix of public and private Linux containers to run the required application workloads using Node.js and .NET Core. The minimal code changes gave them the portability to deploy





Web Apps for Container

Azure Subscription

azure-cli



Installer

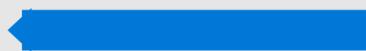


Local Docker

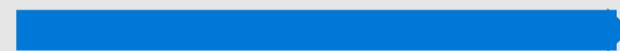
Target



Feature Switches



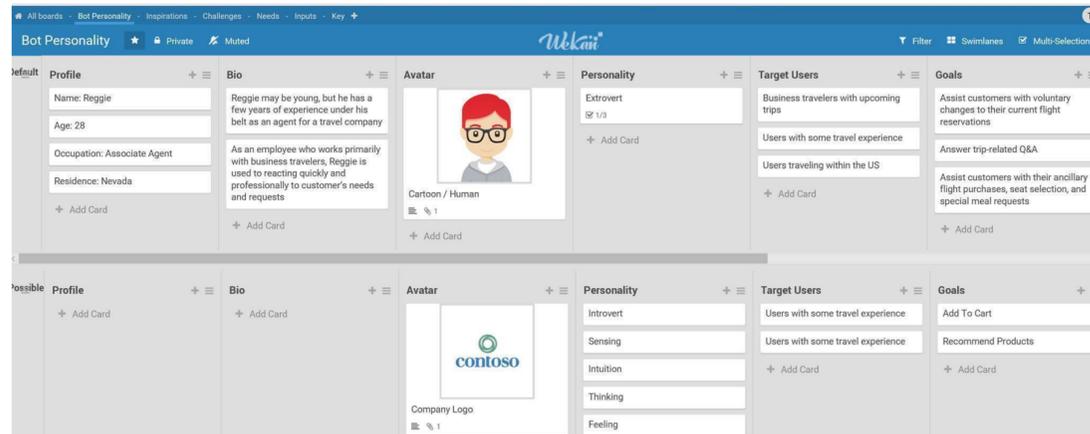
VSTS – CI/CD, Monitoring



Azure Container Registry

# DevOps / Automation

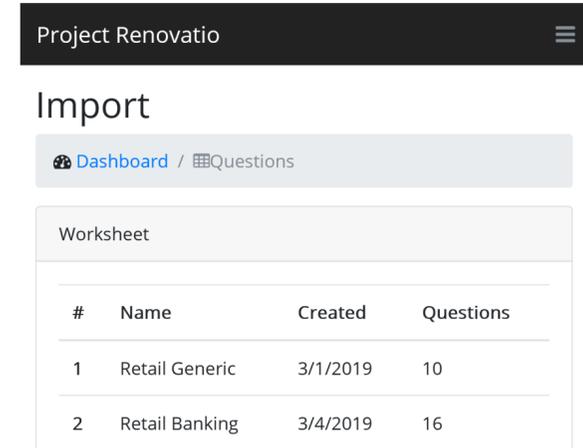
Assessment



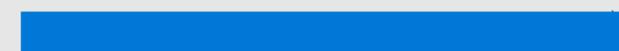
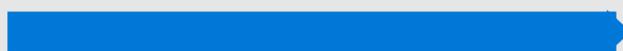
<https://hub.docker.com/r/wekanteam/wekan/>  
<https://github.com/wekan/wekan>

Node.js / Meteor

Admin



.Net Core / Node.js



Cosmos DB API for MongoDB

# Data Sources

# Augmented assessments mean moving from four days of in-person discussions to a set of four one-hour interactive Skype calls.

on local Docker installations and on Azure. For scalability and distributed application deployment, the team used a combination of Azure Kubernetes Services and a VSTS DevOps pipeline to generate containers and package changes.

## **Business Impact**

With this solution, the Global Enterprise Services team has dramatically improved their assessment capabilities—moving from four days of in-person discussions to a set of four one-hour remote Skype calls that can be scaled and conducted globally. With distributed application deployment on Azure, architects and consultants can interact with any number of customers in real time, without worrying about managing the back-end infrastructure. This reduces the solution’s management overhead, which allows the team to focus on their growing global demands. By refactoring the platform using Azure services and a DevOps pipeline, they can quickly validate and deploy builds to live applications and experiment with new features.

May  
2018

Expedition Cloud: inside Microsoft’s cloud migration and application development journey

24

## **Lessons learned**

While moving toward modernization, the Global Enterprise Services team identified several “Lessons learned” that can help shape their solution going forward:

### **Automation is essential**

Automation must be implemented to allow seamless deployment locally and into Azure.

### **Open source tools enable innovation**

Migrating to Linux containers and .NET Core allowed the team to reuse open source tools and existing assets to quickly innovate.

### **Flexible hosting is best achieved with Azure Kubernetes Service**

Azure Kubernetes Services provided the best hosting option for the multiple container services that make up the solution.

**Learn more about application refactoring by exploring these resources:**

[Optimizing Microsoft IT legacy apps with containers](#)

[Five tips for moving virtual machine-based apps to Windows Server Containers](#)

## **Commonly used services**

[Azure App Service](#)

[Azure Kubernetes Service](#)

[Azure Container Instance](#)

[Azure Container Registry](#)

[Azure SQL Database Managed Instance](#)

[Azure Service Fabric](#)

[Azure SQL Database](#)

# Refactor: get modular

**Redesign your application to be modern, transform it with a modular architecture.**

Rearchitecting is to modify or extend the existing application's code base to optimize it for cloud platform and better scalability. Rearchitecting is a good time to break apart a monolithic application into a set of loosely coupled modules or functions that integrate to build a complete app. With this approach, you can group different application functionalities into modules that can then be scaled up or down and tested, deployed, and managed independently. If any code breaks, only that module or function is affected; the rest of the application remains available. And when a module or function no longer meets current business requirements, you can swap it out for a better option.

With its innovative services, Azure is the perfect partner for redesigning your application to make it cloud-ready. You can build your application into modules and use different Azure services for each of them. For example, your application front and logic layers can use Azure Containers, Microservices, and Web Apps, while your database requirements can use Azure SQL Database, Azure Cosmos DB, MySQL, and SQL Data Warehouse. You can also use Azure Databricks, , and Spark Processing for advanced analytics on big data, machine learning, and business intelligence requirements.

### Case study 3 /

## Rearchitecting in action: redesigning a revenue reporting system for cloud architecture

### Situations & challenges

The Microsoft Sales team manages Microsoft data related to product and service sales and gives a consistent view of Microsoft businesses and production revenue. The team was struggling to manage the growth rate of transaction volume as customers shifted to online services. Existing hardware was quickly approaching its transaction-handling capacity and was expensive to upgrade. In addition, business intelligence requirements were beyond what existing solutions could provide. New complexity around transactions and offers did not fit cleanly into existing business rules and reporting constructs.

### Solution

The Microsoft CSEO team planned to transition the Microsoft Sales revenue reporting platform from on-premises datacenters to Azure. This transition was an opportunity to reimagine and redesign the way Microsoft Sales functions in terms of infrastructure components and business benefits. Working with Microsoft design teams and the Azure Customer Advisory team, CSEO devised three options for application modernization:

**Lift and shift:** Lift and shift involves moving an existing on-premises environment into Azure as VMs. Essentially, this means maintaining the same structure and workload distribution, but



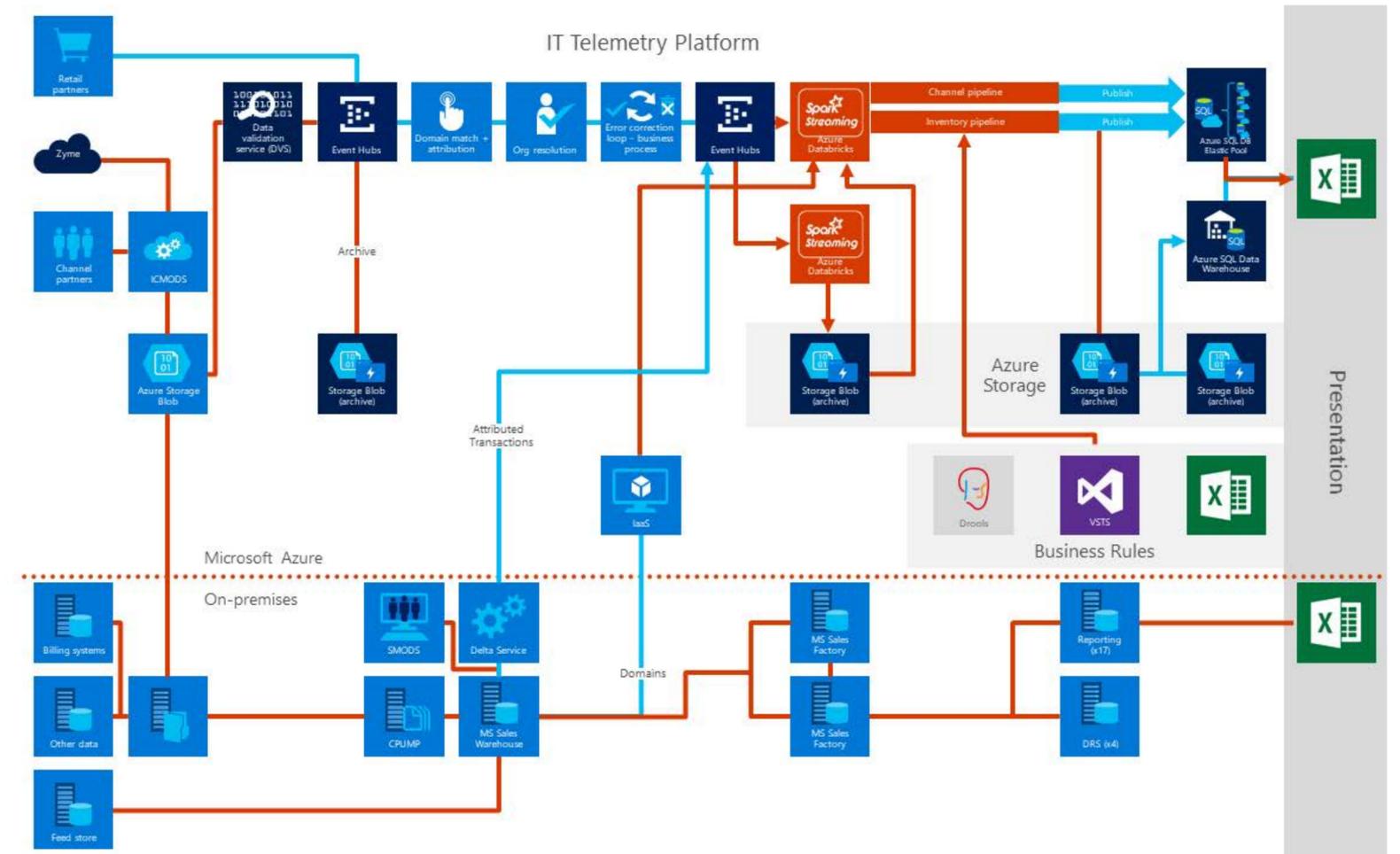
hosting in Azure. Lift and shift provides a low risk factor, but ultimately would force Microsoft Sales to maintain the same structure that was built 20 years ago.

**Microservices:** Establishing microservices would involve a significant redesign to Microsoft Sales, using Azure capabilities and functionality as the default for solution architecture. Breaking Microsoft Sales into microservices could establish a denser compute structure and optimize data flow and generation capabilities. Individual microservices could be treated as atomic units that can be added, removed, or modified.

**Big data:** CSEO understood that the robust data processing capabilities of Azure could be highly beneficial to Microsoft Sales. The inherent parallelism in Azure could drive scale, while also reducing risk. With an array of Azure big data solutions, including Spark and Azure Data Lake Store, the team could use different pieces of technology to complete the overall Microsoft Sales puzzle.

After analyzing these possibilities, CSEO decided that the big data option would meet the maximum requirements. With Azure, they could rethink data distribution and consumption, along with what the data flow looks like. Azure big data services provide a greater level of granularity and native support for data manipulation, leading to more parallel processes and quicker data delivery to end users.

The data flow components in the final Microsoft Sales solution may include Azure Databricks for Apache Spark based analytics processing, Azure Event Hubs, Azure SQL Database, and Azure SQL Data Warehouse. After the redesign is complete, Microsoft Sales can anticipate increased processing and reporting capabilities—and an explosion in transaction size and data up to ten-fold, which will account for the different ways they plan to ingest, process, and distribute data.



# Microsoft Sales has gone from a 24-hour window to less than 45 minutes—an improvement of more than 95%

## Business Impact

Although Microsoft Sales is still migrating, the new platform has already produced several benefits. Microsoft Sales can handle larger transaction volumes with a sublinear correlation in end-to-end time. The team has moved away from the monolithic nature of the earlier code base; they now incorporate modern engineering practices into development, allowing them to add new features quickly and redefine their processes to fit business demands rather than technical limitations. This

modernization has drastically reduced end-to-end latency in the sales process: Microsoft Sales has gone from a 24-hour window to less than 45 minutes—an improvement of more than 95 percent.

## Lessons learned

While moving toward modernization, Microsoft Sales identified several “Lessons learned” that can help shape their solution going forward:

### Understanding of the distributed nature of Azure functionality

Functionality doesn't have to be restricted to the feature set of a single platform. Azure enables flexibility for adopting the right solution for the task at hand.

### Natural movement toward modern engineering principles and practices

Examining Azure has refocused the developer mindset toward modern engineering principles, moving from online analytical processing to big data. This affects how code is written and how solutions are targeted and planned. It also presents an exciting learning opportunity for team members' professional growth.

## Workers who are better tooled for future investment

With a more agile and flexible solution based in Azure, development and DevOps understand modern engineering principles, are familiar with Azure development processes, and are better equipped to move the organization forward—from both the engineering and business perspectives.

## Benefits of using big data for financial and structured data

By testing and examining the Azure environment and available solutions, Microsoft Sales was able to conclude that Azure can help to solve the unique and complex problems presented by financial data.

## Ability to create the rules management process as a discrete component

As a discrete component, the rules management process is much simpler. Rule changes can be made and deployed in production with zero downtime.

## Learn more about application rearchitecting by exploring these resources

[Rearchitecting for the cloud with Robert Venable](#)

[Redesigning our revenue reporting system for cloud architecture](#)

## Commonly used services

[Azure App Service](#)

[Azure Kubernetes Service](#)

[Azure Functions](#)

[Azure API Management](#)

[Azure Logic Apps](#)

[Azure SQL Database](#)

[Azure Databricks](#)

[Azure Analysis Services](#)

# Rebuild: go cloud native

## **Accelerate your development and deployment with cloud native services.**

When building greenfield applications in this Rebuild phase, consider cloud native technologies to leverage the high productivity with PaaS along with rapid application development. IoT (Internet of Things) and Blockchain are great examples of born-in-the-cloud applications.

Azure offers a comprehensive set of cloud platform services that make it easy to navigate architectural approaches and design patterns for building modern apps. It delivers numerous options for application development and deployment like Azure App Service, Azure Kubernetes Service, Service Fabric and more.

In addition, Serverless architecture, a popular choice for cloud native applications, includes:

- Event-driven functions—without the need to explicitly provision or manage infrastructure—with Azure Functions
- Globally distributed, multi-model databases with Azure Cosmos DB.
- Use Azure Functions and Azure Cosmos DB to build scalable, serverless applications.
- Highly available and redundant storage with Azure Storage.
- More than 200 out-of-the-box connectors using Logic Apps to integrate apps, data, systems, and services.

Case study 4 /

# Rebuilding in action: creating an integrated plug-and-play supply chain with serverless computing

### Situations & challenges

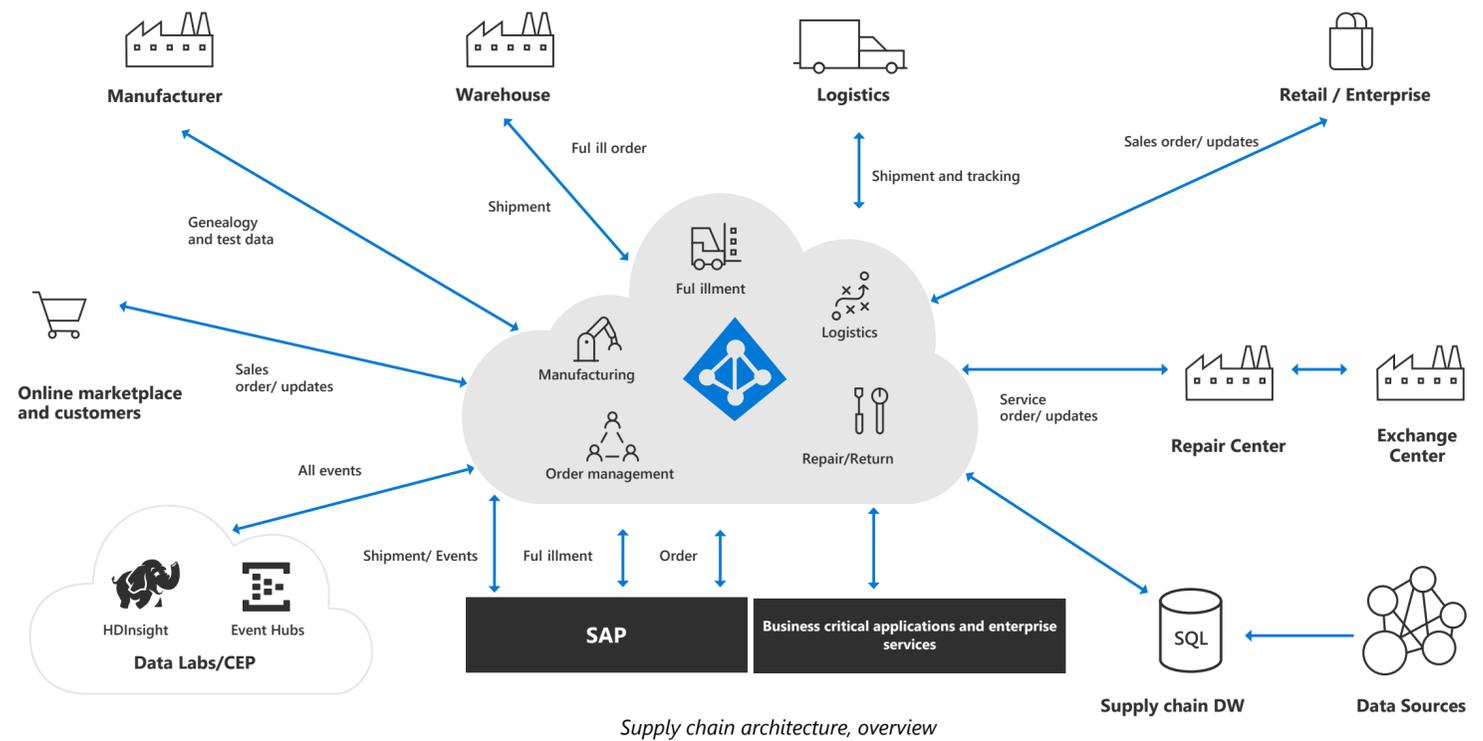
The Microsoft supply chain engages with hundreds of vendors that manufacture and distribute Microsoft products around the world. Partners must be integrated into the Microsoft network and communication stream to create a properly functioning supply chain. The original solution for integrating partners into the Microsoft Devices Supply Chain (MDSC) team was built on an infrastructure that included approximately 200 on-premises servers handling more than 3 million transactions per day. Adding a new partner to the supply chain was a time- and labor-intensive process, as new code needed to be developed for each partner integration. The costs to run infrastructure, upgrades, and maintenance were too high. Plus, visibility into the supply chain process

was limited by the ability to collect telemetry data from partner systems that were either inadequate or unreliable.

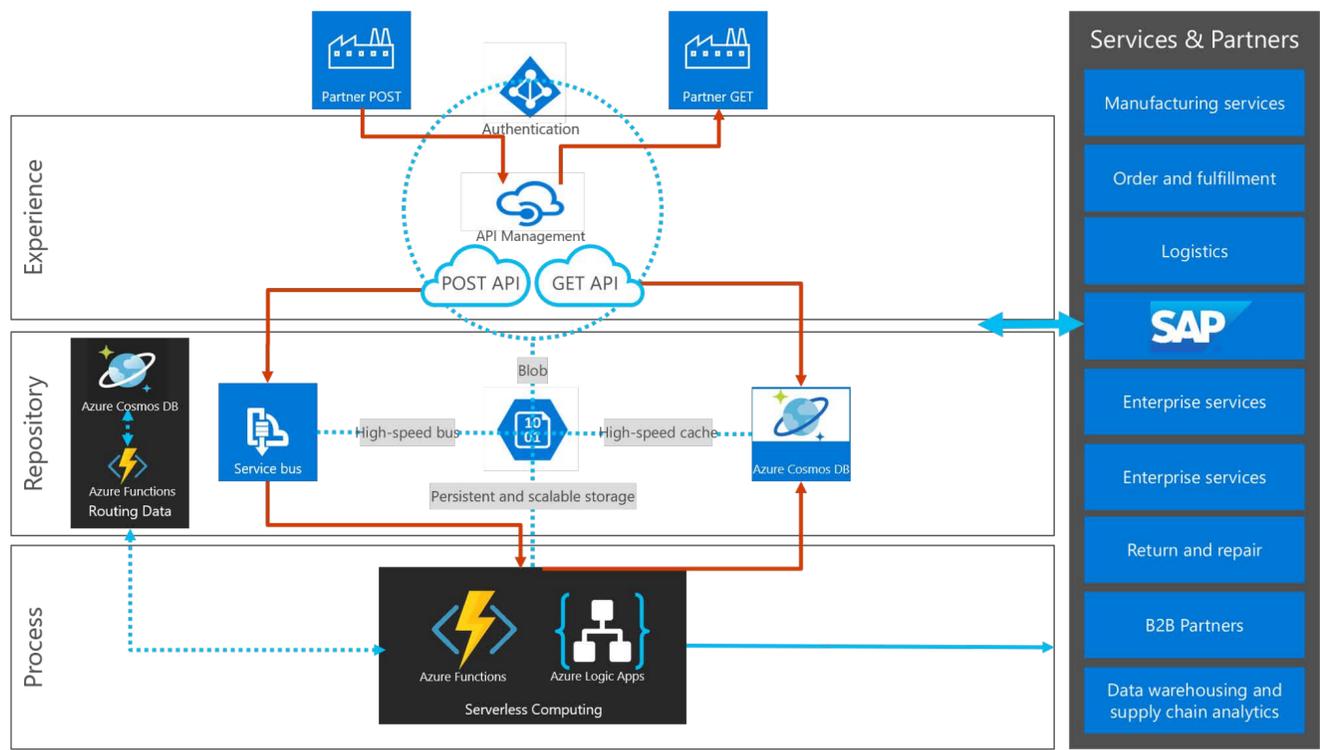
### Solution

The Microsoft CSEO team decided to build a completely new solution to enable digital transformation. They developed a short list of goals for the new supply chain solution and set about creating the core design principles for an integrated supply chain platform. CSEO used microservices and serverless computing in Azure to build the new solution, which allowed the MDSC team to create a simpler, faster, and more dynamic supply chain management process. Now it's easier for partners to onboard, interact, and conduct business with Microsoft,





Supply chain architecture, overview



Supply chain, serverless computing

and MDSC has end-to-end control and visibility. Currently, they're expanding this transformation by exploring technologies like Blockchain, Internet of Things, and cognitive bots. These investments give Microsoft a competitive edge in the devices supply chain and bring maximum value to partners and customers.

Importantly, to offer plug-and-play partner integration and create a manageable end-to-end solution, MDSC had to change the way partners fit into the supply chain. Plug-and-play strategy was a significant driver of the architecture design. CSEO added five integration layers for supply chain components, including partners, marketplace, services edge, ERP and processing, and data warehouse and analytics. All integrations use REST API standards that help Microsoft and its partners connect to endpoints and exchange data as effortlessly and securely as possible. Likewise, all APIs are implemented with Azure API Apps, Azure Functions, and Logic Apps based on the capability needed, latency, and throughput requirements. To create a truly modular, microservices-based architecture, the team used several practical design patterns that helped keep integration both efficient and secure

Cosmos DB helps contribute to an event driven application with change feeds, making it a plug & play component of this Serverless architecture. The service can be easily integrated into Azure functions & Logic Apps with "zero" code. A 100% hassle-free way to build world class application in no time. Elimination of trivial infrastructure maintenance has also enabled the team to transition towards DevOps culture with ease.

**Business Impact**

A plug-and-play provisioning platform benefits both Microsoft and its partners. With a streamlined, trustworthy, and repeatable onboarding process, partners enjoy a more modular way of engaging with Microsoft. The new solution reduces on-boarding for new vendors and partners from weeks to minutes. The new supply chain platform built on PaaS architecture and microservices reduces management overhead, as there are no servers to patch and manage. It also allows CSEO to change small parts of the solution without impacting the entire app. In addition, the team can use infrastructure as code to quickly provision and manage new resources, and the native scaling capability of Azure PaaS can handle volumes that the previous solution simply could not.

The solution reduces onboarding for vendors and partners from weeks to minutes—helping them focus on business innovations to gain speed and scalability, instead of managing servers and infrastructure.

Supply chain is a low margin business requiring a high order of agility to integrate with new partners and is extremely sensitive to data visibility needed for operational performance.

Azure Serverless compute and storage options like Functions, Logic Apps and Cosmos DB have provided the fastest way to connect, integrate and execute business processes with minimal cost and seamless implementation.

Data streaming solutions like Event Hubs, Stream Analytics along with data analytics services like HDInsight and Data Lake helps provide real-time visibility allowing the team to make informed business decisions using an Azure-powered intelligent cloud.

By migrating supply chain services to the cloud, MDSC is delivering a better solution to partners—one that is instantly accessible through a resilient, global presence.

May  
2018

Expedition Cloud: inside Microsoft's cloud migration and application development journey

32

### Lessons learned

While moving toward modernization, the MDSC team identified several “Lessons learned” that can help shape their solution going forward:

Shifted paradigm to a plug-and-play model  
This transformation has helped to shift the MDSC paradigm from a monolithic application model to a plug-and-play model that results in a dynamic and demand-driven supply chain.

Generated ideas as a team to drive changes  
Routine design whiteboard sessions provided opportunities for discussing new ideas around improving business processes.

### Understood design patterns to get truly modular

To create a truly modular, microservices-based architecture, the CSEO team used several practical design patterns to help keep integration both efficient and secure.

### Identified opportunities for further transformation

The MDSC team is considering technologies like Blockchain, Internet of Things, and cognitive bots to further transform their solution and deliver maximum value to partners and customers.

Learn more about application rearchitecting by exploring these resources:

[Creating an integrated plug-and-play supply chain with serverless computing](#)

### Commonly used services

[App Service](#)

[Azure Kubernetes Service](#)

[Azure Functions](#)

[Azure API Management](#)

[Azure Logic Apps](#)

[Azure SQL Database](#)

[Azure Cosmos DB](#)

[Azure Bot Service](#)

[Cognitive Services](#)

# Replace: use business software as a service

**Use managed yet configurable applications to replace your legacy apps—without building or customizing application code.**

Legacy applications based on traditional models and older technologies usually require upgrading, along with new investment to enhance features and capabilities. SaaS cloud applications provide a better alternative to replacing legacy apps, helping your organization go to market more quickly and eliminate the need to build or customize application code.

Microsoft provides managed yet configurable applications to replace your legacy applications for modernization. You don't need to purchase, install, update, or maintain any hardware, middleware, or software—just subscribe to the complete software solution on a pay-as-you-go basis.

You can mobilize your workforce with these managed apps, which can be accessed from any internet-connected endpoint or mobile device—all without the need to make code changes or develop different applications for different device platforms.

Microsoft managed enterprise applications include:

**Dynamics 365** for ERP and CRM, powered by data and intelligence to capture new business opportunities.

**Office 365** for messaging, collaboration, and other productivity tools.

**Power BI** for business analytics requirements, with interactive visualizations and business intelligence.

**PowerApps** for building custom business applications that connect to your data and work—without the time and expense of custom software development.

Case study 5 /

# Replacing in action: simplifying the sales platform and deploying Dynamics CRM Online

### Situations & challenges

CRM systems at Microsoft are a critical component of the overall business, managing almost 50 percent (\$40B annually) of total Microsoft revenue. Microsoft wanted to simplify its direct sales platform across the globe because existing sales processes were overly complex, the customized CRM system was high maintenance, and related tools and technologies were disjointed. Instead of fully focusing on connecting with customers, sellers were spending as many as 1.5 days per week completing administrative processes and tasks. Integration and changes made across multiple systems slowed bi-directional data synchronization across all CRM instances and associated tools, and synchronization required

a four-hour SLA. It took months to update apps, yet they still included little automation for testing processes or deployment tools.

### Solution

Based on critical feedback and lagging sales productivity, Microsoft chose to both simplify the sales process itself and deploy a new CRM system to support the process. First, Microsoft radically simplified its direct sales process. Then, it mapped the process to features of Dynamics CRM Online and enabled native Office 365 apps and collaboration tools. Microsoft built and deployed a cloud-first, mobile-first platform that supports seller-specific customer relationship management (Dynamics CRM Online), business intelligence,

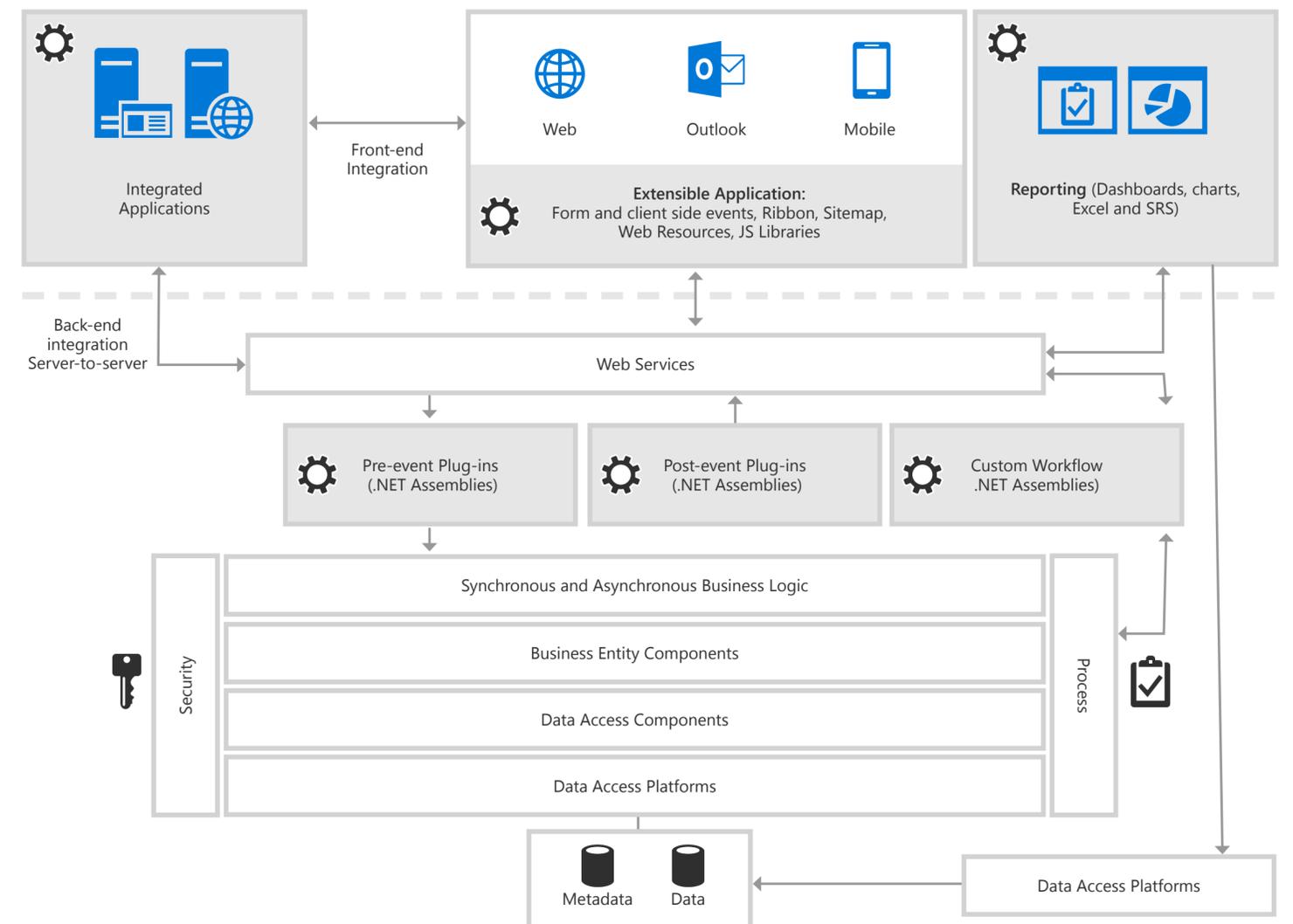


and connected seller tools (using Windows App Portal for Sales). Native features of Dynamics CRM Online helped to create a framework with built-in agility and speed to support continually evolving business and data management needs. The platform uses an Azure service bus to receive messages from Dynamics CRM Online and leverages queues to ensure that messages are not dropped. At peak times, autoscaling initiates to rapidly process queued messages.

The CRM platform deployment was phased and automated as much as possible, and the solution rolled out globally in nine months, with multiple sprints. The rollout of Dynamics CRM Online has fundamentally changed the support relationship between CSEO and the business group. Because the system is largely built out of the box, bugs are routed directly to the Dynamics CRM product development group for resolution, and the product group applies fixes to get services up and running. Today, more than 25,000 sellers and sales operations staff use the system in more than 100 countries.

**Business Impact**

The new sales platform and process deliver notable advantages to Microsoft teams, including better cost, productivity, and scalability. Dynamics CRM Online runs more users on a single instance and immediately scales to facilitate large sets of data—yet this system is significantly faster than the legacy system. The capability for bulk data import is much faster now, requiring only days to create a new template and import the enterprise data management capability (instead of months with the legacy system). The legacy customized method cost \$1 million; the new system costs one-tenth as much. Because data is now immediately synchronized between Dynamics CRM Online and the Azure ODS, there's a single version of data across the system, and latency doesn't generate data inconsistencies. In addition, because Microsoft used out-of-the-box features of Dynamics CRM Online, CSEO



is now a software service provider and acts as a liaison between the business group and the product development group. Manual fixes are no longer necessary due to the lack of custom code. Plus, the modular nature of the Windows App Portal for Sales allows teams to work in parallel and produce more. While requests in the original system would be queued for long periods, applications now can be rapidly developed and plugged into the portal.

#### **Lessons learned**

While moving toward modernization, Microsoft teams identified several "Lessons learned" that can help shape the solution going forward:

#### **Simplified process and business rules first**

CSEO took the time to partner across the enterprise and deeply understand business requirements and processes. The team then reduced hundreds of business rules down to a handful and greatly simplified the sales process across the company.

#### **Implemented data hygiene**

This activity involves proactively executing premigration, data audits, and cleansing to improve data hygiene before migration. For Microsoft, strong data hygiene improved data relevance to sellers after migration.

#### **Applied parallelism**

For major subsidiaries, parallelism reduced the time required for data migration from what would have been weeks to a three-day weekend. Eight million US records were processed in 13 hours.

#### **Ensured sufficient bandwidth**

While importing large amounts of data to CRM, the CSEO team learned to ensure sufficient bandwidth, as sometimes CRM services may throttle the data import to prevent a denial of service attack.

#### **Migrated organizations**

The team migrated entire units of sales organizations simultaneously instead of subcomponents of a specific sales organization.

#### **Partnered with business teams**

CSEO learned to move quickly with a small team at first. They created a base platform, and then engaged business stakeholders about what the system was trying to achieve. The team also obtained sponsorship at the executive and local levels. Finally, they set aside time to talk through changes with local teams and focus on user-centered design, finding that it's easier to get adoption if people are already in agreement with the design.

**Learn more about application rearchitecting by exploring these resources:**

[Microsoft simplifies its sales platform and deploys Microsoft Dynamics CRM Online](#)

#### **Commonly used services**

[Azure Functions](#)

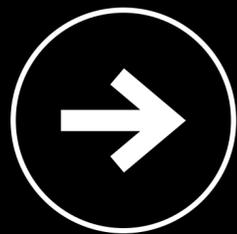
[Azure Logic Apps](#)

[Azure Cosmos DB](#) [Azure](#)

[Bot Service](#) [Azure](#)

[Cognitive Services](#)

# We meet you where you are. Azure, cloud for all.



Get help with your cloud migration. Contact  
an Azure specialist at [aka.ms/azurespecialist](http://aka.ms/azurespecialist)

Application modernization is a journey to digitally transform your business applications to make them future-ready. Modernizing your legacy applications through proven DevOps practices leads to teams that are equipped for better productivity, agility, flexibility, and operational efficiency. Microsoft did it, so can you.

Assess your application portfolio, categorize the applications properly, have a strategic plan, and start your journey to the cloud. The journey includes selecting from among five tested migration and modernization approaches—Rearchitect, Rebuild, Rehost, Refactor, and Replace—and combining them in a way that makes the most sense for your business requirements. Azure provides robust cloud offerings that deliver an end-to-end solution—from strategy and planning through deployment, migration, and ongoing support.

Find inspiration, tools and guidance on how to lead your business through its digital transformation. Download the free e-book, *Designed to Disrupt* at <http://aka.ms/disruptebook>