Microsoft Azure

Hyper-converged solutions based on serverless and containers

Łukasz Kałużny, Marek Grabarz

Microsoft Azure

Agenda

- Azure Functions v2 runtime
- Kubernetes in Azure (AKS / ACI)
- Azure Service Fabric Mesh

Five demos! :-)

What is serverless?

Full abstraction of servers

Developers can just focus on their code—there are no distractions around server management, capacity planning, or availability.

Instant, event-driven scalability

Application components react to events and triggers in near real-time with virtually unlimited scalability; compute resources are used as needed.

Pay-per-use

Only pay for what you use: billing is typically calculated on the number of function calls, code execution time, and memory used.*

*Supporting services, like storage and networking, may be charged separately.

What are the benefits?

Focus

Solve business problems—not technology problems related to undifferentiated heavy lifting

Efficiency

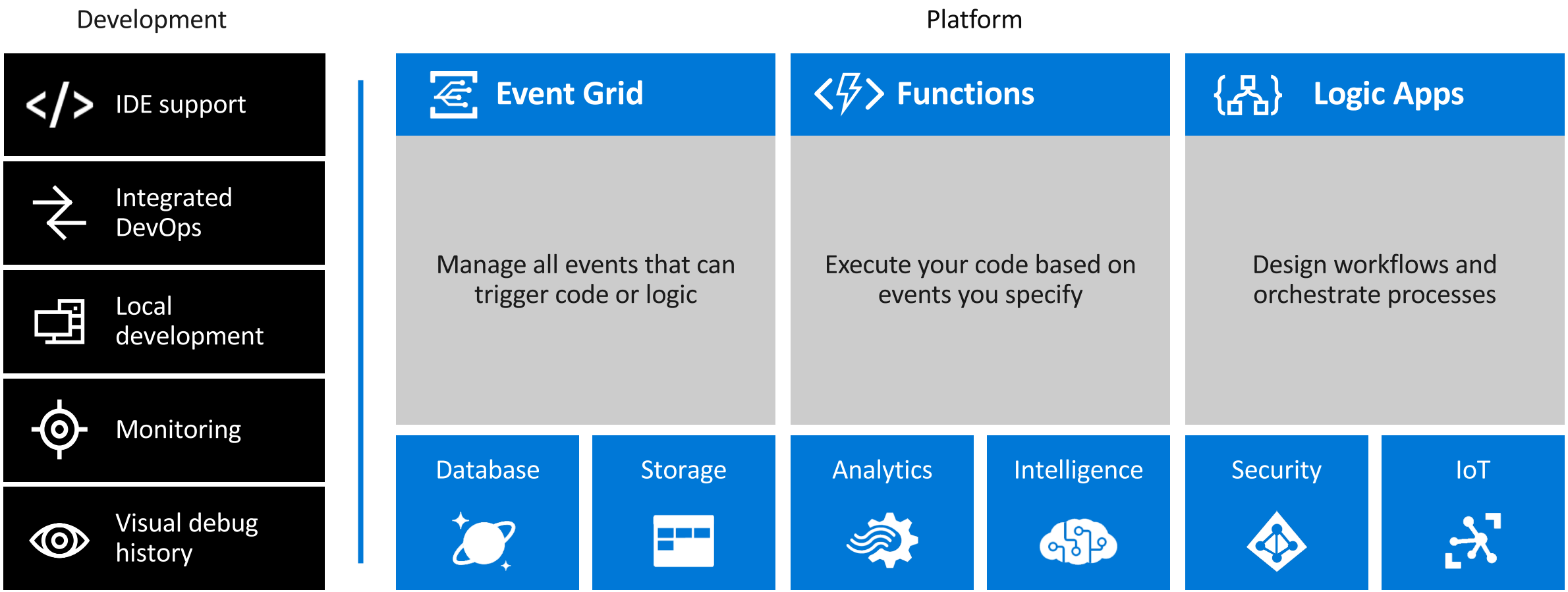
Shorter time to market
Fixed costs converted to variable costs
Better service stability
Better development and testing management
Less waste

Flexibility

Simplified starting experience
Easier pivoting means more flexibility
Easier experimentation
Scale at your pace—don't bet the farm on day 1
Natural fit for microservices

Full integration with Azure ecosystem

Functions is the center piece of the Serverless platform



Azure Functions

Events

React to timers, HTTP, or events from your favorite Azure services, with more on the way

Code

Author functions in C#, F#, Node.JS, Java, and more

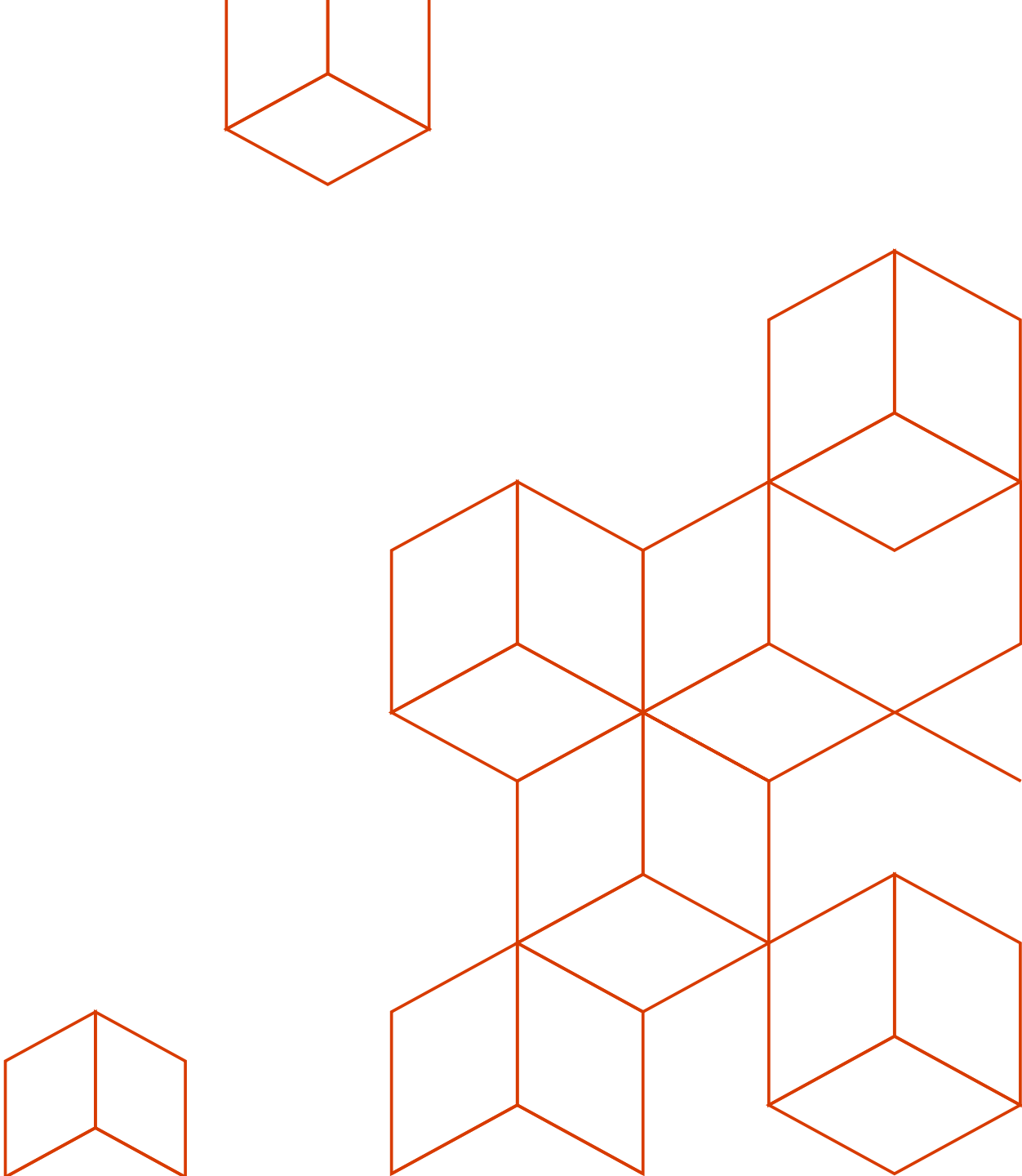
Outputs

Send results to an ever-growing collection of services

Functions 1.0 challenges

- Need for additional language support, e.g. Java, Python, Powershell
- Only able to host on Windows
- No support for development on Mac and Linux
- Assembly probing and binding issues for .NET developers
- Performance issues on a range of scenarios / languages
- Lack of UX guidance to production success

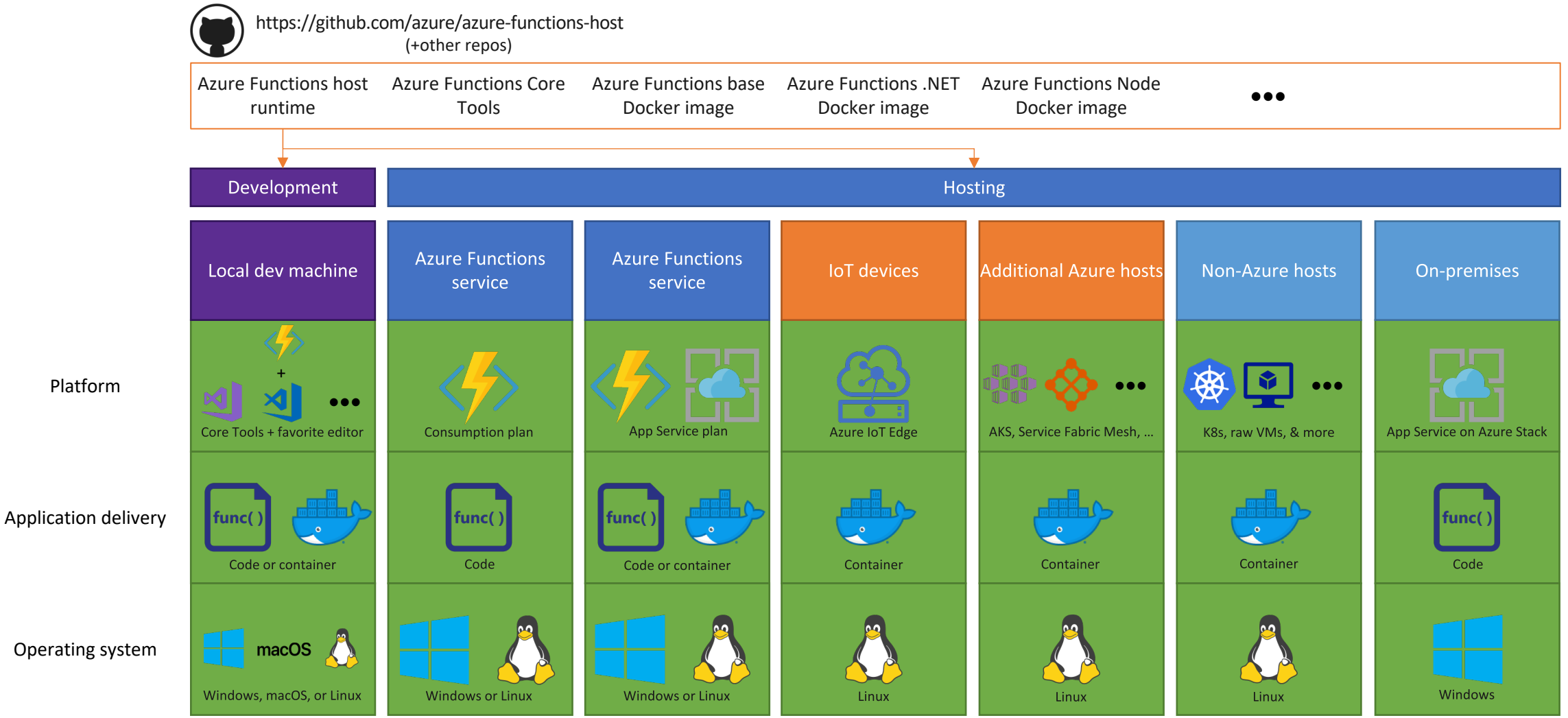
Introducing…
Azure Functions 2.0

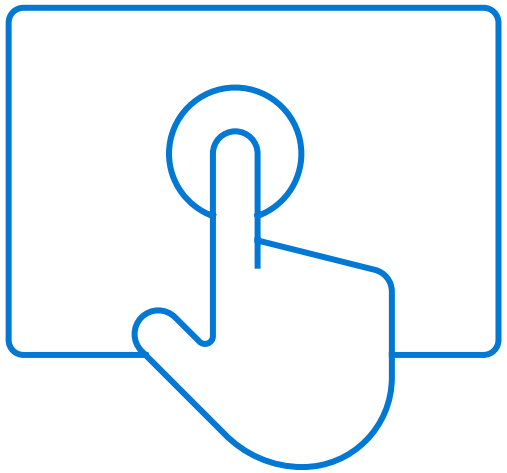


Functions runtime 1.0 and 2.0 key differences

	Functions 1.0	Functions 2.0
.NET Support	.NET Framework 4.7.1	.NET Core 2.1
Assembly isolation	No	Yes
Bindings versions	Runtime versions	User controlled
Language options	Limitations in languages and versions	Languages are external to the host
Node.js version	Node.js 6 only	Node.js 8 & 10 + future versions
Node.js native modules	Not supported	Supported
HTTP triggers	HTTP and specialized Webhooks	HTTP (supports Webhooks)
Language Runtime	Multiple languages per function app	Single language per function app
Functions Proxies	GA	GA
OpenAPI definition	Preview	Not yet available
Observability	Application Insights/WebJobs dashboard	App Insights

Functions everywhere





Demo

Azure Functions v2 in Kubernetes

Language options

Generally available

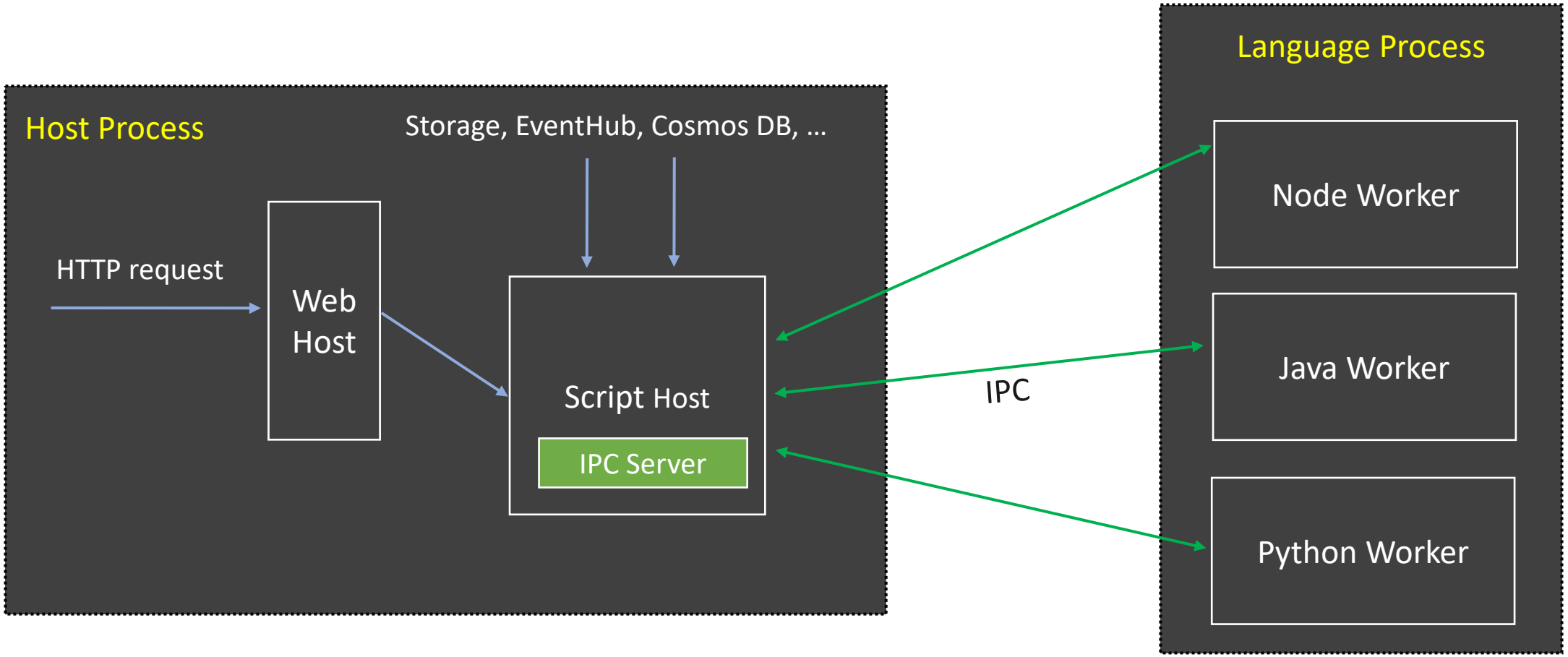
Public preview

Private preview

New!

More on the way!

Language Extensibility



- Worker and host broken into 2 separate processes
- Development of new language workers can happen independently
- Worker process crashes doesn’t bring down the host

Bindings and integrations

Functions 1.0

Microsoft.NET.Sdk.Functions (.NET Framework 4.6)

- HTTP
- Timer
- Storage
- Service Bus
- EventHubs
- Cosmos DB

Functions 2.0

Microsoft.NET.Sdk.Functions (.NET Standard 2.0)

- HTTP
- Timer

Microsoft.Azure.WebJobs.Extensions.Storage 3.0.0

Microsoft.Azure.WebJobs.Extensions.ServiceBus 3.0.0

Microsoft.Azure.Webjobs.Extensions.EventHubs 3.0.0

Microsoft.Azure.WebJobs.Extensions.CosmosDB 3.0.0

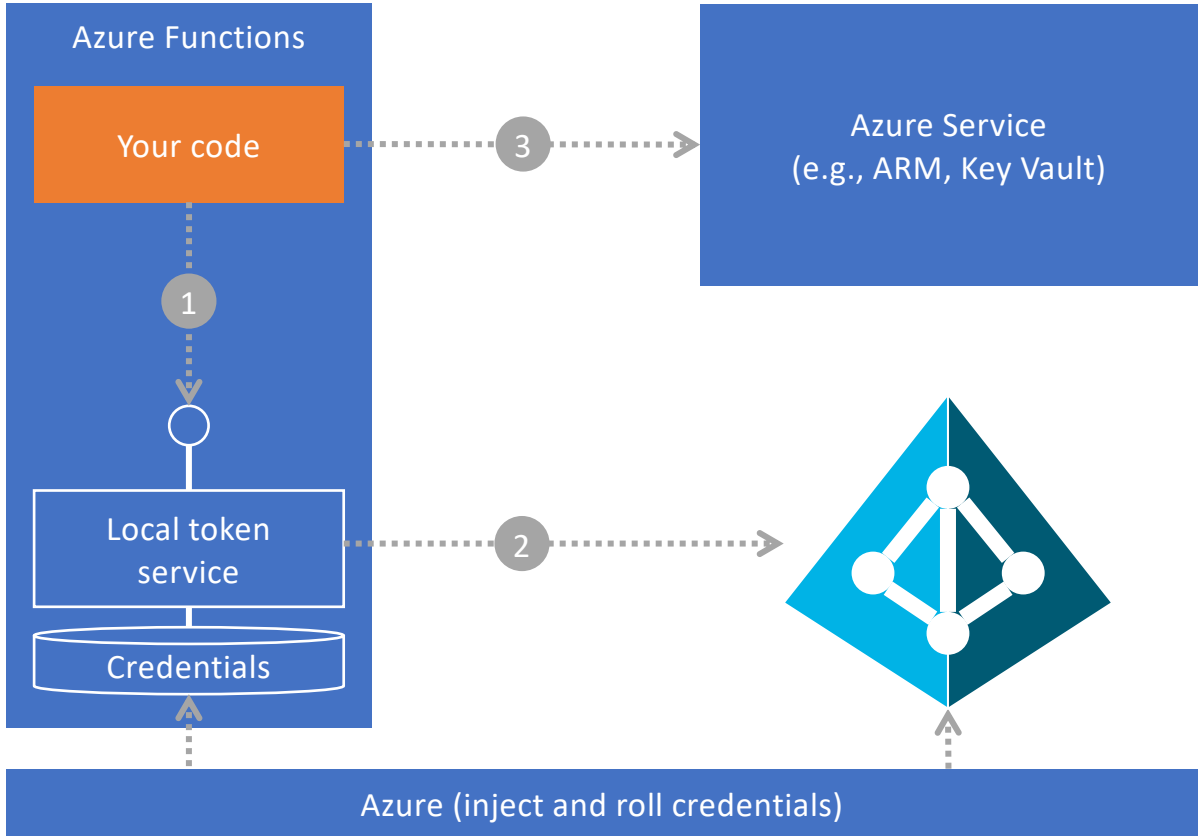
Microsoft.Azure.Webjobs.Extensions.EventGrid 2.0.0

Microsoft.Azure.WebJobs.Extensions.DurableTask 1.4.0

Microsoft.Azure.Webjobs.Extensions.MicrosoftGraph 1.0.0-beta

Managed identities for Azure Functions

- Keep credentials out of code
- Auto-managed identity in Azure AD for Azure resource
- Use local token endpoint to get access tokens from Azure AD
- Direct authentication with services, or retrieve creds from Azure Key Vault



Co

Gets
and i

Lever
your

Versi
previ

Simplifying security for serverless and web apps with Azure Functions and App Service

Opublikowano: 28 listopada, 2018

Matthew Henderson, Senior Program Manager, Azure Functions

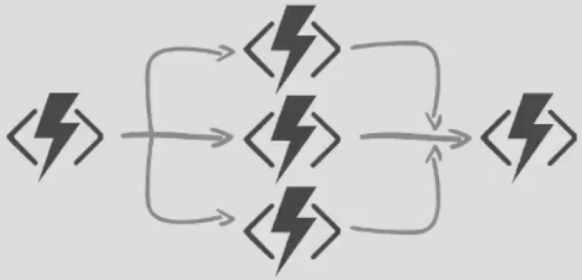
Serverless and PaaS are all about unleashing developer productivity by reducing the management burden and allowing you to focus on what matters most, your application logic. That can't come at the cost of security, though, and it needs to be easy to achieve best practices. Fortunately, we have a whole host of capabilities in the App Service and Azure Functions platform that dramatically reduce the burden of securing your apps.

Today, we're announcing new security features which reduce the amount of code you need in order to work with identities and secrets under management. These include:

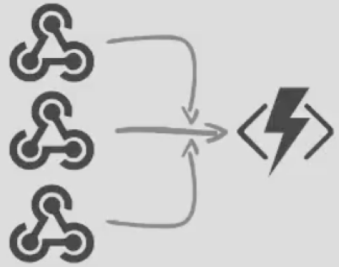
- Key Vault references for Application Settings (public preview)
- User-assigned managed identities (public preview)
- Managed identities for App Service on Linux/Web App for Containers (public preview)
- ClaimsPrincipal binding data for Azure Functions
- Support for Access-Control-Allow-Credentials in CORS config

What’s still hard?

Manageable Sequencing
+ Error Handling / Compensation

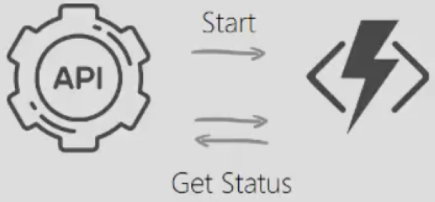


Fanning-out & Fanning-in



External Events Correlation

Flexible Automated Long-running
Process Monitoring



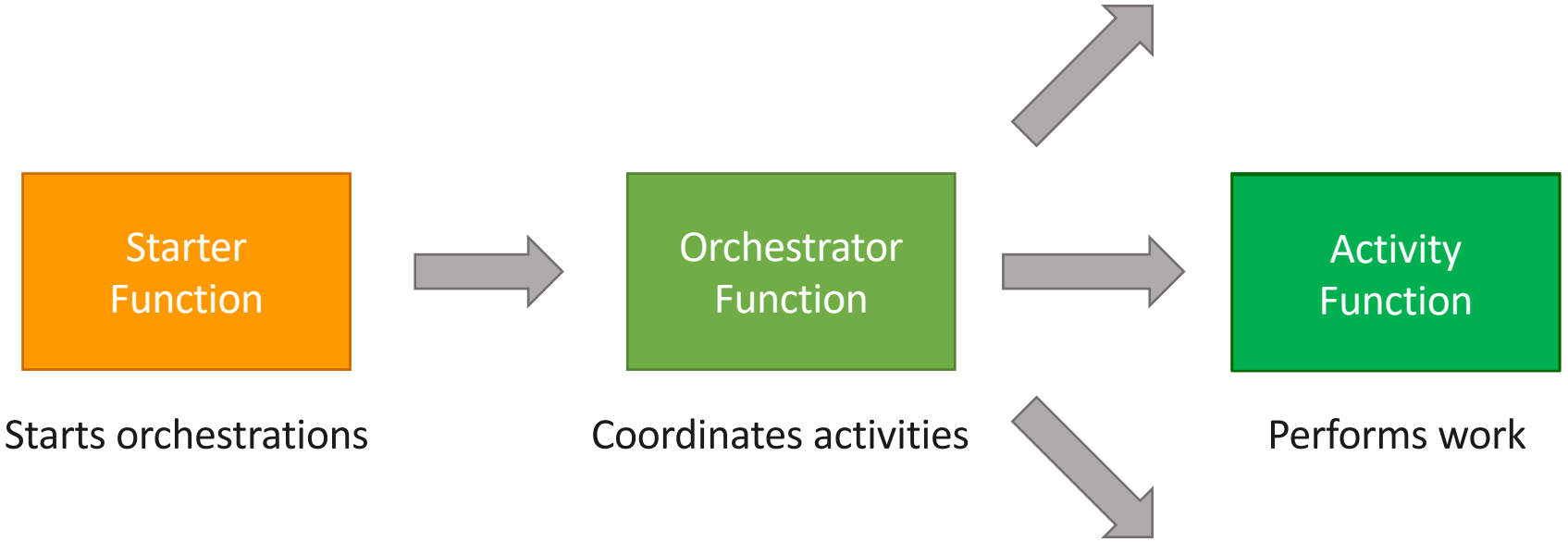
Http-based
Async Long-running APIs

Human Interaction

Durable Functions

- Write long-running orchestrations as a single function while maintaining local state.
- Simplify complex transactions and coordination (chaining, etc.) Easily call a Function from another Function, synchronously or asynchronously.
- All of the above using code-only. No JSON schemas. No graphical designer.
- Announcing: **Generally available (v2) - C# and JavaScript**

Components



OrchestrationClient

OrchestrationContextTrigger

ActivityTrigger

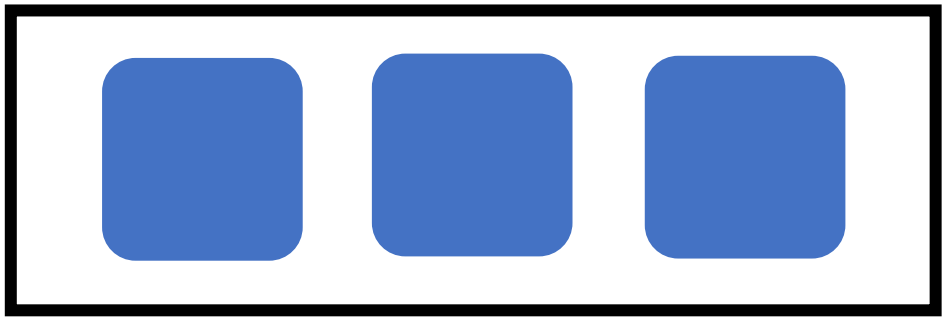
Azure Functions Hosting Options

Consumption

- Rapid scale out
- ”Unbounded” scale out
- No VNet connectivity available
- 10 minute execution
- Small instance size
- Scale to zero (cold start)

App Service Plan / Environment

- Auto-scale out (~5 min)
- Fixed scale out
- VNet connectivity / hybrid
- Unlimited execution duration
- Premium instance size
- Always on



Azure Functions Hosting Options

PRIVATE PREVIEW

Consumption

- Rapid scale out
- ”Unbounded” scale out
- No VNet connectivity available
- 10 minute execution
- Small instance size
- Scale to zero (cold start)

Functions premium plan

- Rapid scale out
- “Unbounded” scale out
- VNet connectivity / hybrid
- Unlimited execution duration
- Premium instance size
- Always on

Containers in Azure

App Service

Deploy web apps or APIs using containers in a PaaS environment

Service Fabric

Modernize .NET applications to microservices using Windows Server containers

Kubernetes Service

Scale and orchestrate Linux containers using Kubernetes

Container Instance

Elastically burst from your Azure Kubernetes Service (AKS) cluster

Ecosystem

Bring your Partner solutions that run great on Azure

Azure Container Registry

Docker Hub

Choice of developer tools and clients

Kubernetes: the industry leading orchestrator

Portable

Public, private, hybrid, multi-cloud

Extensible

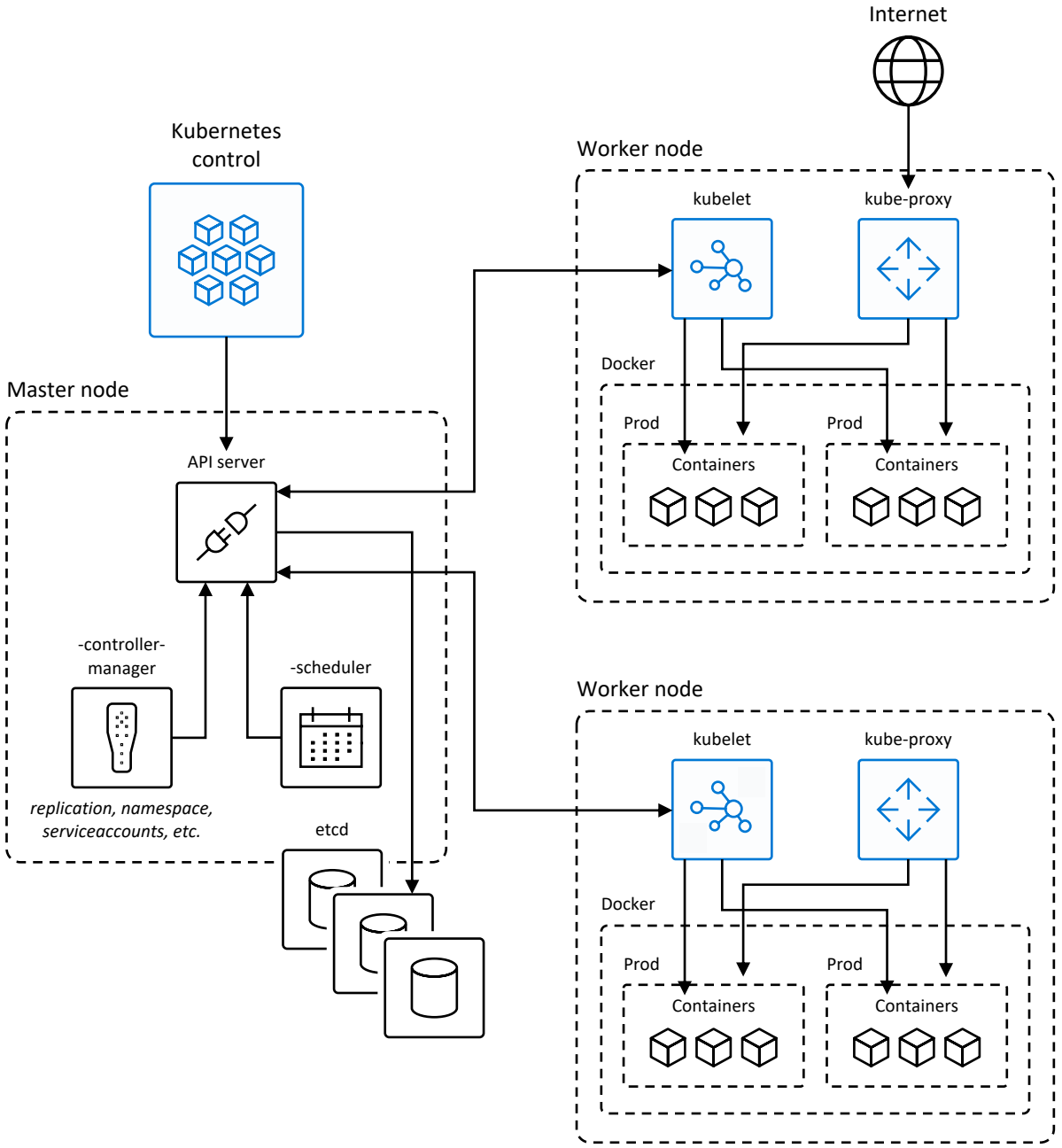
Modular, pluggable, hookable, composable

Self-healing

Auto-placement, auto-restart, auto-replication, auto-scaling

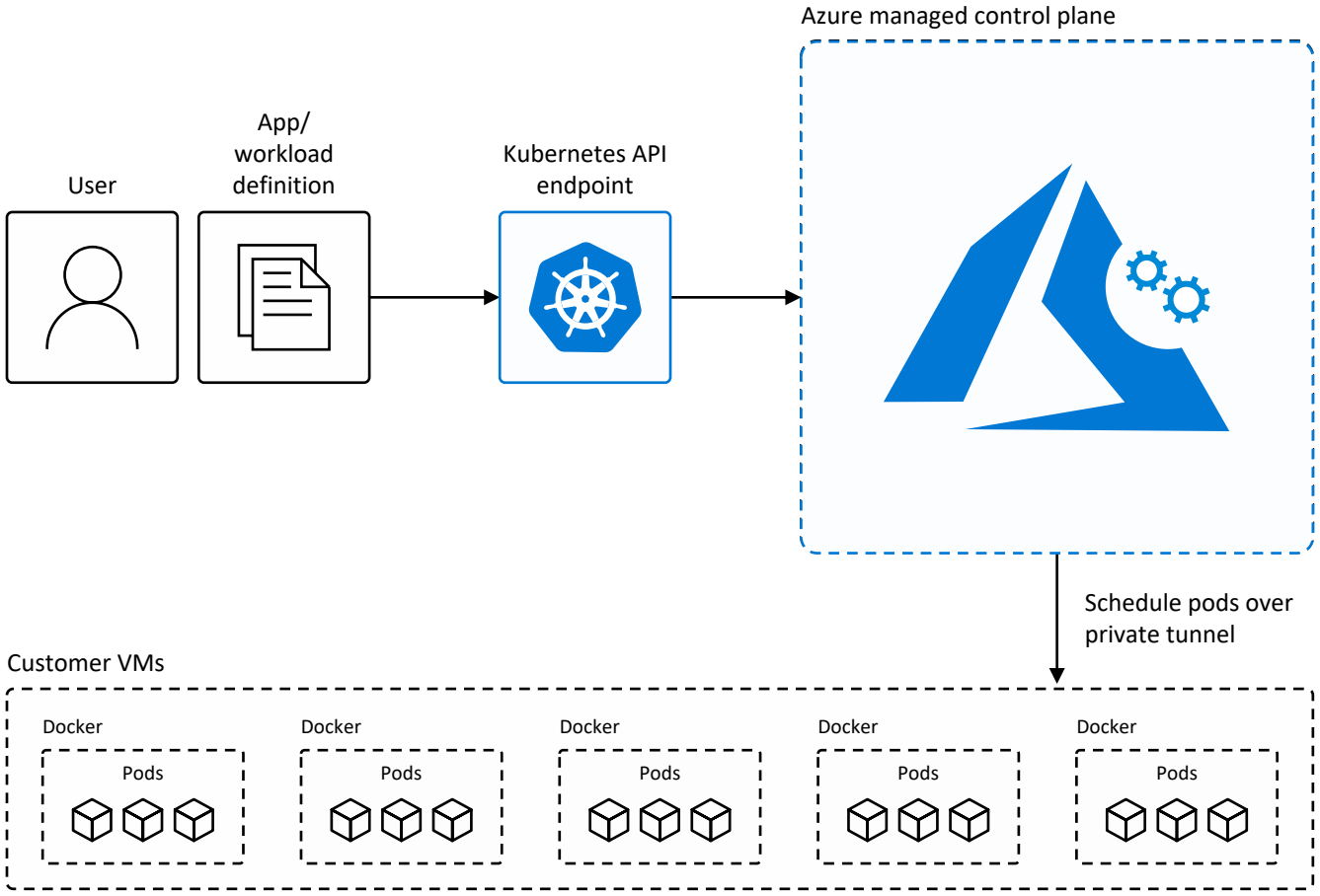
Kubernetes 101

1. Kubernetes users communicate with API server and apply desired state
2. Master nodes actively enforce desired state on worker nodes
3. Worker nodes support communication between containers
4. Worker nodes support communication from the Internet



How managed Kubernetes on Azure works

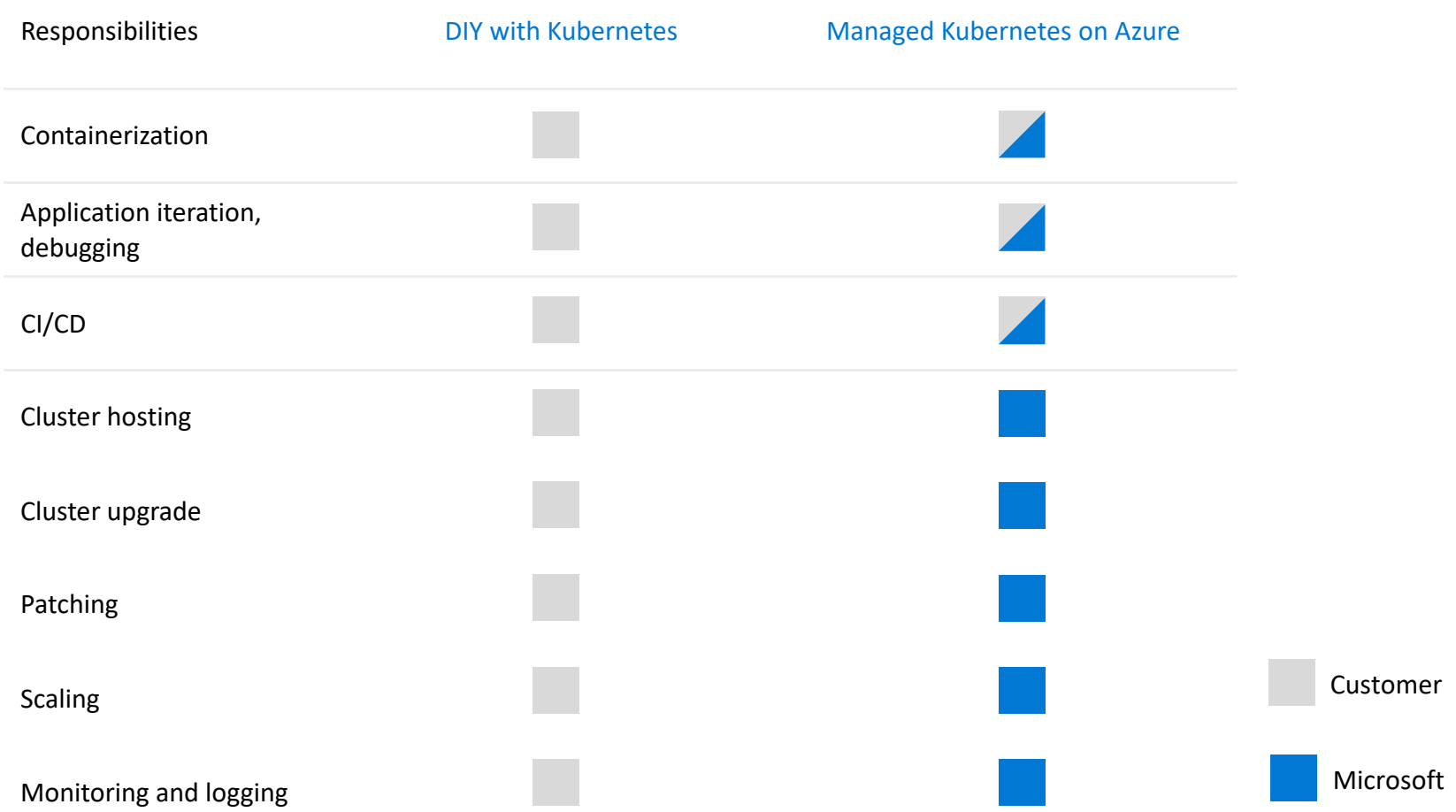
- Automated upgrades, patches
- High reliability, availability
- Easy, secure cluster scaling
- Self-healing
- API server monitoring
- At no charge



How managed Kubernetes on Azure works

Managed Kubernetes empowers you to do more

Focus on your containers and code, not the plumbing of them



AKS Get started easily

```
$ az aks create -g myResourceGroup -n myCluster --generate-ssh-keys
\ Running ..

$ az aks install-cli
Downloading client to /usr/local/bin/kubectl ..

$ az aks get-credentials -g myResourceGroup -n myCluster
Merged "myCluster" as current context ..

$ kubectl get nodes
NAME                       STATUS    AGE       VERSION
aks-mycluster-36851231-0   Ready     4m        v1.8.1
aks-mycluster-36851231-1   Ready     4m        v1.8.1
aks-mycluster-36851231-2   Ready     4m        v1.8.1
```

Manage an AKS cluster

```
$ az aks list –o table
Name                Location    ResourceGroup     KubernetesRelease  ProvisioningState
------------------  ----------  --------------  -------------------  ------------------
myCluster           westus2     myResourceGroup               1.7.7  Succeeded
```

```
$ az aks upgrade -g myResourceGroup -n myCluster –-kubernetes-version 1.8.1
\ Running ..
```

```
$ kubectl get nodes
NAME                        STATUS    AGE       VERSION
aks-mycluster-36851231-0    Ready     12m       v1.8.1
aks-mycluster-36851231-1    Ready     8m        v1.8.1
aks-mycluster-36851231-2    Ready     3m        v1.8.1
```

```
$ az aks scale -g myResourceGroup -n myCluster --agent-count 10
\ Running ..
```

Azure and Kubernetes integration

Cluster autoscaler

Storage

- Disks
- Files

Networking

- Azure CNI
- Load Balancer
- Application Gateway
- DNS (external-dns)

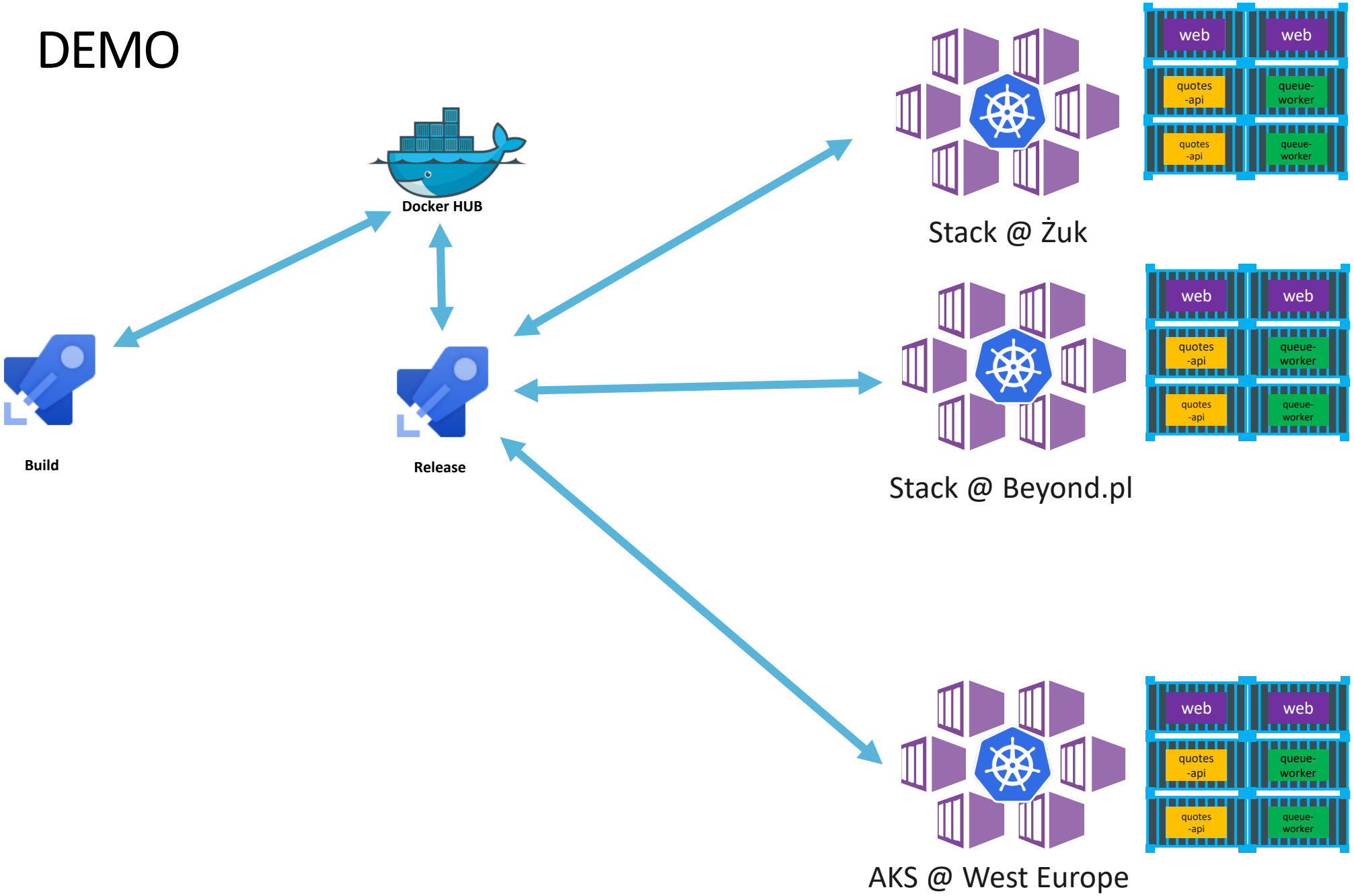
Azure AD

- RBAC
- Pod Identity

Key Vault

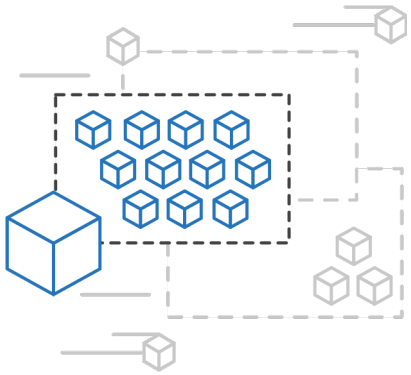
ACI

DEMO

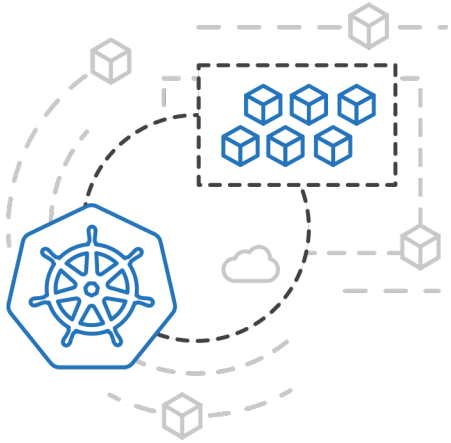

Docker HUB
Build
Release
web
web
quotes -api
queue- worker
quotes -api
queue- worker
Stack @ Żuk
web
web
quotes -api
queue- worker
quotes -api
queue- worker
Stack @ Beyond.pl
web
web
quotes -api
queue- worker
quotes -api
queue- worker
AKS @ West Europe

Azure Container Instances (ACI)

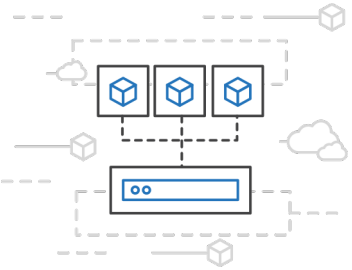
Easily run serverless containers



Run containers without managing servers

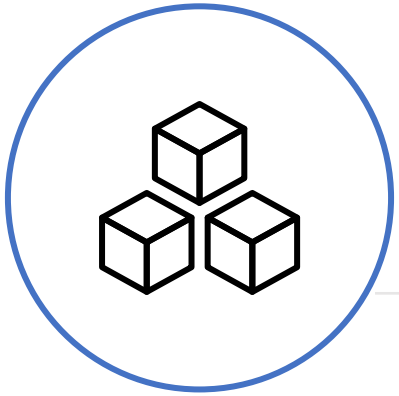


Containers as a primitive billed per second



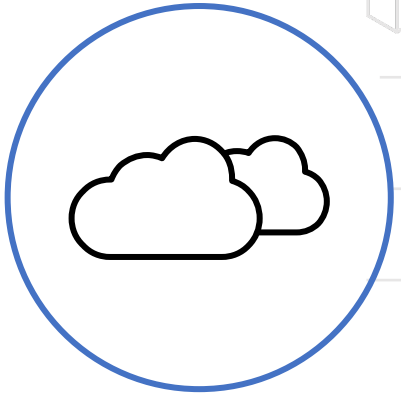
Secure applications with hypervisor isolation

What can you build with ACI today



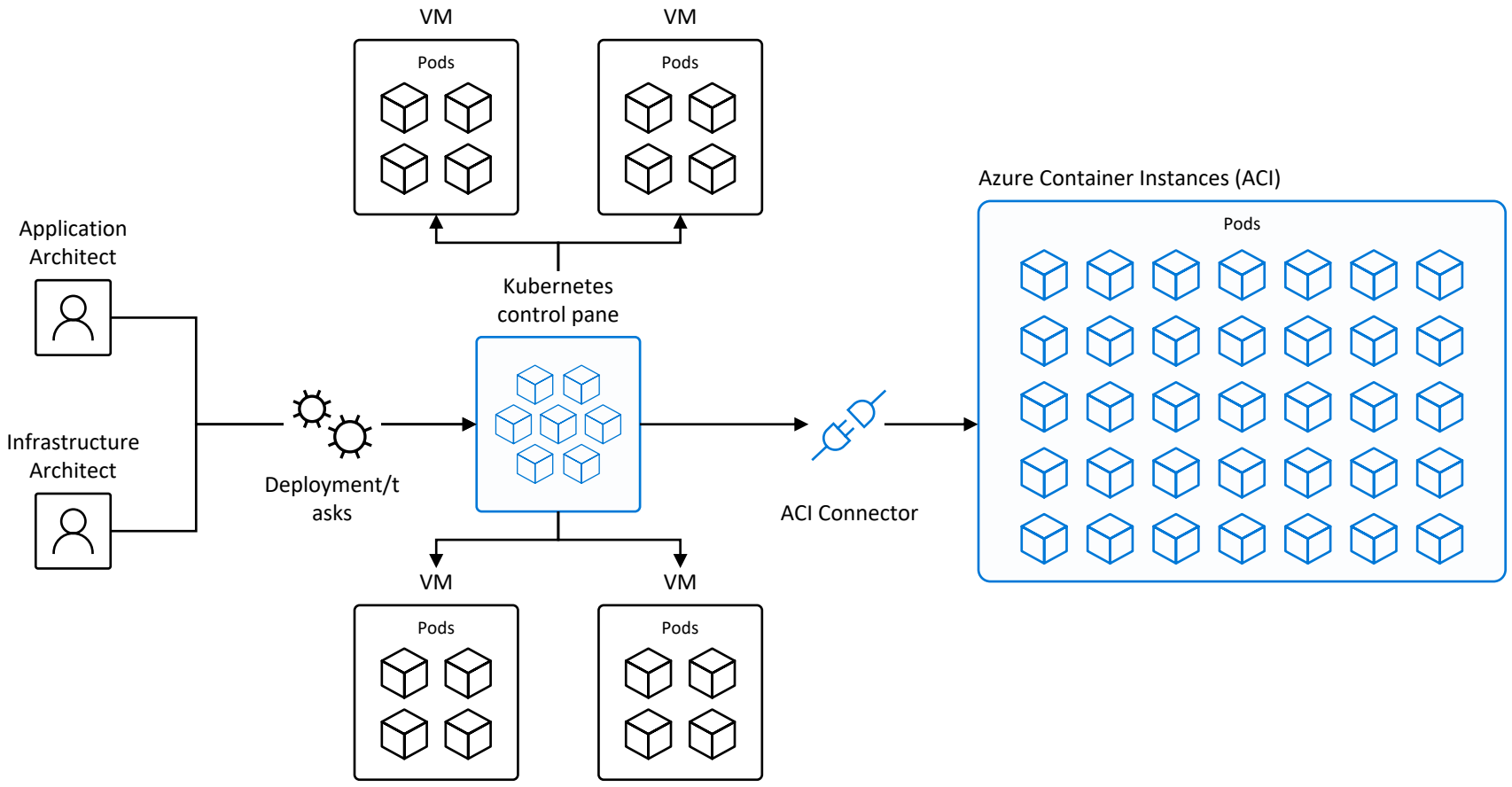
Modular apps

Event Driven Apps



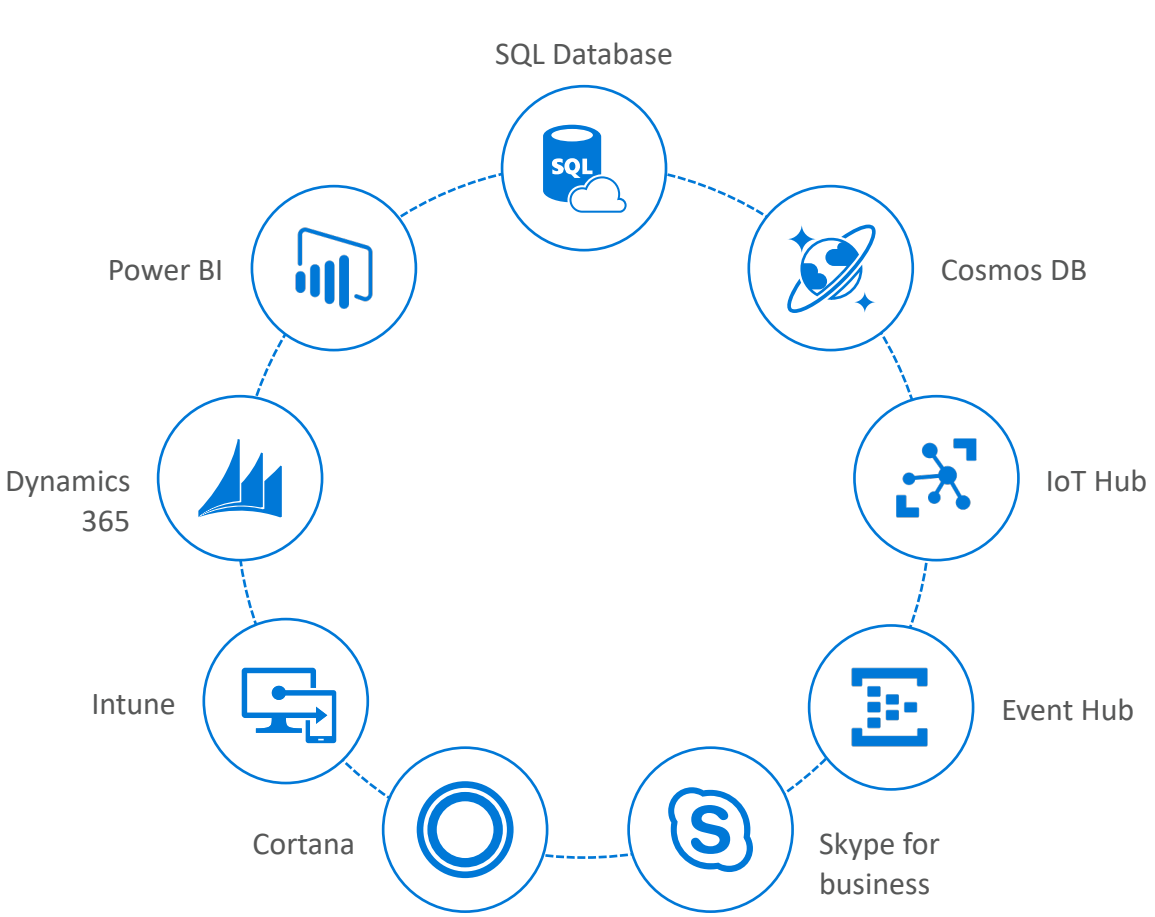
Elastic Bursting (AKS)

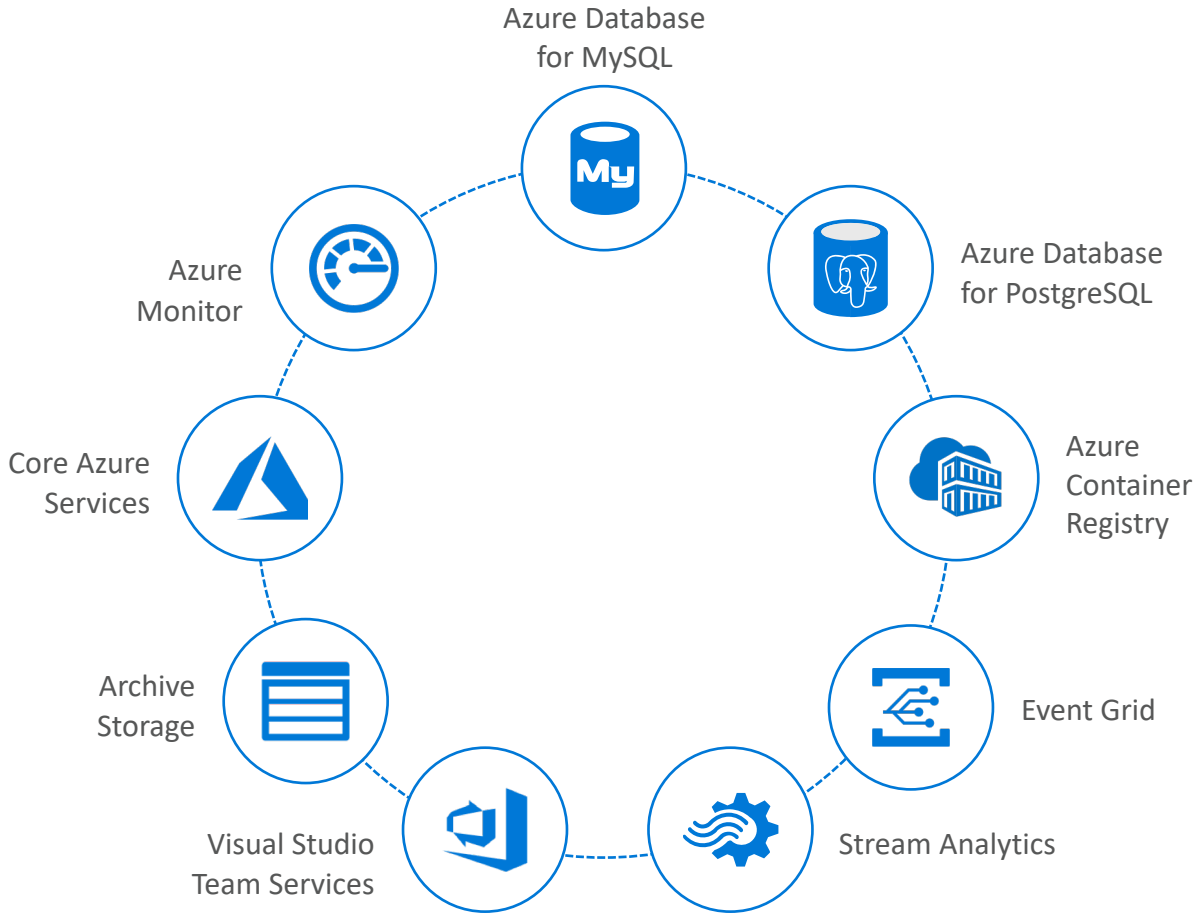
Bursting with the ACI Connector/ Virtual Kubelet



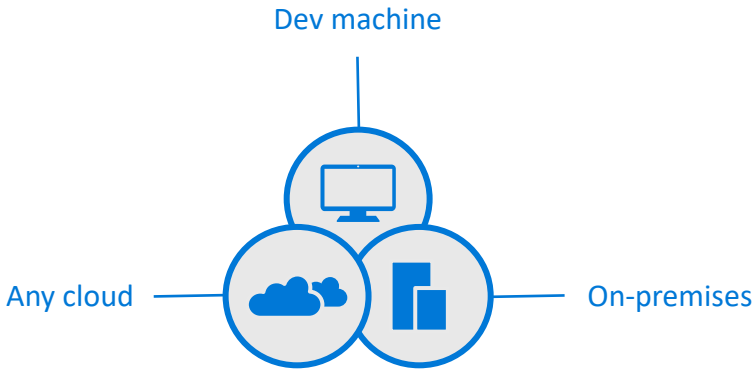
Powering Azure and Microsoft services

Azure Service Fabric is designed for mission-critical services





Azure Service Fabric offerings



Service Fabric Standalone

Bring your own infrastructure

Azure Service Fabric

Dedicated Azure clusters

Azure Service Fabric Mesh

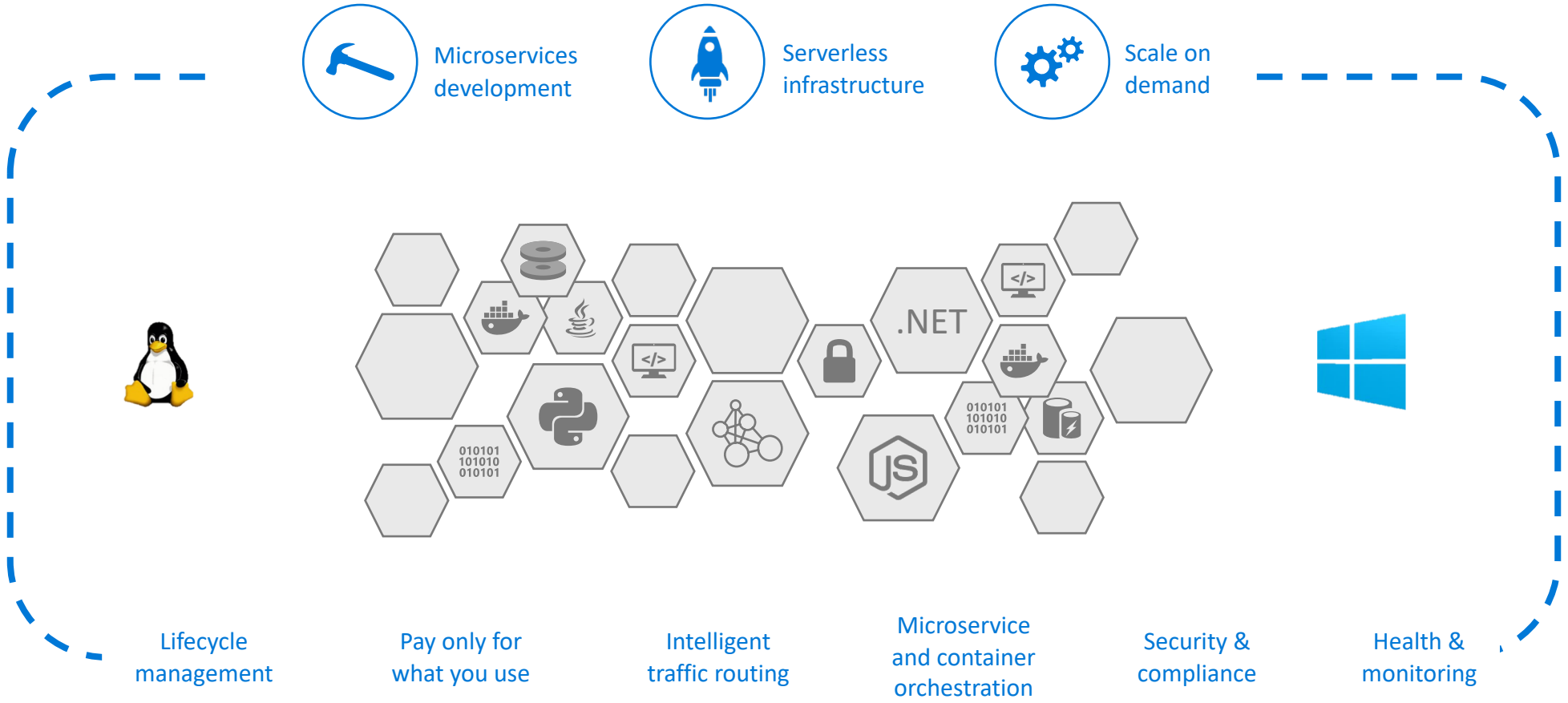
Serverless microservices

Full Control ⟷ *Fully managed*

Service Fabric

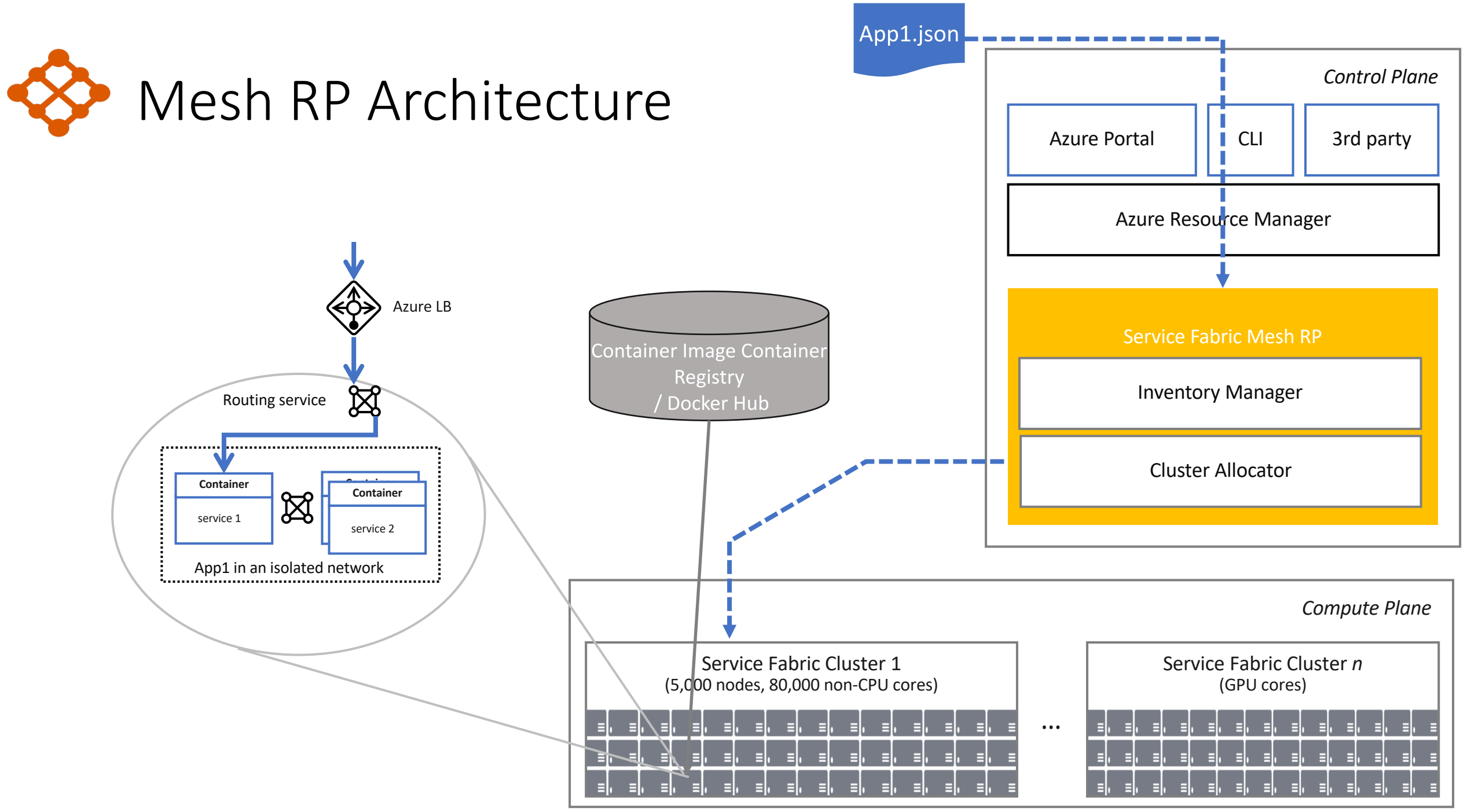
Azure Service Fabric Mesh

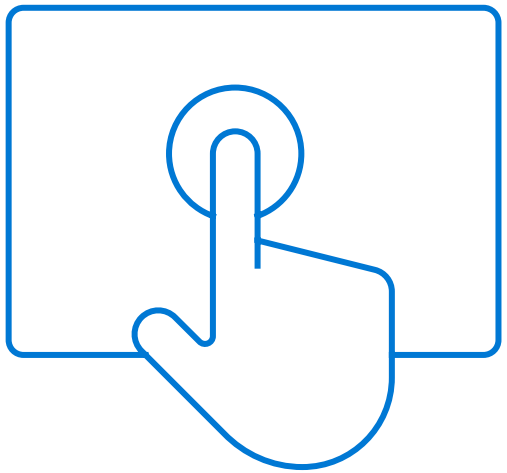
A Serverless microservices platform for business critical applications



Service Fabric Mesh

Mesh RP Architecture





Demo

Portal, Application logs, Cloud shell

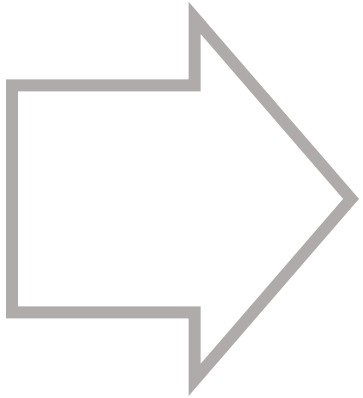
Service Fabric Resource Model

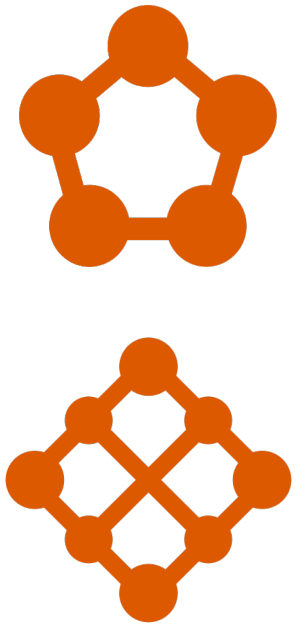
Simple, declarative way to deploy and manage applications.

Individual resources can be deployed to any Service Fabric environment.

Compatible with Azure Resource Manager.

- Applications and Services
- Networks
- Gateways
- Secrets
- Volumes
- Routing rules

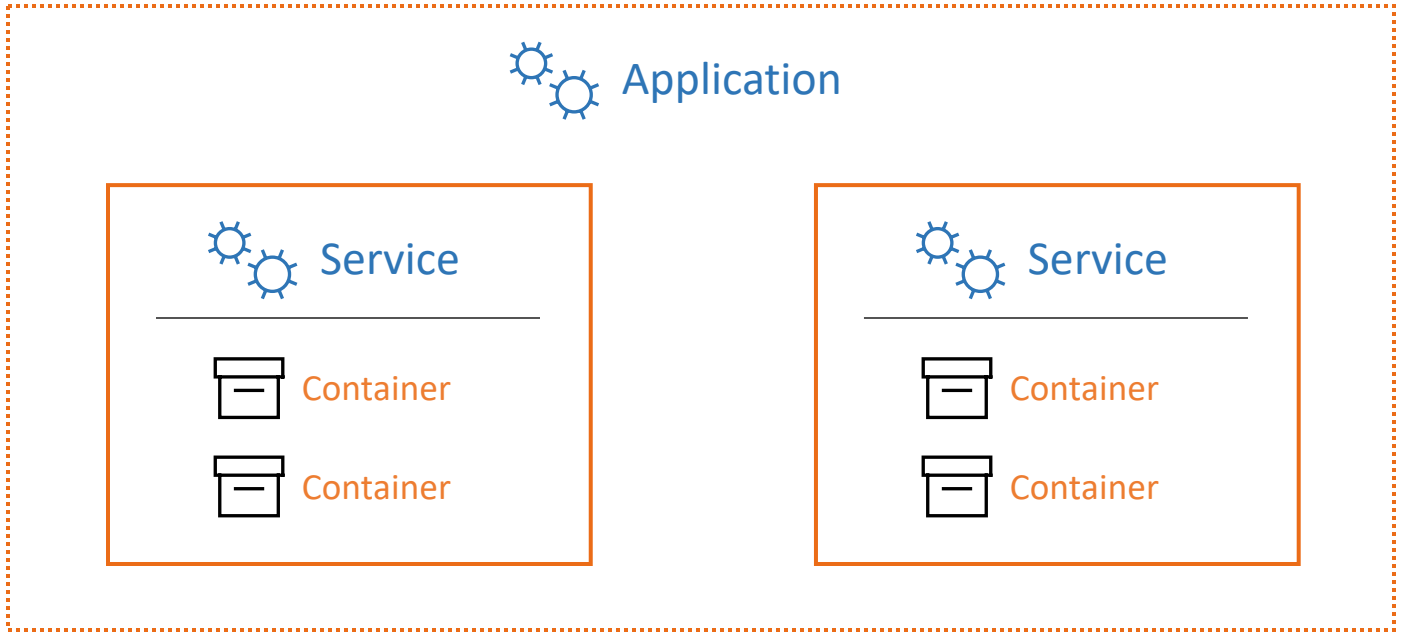




Application and Service Resources

Services describe how a set of containers run

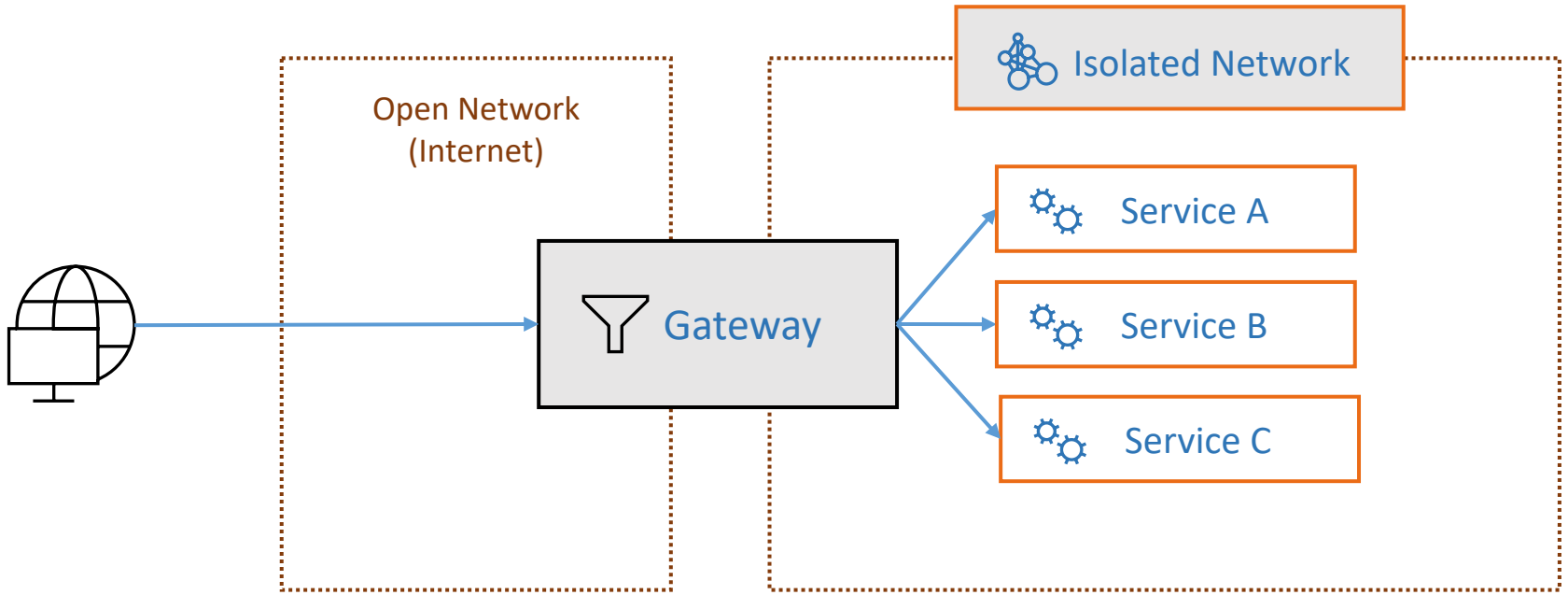
- Container image, environment variables, CPU/Memory requirements, endpoints, etc.
- References to other resources, such as networks, volumes, and secrets.
- Applications are a group of services.



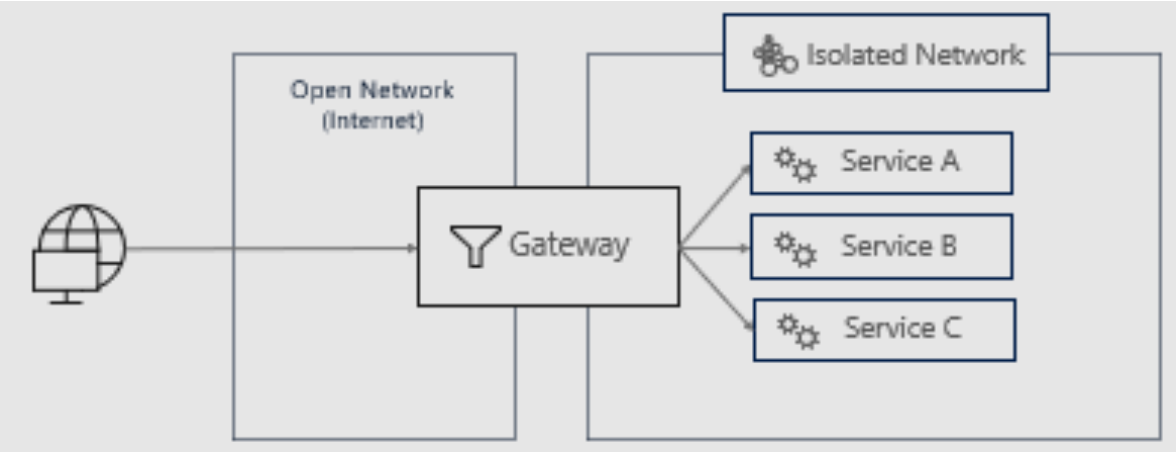
Gateway and networks

Gateway connects two networks and routes traffic

- L4 routing: TCP
- L7 routing: HTTP(S)



Gateway resource - Snippet

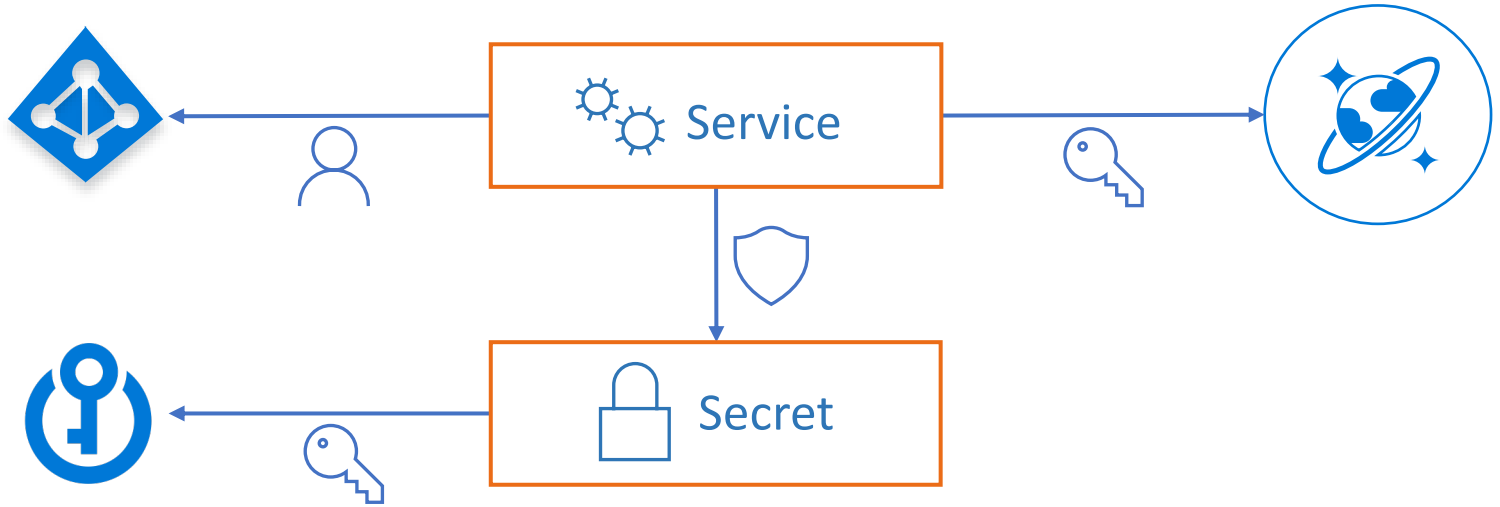



```
{
  "apiVersion": "2018-09-01-privatepreview",
  "name": "Demo1Gateway",
  "type": "Microsoft.ServiceFabricMesh/gateways",
  "location": "[parameters('location')]",
  "dependsOn": [
    "Microsoft.ServiceFabricMesh/networks/Demo1Network"
  ],
  "properties": {
    "description": "Gateway for Demo 1",
    "sourceNetwork": {
      "name": "Open"
    },
    "destinationNetwork": {
      "name": "[resourceId('Microsoft.ServiceFabricMesh/networks', 'Demo1Network')]
    },
    "http": [
      {
        "name": "aspnetsite",
        "port": 80,
        "hosts": [
          {
            "name": "contoso.com",
            "routes": [
              {
                "name": "index",
                "match": {
                  "path": {
                    "value": "/myapp",
                    "rewrite": "/",
                    "type": "prefix"
                  }
                },
                "destination": {
                  "applicationName": "Demo1Application",
                  "serviceName": "AspNetCoreWebService",
                  "endpointName": "AspNetCoreWebServiceEndpoint"
                }
              }
            ]
          }
```

Secrets Resource

Inline or stored in Azure Key Vault

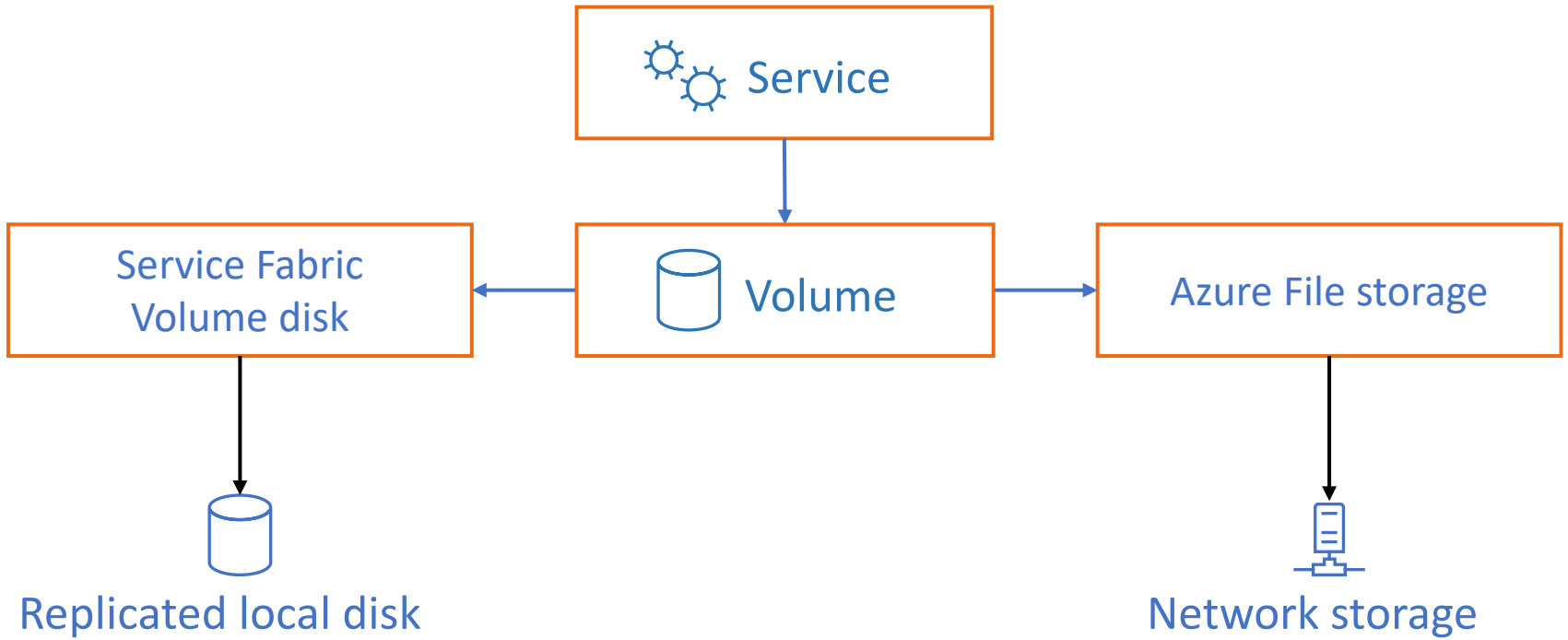
- Applications and service resources have Managed Service Identity (MSI) with AAD to be able to access secrets in Azure Key Vault
- Secrets and certificates can be auto-rolled over with policies



Volume Resource

General-purpose file storage

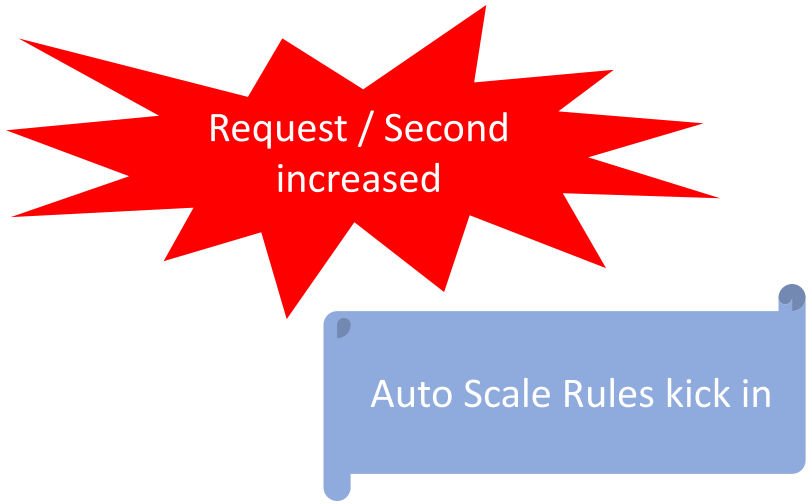
- Read and write files using normal disk I/O file APIs.
- Backed by Azure File storage or Service Fabric Volume disk

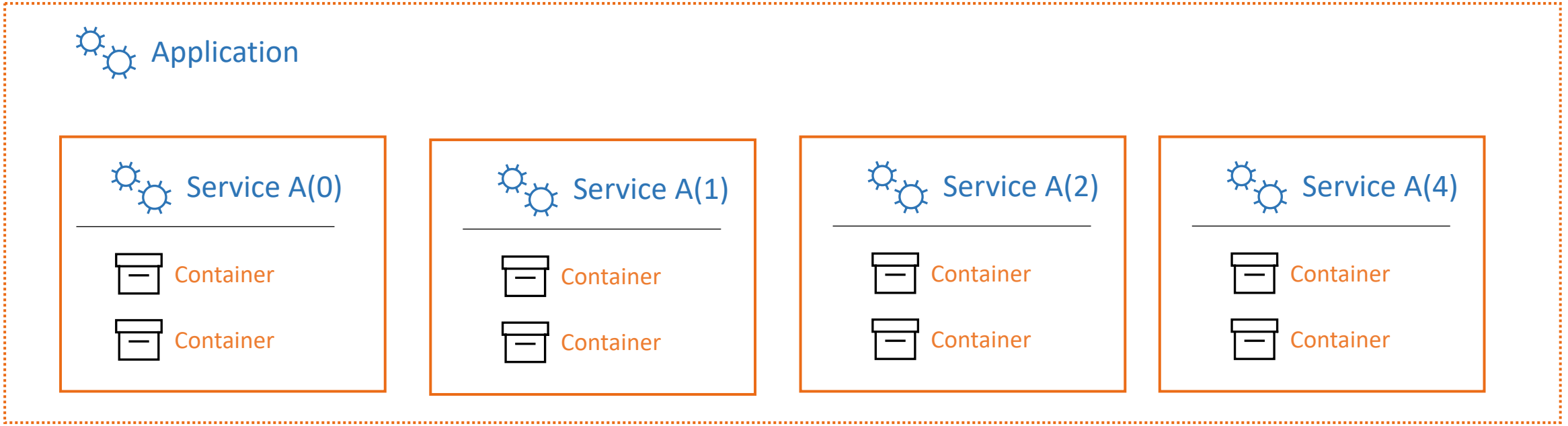


Auto Scale property

Horizontal scaling of services based on

- CPU utilization
- Memory utilization
- Application provided custom metric





Defining the Auto Scale property (JSON)

<excerpt from service JSON>

```
…
replicaCount": 1,
"autoScalingPolicies": [
  {
    "name": "AutoScaleWorkerRule",
    "trigger": {
      "kind": "AverageLoad",
      "metric": {
        "kind": "Resource",
        "name": "cpu"
      },
      "lowerLoadThreshold": "0.2",
      "upperLoadThreshold": "0.8",
      "scaleIntervalInSeconds": "60"
    },
    "mechanism": {
      "kind": "AddRemoveReplica",
      "minCount": "1",
      "maxCount": "40",
      "scaleIncrement": “2"
    }
  }
]
…
```

In addition to average load for the metric over the time period, what other options does a user have?

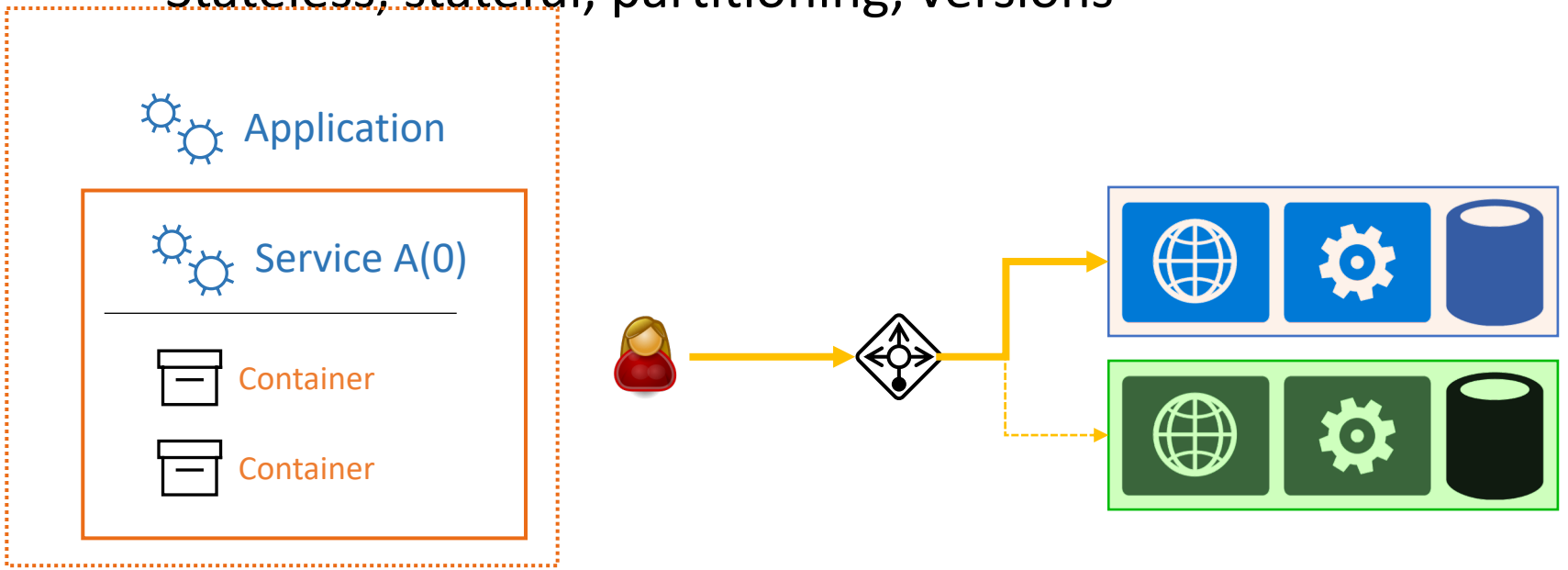
Does this mean scaling happens once every 60 seconds? What is the min / max interval?

What other options does a user have?

Does this mean 2 replicas at a time? Ie if this app were to scale to max, we’d have to see average cpu > 0.8 across replicas and still see 2 more being added every 60 seconds, so about 60*(38/2) sec?

Routing rules resource - Intelligent traffic routing

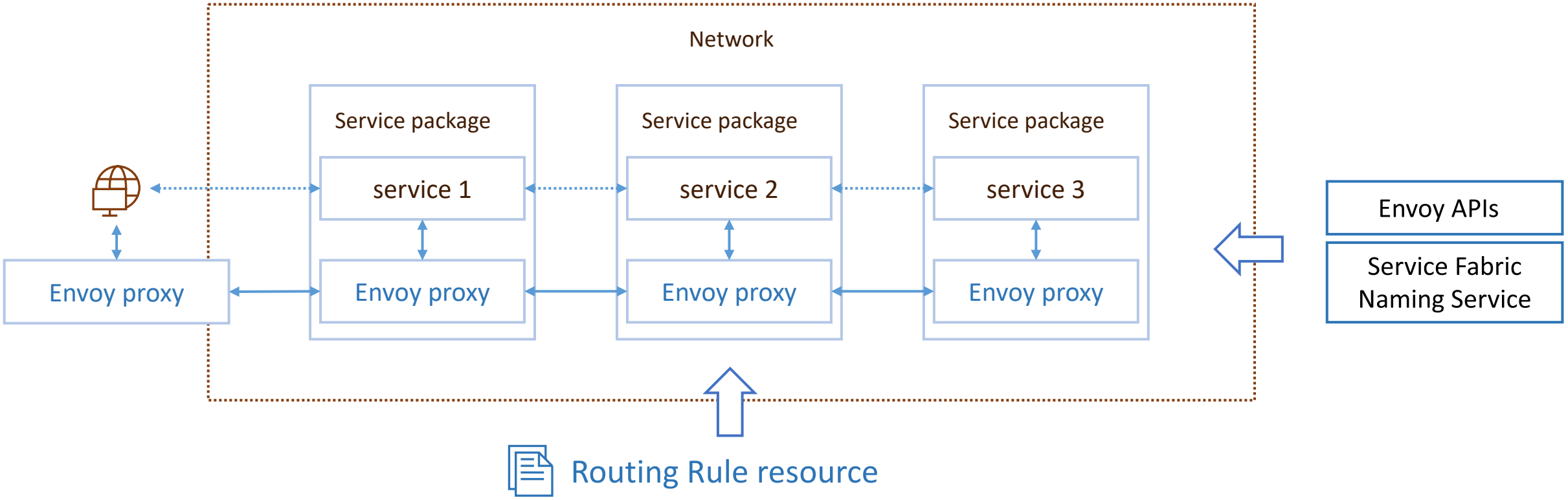
- Services address each other by hostnames.
- Services do not implement platform-specific discovery APIs.
- Services do not deal with network-level details.
  - Retries, circuit breakers, throttling
- Services are unaware of the implementation details of other services.
  - Stateless, stateful, partitioning, versions



Routing Rule Resource - Intelligent traffic routing

Advanced HTTP/HTTPS traffic routing with load balancing

Proxy handles partition resolution and key hashing

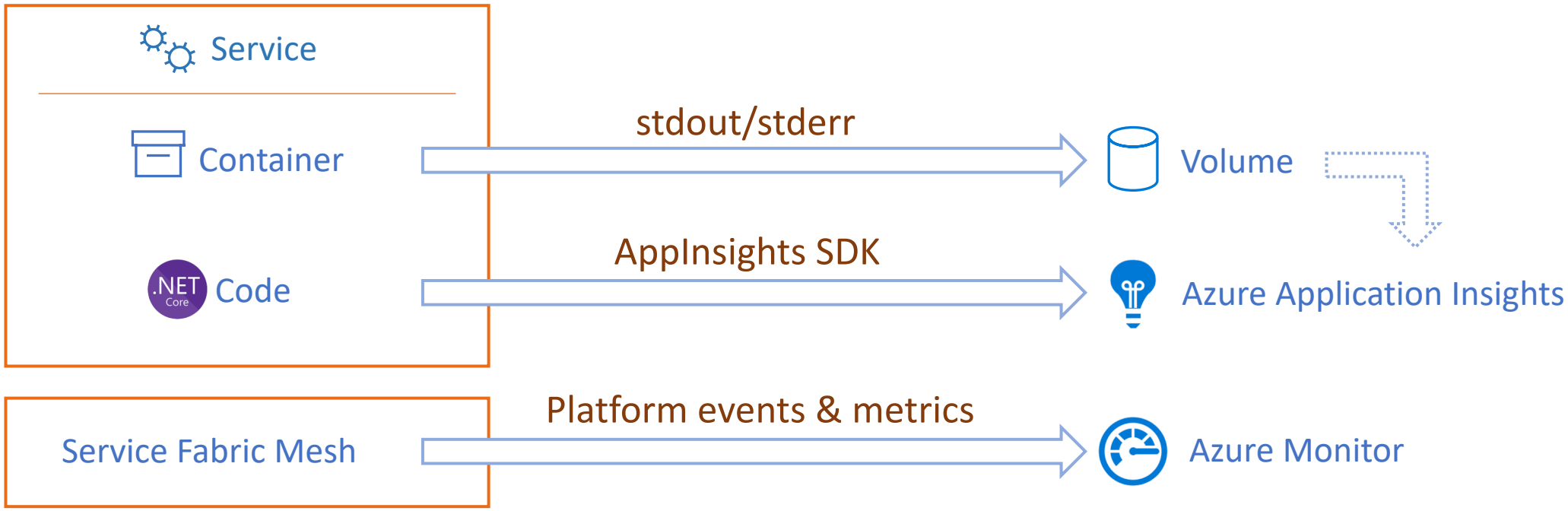


Diagnostics and Monitoring

Use your favorite APM platform to monitor apps inside containers

Containers write out stdout/stderr logs

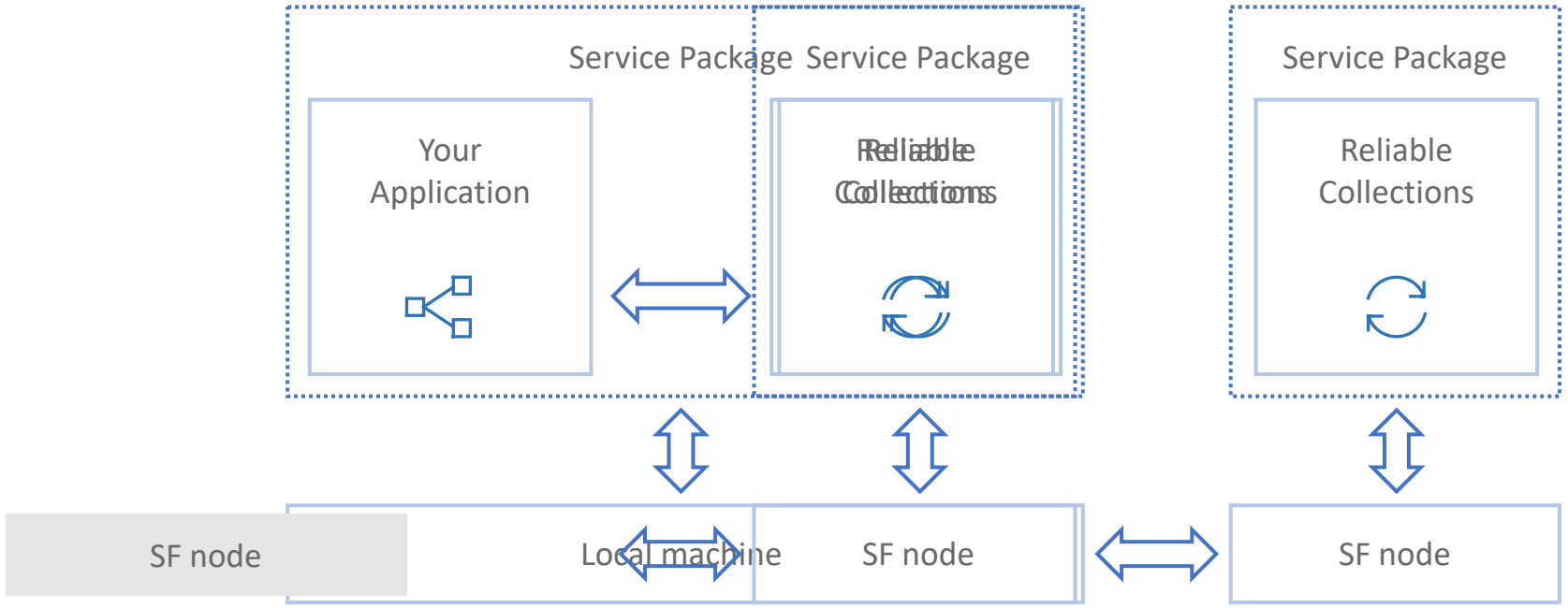
Azure Monitor for platform events and container metrics



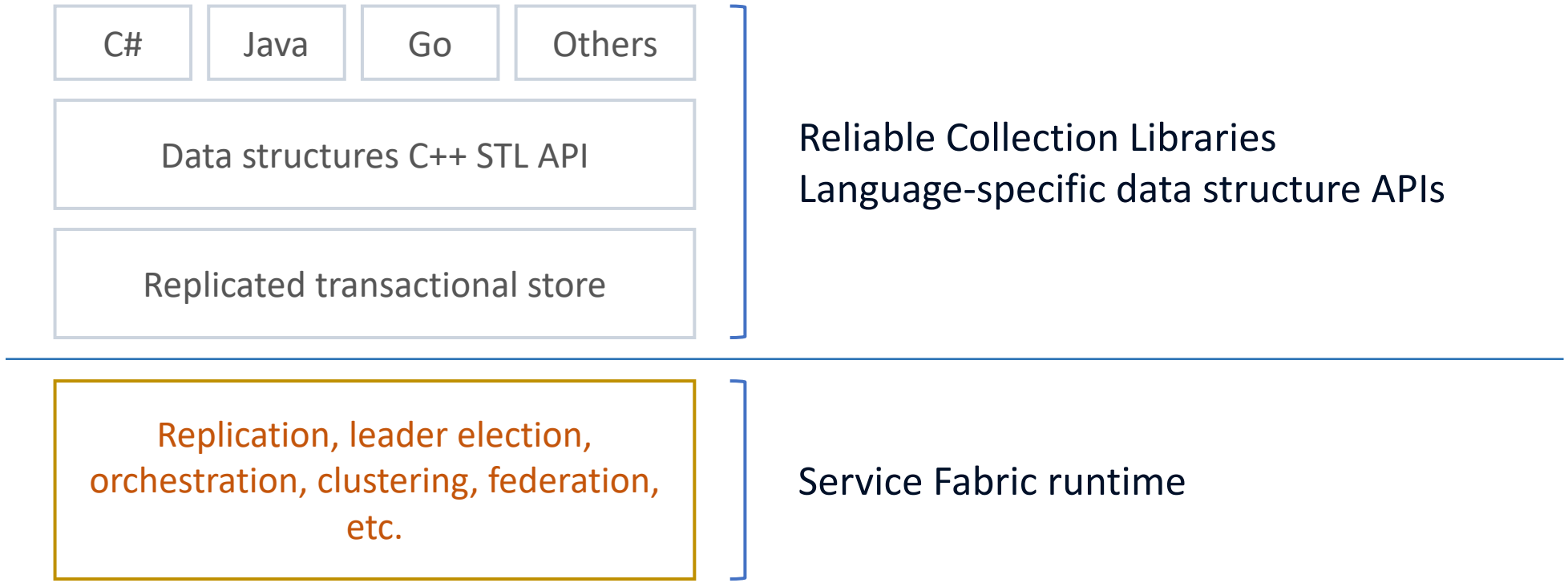
Reliable Collections

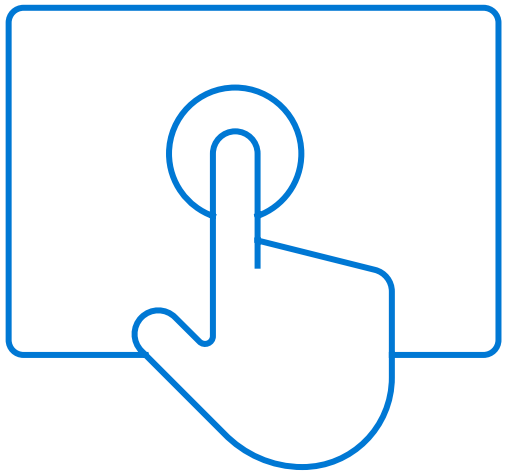
Runs anywhere

- Local disk storage when running outside of Service Fabric
- + Replication and partitioning when running on Service Fabric



Reliable Collections



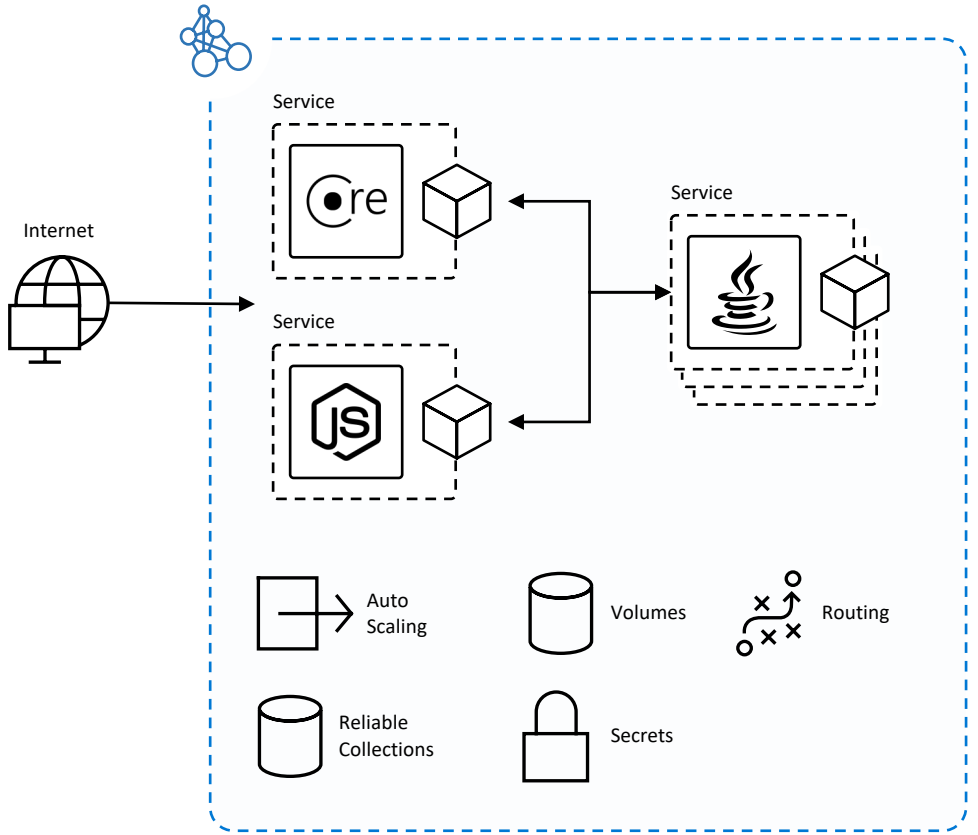


Demo

Visual Studio Experience

Modernize .NET application with Service Fabric

- Polyglot services, described by resources
- Deploy anything and everything in a container
- Bring your own network to connect to other systems
- No code changes required
- No servers or VMs to manage with Service Fabric Mesh
- Enhanced with Reliable Collections and volume drives for state
- Connected through intelligent routing



Dev machine

Any cloud

On-prem

Azure clusters

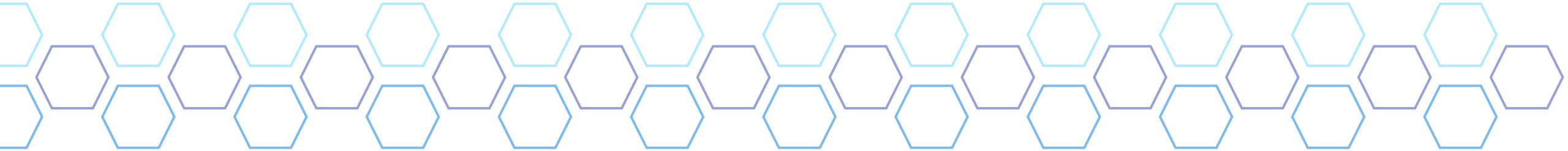
Mesh

Enables application modernization

- Deploy anything and everything in a container
- Bring your own network to connect to other systems
- Minimal code changes required
- No servers or VMs to manage with Service Fabric Mesh

NGINX

IIS



Service Fabric Mesh

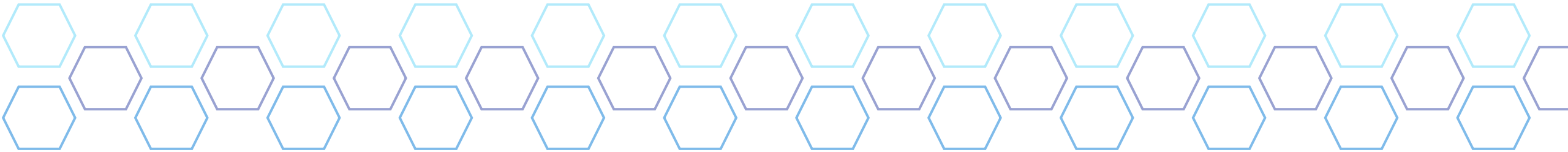
Ideal for Cloud-native applications

Any language, any framework

Libraries to integrate with your favorite languages

Easy H/A state storage with Reliable Collections

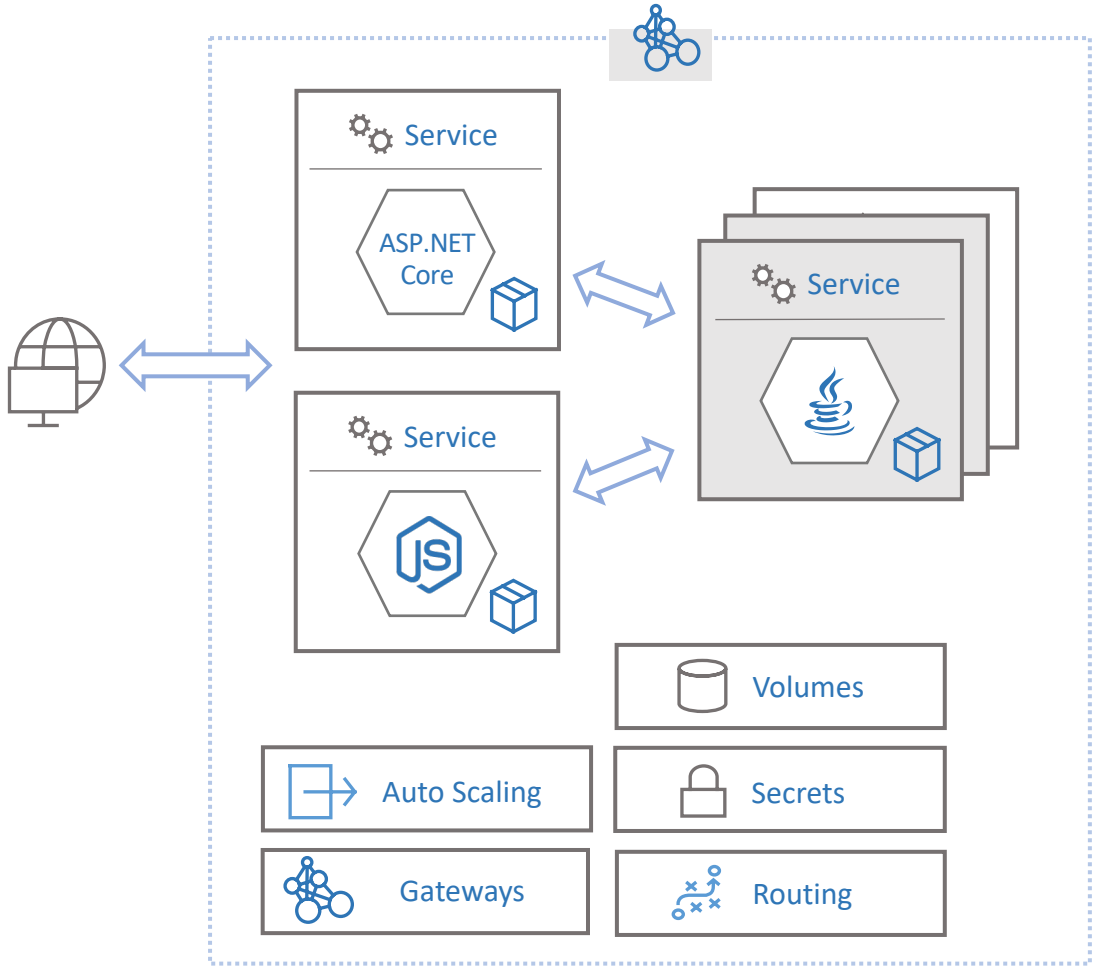
Intelligent traffic routing and connectivity



Service Fabric Mesh

Future of application development

- Polyglot applications described by resources
- In multi OS environments
- Deploy anything in a container
- Bring your own network to connect to your other services
- State management and data-aware containers with Reliable Collections and volume drives
- Connected through intelligent routing



Dev machine | Any cloud | On-prem | Azure clusters | Mesh

Microsoft Azure

Thanks!