Azure Service Fabric

Jian Wu
China CAT, Microsoft

Microsoft

Agenda

- Microservice and Service Fabric Cluster
- First microservice in Service Fabric
- Building microservice applications with Service Fabric
- Managing Service Fabric and microservice
- Service Fabric Cases
- Resources and Getting Started

Microservice and Service Fabric Cluster

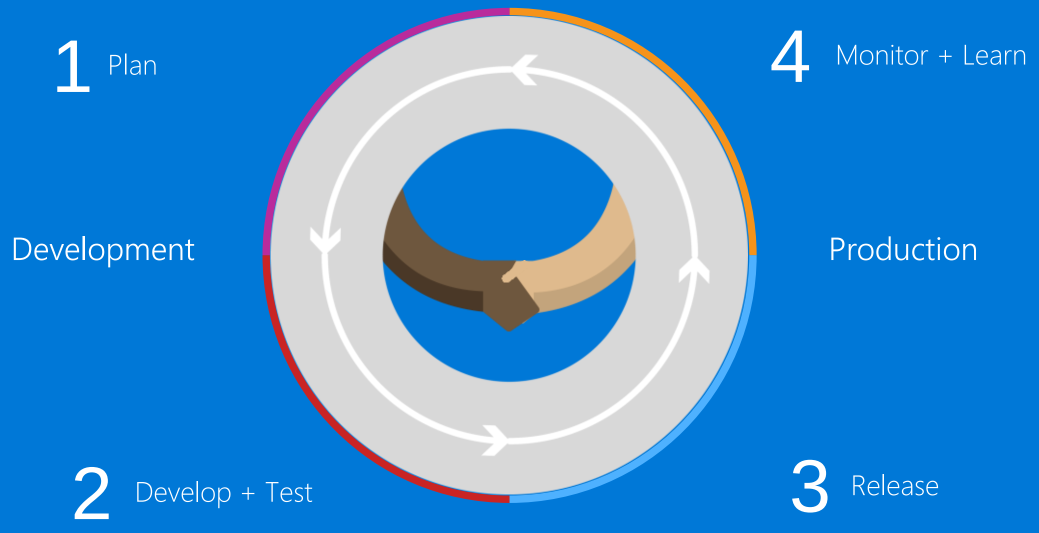
Microsoft

Some Application Considerations

- You do not own the hardware
- Be prepared for failures
- Scale is unpredictable. Can you scale efficiently?
- Managing services is often harder than building services. Do you have the right operational telemetry for visibility?
- No downtime upgrades
- Are the costs understandable and controllable? Do you know the density and capacity of your services?
- Care upfront about security
- Developer productivity. Build the app not a platform.

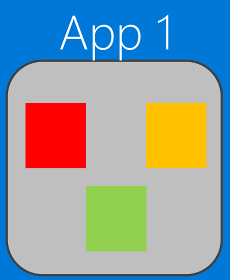
Why a microservices approach?

- Continually evolving applications
- Faster delivery of features and capabilities to respond to customer expectations
- Build and operate a service at scale

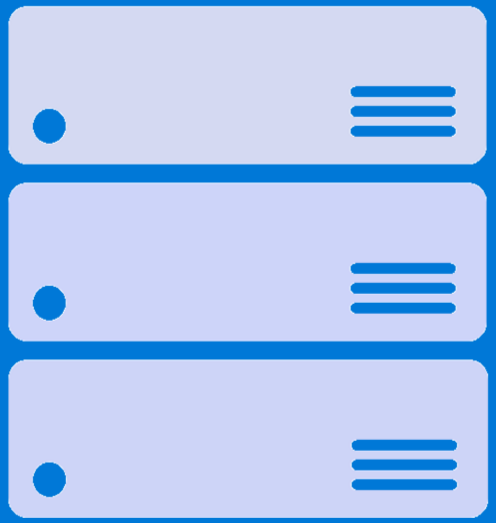


Monolithic application approach

- A monolith app contains domain specific functionality and is normally divided by functional layers such as web, business and data

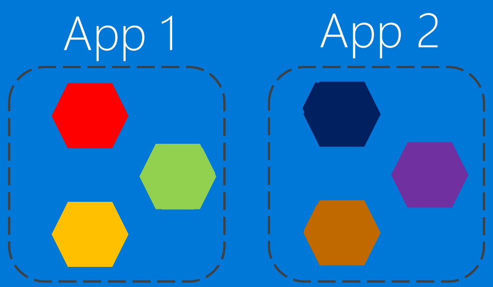


- Scales by cloning the app on multiple servers/VMs/Containers



Microservices application approach

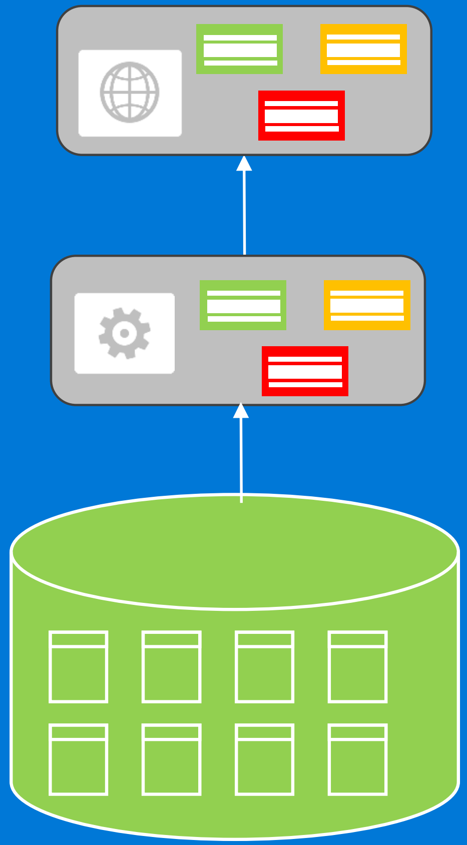
- A microservice application separates functionality into separate smaller services.



- Scales out by deploying each service independently creating instances of these services across servers/VMs/containers

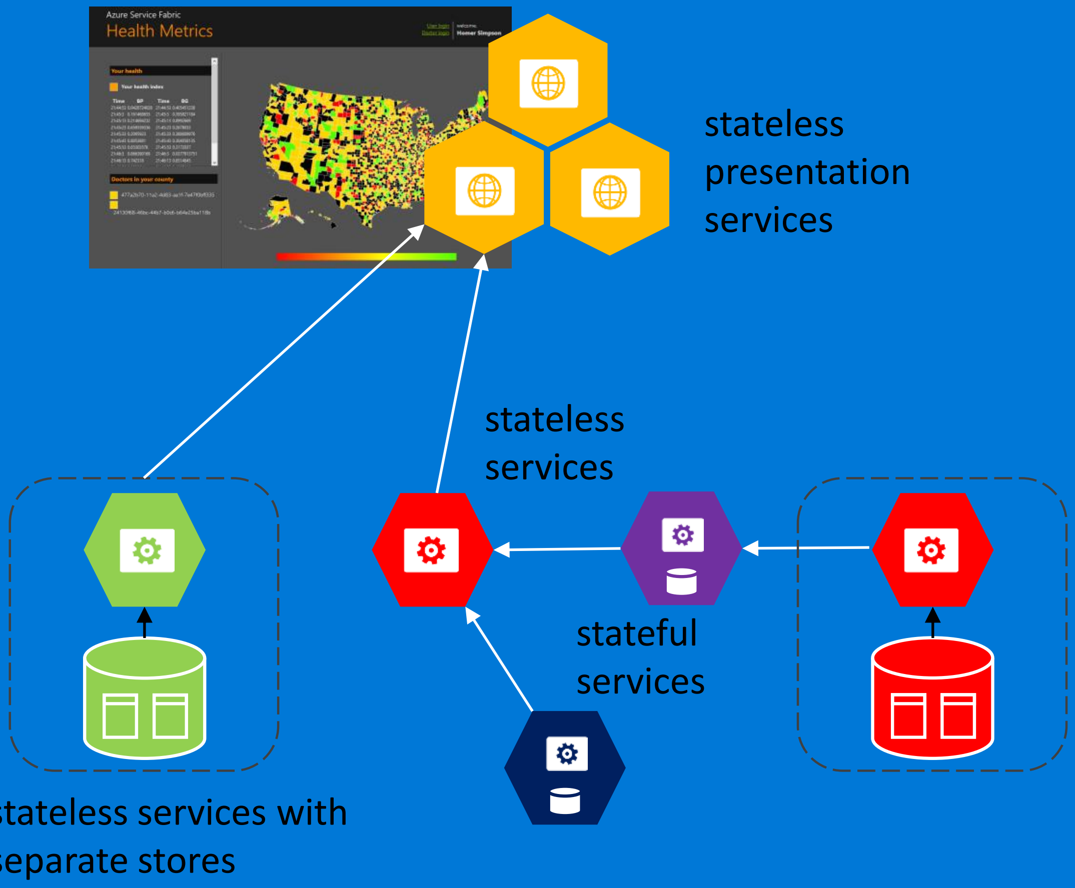
State in Monolithic approach

- Single monolithic database
- Tiers of specific technologies

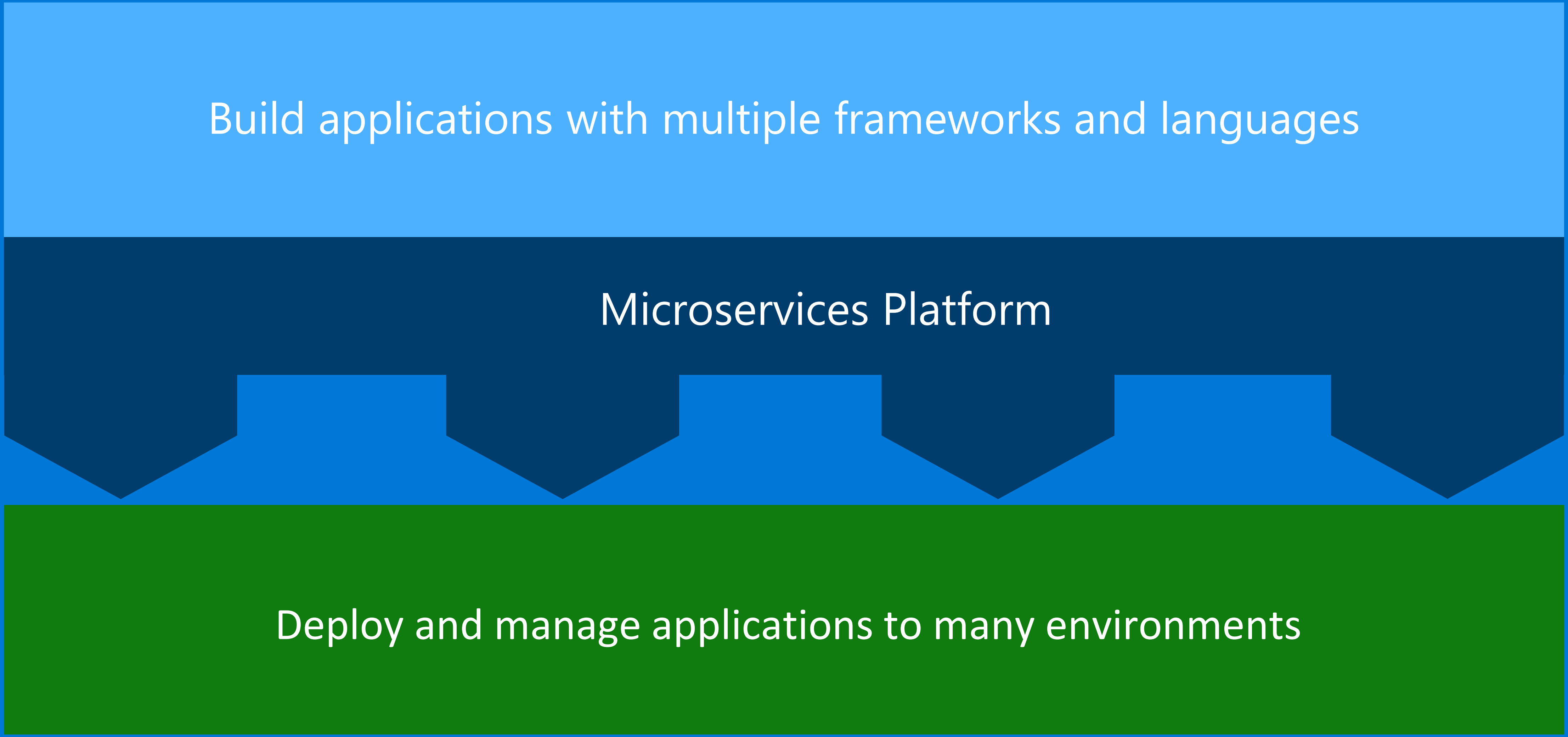


State in Microservices approach

- Graph of interconnected microservices
- State typically scoped to the microservice
- Variety of technologies used



Microservices platform requirements

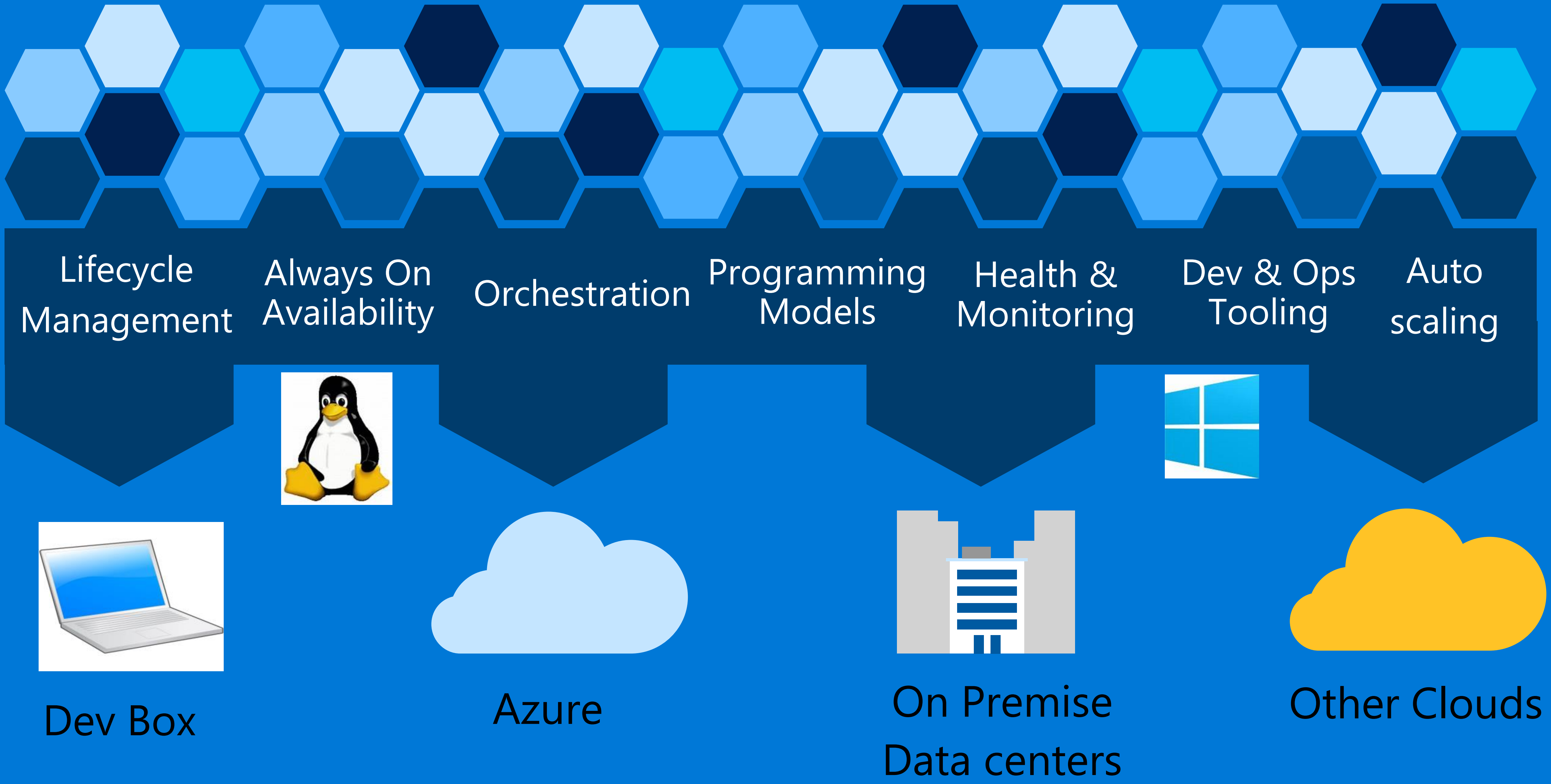


Build applications with multiple frameworks and languages

Microservices Platform

Deploy and manage applications to many environments

Azure Service Fabric



Lifecycle Management

Always On Availability

Orchestration

Programming Models

Health & Monitoring

Dev & Ops Tooling

Auto scaling

Dev Box

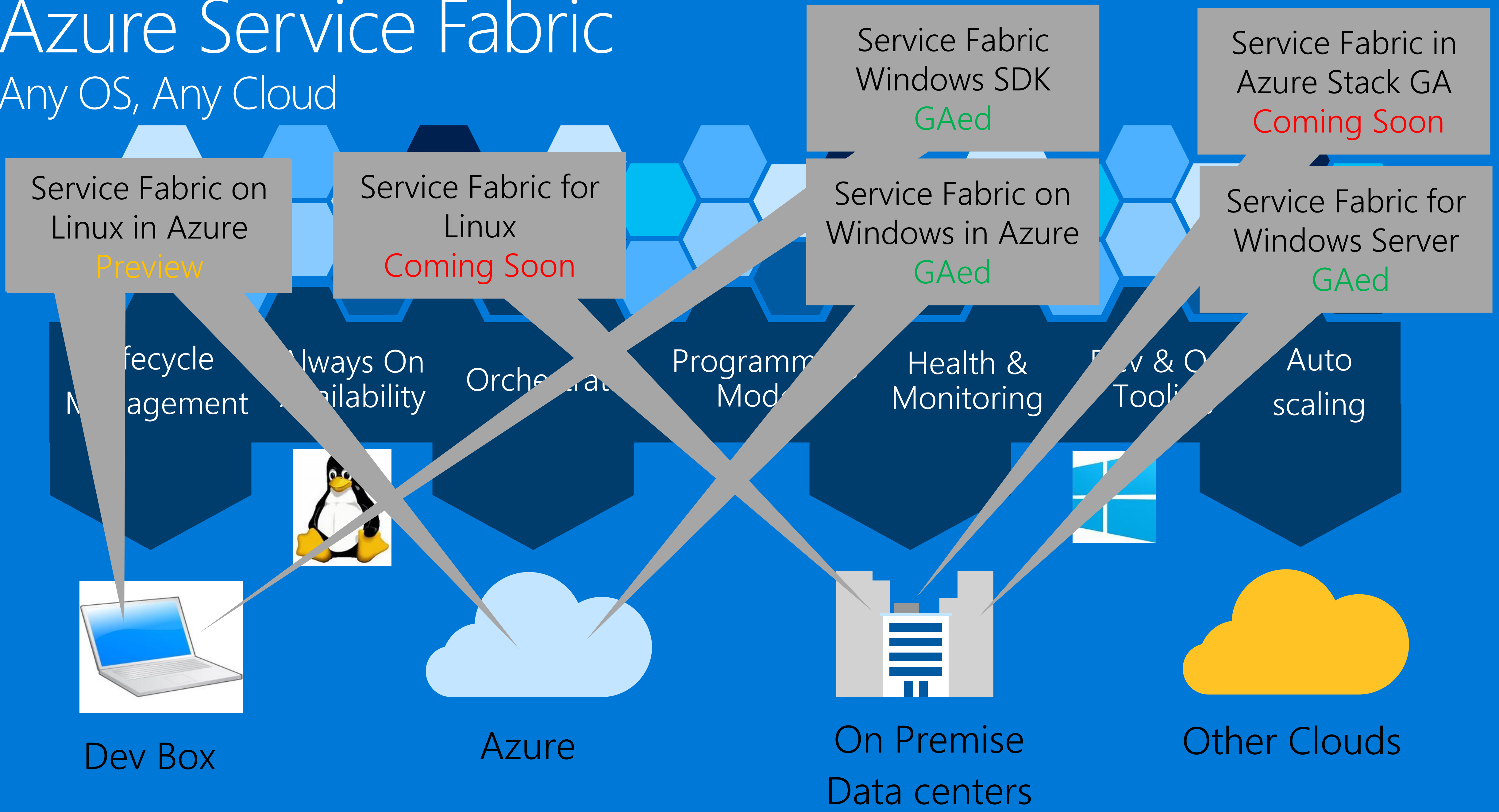
Azure

On Premise Data centers

Other Clouds

Azure Service Fabric

Any OS, Any Cloud



Service Fabric on Linux in Azure Preview

Service Fabric for Linux Coming Soon

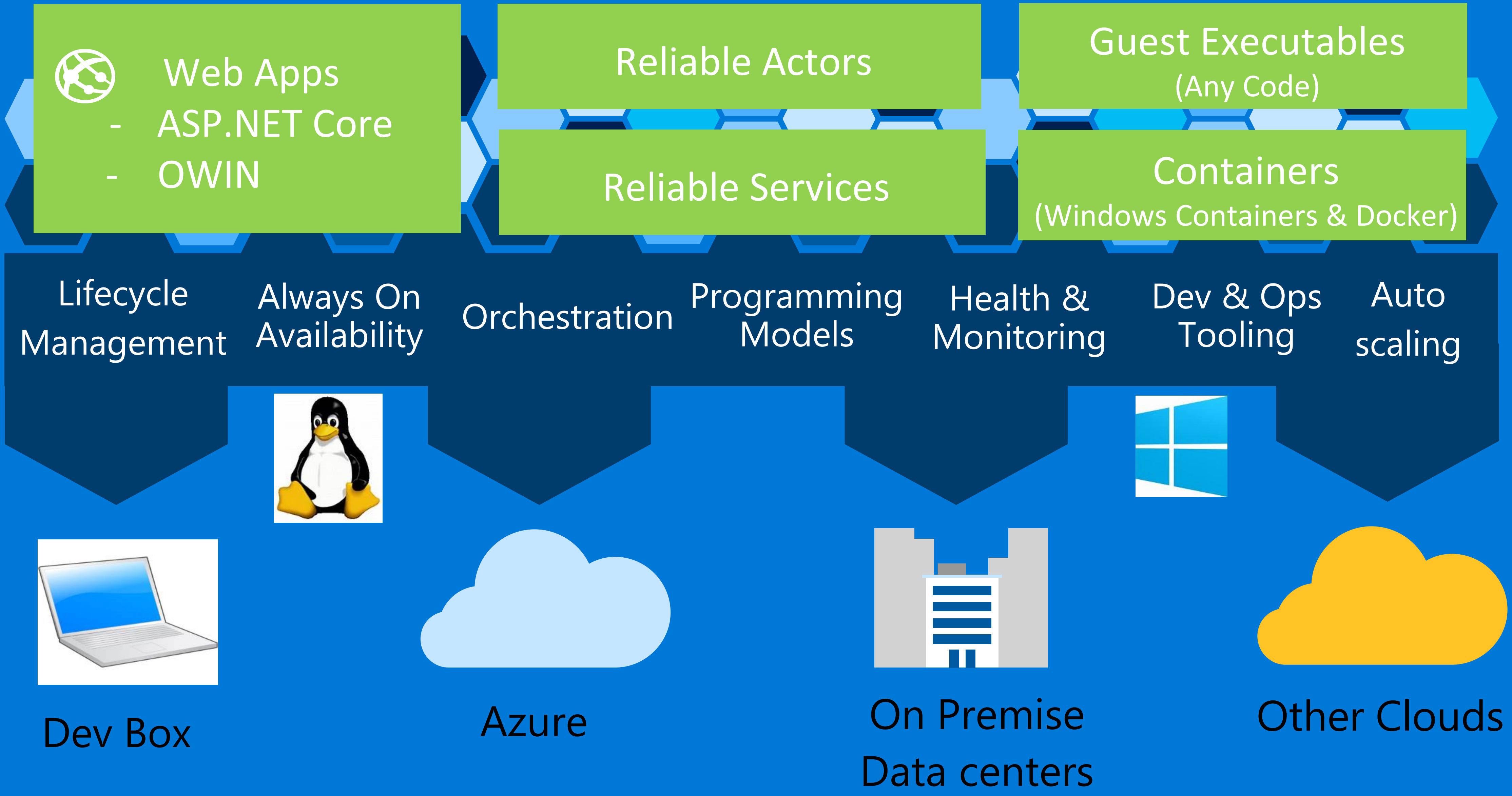
Service Fabric Windows SDK GAed

Service Fabric on Windows in Azure GAed

Service Fabric in Azure Stack GA Coming Soon

Service Fabric for Windows Server GAed

Azure Service Fabric microservices



Web Apps

- ASP.NET Core
- OWIN

Reliable Actors

Reliable Services

Guest Executables
(Any Code)

Containers
(Windows Containers & Docker)

Lifecycle Management

Always On Availability

Orchestration

Programming Models

Health & Monitoring

Dev & Ops Tooling

Auto scaling

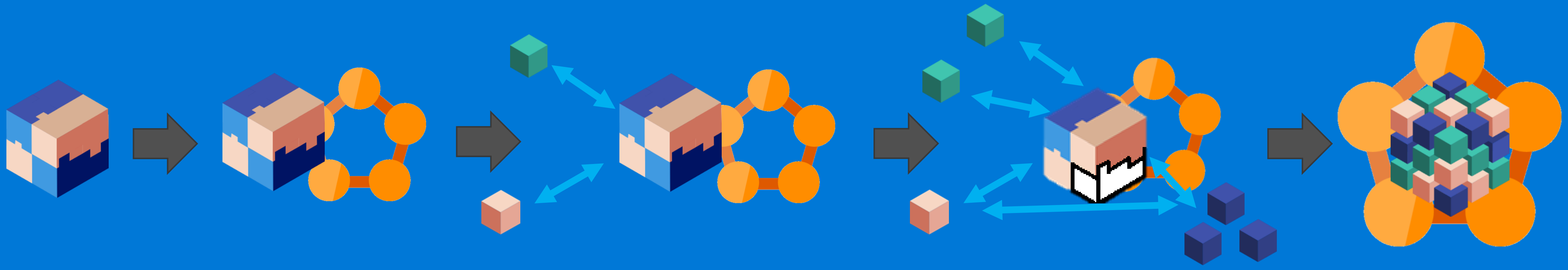
Dev Box

Azure

On Premise Data centers

Other Clouds

Migrating a traditional application to microservices

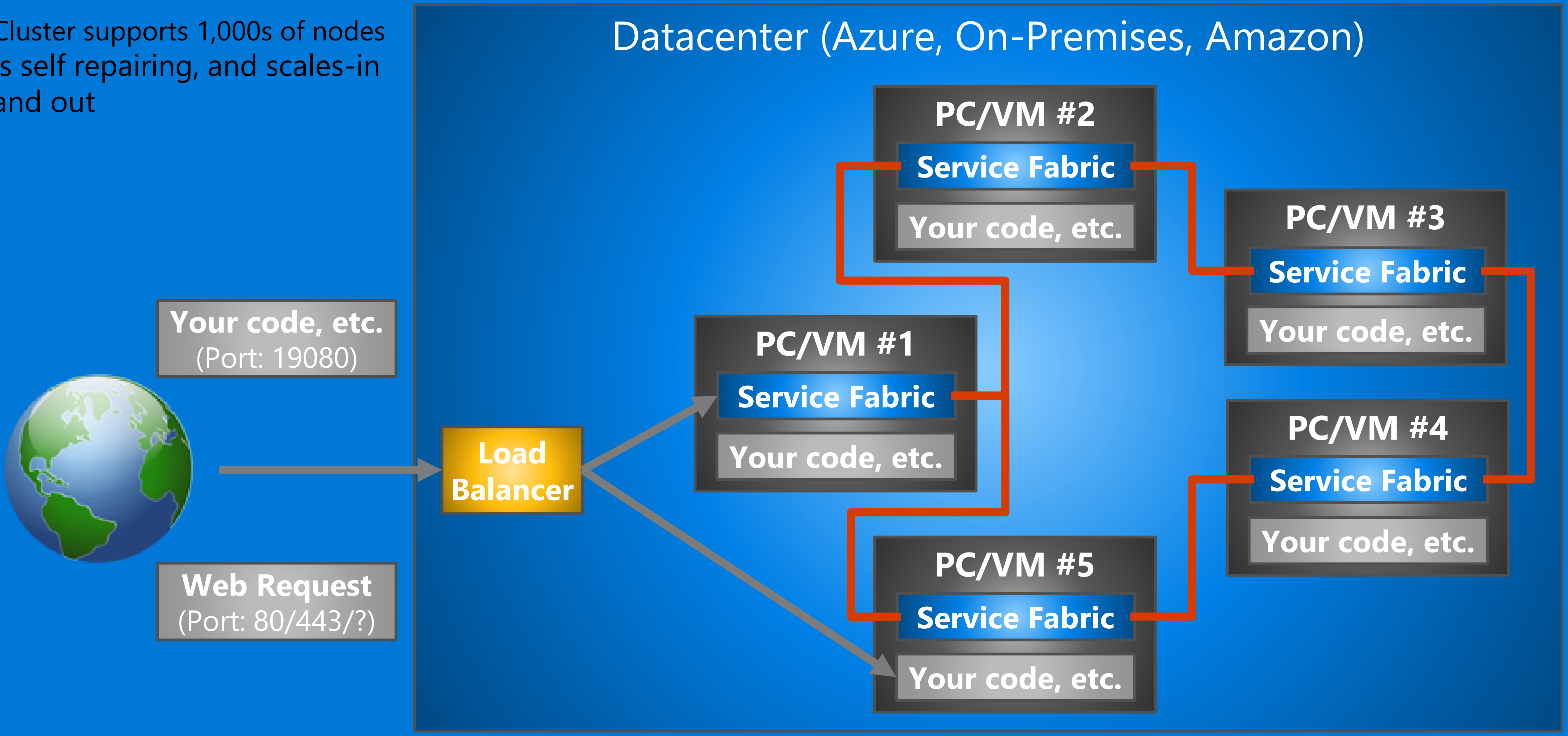


1) Traditional app

...You can stop at any stage

Service Fabric Cluster in Azure

Cluster supports 1,000s of nodes is self repairing, and scales-in and out



Your code, etc.
(Port: 19080)

Web Request
(Port: 80/443/?)

Datacenter (Azure, On-Premises, Amazon)

Load Balancer

PC/VM #1
Service Fabric
Your code, etc.

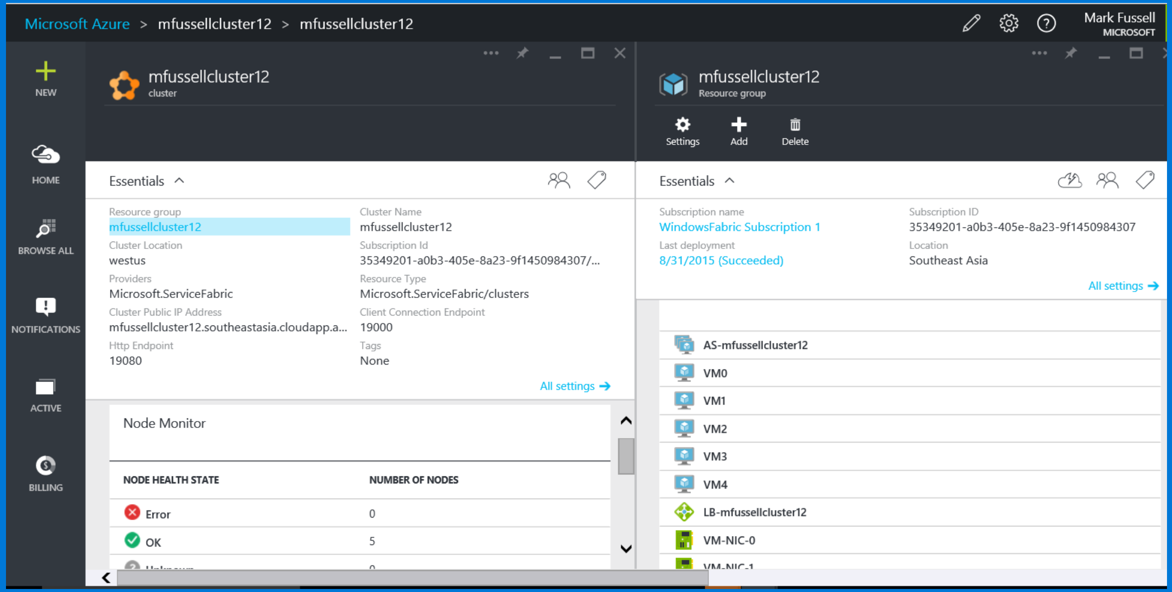
PC/VM #2
Service Fabric
Your code, etc.

PC/VM #3
Service Fabric
Your code, etc.

PC/VM #4
Service Fabric
Your code, etc.

PC/VM #5
Service Fabric
Your code, etc.

Demo: Service Fabric cluster via the portal



mfussellcluster12
cluster

Essentials

Resource group mfussellcluster12	Cluster Name mfussellcluster12
Cluster Location westus	Subscription Id 35349201-a0b3-405e-8a23-9f1450984307/...
Providers Microsoft.ServiceFabric	Resource Type Microsoft.ServiceFabric/clusters
Cluster Public IP Address mfussellcluster12.southeastasia.cloudapp.a...	Client Connection Endpoint 19000
Http Endpoint 19080	Tags None

Node Monitor

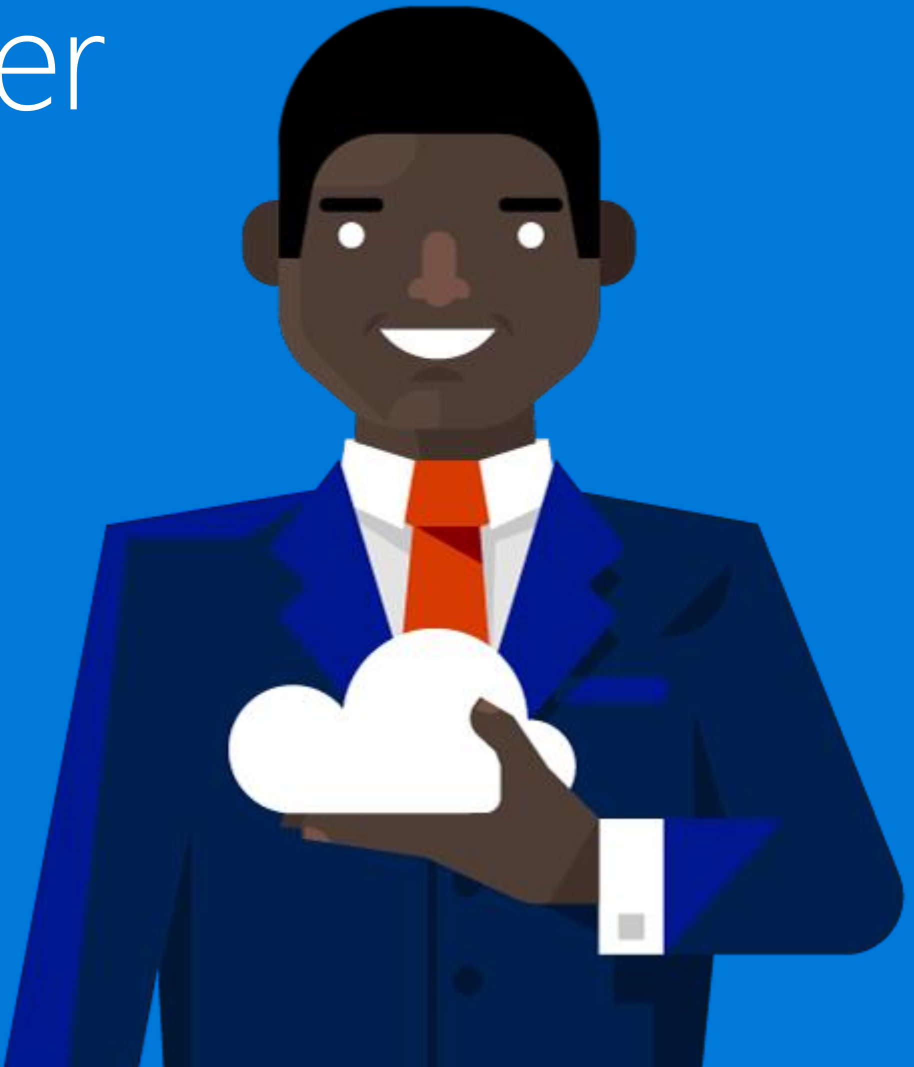
NODE HEALTH STATE	NUMBER OF NODES
Error	0
OK	5
Unknown	0

mfussellcluster12
Resource group

Essentials

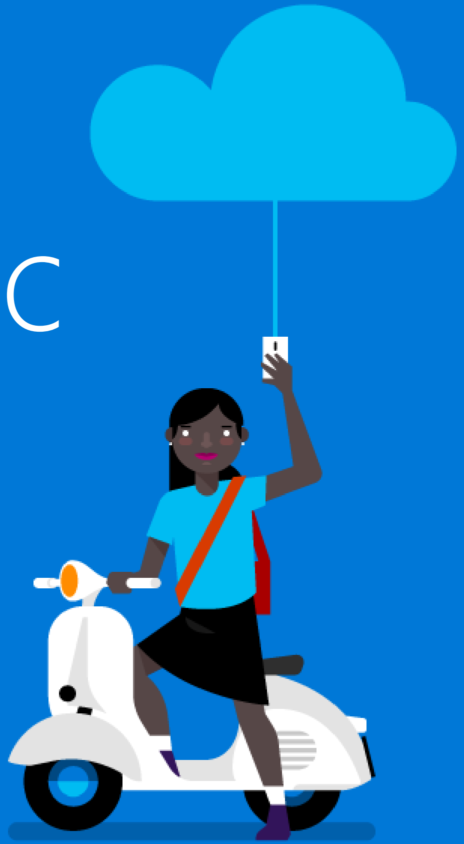
Subscription name WindowsFabric Subscription 1	Subscription ID 35349201-a0b3-405e-8a23-9f1450984307
Last deployment 8/31/2015 (Succeeded)	Location Southeast Asia

- AS-mfussellcluster12
- VM0
- VM1
- VM2
- VM3
- VM4
- LB-mfussellcluster12
- VM-NIC-0
- VM-NIC-1



Microsoft

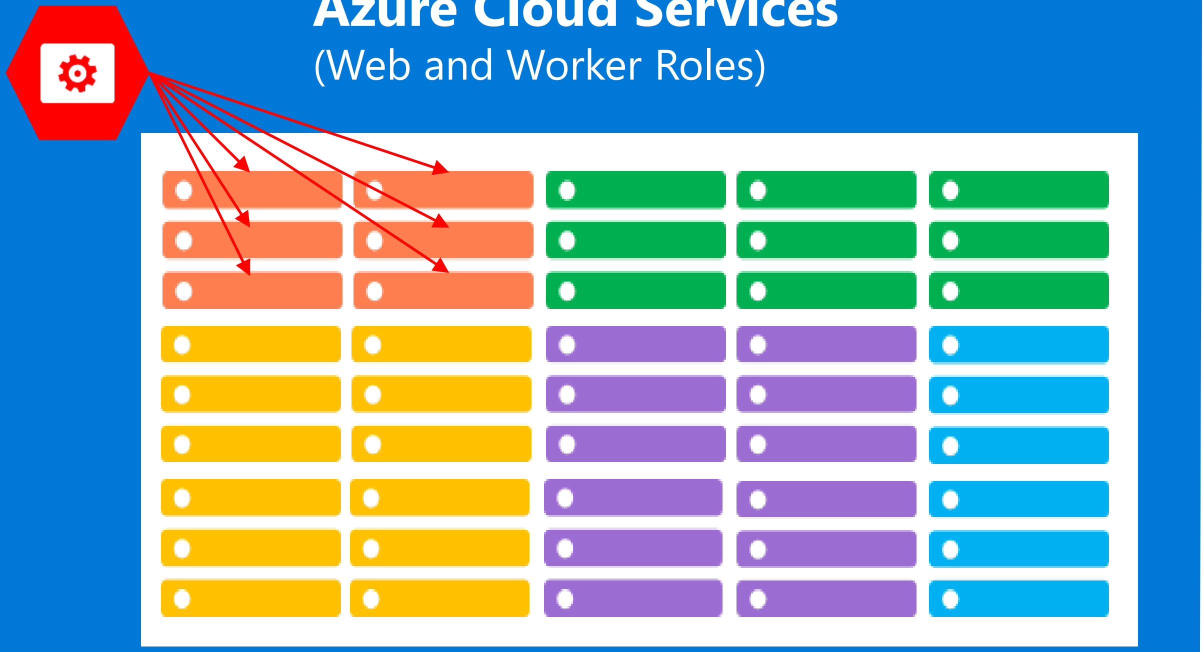
First microservice in Service Fabric



Comparing Azure Cloud Services vs. Azure Service Fabric

Azure Cloud Services

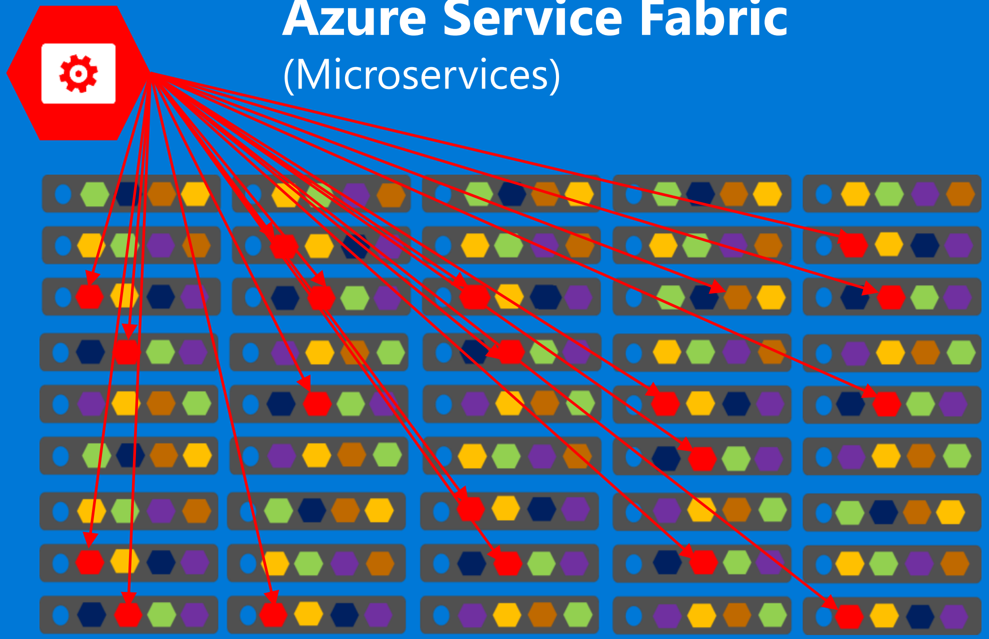
(Web and Worker Roles)



- 1 service instance per VM with uneven workloads
- Lower compute density
- Slow in deployment & upgrades
- Slower in scaling and disaster recovery

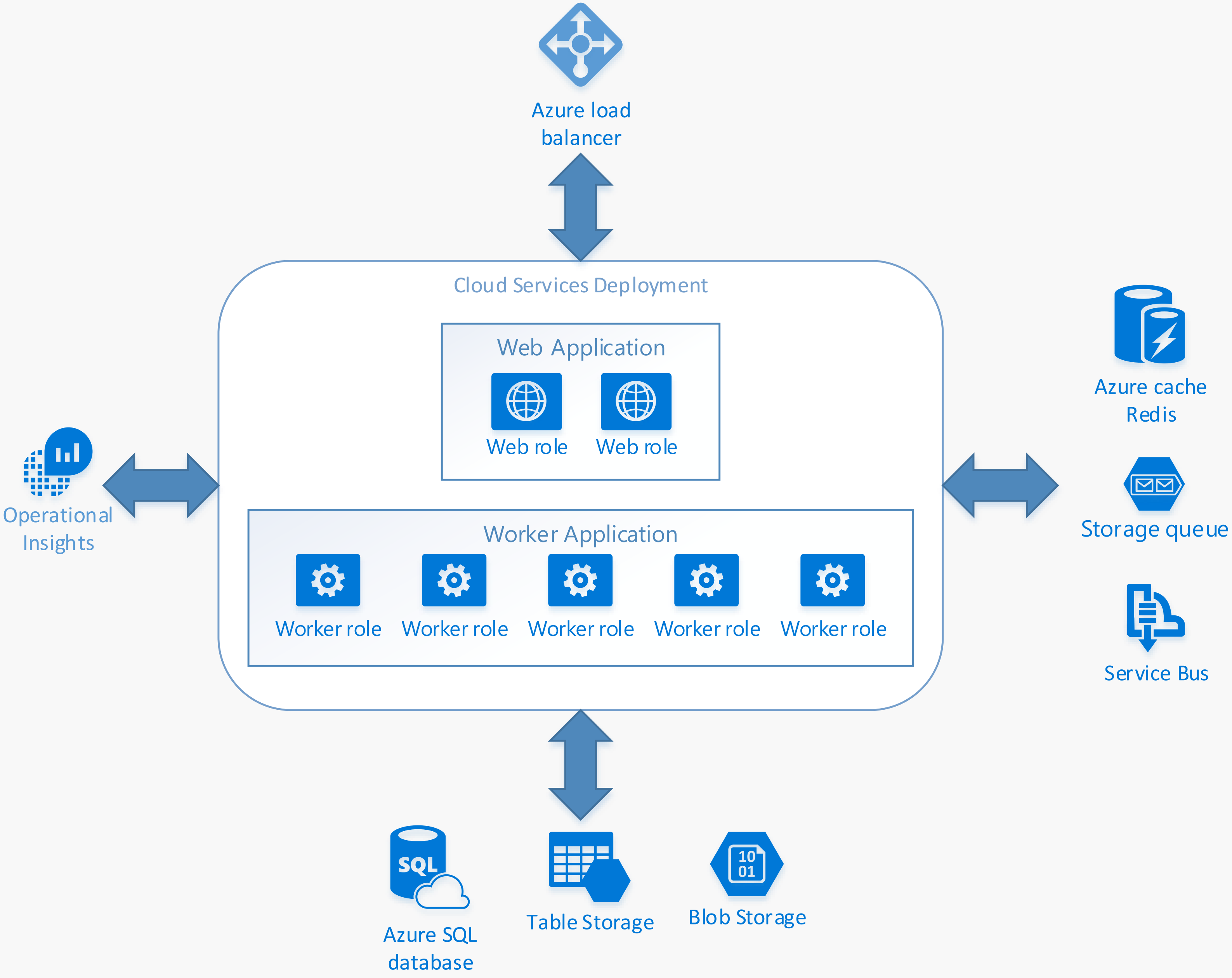
Azure Service Fabric

(Microservices)

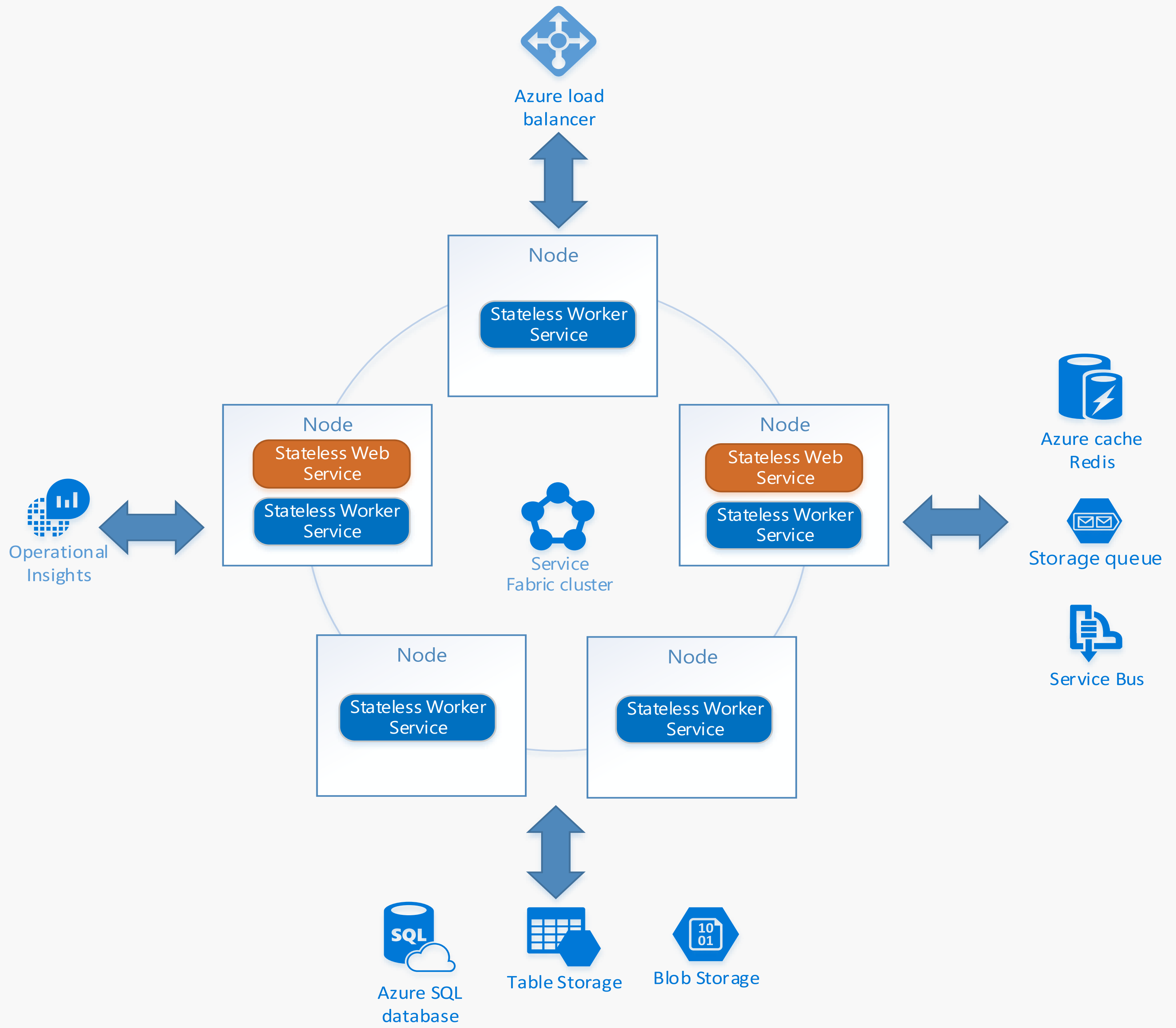


- Many microservices per VM
- High microservices density
- Fast deployment & upgrades
- Fast scaling microservices across the cluster

Cloud Service Application Design



Service Fabric Application Design

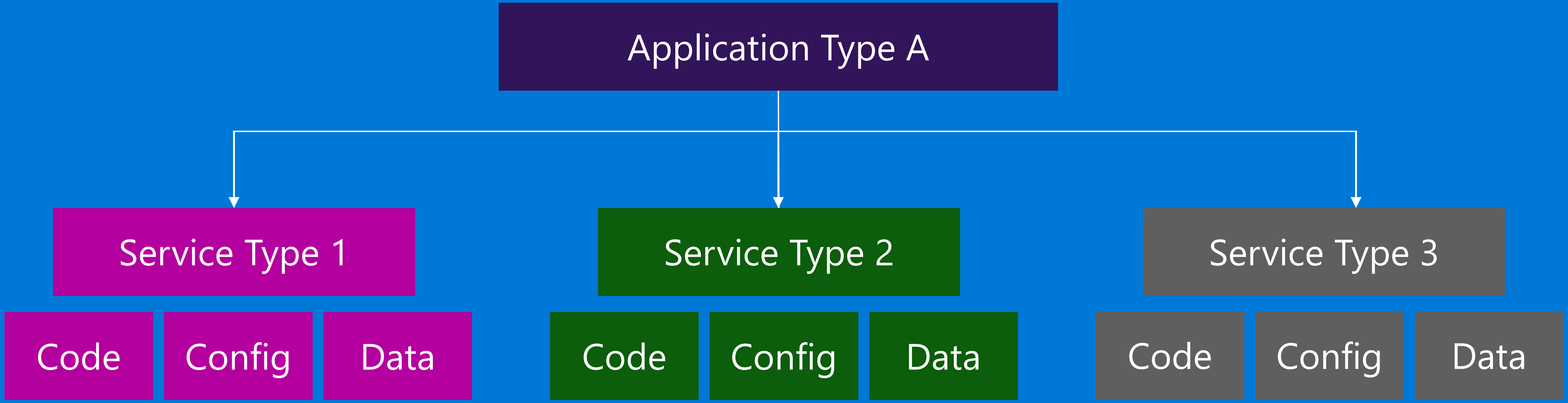


Types of microservices
from a Service Fabric perspective

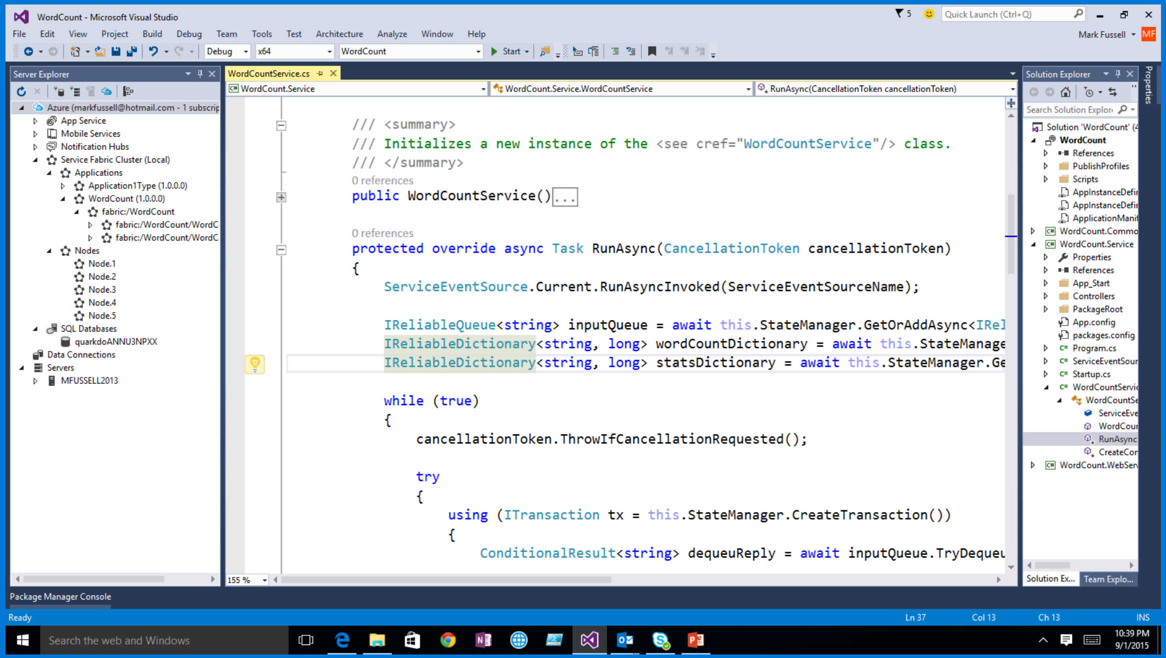
- Stateless microservice
  - Has either no state or it can be retrieved from an external store
  - There can be N instances
  - e.g. web frontends, protocol gateways, Azure Cloud Services etc.
- Stateful microservice
  - Maintain hard, authoritative state
  - N consistent copies achieved through replication and local persistence
  - e.g. database, documents, workflow, user profile, shopping cart etc.

Application type

- Declarative template for creating an application
- Based on a set of service types
- Used for packaging, deployment, and versioning



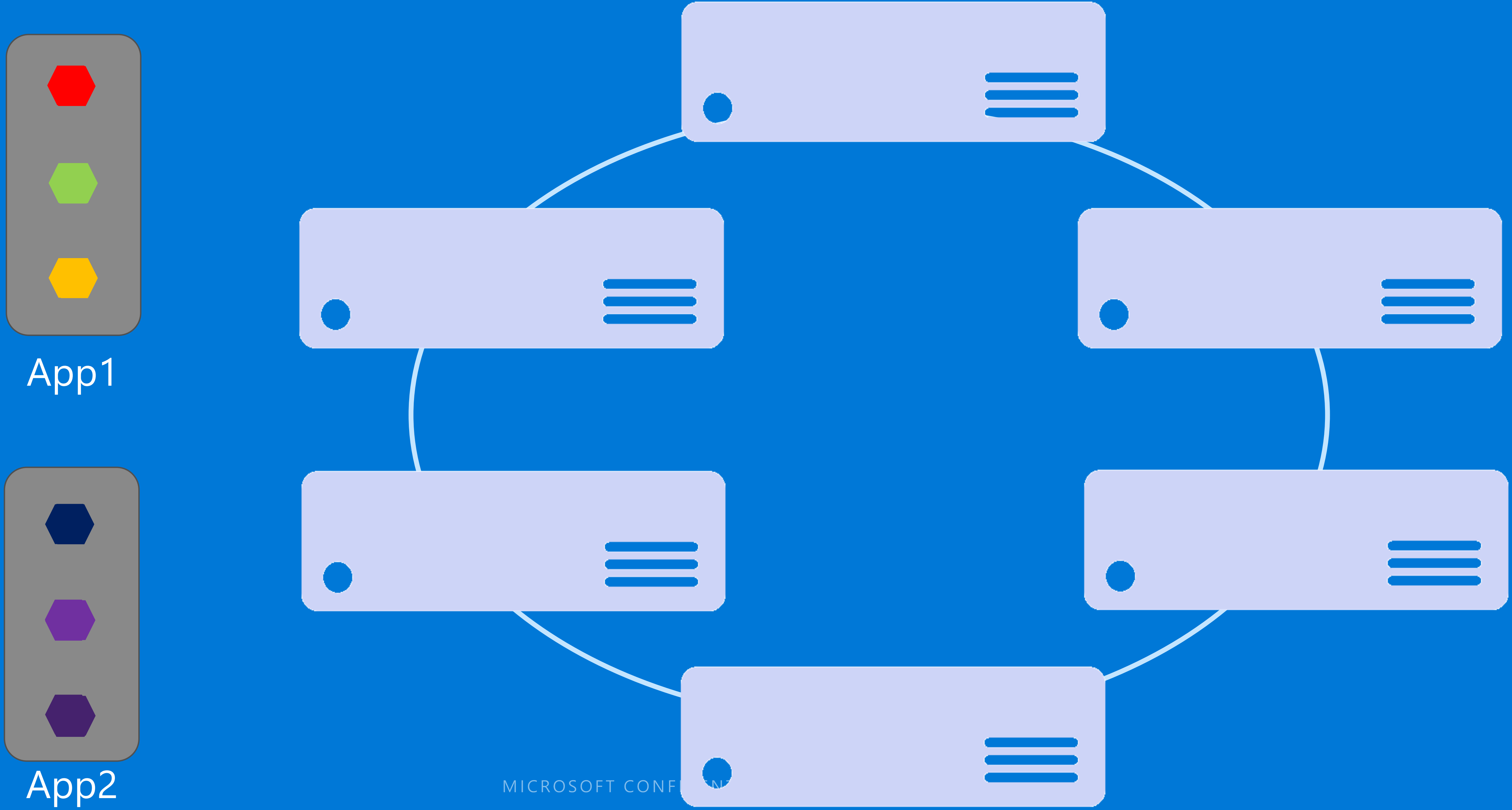
Demo: Building your first microservice with Service Fabric



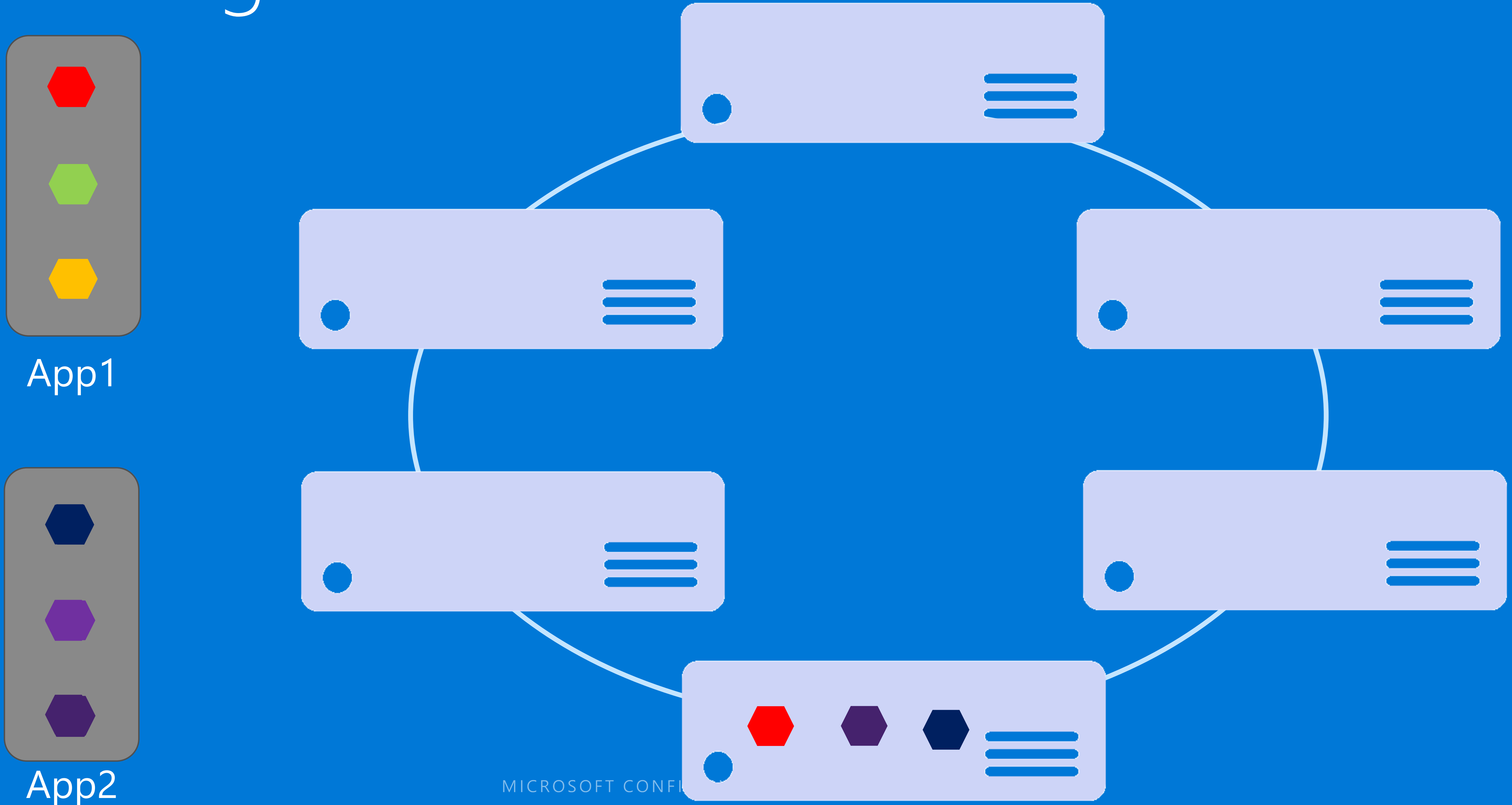
```
/// <summary>
/// Initializes a new instance of the <see cref="WordCountService"/> class.
/// </summary>
0 references
public WordCountService()...

0 references
protected override async Task RunAsync(CancellationToken cancellationToken)
{
    ServiceEventSource.Current.RunAsyncInvoked(ServiceEventSourceName);

    IReliableQueue<string> inputQueue = await this.StateManager.GetOrAddAsync<IRel
    IReliableDictionary<string, long> wordCountDictionary = await this.StateManage
    IReliableDictionary<string, long> statsDictionary = await this.StateManager.Ge

    while (true)
    {
        cancellationToken.ThrowIfCancellationRequested();

        try
        {
            using (ITransaction tx = this.StateManager.CreateTransaction())
            {
                ConditionalResult<string> dequeReply = await inputQueue.TryDequeu
```

Microsoft

Service Fabric cluster with microservices

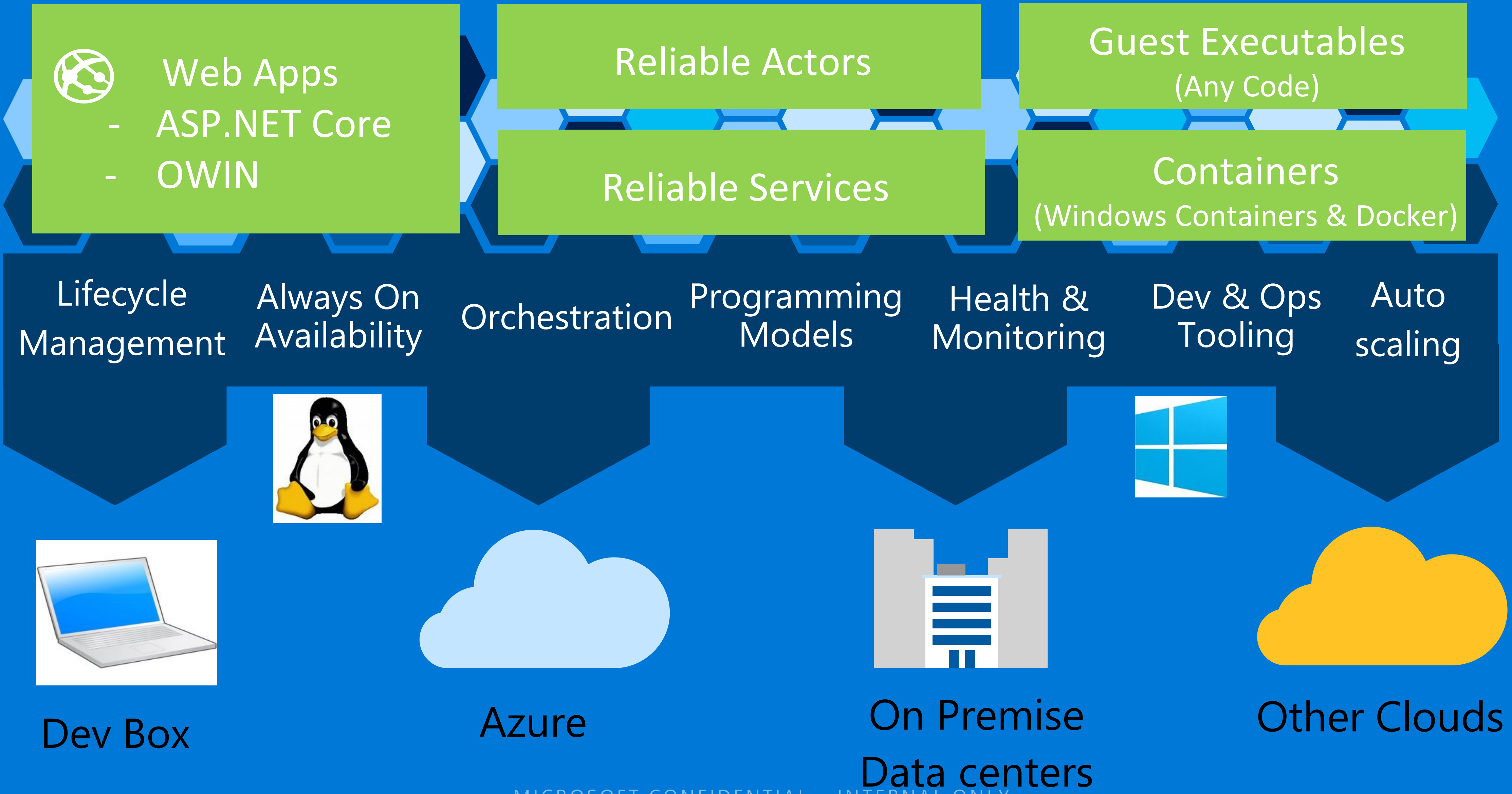


Handling machine failures



Building microservice applications with Service Fabric

Azure Service Fabric microservices



Web Apps
- ASP.NET Core
- OWIN

Reliable Actors

Reliable Services

Guest Executables (Any Code)

Containers (Windows Containers & Docker)

Lifecycle Management

Always On Availability

Orchestration

Programming Models

Health & Monitoring

Dev & Ops Tooling

Auto scaling

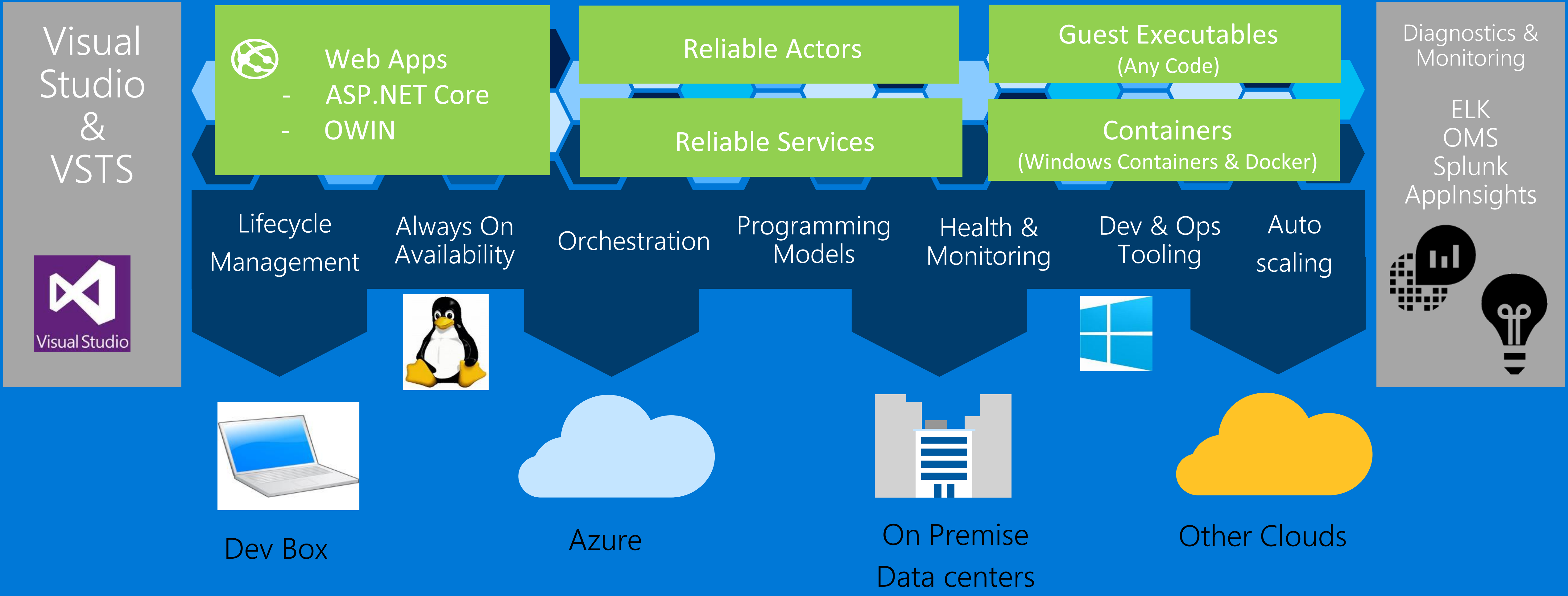
Dev Box

Azure

On Premise Data centers

Other Clouds

Service Fabric Programming Models & CI/CD



Visual Studio & VSTS

Visual Studio

Web Apps
- ASP.NET Core
- OWIN

Reliable Actors

Reliable Services

Guest Executables (Any Code)

Containers (Windows Containers & Docker)

Diagnostics & Monitoring

ELK
OMS
Splunk
AppInsights

Lifecycle Management

Always On Availability

Orchestration

Programming Models

Health & Monitoring

Dev & Ops Tooling

Auto scaling

Dev Box

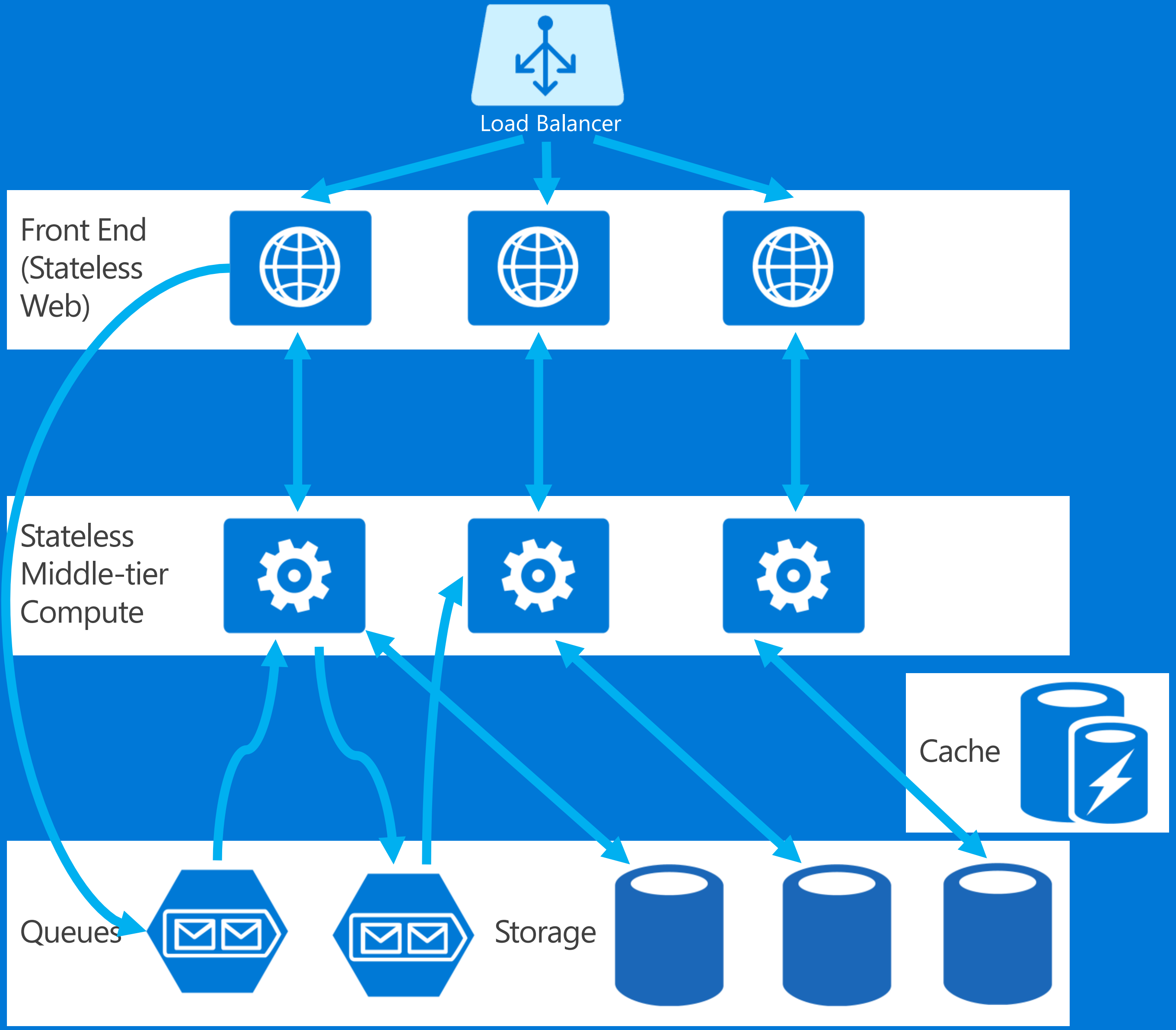
Azure

On Premise Data centers

Other Clouds

Stateless Services Pattern

- Scale stateless services backed by partitioned storage
- Increase reliability and ordering with queues
- Reduce read latency with caches
- Manage your own transactions for state consistency
- More moving parts each managed differently



Demo: stateless service

Microsoft

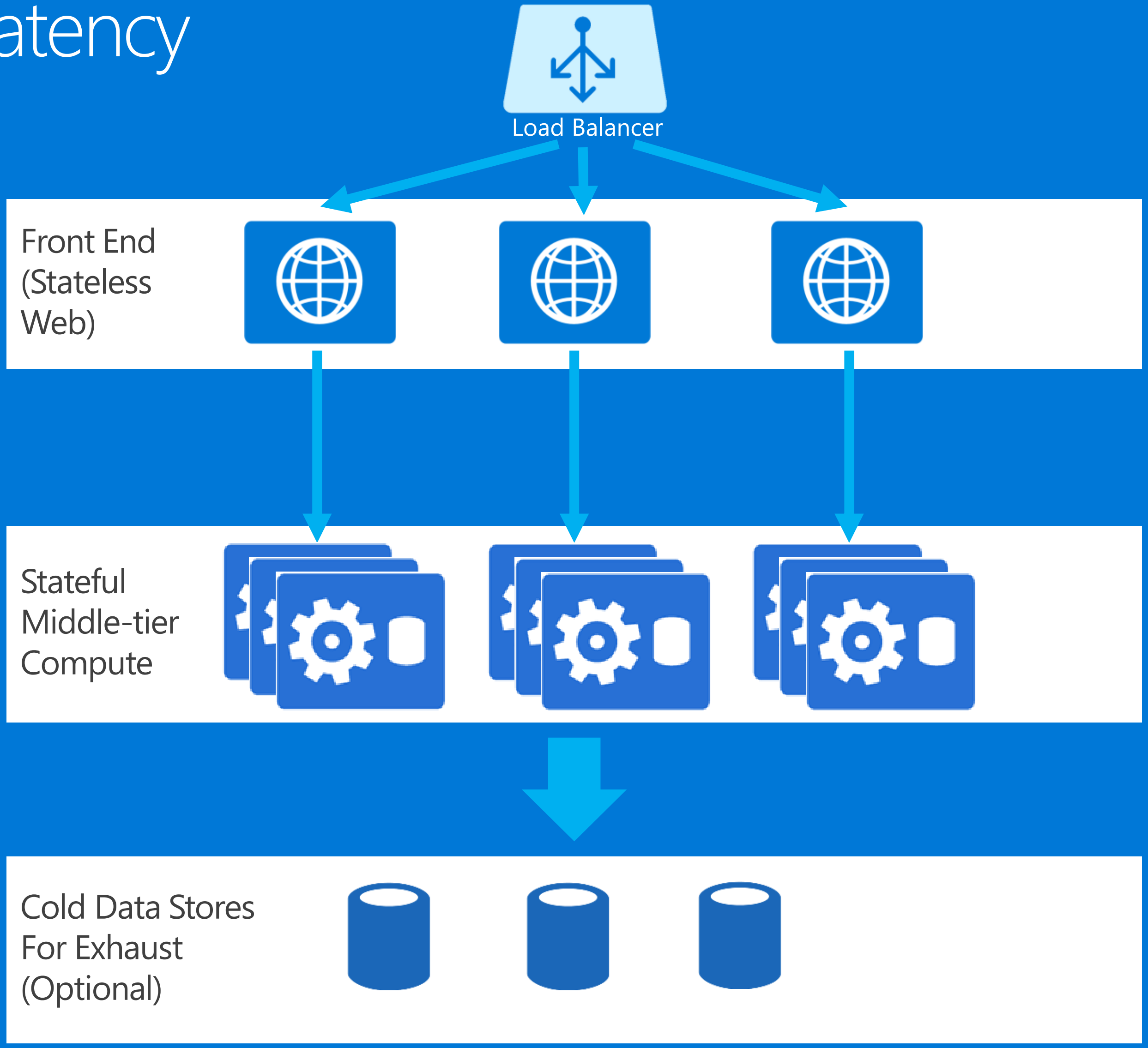
Demo: stateless web api

Microsoft

Stateful Services Pattern

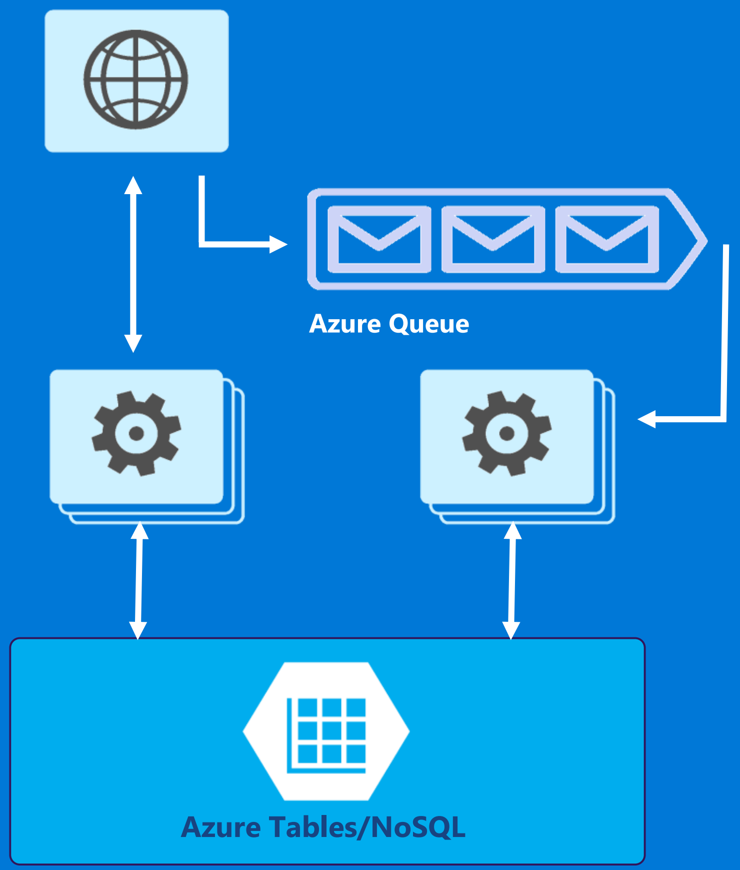
Simplify design, reduce latency

- Application state lives in the compute tier
- Low Latency reads and writes
- Partitions are first class at the service layer for scale-out
- Built in transactions
- Fewer moving parts
- External stores for exhaust and offline analytics

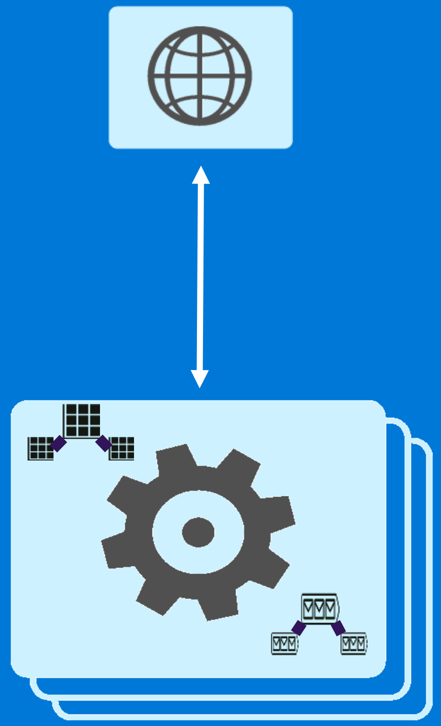

Load Balancer
Front End (Stateless Web)
Stateful Middle-tier Compute
Cold Data Stores For Exhaust (Optional)

Comparing Azure Cloud Services vs. Azure Service Fabric

Cloud Services OR Service Fabric Stateless Service

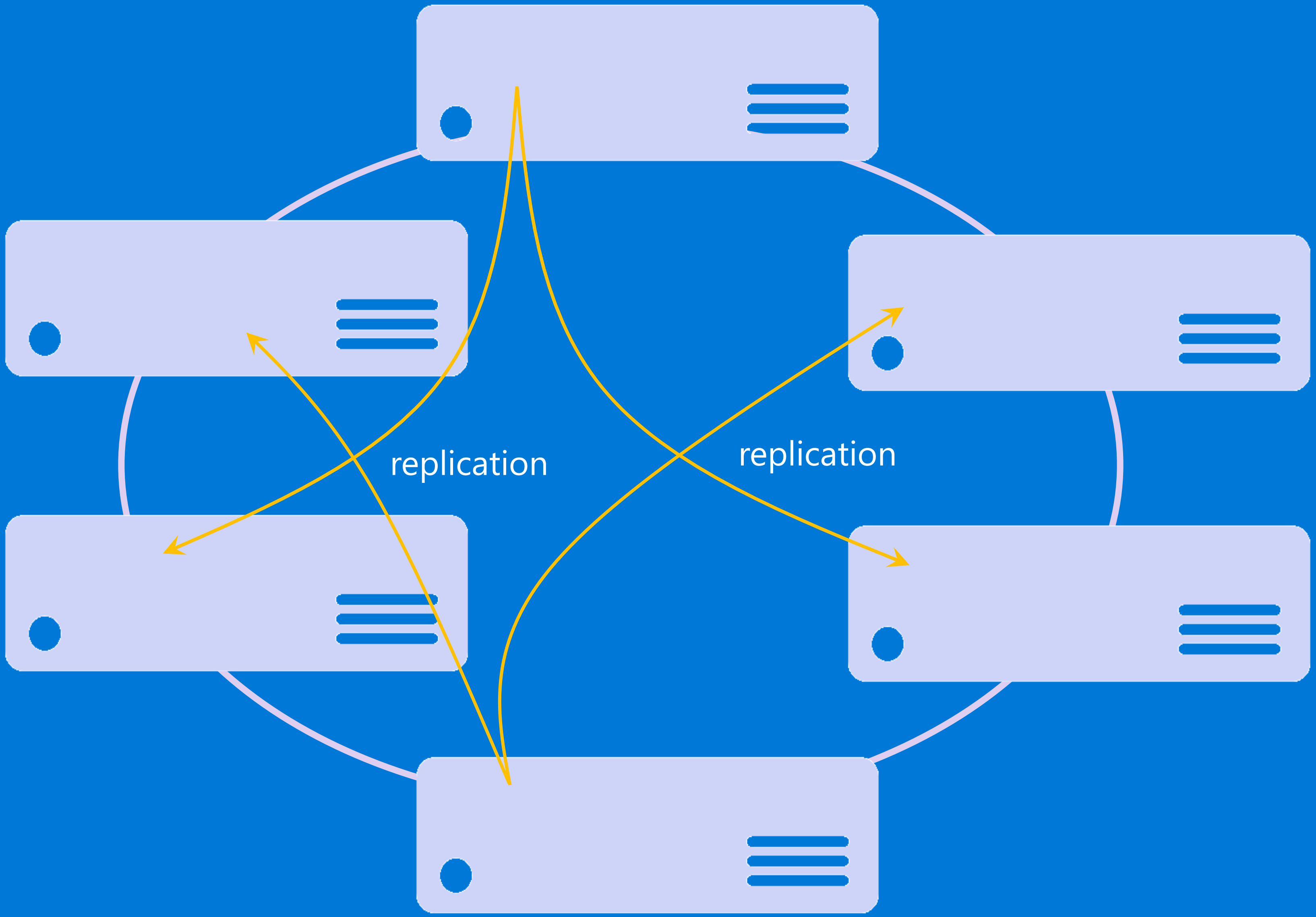


Service Fabric Stateful Service



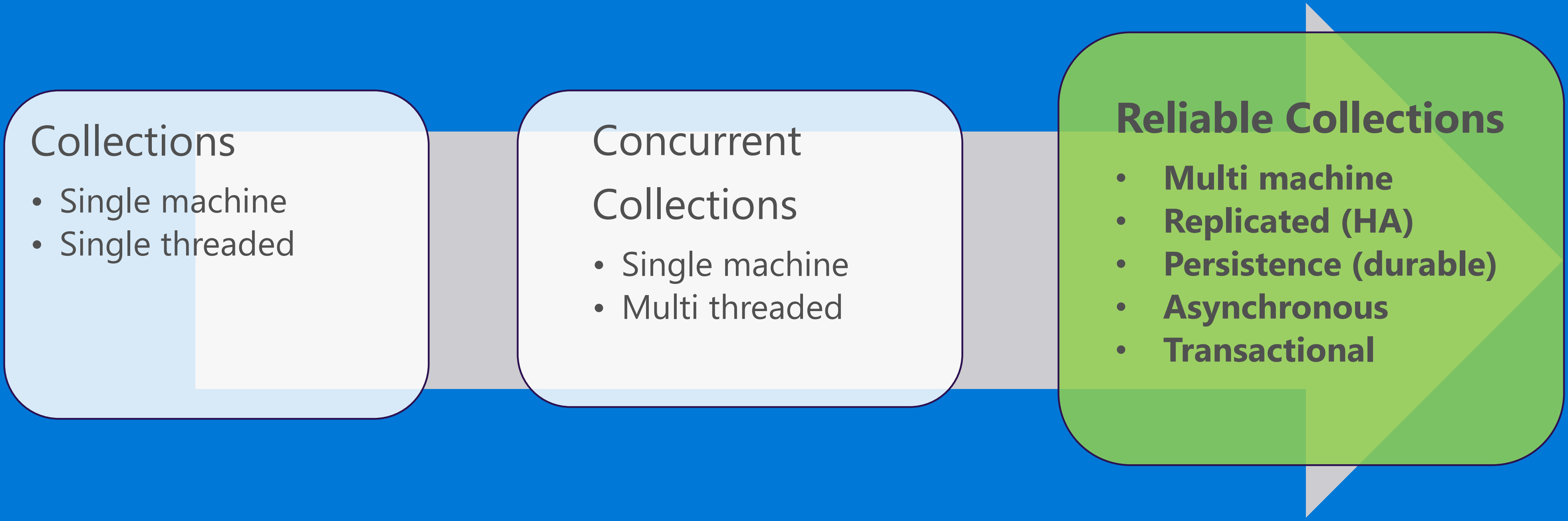
Stateful microservice

Application Package



Reliable Collections

- Reliable collections make it easy to build stateful services.
- An evolution of .NET collections for the cloud.



Collections

- Single machine
- Single threaded

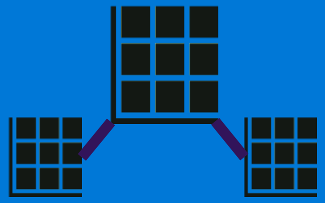
Concurrent Collections

- Single machine
- Multi threaded

Reliable Collections

- **Multi machine**
- **Replicated (HA)**
- **Persistence (durable)**
- **Asynchronous**
- **Transactional**

Reliable Collections

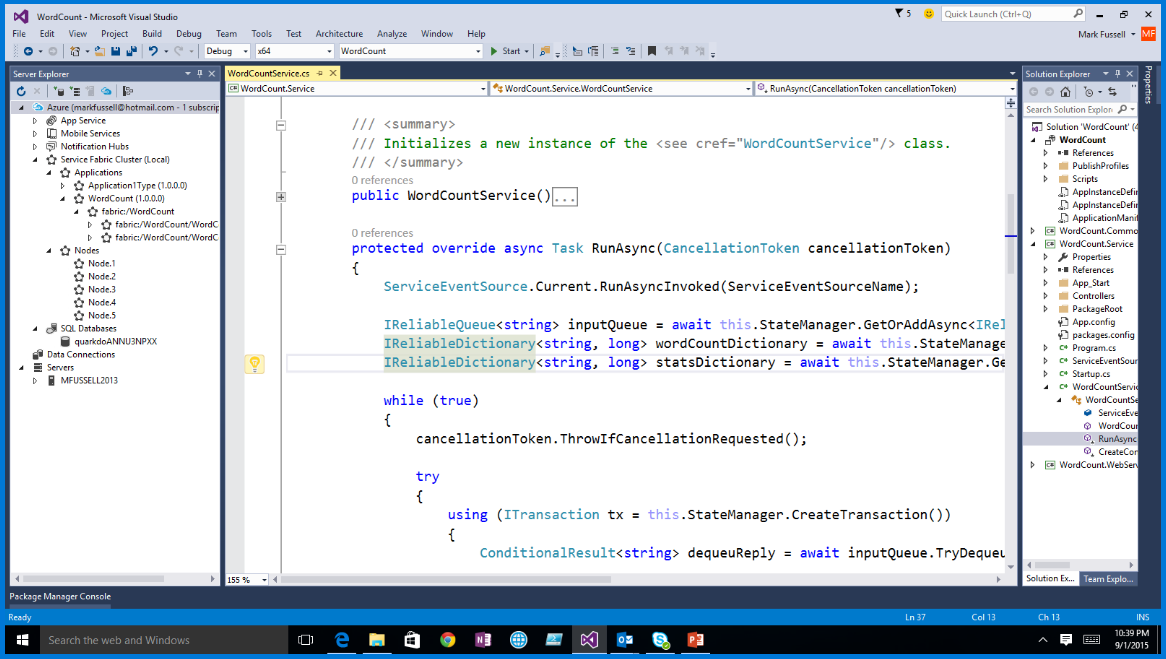


IReliableDictionary<K,V>

IReliableQueue<T>

- Data is replicated and durably stored on multiple replicas.
- Atomically update one or more collections using transactions.
- Supports LINQ.

Demo: stateful service

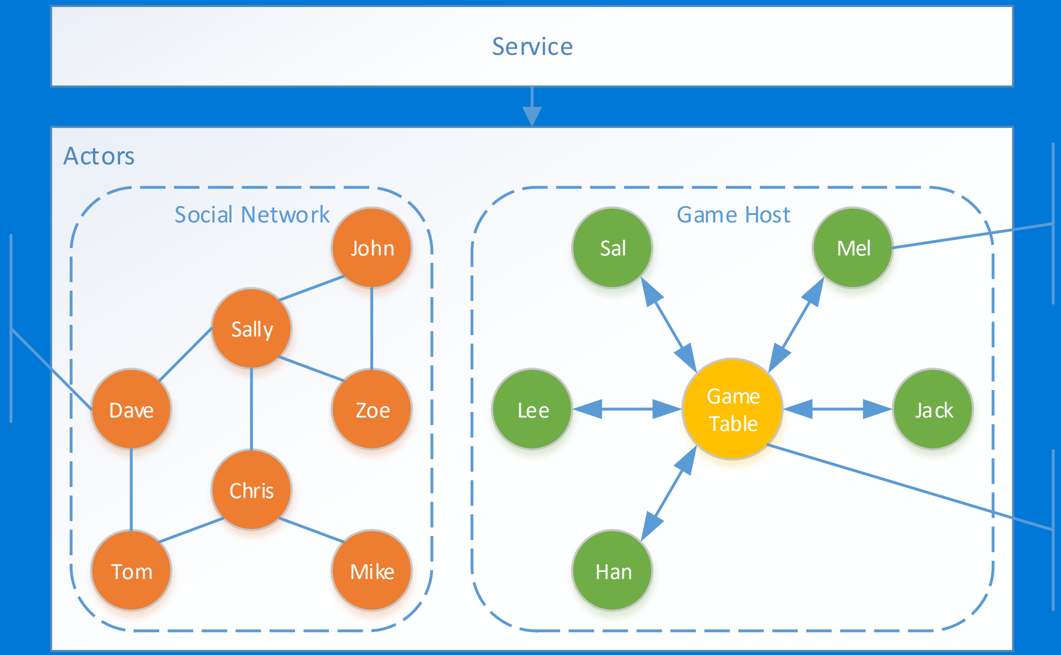


```
/// <summary>
/// Initializes a new instance of the <see cref="WordCountService"/> class.
/// </summary>
0 references
public WordCountService()...

0 references
protected override async Task RunAsync(CancellationToken cancellationToken)
{
    ServiceEventSource.Current.RunAsyncInvoked(ServiceEventSourceName);

    IReliableQueue<string> inputQueue = await this.StateManager.GetOrAddAsync<IRel
    IReliableDictionary<string, long> wordCountDictionary = await this.StateManage
    IReliableDictionary<string, long> statsDictionary = await this.StateManager.Ge

    while (true)
    {
        cancellationToken.ThrowIfCancellationRequested();

        try
        {
            using (ITransaction tx = this.StateManager.CreateTransaction())
            {
                ConditionalResult<string> dequeReply = await inputQueue.TryDequeu
```

Microsoft

Actor Pattern

- Isolated, independent unit of compute and state with single-threaded execution
- Computational model for concurrent or distributed systems
- An implementation of the actor design pattern



Social Person represents a person or an entity that is part of a social network. Its key behaviour includes updating their status and viewing their friends feed.

Player Actor allows players to join, play and leave games. It maintains player state and players in game state.

Game Actor represents the game. It maintains the game state as well as players in game and their state.

What is an Actor?

- An independent unit of compute and state with large number of them executing in parallel
- Communicates with other actors using asynchronous messaging
- Has single threaded execution (turn based concurrency)

Create a Reliable Actor

- Define Actor Interface

```
public interface IHelloWorld : IActor {
... }
```

Implement Actor Interface

```
public class HelloWorldActor : Actor, IHelloWorld {
... }
```

Register Actor Implementation with Runtime

```
using (var fabricRuntime = FabricRuntime.Create())
{
    fabricRuntime.RegisterActor(typeof(HelloWorldActor));
```

Demo: actor model

Microsoft

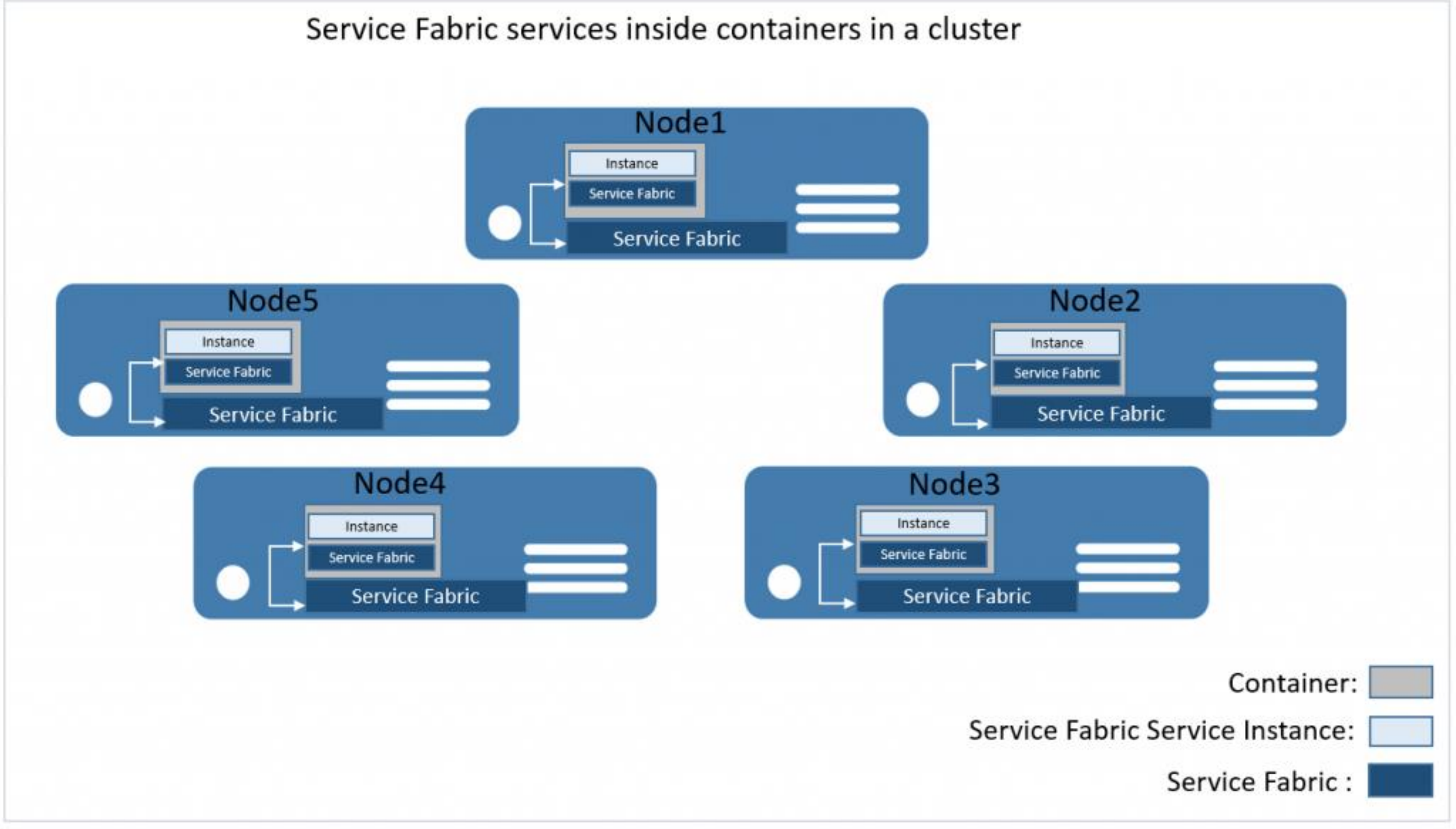
Guest Pattern

Any code, any container

- Run any type of application, such as Node.js, Java, or native applications in Azure Service Fabric

- Guest executables are treated by Service Fabric like stateless services

- Supports deployment of Docker containers on Linux and Windows Server containers on Windows Server 2016



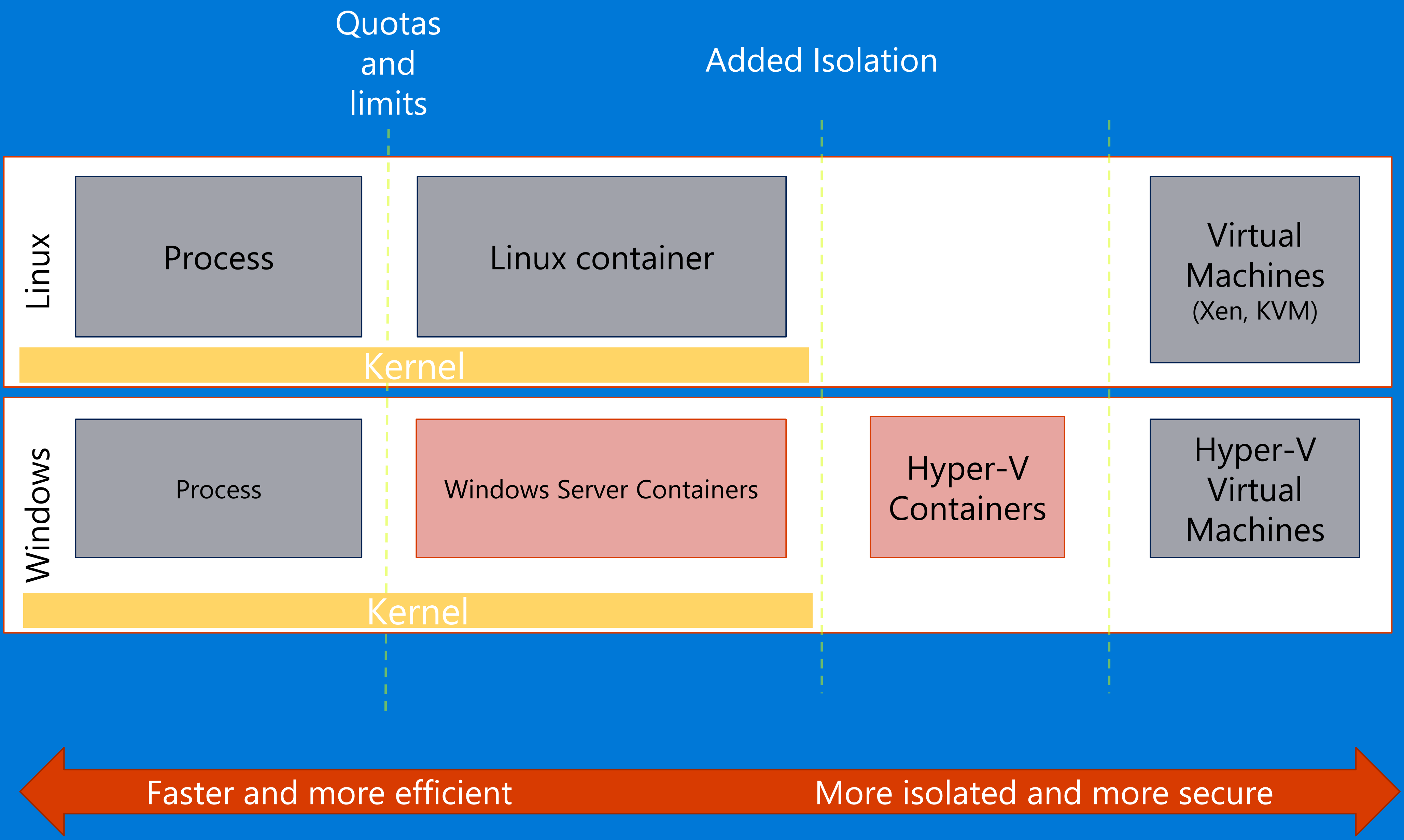
Demo: guest executables

Microsoft

Service Fabric is a Container Orchestrator

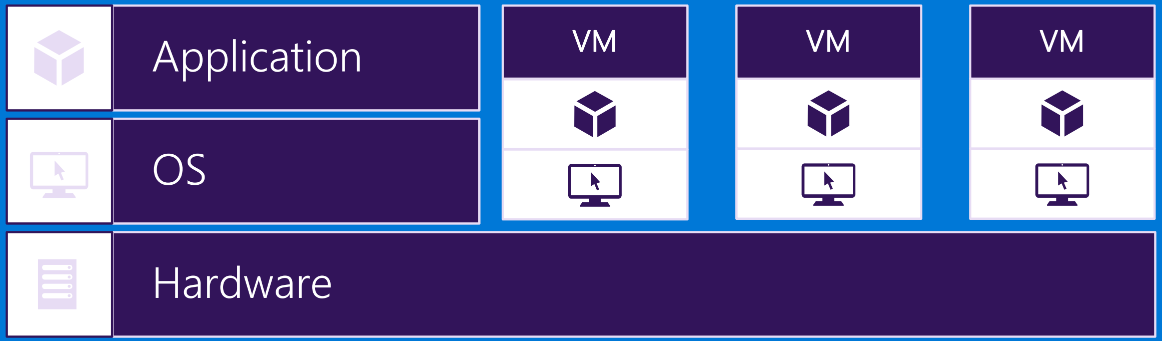
- Service Fabric performs the placement and failover of containers in the cluster
- Deploy Docker containers on Linux
- Deploy Windows containers on Windows (which communicate via the Docker Agent)
- Containers can be deployed with
  - Code with guest executables or Service Fabric programming models
  - Endpoints in a container can be registered with Service Fabric for discoverability

Containers


Quotas and limits
Added Isolation
Linux
Process
Linux container
Virtual Machines (Xen, KVM)
Kernel
Windows
Process
Windows Server Containers
Hyper-V Containers
Hyper-V Virtual Machines
Kernel
Faster and more efficient
More isolated and more secure

Containers on Windows

Traditional virtual machines = hardware virtualization



Application

OS

Hardware

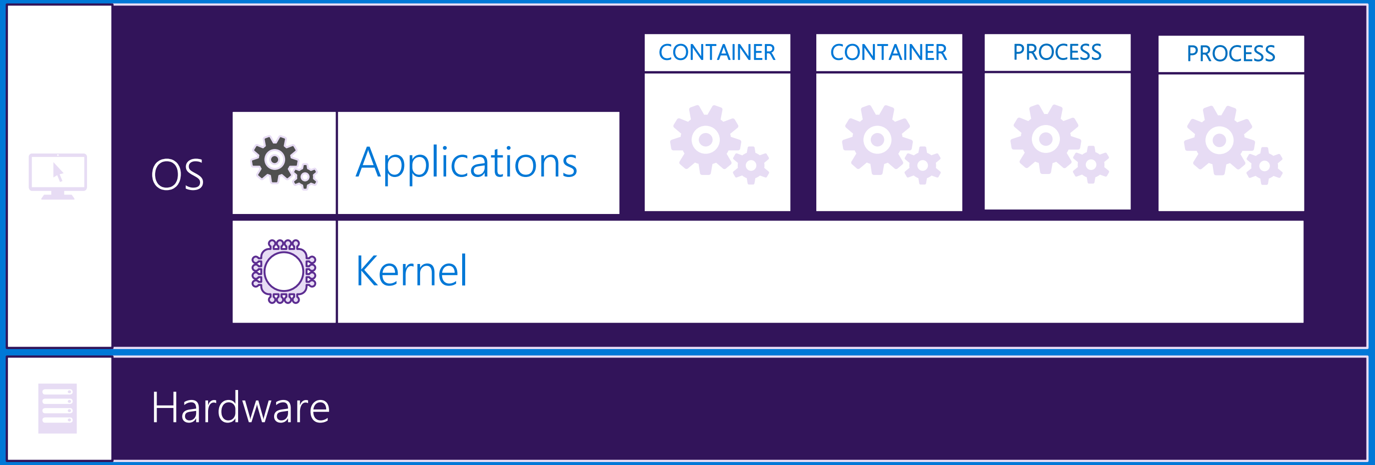
VM

VM

VM

Processes = an executing program

Containers = Operating system virtualization



OS

Applications

Kernel

CONTAINER

CONTAINER

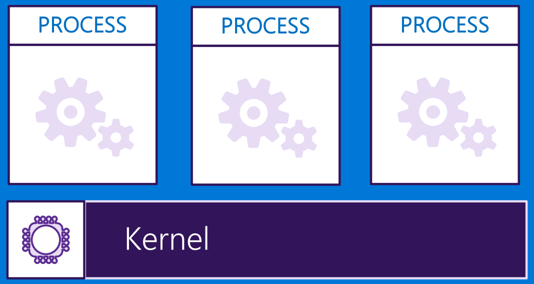
PROCESS

PROCESS

Hardware

Processes

Maximum speed and density



PROCESS

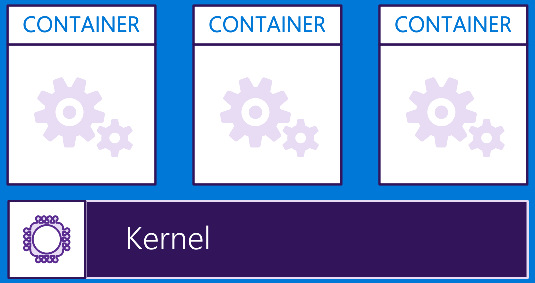
PROCESS

PROCESS

Kernel

Windows Server Containers

Speed and density



CONTAINER

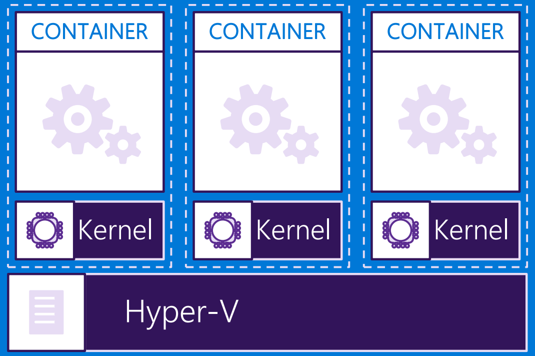
CONTAINER

CONTAINER

Kernel

Hyper-V Containers

Isolation plus performance



CONTAINER

CONTAINER

CONTAINER

Kernel

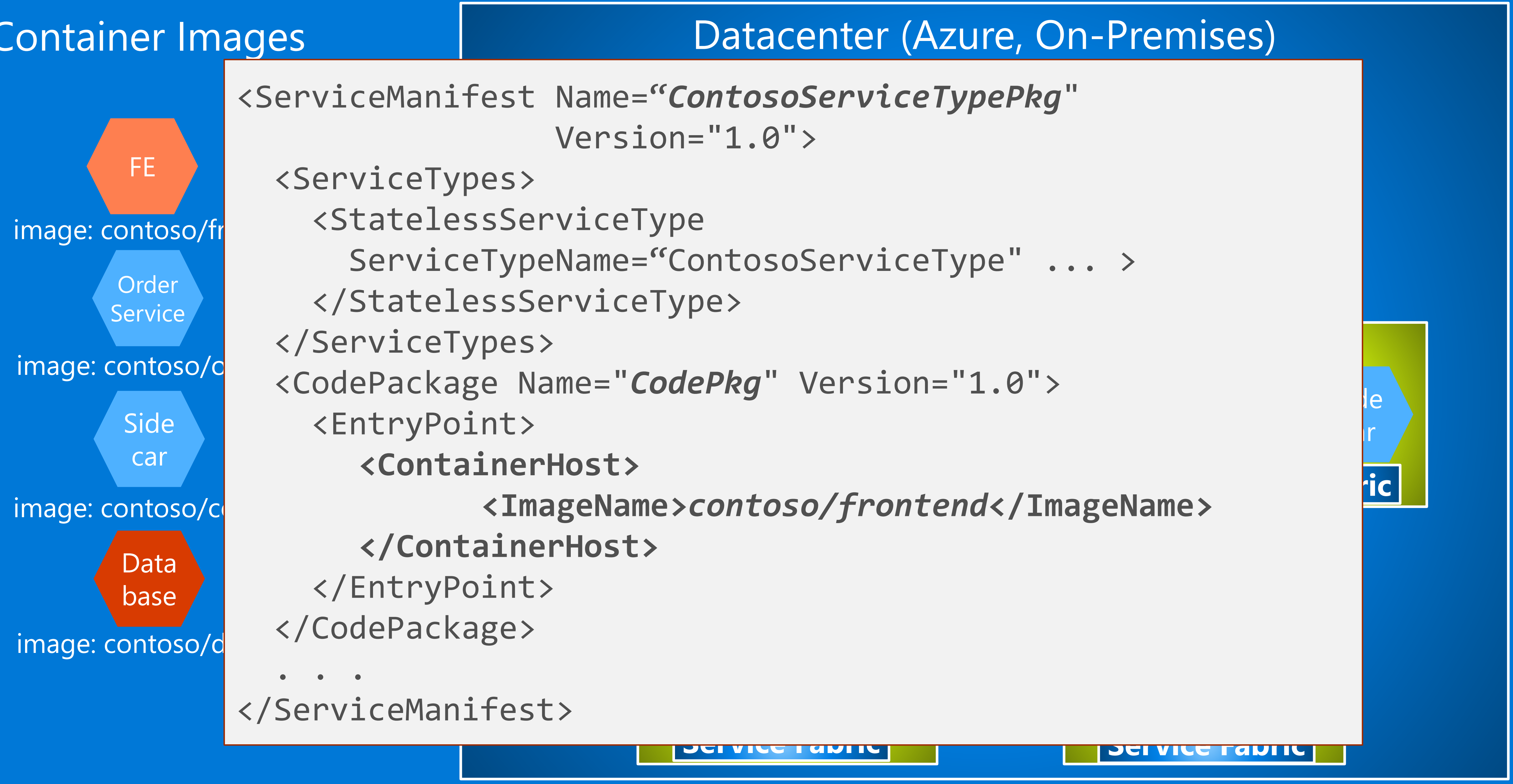
Kernel

Kernel

Hyper-V

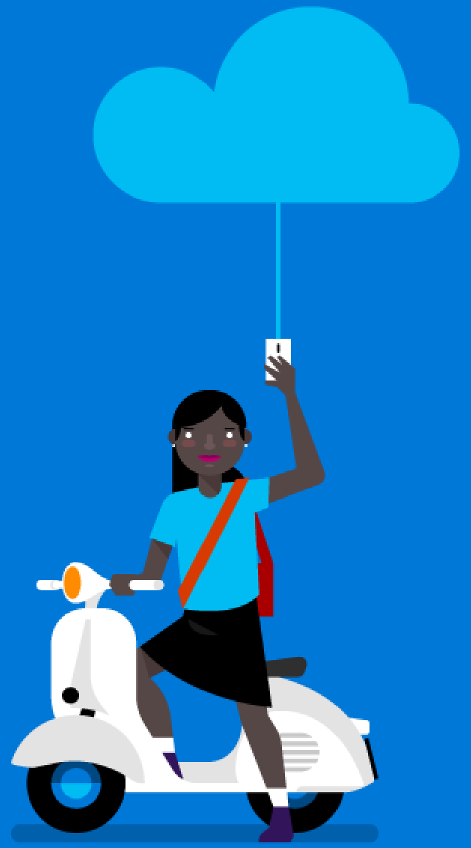
Service Fabric Container Integration - Guest Container

Container Images



```
<ServiceManifest Name=“ContosoServiceTypePkg"
                 Version="1.0">
  <ServiceTypes>
    <StatelessServiceType
      ServiceTypeName=“ContosoServiceType" ... >
    </StatelessServiceType>
  </ServiceTypes>
  <CodePackage Name="CodePkg" Version="1.0">
    <EntryPoint>
      <ContainerHost>
             <ImageName>contoso/frontend</ImageName>
      </ContainerHost>
    </EntryPoint>
  </CodePackage>
  . . .
</ServiceManifest>
```

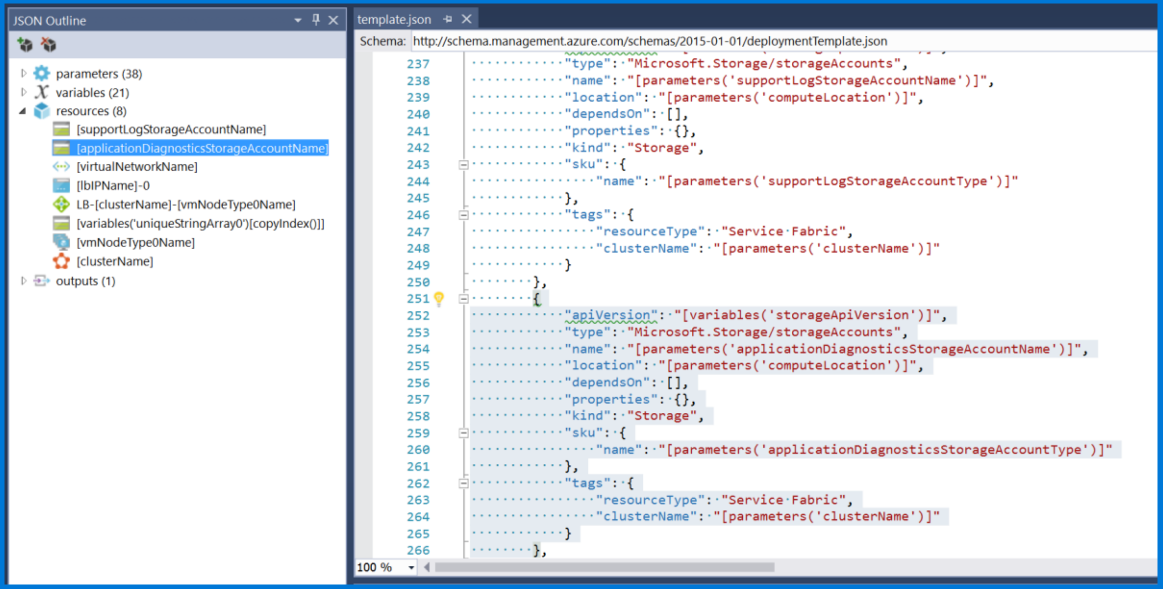
Managing Service Fabric and microservices



Microsoft

Manage the cluster

- Define customer cluster and generate template (Azure portal)

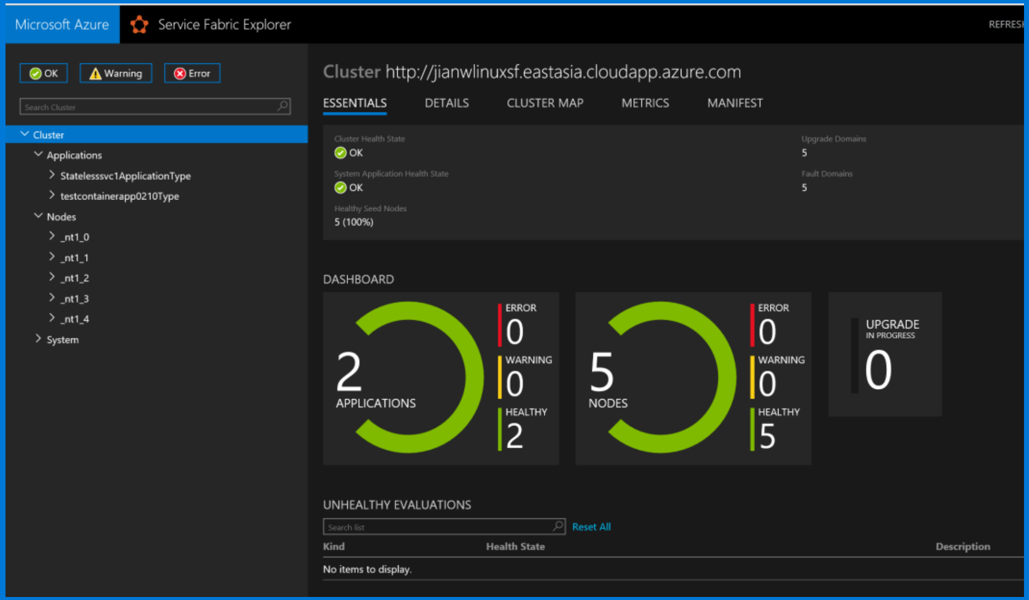


- parameters (38)
- variables (21)
- resources (8)
  - [supportLogStorageAccountName]
  - [applicationDiagnosticsStorageAccountName]
  - [virtualNetworkName]
  - [lbIPName]-0
  - LB-[clusterName]-[vmNodeType0Name]
  - [variables('uniqueStringArray0')[copyIndex()]]
  - [vmNodeType0Name]
  - [clusterName]
- outputs (1)

Schema: http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json

```
            "type": "Microsoft.Storage/storageAccounts",
            "name": "[parameters('supportLogStorageAccountName')]",
            "location": "[parameters('computeLocation')]",
            "dependsOn": [],
            "properties": {},
            "kind": "Storage",
            "sku": {
                "name": "[parameters('supportLogStorageAccountType')]"
            },
            "tags": {
                "resourceType": "Service Fabric",
                "clusterName": "[parameters('clusterName')]"
            }
        },
        {
            "apiVersion": "[variables('storageApiVersion')]",
            "type": "Microsoft.Storage/storageAccounts",
            "name": "[parameters('applicationDiagnosticsStorageAccountName')]",
            "location": "[parameters('computeLocation')]",
            "dependsOn": [],
            "properties": {},
            "kind": "Storage",
            "sku": {
                "name": "[parameters('applicationDiagnosticsStorageAccountType')]"
            },
            "tags": {
                "resourceType": "Service Fabric",
                "clusterName": "[parameters('clusterName')]"
            }
        },
```

- Manage cluster by explorer, Azure portal and command lines



Cluster http://jianwlinuxsf.eastasia.cloudapp.azure.com

Cluster Health State: OK

System Application Health State: OK

Healthy Seed Nodes: 5 (100%)

Upgrade Domains: 5

Fault Domains: 5

DASHBOARD

2 APPLICATIONS — ERROR 0, WARNING 0, HEALTHY 2

5 NODES — ERROR 0, WARNING 0, HEALTHY 5

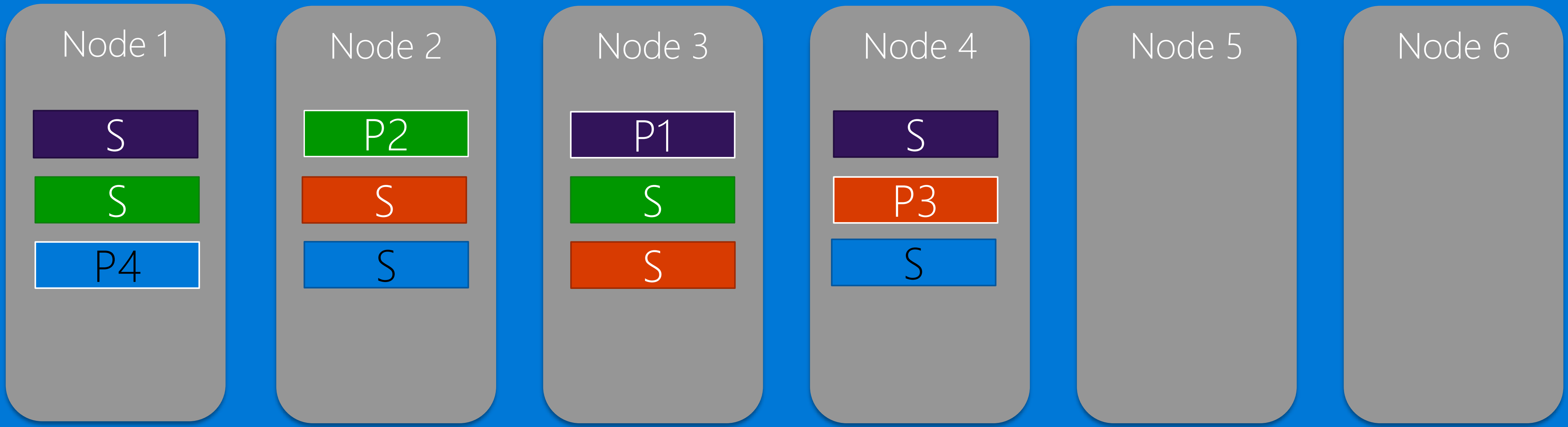
UPGRADE IN PROGRESS 0

UNHEALTHY EVALUATIONS

Kind	Health State	Description
No items to display.		

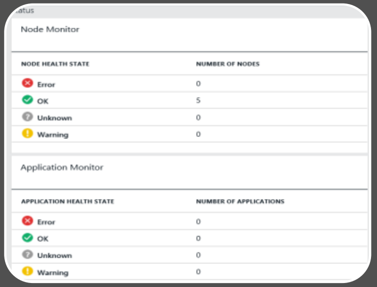
Service Fabric Orchestration Scale-Out

- Services can be partitioned for scale-out.
- You can choose your own partitioning scheme.
- Service partitions are striped across machines in the cluster.
- Replicas automatically scale out & in on cluster changes



Monitoring your Services

Visibility into how your services are doing when running in production



atus

Node Monitor

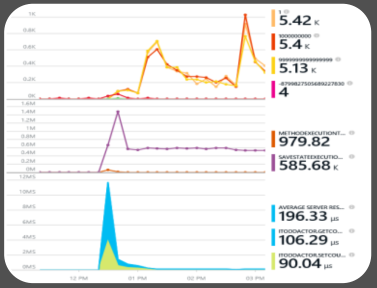
NODE HEALTH STATE	NUMBER OF NODES
✖ Error	0
✔ OK	5
? Unknown	0
! Warning	0

Application Monitor

APPLICATION HEALTH STATE	NUMBER OF APPLICATIONS
✖ Error	0
✔ OK	0
? Unknown	0
! Warning	0

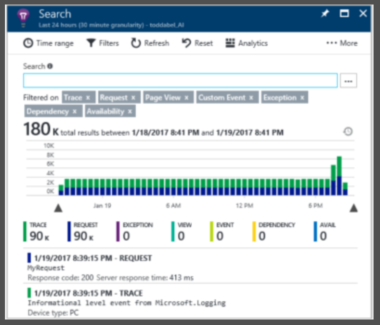
Health status monitoring

- Built-in health status for cluster and services
- Flexible and extensible health store for custom app health reporting
- Allows continuous monitoring for real-time alerting on problems in production



Performance and stress response

- Rich built-in system events for Actors and Services programming models
- Easy to add custom application performance metrics
- System events can be viewed in OMS



Search
Last 24 hours (30 minute granularity) - toddabel_AI

Search

Filtered on Trace × Request × Page View × Custom Event × Exception × Dependency × Availability ×

180 K total results between 1/18/2017 8:41 PM and 1/19/2017 8:41 PM

TRACE 90 K | REQUEST 90 K | EXCEPTION 0 | VIEW 0 | EVENT 0 | DEPENDENCY 0 | AVAIL 0

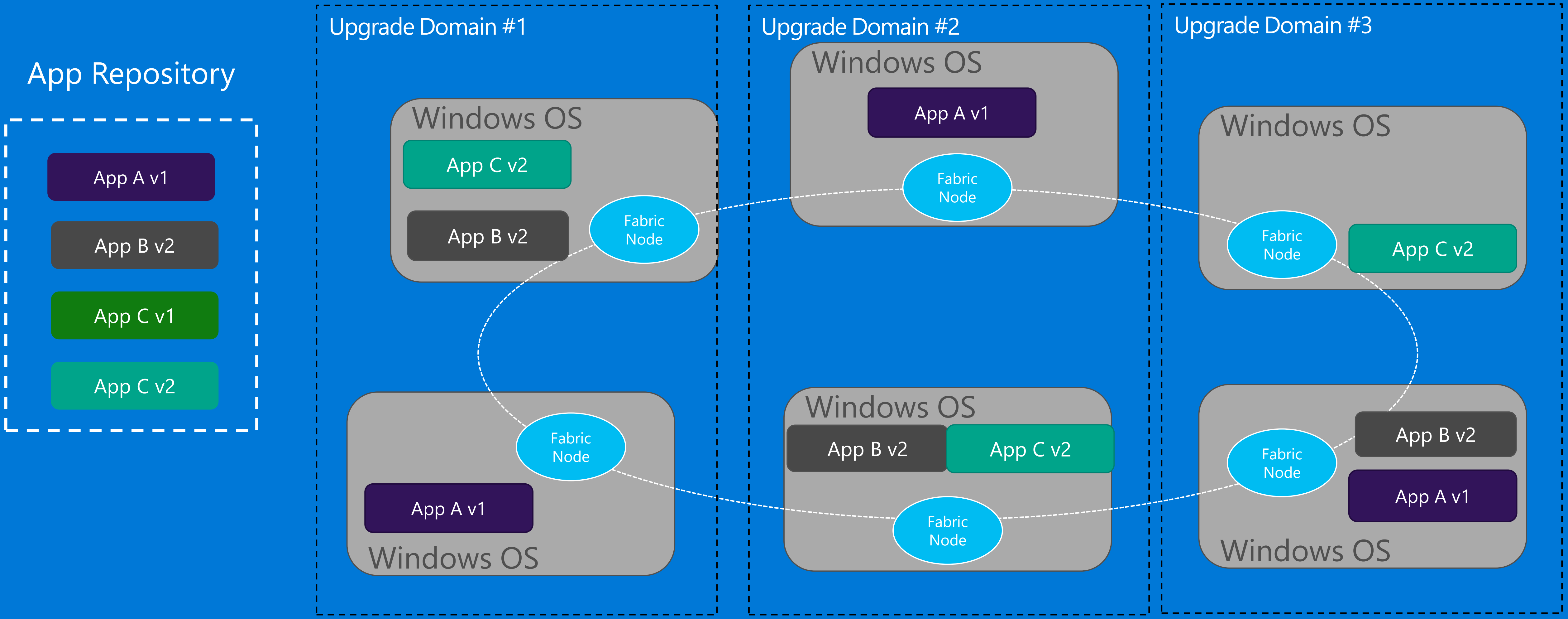
1/19/2017 8:39:15 PM - REQUEST
MyRequest
Response code: 200 Server response time: 413 ms

1/19/2017 8:39:15 PM - TRACE
Informational level event from Microsoft.Logging
Device type: PC

Write application diagnostic event anywhere

- Application events are captured with EventFlow
- Can be pushed to any store for query e.g. AppInsights, ELK, Serilog

Application Upgrade



Service Fabric Cases and Samples

Microsoft

Services Powered by Service Fabric

SQL

SQL Database

2.0 million DBs

{ }

Document DB

Billions transactions/day

IoT Hub

10 of Ks devices &
millions of messages

Event Hubs

20bn events/day

Skype
for Business

Skype

Cortana

Windows Intune

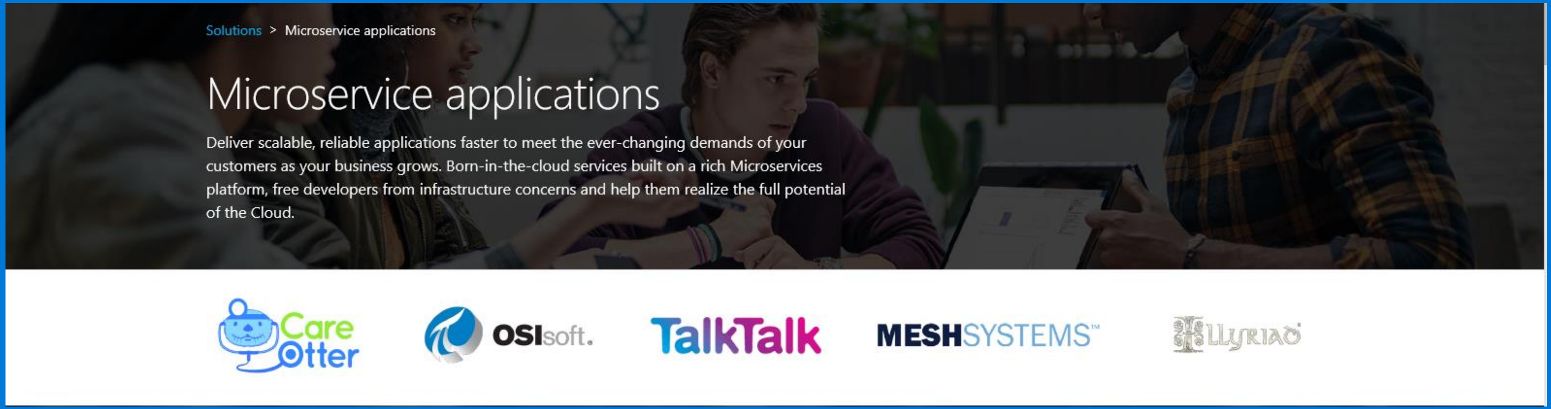
Intune

Microsoft
Dynamics 365

Dynamics

Power BI

Case studies



Solutions > Microservice applications

Microservice applications

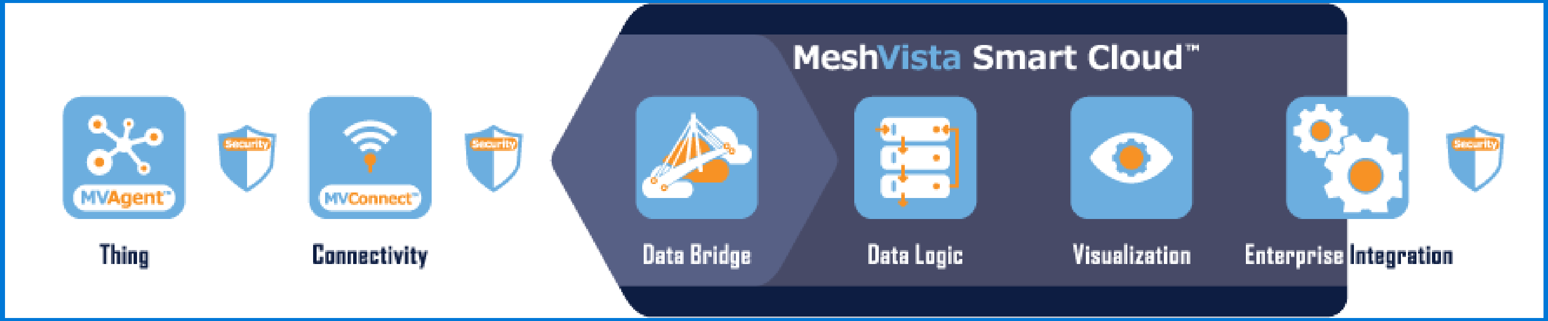
Deliver scalable, reliable applications faster to meet the ever-changing demands of your customers as your business grows. Born-in-the-cloud services built on a rich Microservices platform, free developers from infrastructure concerns and help them realize the full potential of the Cloud.

https://azure.microsoft.com/en-us/solutions/microservice-applications/

Service Fabric Team blog
https://blogs.msdn.microsoft.com/azureservicefabric/tag/case-study/

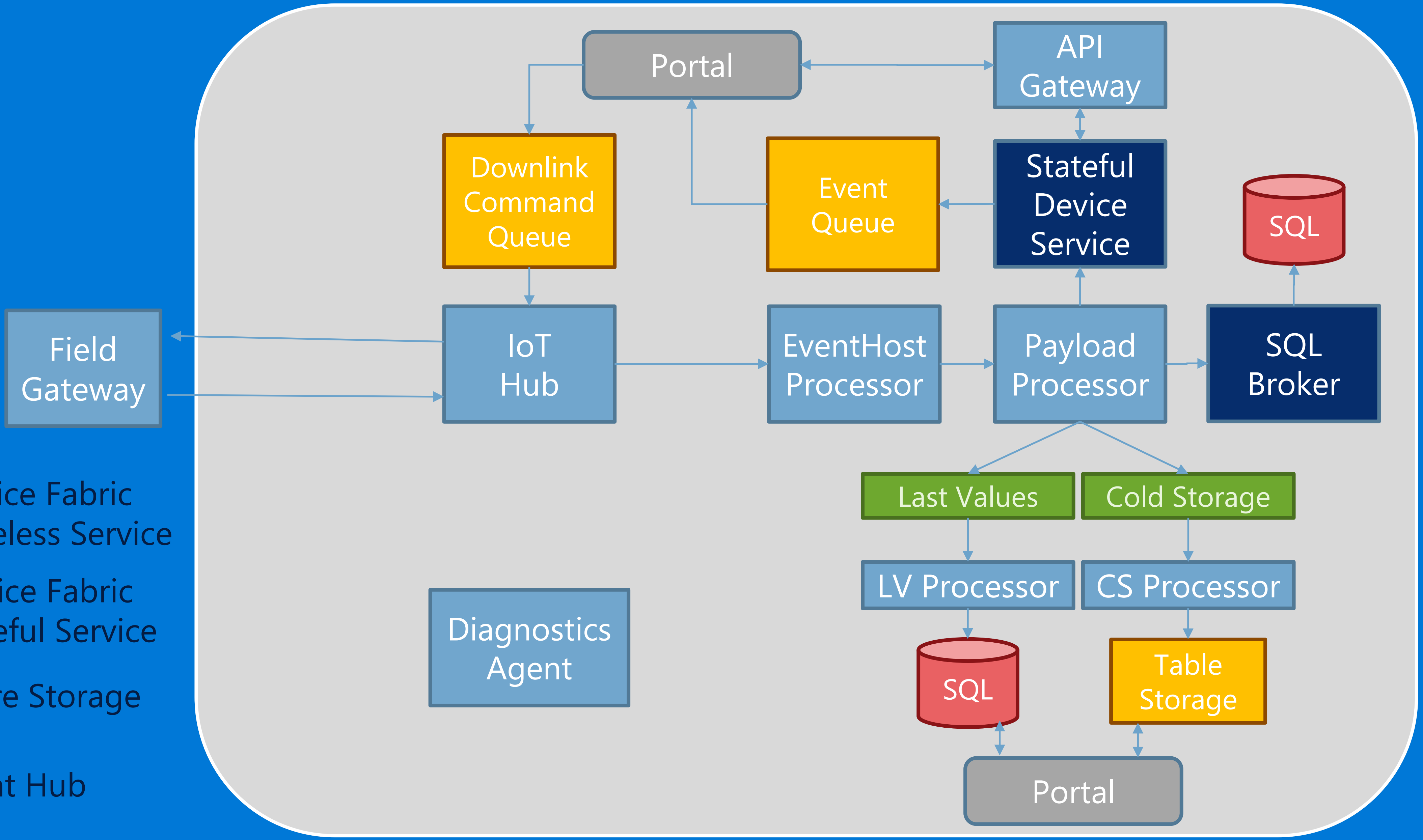
MESHSYSTEMS™

Mesh Systems IoT software and services



- Replaced existing Cloud Services solution for cost and microservices architecture for managing IoT devices
- Read the Mesh System blog post on Service Fabric team blog
  - https://blogs.msdn.microsoft.com/azureservicefabric/2016/06/20/service-fabric-customer-profile-mesh-systems/

Mesh Vista Architecture



Portal

API Gateway

Downlink Command Queue

Event Queue

Stateful Device Service

SQL

Field Gateway

IoT Hub

EventHost Processor

Payload Processor

SQL Broker

Last Values

Cold Storage

LV Processor

CS Processor

Diagnostics Agent

SQL

Table Storage

Portal

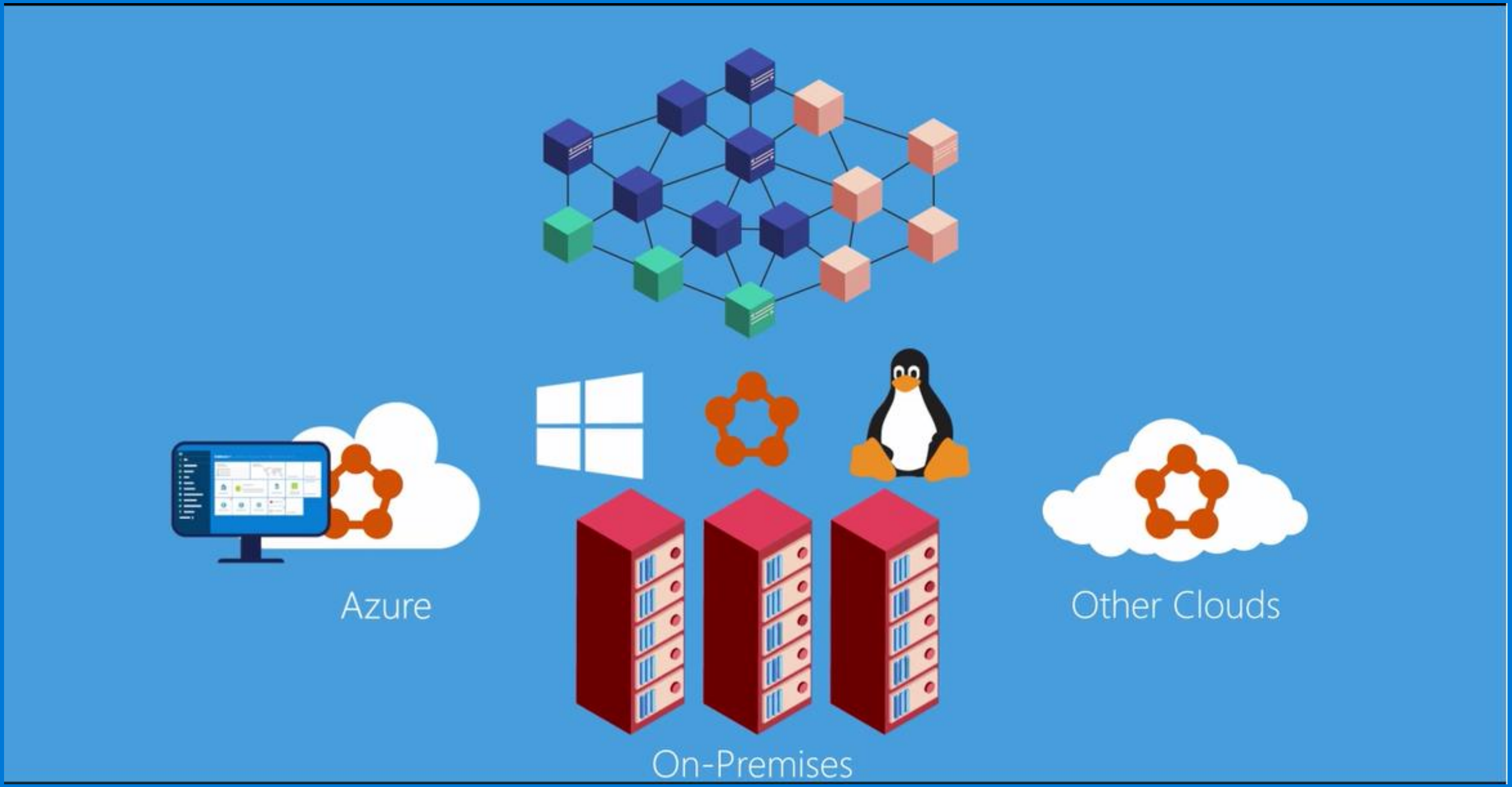
🟦 Service Fabric Stateless Service

■ Service Fabric Stateful Service

🟨 Azure Storage

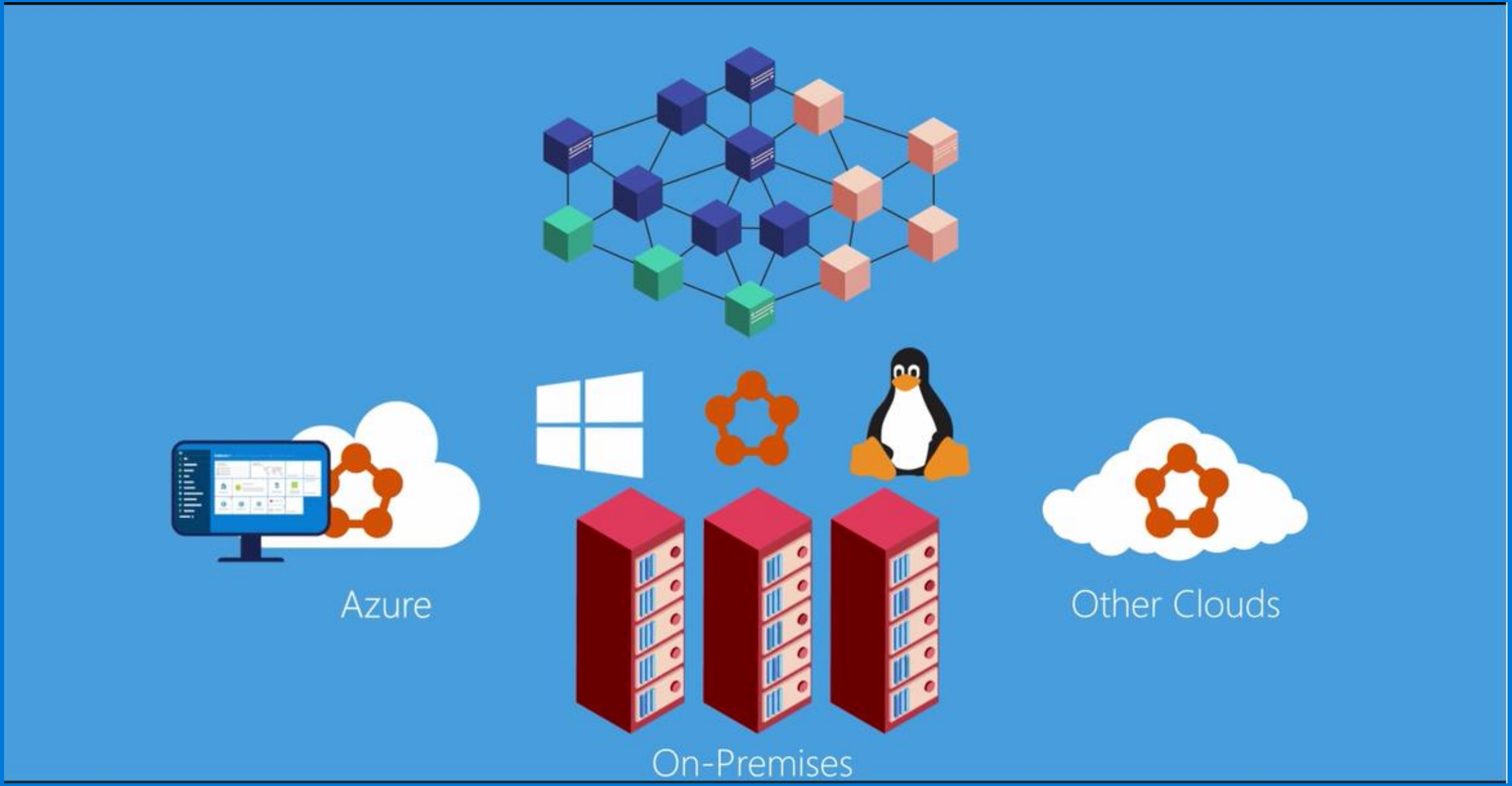
🟩 Event Hub

Word Count Sample



http://jianwsfcluster1g.eastasia.cloudapp.azure.com:8081/wordcount/

Visual Objects Sample



http://jianwsfcluster1g.eastasia.cloudapp.azure.com:8088/visualobjects/

Next steps and resources

- Docs Learning map and overview videos
  - https://azure.microsoft.com/en-us/documentation/services/service-fabric/

- Learn from samples, free clusters and labs
  - http://aka.ms/ServiceFabricSamples
  - https://github.com/Azure/azureservicefabricchina

- Questions? Comments? Issues? Join the monthly community call
  - https://stackoverflow.com/questions/tagged/azure-service-fabric
  - http://aka.ms/ServiceFabricForum
  - https://github.com/azure/service-fabric-issues

Q&A