Microsoft

Enhance your workload capabilities on AKS

Kendall Roden & Ray Kao, Cloud Native Global Blackbelt

Agenda

01 Welcome and Introduction

02 AKS developer considerations

03 Cloud Native themes

04 App Innovation

05 Review & next steps

Meet your instructors

@kendallroden

@RayKao

Webinar Series Overview

Configure your Cluster with Confidence

Optimize your Cluster for Security and Compliance

Today's session
Extend your Workload Capabilities

Cluster set-up review

	Compute and Infrastructure	Process	Networking	Observability
Webinar 1	· Managed Identity · Uptime SLA · System and User Node Pools	· ACR integration · Upgrade plan · Azure AD + RBAC	· Azure CNI · Network Policy with Calico	· Container monitoring & Log analytics
Webinar 2		· Azure Policy Integration · Azure Security Center · Pod Identity & Secret Store CSI Driver	· Private Cluster · AzFirewall integration · Service Mesh	· Intra-cluster scanning

[CloudNativeGBB/webinars(github.com)]

Kubernetes on Azure | Enterprise-grade by design

Development tools

- Visual Studio Code
- GitHub
- Bridge to Kubernetes
- Azure Container Registry
- Azure Monitor

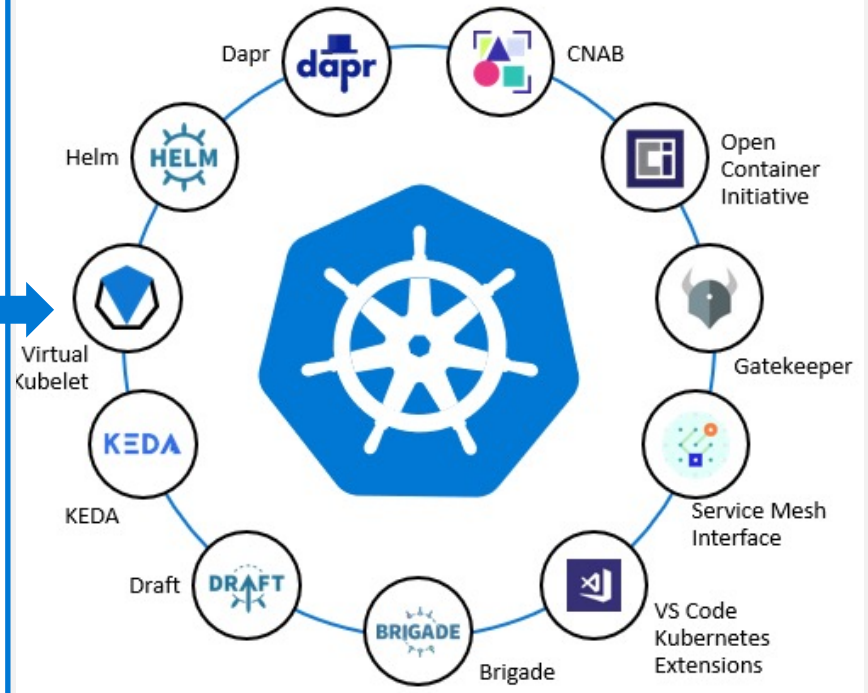
Platform

Active Directory | Azure Policy | Security Center | Key Vault | Azure Advisor

Azure Kubernetes Service | Azure Container Instances | Azure Red Hat OpenShift

Azure Arc
Management across environments

Community



Cloud Native Themes

Event-driven, distributed systems

- Dapr
- KEDA

Shifting left

- IDE integrations and tooling
- Bridge to K8s

Hybrid/Multi-cloud

- GitOps (Infra-as-Code)
- Azure Arc
- Self-hosted Azure services

Deep Integration

- Azure platform integration

AKS considerations for developers

Developer considerations

Pod resource requests and limits

- **Pod requests** => CPU and memory baseline need
- **Pod Limits** => max amount of CPU & memory a pod can use
- Dependency on **Resource Quotas** set at the ns level by operator

Limit Pod access to nodes

- Follow secure guidelines and avoid allowing privilege escalation

Avoid credential exposure

- Use a secure credential store i.e., Key Vault
- Use upstream projects i.e., Pod Identity & Secret Store CSI driver

Make use of developer tooling

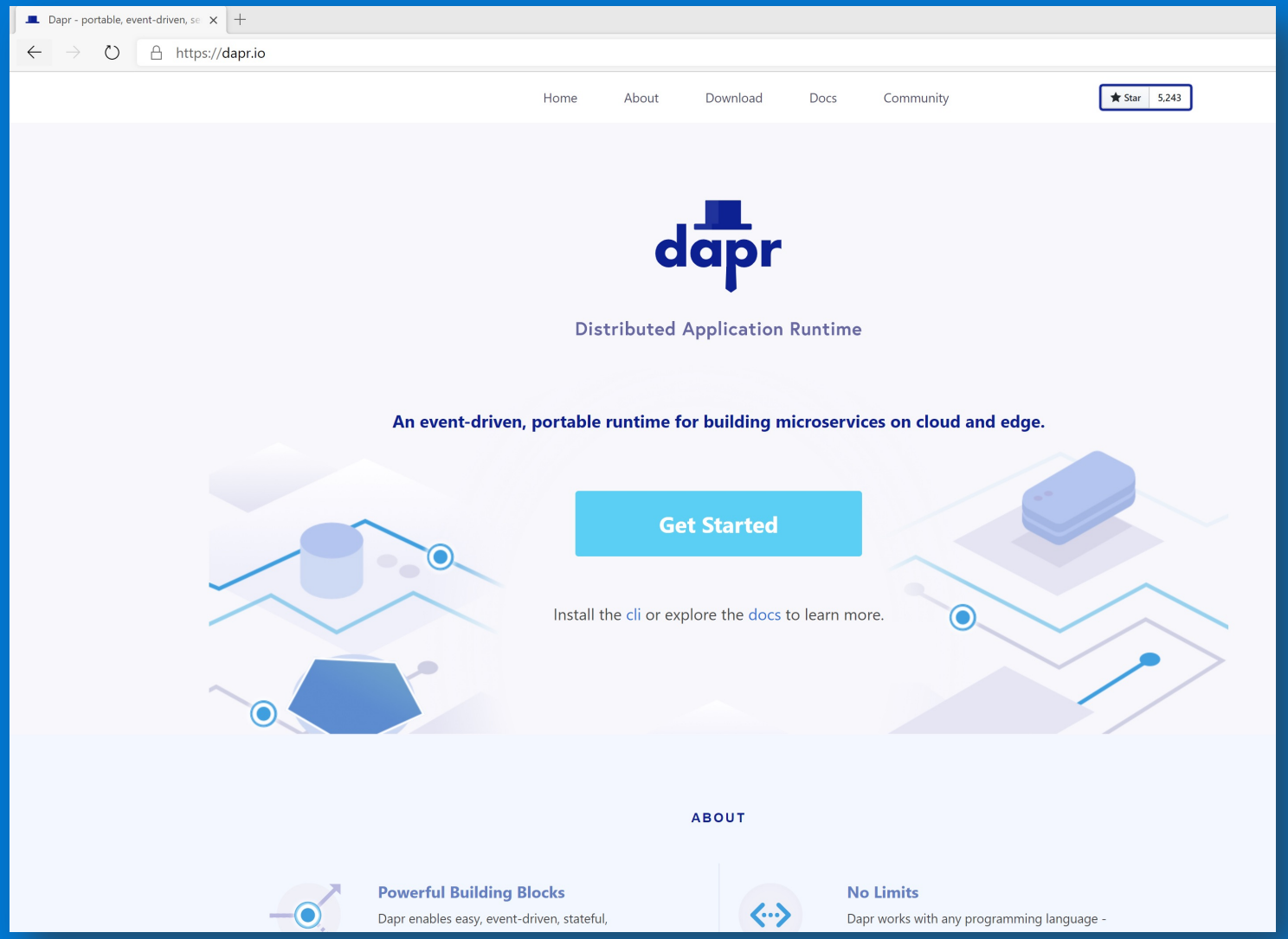
- VS Code extensions
- Bridge to K8s for debugging

Dapr

Distributed Application Runtime

Portable, event-driven, runtime for building distributed applications across cloud and edge

https://dapr.io



Microservice development challenges

- How do I **integrate with external systems** that my app has to react and respond to?
- How do I **create event driven apps** which reliably send events from one service to another?
- How do I **create long running, stateful services** that can recover from failures?
- How do I observe the calls and events between my services to **diagnose issues in production**?
- How do I **access secrets securely** from within my application?
- How do I **discover other services and call methods** on them?
- How do I **secure communication** between services?
- How do I **handle connection failures and build resiliency** into my application?
- How do I **prevent committing to a technology early** and have the flexibility to swap out an alternative based on project or environment changes?
- How do I write an app that **runs in multiple environments** without code changes?

Dapr Goals

Best-practices building blocks

Any language or framework

Consistent, portable, open APIs

Adopt standards

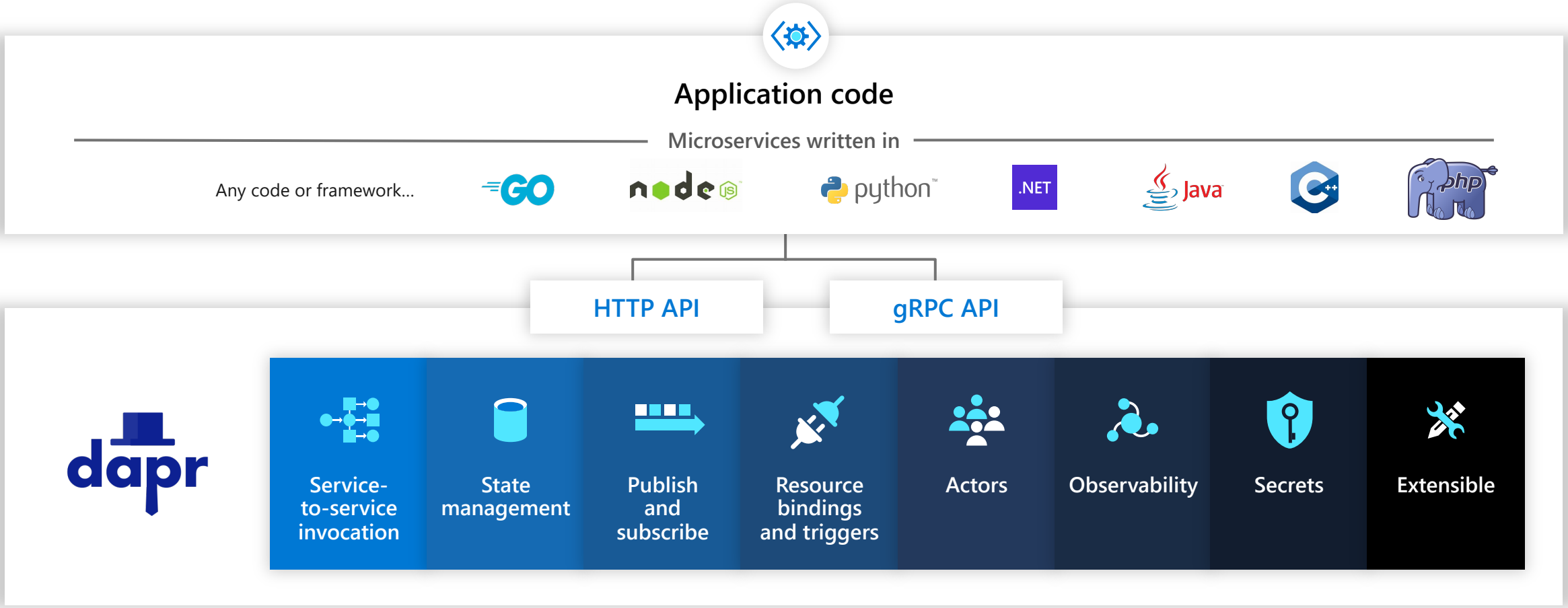
Extensible and pluggable components

Platform agnostic cloud + edge

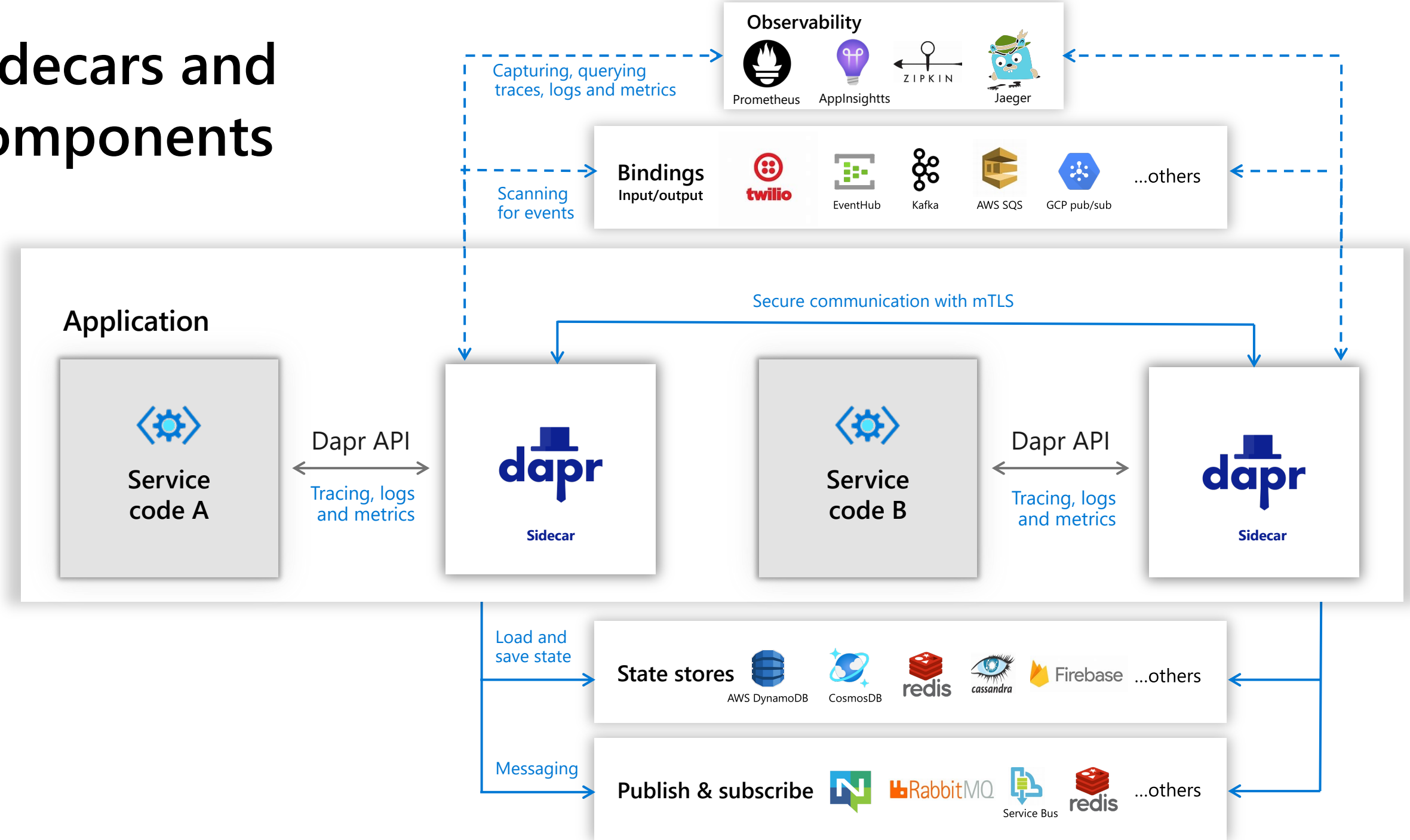
Community driven vendor neutral

Microservice building blocks

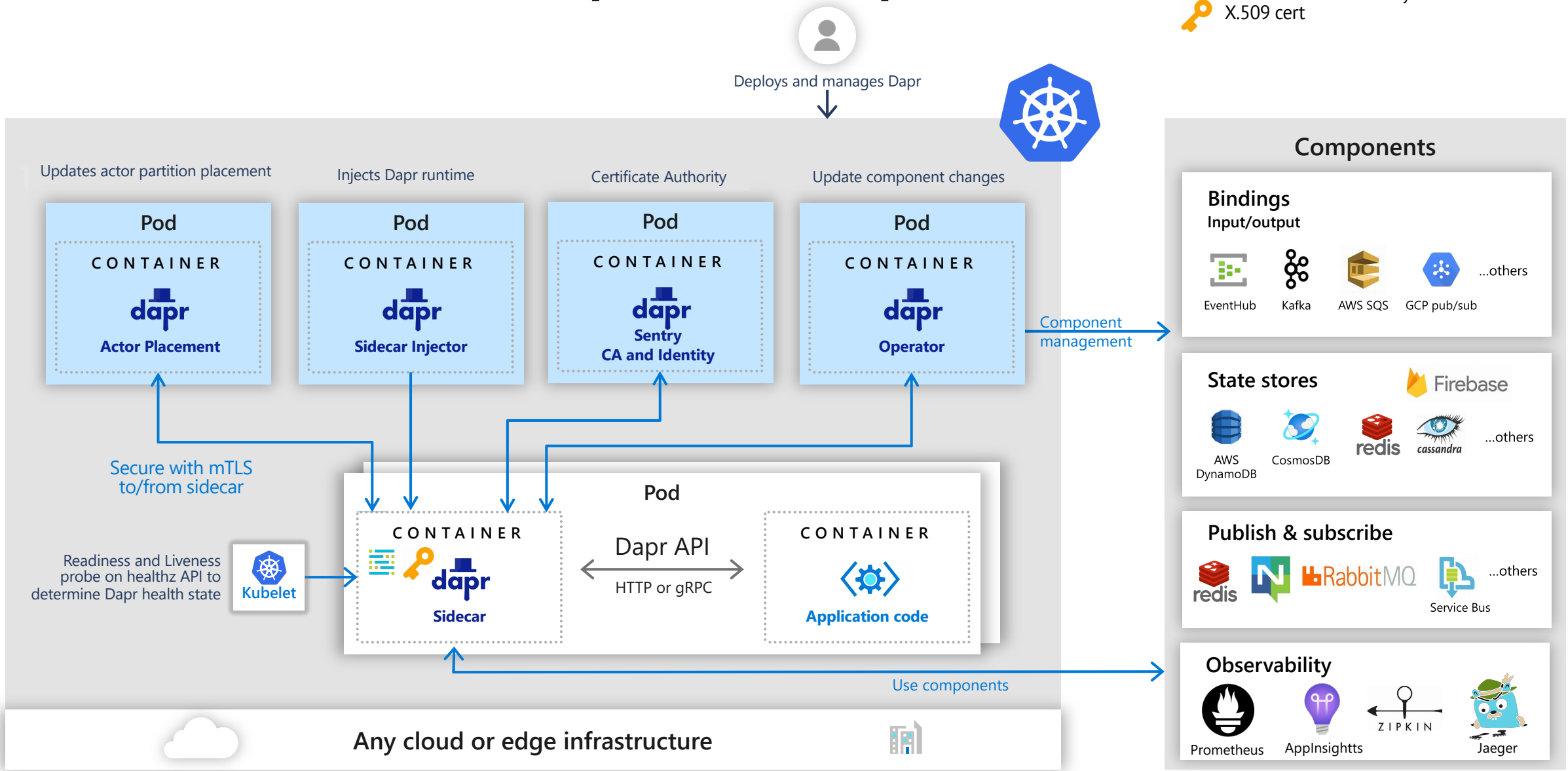
- Standard APIs accessed over http/gRPC protocols from user service code
  http://localhost:3500/v1.0/invoke/cart/method/neworder
  http://localhost:3500/v1.0/state/inventory/item67
- Runs as local "side car library" dynamically loaded at runtime for each service



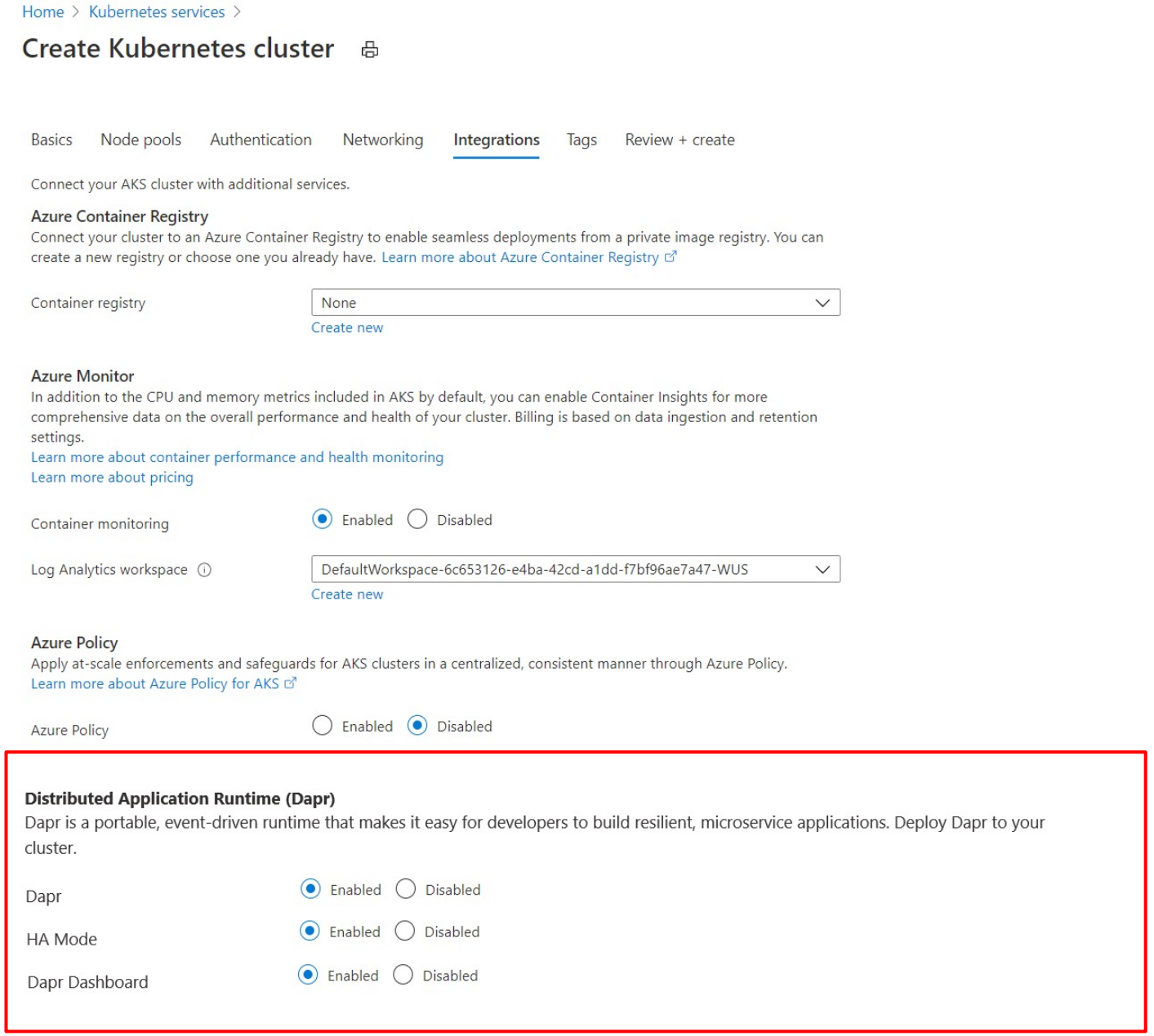
Sidecars and components



Kubernetes hosted Dapr control plane



Dapr
AKS Add-on

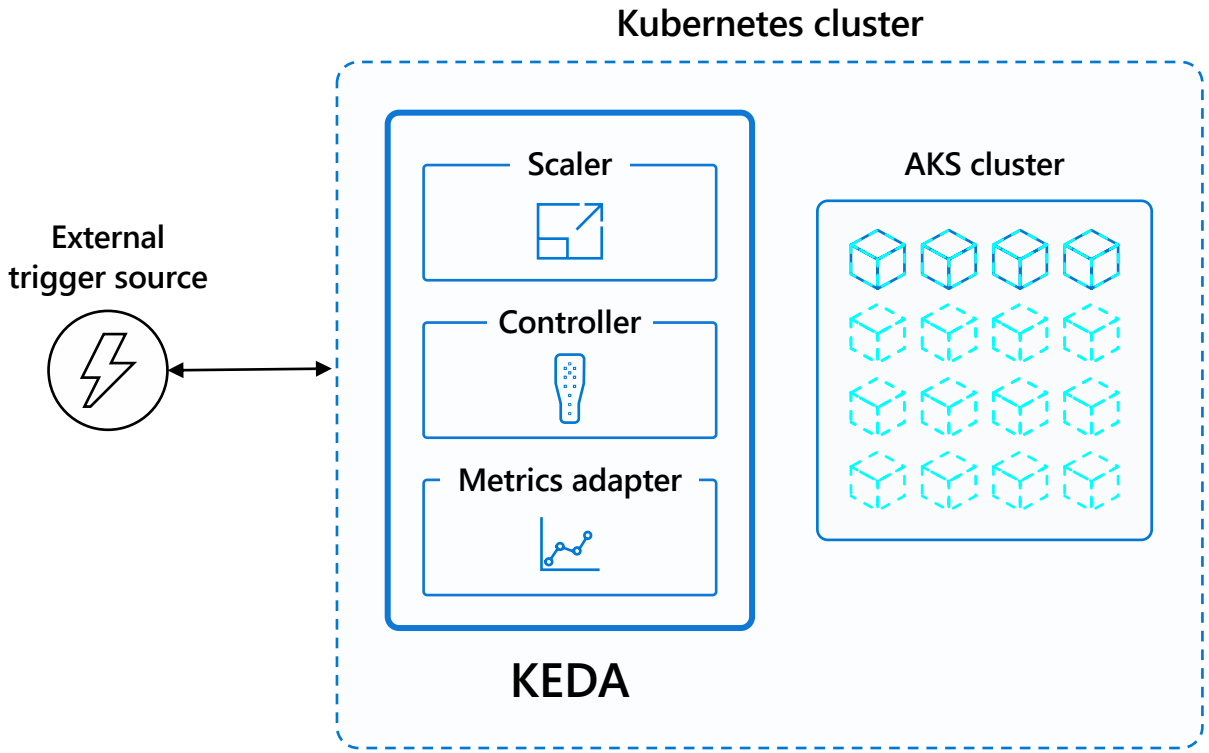


KEDA

Kubernetes-based event-driven auto-scaling (KEDA)

Open-source component jointly built by Microsoft and RedHat

- **Event-driven container creation & scaling**
  Allows containers to "scale to zero" until an event comes in
- **Native triggers support**
  Containers can consume events directly from the event source, instead of routing events through HTTP
- **Can be used in any Kubernetes service**
  This includes in the cloud (e.g., AKS, EKS, GKE, etc.) or on-premises with OpenShift



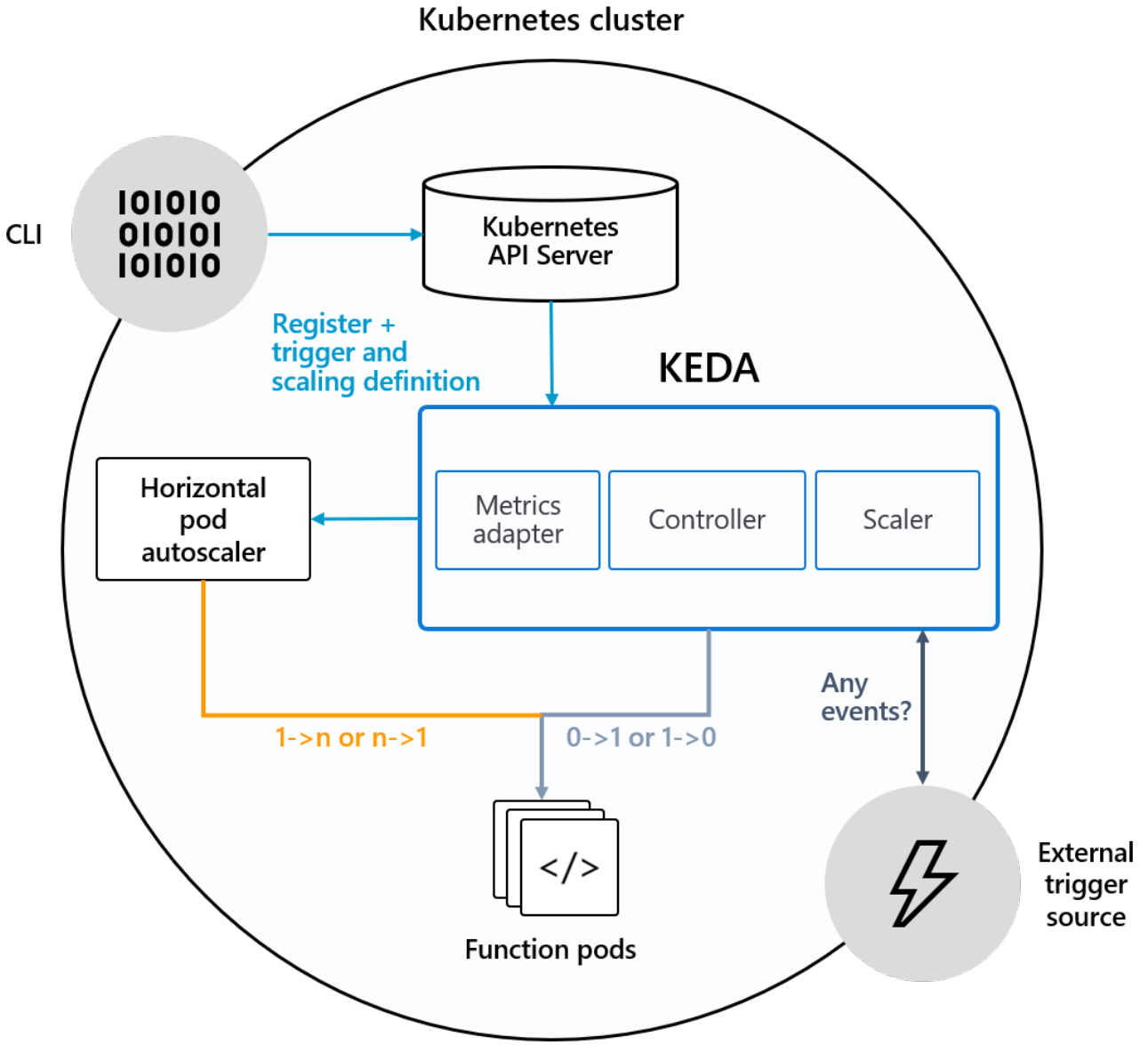
How KEDA works

1. Agent

KEDA activates and deactivates Kubernetes Deployments to scale to and from zero on no events (utilizes the keda-operator)

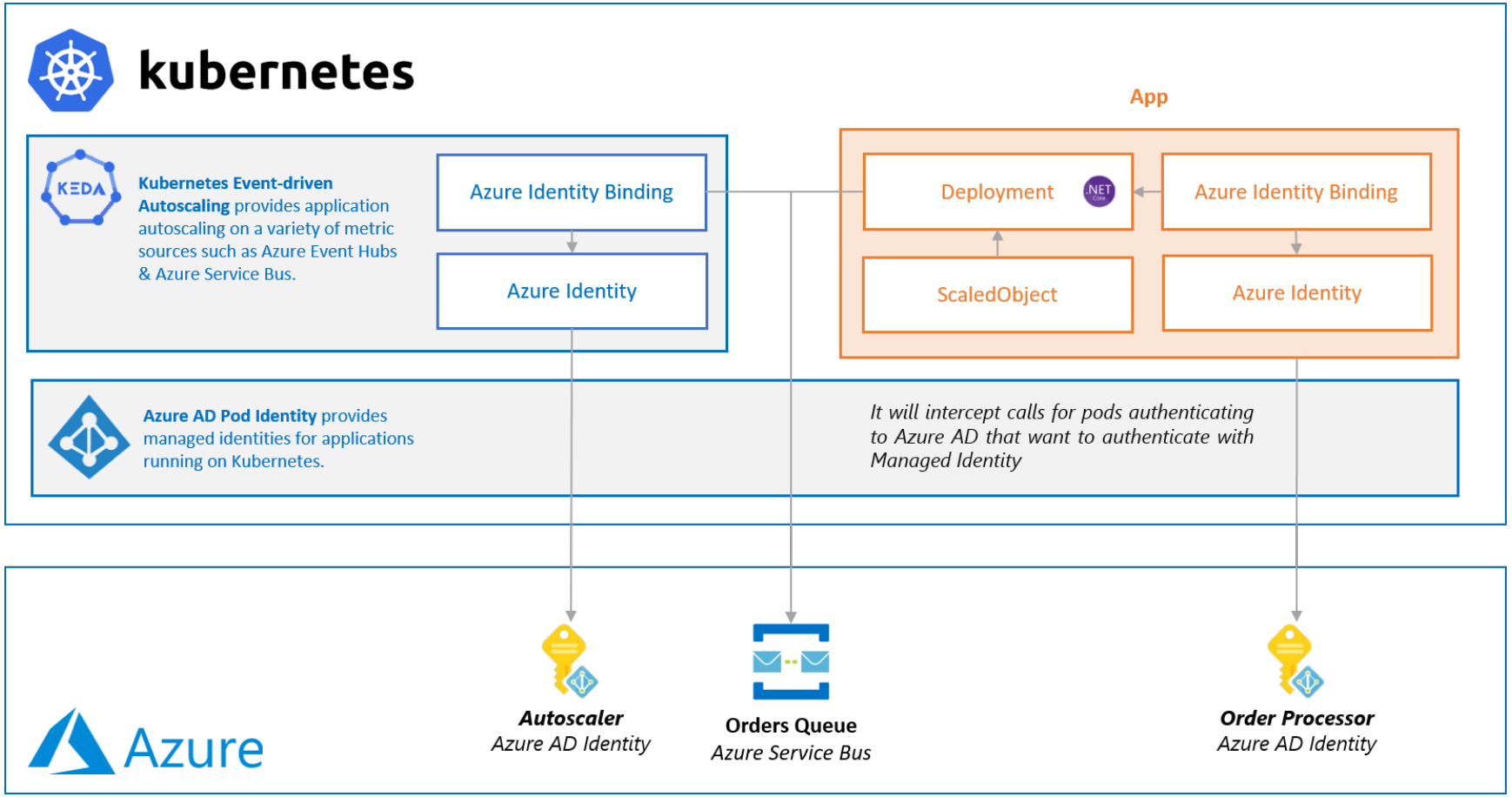
2. Metrics

KEDA acts as a Kubernetes metric server that exposes rich event data to the HPA to drive scale out. The deployment consumes the events directly from the source (utilizes the keda-operator-metrics-apiserver)



Reference: https://keda.sh/docs/2.0/concepts/#architecture

KEDA + Azure AD Pod Identity



Reference: https://github.com/kedacore/sample-dotnet-worker-servicebus-queue/

Why use KEDA if you are using Azure Functions?

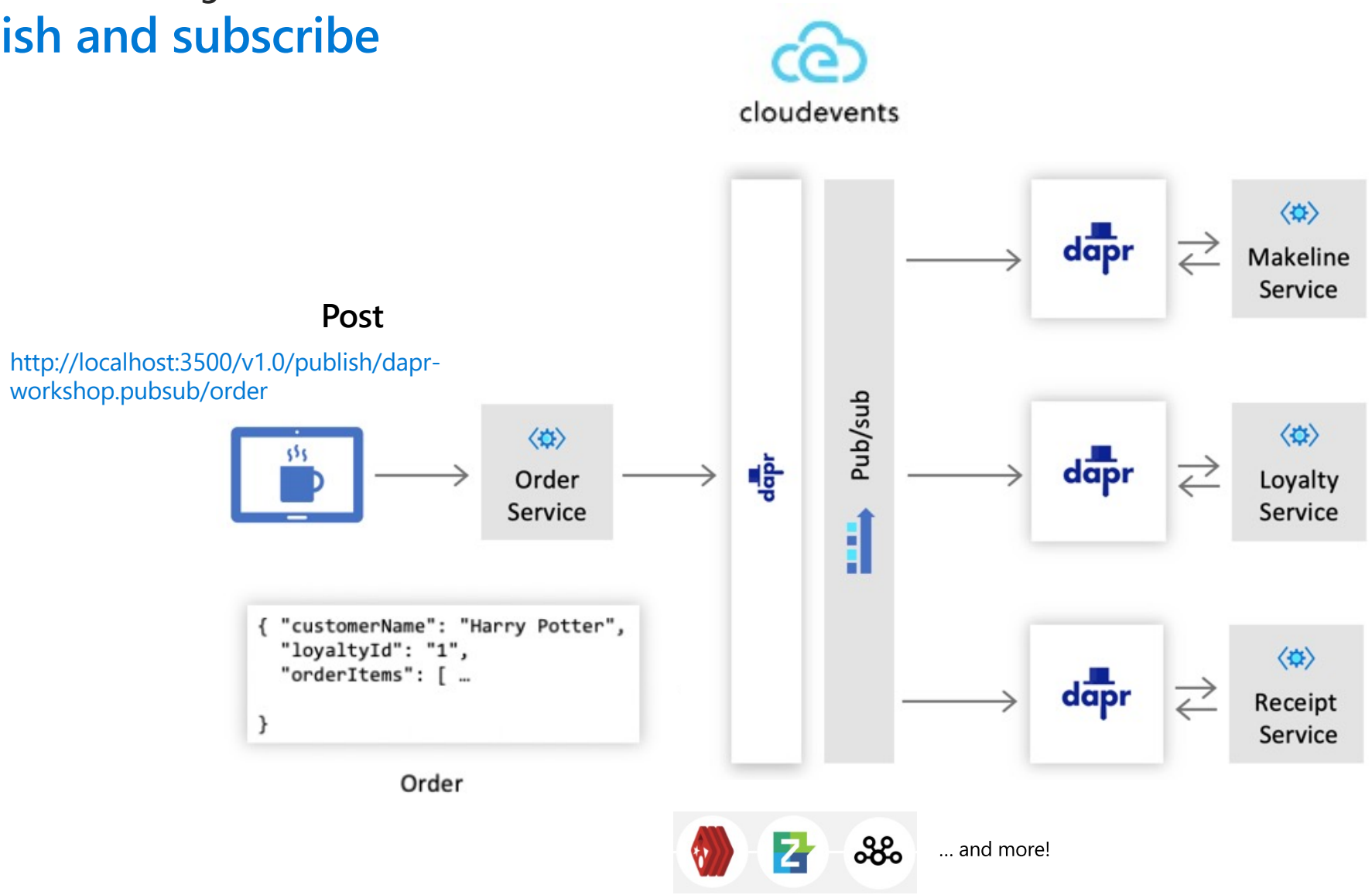
- Run functions on-premises (potentially in something like an ‘intelligent edge’ architecture)
- Run functions alongside other Kubernetes apps (maybe in a restricted network, app mesh, custom environment, etc.)
- Run functions outside of Azure (no vendor lock-in)
- Specific need for more control (GPU enabled compute clusters, policies, etc.)

Reference: https://keda.sh/docs/2.0/faq/

DEMO: Dapr + Keda in action

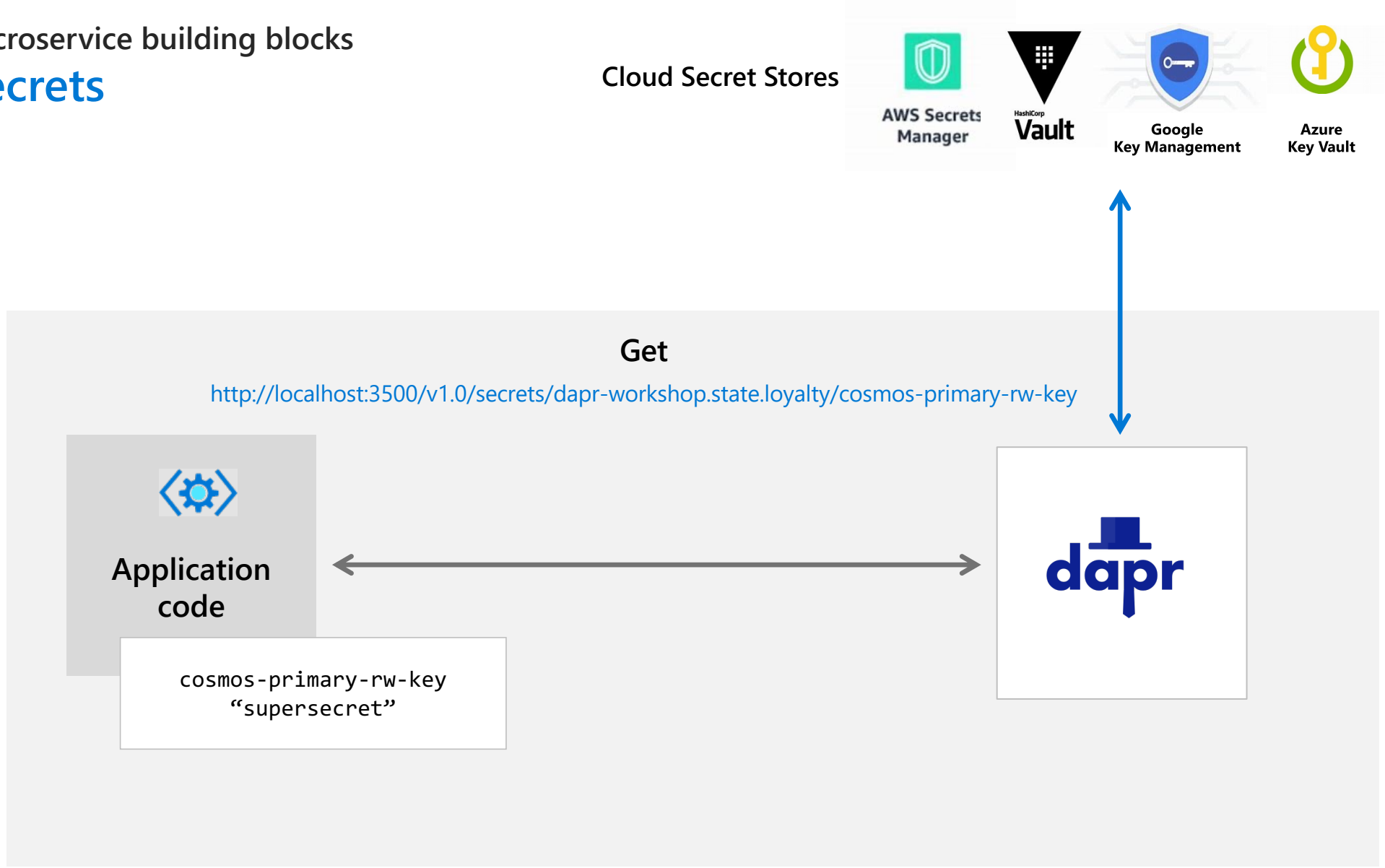
Microservice building blocks

Publish and subscribe



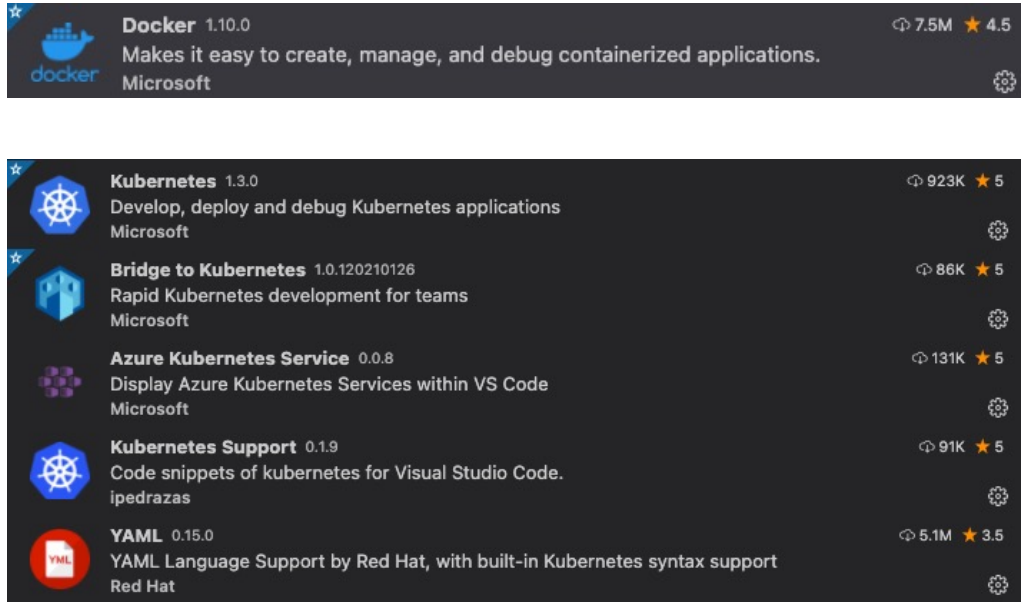
Microservice building blocks

Secrets



VS Code integrations

Local Tooling to get started with containers and k8s



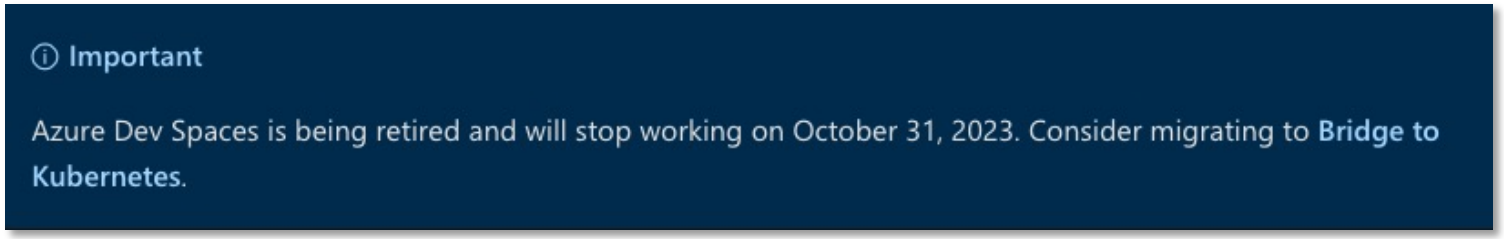
- `draft create` to containerize your app based on Draft packs.
- `draft up` to deploy your application to a Kubernetes dev sandbox, accessible via a public URL.
- Use a local editor to modify the application, with changes deployed to Kubernetes in seconds.

Bridge to K8s

Azure Dev Spaces

Debug and test code changes in the context of the larger application running in Kubernetes

While the value prop resonated, syncing code into the cluster came with friction revolving around operational complexities



Client-side tool integrated into VS and VS Code

Simplify Microservice Development

Eliminate the need to manually source, configure and compile external dependencies

Expedite inner loop

Sidestep operational complexities of building and deploying code into the cluster to test and debug

Debug and test end-to-end

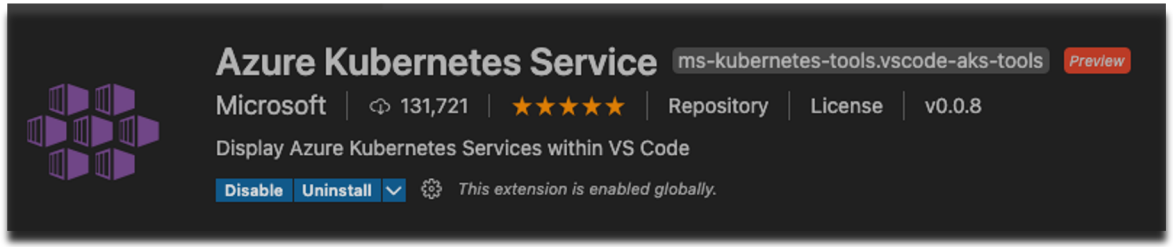
Route traffic from the cluster to development workstations and back seamlessly

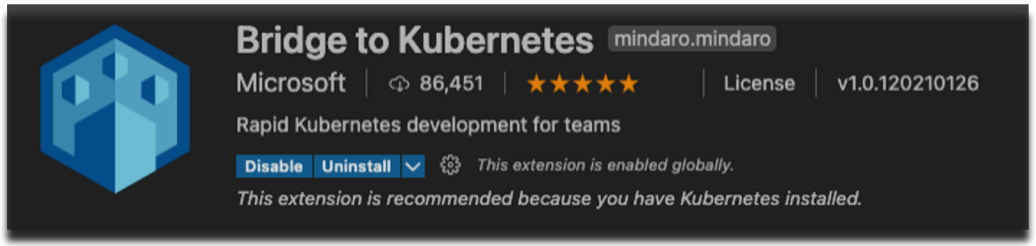
Work in isolation in a shared dev environment

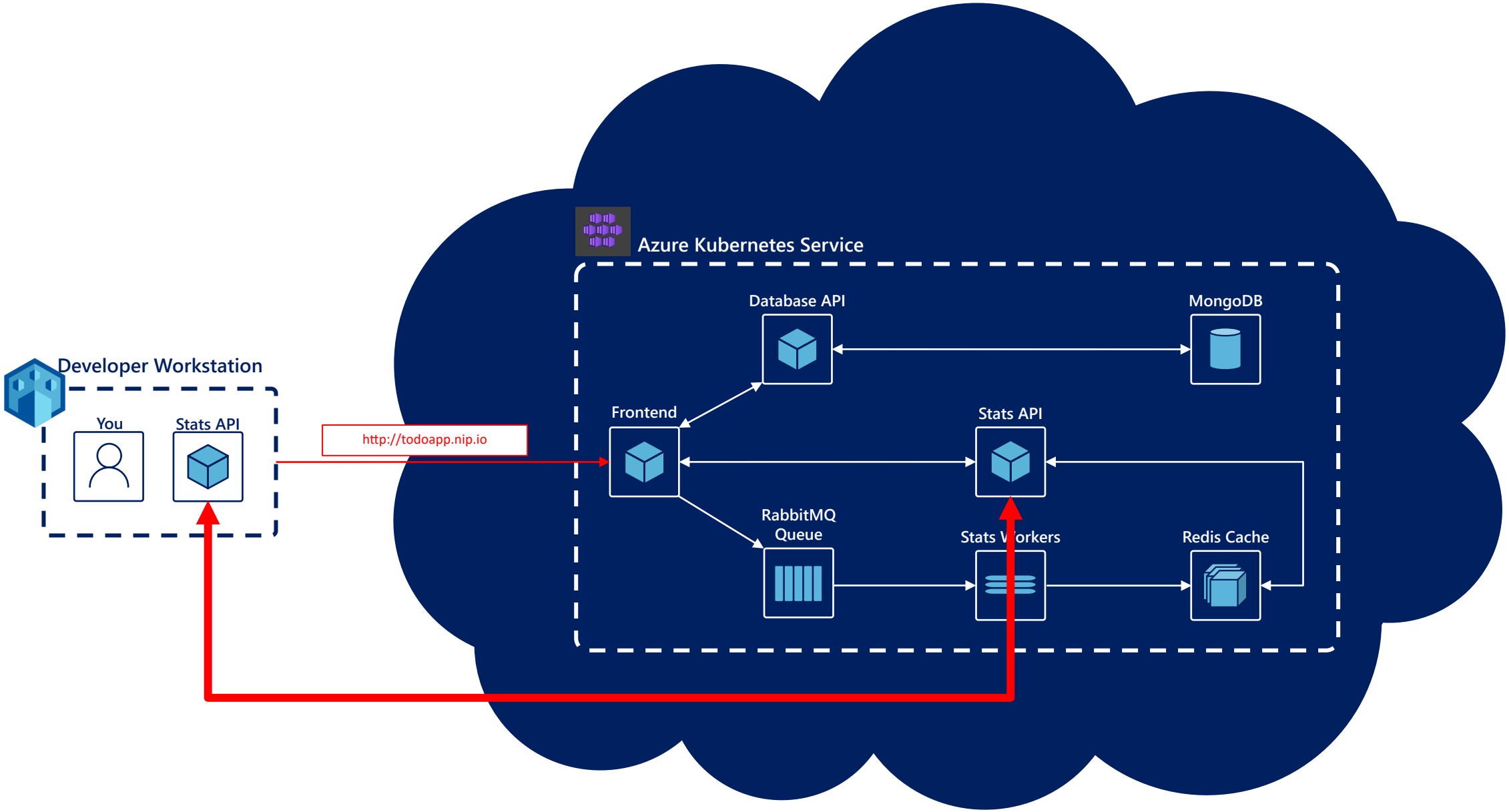
Work in a private “sandbox” environment by routing specific traffic locally.

Connect to any k8s cluster

Support for local and remote clusters



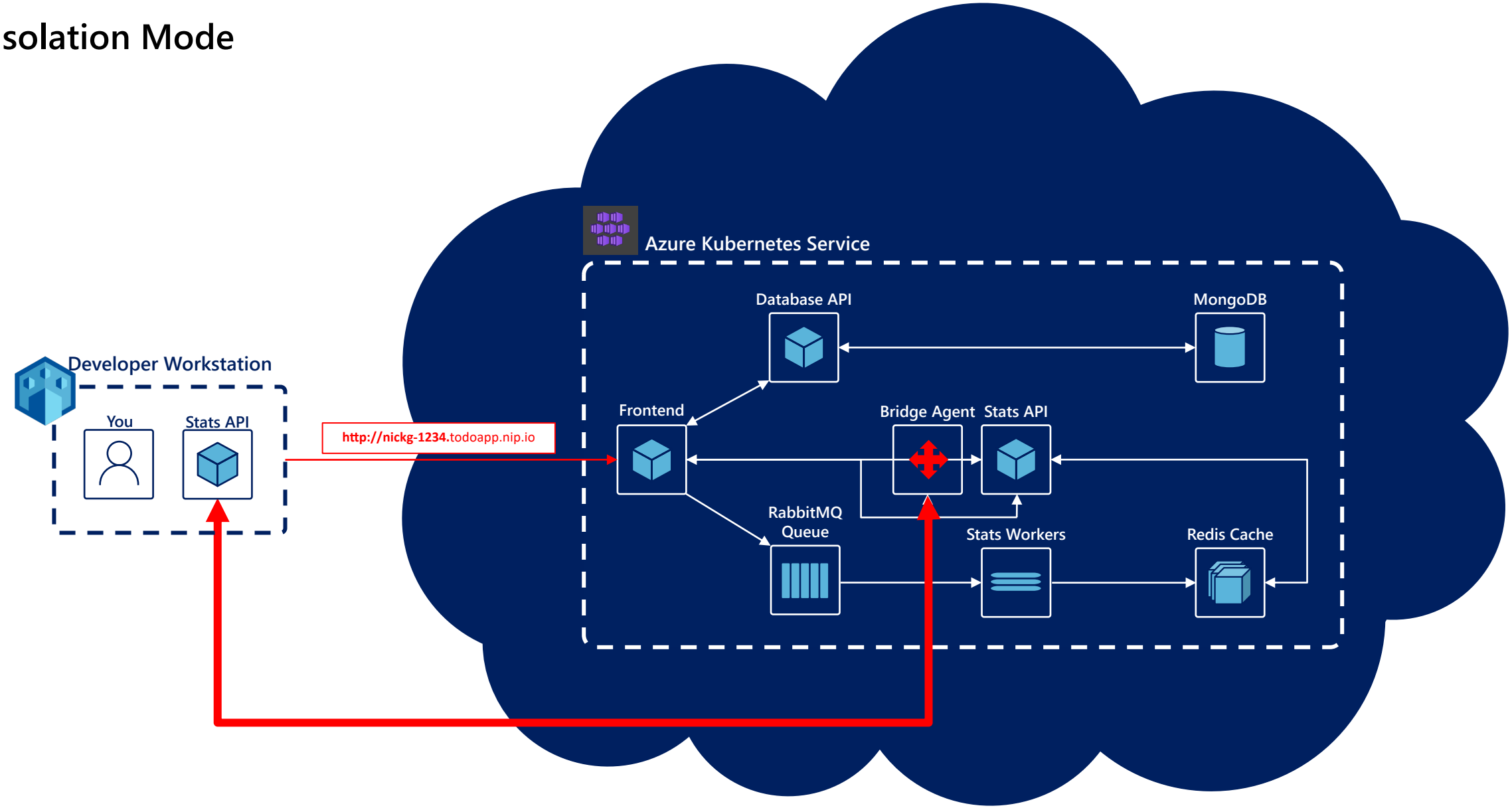



Azure Kubernetes Service
Database API
MongoDB
Frontend
Stats API
RabbitMQ
Queue
Stats Workers
Redis Cache
Developer Workstation
You
Stats API
http://todoapp.nip.io

Bridge to Kubernetes

- Simplifies microservice development
  - Eliminate the need to manually source, configure and compile external dependencies
- Develop applications faster
  - Sidestep operational complexities of building and deploying code into the cluster to test and debug
- Debug and test end-to-end
  - Route traffic from the cluster to development workstations and back seamlessly
- Work in isolation in a shared development environment
  - Work in a private “sandbox” environment by routing specific traffic locally

Isolation Mode



Helm/Kustomize and GitOps

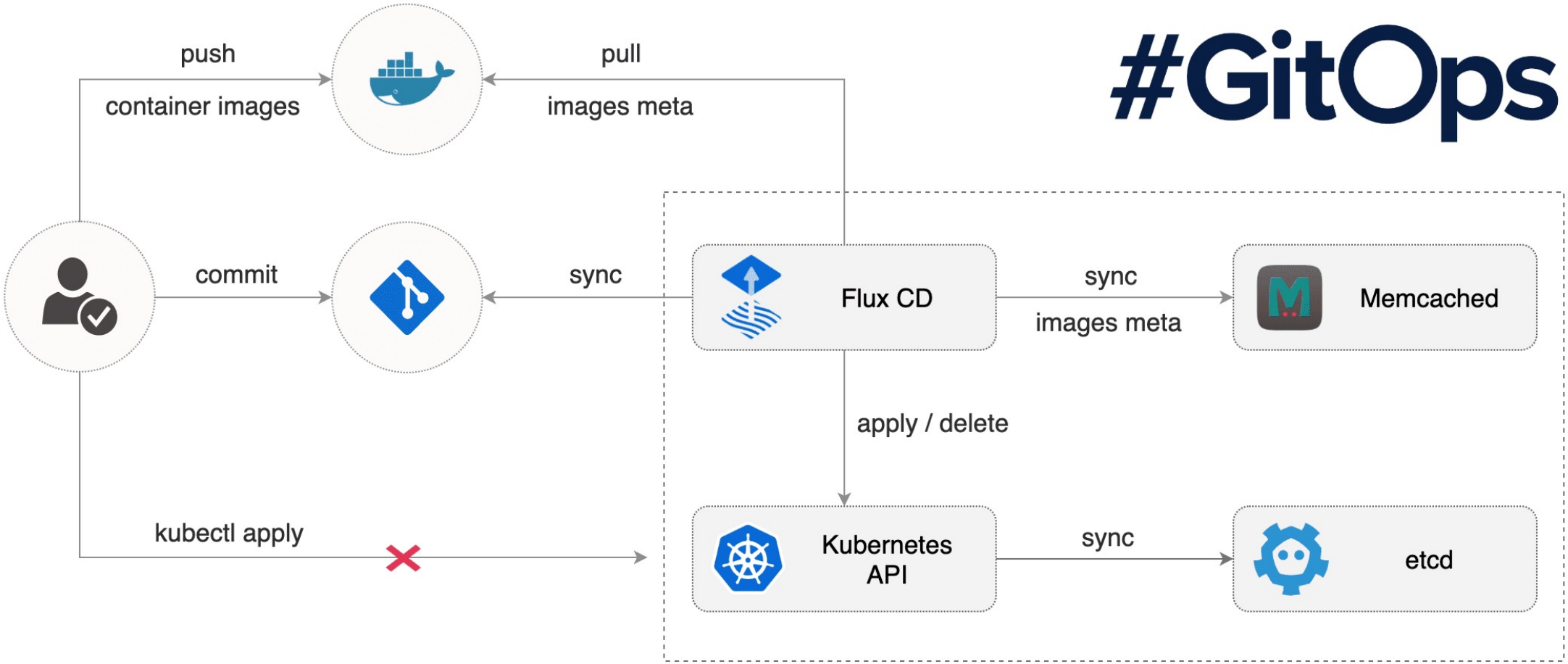
K8s Config/Package Management

- Helm
  - Defacto Pacakage Management for K8s (charts)
    - Think APT/RPM etc.
  - Powerful Templating Language
  - OCI Registry Compliant
    - Can save/distribute via container registry
  - Good for complex deployments or if you're managing a "solution"
  - Requires separate CLI tool (Helm)
- Kustomize
  - Simpler templating engine
  - Can define properties not found in original template – easier to extend/update
  - Best for simple deployment maintenance
  - Built into kubectl – however the feature drift can be noticeable
  - Should use separate cli to stay on latest features/updates

Push/Pull Models

- Push Model
  - CI/CD Pipeline has the authentication credentials
  - Pipeline directly operates against the cluster
  - Pipeline is the authoritative control center
  - Triggered by push/pr to source control
  - If it’s pushed to pipeline it moves forward – rollback requires reverting to previous commit
- Pull Model
  - Cluster Agent has the authentication credentials
  - Cluster Agent operates against the cluster
  - Cluster Agent is the authoritative control center
  - Checks source control periodically for updates/changes
  - Agents can prevent updates and do rollback automatically

From Flux:



https://github.com/fluxcd/flux

Tooling

- Flux
- Argo
- Azure Arc for Kubernetes

Azure Arc

Customer environments are increasingly complex

10s - 1,000s of apps


VMs


Databases


Containers


Serverless

Diverse infrastructure


Datacenters


Hosters


Branch offices


OEM hardware


IoT devices


Edge

Multi-cloud

Azure Arc

Bring Azure services and management to any infrastructure

Run Azure data services anywhere

Extend Azure management across your environments

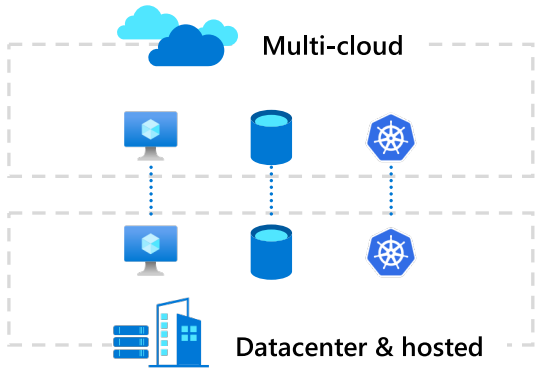
Adopt cloud practices on-premises

Implement Azure security anywhere

Azure Arc is a set of technologies that extends Azure management and enables Azure services to run across on-premises, multi-cloud, and edge.

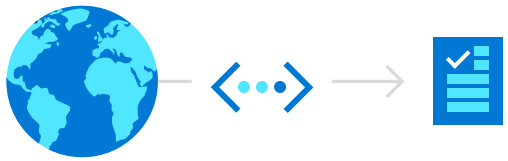
Azure Arc

Customer use cases



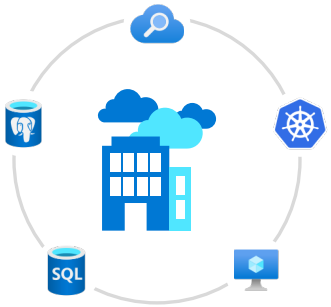
Organize and govern across environments

Get Kubernetes clusters and servers that are sprawling across clouds, datacenters and edge under control by centrally organizing and governing from a single place.



At-scale Kubernetes app management

Deploy and manage Kubernetes applications at scale across environments using DevOps techniques. Ensure that applications are deployed and configured consistently from source control, at scale.



Run data services anywhere

Deploy and manage data services where you need it for latency or compliance reasons. Always use the most current technology and seamlessly manage and secure your data assets across on-premises, clouds and edge.

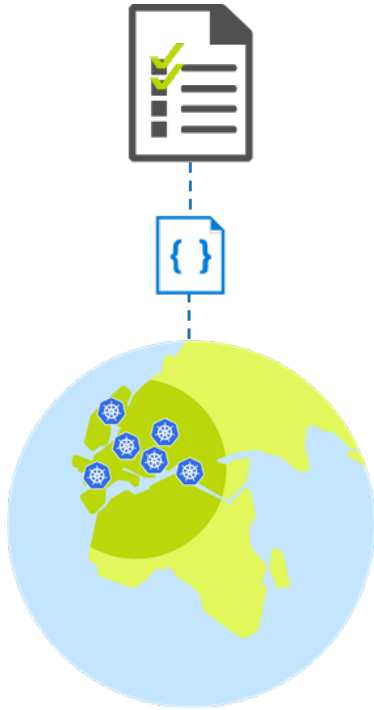
Customer scenario

At-scale Kubernetes App management

Overview

A retailer with 100s of stores would like to move all in-store applications to containers running on a K8s clusters.

They are faced with the challenge of how to uniformly deploy, configure and manage their containerized applications across multiple locations.



Business requirements

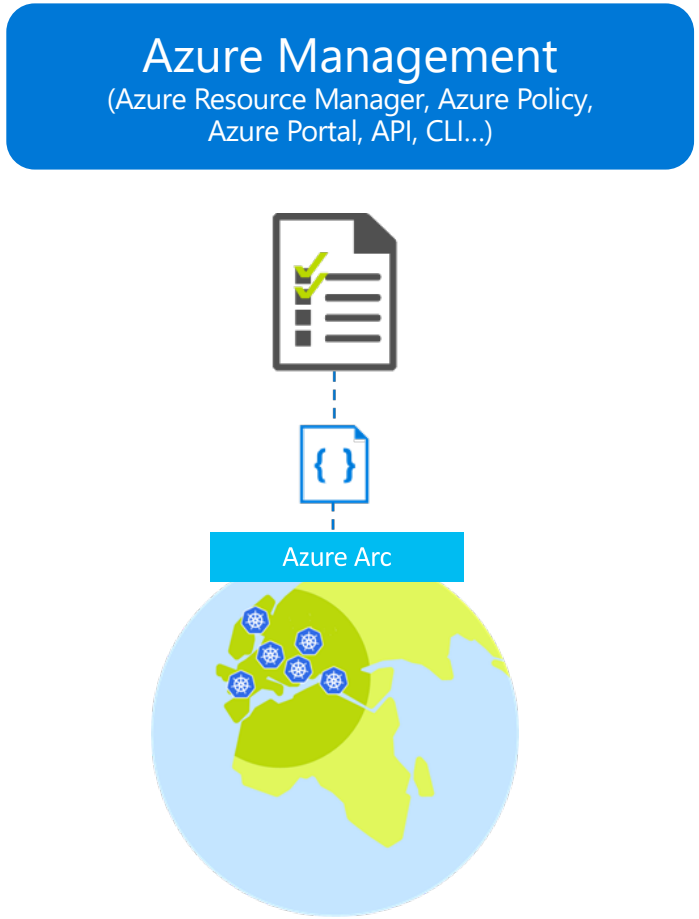
- Bootstrap a new store to fully run with the applications and configuration that this store requires
- Enable IT to apply and monitor at scale governance across all stores
- Monitor the state of applications and configuration in all stores
- Integrate DevOps and Safe Deployment Practices for applications running in stores
- Allow region/store IT to monitor and troubleshoot configuration issues for their stores

Customer scenario

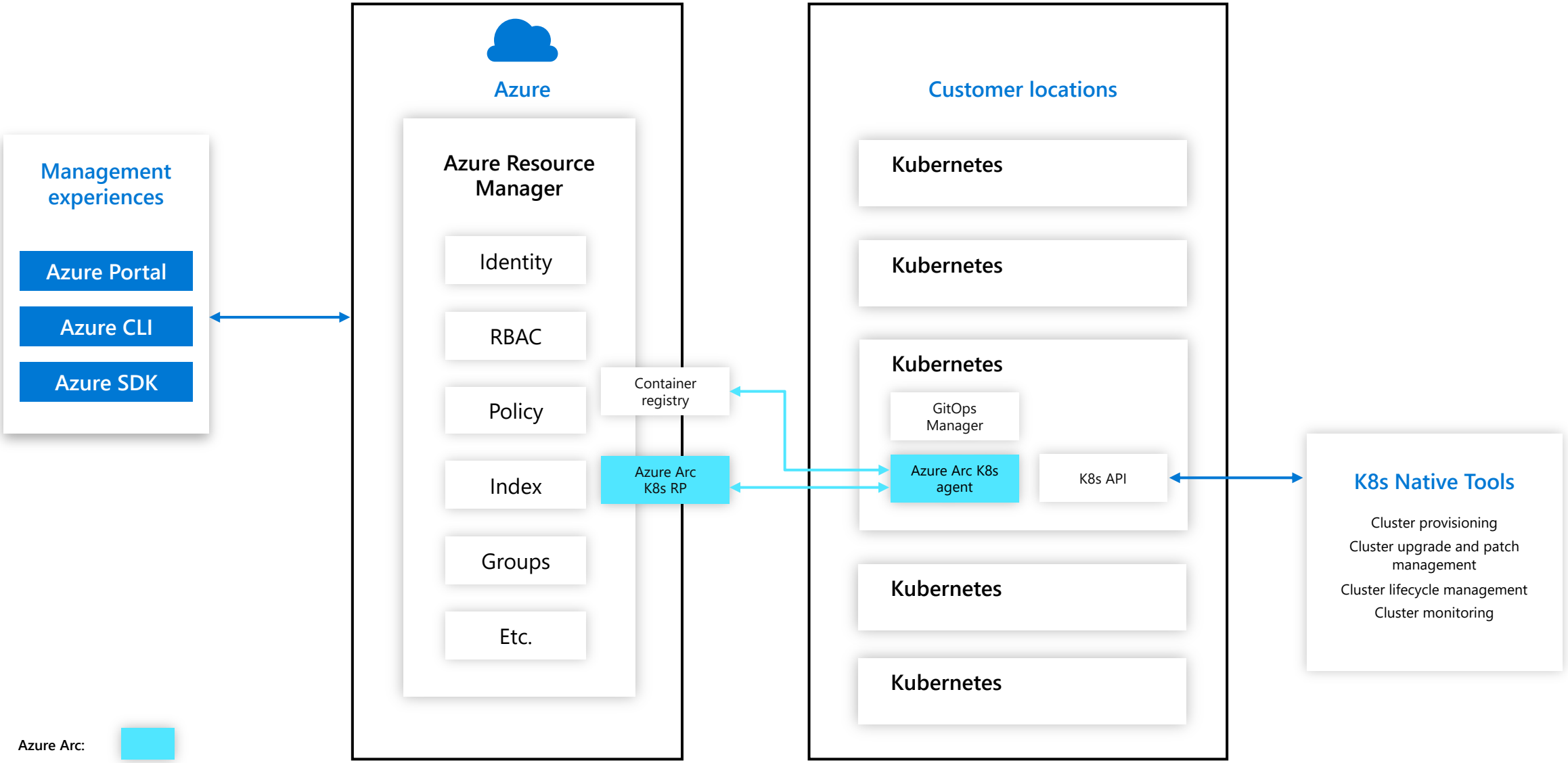
Solution

Key benefits from Azure Arc

- Asset organization and inventory with a unified view in the Azure Portal across all locations
- At scale configuration and deployment based on subscription, resource groups, and tags
- GitOps-based model for deploying configuration as code to one or many clusters
- Application deployment and update at scale
- Source control based Safe Deployment Procedures when rolling new applications and configurations
- Developer tooling agnostic—use the tools they want



Azure Arc for Kubernetes



Key takeaways for Azure Arc for Kubernetes

Central management

Cluster organization and inventory with a unified view in the Azure Portal across all locations

At-scale control

At-scale configuration and workload management

GitOps

GitOps model for configuration and app deployment from single sources of truth to one or many clusters

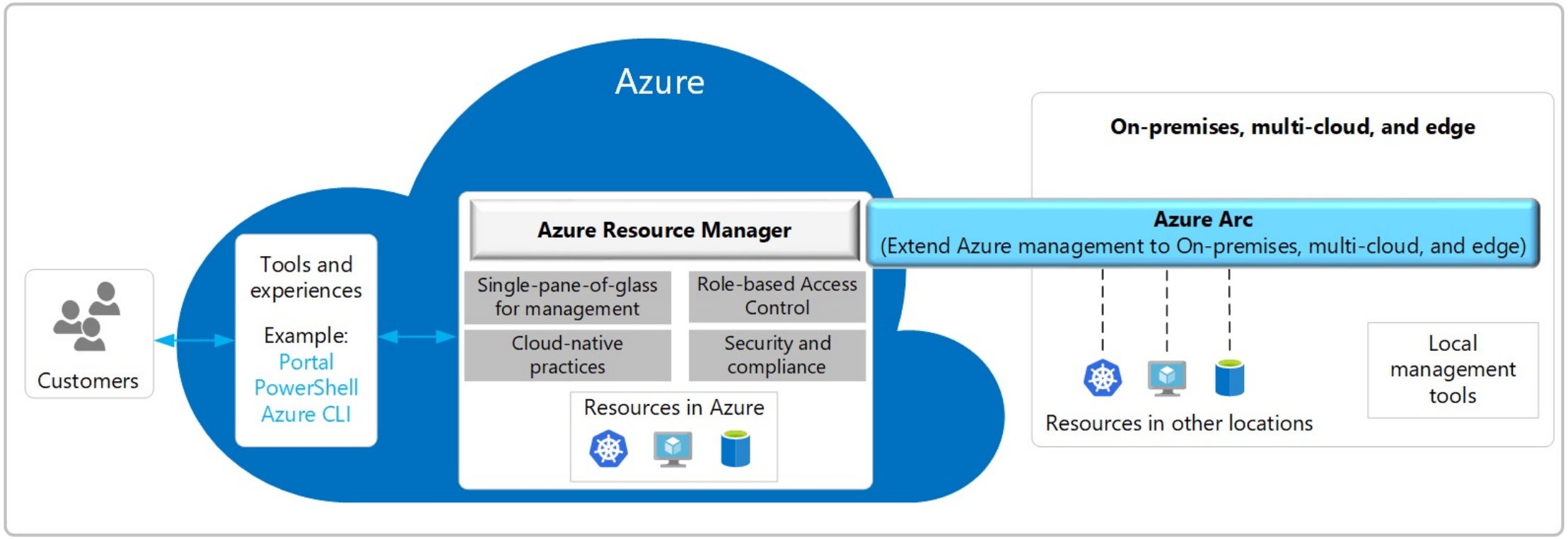
Integrates with your dev tooling and CI pipeline

Azure management

Azure management capabilities brought to all clusters for consistent management

Dapr & Azure: ARC for Kubernetes Dapr Extension

- Deploy and upgrade Dapr to Kubernetes clusters via Azure resource manager
- Dapr extension publish to ARC extension repository. Uses Helm under the covers to deploy to Kubernetes



Azure Platform Integrations

Kubernetes and Beyond

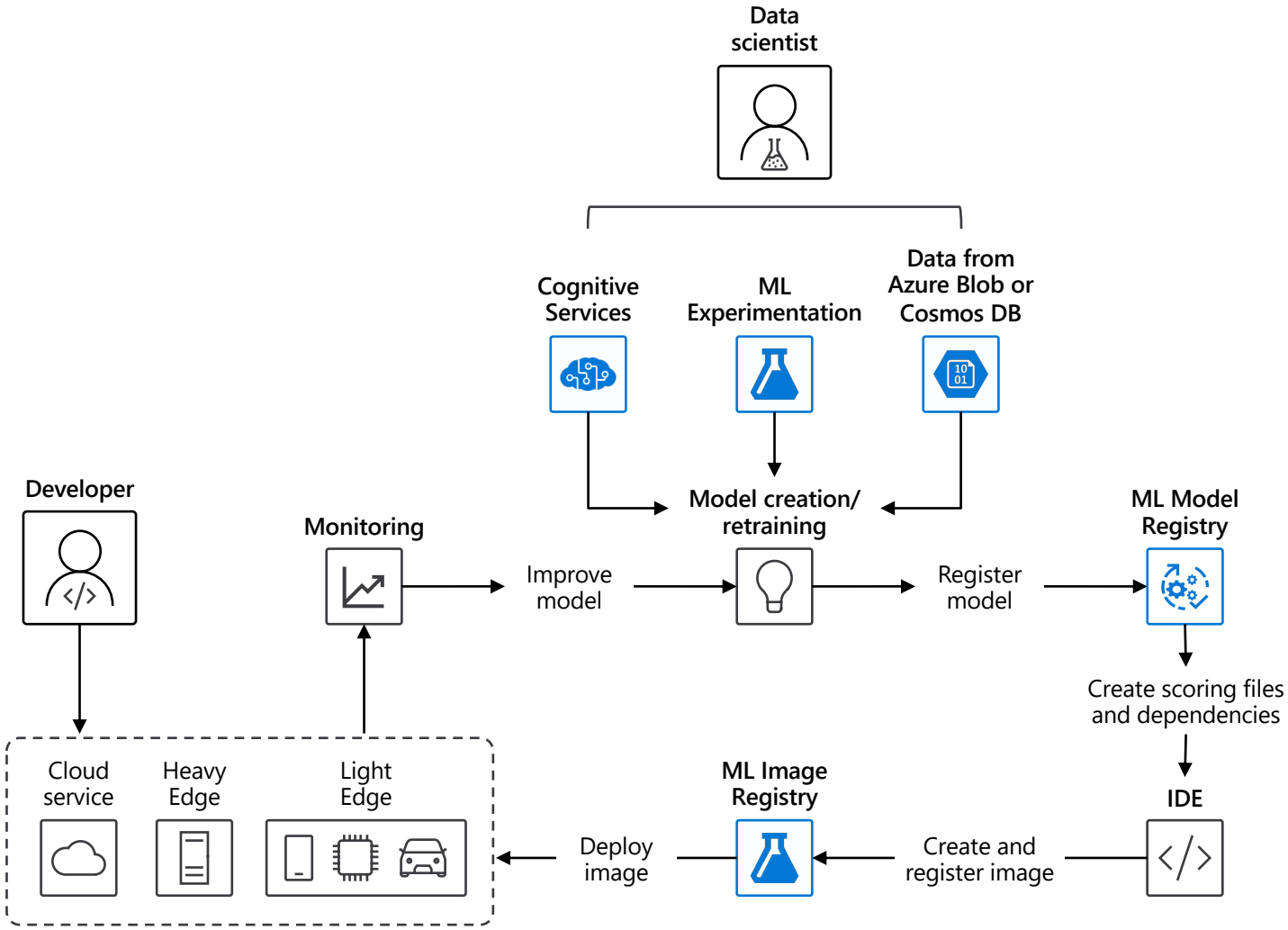
Native Integrations

- **Azure CNI**
- **Azure CSI**
  - Azure Disk
  - Azure Files
  - Azure Key Vault

Data scientist in a box

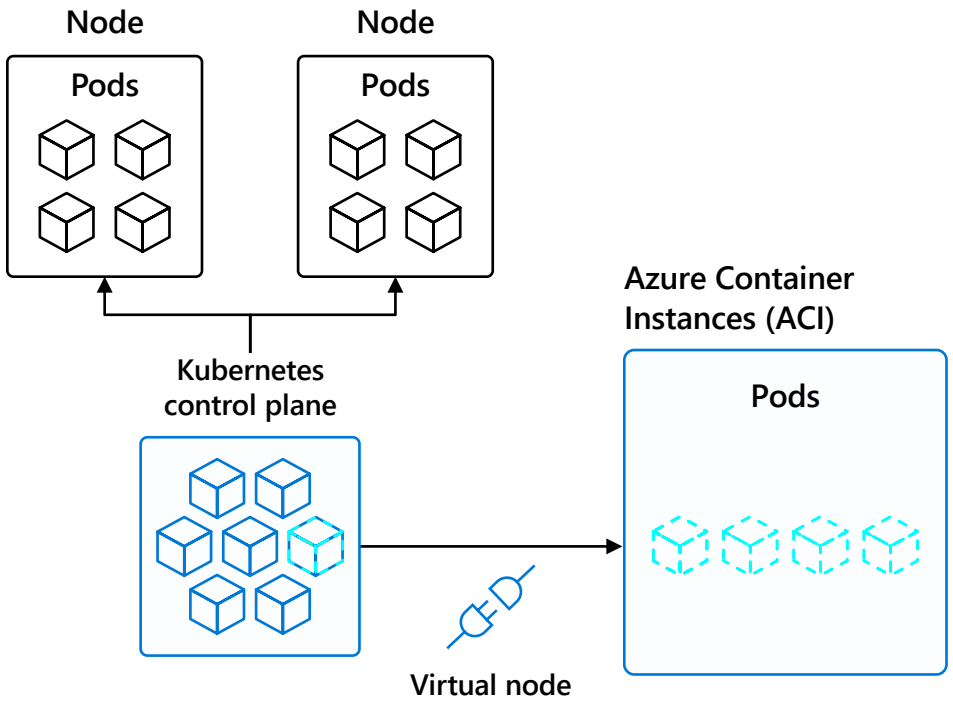
- Quick deployment and high availability
- Low latency data processing
- Consistent environment across test, control and production

How to deploy



Serverless Kubernetes using AKS virtual nodes

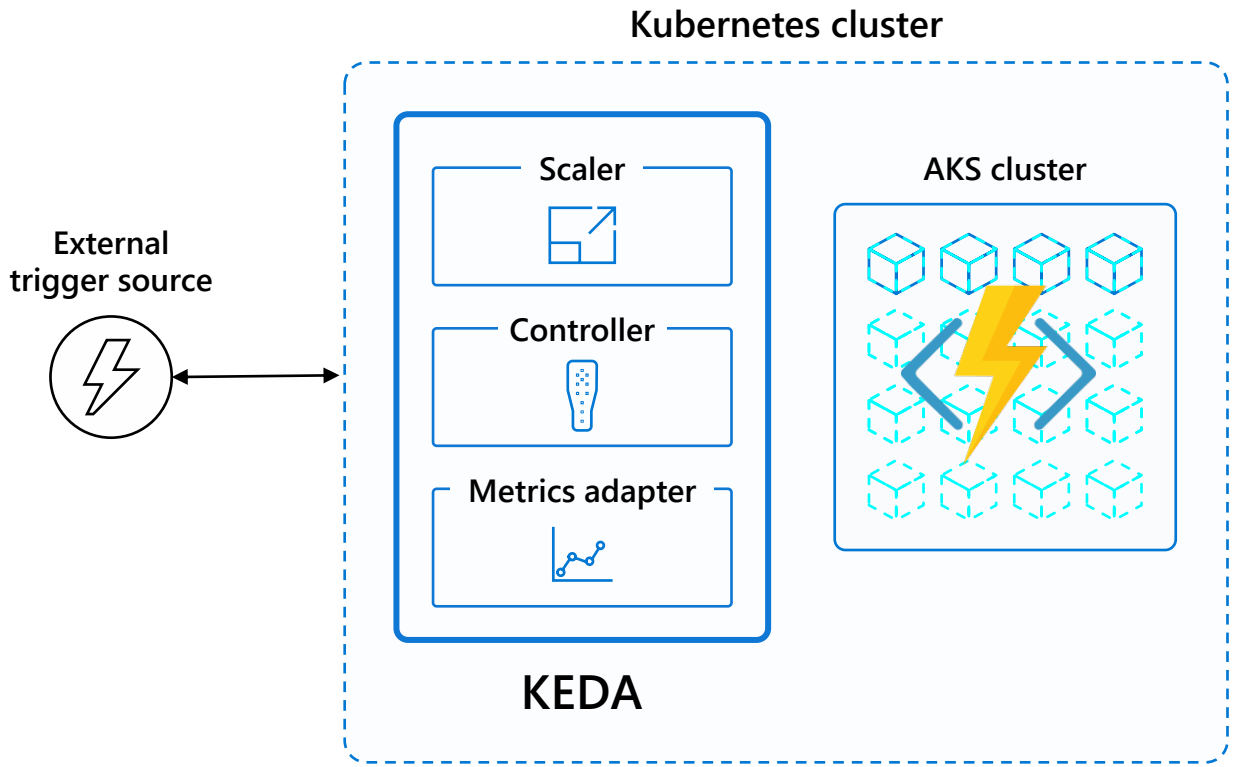
- Elastically provision compute capacity in seconds
- No infrastructure to manage
- Built on open sourced Virtual Kubelet technology, donated to the Cloud Native Computing Foundation (CNCF)



Containerization of Azure Services

- APIm (Self Hosted)
- Logic Apps
- Azure Functions
- Azure “Everywhere”
- More…

- Function runtime as a container available in multiple languages
- Write the same code run anywhere
- Use KEDA to auto scale the application
- Scale on Prometheus Metrics or external message queue/event (http trigger) as expected



Dapr and Azure Functions

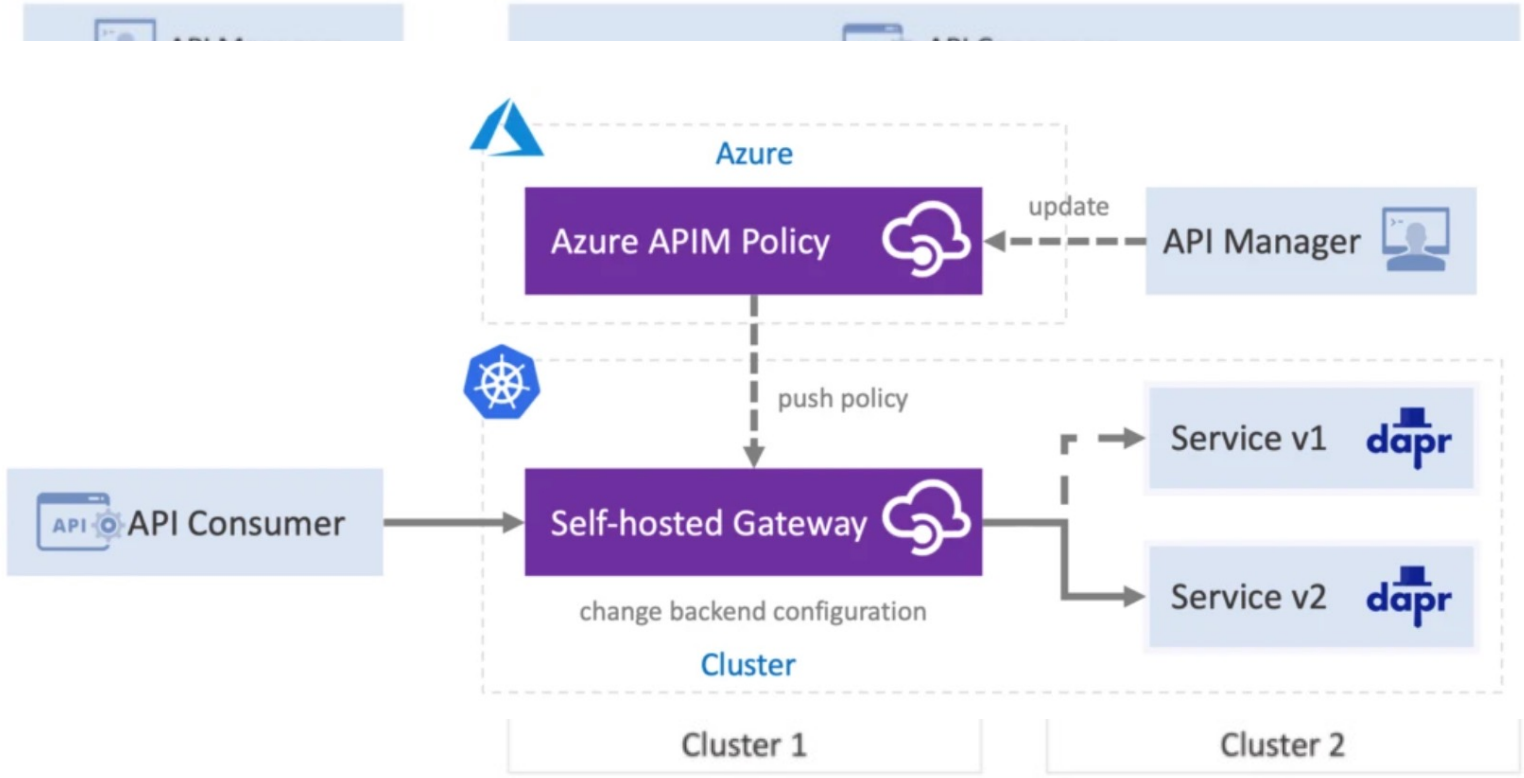
- Building an Azure Functions Dapr extension for both Triggers & Bindings
- Support for svc invocation, state, secrets, bindings & pub/sub
- Interact with Dapr capabilities in Kubernetes, IoT Edge and self-hosted mode

```
[FunctionName("StateInputBinding")]
public static async Task<IActionResult> Run(
    [HttpTrigger(AuthorizationLevel.Function, "get", Route = "state/{key}")] HttpRequest req,
    [DaprState(StateStore = "statestore", Key = "{key}")] string state,
    ILogger log)
{
    log.LogInformation("C# HTTP trigger function processed a request.");

    return new OkObjectResult(state);
}
```

Dapr + API Management

- Self-hosted gateway integration
- Svc Invocation, Pub/sub and resource binding
- Use APIM's built-in policies
- Svc discovery, retries, tracing, etc all from Dapr
- Expose your Dapr APIs in a single interface



Additional Q&A

Join us for AKS Office Hours!

Hosted by the Cloud Native GBB Team every other Thursday from 11-12 CST!

- Provide AKS customers with updates pertaining to AKS and the Cloud Native Ecosystem
- Host a short talk and/or demo on Cloud Native technologies related to Kubernetes and AKS
- Collect feedback from customers on issues, blockers, use cases, and questions related to AKS

Other Resources

AKS Public Office Hours

https://aka.ms/akspublicofficehours

Microsoft Cloud Native GBB YouTube Channel:

https://www.youtube.com/channel/UCvdABD6_HuCG_to6kVprdjQ

Kubernetes Learning Path:

https://azure.microsoft.com/en-us/resources/kubernetes-learning-path/

AKS Checklist:

https://www.the-aks-checklist.com

AKS Solution Journey

https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-start-here

AKS Workshop (MS Learn):

https://docs.microsoft.com/en-us/learn/modules/aks-workshop/

GBB AKS Secure Workshop:

https://github.com/CloudNativeGBB/aks-secure-workshop