

# Cloud-Native Architecture For .NET Developers

# Hi, I'm J.

Jonathan "J." Tower

Partner & Principal Consultant

Trailhead Technology Partners

🏆 Six-time Microsoft MVP

👤 Business owner

📅 Beer City Code organizer



✉️ [jtower@trailheadtechnology.com](mailto:jtower@trailheadtechnology.com)

🌐 [trailheadtechnology.com/blog](https://trailheadtechnology.com/blog)

🐦 [jtowermi](https://twitter.com/jtowermi)

[github.com/jonathantower/cloud-native-dotnet](https://github.com/jonathantower/cloud-native-dotnet)

# Overview

Assumptions

Background

7 Patterns

11 Technologies



# If You Give \$200, So Will I!

## **[bit.ly/chicago-cloud](https://bit.ly/chicago-cloud)**

*"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 56,759 projects in 29 countries, benefiting over 11.4 million people."*

*"4/4 Stars"  
- CharityNavigator.org*



charity: water

# Assumptions

Experience or preference for Microsoft stack—C#, .NET, Azure, etc.

**You want a high-level and conceptual overview**

# Cloud-Native Terms

To Make Sure We're All On The Same Page

# What is “Cloud Native”?

Cloud native technologies empower organizations to build and run **scalable applications** in **modern, dynamic environments** such as public, private, and hybrid **clouds**. **Containers, service meshes, microservices, immutable infrastructure, and declarative APIs** exemplify this approach.

These techniques enable **loosely coupled** systems that are **resilient, manageable, and observable**. Combined with **robust automation**, they allow engineers to make high-impact **changes frequently** and predictably with **minimal toil**.

Cloud Native Computing Foundation (CNCF)

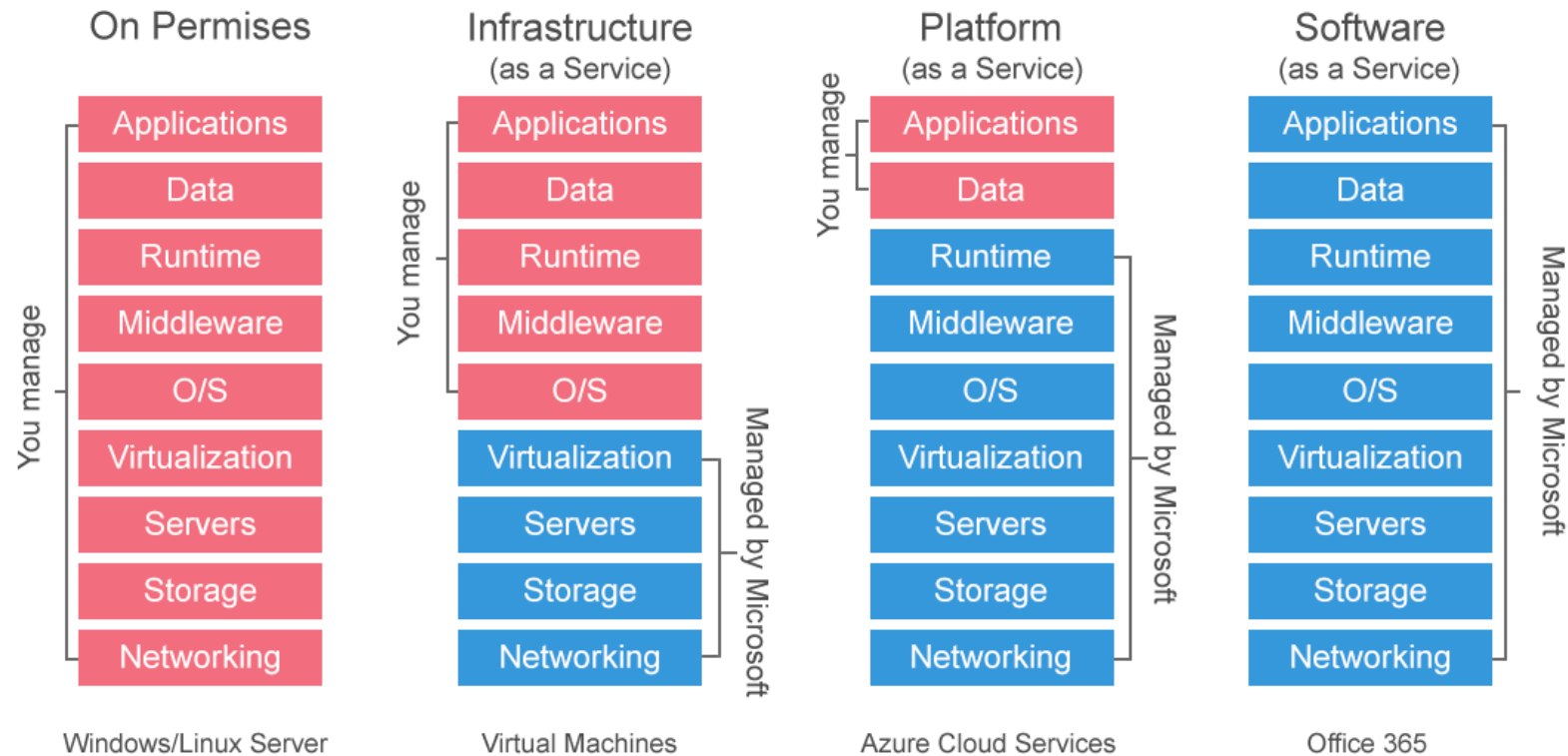
<https://github.com/cncf/foundation/blob/master/charter.md>

# What is "Cloud Native"?

**scalable applications**  
**loosely coupled** **service meshes**  
**containers** **immutable infrastructure**  
**declarative APIs** **changes frequently** **cloud**  
**minimal toil** **robust automation**  
**microservices** **modern, dynamic environments**

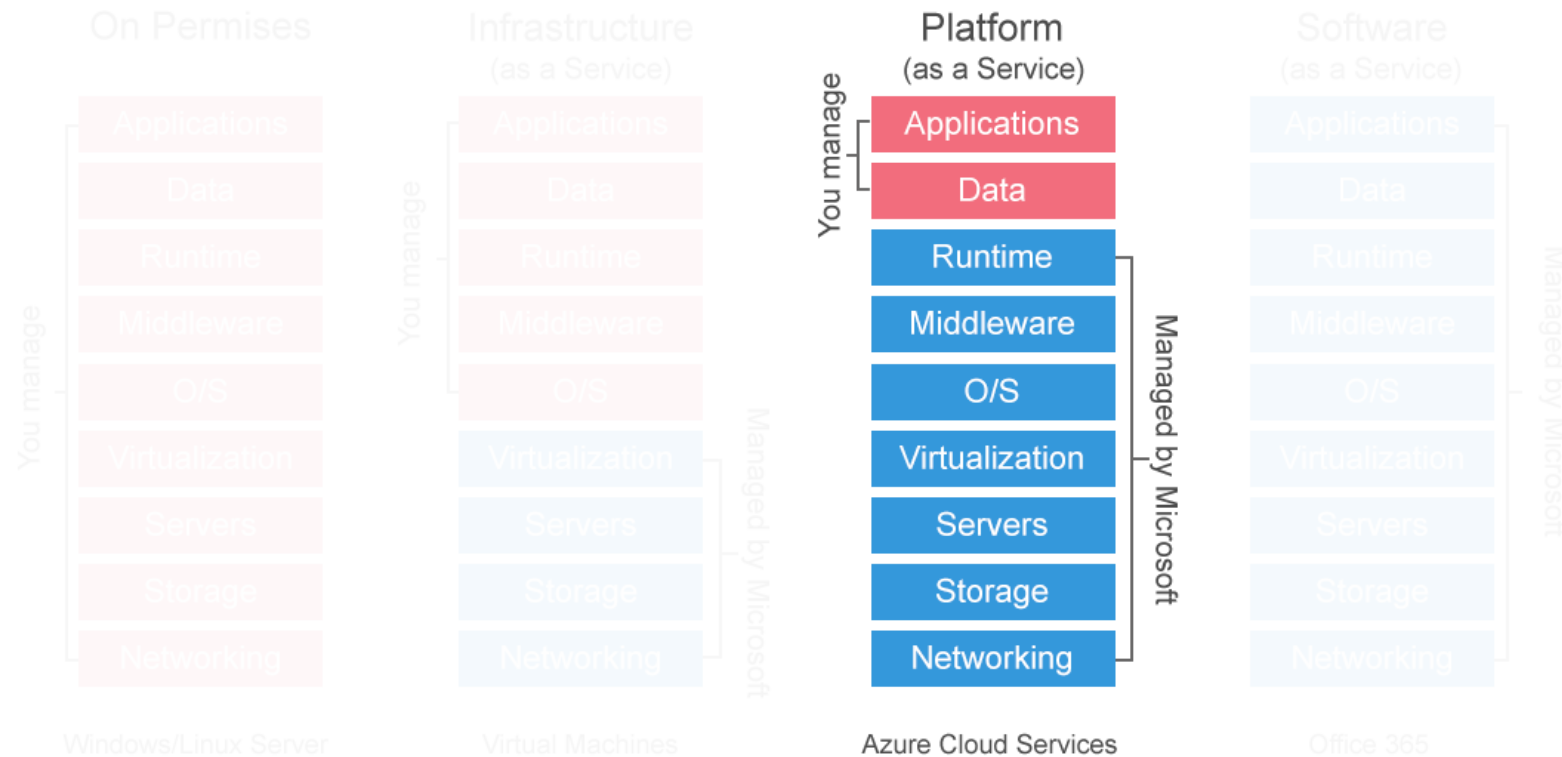


# IaaS, PaaS, SaaS



# IaaS, PaaS, SaaS

**Cloud-Native**



# Alt Definition: 12-Factor Applications

## I. **Codebase**

One codebase tracked in revision control, many deploys

## II. **Dependencies**

Explicitly declare and isolate dependencies

## III. **Config**

Store config in the environment

## IV. **Backing services**

Treat backing services as attached resources

## V. **Build, release, run**

Strictly separate build and run stages

## VI. **Processes**

Execute the app as one or more stateless processes

## VII. **Port binding**

Export services via port binding

## VIII. **Concurrency**

Scale out via the process model

## IX. **Disposability**

Maximize robustness with fast startup and graceful shutdown

## X. **Dev/prod parity**

Keep development, staging, and production as similar as possible

## XI. **Logs**

Treat logs as event streams

## XII. **Admin processes**

Run admin/management tasks as one-off processes



Source: <https://12factor.net/>

# Beyond 12-Factor

## XIII. **API First**

Make everything a service. Assume your code will be consumed by a front-end client, gateway, or another service.

## XIV. **Telemetry**

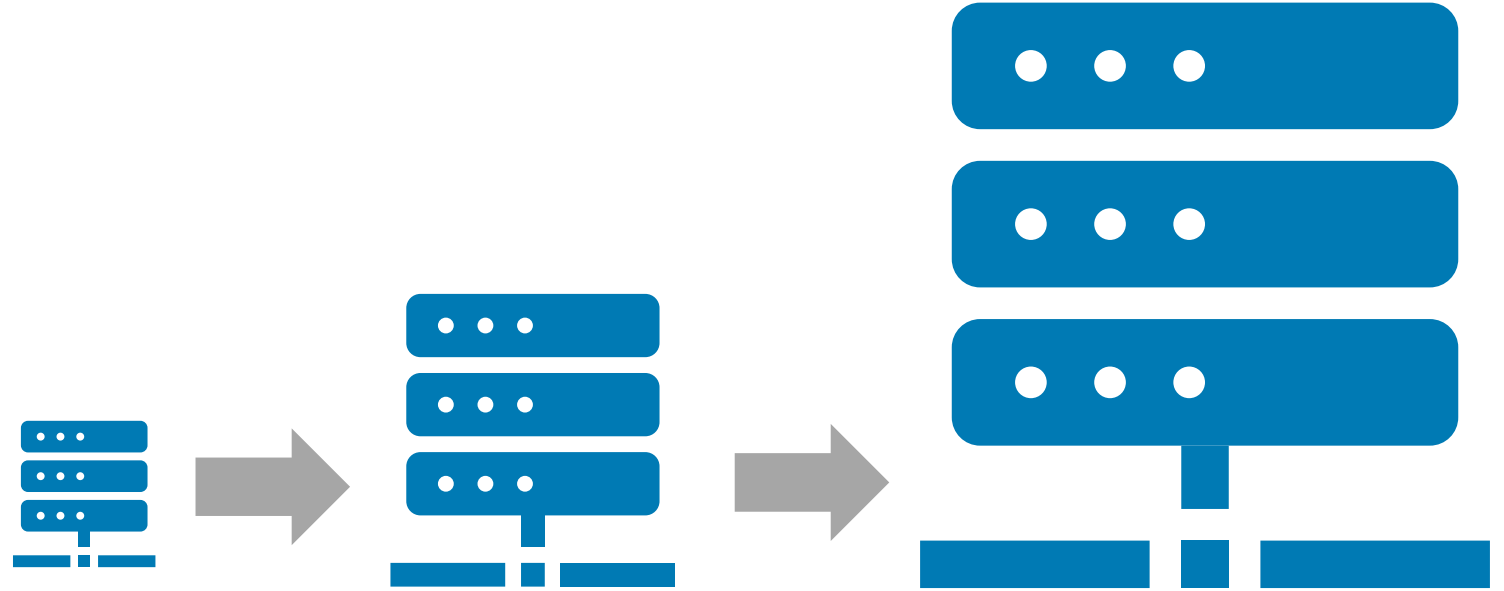
On a workstation, you have deep visibility into your application and its behavior. In the cloud, you don't. Make sure your design includes the collection of monitoring, domain-specific, and health/system data.

## XV. **Authentication/Authorization**

Implement identity from the start. Consider RBAC (role-based access control) features available in public clouds.

# Scaling

Scale Up (Vertical)



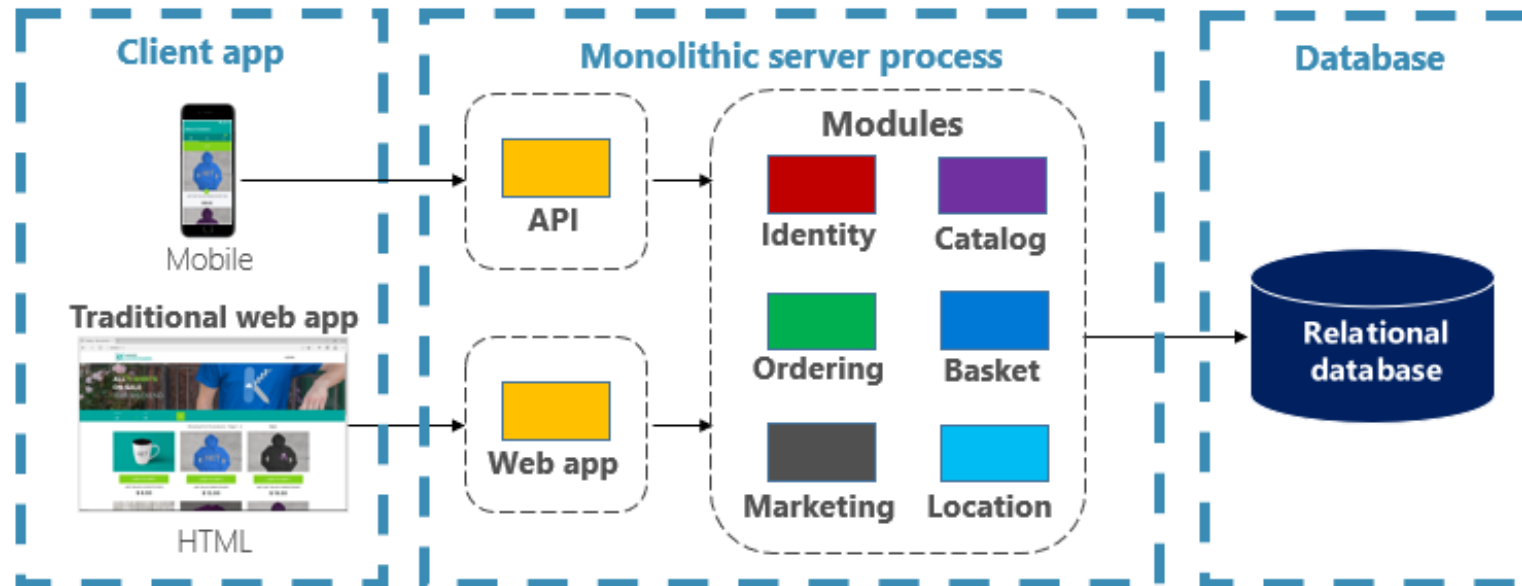
Scale Out (Horizontal)



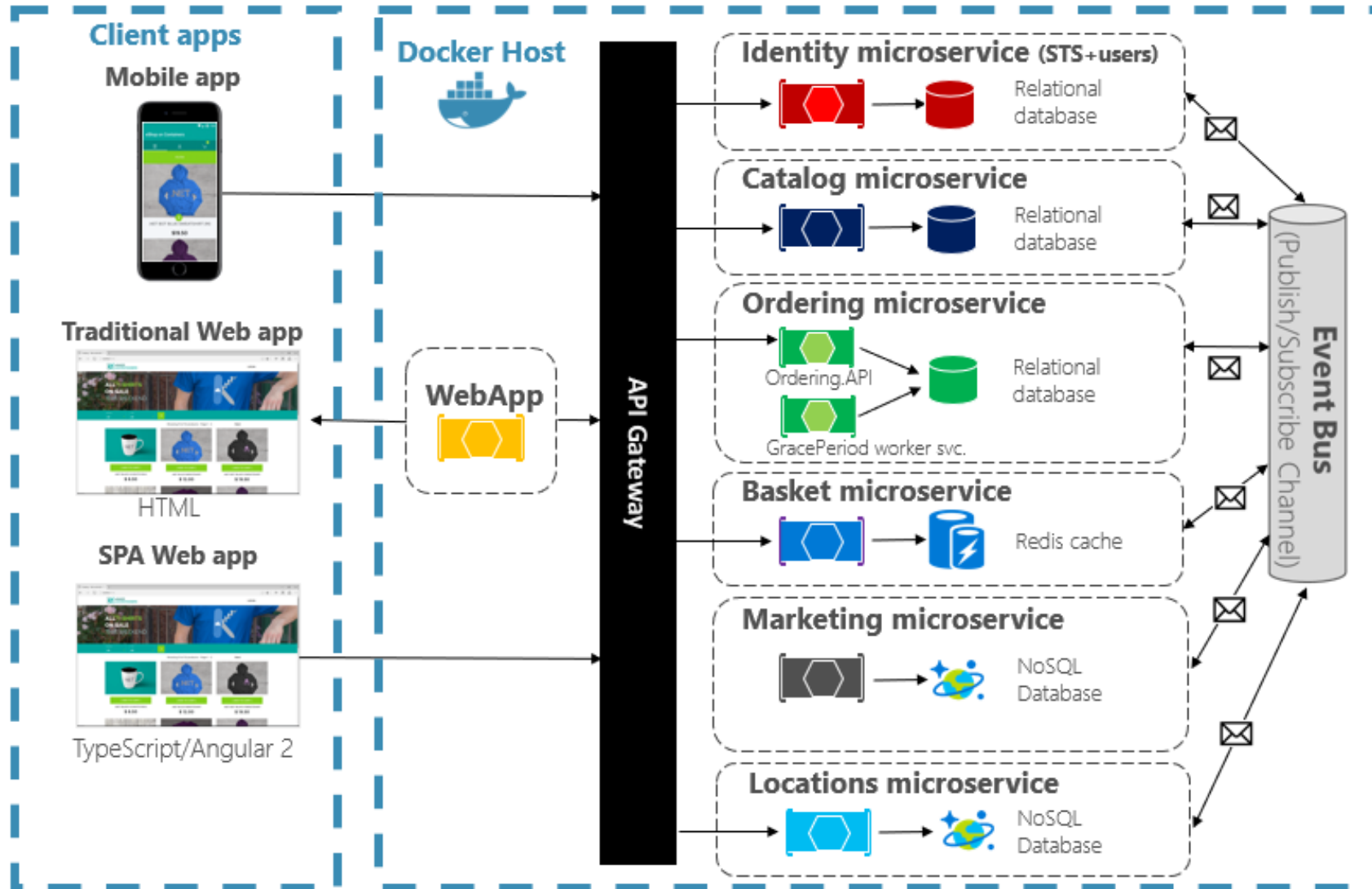
# Cloud-Native Patterns

And How They Help

# 1 | Microservices



# Microservices





# Microservices Advantages



Isolation



Scalability



Productivity



Flexibility

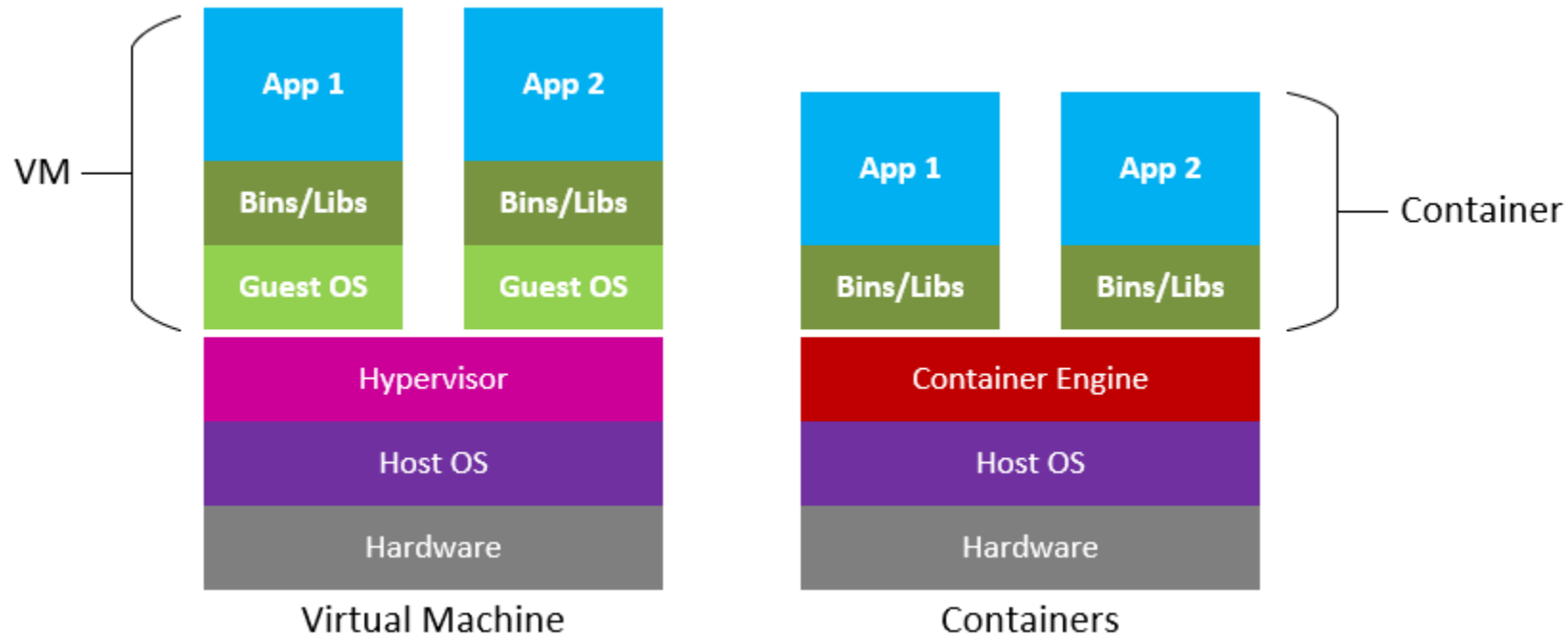


Faster  
development



Evolutionary

## 2 | Containerization



# Containerization



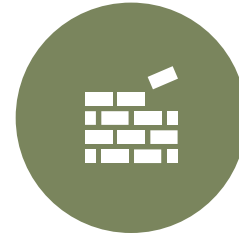
ANY CLOUD



FOCUS ON  
APPLICATIONS, IGNORE  
SERVER NUANCES



CAN BE FOUNDATION  
FOR SERVERLESS



ISOLATION

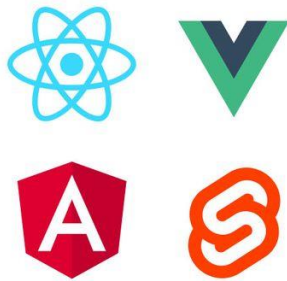


ORCHESTRATION

# 3 | JAMstack

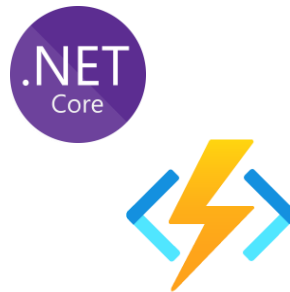
## J

Javascript



## A

APIs

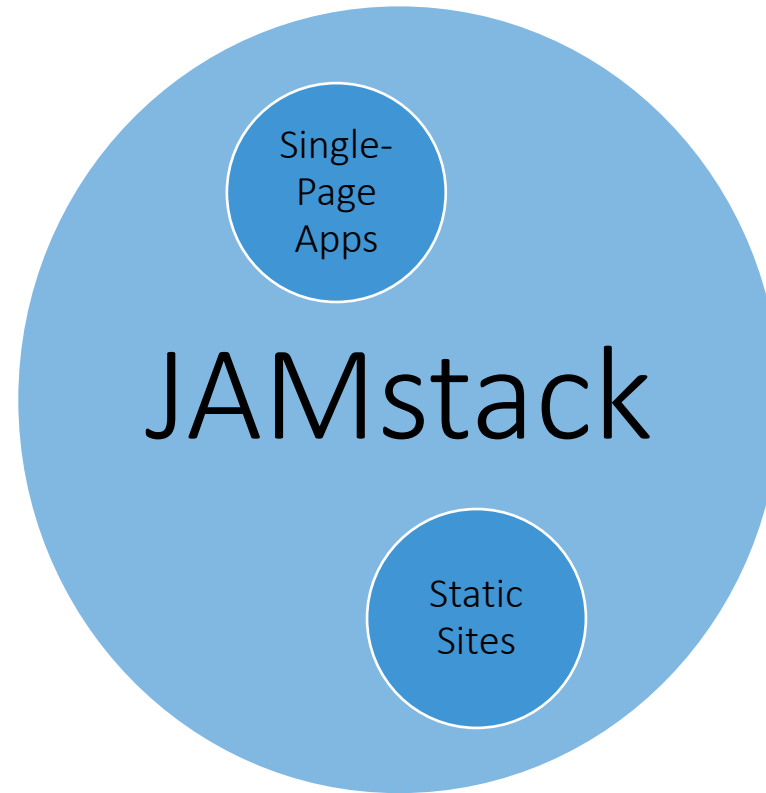


## M

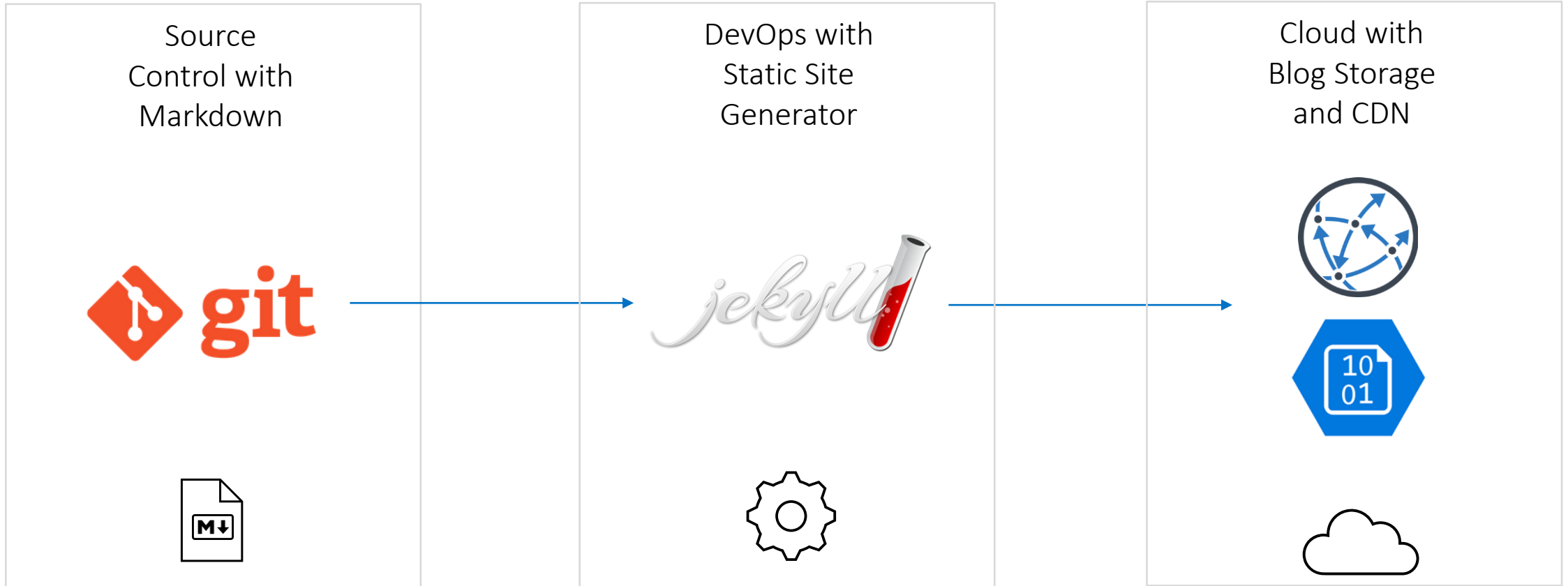
Markup



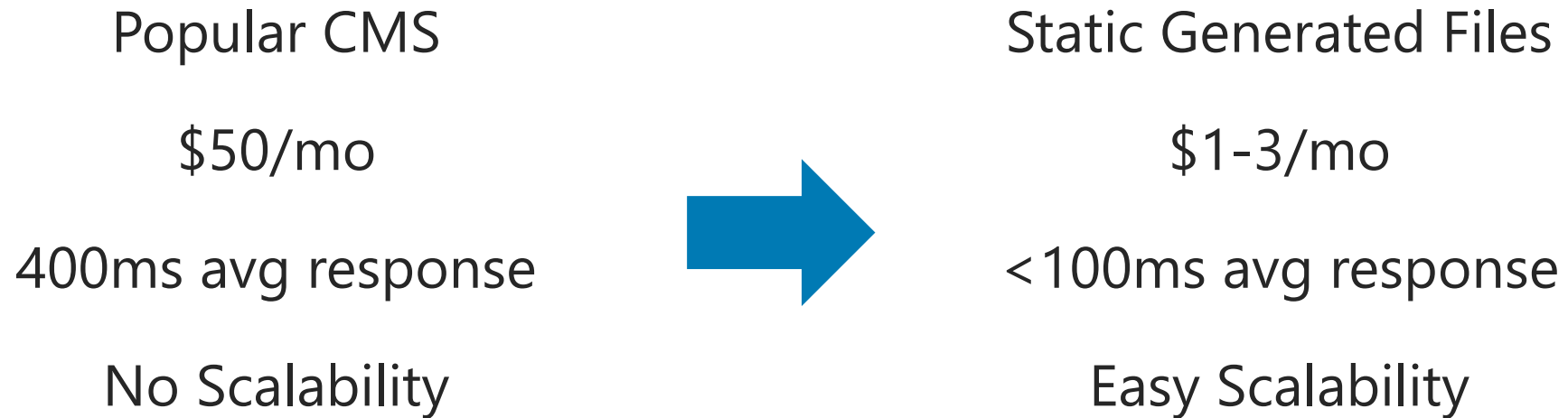
# JAMstack



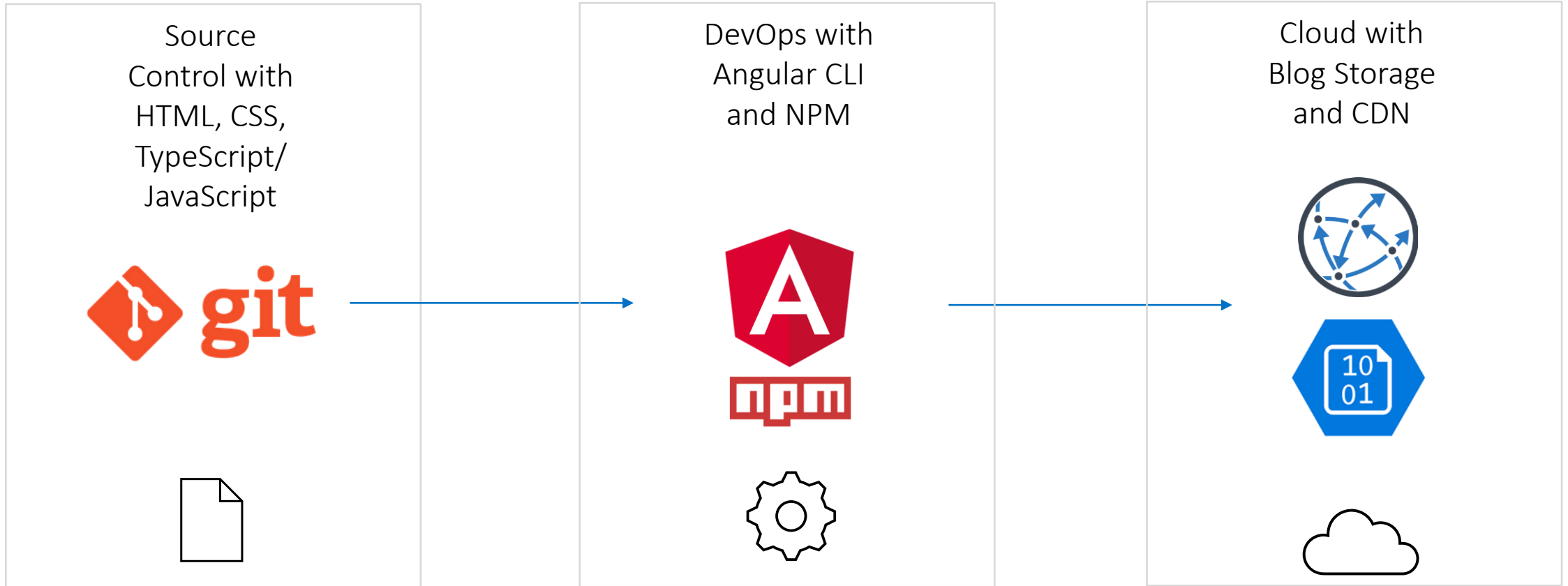
# JAMstack Scenario: Static Site



# Static Site Case Study



# JAMstack Scenario: SPA





# Popular Static Site Generators



# .NET/Razor Static Site Generators

<div>Statiq</div> <div><div>★ 1278 +8</div><div>🕒 183 +1</div><div>🔗 214 +1</div><div>🐦 N/A</div></div> <div><p>Statiq Web is a flexible static site generator written in .NET</p><p><b>Languages:</b> .Net</p><p><b>Templates:</b> Razor, Markdown</p><p><b>License:</b> Prosperity Public License 3.0.0</p></div>	<div>Wyam</div> <div><div>★ 74 +2</div><div>🕒 13 --</div><div>🔗 17 --</div><div>🐦 N/A</div></div> <div><p>A simple to use, highly modular, and extremely configurable static content generator.</p><p><b>Languages:</b> .Net</p><p><b>Templates:</b> Razor, Markdown</p><p><b>License:</b> MIT</p></div>	<div>Misakai Baker</div> <div><div>★ 41 --</div><div>🕒 1 --</div><div>🔗 5 --</div><div>🐦 N/A</div></div> <div><p>Slice and Optimize with Razor and Markdown</p><p><b>Languages:</b> .Net</p><p><b>Templates:</b> Razor</p><p><b>License:</b> Apache-2.0</p></div>
<div>graze</div> <div><div>★ 128 --</div><div>🕒 4 --</div><div>🔗 23 --</div><div>🐦 N/A</div></div> <div><p>Static site generator using Razor.</p><p><b>Languages:</b> .Net</p><p><b>Templates:</b> Razor</p><p><b>License:</b> MIT</p></div>	<div>IronBeard</div> <div><div>★ 15 --</div><div>🕒 1 --</div><div>🔗 5 --</div><div>🐦 N/A</div></div> <div><p>Simple, zero-configuration static site generator written in .NET Core.</p><p><b>Languages:</b> .Net</p><p><b>Templates:</b> Razor, Markdown</p><p><b>License:</b> MIT</p></div>	<div>Record Collector</div> <div><div>★ 2 --</div><div>🕒 0 --</div><div>🔗 0 --</div><div>🐦 3 --</div></div> <div><p>An ASP.NET Core MVC 3.1 static site toolkit.</p><p><b>Languages:</b> C#</p><p><b>Templates:</b> Razor</p><p><b>License:</b> MIT</p></div>

# If You Give \$200, So Will I!

## **bit.ly/chicago-cloud**

*"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 22,936 projects in 24 countries, benefiting over 4.6 million people." - Wikipedia*

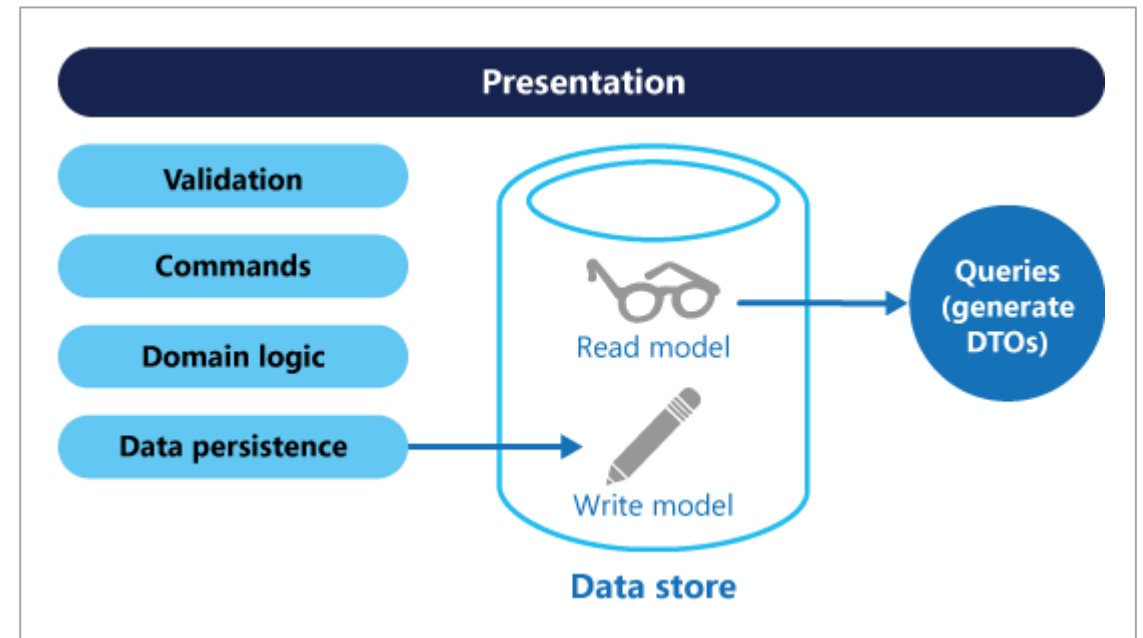
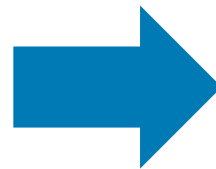
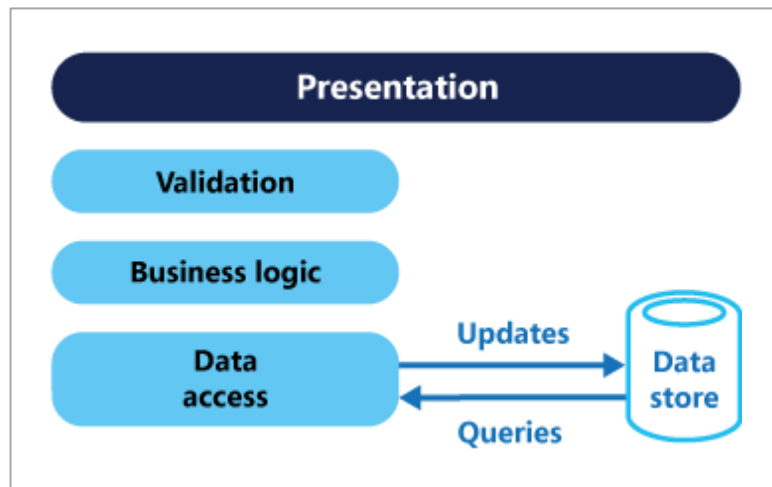
*"4/4 Stars"  
- CharityNavigator.org*



charity: water

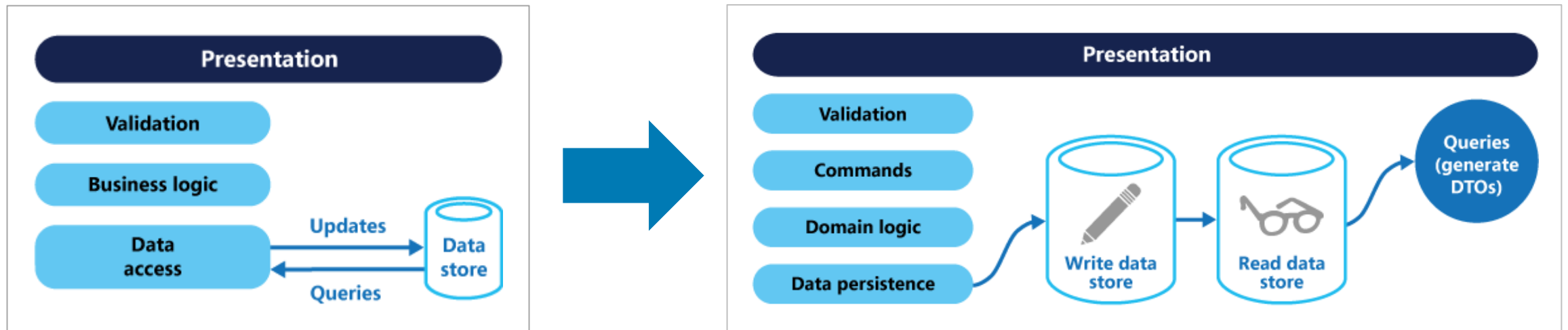
# 4 | CQRS Pattern

“Command and Query Responsibility Segregation”

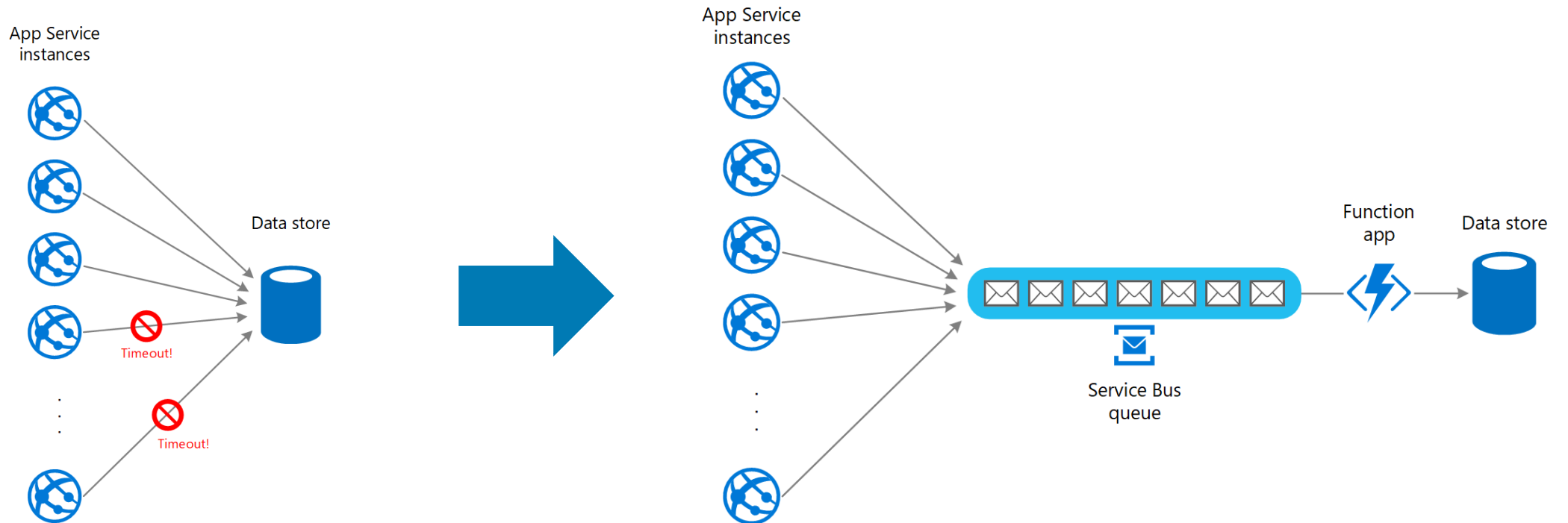


# CQRS Pattern

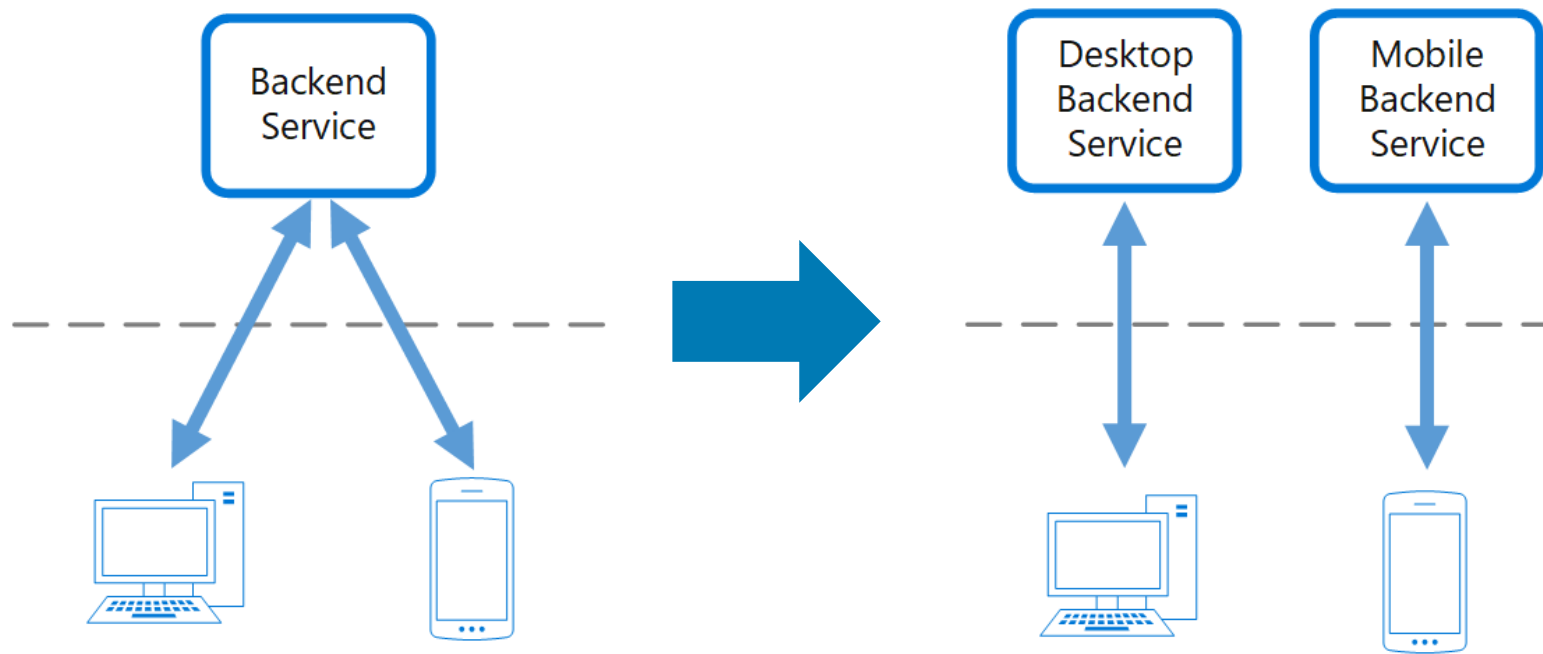
“Command and Query Responsibility Segregation”



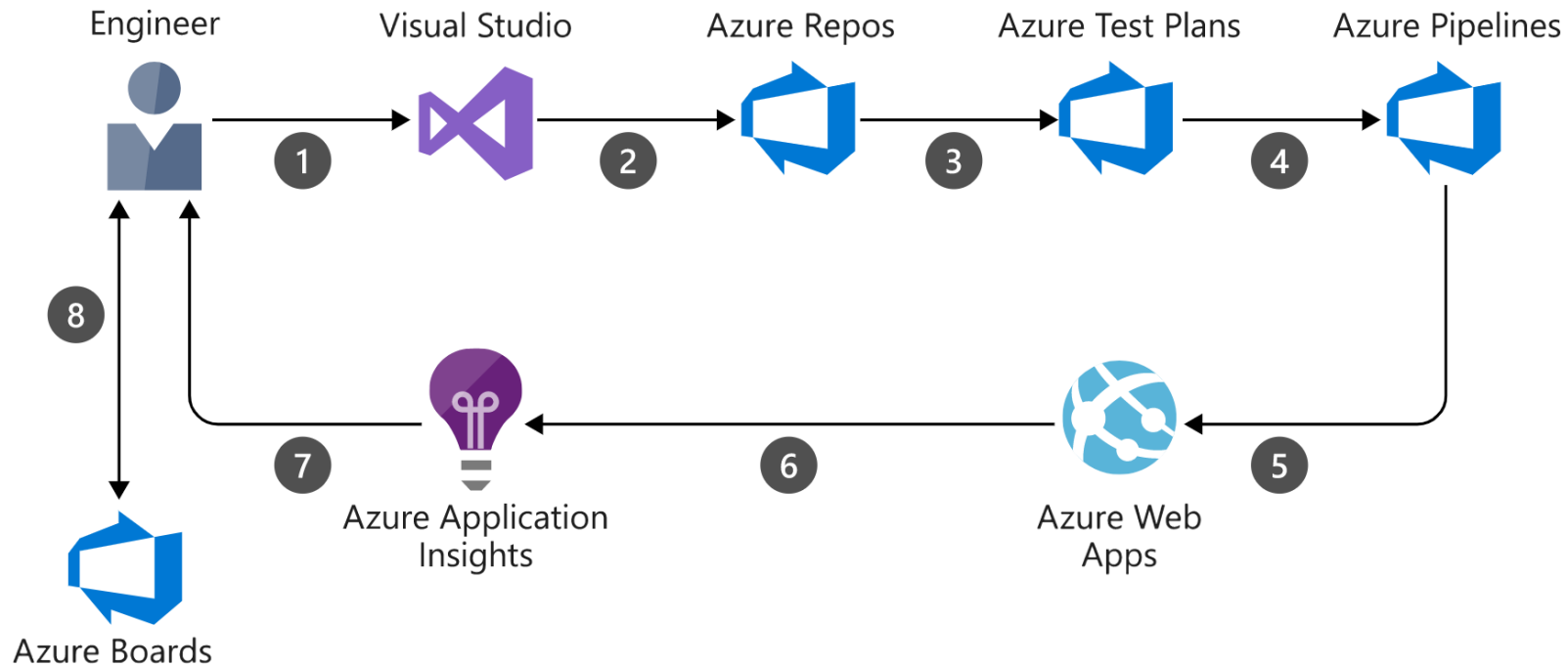
# 5 | Queue-Based Load Leveling



# 6 | Backends For Frontends



# 7 | Continuous Integration & Deployment





# Cloud-Native Technologies

And When to Use Them

# 1 | .NET Core



It's (past?)  
time to switch

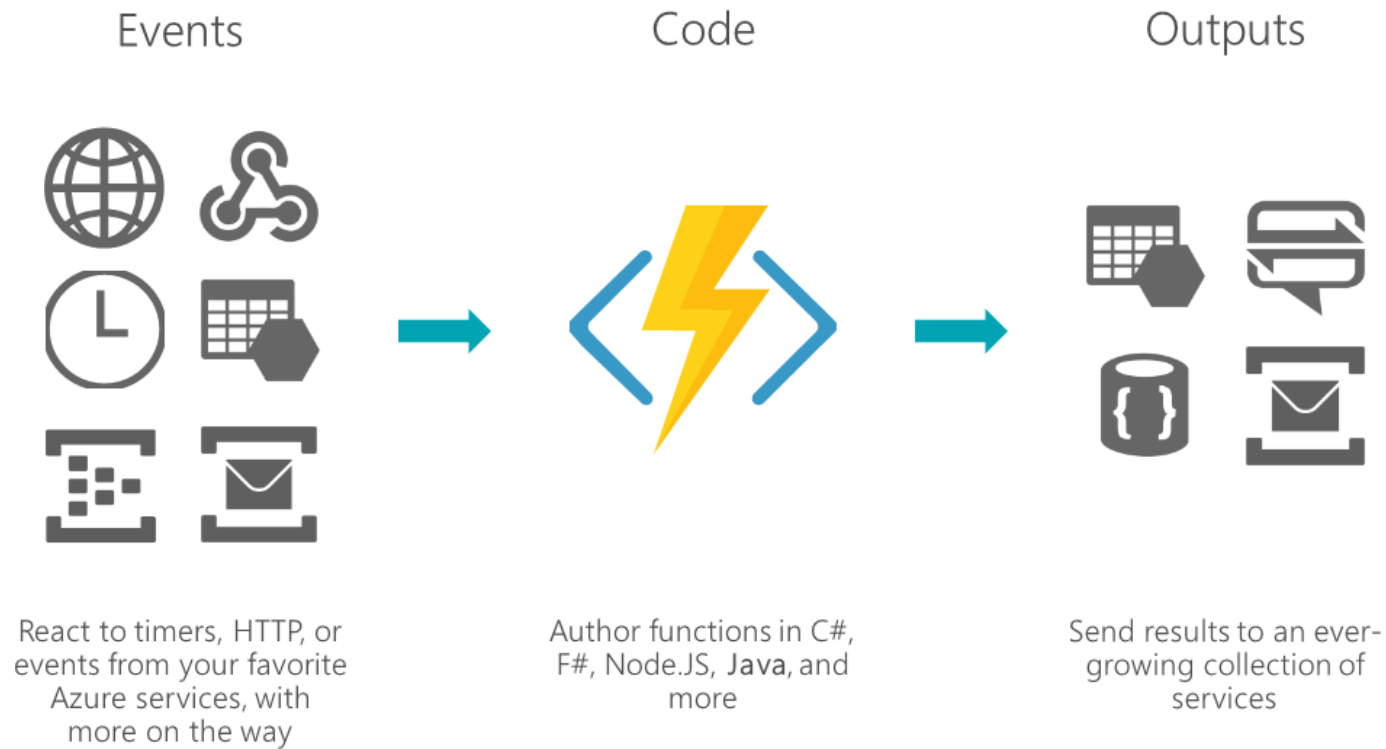


.NET Unified  
by .NET 5

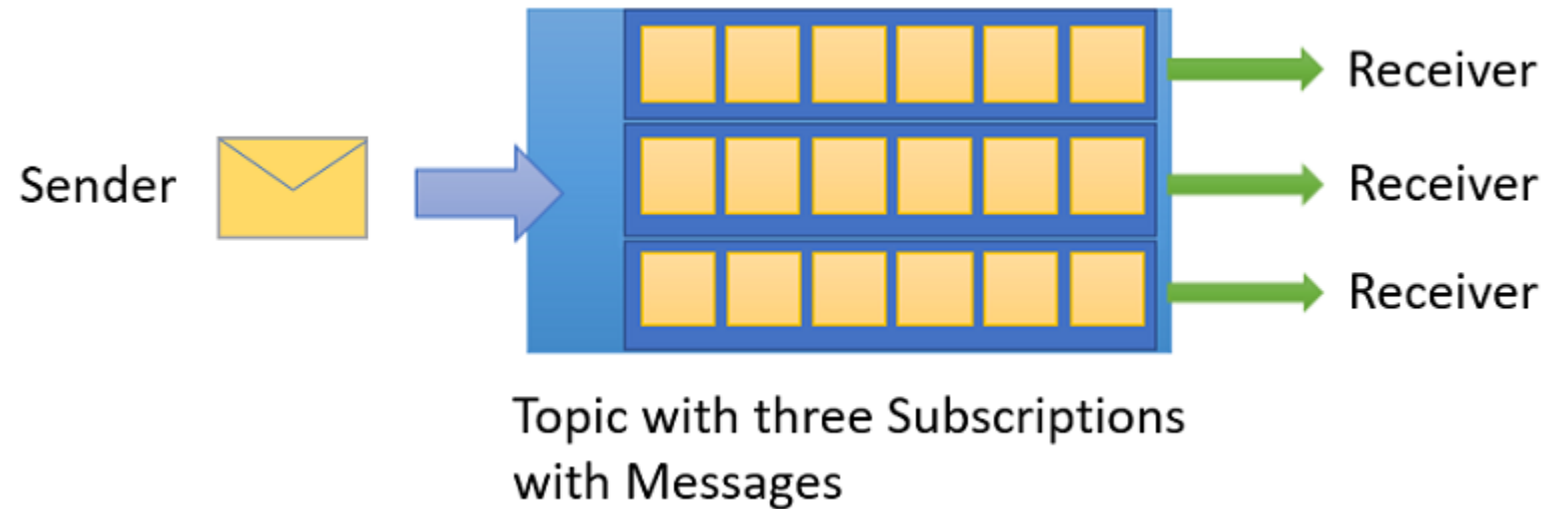


The .NET  
Standard

## 2 | "Serverless" with Azure Functions



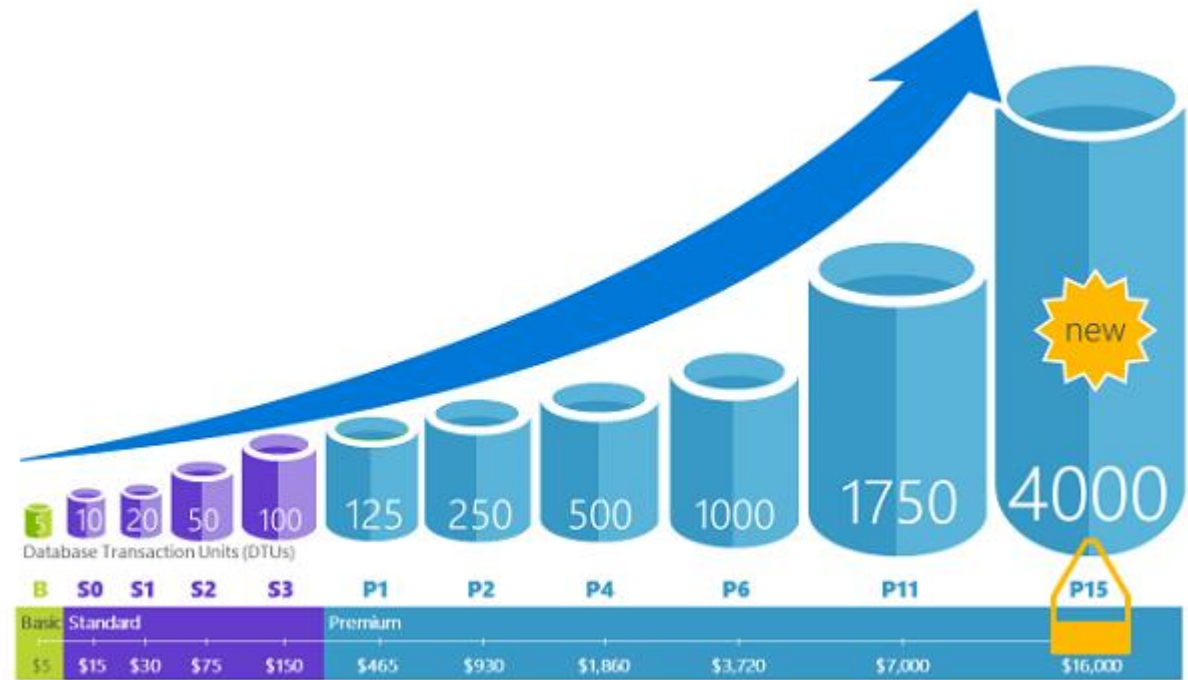
# 3 | Azure Service Bus



# 4 | Scalable RDBMS on SQL Azure

Automated scaling on-demand

NOTE: scales VERTICAL only

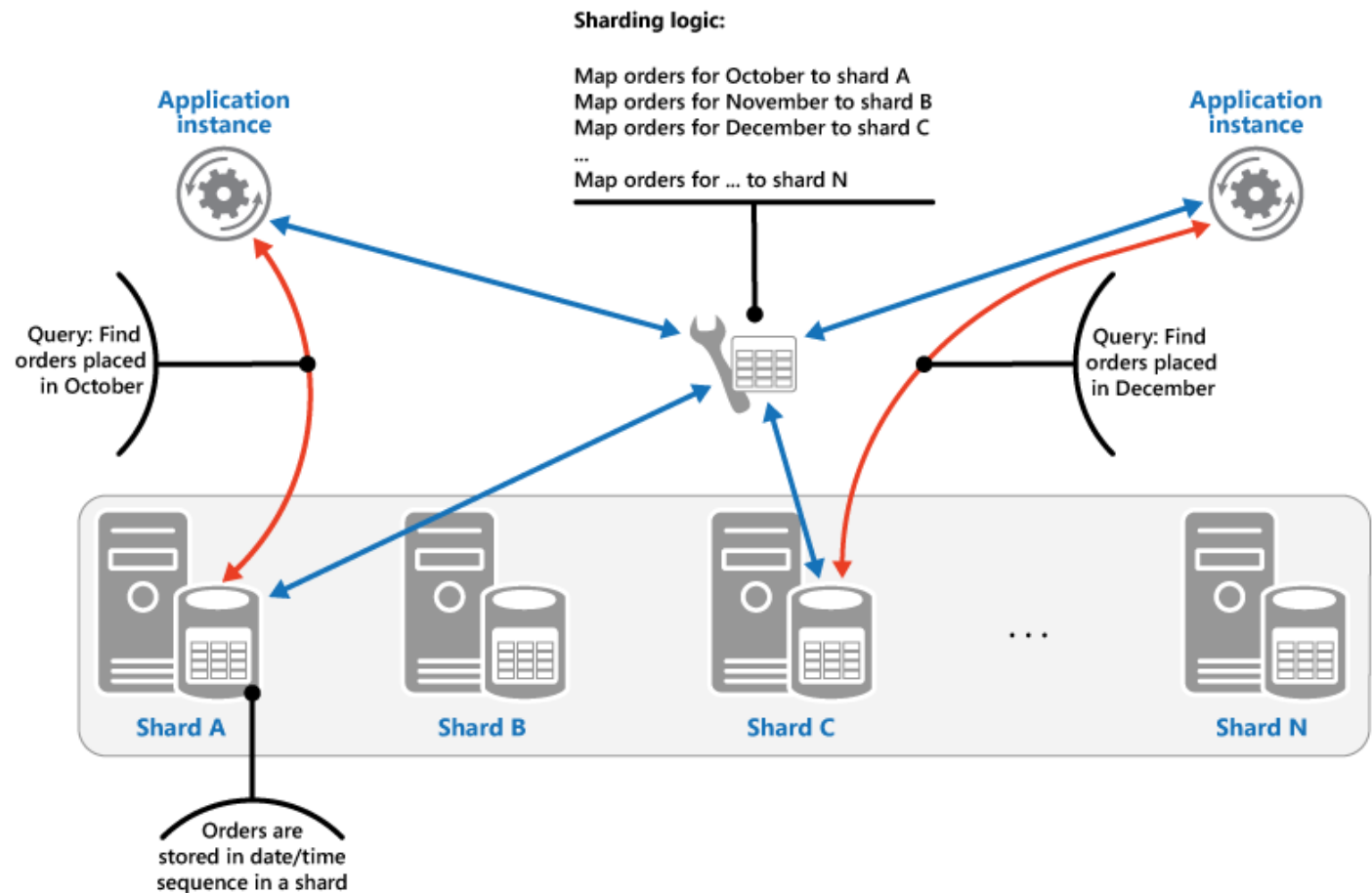


# 5 | Scalable RDBMS with Sharding

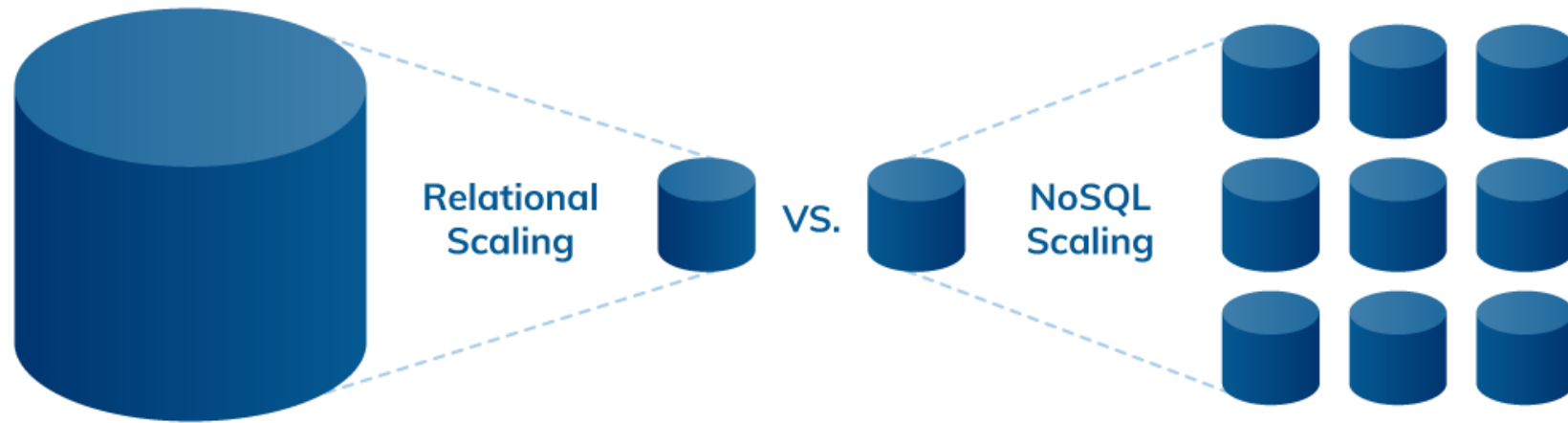
Data Management,  
Performance and  
Scalability

Sharding divides data  
into set of horizontal  
partitions or "shards"

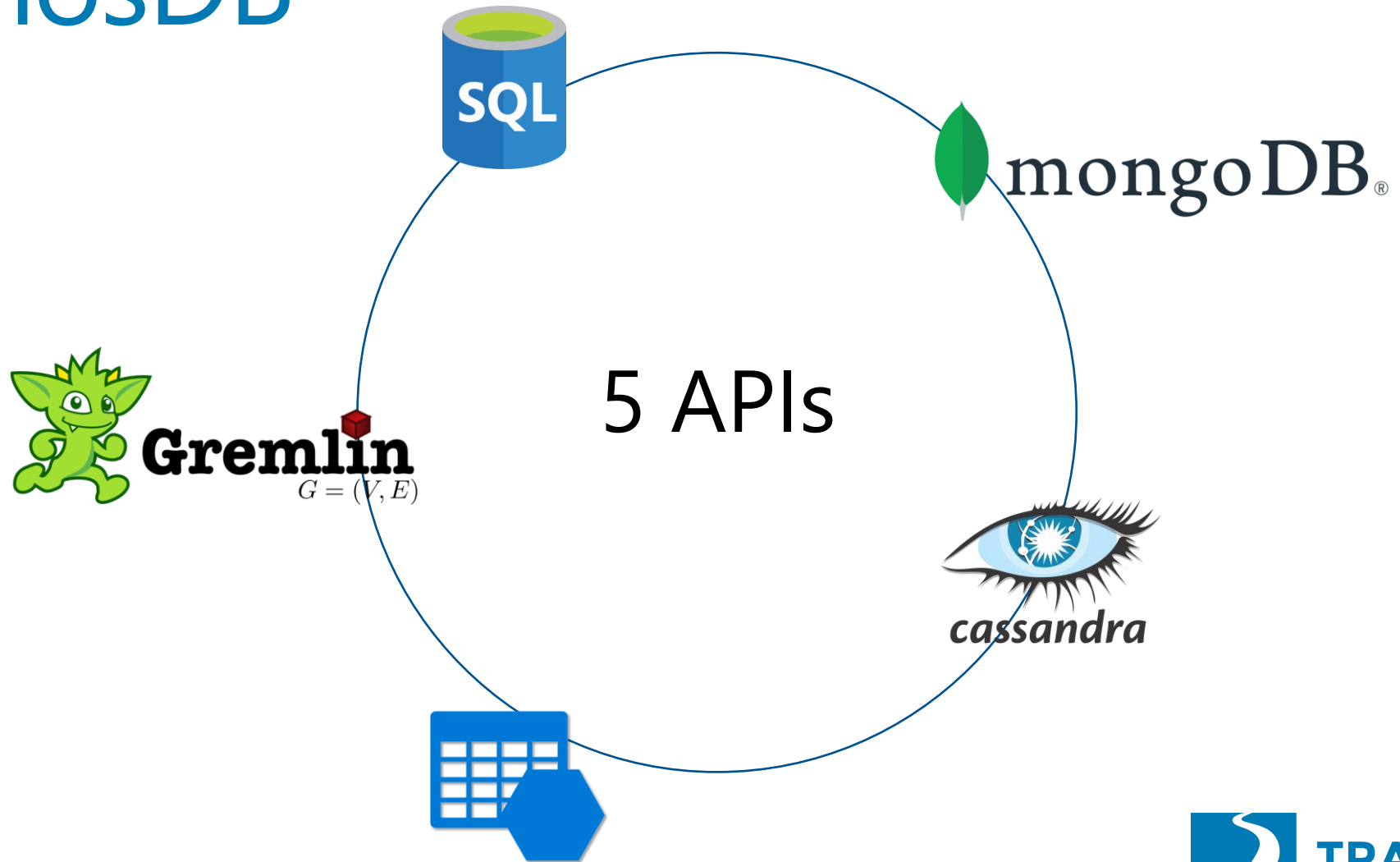
Ex: Year, Tenant  
(user/account)



# 6 | CosmosDB



# CosmosDB





# 7 | DevOps in the Cloud



Azure  
Boards

Plan, track, and discuss work across teams, deliver value to your users faster.



Azure  
Repos

Unlimited cloud-hosted private Git repos. Collaborative pull requests, advanced file management, and more.



Azure  
Pipelines

CI/CD that works with any language, platform, and cloud. Connect to GitHub or any Git provider and deploy continuously to any cloud.



Azure  
Test Plans

The test management and exploratory testing toolkit that lets you ship with confidence.



Azure  
Artifacts

Create, host, and share packages. Easily add artifacts to CI/CD pipelines.

# 8 | Containers and Orchestration

~~Azure Container Instances (ACI)~~

VMs

**Azure Container Service (ACS)**

Kubernetes, DC/OS, or Docker Swarm clusters

**Azure Kubernetes Service (AKS)**

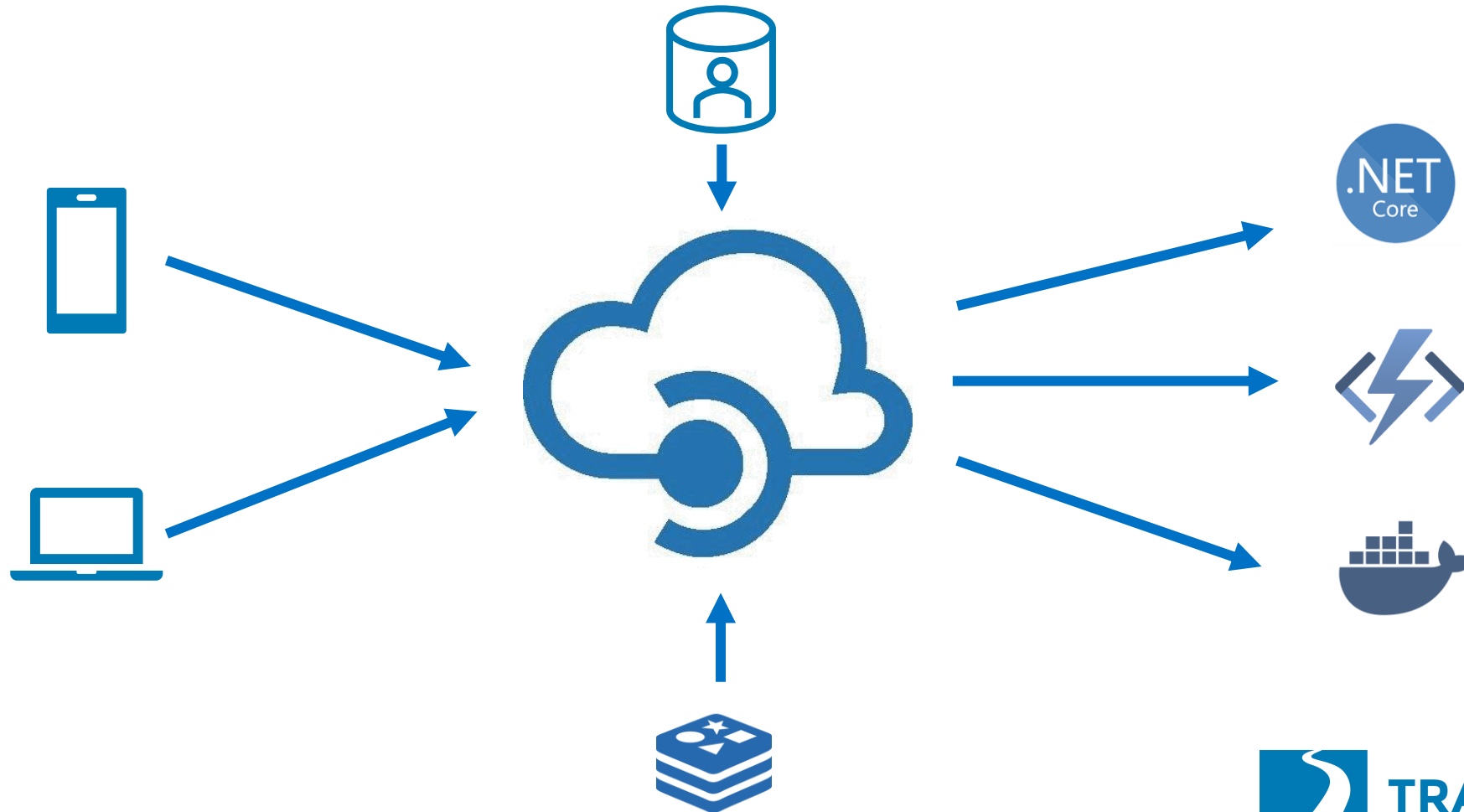
"Serverless" Kubernetes

**Web App for Containers**

App Service for Docker Containers

**Azure Service Fabric**

# 9 | Azure API Management



# 10 & 11 | Azure Service Fabric & Service Fabric Mesh

## **Azure Service Fabric**

- ≈ Like Kubernetes+ (adds SDK, updates, CI/CD, rollback, etc)
- ≈ Dog food
- ≈ On-prem, multi-cloud, and OS-neutral
- ≈ Cluster/node/VM-based (CPaaS)

## **Azure Service Fabric Mesh**

- ≈ Like AKS+
- ≈ True PaaS, managed clusters
- ≈ "Serverless" Service Fabric

# If You Give \$200, So Will I!

## **[bit.ly/chicago-cloud](https://bit.ly/chicago-cloud)**

*"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 56,759 projects in 29 countries, benefiting over 11.4 million people."*

*"4/4 Stars"  
- CharityNavigator.org*



charity: water

# Recap

Assumptions

Background

7 Patterns

11 Technologies

# Further Reading

## **Architecting Cloud Native .NET Applications for Azure**

<https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/>

## **eShopOnContainers reference app**

<https://github.com/dotnet-architecture/eShopOnContainers>

## **Serverless apps: Architecture, patterns, and Azure implementation**

<https://docs.microsoft.com/en-us/dotnet/architecture/serverless/>

## **.NET Microservices: Architecture for Containerized .NET Applications**

<https://docs.microsoft.com/en-us/dotnet/architecture/microservices/>

# Thanks! Questions?

Jonathan "J." Tower

Partner & Principal Consultant

Trailhead Technology Partners



🏆 Six-time Microsoft MVP

👤 Business owner

📅 Beer City Code organizer

✉️ [jtower@trailheadtechnology.com](mailto:jtower@trailheadtechnology.com)

🌐 [trailheadtechnology.com/blog](https://trailheadtechnology.com/blog)

🐦 [jtowermi](https://twitter.com/jtowermi)

[github.com/jonathantower/cloud-native-dotnet](https://github.com/jonathantower/cloud-native-dotnet)