



Azure webinar series

Migrate your .NET  
web applications to  
Azure for scale and  
agility



# Welcome

## How do I ask a question?

If you have a technical or content-related question, please use the Q&A window

We will address the questions as they come in

## Can I view this presentation after the webinar?

Yes, this presentation is being recorded

A link to the recorded presentation will be sent to the email address you used to register

# Meet our speaker



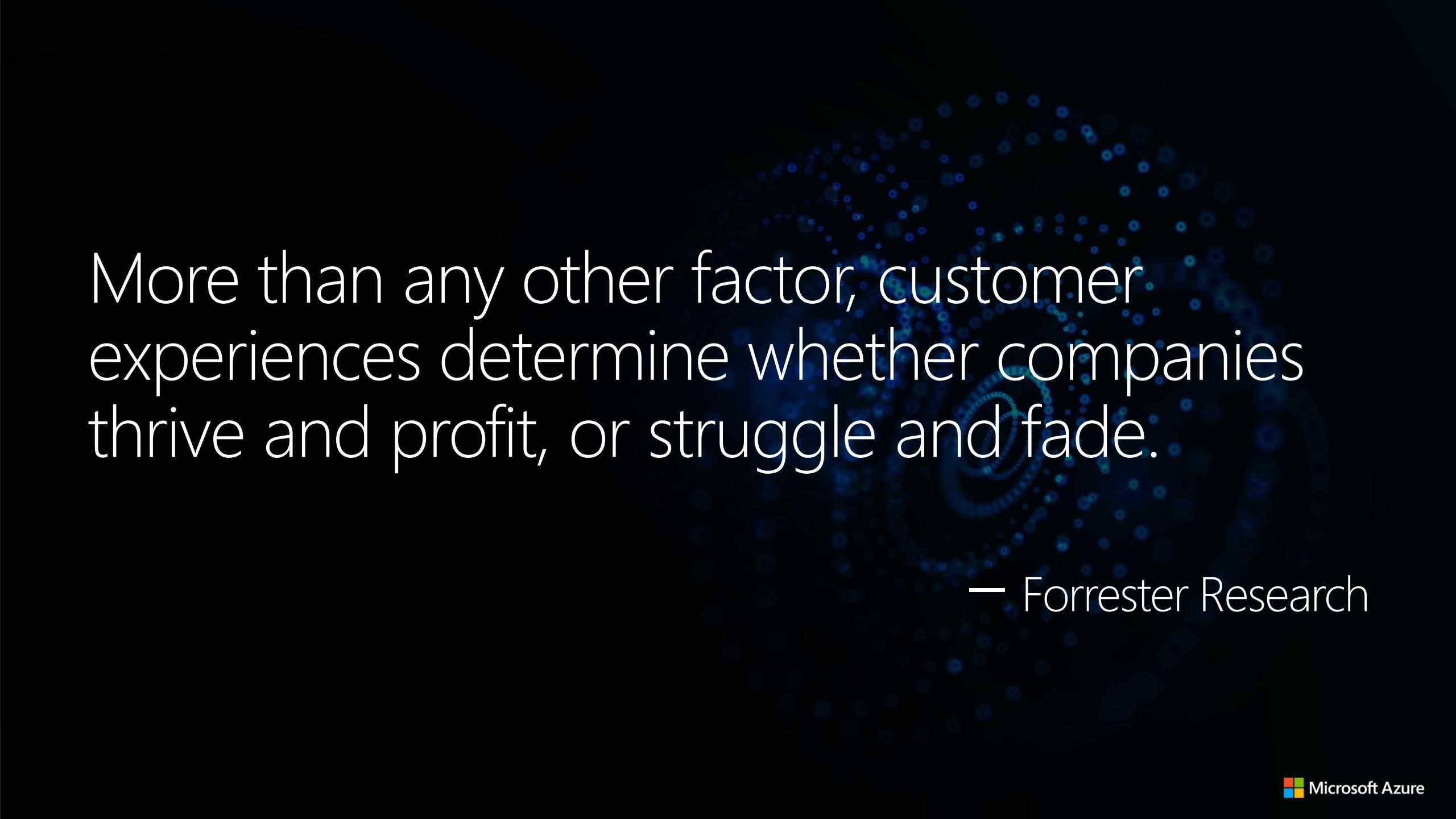
## Jay Schmelzer

Director, Developer Division Customer  
Engagement Team

[JaySch@Microsoft.com](mailto:JaySch@Microsoft.com)

# Agenda

- Customer expectations vs. reality
- .NET Momentum & Roadmap
- Migration & modernization strategies
- Migration & modernization case studies



More than any other factor, customer experiences determine whether companies thrive and profit, or struggle and fade.

— Forrester Research

# Your **customers** have high expectations

## Speed

79% won't return to a slow website

## Personalization

38% won't call again if they have to repeat themselves

## Cross-device

65% get frustrated with an inconsistent experience



# What does it take to realize these benefits?





Open & Scalable  
Cloud Platform



Data-driven  
Intelligence

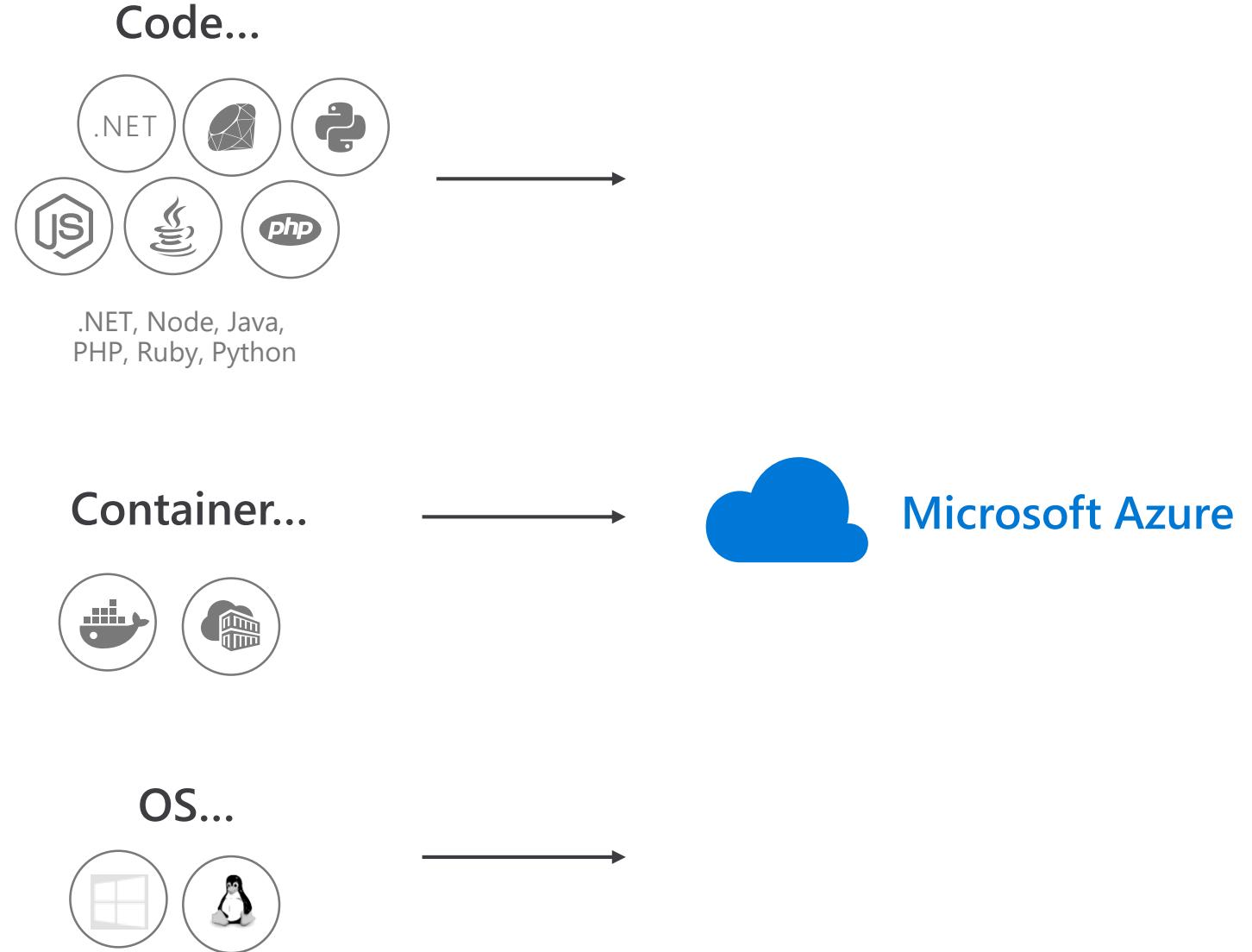


Cross-platform  
Experiences

Continuous Innovation

# Azure is for all your web apps

Use the framework,  
container, or OS of your  
choice on a fully managed  
platform.

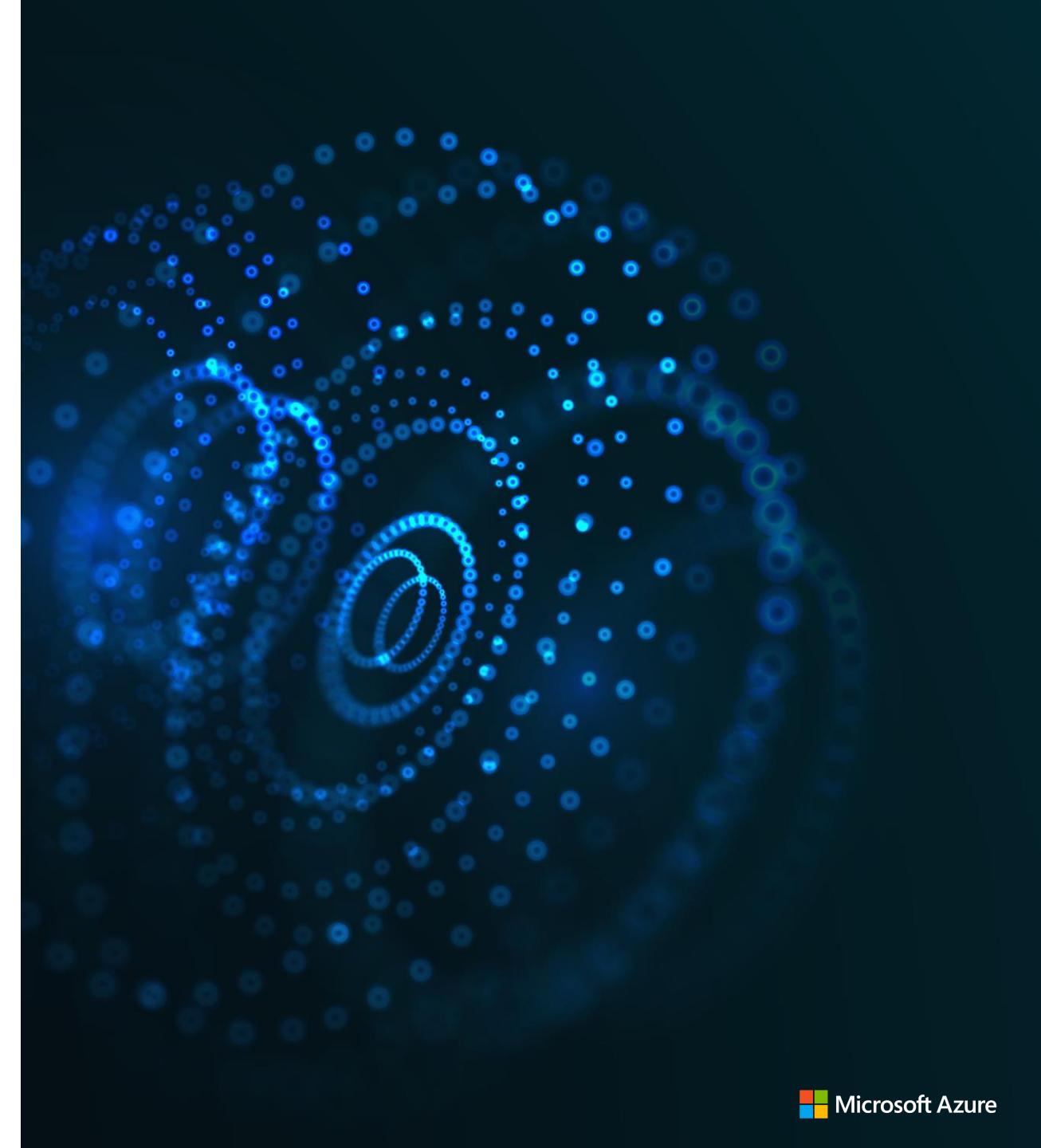


		 Save money. Live better.			
<b>Rockwell Automation</b>		 Johnson Controls	 Empowering Healthcare	 Lufthansa	 SAMSUNG
 HEINEKEN	 mazda	 SKANSKA	 Ford	 ROLLS ROYCE	 xeroX
 Honeywell	 Schlumberger	 Building a better working world	 CBC & Radio-Canada	 Alaska	
 ThyssenKrupp	 FUJITSU	 TEMENOS The Banking Software Company	 Wellmark BlueCross BlueShield	 esri	 HarperCollins

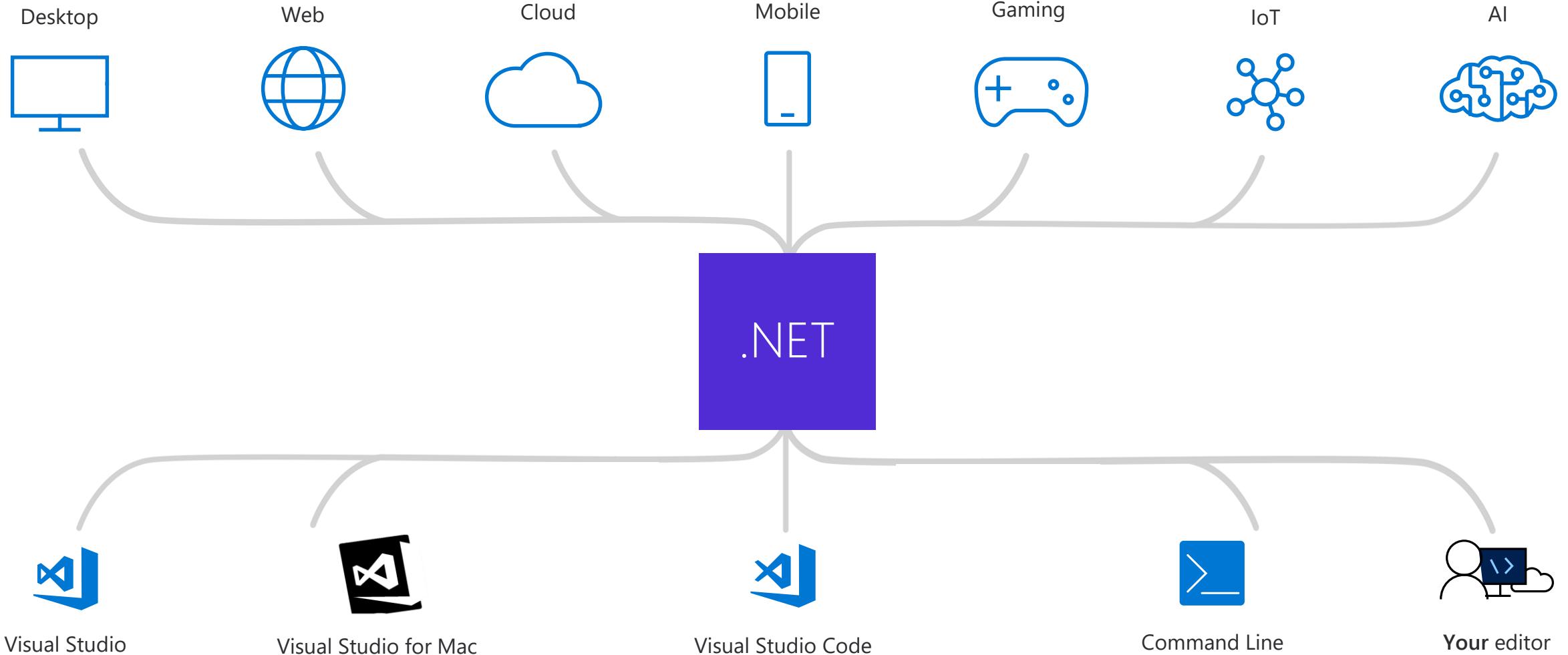
>90%

of Fortune 500 use  
Microsoft Azure

# .NET Momentum & Roadmap



# Your platform for building **anything**





Open source & cross platform



750K

.NET Core developers



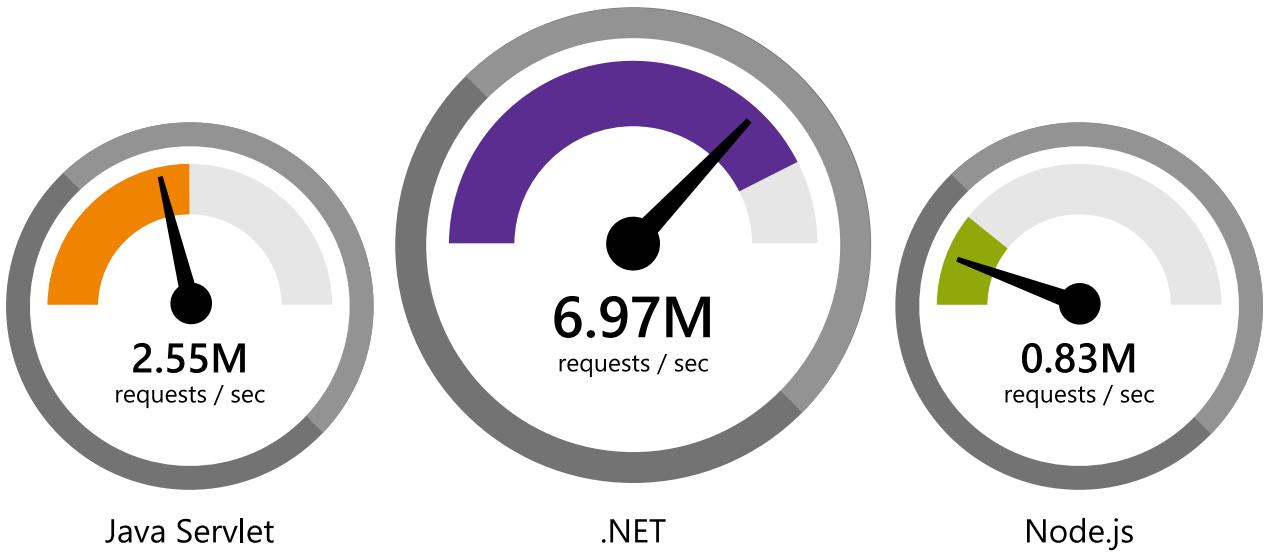
+1M

New .NET developers  
in last year

# .NET

Faster than any popular framework

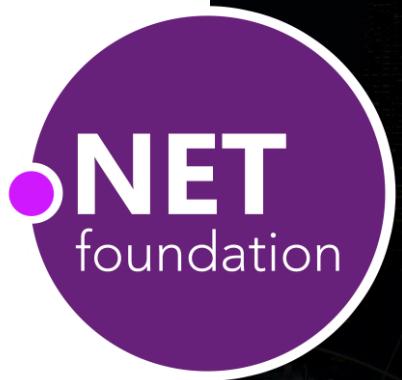
"Using the same-size server, we were able to go from 1,000 requests per second per node with Node.js to 20,000 requests per second with .NET Core." — Raygun



[www.dot.net/customers](http://www.dot.net/customers)

Data sourced from official tests available at [TechEmpower Round 16](https://techEmpower.com).

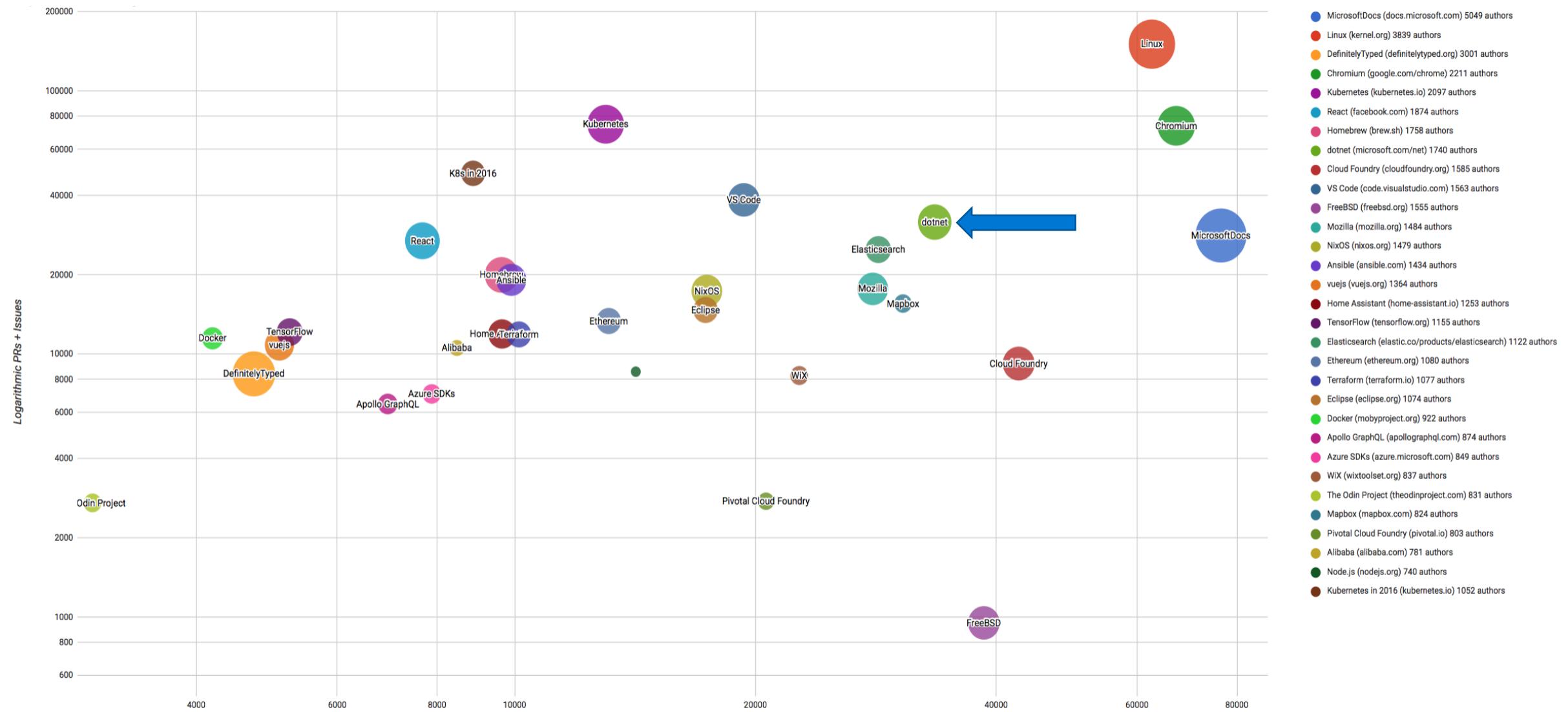
# .NET open source momentum



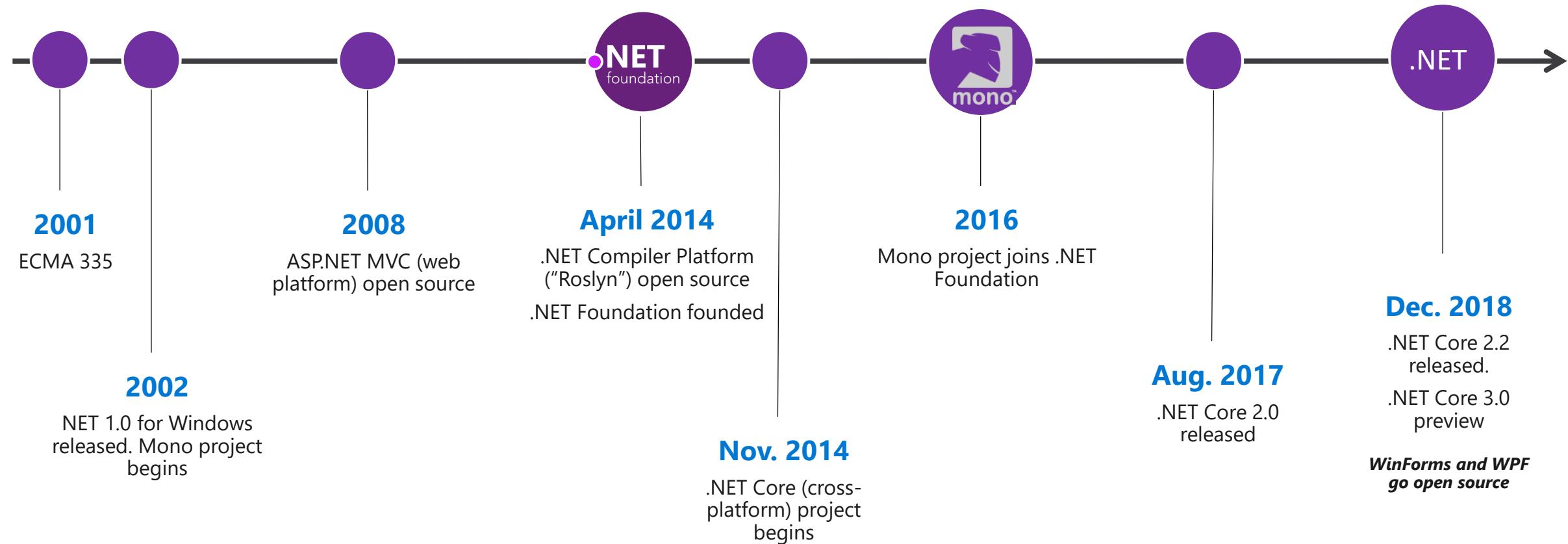
87%  
contributors outside Microsoft

61,000+  
accepted PRs from community

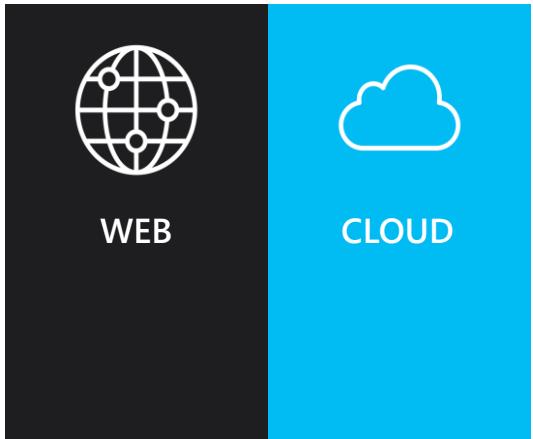
# Top 30 Highest Velocity OSS Projects on GitHub



# .NET Open Source Journey



# .NET Core 3



.NET Core is perfectly suited for the requirements of cross-platform workloads



.NET Core 3

LIBRARIES

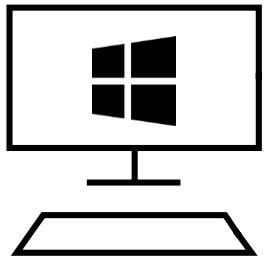
INFRASTRUCTURE

RUNTIME COMPONENTS

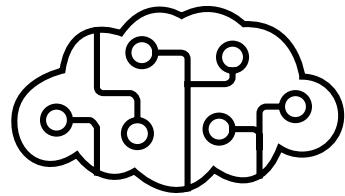
COMPILERS

LANGUAGES

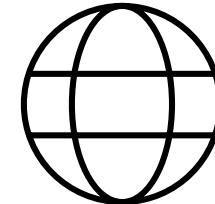
# .NET Core 3.0 themes



**Desktop Workloads  
& UI Interop**



**Artificial Intelligence  
& Machine Learning**

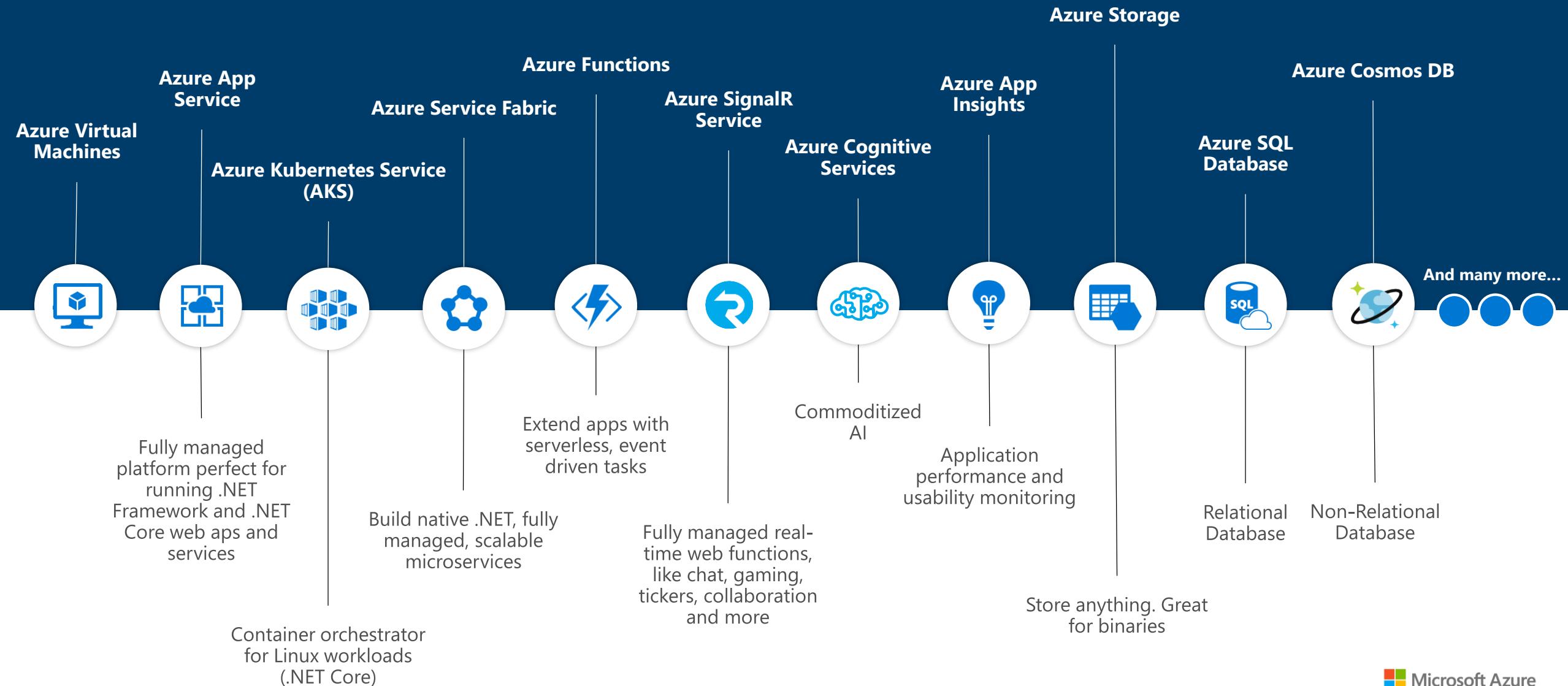


**Web App Development  
Productivity & Microservices**

# The **future** of .NET Framework

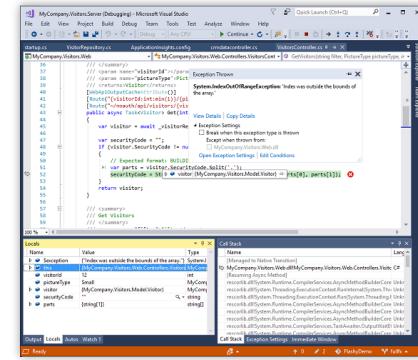
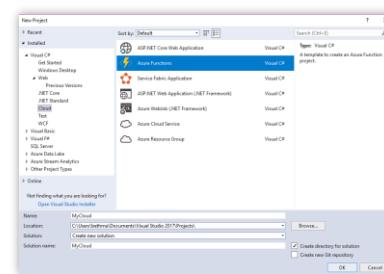
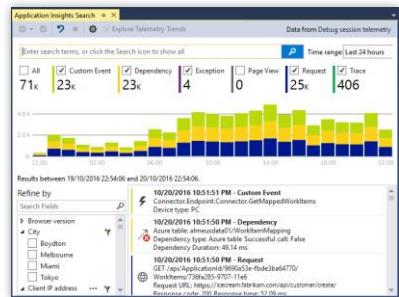
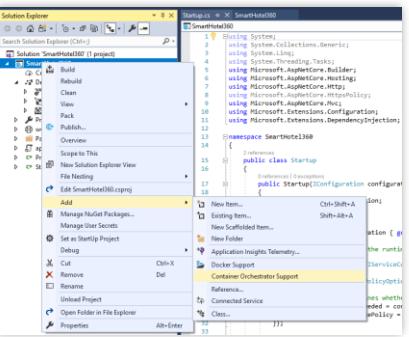
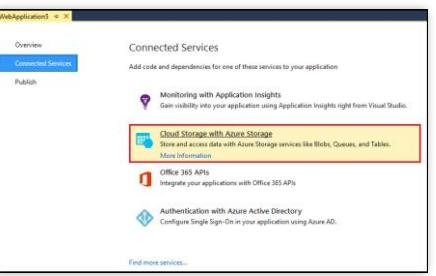
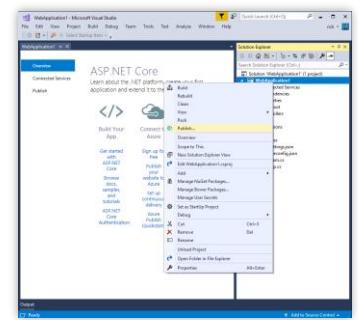
- .NET Framework is slowing down
- Why? We want to stop breaking customers apps
  - Billions of installs. Getting much harder to release major features without compatibility issues
  - Can innovate much faster with .NET Core because of its side-by-side nature
- We will continue to update
  - Many major product lines within Microsoft depend on .NET Framework
  - We will continue to update .NET Framework for years to come. High-compat features only.
  - Things like... new security protocols, bug fixes, Windows features, etc.
- No changes to support policy
  - Support for 10 years (5 mainstream+5 custom) per Windows support policy for each release
  - .NET Framework will remain a Windows component and service policy is the same as Windows
- Recommend new development on .NET Core

# .NET cloud apps run best in Azure





## Accelerate your cloud development



### Seamless Azure Integration

Publish directly to Azure without leaving the IDE

### Connected Services

A new experience for easily discovering and extending your app with cloud services

### Container Tools

Quickly add Docker compose and container orchestrator support to your apps

### App Insights with Code Lens

Get a 360° view of your app that includes availability, performance and user behavior

### Azure Functions

Get started quickly with serverless using templates and tools for development and debugging

### Snapshot Debugging

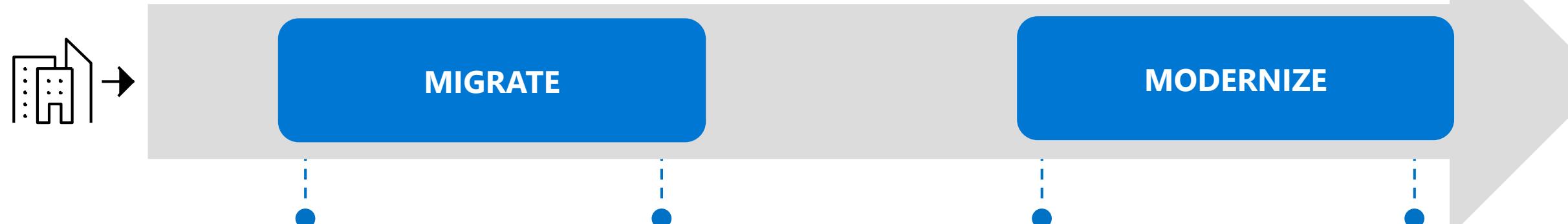
Production debugging your cloud apps has never been so simple



# Migration & modernization strategies



# Cloud migration vs. modernization



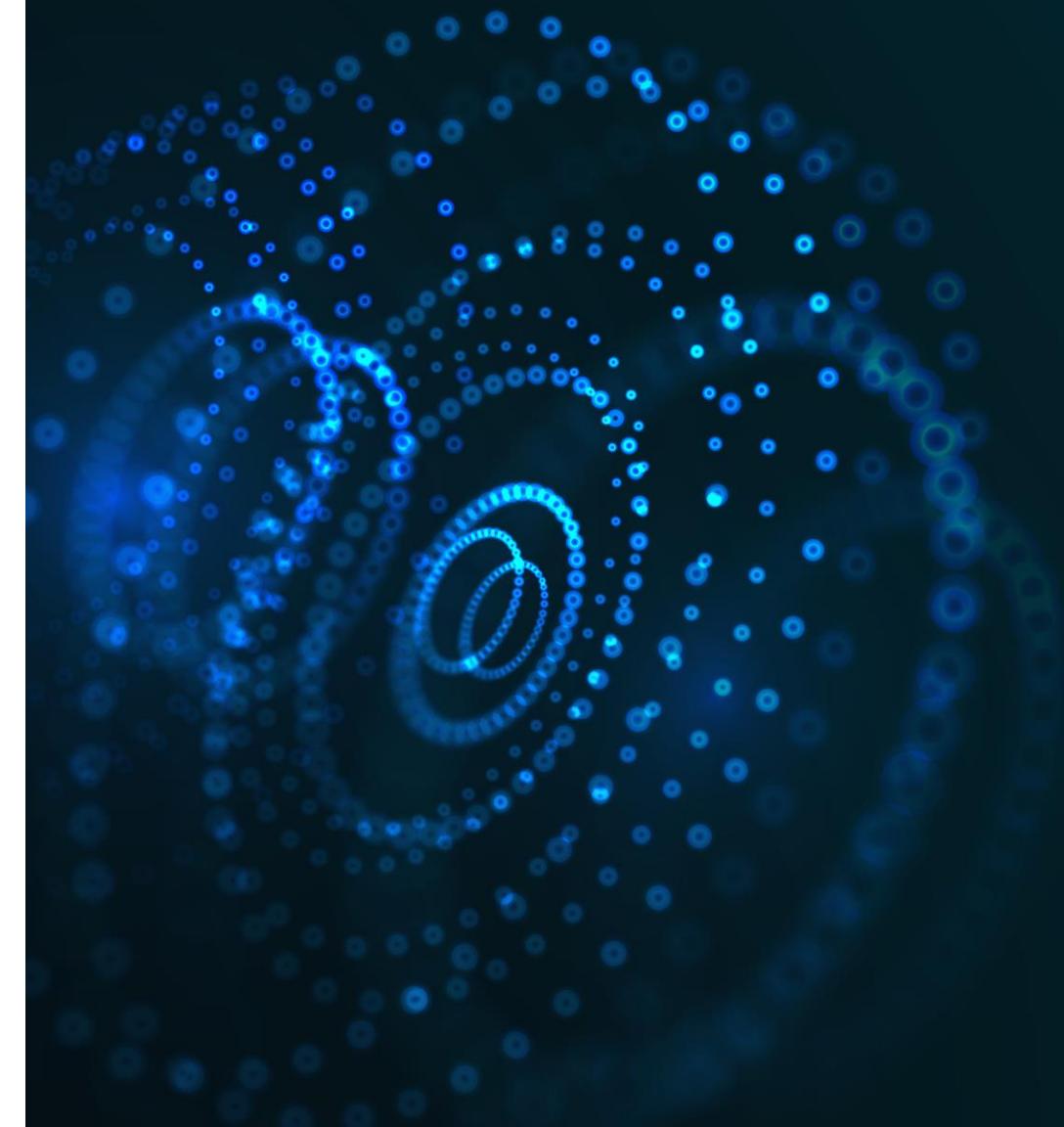
	MIGRATE	MODERNIZE		
Description	Redeploy as-is to cloud	Minimally alter to take better advantage of cloud	Materially alter/decompose application to services	New code written with cloud native approach
Drivers	<ul style="list-style-type: none"><li>Reduce Capex</li><li>Free up datacenter space</li><li>Quick cloud ROI</li></ul>	<ul style="list-style-type: none"><li>Faster, shorter, updates</li><li>Code portability</li><li>Greater cloud efficiency (resources, speed, cost)</li></ul>	<ul style="list-style-type: none"><li>App scale and agility</li><li>Easier adoption of new cloud capabilities</li><li>Mix technology stacks</li></ul>	<ul style="list-style-type: none"><li>Accelerate innovation</li><li>Build apps faster</li><li>Reduce operational cost</li></ul>
Technologies	IaaS	PaaS Containers	PaaS Serverless Microservices	
Results	IaaS: 435% ROI, 73% reduction in datacenter footprint and 83% reduced IT outsourcing cost. <sup>1</sup>	PaaS: 466% ROI, 80% time saved, 5.91M NPV, 50% faster deployments. <sup>1</sup> Container: 13x more releases, 10x cost reduction, 65% faster developer onboarding, 62% better availability. <sup>2</sup>		

# Migrating to the Cloud, Use Cases

Hosting existing ASP.NET web apps, APIs & WCF services on Azure

Cloud Infra Ready	Cloud Optimized Containers	Cloud Optimized
<b>IaaS</b> <b>(Virtual Machine)</b> <ul style="list-style-type: none"><li>✓ No re-architect or new code</li><li>✓ Least effort for quick migration</li><li>✓ Similar deployment model to on prem</li></ul>	<b>Windows Containers</b> <b>(App Service, Service Fabric, ACI, VM with Windows Containers)</b> <ul style="list-style-type: none"><li>✓ No re-architect, minimal code or config settings change</li><li>✓ Improved deployment and DevOps agility because self contained dependencies</li><li>✓ Automatic deployment with CI/CD</li><li>✓ High availability and scalability with Orchestrators</li><li>✓ Portability of apps through containers/docker, multi-cloud enabled</li></ul>	<b>PaaS</b> <b>(App Service)</b> <ul style="list-style-type: none"><li>✓ No infrastructure to manage</li><li>✓ Built-in high availability, scaling, and security</li><li>✓ Multiple versions of .NET supported</li><li>✓ Automatic deployment with CI/CD</li><li>✓ Advanced monitoring, diagnostics and debugging</li></ul>
<ul style="list-style-type: none"><li>x Manual patching and upgrade</li><li>x Manual scaling and availability</li></ul>	<ul style="list-style-type: none"><li>x Containerization has a learning curve</li></ul>	<ul style="list-style-type: none"><li>x Not all apps are eligible</li><li>x May require some code refactoring</li></ul>

# .NET Framework migration: [Try App Service](#)



# App Service differentiator

## Benefits of App Service for .NET Developers

High productivity	Fully managed	Enterprise grade
<ul style="list-style-type: none"><li>Live production debugging with Visual Studio Snapshot Debugger</li><li>App telemetry, anomaly detection, and site diagnostics with App Insights</li><li>Automatic OS and framework patching (.NET Framework &amp; .NET Core)</li><li>Site staging slots</li><li>Continuous integration/deployment with Git, Visual Studio, Docker Hub, and GitHub</li><li>Site extensions support &amp; gallery</li><li>Auto-healing</li><li>Logging and auditing</li><li>Admin-site</li></ul>	<ul style="list-style-type: none"><li>Automated deployment</li><li>AutoScale</li><li>Built-in load balancing</li><li>WW datacenter coverage</li><li>End point monitoring and alerts</li><li>App gallery</li><li>DR site support</li><li>WildCard support</li><li>Dedicated IP address</li><li>HTTP compression</li><li>CDN support for websites</li><li>App Services Environments</li></ul>	<ul style="list-style-type: none"><li>Hybrid connections/VPN support</li><li>Scheduled backup</li><li>Azure Active Directory Integration</li><li>Site resiliency, HA, and DR</li><li>Web jobs</li><li>Role base access control</li><li>Audit/compliance</li><li>Enterprise migration</li><li>Client certs</li><li>Cache</li><li>IP restrictions/SSL</li><li>Web sockets</li><li>SQL, MySQL, CosmosDB</li><li>Sticky sessions</li><li>Authorization/authentication</li></ul>

# What types of .NET apps are ideal for App Service?

## ASP.NET

Webforms, MVC, Web API

## WCF services

Http / Https bindings \*

**Self contained monolithic/N-Tier web apps & services talking to a database are particularly well suited for Azure App Service as well as apps that do not have complex dependencies.**

\* Supported WCF bindings: BasicHttp, WSHttp, NetHttpBinding, NetHttpsBinding, BasicHttpContextBinding, WebHttpBinding, WSHttpContextBinding

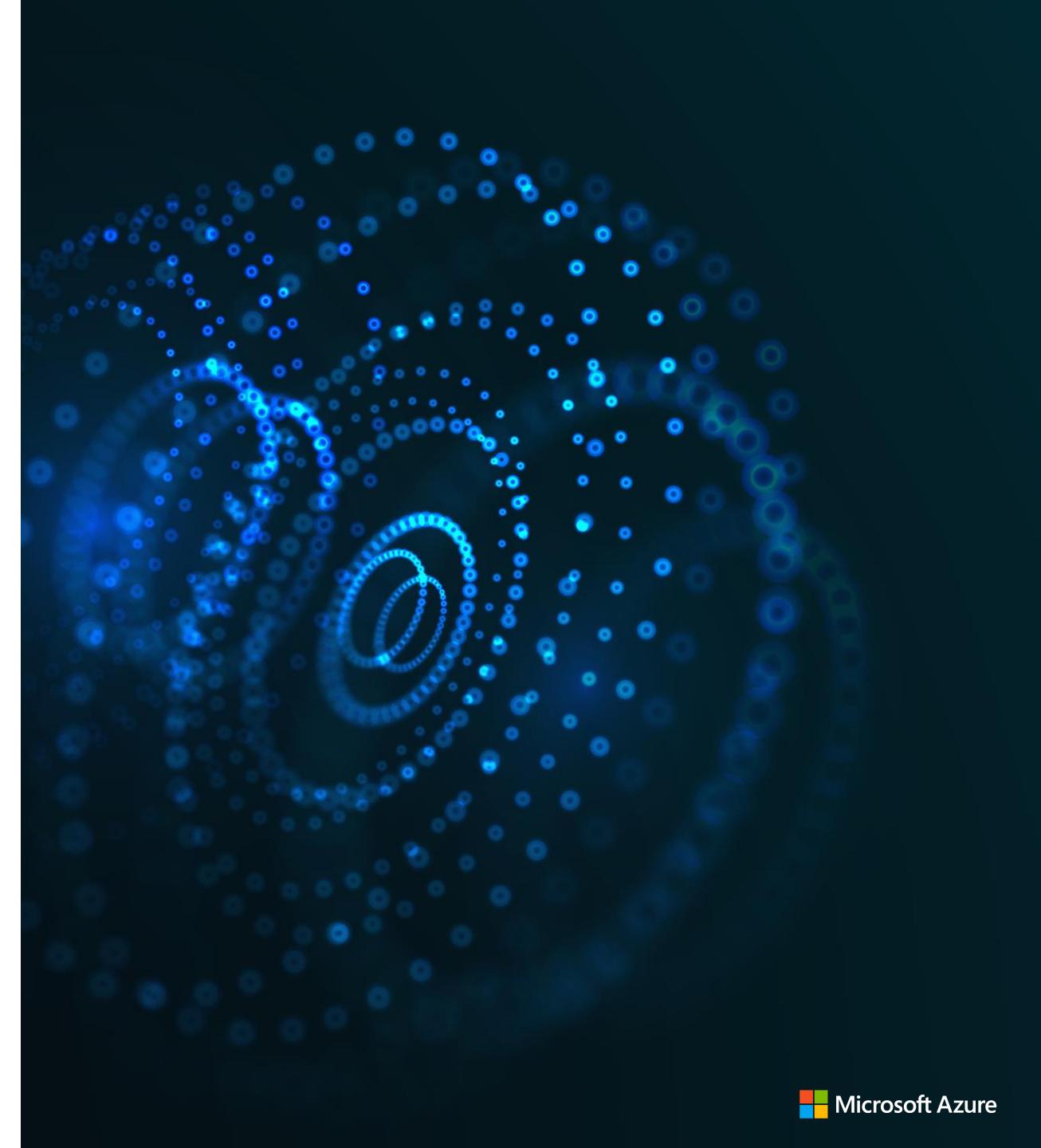
# Compatibility Checklist – Can my .NET app move?

## Checklist

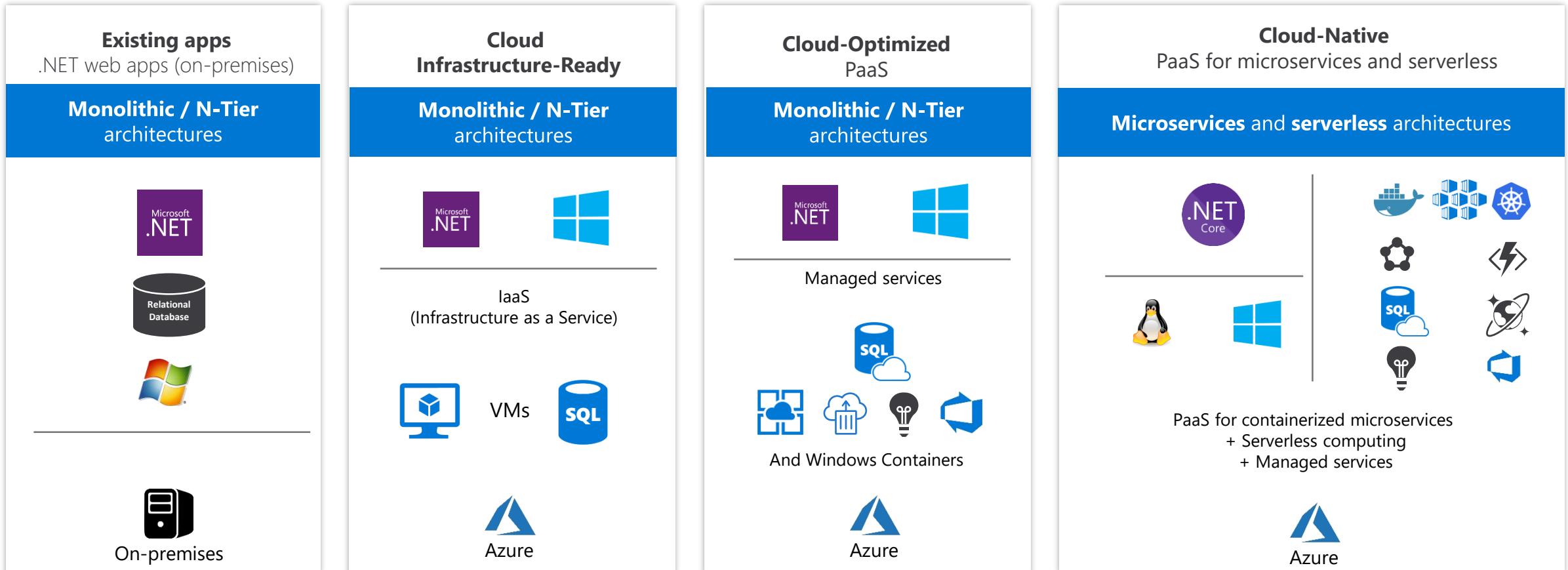
- ✓ Ensure port bindings 80 (Http) or 443 (Https)
- ✓ Anonymous & forms authentication supported. Windows auth with Azure Active Directory or ADFS & VPN
- ✓ No GAC (Global Assembly Cache) usage
- ✓ Single Application Pool per site
- ✓ IIS7+ Schema Compliance
- ✓ No COM/COM+ components
- ✓ No reference to physical directories. Use blob storage to mimic file access.
- ✓ Verify access to on-premises resources as they may need to be migrated/changed
- ✓ SQL Server, Oracle, MySQL databases
- ✓ ISAPI Filters DLL(s) need to be deployed locally and registered via web.config

Follow this checklist to determine if the app needs some refactoring

# .NET Core modernization: Phased approach to Cloud-Native



# Maturity model for .NET application modernization



**Base Cloud Environment and cross-cutting concerns:** Network, Hybrid-cloud, Identity/Auth, Cost control and Operations model

Migrate / Rehost

Minimal code changes

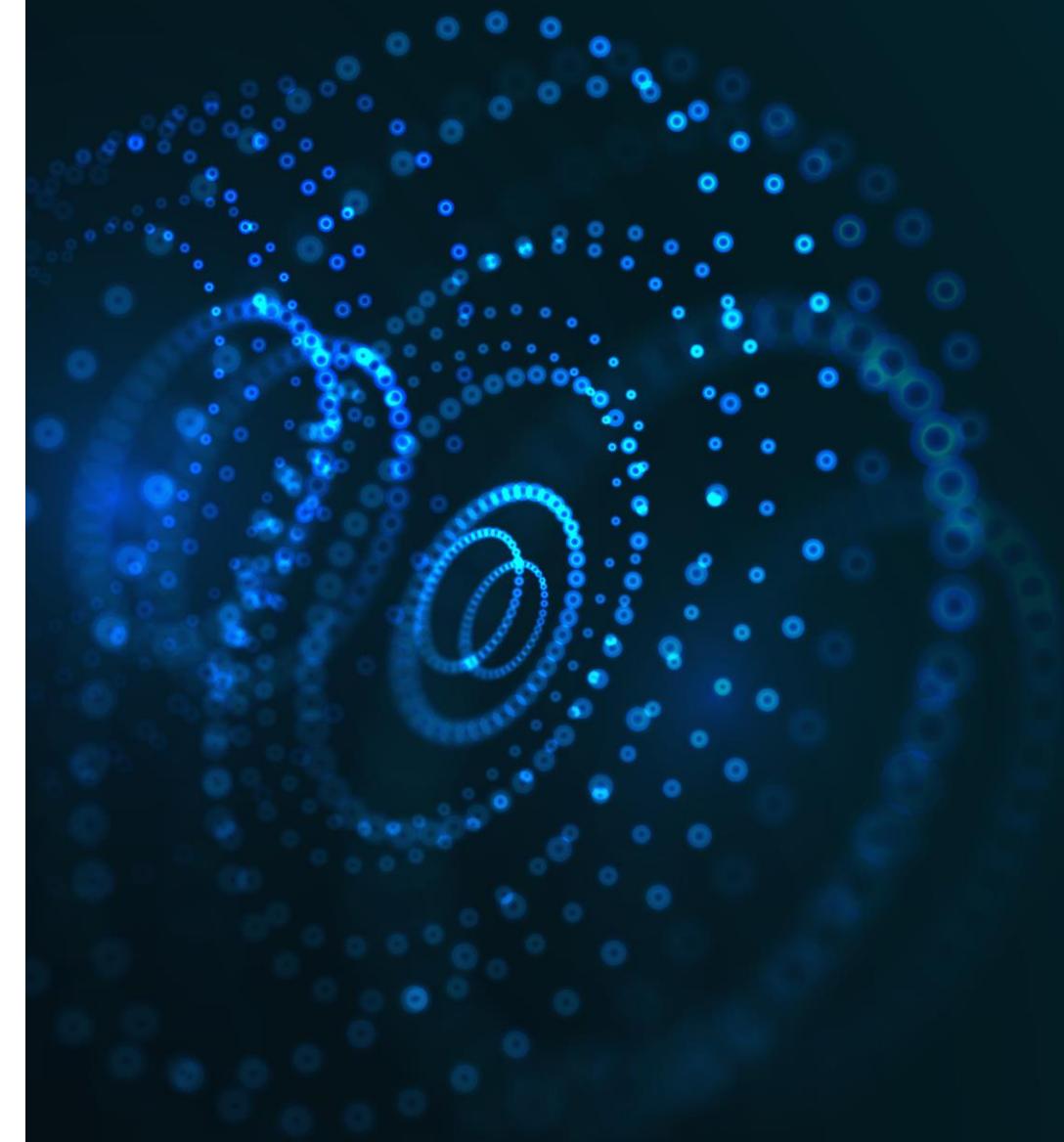
Modernize

Architected for the cloud, new code

# Azure Services Recommendations for .NET Apps

	Azure Services						
	Azure Virtual Machines (VMs)	Azure App Service	Azure Kubernetes Service (AKS)	Azure Functions	Azure Batch	Azure Container Instances (ACI)	
Application Architecture	<b>Web apps</b> (Monolithic architecture)	✓	✓ Recommended	✓			
	<b>N-Tier apps</b> (Coarse-grain services)	✓	✓  Recommended	✓  Recommended			
	<b>Cloud-Native</b> (Microservices architecture)			✓ Recommended	✓ Azure event-driven Recommended		
	<b>Batch / Jobs</b> (Background tasks)	✓	✓	✓	✓ Application's background tasks Recommended	✓ Large Batch scale Recommended	✓
✓ Recommended		✓ Possible					

# Migration & modernization case studies



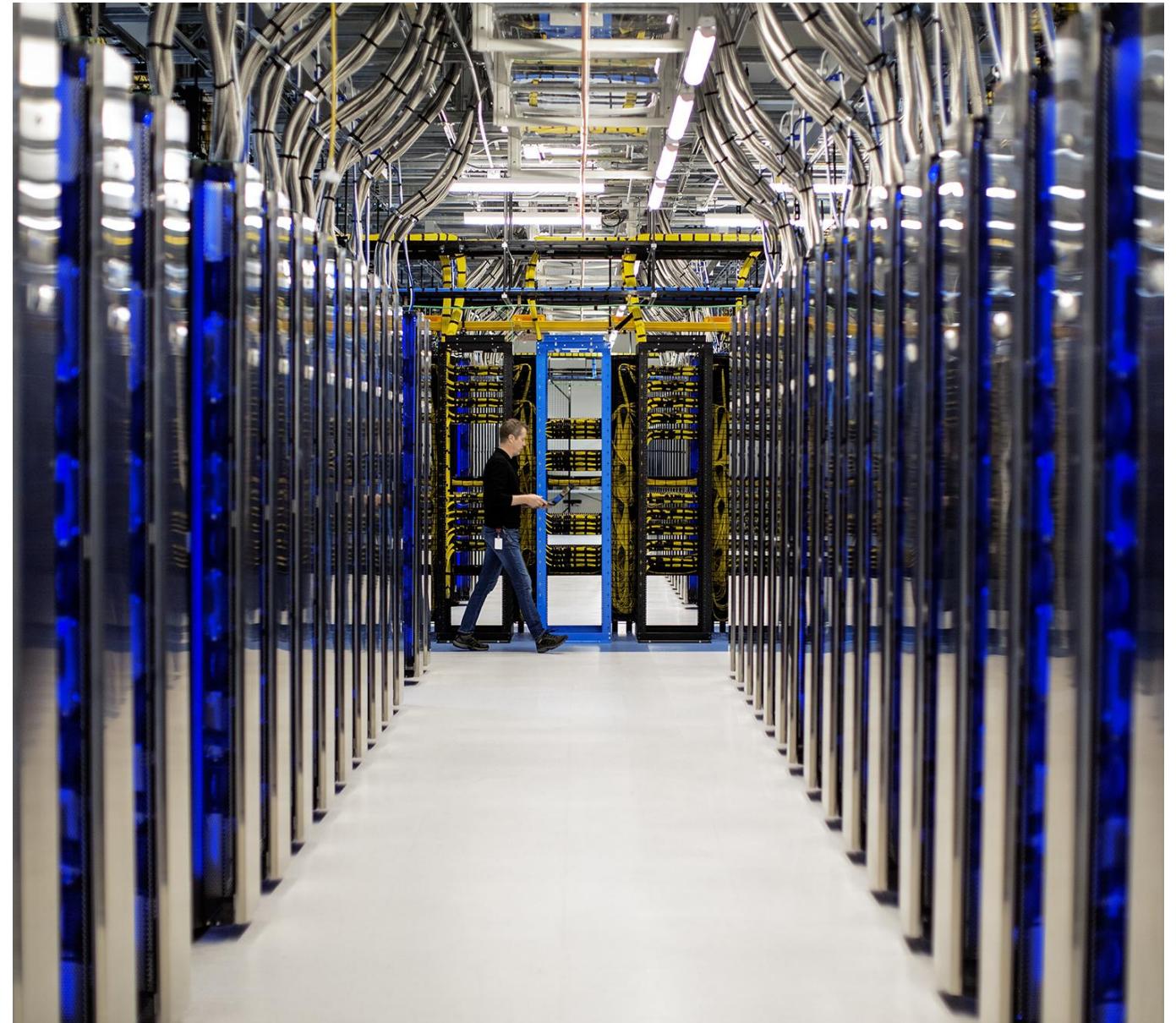
# Customer themes

## Data Center modernization

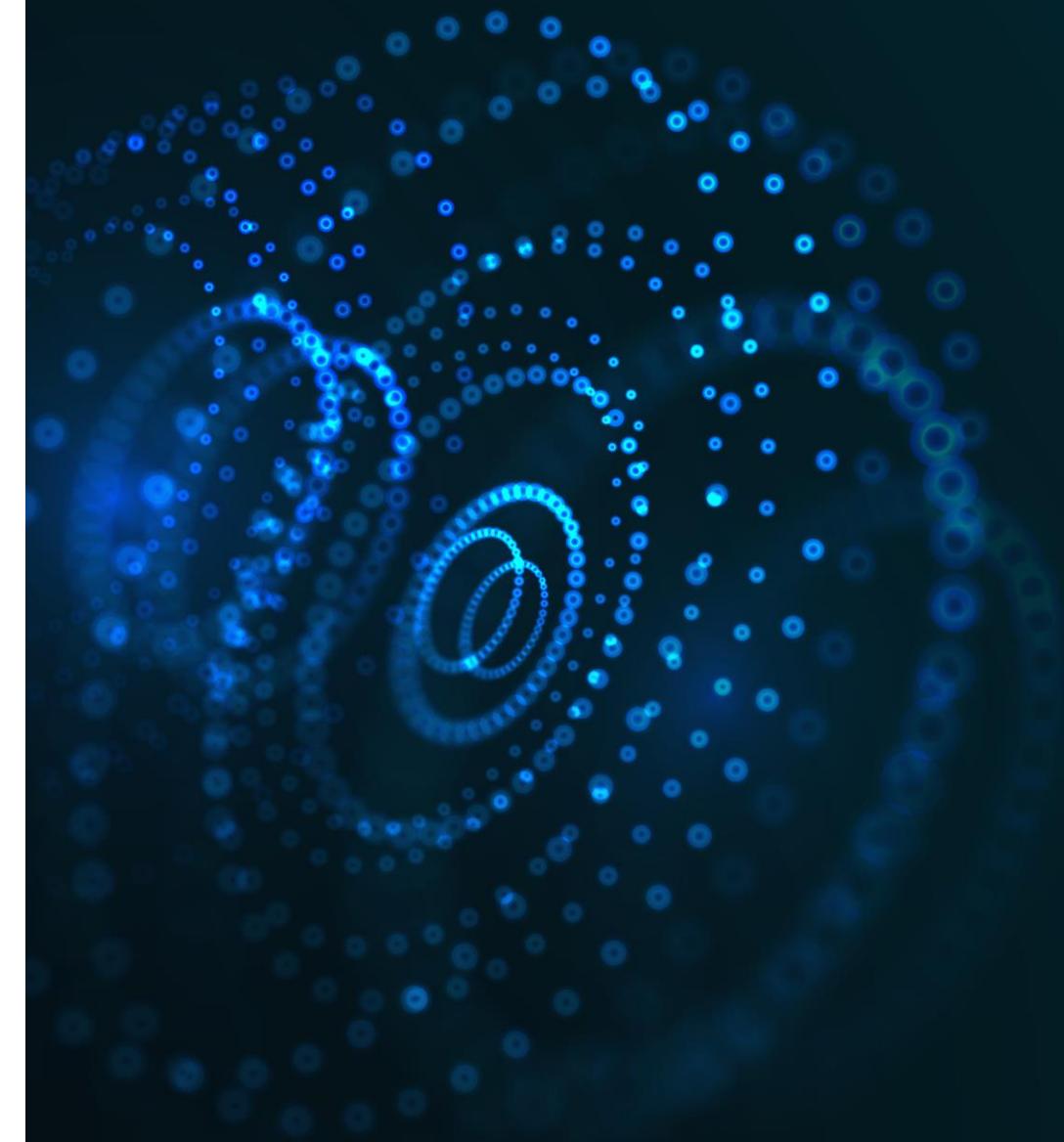
- Cloud Lift and Shift

## Application transformation

- Expand application scenarios
- Cross platform
- Containers + Microservices
- Cloud Native



# Application migration



# ASP.NET to App Service

## Problem

- Customer wants DevOps agility & App scalability
- App has to stay on .NET Framework (i.e. WebForms, WCF)
- Monolithic or n-tier (course grained services)

## Solution

- Migration to App Service with minimal code changes

# .NET Framework containerization

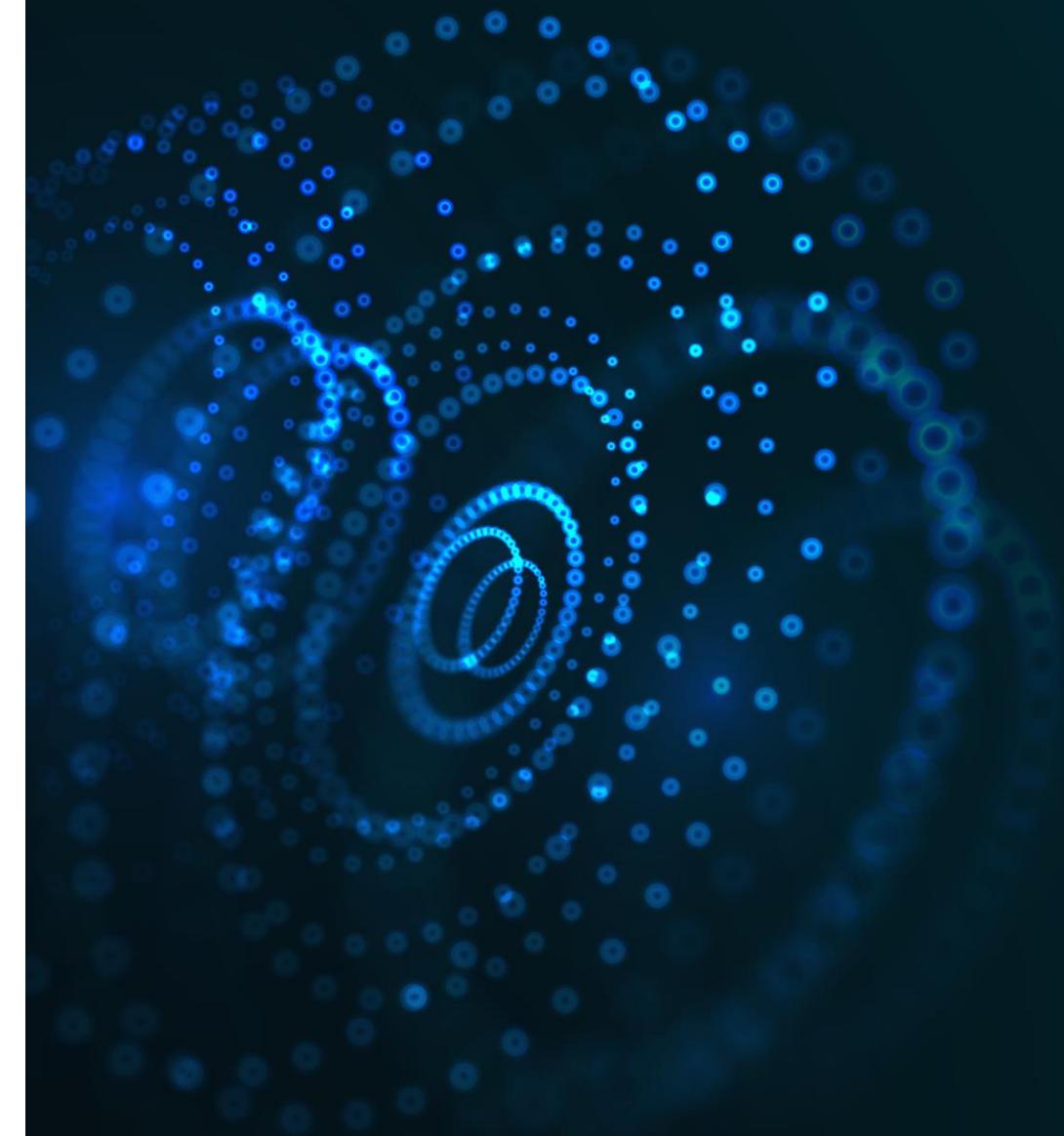
## Problem

- App has dependencies on Windows/.NET Framework (i.e. GAC, MSIs)
- Monolithic or n-tier (course grained services)

## Solution

- Windows Containers
  - App Service w/ Windows Containers
  - AKS

# Application modernization



# .NET Core Microservices on AKS

"Internal and external developers can independently release microservices at any point in time, which makes development faster and enables a continuous delivery approach completely based on Azure. We have set an astonishing speed for product development."

Thomas Friese: Vice President, Digital Ecosystem Platform  
Siemens Healthineers



Siemens Healthineers moves more computing to the cloud with .NET Core and Azure Kubernetes Service (AKS)

Customer Story:  
<https://aka.ms/SiemensCustomerStory>





# App performance monitoring provider increases throughput by 2,000 percent with .NET Core

The Raygun platform offers a cloud-based set of tools that help developers build better-performing, more reliable software. Its Real-User Monitoring and Crash Reporting applications provide developers with deep insights into problems that end users are experiencing, so they can diagnose and fix issues faster. By moving its API nodes, which run on Linux, from Node.js to .NET Core, Raygun increased its request throughput from 1,000 to 20,000 requests per second per node, increasing scalability and reducing the number of nodes required.



Products & Services	Org Size	Industry	Country	Business Need
Microsoft .NET Core Microsoft Visual Studio	40 employees	Telecommunications	New Zealand	Software error and crash reporting



"To be one of the best e-commerce destinations in the US, we will have to handle millions of customers... That requires a top-class e-commerce system built on a flexible, open cloud platform. That is exactly what we got with Azure.

Being able to leverage so many off-the-shelf services and tools from Azure enabled us to go from zero to a full-fledged e-commerce marketplace in just about 12 months."

Mike Hanrahan

CTO  
Jet.com





Tencent 腾讯



腾讯支付基础平台与金融应用线

# Tencent updates web and mobile payment infrastructure for better scalability and agility

Tencent  
腾讯

Products & Services	Org Size	Industry	Country	Business Need
.NET Core ASP.NET Core	34,000 employees	Professional Services	China	IT Development



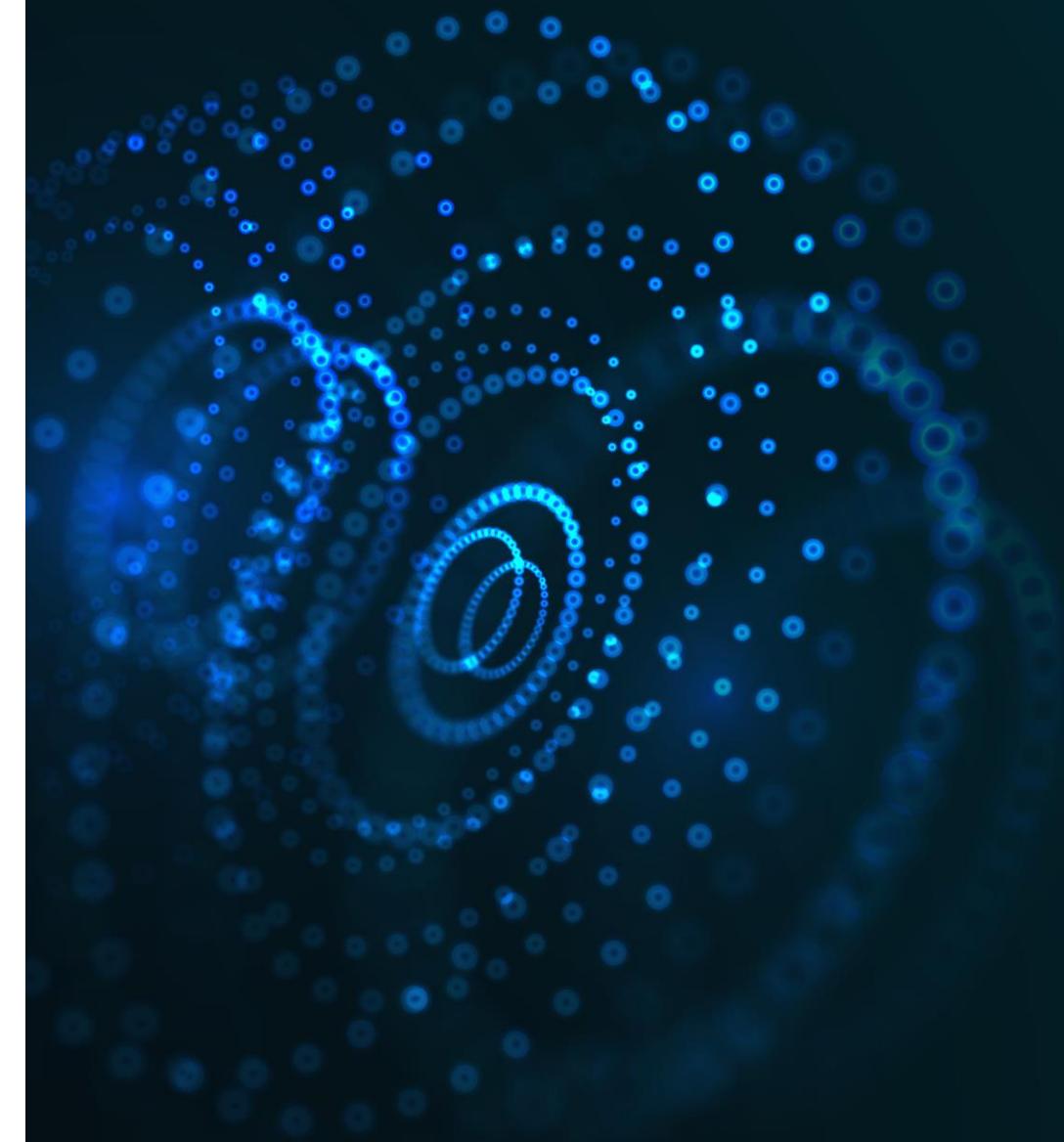
# GoDaddy ports communications services to Linux in six weeks using .NET Core

From building websites to hosting productivity tools, GoDaddy provides technology that helps small businesses succeed. As part of its goal to remain the go-to cloud platform for wide-ranging web services, GoDaddy acquired FreedomVoice in 2016 to enlarge the company's portfolio of communication offerings. GoDaddy developers quickly and easily ported the acquired firm's Microsoft .NET-based technology stack to [.NET Core](#) — a step critical to creating a new end-to-end business telephony solution running on Linux.



Products & Services	Org Size	Industry	Country	Business Need
.NET Core Microsoft Visual Studio .NET Framework	6,000 employees	Professional Services	United States	Platform services Data migration

# Migration & modernization resources



# Get help moving .NET apps to Azure

## Assess & Migrate

<https://aka.ms/MigrateToTheCloud>

## The Azure Hybrid Benefit

Migrate your Windows Server and SQL Server-based web apps to Azure with your on-premises licenses

<https://azure.microsoft.com/en-us/pricing/hybrid-benefit/>

## Microsoft Engineering Support

## Migration Tools for Visual Studio Enterprise Subscribers

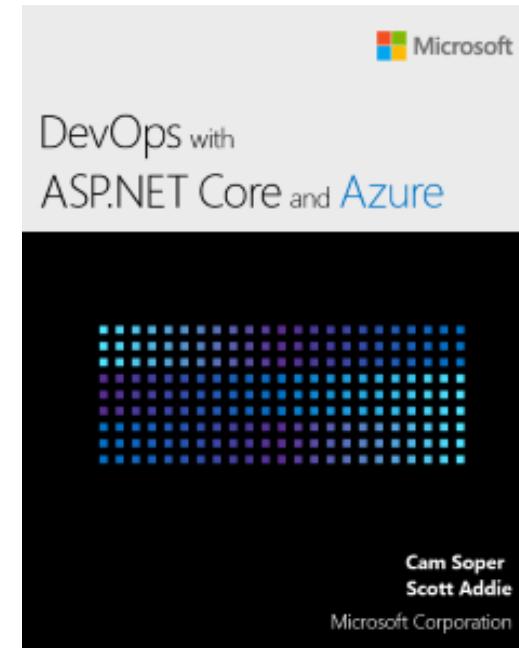
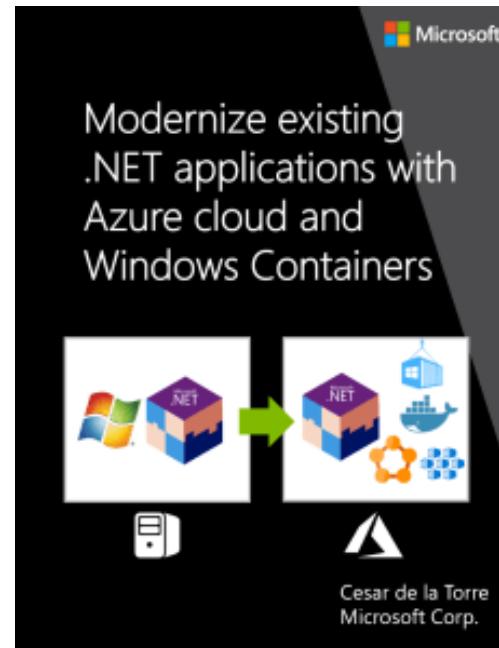
- **CAST Highlight** - one full 30-day subscription up to 5 apps
- **CloudPilot** - 2 licenses for 90 days, unlimited scans of 1 app per license

<https://visualstudio.microsoft.com/subscriptions/>

# Get help migrating .NET apps to Azure

Download free e-books and assessment resources

[aka.ms/migratetothecloud](https://aka.ms/migratetothecloud)



Customers can apply for **free** assistance from the .NET team

# .NET Architecture Guides



## Microservices

Build resilient, scalable, and independently deployable microservices using .NET and Docker.

[Microservices guidance](#)



## DevOps

DevOps and application lifecycle best practices for your .NET applications.

[DevOps guidance](#)



## Modernizing web & server

Options for modernizing your existing web and server applications for the cloud.

[Modernization guidance](#)



## Azure cloud apps

Build production-ready cloud applications for scalability, security, resiliency, and more using Azure.

[Azure guidance](#)



## ASP.NET apps

Quickly build, test, and deploy data-driven web applications using the ASP.NET web framework.

[ASP.NET guidance](#)



## Mobile apps

Build apps for iOS, Android, and Windows using .NET. Leverage native APIs on every platform while maximizing code-sharing across all of them.

[Mobile guidance](#)



## UWP desktop apps

Build modern desktop experiences that empower your customers to do more with the Universal Windows Platform (UWP).

[UWP guidance](#)

[www.dot.net/architecture](http://www.dot.net/architecture)

# Q&A



# Resources

Docs, samples, e-books and request for free migration assistance from the .NET team

<http://aka.ms/MigrateToTheCloud>

Free .NET architecture guides and samples

[www.dot.net/architecture](http://www.dot.net/architecture)

Azure migration center

<https://azure.microsoft.com/en-us/migration/>

Azure reference architecture guides

<https://docs.microsoft.com/en-us/azure/architecture/guide/>

This deck can be found at

<https://github.com/dotnet-presentations/home>



Thank you for joining us.