Microsoft

CloudLand
www.cloudland.org

Implementing distributed applications with Dapr and Azure Container Apps

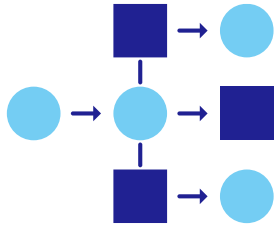
Ricardo Niepel
Senior Cloud Solution Architect – Azure App Innovation
ricardo.niepel@microsoft.com

RicardoNiepel

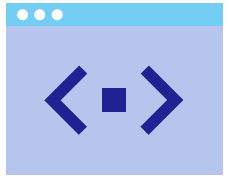
@RicardoNiepel

Modern microservice architectures

Deploying scale-out apps for flexibility, cost, and efficiency

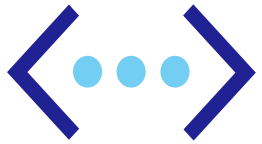


Developing resilient, scalable, microservice-based apps that interact with services



Focusing on building applications, not infrastructure

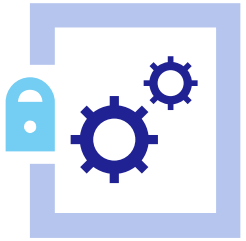
Trending toward serverless platforms with simple code to cloud pipelines



Using multiple languages and frameworks during development

Common challenges in microservice development

Disjointed tools and runtimes to build distributed applications

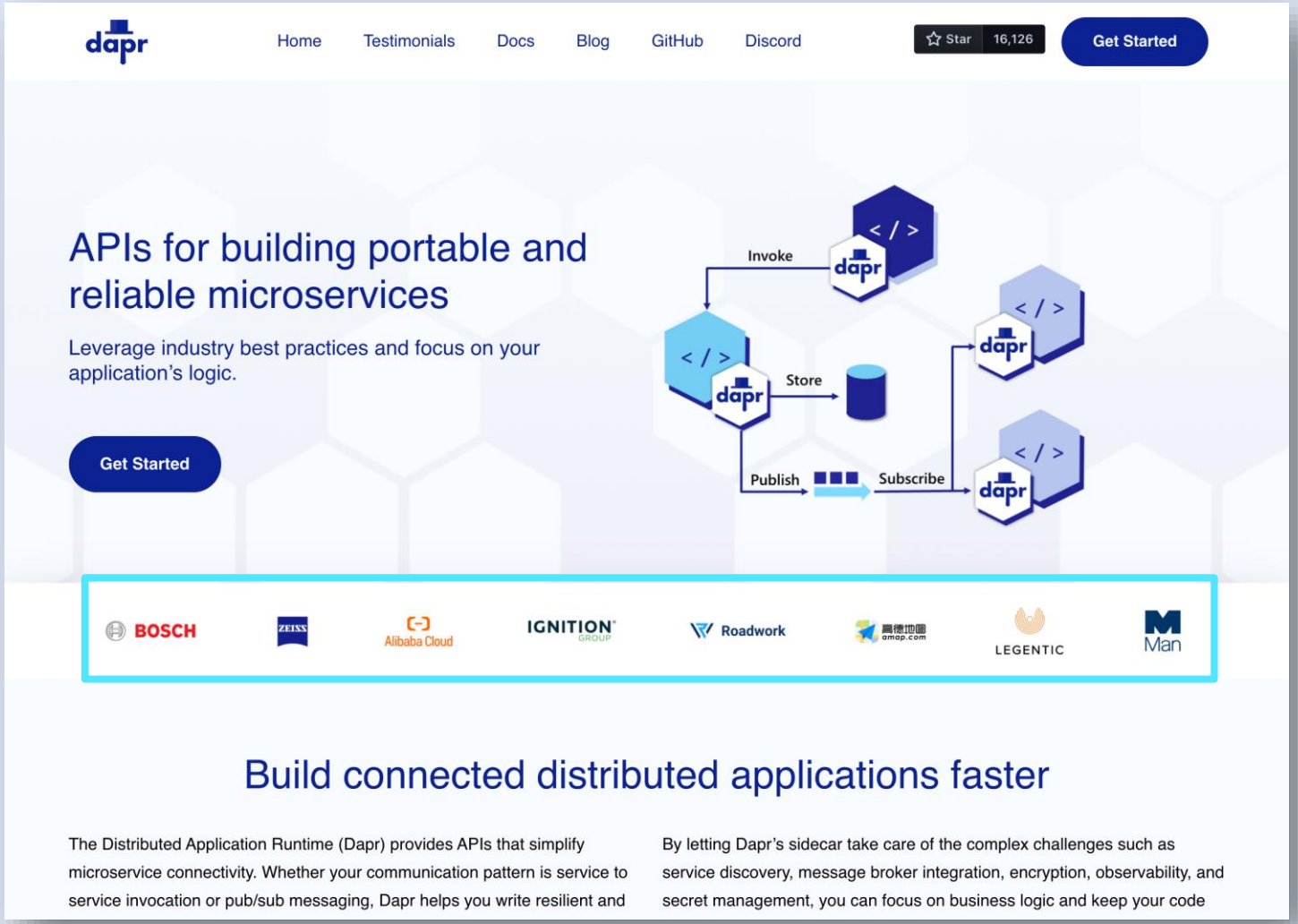


Runtimes have limited language support and tightly controlled feature sets

Runtimes only target specific infrastructure platforms with limited portability

Distributed Application Runtime

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



dapr.io

CLOUD NATIVE COMPUTING FOUNDATION

cncf.io/projects/dapr

CLOUD NATIVE COMPUTING FOUNDATION

About ▾ Projects ▾ Certification ▾ Community ▾ Blog & news ▾ JOIN NOW

CONTOUR
Service Proxy

cortex
Monitoring

cri-o
Container Runtime

Crossplane
Scheduling & Orchestration

dapr
Framework

Dragonfly
Container Registry

EMISSARY INGRESS
API Gateway

falco
Security & Compliance

flagger
Continuous Integration & Delivery

flux
Continuous Integration & Delivery

gRPC
Remote Procedure Call

KEDA
Installable Platform

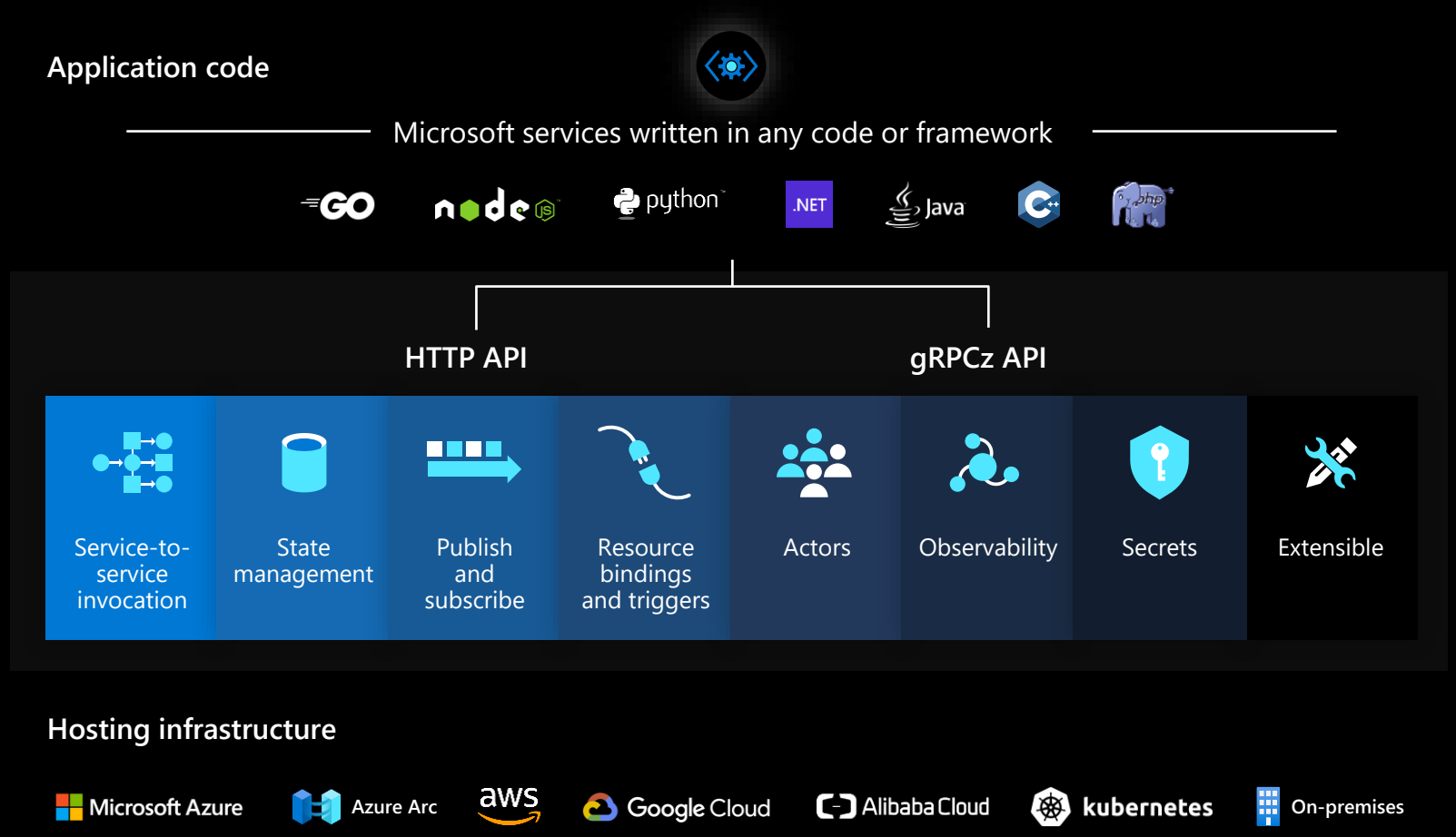
KubeEdge
Automation & Configuration

LONGHORN
Cloud Native Storage

NATS
Streaming & Messaging

Microservices using any language or framework

Any cloud or edge infrastructure

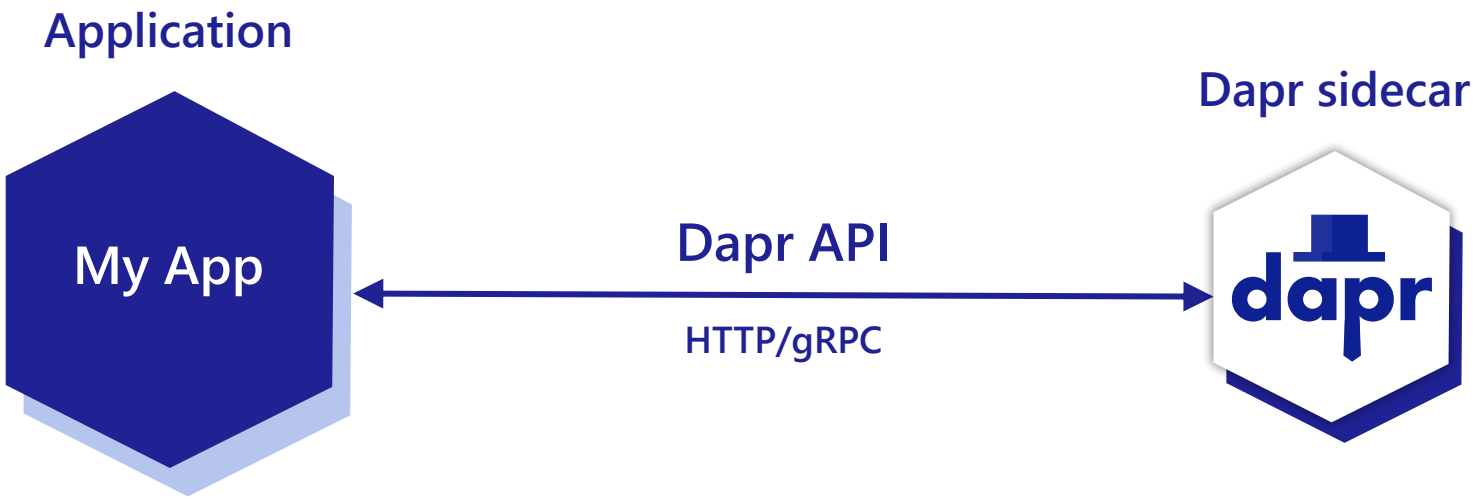


Dapr value pillars

- 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

Sidecar model



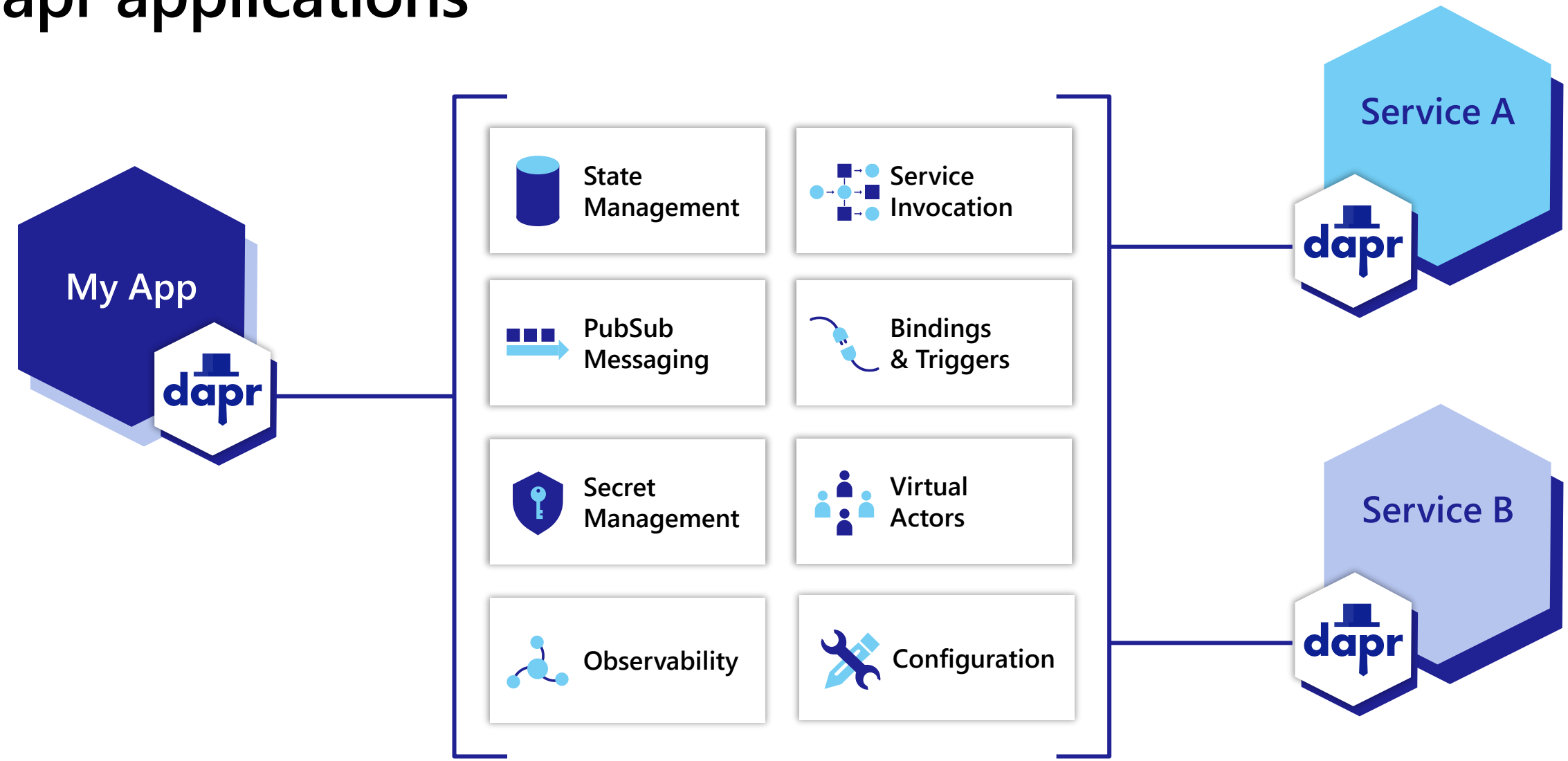

```
POST http://localhost:3500/v1.0/invoke/cart/method/neworder

GET http://localhost:3500/v1.0/state/inventory/item67

POST http://localhost:3500/v1.0/publish/shipping/orders

GET http://localhost:3500/v1.0/secrets/keyvault/password
```

Dapr applications

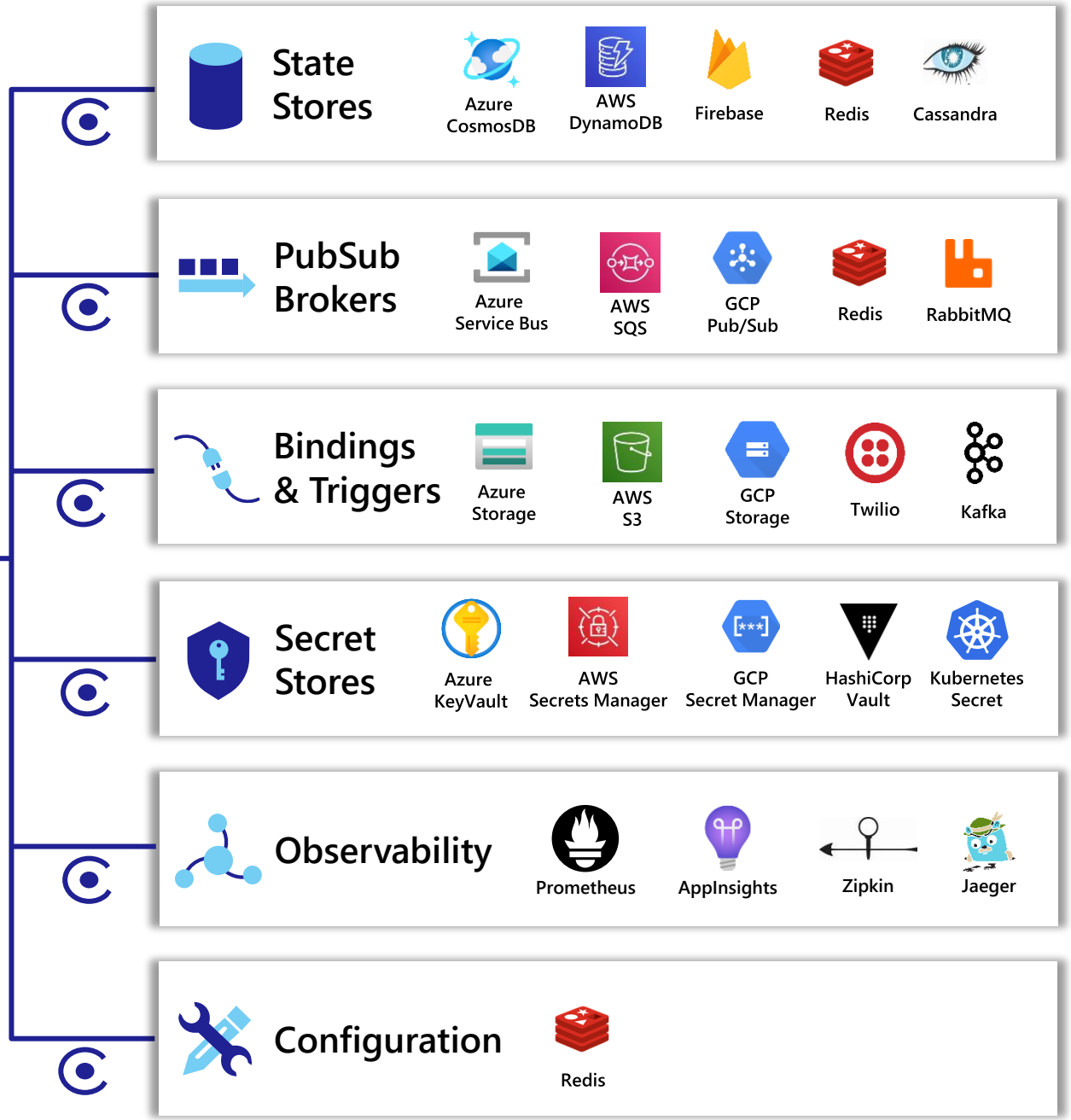


Dapr components

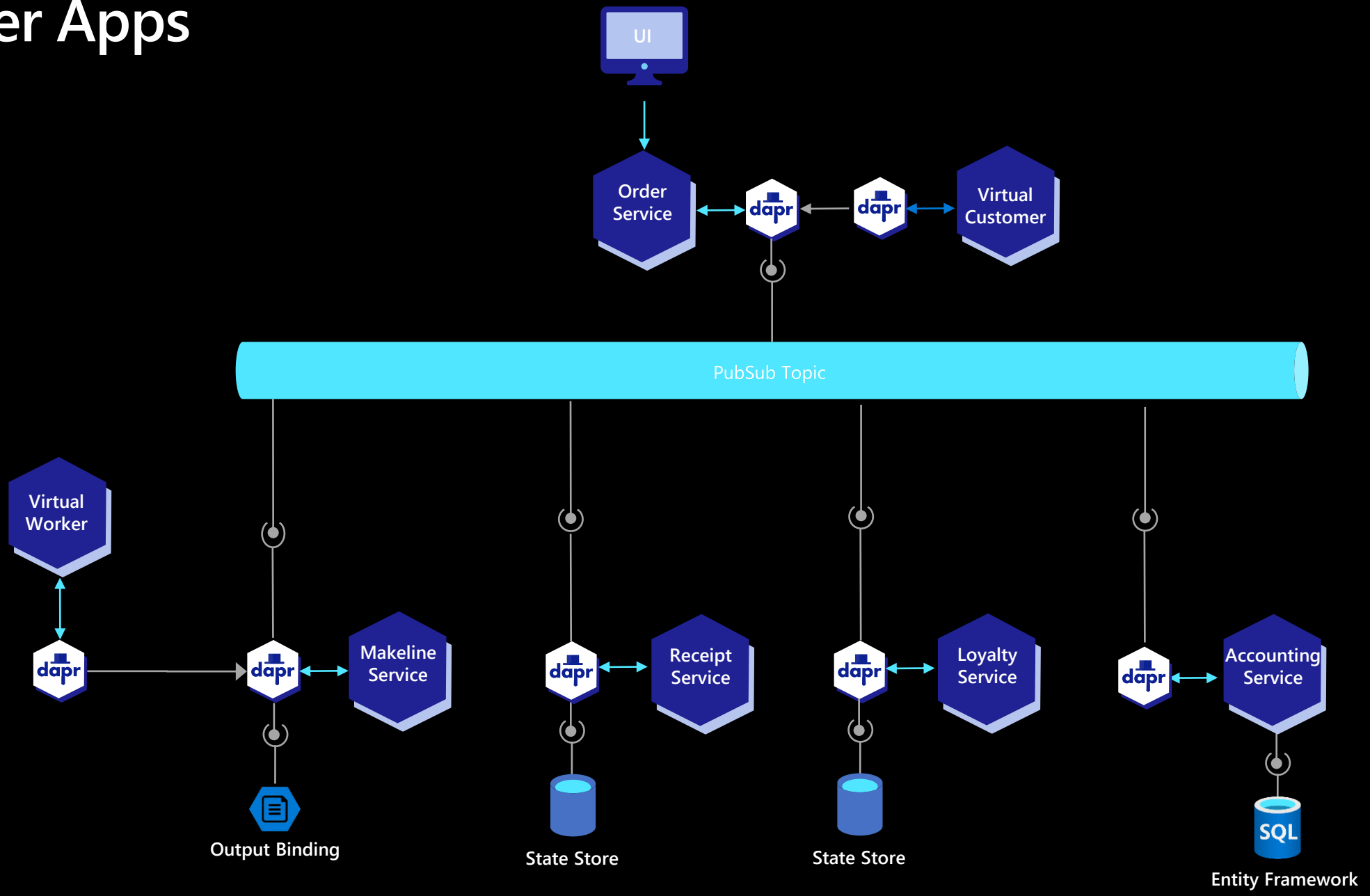
Swappable YAML files with resource connection details

Over 70 components available

Create components for your resource at: github.com/dapr/components-contrib

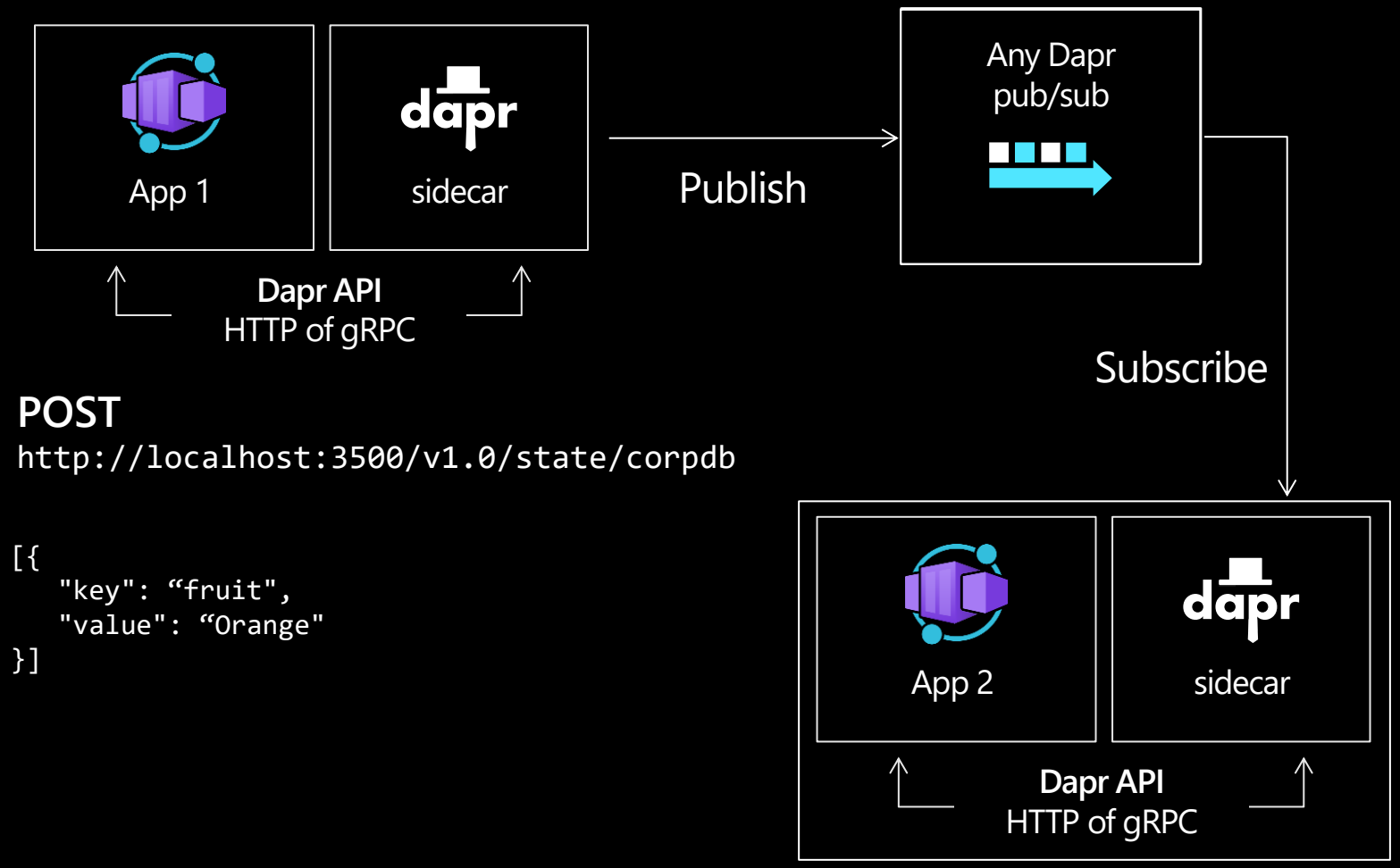


Container Apps



Publish and subscribe

Create event-driven, loosely coupled architectures where producers send events to consumers via topics.

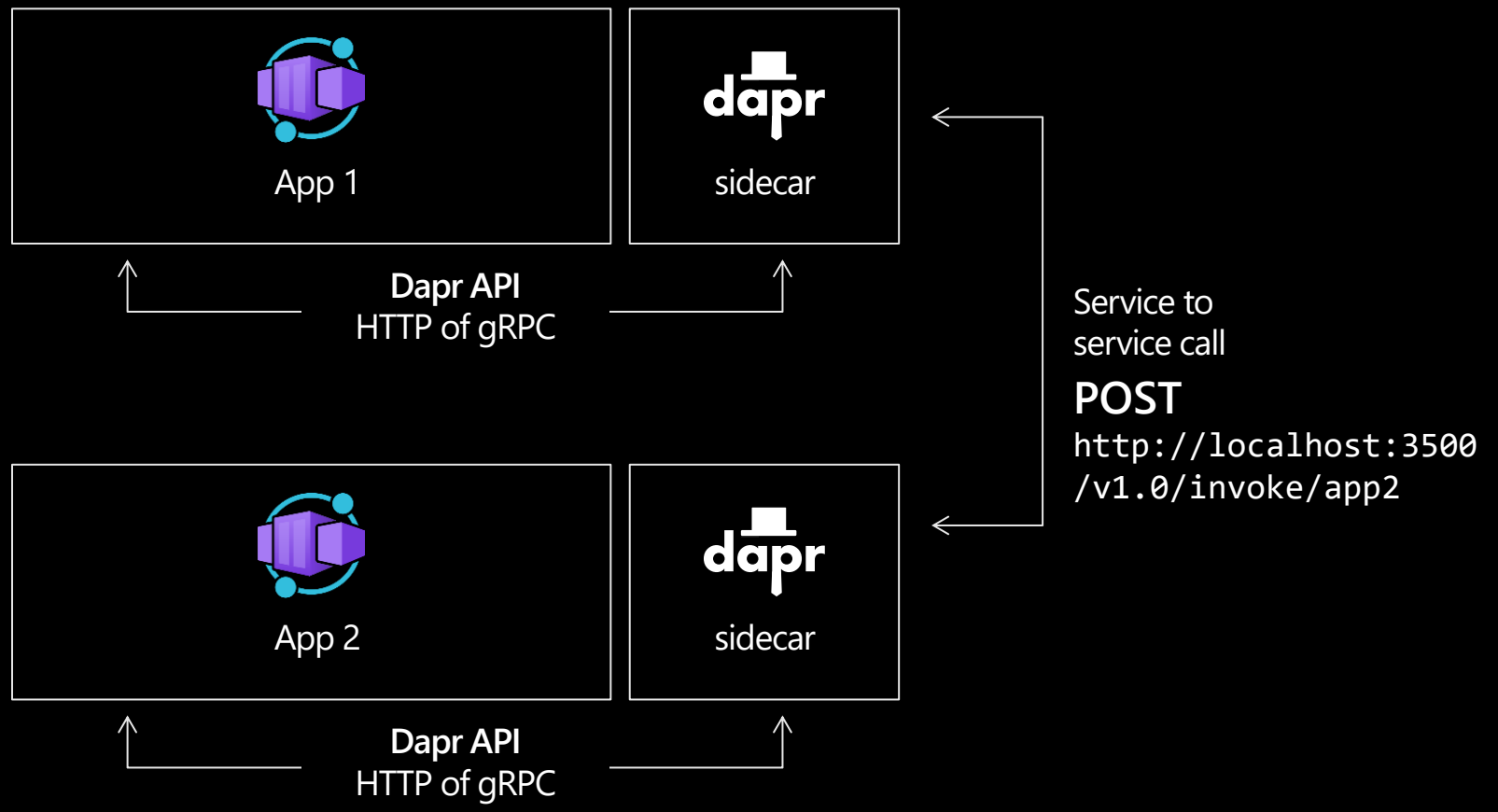


POST
`http://localhost:3500/v1.0/state/corpdb`

```
[{
   "key": “fruit",
   "value": “Orange"
}]
```

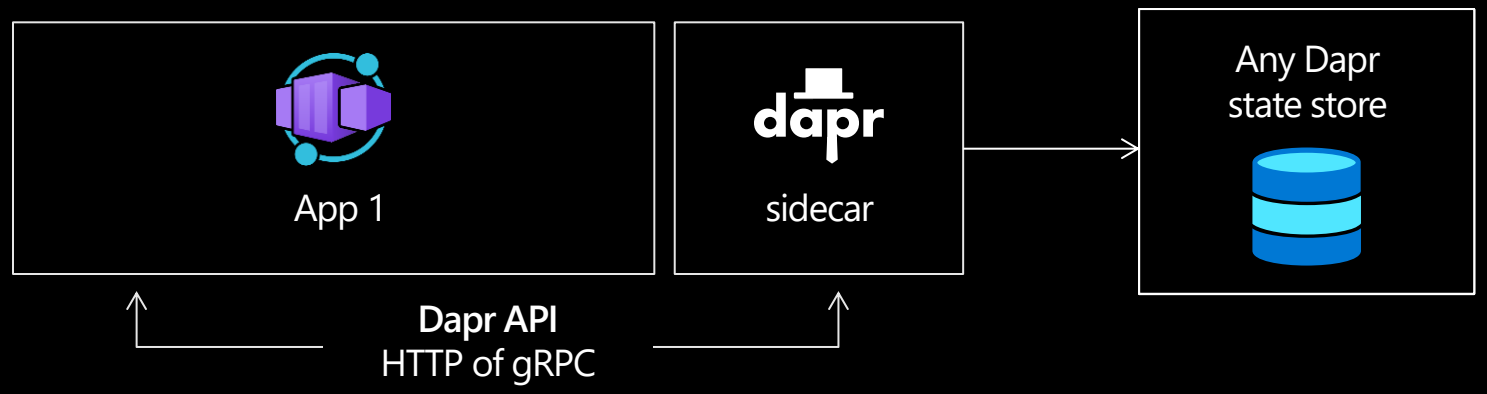
Service to service invocation

Fully managed Dapr APIs provide a rich set of capabilities and productivity gains



State management

Dapr provide apps with state management capabilities for CRUD operations, transactions and more



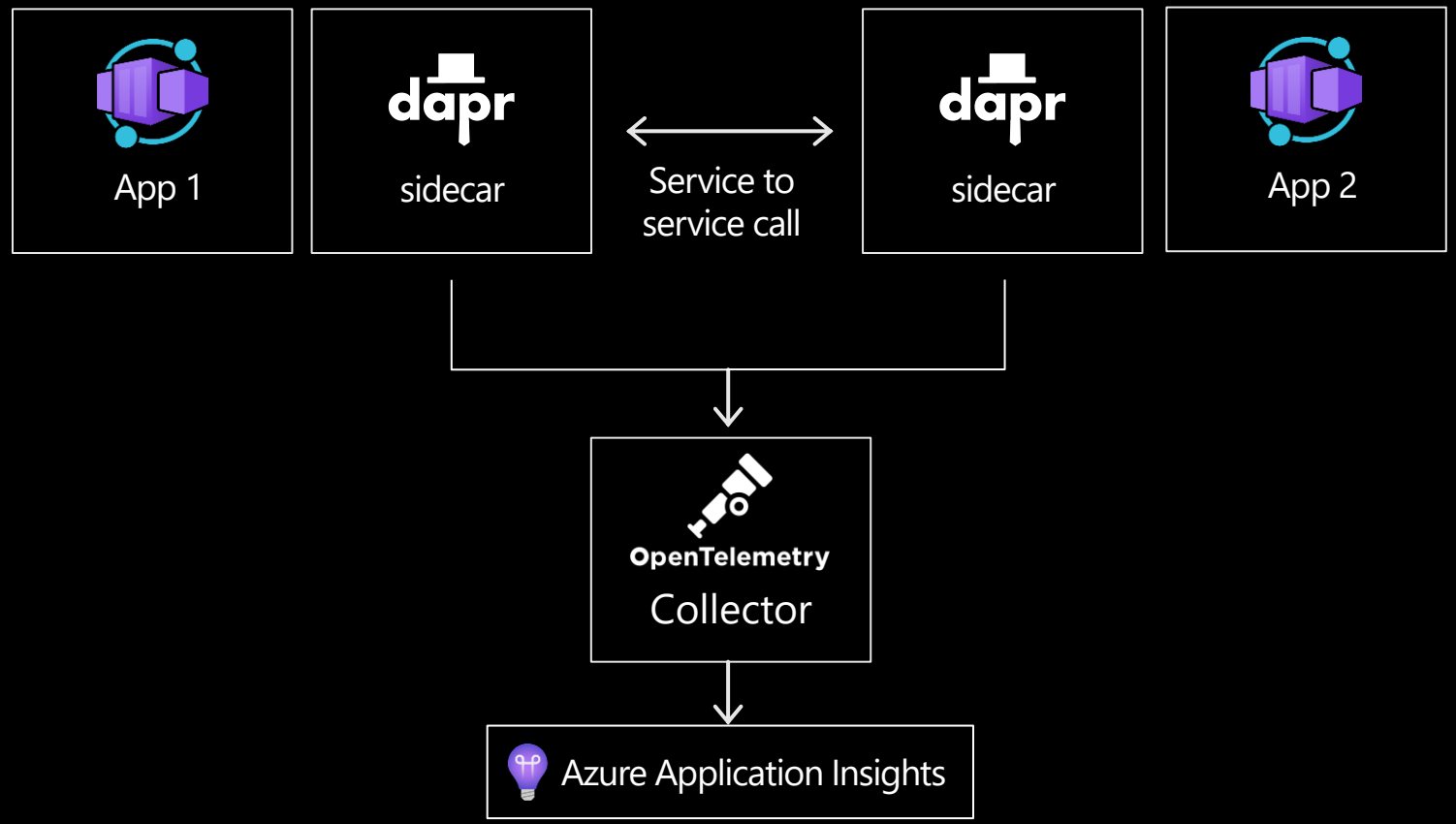
POST

```
http://localhost:3500/v1.0/state/corpdb
```

```
[{
   "key": “fruit",
   "value": “Orange"
}]
```

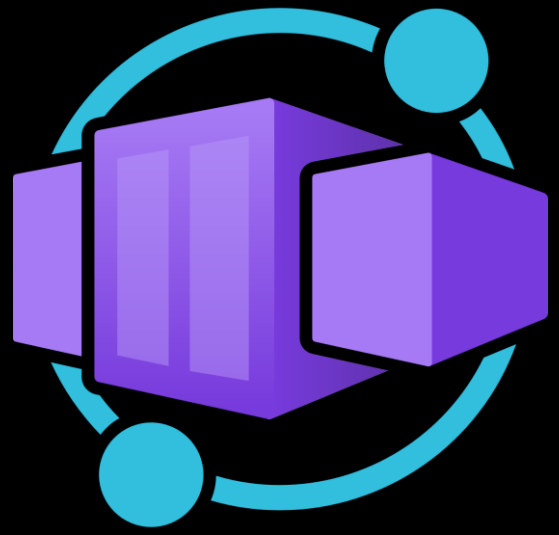
Observability

Intercept traffic and extract tracing, metrics, and logging information. Configure Azure Application Insights for distributed tracing across your services



Azure Container Apps

GA



Serverless containers for microservices

Build modern apps on open source

Focus on apps, not infrastructure

Seamlessly port to Kubernetes

Kubernetes KEDA dapr DAPR envoy Envoy

Azure containers portfolio

	Infrastructure control	Serverless applications
	Use the Kubernetes APIs to manage, deploy and scale the cluster in addition to application development	Platform handles the orchestration details with a focus on application development
Unopinionated containers	Any container image, no restrictions	Azure Kubernetes Service Azure Red Hat OpenShift	Azure Container Instances New: Azure Container Apps
Opinionated containers	Specific base image or programming model required		Azure App Service* Azure Functions*

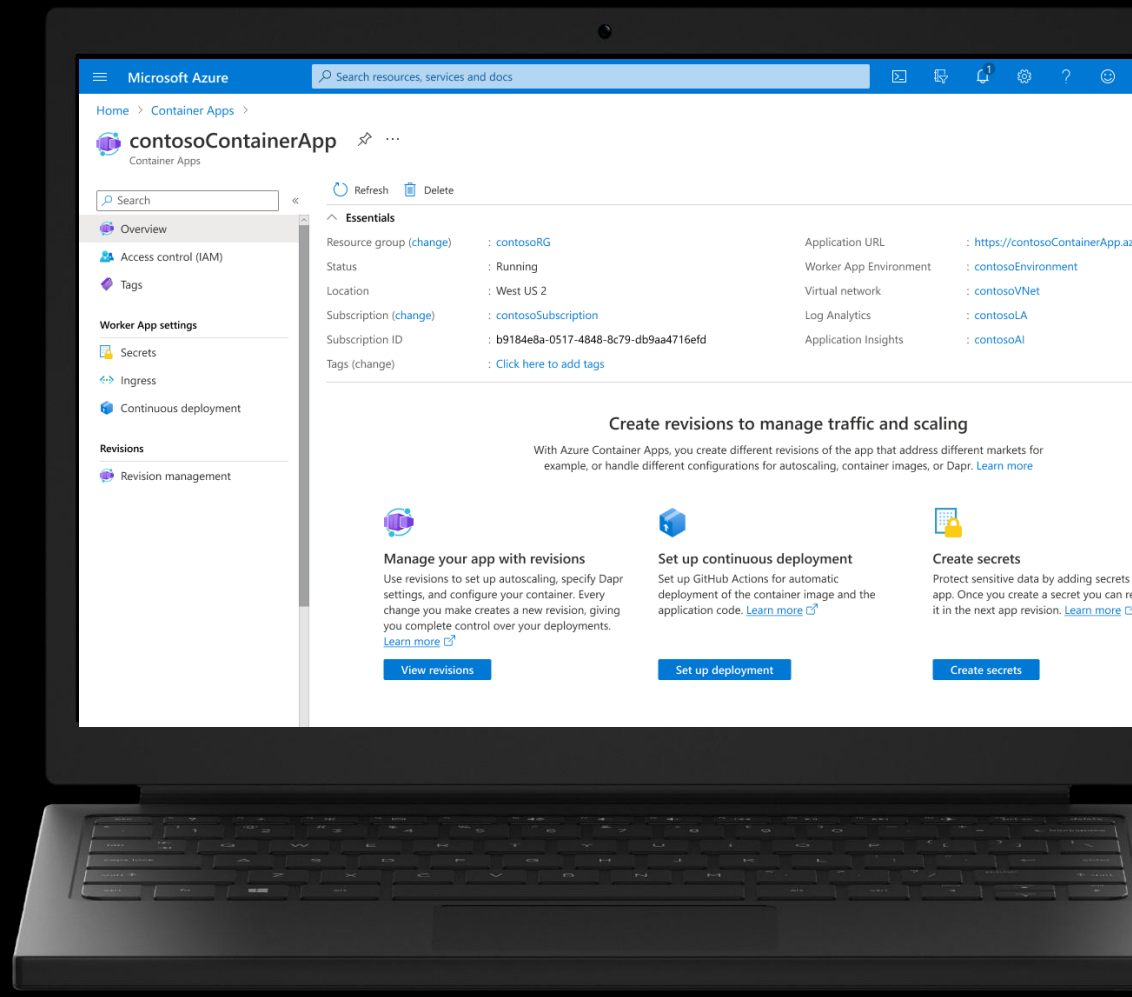
* When used with containers

Scale with flexible serverless containers

Run containers and scale in response to HTTP traffic or a growing list of KEDA-supported scale triggers including Azure Event Hub, Apache Kafka, RabbitMQ Queue, MongoDB, MySQL, and PostgreSQL

Get robust autoscaling capabilities without the overhead of managing complex infrastructure.

Scale to zero and pay for only what you use, by the second.

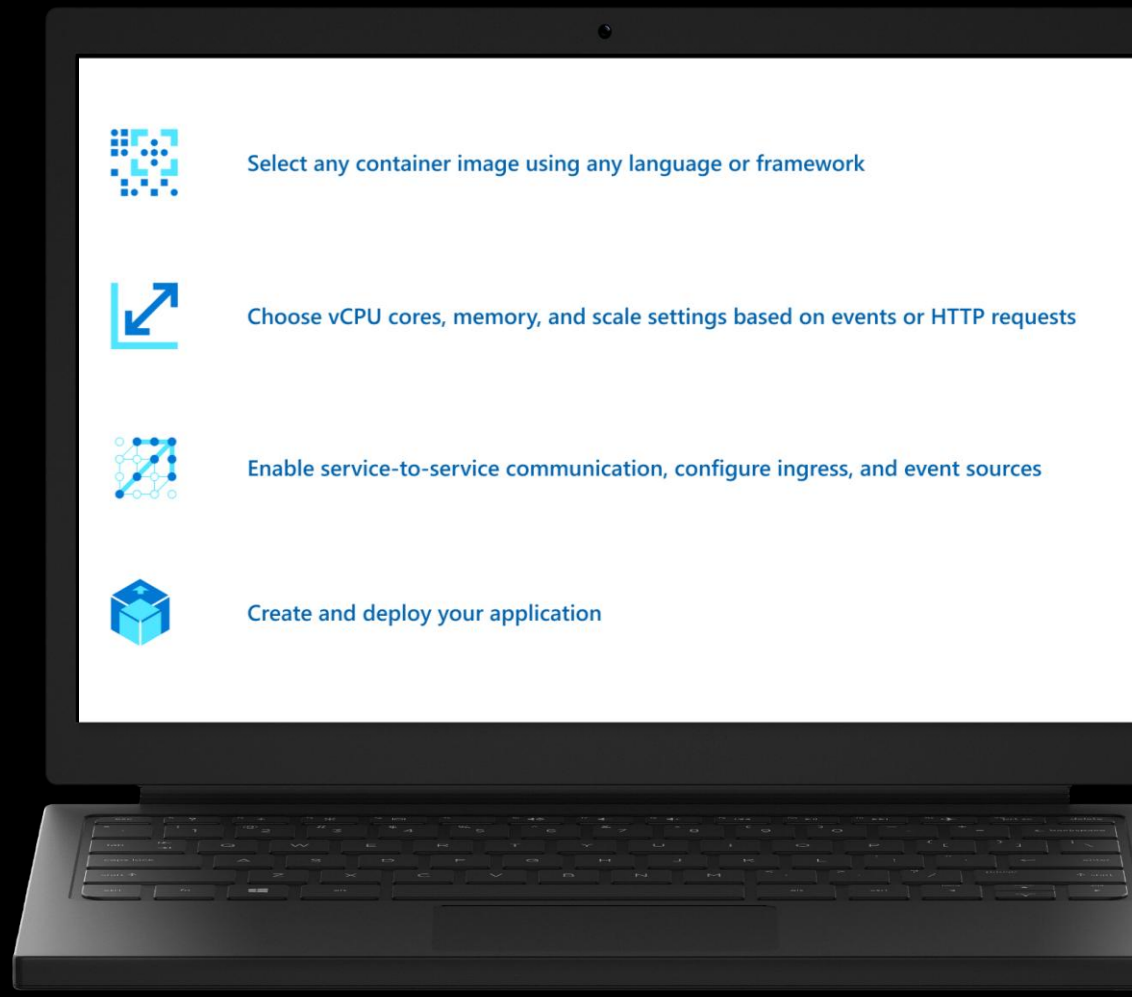


Accelerate developer productivity

Build microservices, APIs, event processing workers, and background jobs using containers.

Write code in your favorite programming language and accelerate development with built-in Distributed Application Runtime (Dapr) integration to simplify common tasks like event processing, pub/sub, and service invocation.

Set up a code-to-cloud pipeline using GitHub Actions.

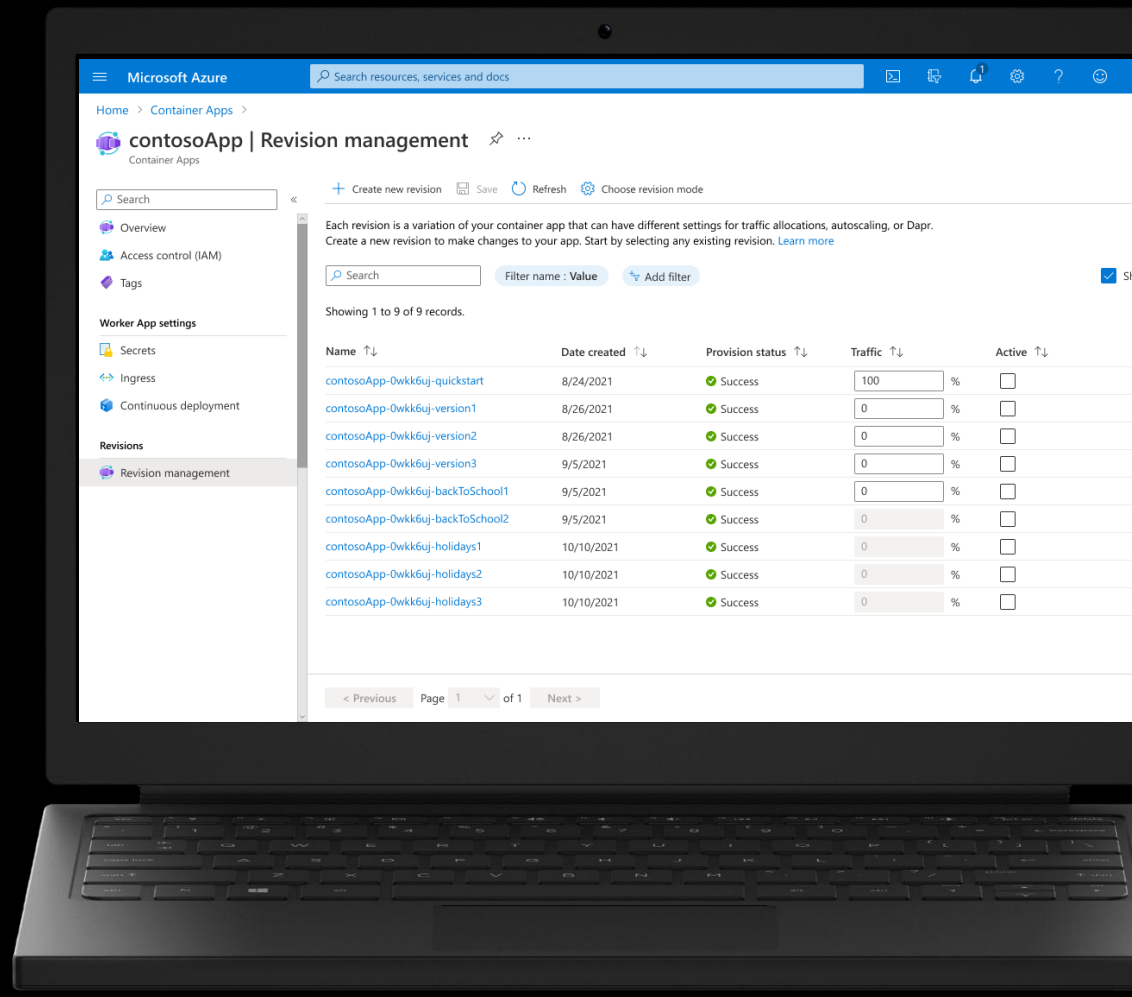


Build modern apps on open-source

Create modern apps with open standards on a Kubernetes foundation and portability in mind.

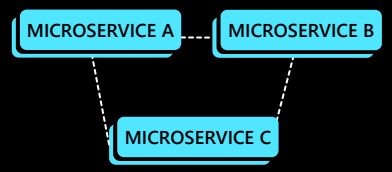
Contribute directly to OSS projects to influence product capabilities.

Rely on streamlined application lifecycle tasks such as application upgrades and versioning, traffic shifting, service discovery, and monitoring.



What can you build with Azure Container Apps?

Microservices



Deploy and manage a microservices architecture with the option to integrate with Dapr.

AUTO-SCALE CRITERIA

Individual microservices can scale independently using any KEDA scale triggers

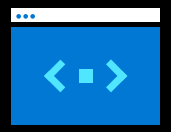
Event-driven processing

E.g., queue reader application that processes messages as they arrive in a queue.

AUTO-SCALE CRITERIA

Scaling is determined by the number of messages in the queue

Web Applications

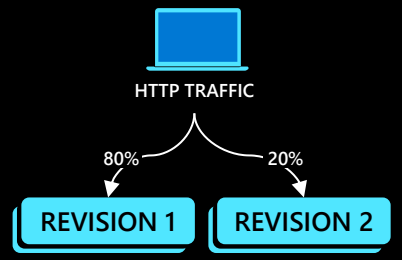


Deploy web apps with custom domains, TLS certificates, and integrated authentication.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP requests

Public API endpoints



HTTP requests are split between two revisions of the app — the first revision gets 80% of the traffic, while a new revision receives 20%.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP requests

Background processing

E.g., continuously-running background process that transforms data in a database.

AUTO-SCALE CRITERIA

Scaling is determined by the level of CPU or memory load

Engagements

Milliman, Arius Team

Customer: The application allows users to submit a job to perform various python-based model simulations in Azure via AKS. The project uses AKS to scale-out hundreds of nodes, which perform various model calculations in parallel, and then when all nodes are done, the results are aggregated and returned to the user.

+ Ease of deployment, via CLI and ARM. No overhead of AKS, operations and maintenance.

+ We benchmarked initial scaling times on AKS versus Container Apps. Container Apps was faster.

- Unfortunately, their calculations require a lot more application and cluster level scaling.

American Airlines, Customer HUB Application

Customer: Very large complex application design modernized to Azure using many Azure Services. Majority of code is Core Java. One of the use cases is JMS Listener processing messages from Service Bus Queue.

+ The JMS listener is deployed as a Linux Container using the ARM template and specifying the scaling criteria, and deploying the Container was a very positive experience. The scale limit to 10 replicas is more than enough to meet the scaling need for this use case, and as soon as messages arrive in the Service Bus Queue, JMS Listener processes them and then scales back down.

Engagements

Customer: Gluwa, an ISV

- Worked with them to implement ACA with Dapr. A lot of their issues ended up being with Dapr itself, especially around state store with MongoDB, not really with ACA.

+ Very simple to use, easy to deploy. They were up and running in a week or so with their MVP

+ PG has been extremely helpful, was available for a call and helped customer out.

- Latest version of Dapr is 1.6, but the supported version of Dapr for ACA is 1.4.2

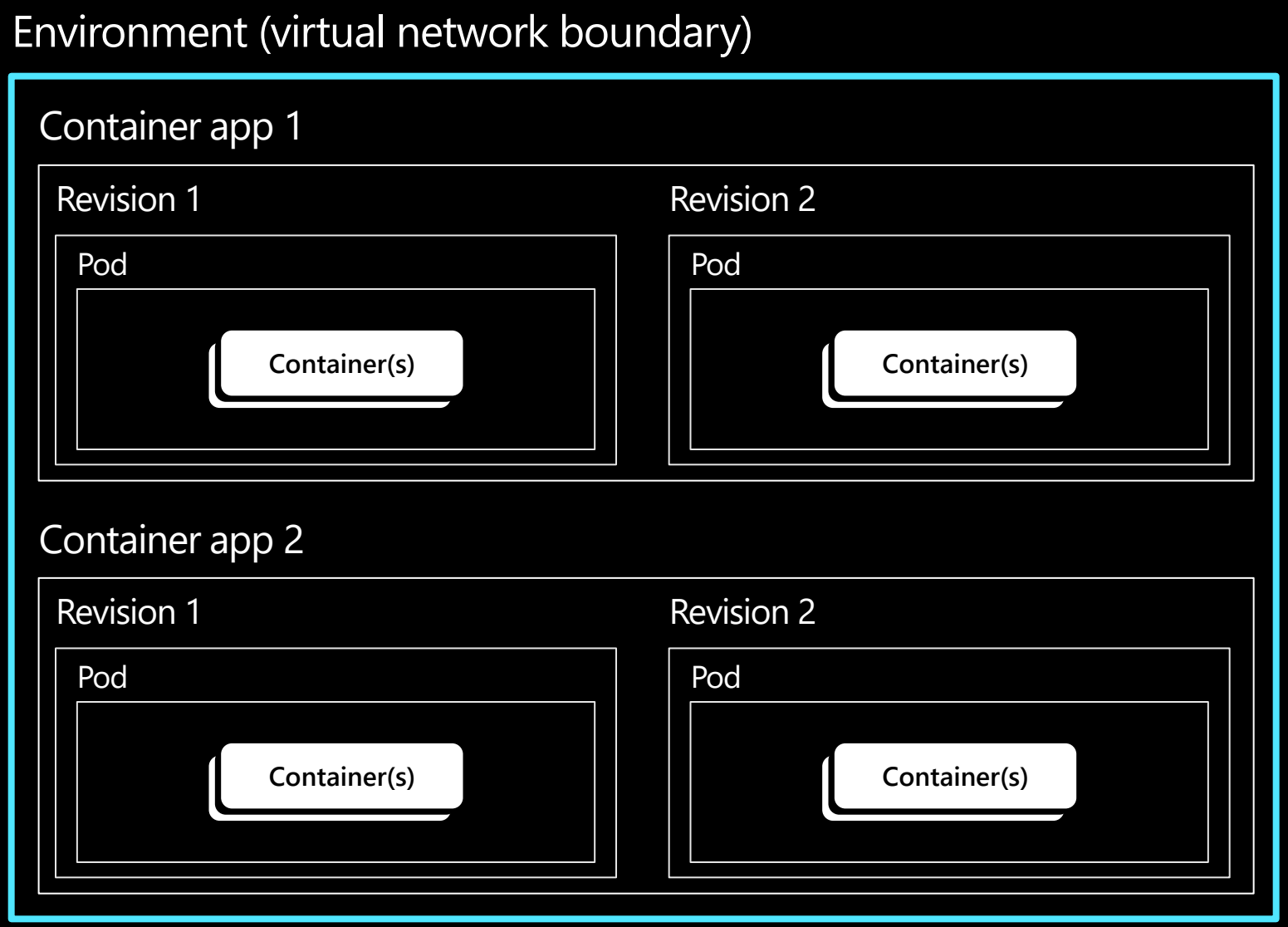
- Only 2 environments allowed per subscription. Customer was looking for 4 to support each lifecycle environment.

- Dapr challenges

Deep dive

Environments

Environments define an isolation and observability boundary around a collection of container apps deployed in the same virtual network



Recent news/announcements

- BYO Virtual Network
  - Still working on documentation and evaluating integration with other networking services in Azure

Azure Container Apps Virtual Network Integration

Subscribe

By Kendall Roden

Published Feb 02 2022 10:51 AM 6,794 Views

Azure Container Apps Virtual Network Integration release announcement

At Ignite, we announced Azure Container Apps, a serverless application-centric hosting service that enables users to quickly deploy containerized applications and microservices to Azure without the need to configure and manage underlying infrastructure resources or container orchestrators. Azure Container Apps runs on Azure Kubernetes Service, and includes several open-source projects: Kubernetes Event Driven Autoscaling (KEDA), Distributed Application Runtime (Dapr), and Envoy. This open-source foundation enables teams to build and run portable applications powered by Kubernetes and open standards without the operational overhead and management complexity of working with the platform directly.

Since its preview release, the Azure Container Apps service has continued evolving to address customer needs and respond to customer feedback. One point of focused investment has been addressing asks related to network isolation. For security and compliance reasons, many enterprise customers require isolation to ensure internal, mission critical applications are inaccessible from the public internet. In addition, customers typically have existing cloud resources like databases, queues, file shares, etc. that an application needs to integrate with over this internal, private network. Today, we are excited to announce that Azure Container Apps can be deployed into your custom Azure Virtual Network! In the remainder of this blog, we will walk through configuring inbound and outbound virtual network integration for an Azure Container Apps environment and subsequently deploying Container Apps to the running environment.

Configure Virtual Network for an Azure Container Apps Environment

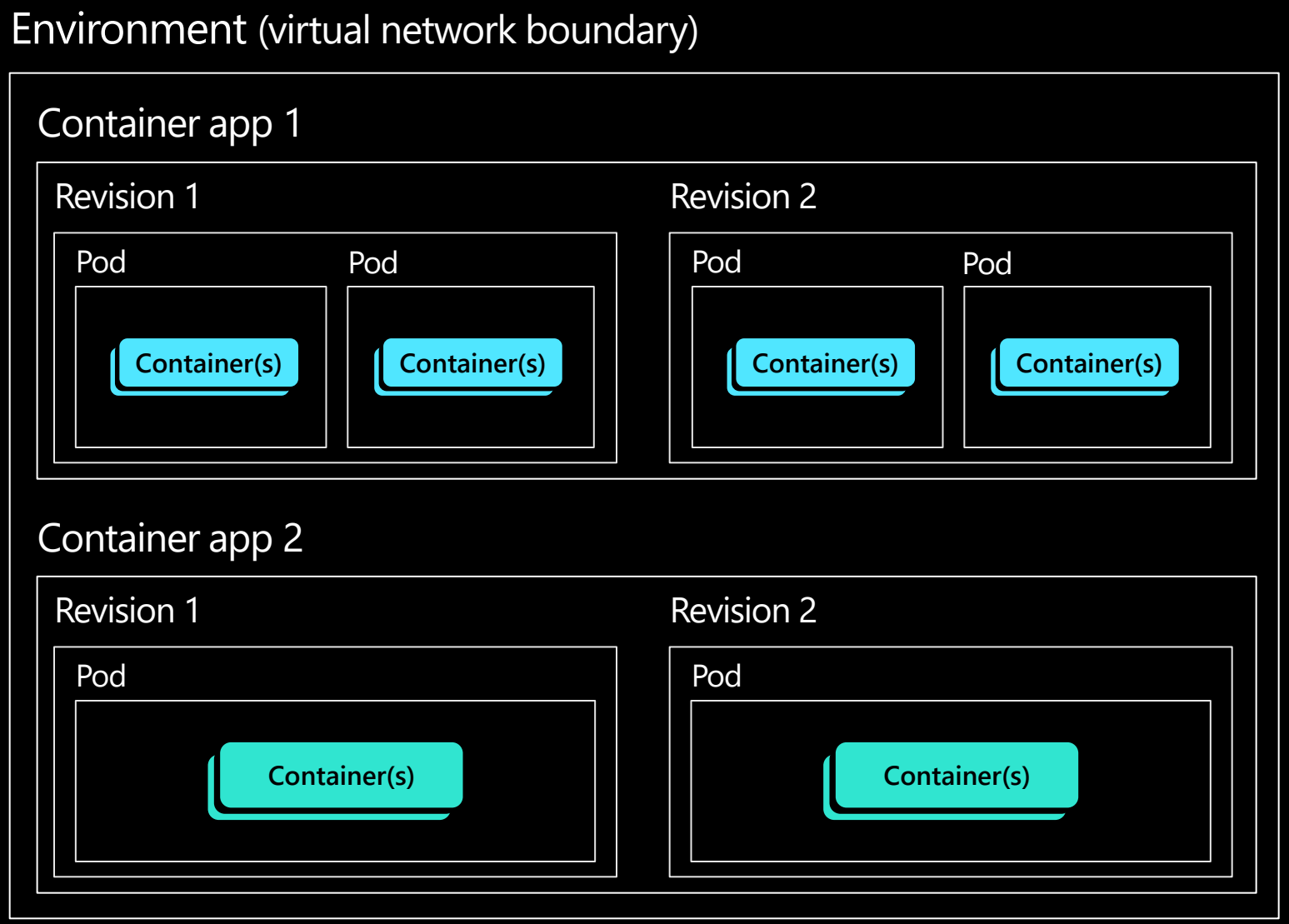
Azure Container Apps Virtual Network Integration - Microsoft Tech Community

```
~> az network vnet create -g $RG -n $VNET_NAME --address-prefix 10.110.0.0/16
```

aka.ms/containerappsbyovnet

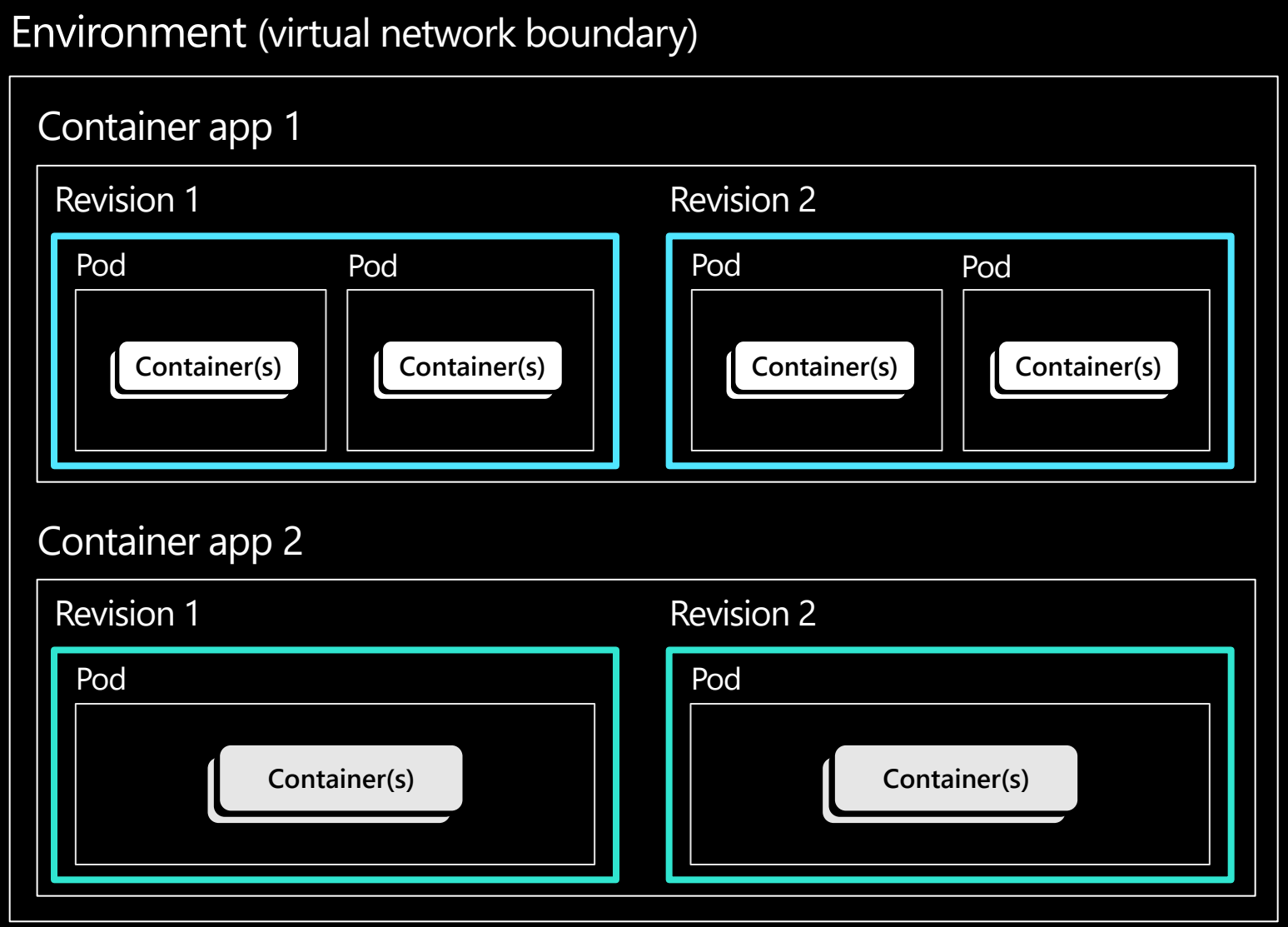
Containers

Containers in Azure Container Apps can use any and development stack of your choice



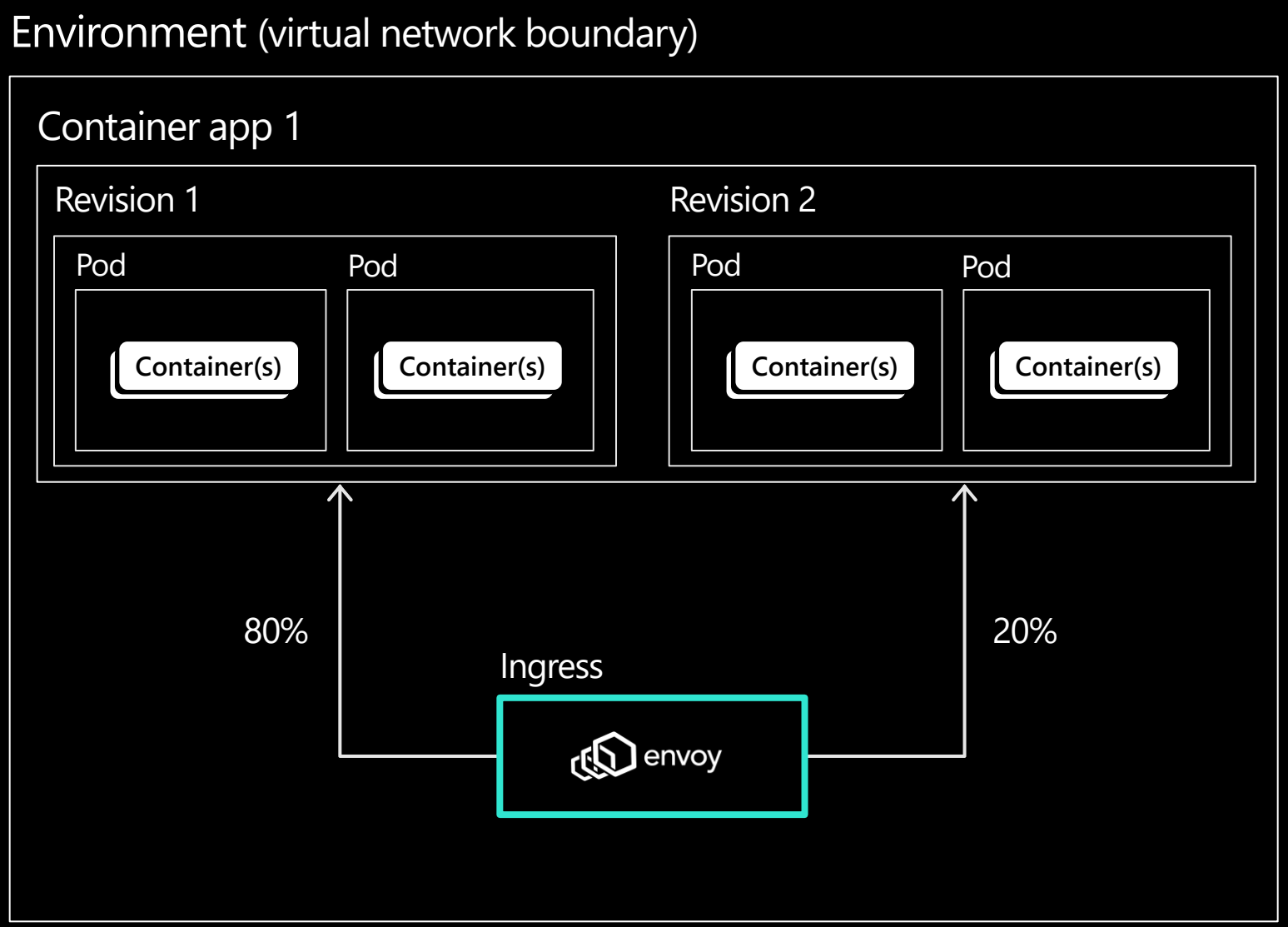
Revisions

Revisions are immutable version snapshots of a container app



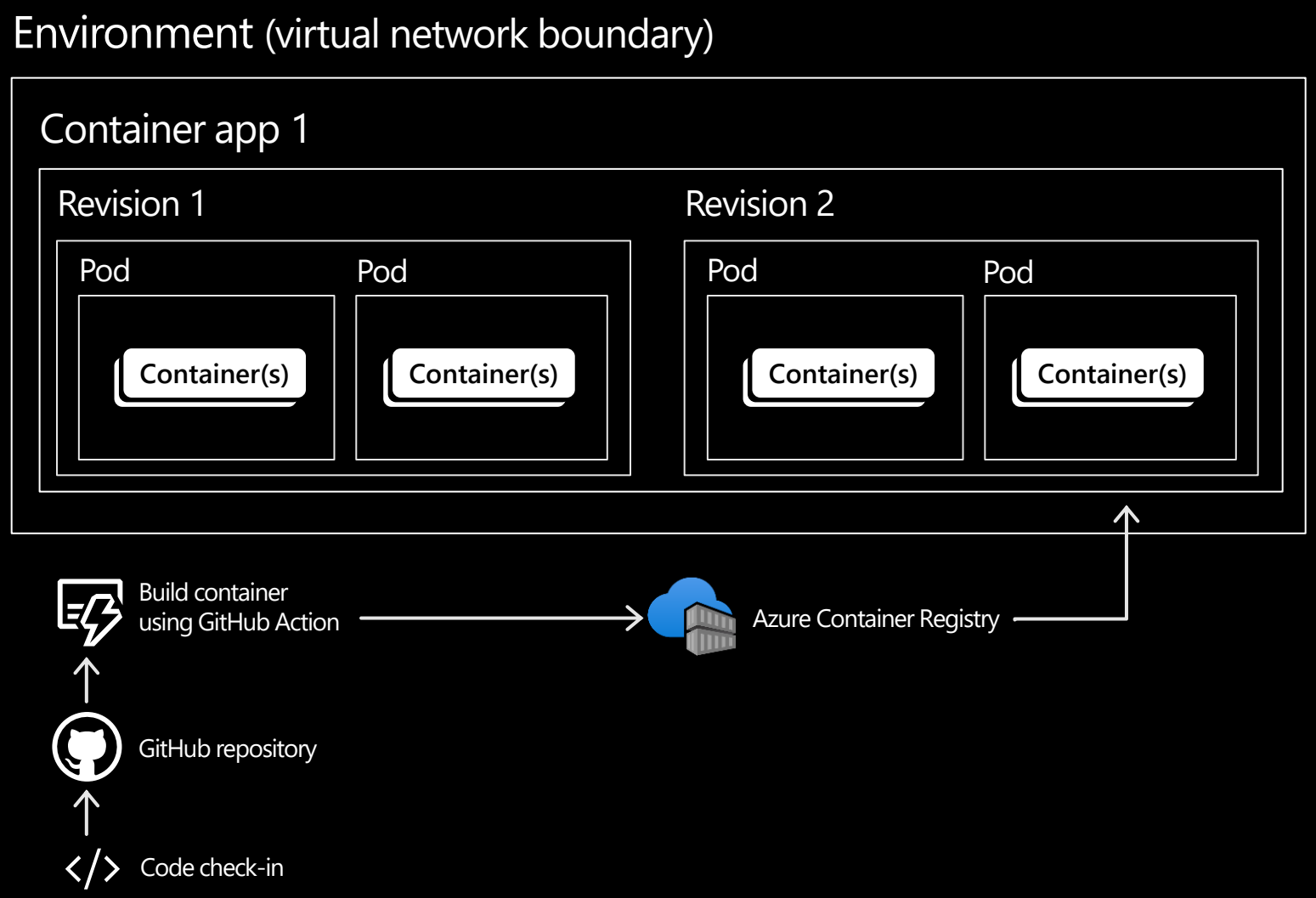
Internal or external visibility with TLS termination and support for HTTP/1.1 and HTTP/2

Ingress



GitHub Actions integration

Publish revisions as commits are pushed to your GitHub repository by triggering a GitHub Action to build a new container image



Secrets management

Securely store sensitive configuration elements that are then available to containers through environment variables, scale rules, and Dapr

```
"template": {
    "containers": [
        {
            "image": "myregistry/myQueueApp:v1",
            "name": "myQueueApp",
            "env": [
                {
                    "name": "QueueName",
                    "value": "myqueue"
                },
                {
                    "name": "ConnectionString",
                    "secretref": "queue-connection-string"
                }
            ]
        }
    ],
}
```

Managed Identity

Coming soon

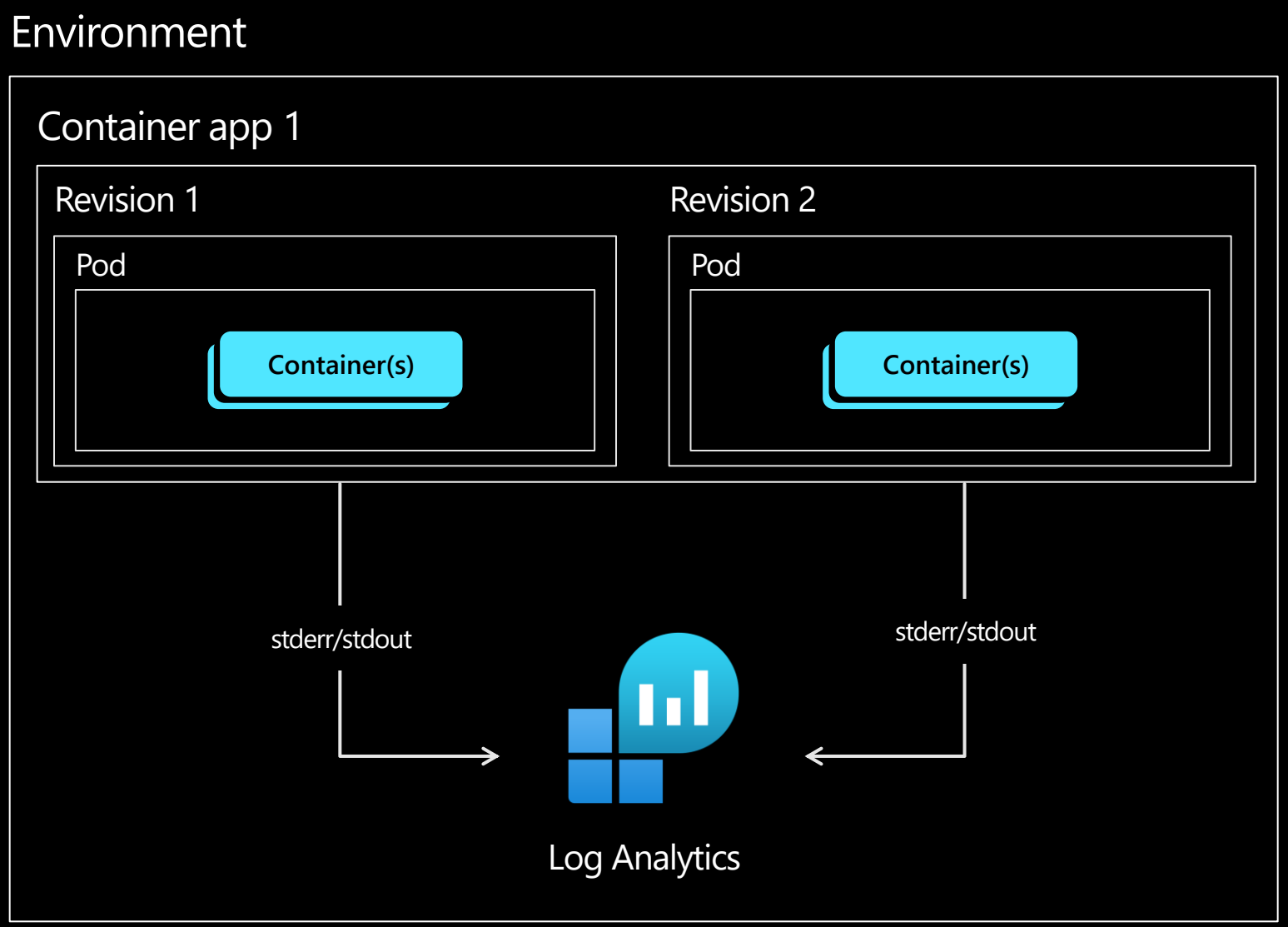
- Enable managed identity for a container app
- Can be system-assigned or user-assigned

Use Cases

- Identity-based connections in app (e.g., connect to SQL Server) – ETA March
- Pull images from Azure Container Registry – planned
- KEDA scaler configuration – planned
- Dapr component configuration – investigating
- Key Vault references – investigating

Logging

Containers write logs to standard output or standard error streams surfaced via Log Analytics



Observability

Available Now

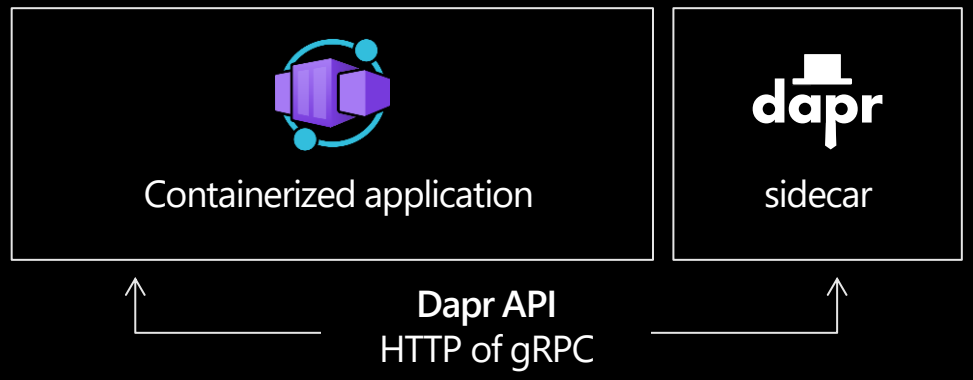
- Log Analytics – stderr/stdout, small delay
- Metrics – CPU, Memory, Bytes in/out, Requests
- Alerts – based on metrics, log search, admin signals (e.g., create, update, delete container app)

Upcoming Investments

- Streaming Logs – stderr/stdout, real-time
- Connect to Console – connect to run shell commands
- Events – emitted from underlying orchestrator (e.g., container start failure, scale up/down)

Using the Distributed Application Runtime (Dapr)

Fully managed Dapr using the sidecar model



Service-to-service invocation

```
POST http://localhost:3500/v1.0/invoke/cart/method/neworder
```

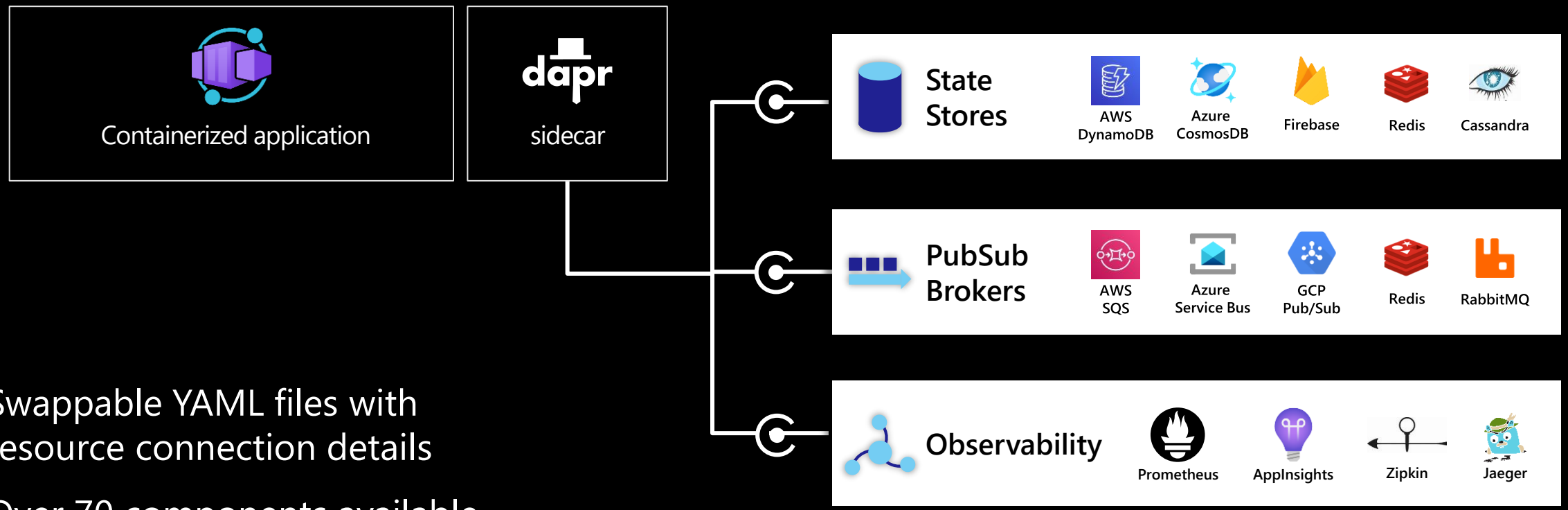
State management

```
GET http://localhost:3500/v1.0/state/inventory/item67
```

Publish and subscribe

```
POST http://localhost:3500/v1.0/publish/shipping/orders
```

Dapr components



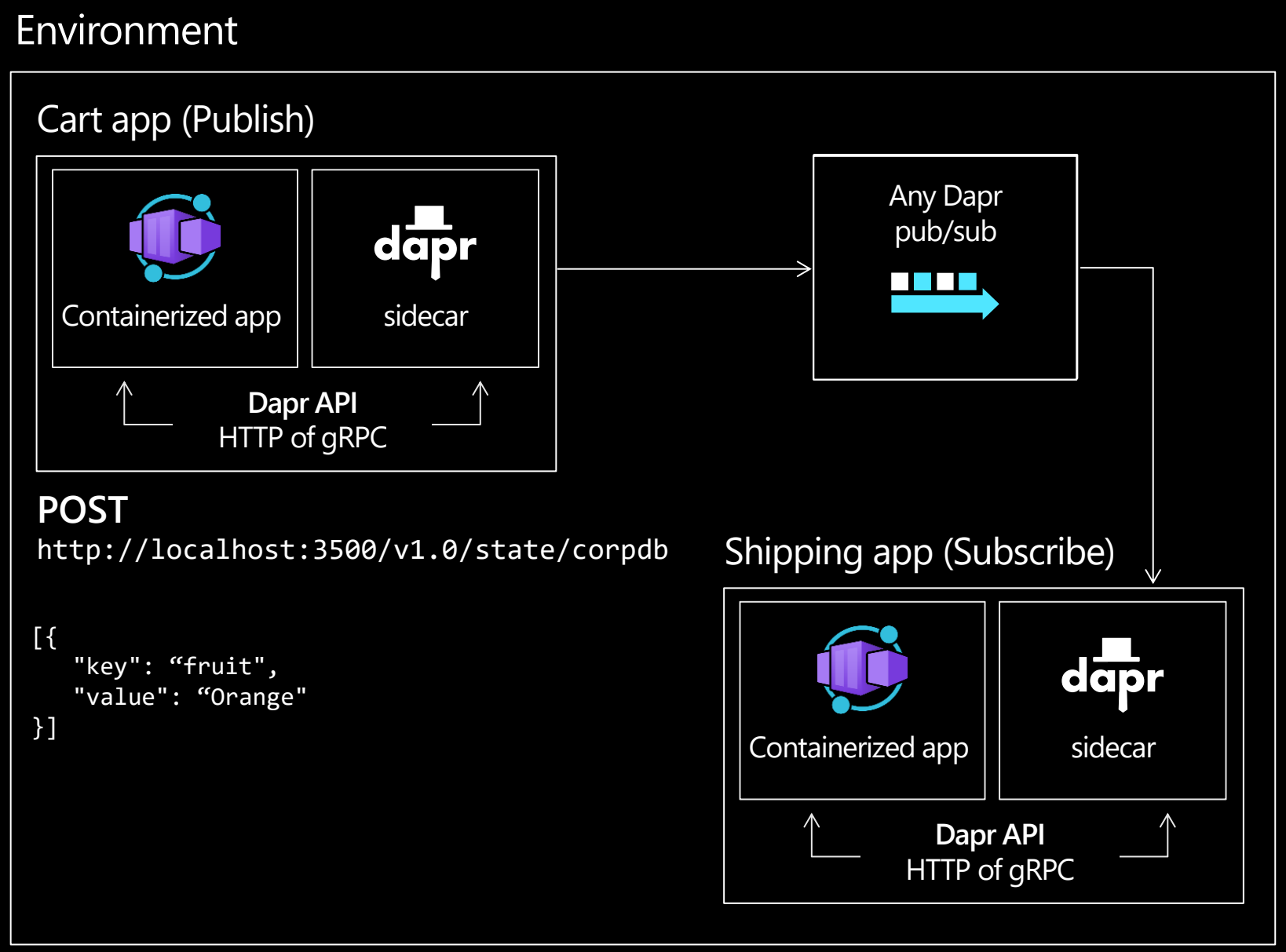
Swappable YAML files with resource connection details

Over 70 components available

Create components for your resource at: github.com/dapr/components-contrib

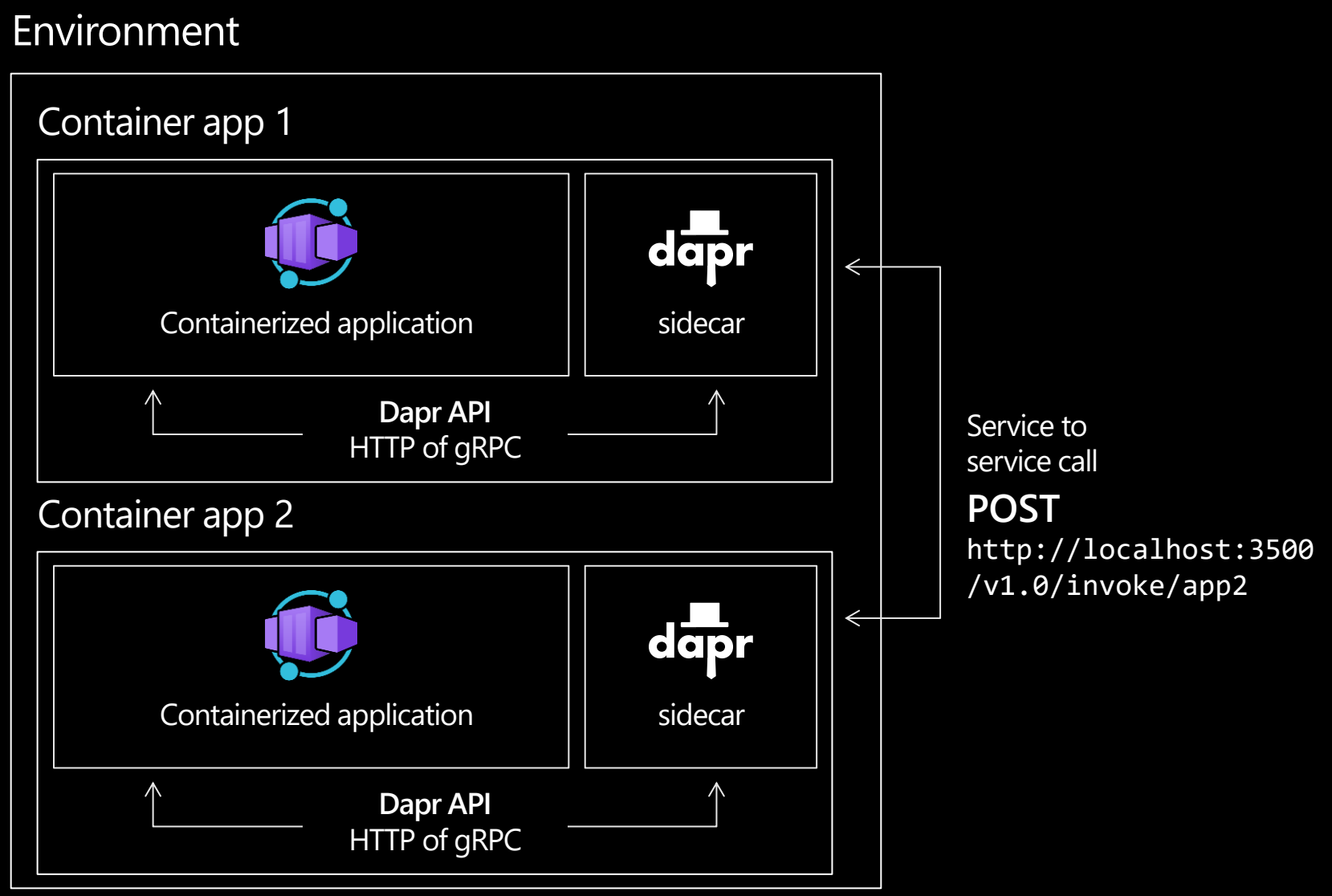
Publish and subscribe

Create event-driven, loosely coupled architectures where producers send events to consumers via topics.



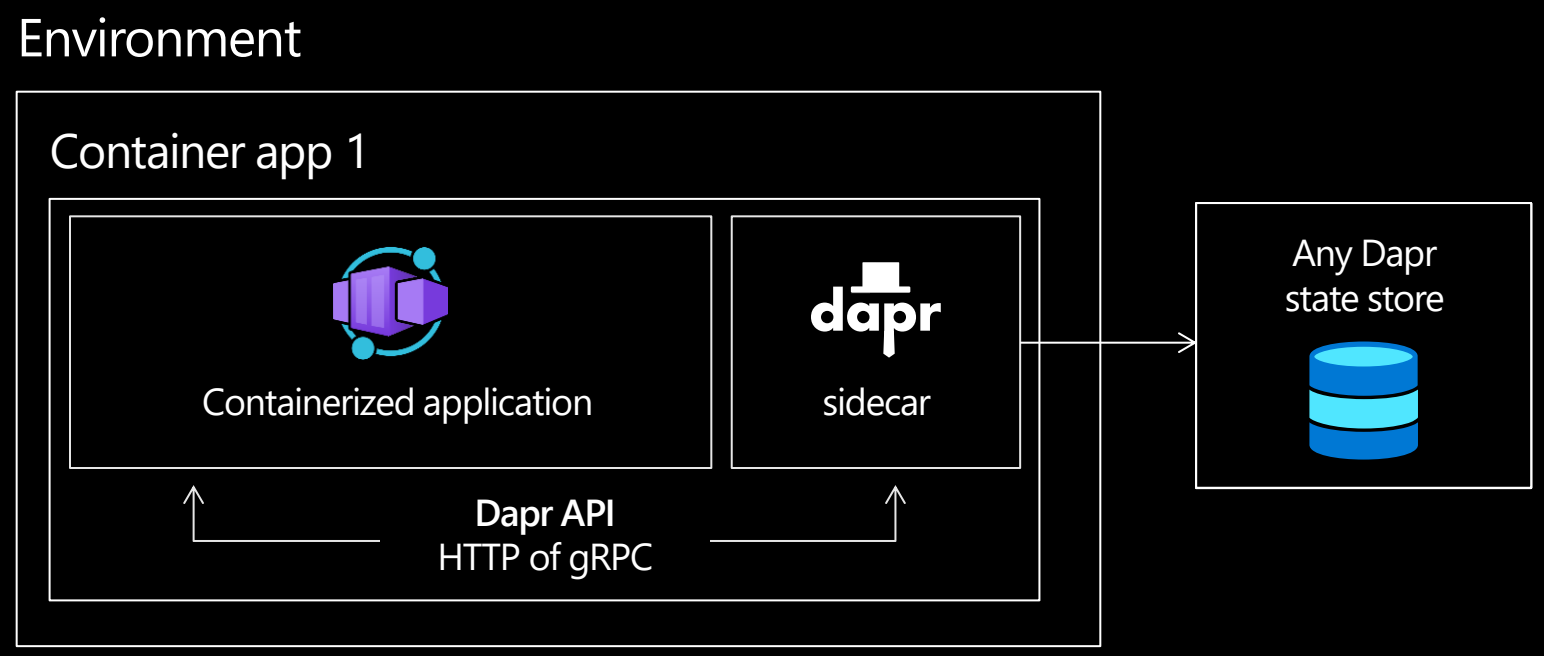
Service to service invocation

Fully managed Dapr APIs provide a rich set of capabilities and productivity gains



State management

Dapr provide apps with state management capabilities for CRUD operations, transactions and more



POST

```
http://localhost:3500/v1.0/state/corpdb
```

```
[{
   "key": “fruit",
   "value": “Orange"
}]
```

Demo

Scaling and using the Kubernetes Event Driven Autoscaling (KEDA)

Application autoscaling made simple

Open-source, extensible, and vendor agnostic

KEDA

Kubernetes-based Event Driven Autoscaler

Drive the scaling of any container based on a growing list of 35+ event sources, known as: scalers

Metrics Adapter | Controller | Scaler

Event-driven

Intelligently scale your event-driven applications

Built-in scalers

Out-of-the-box scalers for various vendors, databases, messaging systems, telemetry systems, CI/CD, and more

Vendor-agnostic

Support for triggers across variety of cloud providers & products

Rich capabilities

Bring rich scaling to every workload

keda.sh

CLOUD NATIVE COMPUTING FOUNDATION

Scaling

HTTP

```
{
 "name": "http-rule",
  "http": {
   "metadata": {
    "concurrentRequests": 50
   }
  }
}
```

Event-driven

artemis-queue, kafka, aws-cloudwatch, aws-kinesis-stream, aws-sqs-queue, azure-blob, azure-eventhub, azure-servicebus, azure-queue, cron, external, gcp-pubsub, huawei-cloudeye, ibmmq, influxdb, mongodb, mssql, mysql, postgresql, rabbitmq, redis, redis-streams, selenium-grid, solace-event-queue, ..

CPU

```
{
 "name": "cpu-rule",
  "custom": {
   "type": "cpu",
    "metadata": {
     "type": "Utilization",
      "value": "50"
   }
  }
}
```

Memory

```
{
 "name": "mem-rule",
  "custom": {
   "type": "memory",
    "metadata": {
     "type": "AverageValue",
      "value": "512"
   }
  }
}
```

Support for scale to zero and specifying minimum/maximum replicas

Support for specifying minimum/maximum replicas

Get started with Azure Container Apps today

https://aka.ms/containerapps

Build your apps in Azure

Microsoft