Microsoft

Spring Cloud in the Cloud

For Frictionless Microservices

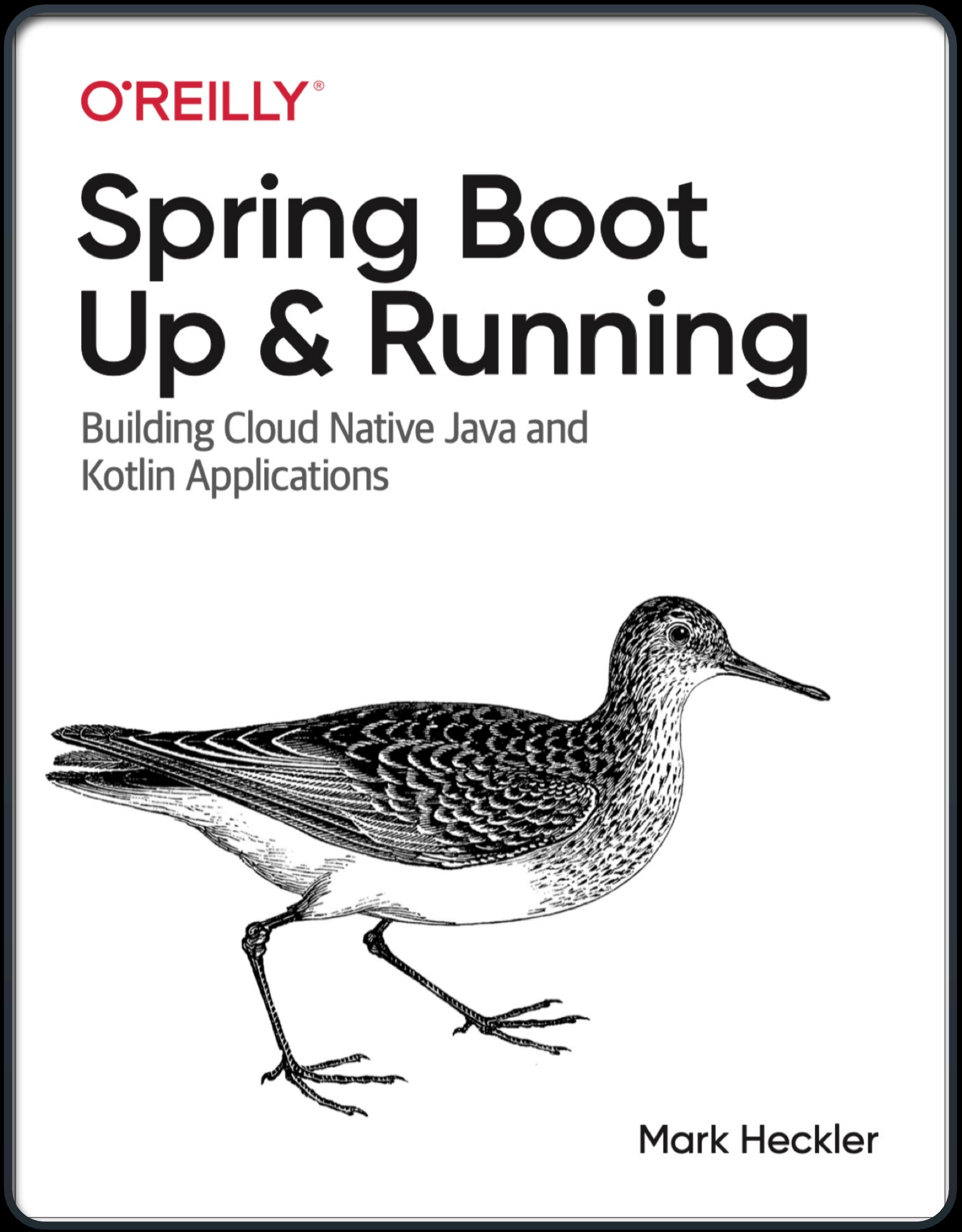
Mark Heckler
Principal Cloud Advocate, Java/JVM Languages
markheckler@microsoft.com
mark@thehecklers.com
@mkheck

Who am I?

- Architect & Developer
- Advocate
- Author
- Java Champion, Rockstar
- Kotlin Developer Expert
- Pilot

Latest book

https://bit.ly/springbootbook
@springbootbook



O'REILLY®

Spring Boot Up & Running

Building Cloud Native Java and Kotlin Applications

Mark Heckler

The Plan

“To achieve great things, two things are needed:
a plan, and not quite enough time.”

- Leonard Bernstein

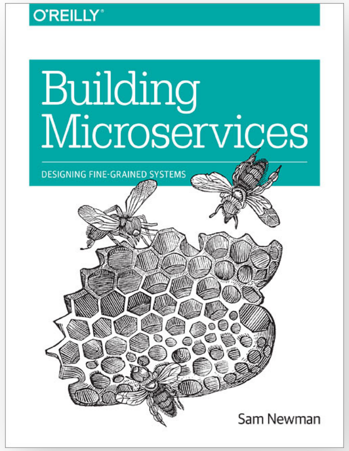
The Plan

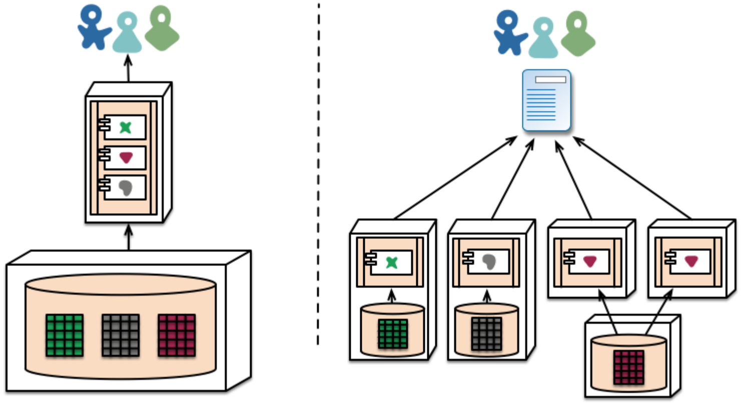
- Increase abilities
- Decrease complexities
- Deliver capabilities

Cloud-native architectures

Benefits of microservices architectures

1. Scalability: scale **each** service based on load
2. High availability: create more resilient systems
3. Velocity: continuous delivery, team autonomy, and unbundled release trains





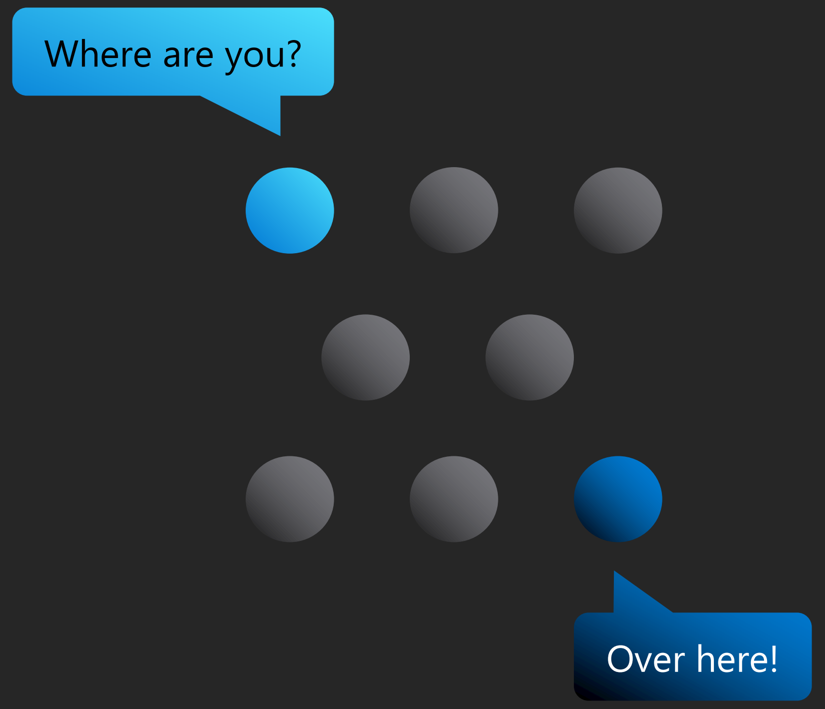
monolith - single database

microservices - application databases

Service Discovery

A microservice needs to discover the IP address/port number of dependency

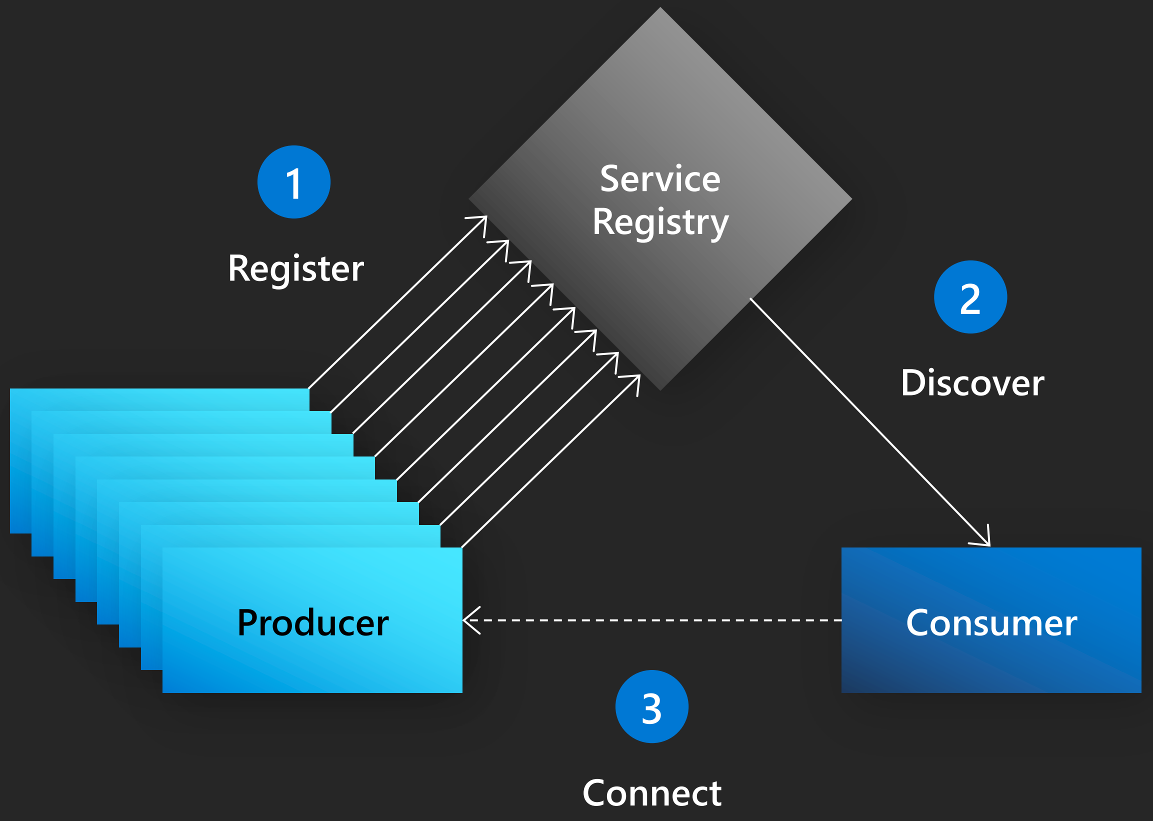
- Microservices are scaled out and in dynamically, thus have ephemeral IP addresses
- How does one determine which microservices are healthy and ready to accept requests?



Service Discovery (cont.)

Uses Service IDs, not URLs, to locate services

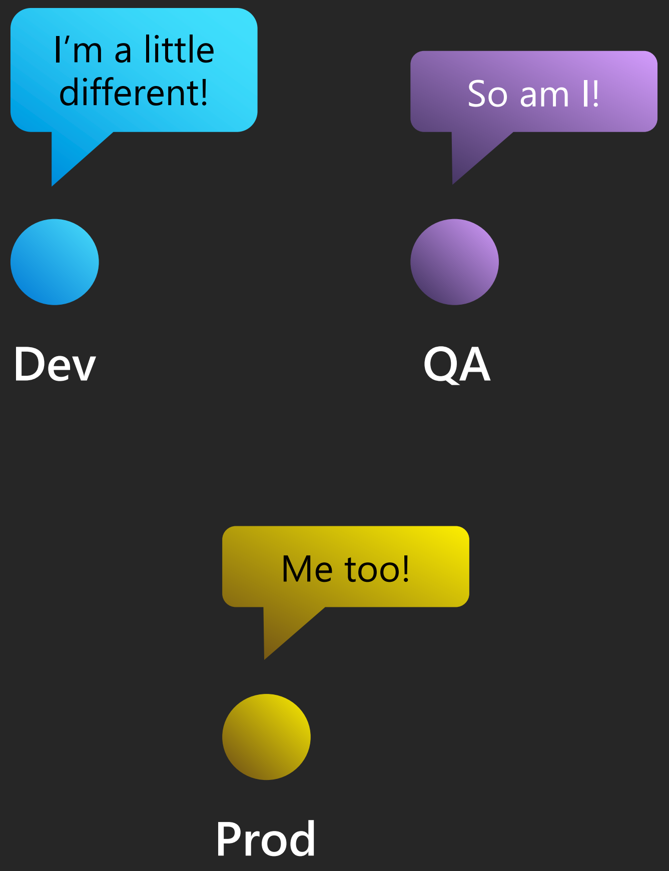
Client-side or server-side load balancing



Configuration, externalized

A microservice is deployed to multiple environments (dev/QA/prod)

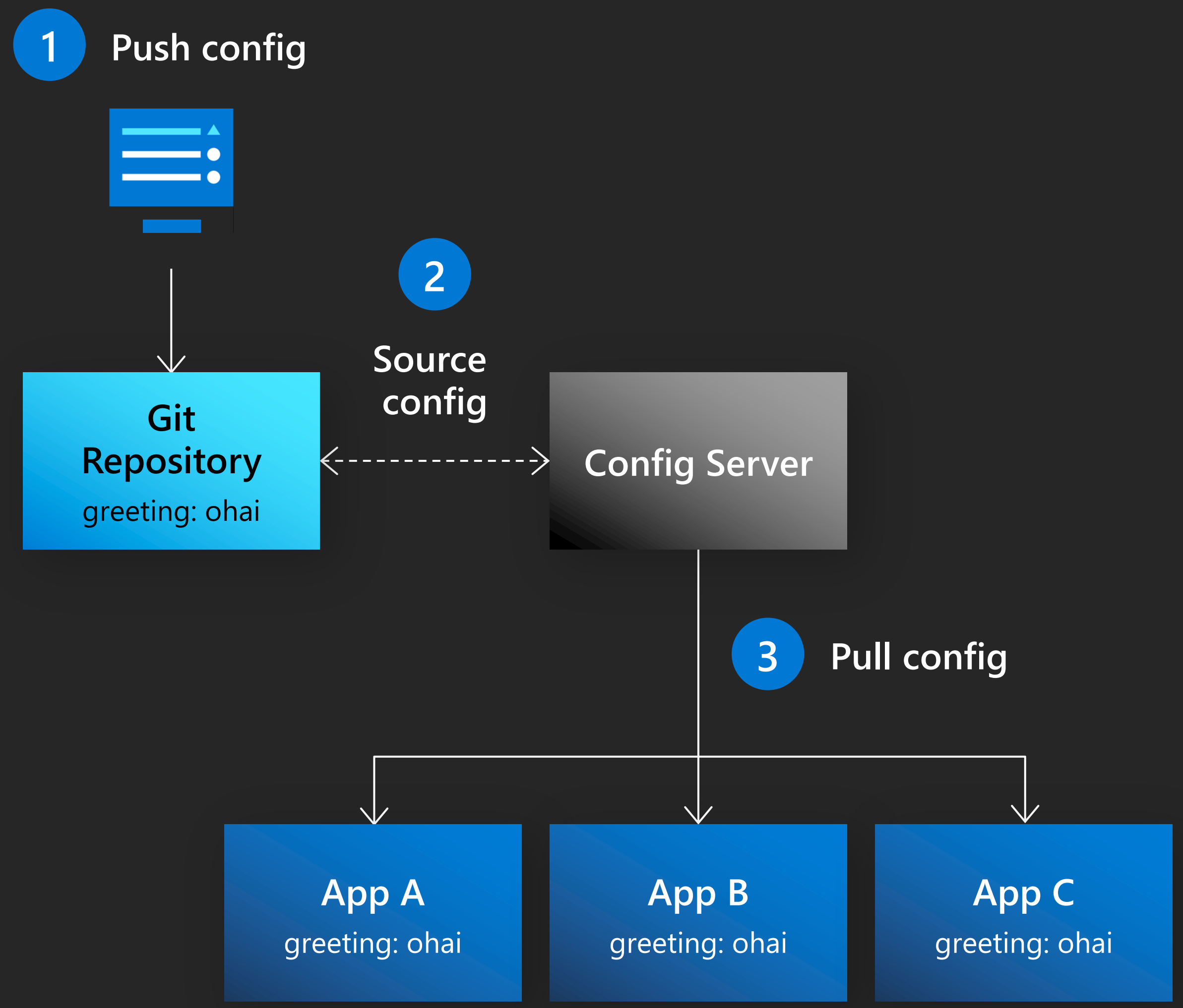
- There are configuration differences between each environment
- Multiple instances of a microservice run in each environment to meet scalability and availability requirements
- Multiple versions of a microservice can be running at the same time to ensure zero downtime deployments, A/B testing, or backward compatibility
- Risk management requires an audit trail of every configuration change made



Configuration (cont.)

Local or remote git repositories

Configuration specified by environment, service, or system



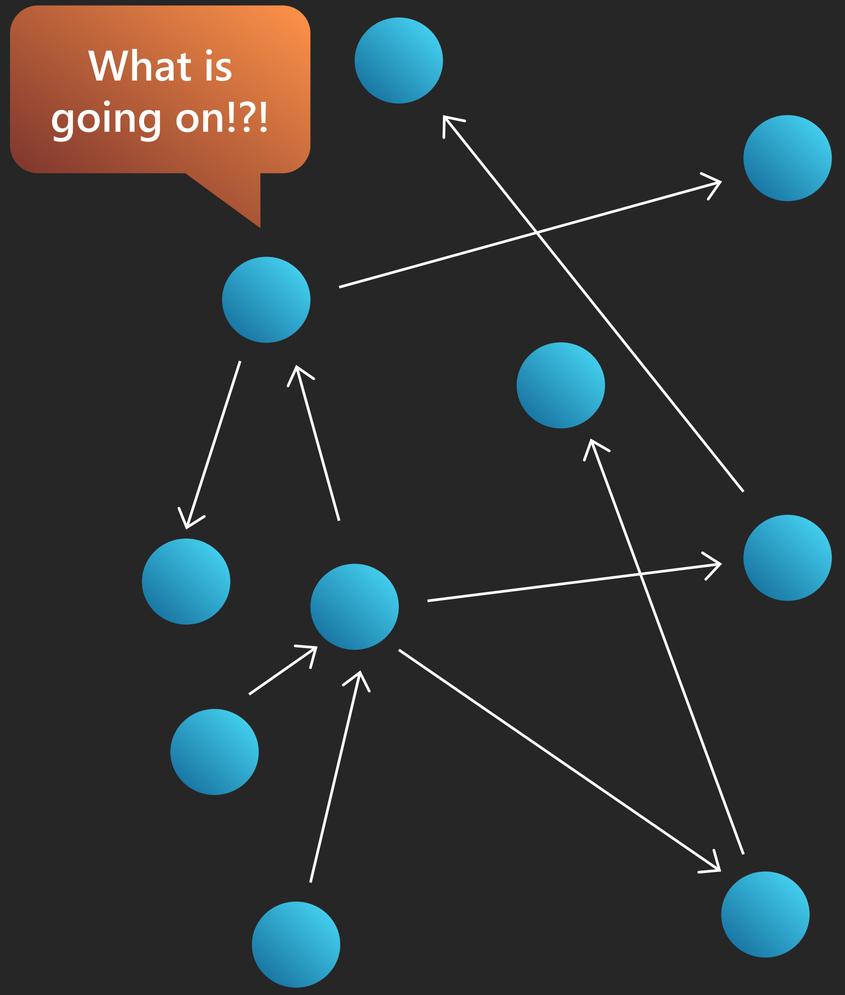
Distributed Tracing

Troubleshooting latency:

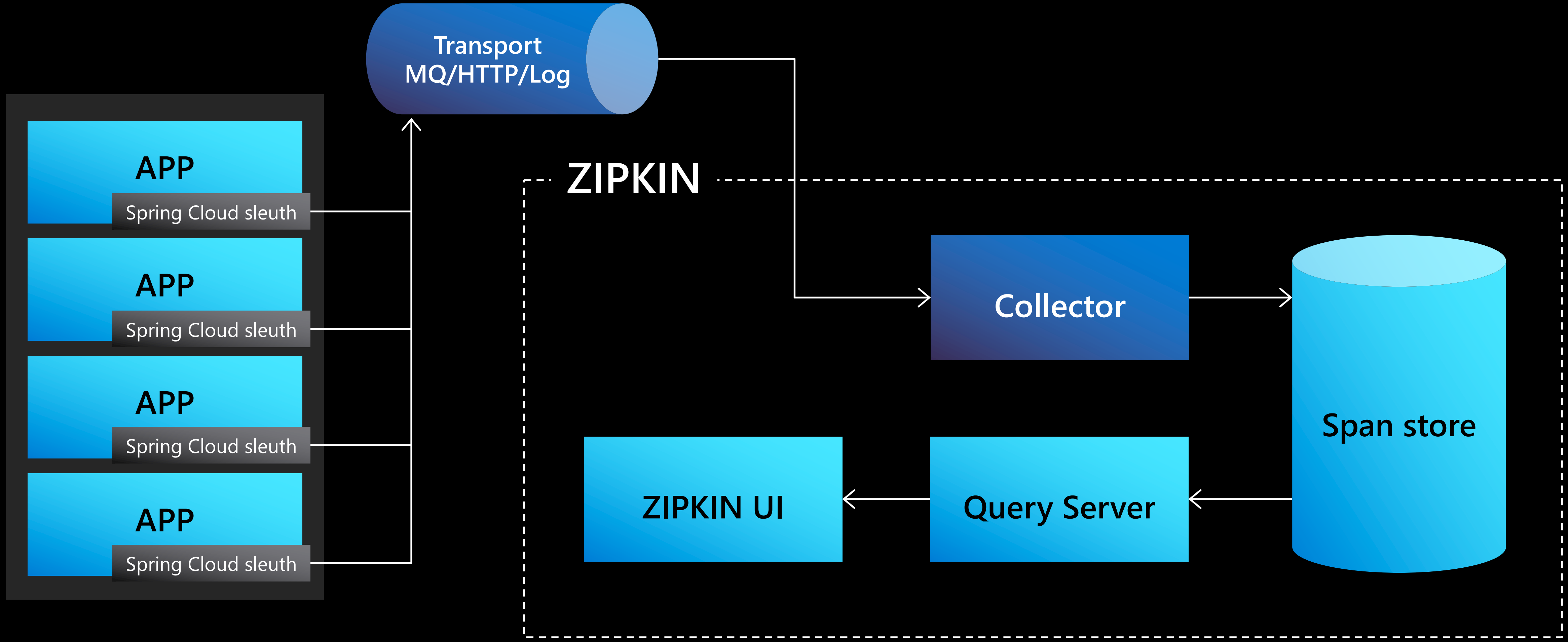
- When was the event? How long did it take?
- How do I know it was slow?
- Why did it take so long?
- Which microservice was responsible?

Distributed tracing:

- Distributed tracing is a process of collecting end-to-end transaction graphs in near real-time
- A trace represents the entire journey of a request



Distributed tracing throughout



What is Spring?

The best Java application development framework!

Productive

- Accomplish practically any Java development task
- Simple, fully comprehensive, and modular
- Deployable/testable/portable
- Clear and well-documented with great online guides

Popular

- More than 50% of Java developers use Spring Boot (JVM Mag)
- Over 3 million visits to Spring Initializr *every year* (& growing)!
- Permissive Open-Source License (Apache 2.0)
- Vibrant, passionate, and committed community

Trusted

- World’s #1 IoC/Dependency Injection Framework
- Proven in production since 2004
- Enterprise distribution and commercial support available

Spring-based microservices development

Spring Boot

Build anything

Designed to get you up and running as quickly as possible, with minimal upfront configuration of Spring

Spring Cloud

Coordinate anything

Provides a set of tools that makes communication between microservices easier

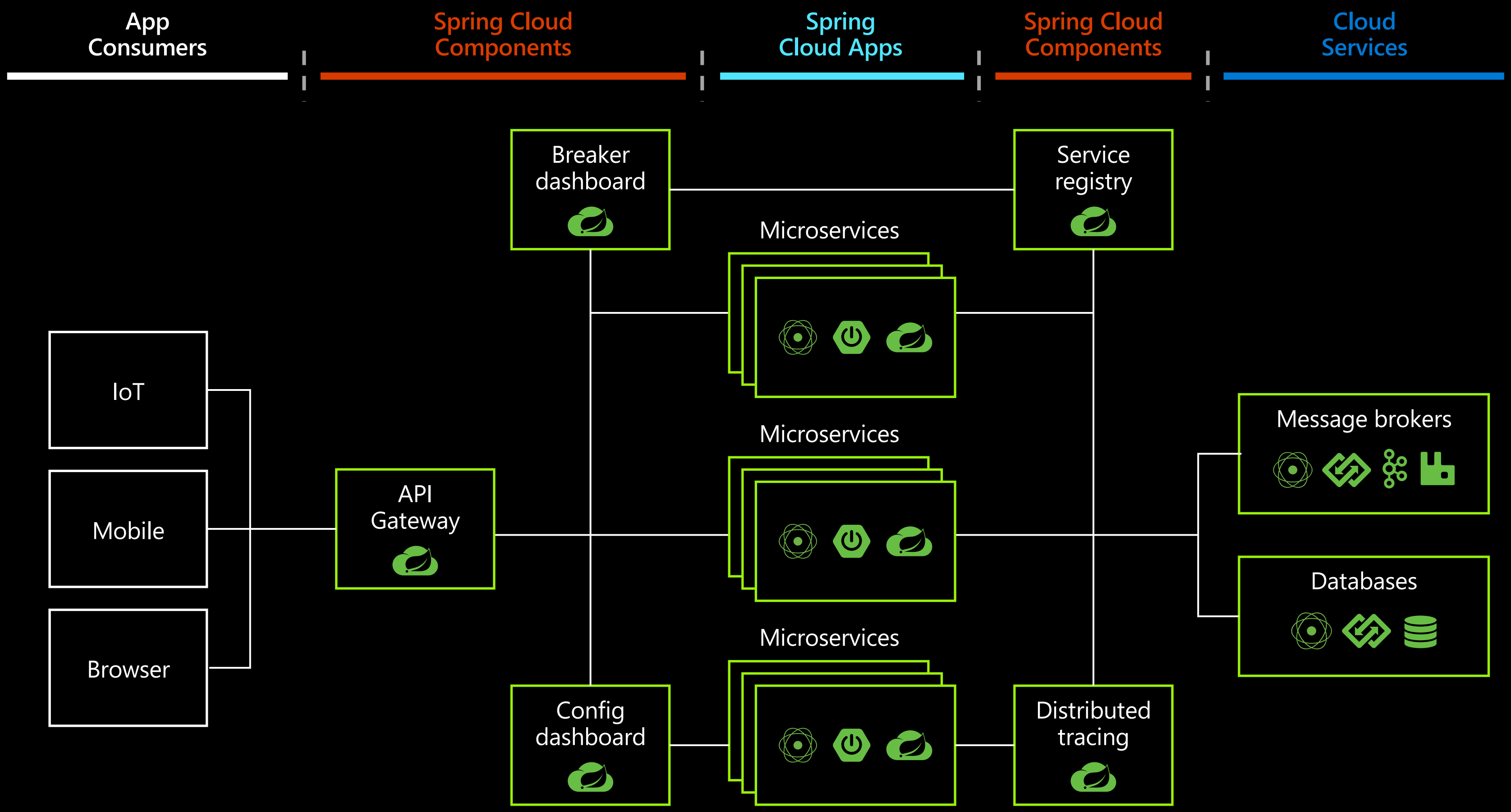
Common production challenges

Configuration, coordination, communication, monitoring, infrastructure, updates, troubleshooting, scaling…

High effort required to manage cloud infrastructure for Spring Boot applications

Application lifecycle is **difficult to manage**

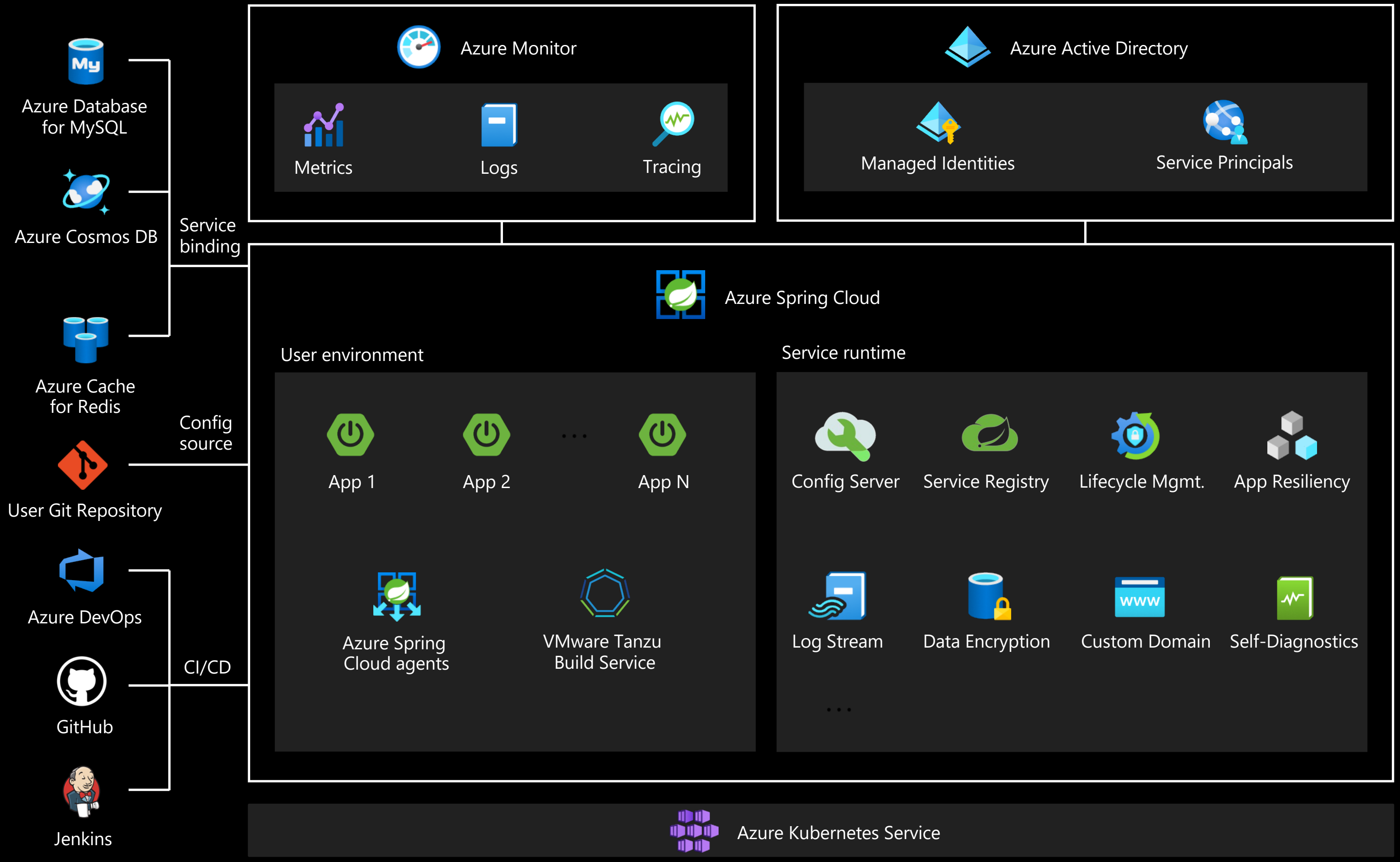
Painful to troubleshoot application issues



Simplify development and deployment

Responsibilities	DIY with Spring Boot	Azure Spring Cloud Service
Application iteration, debugging	⬜	⬜
CI/CD	⬜	🟩⬛
Build and manage clusters	⬜	🟩⬛
Host Spring Cloud middleware	⬜	🟩⬛
Monitoring and logging	⬜	🟩⬛
Scaling	⬜	🟩⬛
Patching	🟩⬜	🟩⬛
Support	🟩⬜	🟩⬛

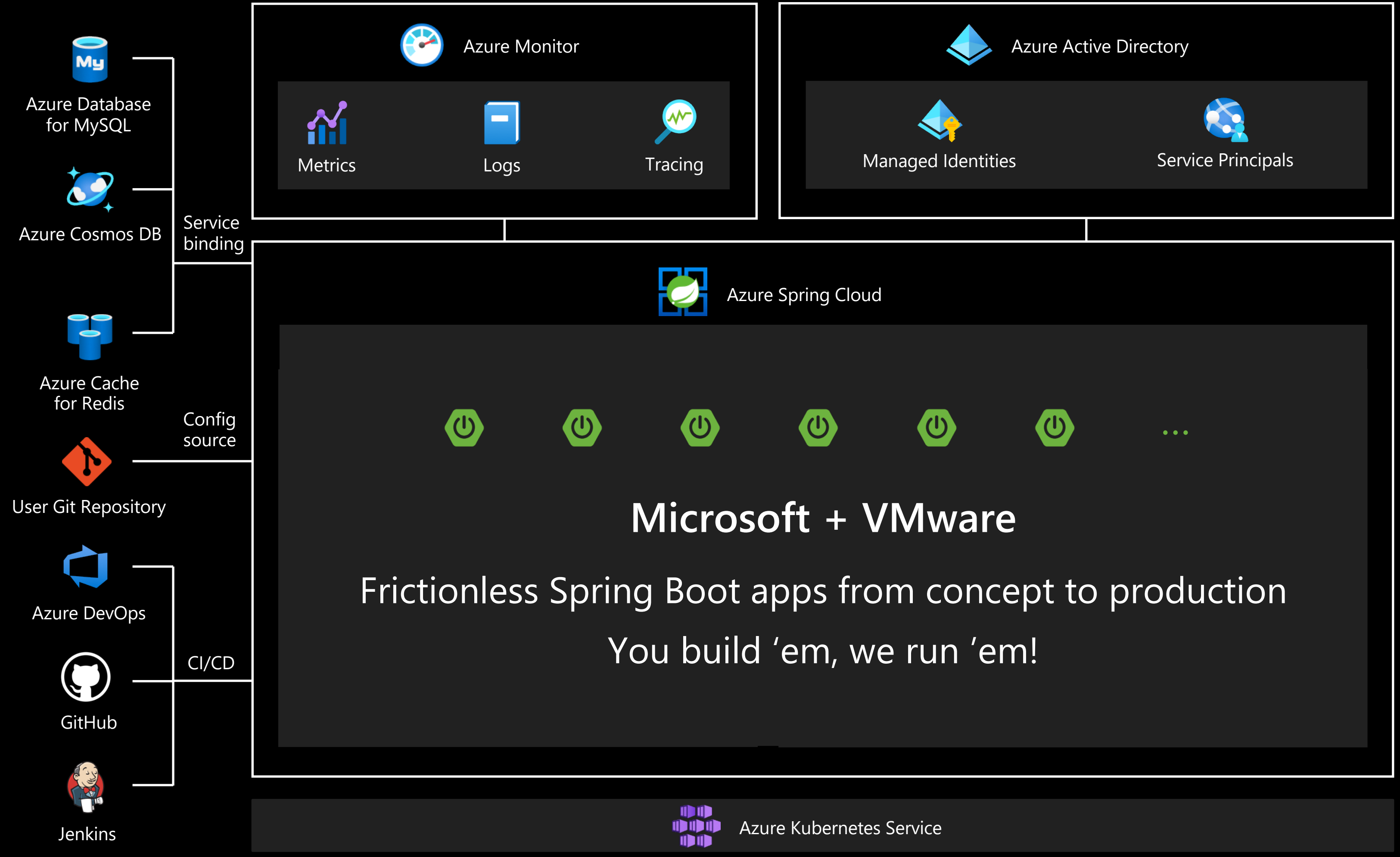
⬜ Customer 🟩 VMware ⬛ Microsoft



Developer-first focus for entire lifecycle

Responsibilities	DIY with Spring Boot	Azure Spring Cloud Service
Application iteration, debugging	⬜	⬜
CI/CD	⬜	🟩🟦
Build and manage clusters	⬜	🟩🟦
Host Spring Cloud middleware	⬜	🟩🟦
Monitoring and logging	⬜	🟩🟦
Scaling	⬜	🟩🟦
Patching	🟩⬜	🟩🟦
Support	🟩⬜	🟩🟦

⬜ Customer 🟩 VMware 🟦 Microsoft



Microsoft + VMware

Frictionless Spring Boot apps from concept to production

You build ’em, we run ’em!

Azure Spring Cloud

Fully managed Spring Cloud service for distributed systems, jointly built and operated with VMware

More choices and full integration into Azure’s ecosystem and services

Fully managed infrastructure

Built-in app lifecycle management

Ease of monitoring

DEMO TIME!

Spring integrations with Azure

Leverage >30 starters to connect to Azure services with minimal configuration and/or code changes

Spring Cloud

App Configuration
Event Hubs
Service Bus
Storage
Redis
Functions

Spring Resource

Storage

Spring Data

SQL Database
MySQL
PostgreSQL
Maria DB
Cosmos DB

- SQL
- MongoDB
- Cassandra
- Gremlin

Spring Security

Active Directory (AAD)
AAD B2C

R2DBC

SQL Database
PostgreSQL
MySQL

Spring Cache

Redis Cache

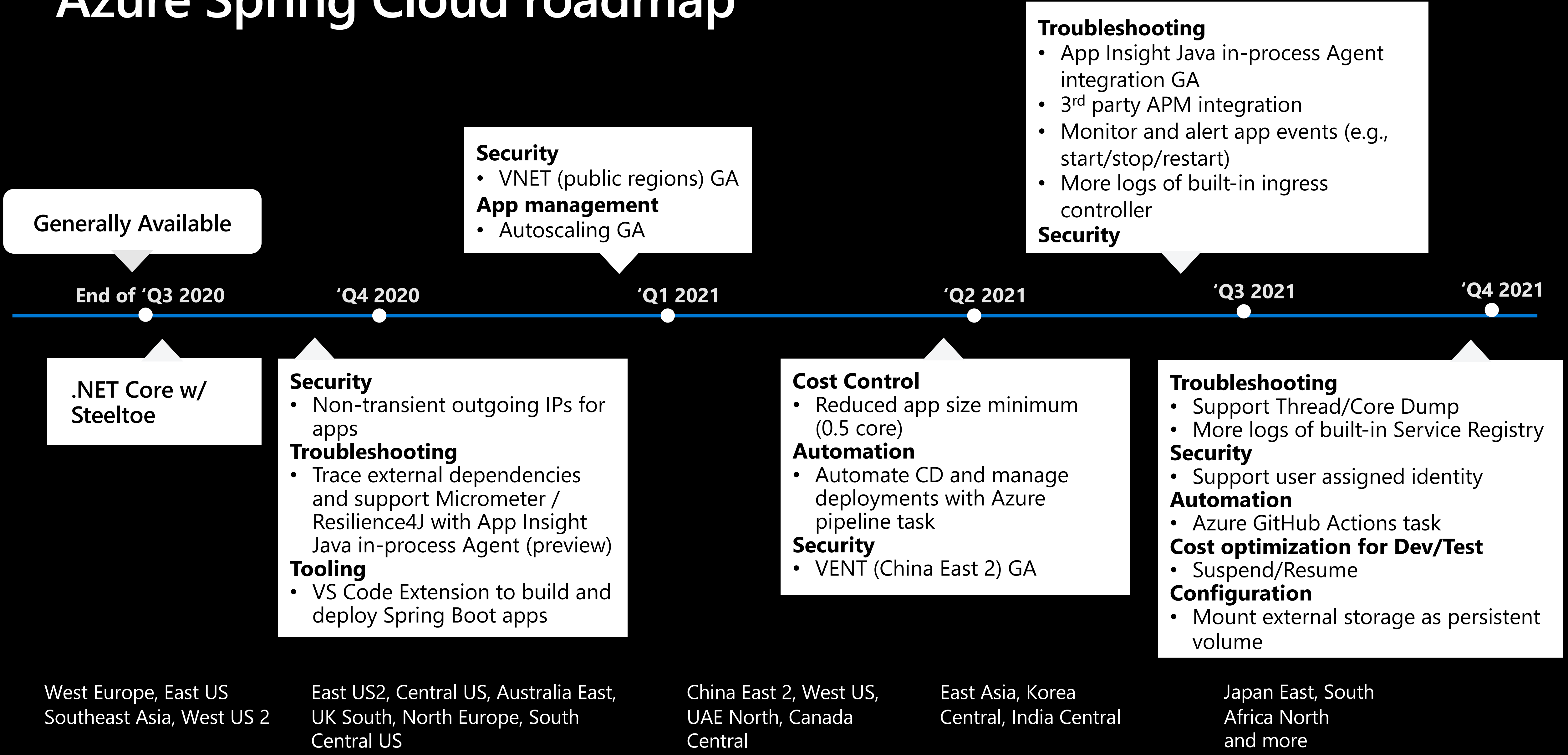
Spring Messaging

Service Bus

Micrometer

Monitor

Azure Spring Cloud roadmap



Generally Available

End of ‘Q3 2020

.NET Core w/ Steeltoe

West Europe, East US Southeast Asia, West US 2

Security
- Non-transient outgoing IPs for apps

Troubleshooting
- Trace external dependencies and support Micrometer / Resilience4J with App Insight Java in-process Agent (preview)

Tooling
- VS Code Extension to build and deploy Spring Boot apps

‘Q4 2020

East US2, Central US, Australia East, UK South, North Europe, South Central US

Security
- VNET (public regions) GA

App management
- Autoscaling GA

‘Q1 2021

China East 2, West US, UAE North, Canada Central

Cost Control
- Reduced app size minimum (0.5 core)

Automation
- Automate CD and manage deployments with Azure pipeline task

Security
- VENT (China East 2) GA

‘Q2 2021

East Asia, Korea Central, India Central

Troubleshooting
- App Insight Java in-process Agent integration GA
- 3rd party APM integration
- Monitor and alert app events (e.g., start/stop/restart)
- More logs of built-in ingress controller

Security

‘Q3 2021

Japan East, South Africa North and more

Troubleshooting
- Support Thread/Core Dump
- More logs of built-in Service Registry

Security
- Support user assigned identity

Automation
- Azure GitHub Actions task

Cost optimization for Dev/Test
- Suspend/Resume

Configuration
- Mount external storage as persistent volume

‘Q4 2021

Learn more

- https://aka.ms/get-started-with-azure-spring-cloud

Spring on Azure resources

- https://azure.microsoft.com/services/spring-cloud/
- https://docs.microsoft.com/azure/developer/java/spring-framework/

Self-paced workshops for Azure Spring Cloud

- https://docs.microsoft.com/learn/modules/azure-spring-cloud-workshop/
- https://github.com/microsoft/azure-spring-cloud-training

Feedback welcome!

- https://aka.ms/springazure
- @mkheck on Twitter

Thank you!