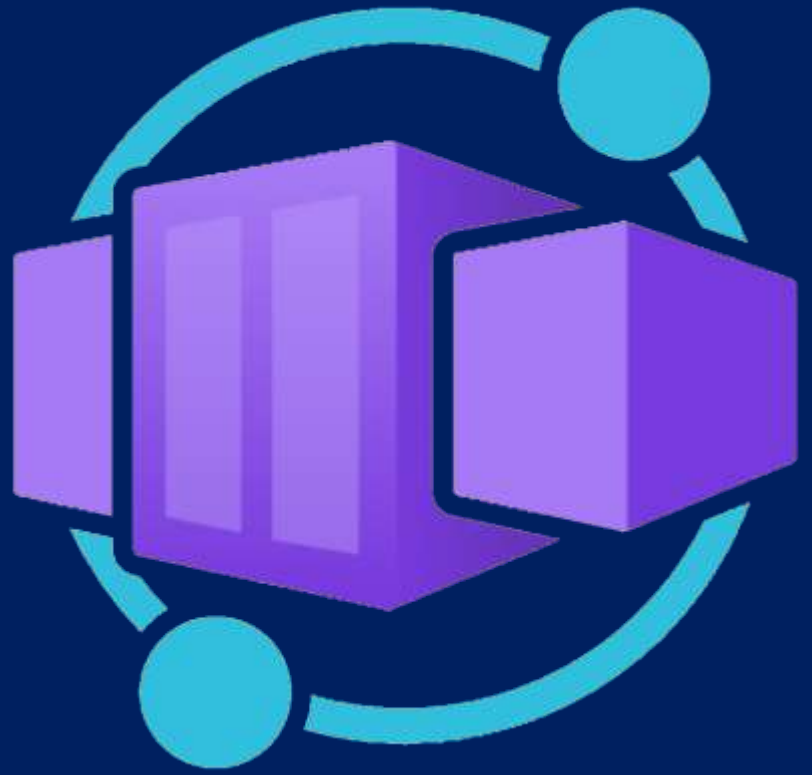
Microsoft

Azure Container Apps

Overview



Kazumi Hirose
Developer Program Manager , Azure
Subject Matter Expert
Microsoft Japan

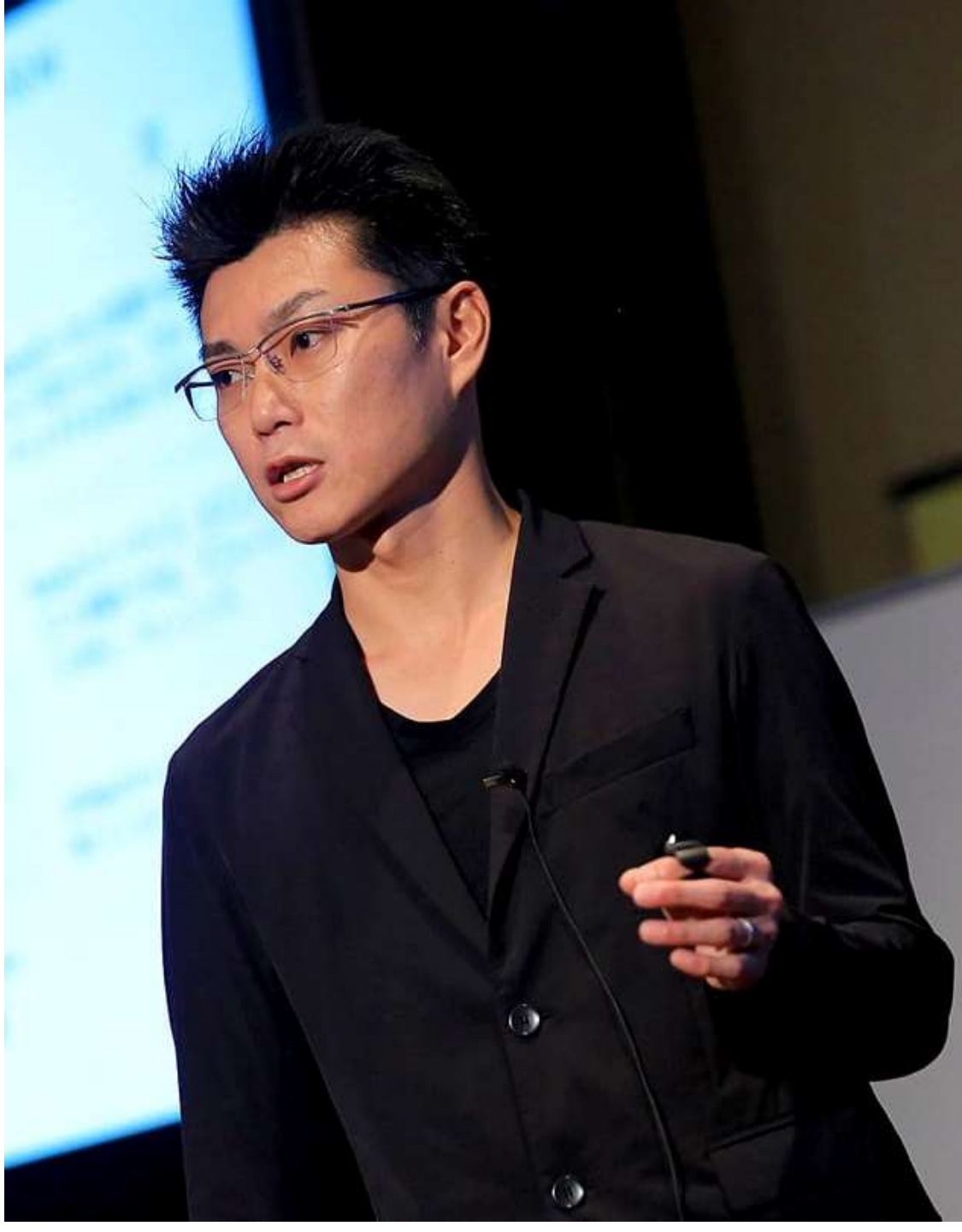
Kazumi Hirose
日本マイクロソフト株式会社
Azure Senior Product Manager / Azure Subject Matter Expert
https://www.linkedin.com/in/kazumihirose/

(社)日本医師会総合政策研究機構 客員研究員、アイレット株式会社 クラウドパック事業部 にて、シニアアーキテクトとして、主にAWSを用いたアーキティクチャを構成する傍ら、Microsoft MVP Azureとして双方のクラウドの知見を有し、2015年12月から日本Microsoft株式会社に在籍、Azure Technology Solutions Professionalで経験を積み、開発テクノロジ、インフラストラクチャ、ネットワーク、データ分析テクノロジなどの幅広い知識を有しています。

エンタープライズ企業顧客を主体に、同社のクラウドであるMicrosoft Azureに関する技術・設計支援・アーキテクチャ提供、技術支援と指導を行っています。

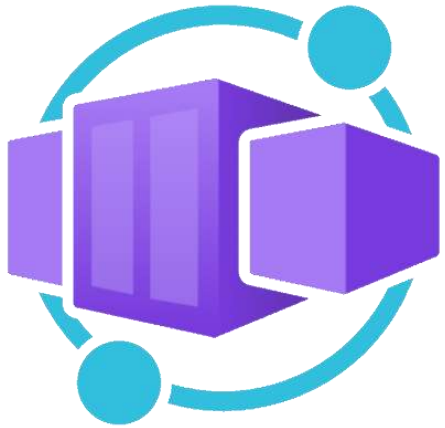
また、現在は開発者へのマーケティングチームの一員として、Azureを含むMicrosoft Cloudのクラウド開発者・インフラエンジニアの成功に焦点を当て、指導や情報提供活動を行っています。

2018年には、Microsoft Founders Award、2020年にはMicrosoft Gold Club Awardを受賞しています。



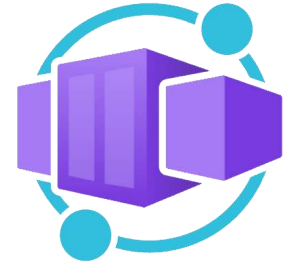
Azure Container Apps

最新のアプリケーションとマイクロサービスを構築するための、新しいサーバーレスコンテナプラットフォーム



AKS, KEDA, Dapr, and Envoy を基盤として、構築

Azure Container Apps



“Azure Container Apps は、どんなプログラミングモデルであるか、ランタイムであるかに関係なく、すべてのコンテナにパッケージされたコードを実行することができます。”

- コンテナーを実行する利点を享受しつつ、クラウド インフラと複雑なコンテナー オーケストレーターの管理を不要にします。
- サーバレス（ゼロからのスケールをサポート）
- HTTPリクエストに伴うスケール、イベント、もしくは、バックグラウンド処理のように常時実行
- イングレス、サービス間の通信の自動暗号化
- AKS, KEDA, Dapr, and Envoy を基に構築

Azure Container Apps: Example scenarios

PUBLIC API ENDPOINTS

HTTP requests are split between two versions of the container app where the first revision gets 80% of the traffic, while a new revision receives the remaining 20%.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP requests.

BACKGROUND PROCESSING

A continuously-running background process that transforms data in a database.

AUTO-SCALE CRITERIA

Scaling is determined by the level of CPU or memory load.

EVENT-DRIVEN PROCESSING

A 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.

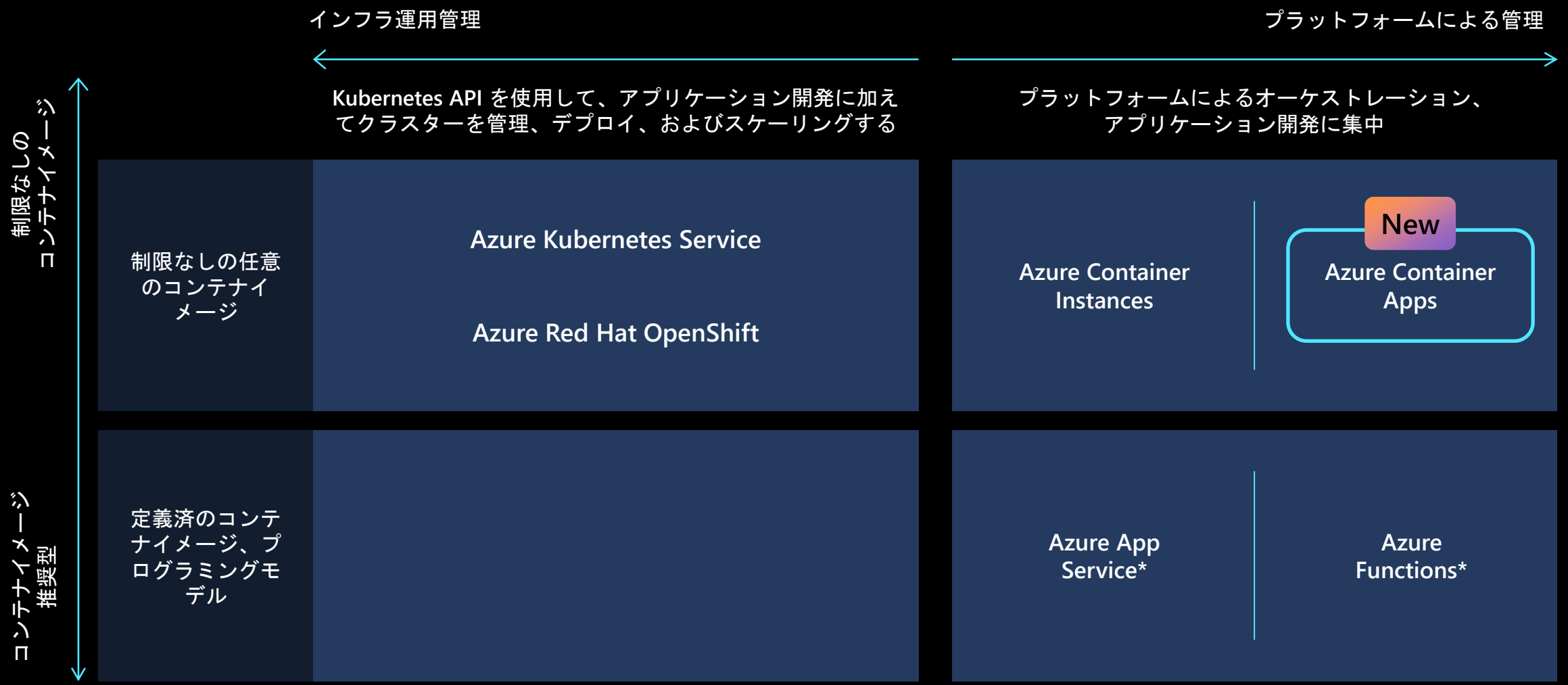
MICROSERVICES

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

AUTO-SCALE CRITERIA

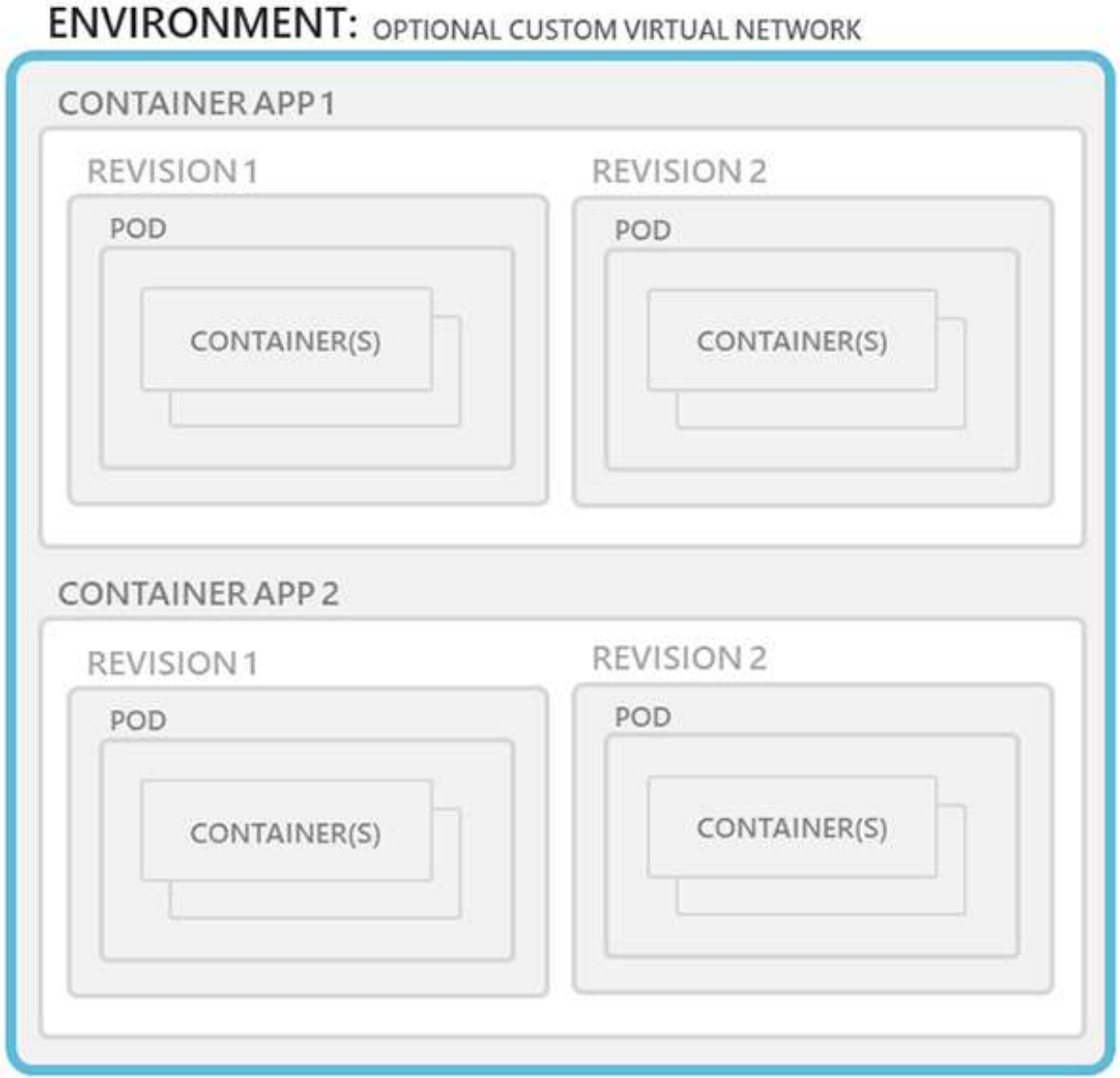
Individual microservices can scale according to any KEDA scale triggers.

Azure containers ポートフォリオ

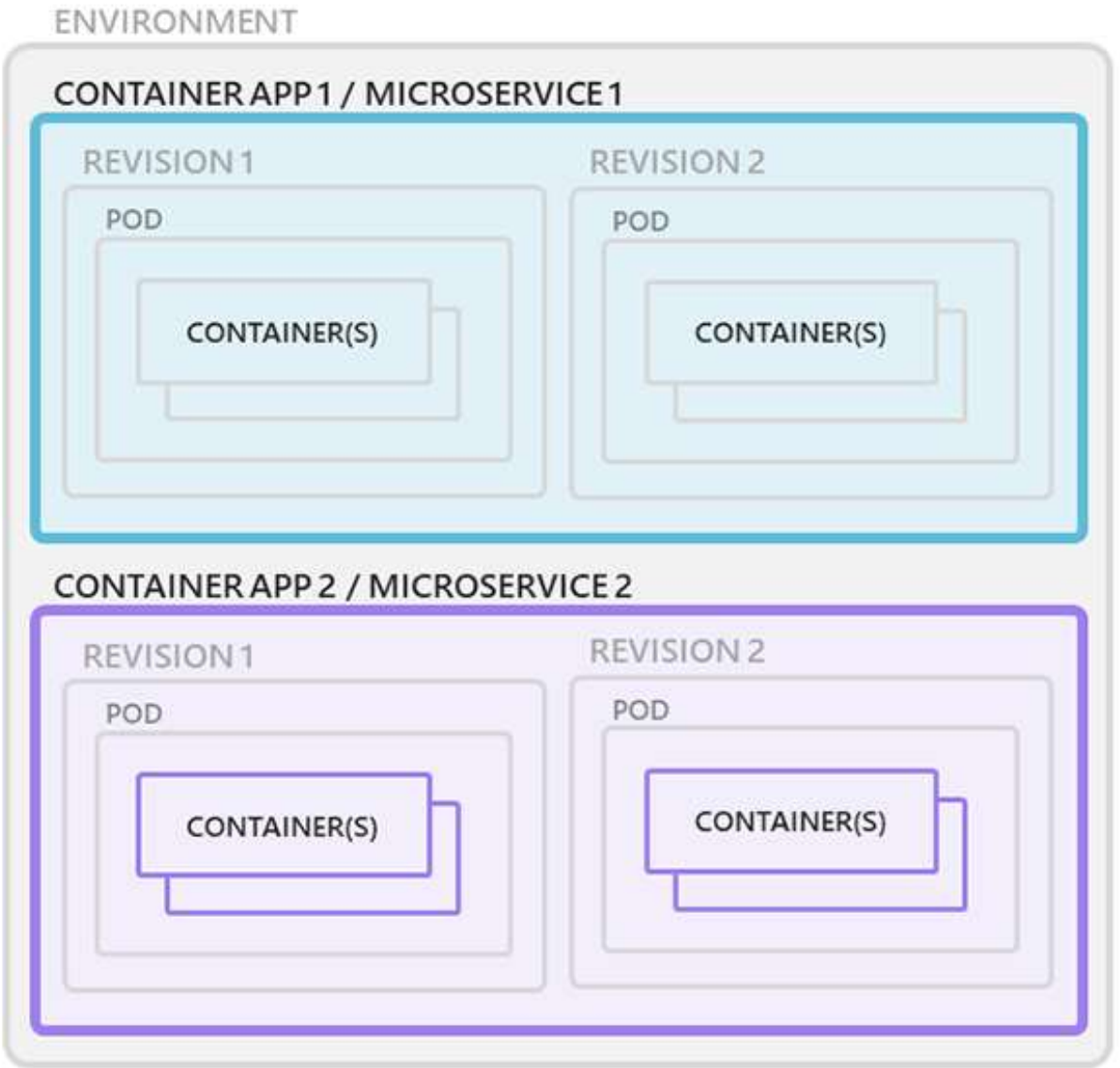


* When used with containers

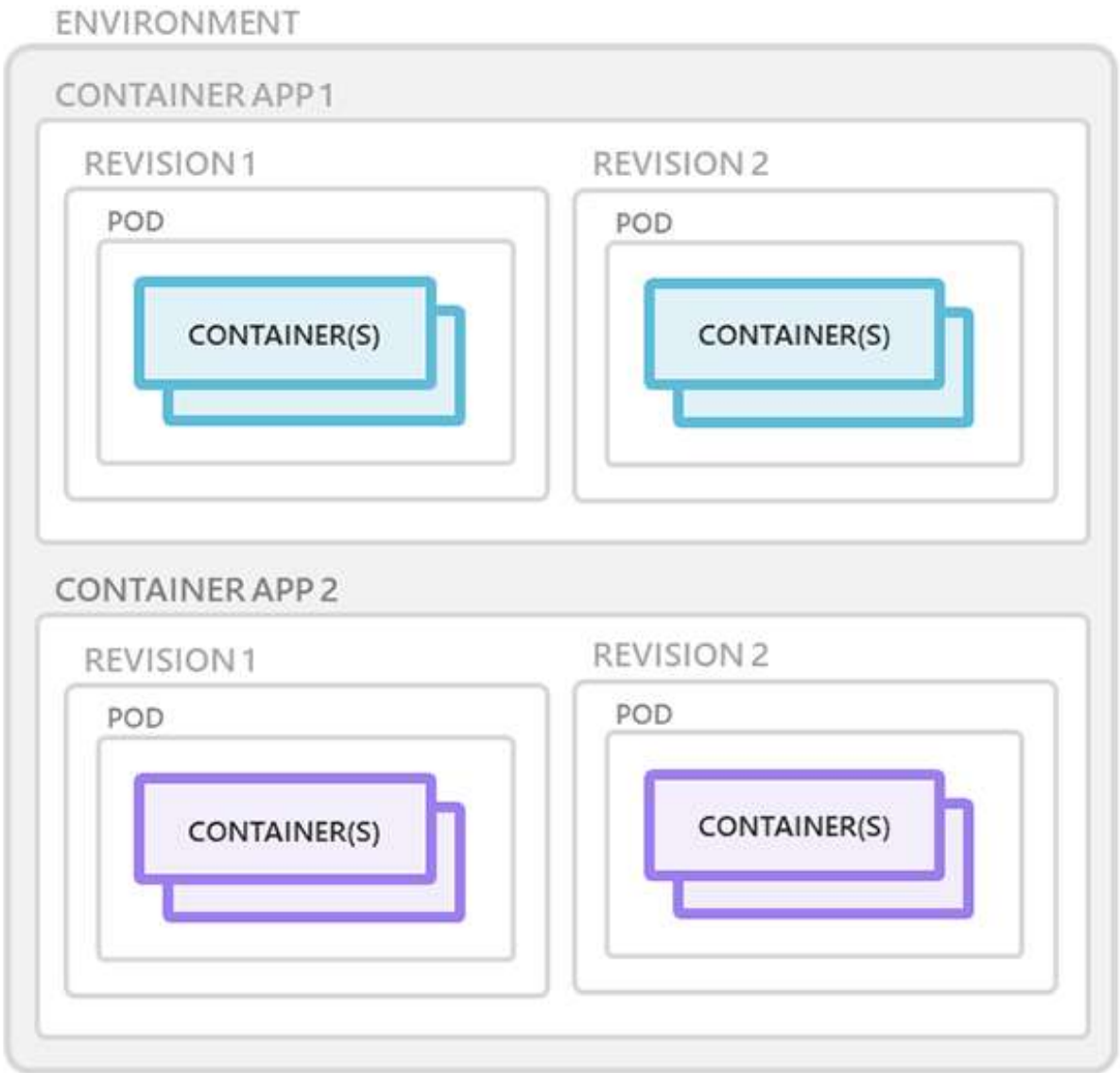
Environments are an isolation boundary around a collection of container apps.

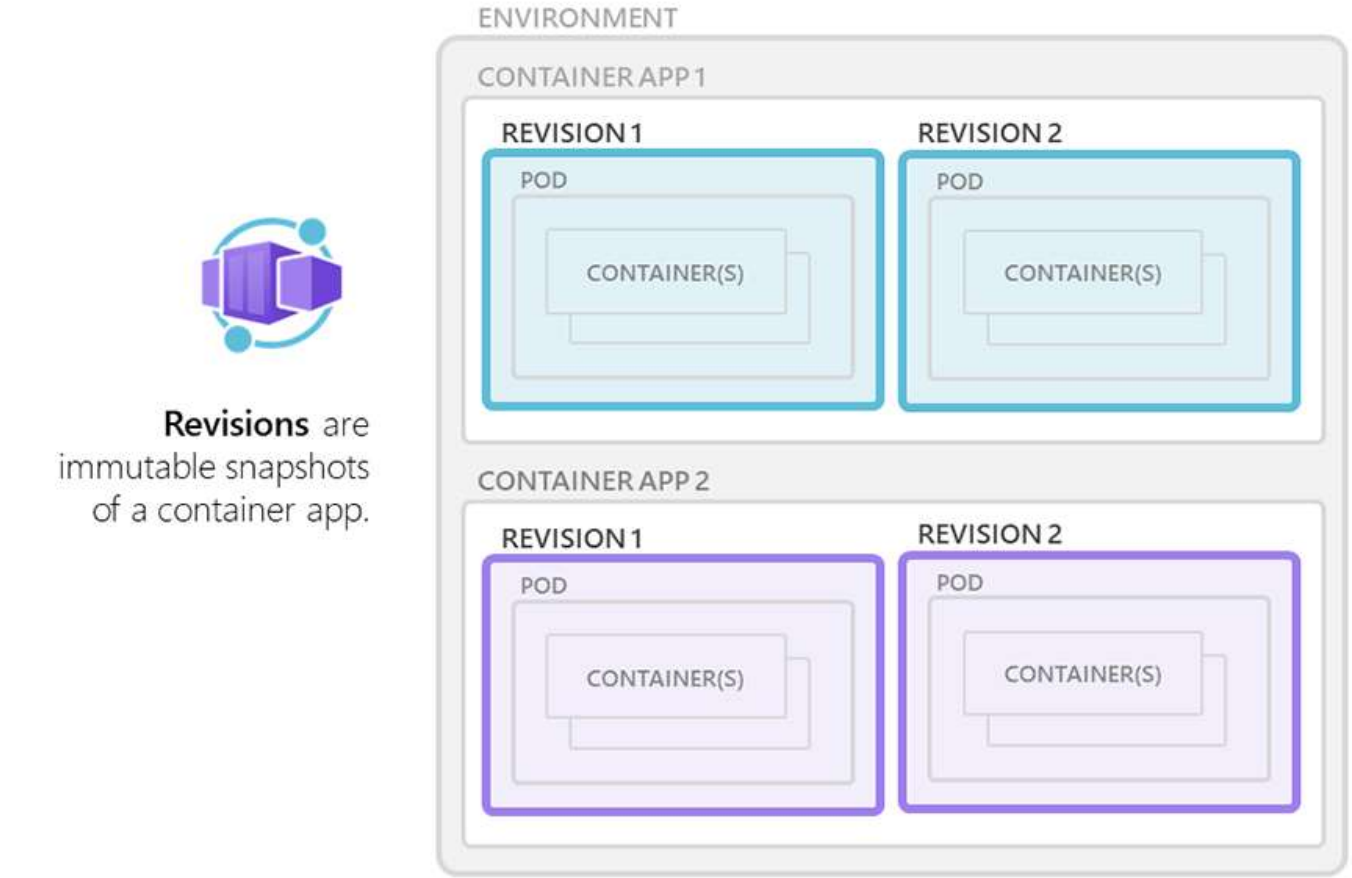


Container apps are deployed as **microservices**.

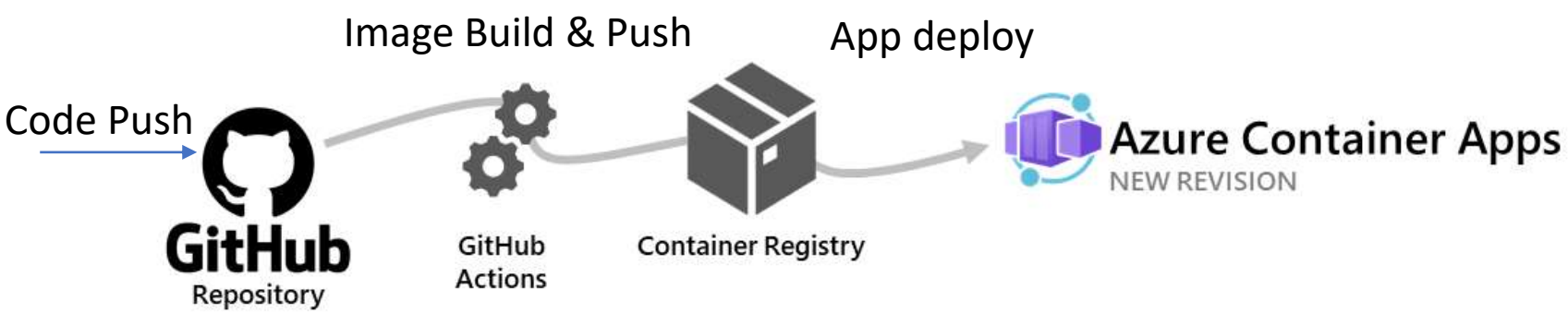


Containers for an Azure Container App are grouped together in pods inside revision snapshots.

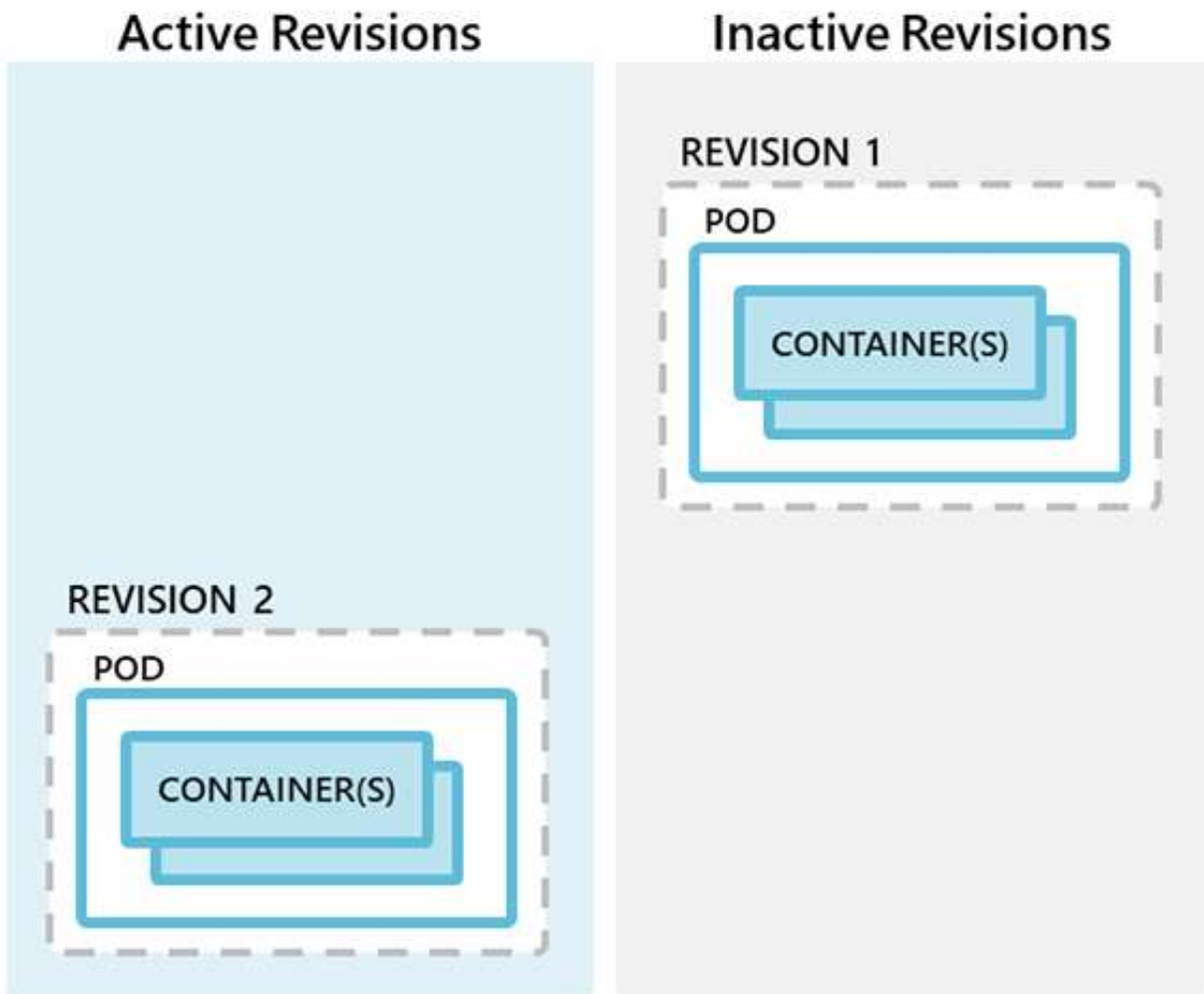




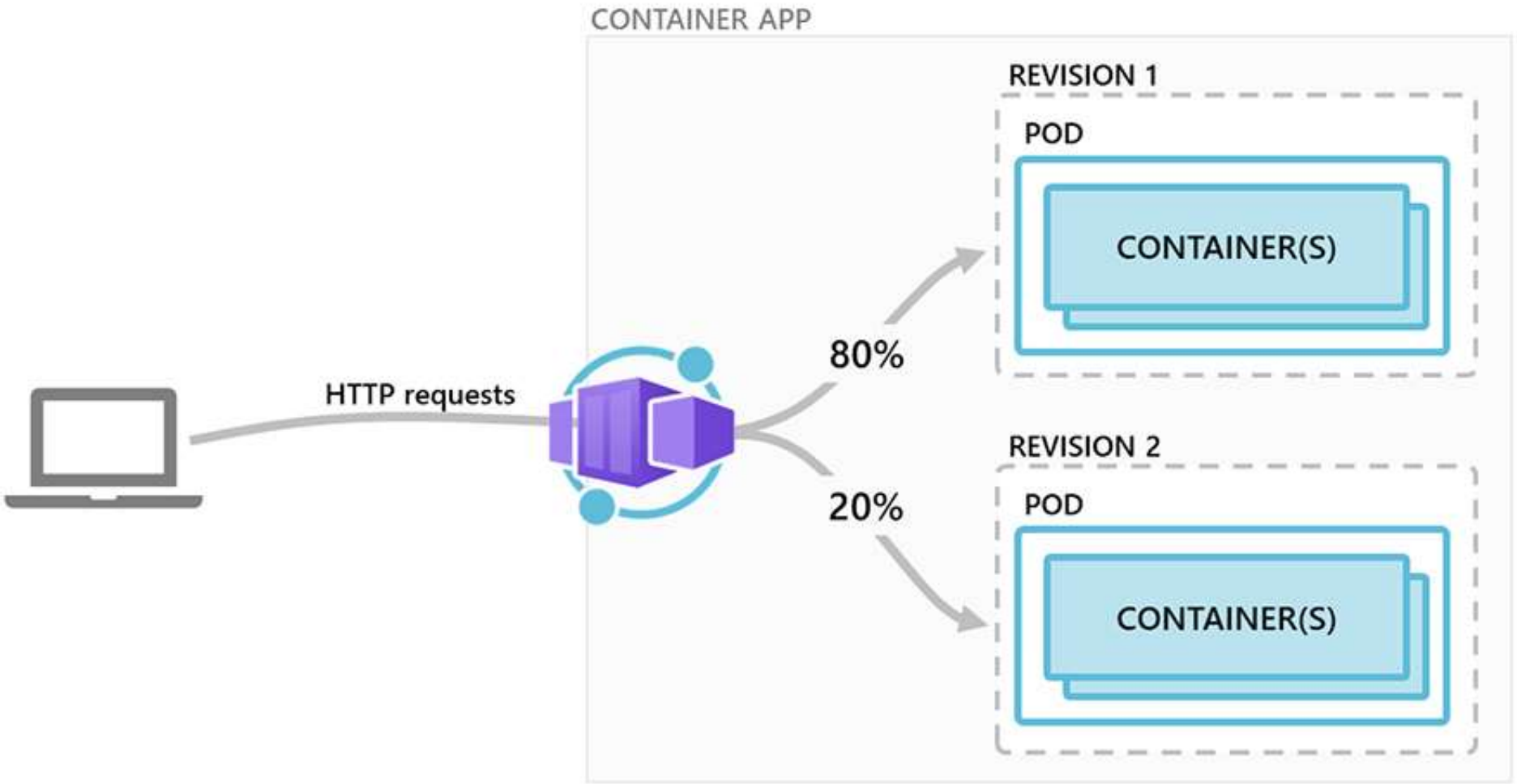
Revisions are immutable snapshots of a container app.



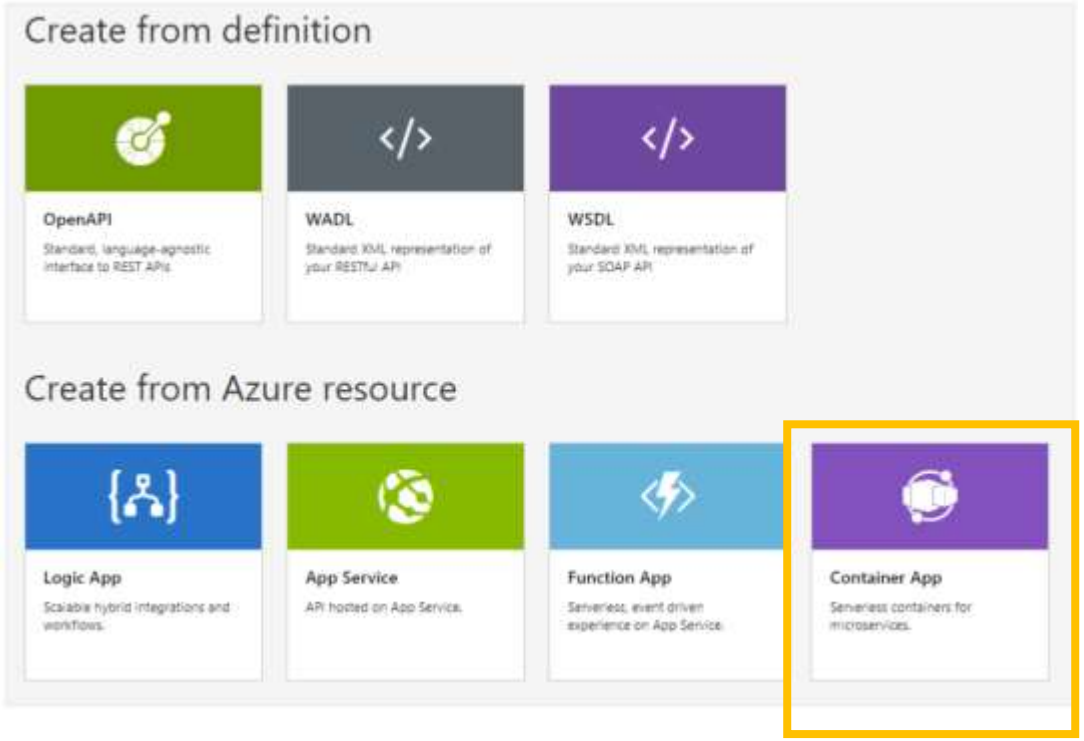
Once a revision is no longer needed, you can **deactivate** individual revisions, or choose to automatically deactivate old revisions.



Ingress traffic splitting



API Management Import

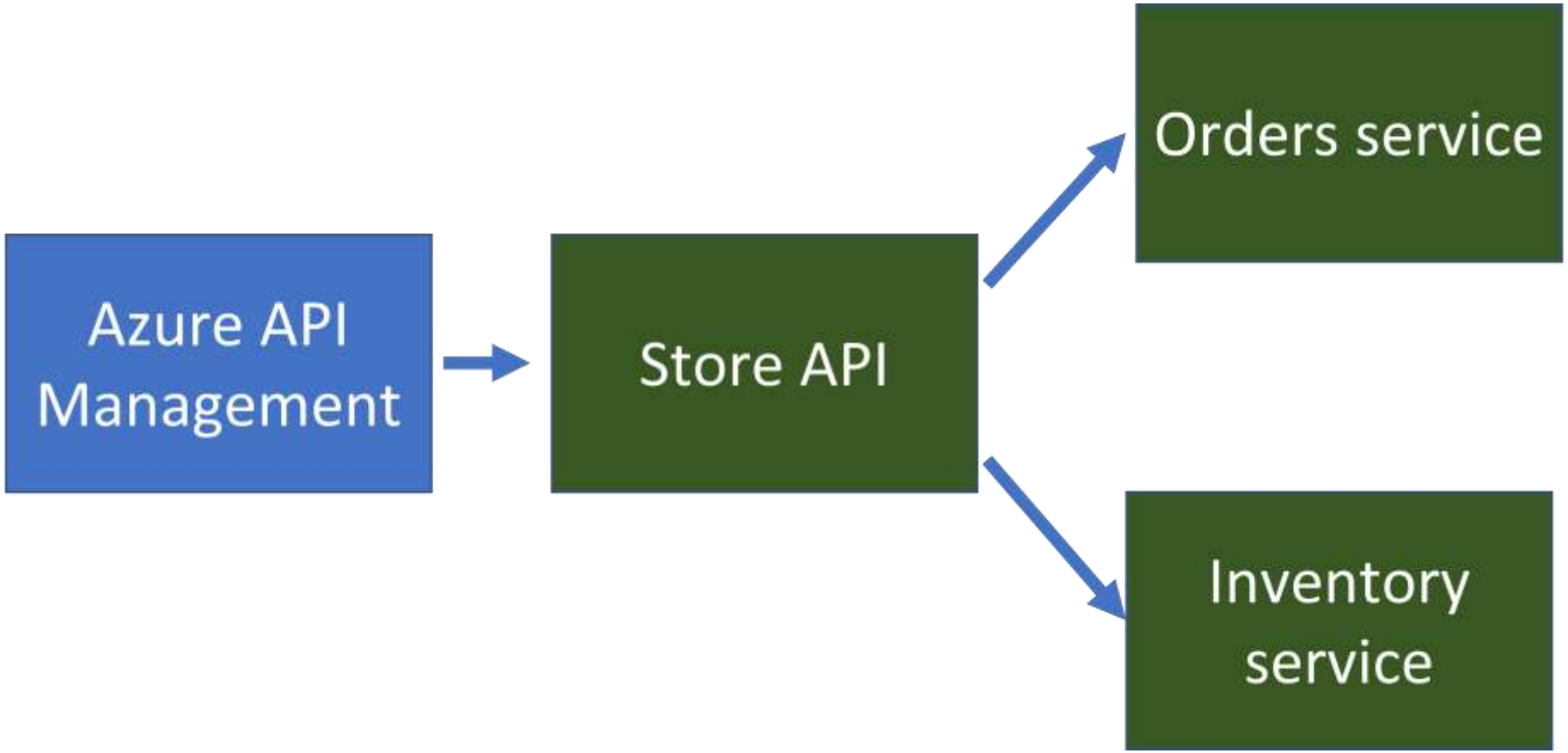


API Management will look in several locations for an OpenAPI Specification:

- The Container App configuration
- /openapi.json
- /openapi.yml
- /swagger/v1/swagger.json

https://docs.microsoft.com/en-us/azure/api-management/import-container-app-with-oas

Container Apps Sample



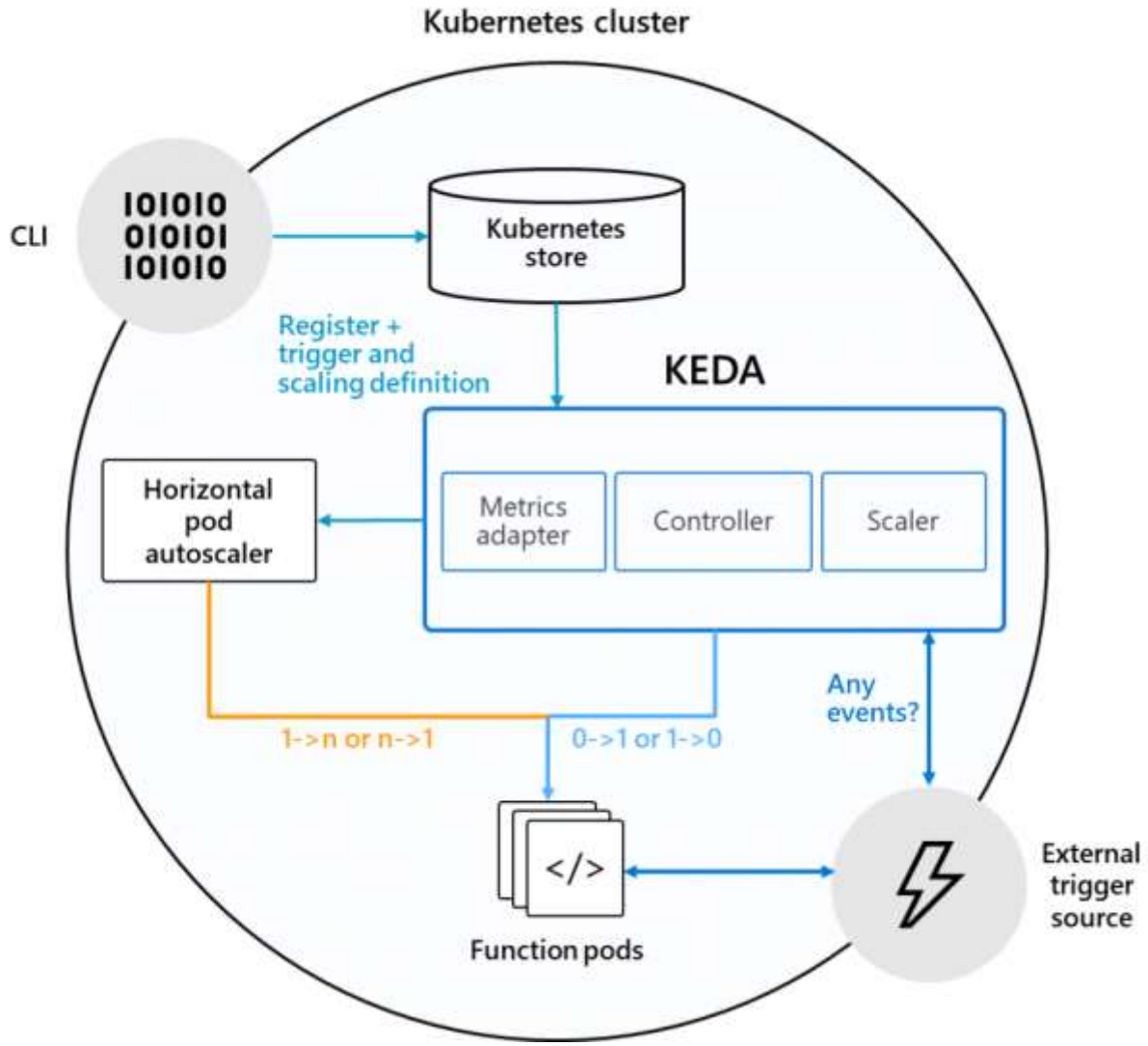
[Container App Store Microservice Sample]

KEDA - Kubernetes Event-driven Autoscaling

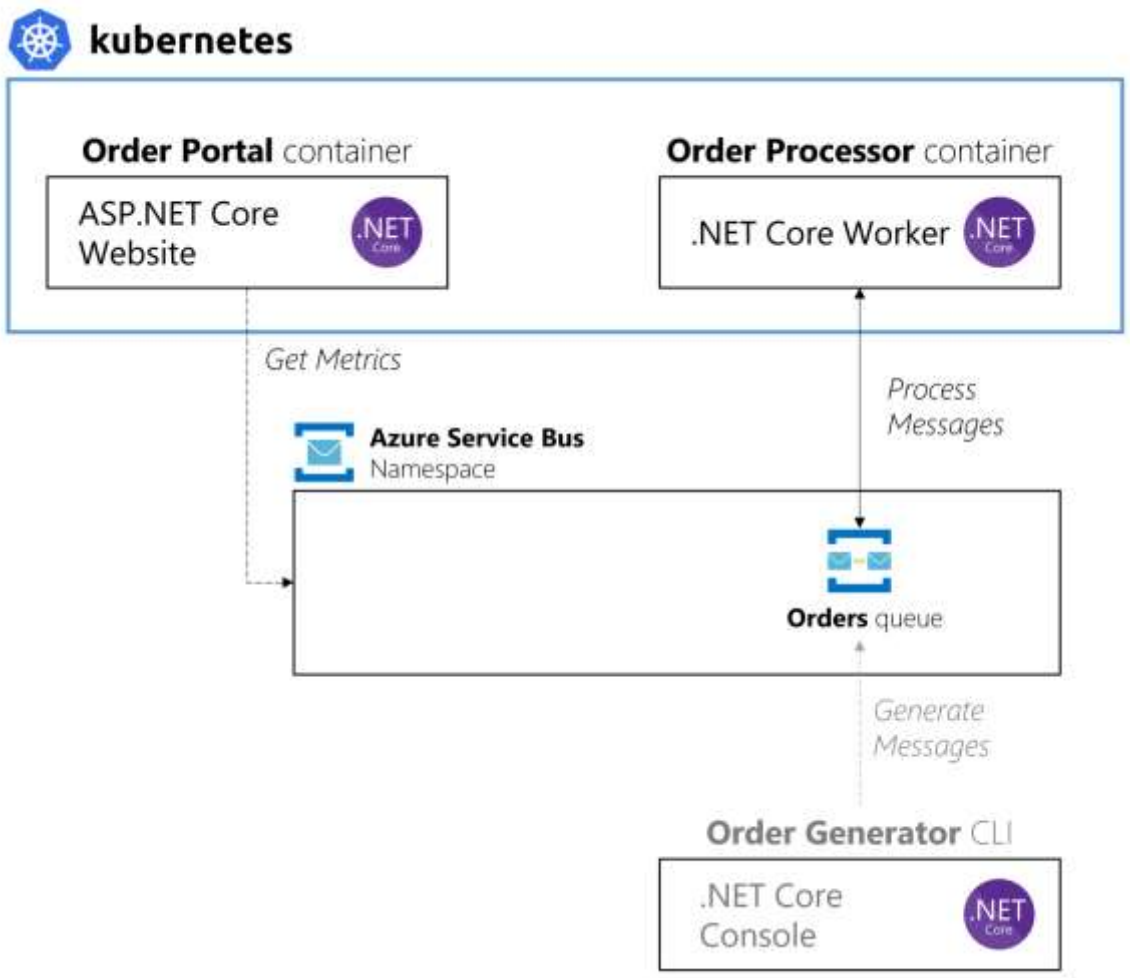
https://keda.sh/
https://www.cncf.io/projects/keda/

- Kubernetesでのイベント駆動型アプリケーションの構築をサポート
- Kubernetesの任意のコンテナのイベントソースからのきめ細かい自動スケール
- Kubernetes/OpenShift が動作する場所ならどこでも動作
- 水平ポッドオートスケーラー(HPA)とのネイティブ統合
- ジョブによるスケーリングをサポート
(1イベント->1ジョブ)
- ポッドはイベントソースに直接アクセスできます
- Kubernetes のコンテナーを介した Azure Functions の新しいホスティング オプション
- Red Hat との共同構築
- CNCFインキュベーションプロジェクト

- Architecture



Example




```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: order-processor-scaler
  labels:
    app: order-processor
    name: order-processor
spec:
  scaleTargetRef:
    name: order-processor
  # minReplicaCount: 0 Change to define how many minimum replicas you want
  maxReplicaCount: 10
  triggers:
  - type: azure-servicebus
    metadata:
      queueName: orders
      queueLength: '5'
    authenticationRef:
      name: trigger-auth-service-bus-orders
```

```yaml
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: trigger-auth-service-bus-orders
spec:
  secretTargetRef:
  - parameter: connection
    name: secrets-order-management
    key: servicebus-order-management-connectionstring
```

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

KEDA – Event Sources and Scalers

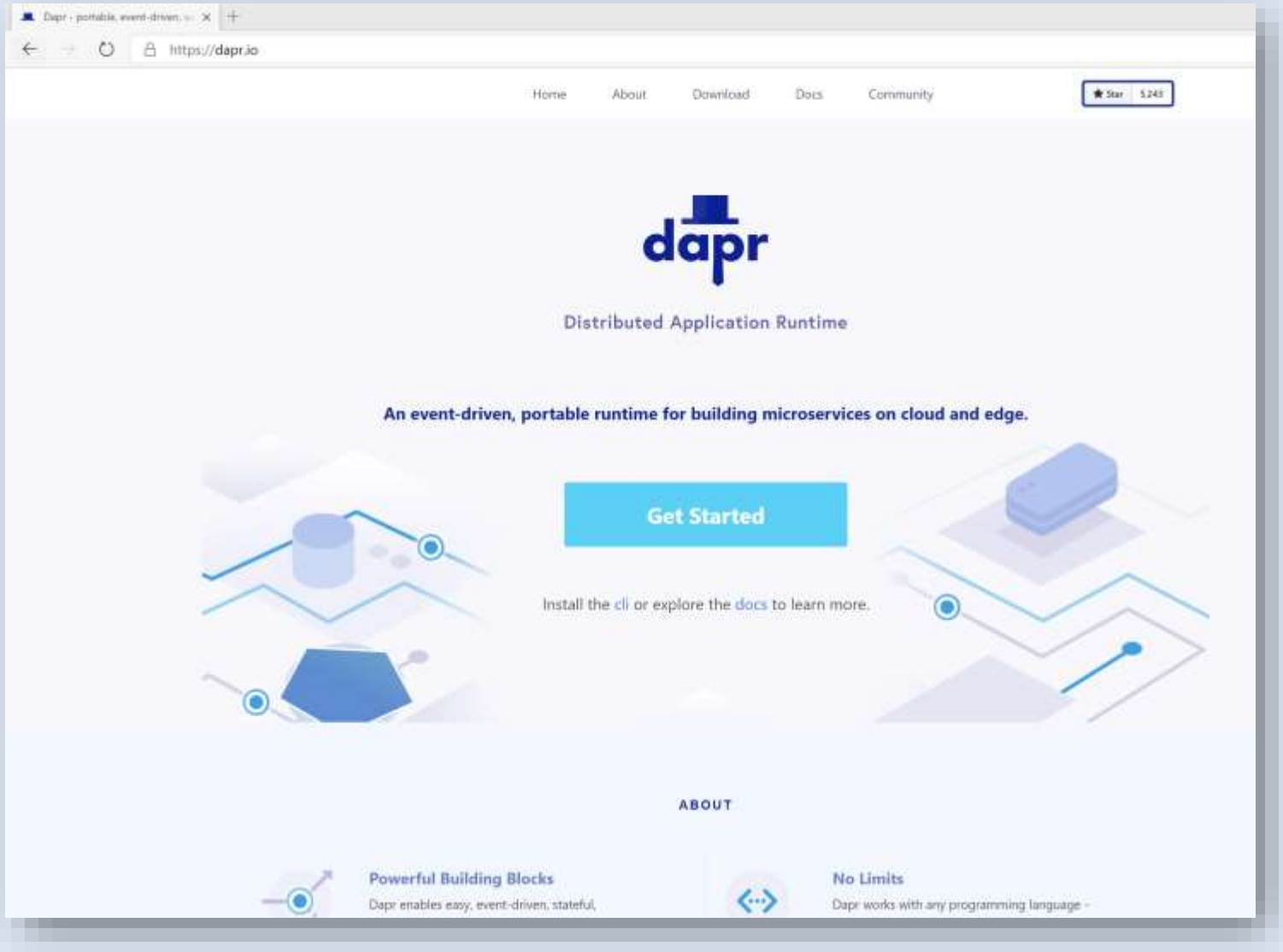
Currently available scalers for KEDA

- ActiveMQ Artemis
- Apache Kafka
- AWS CloudWatch
- AWS Kinesis Stream
- AWS SQS Queue
- Azure Blob Storage
- Azure Event Hubs
- Azure Log Analytics
- Azure Monitor
- Azure Pipelines
- Azure Service Bus
- Azure Storage Queue
- CPU
- Cron
- External
- External Push
- Google Cloud Platform Pub/Sub
- Huawei Cloudeye
- IBM MQ
- InfluxDB
- Kubernetes Workload
- Liiklus Topic
- Memory
- Metrics API
- MongoDB
- MSSQL
- MySQL
- NATS Streaming
- OpenStack Metric
- OpenStack Swift
- PostgreSQL
- Prometheus
- RabbitMQ Queue
- Redis Lists
- Redis Lists (supports Redis Cluster)
- Redis Streams
- Redis Streams (supports Redis Cluster)
- Selenium Grid Scaler
- Solace PubSub+ Event Broker

Distributed Application Runtime

ポータブル, イベント駆動, クラウドとエッジ双方に分散アプリケーションを構築する為のランタイム

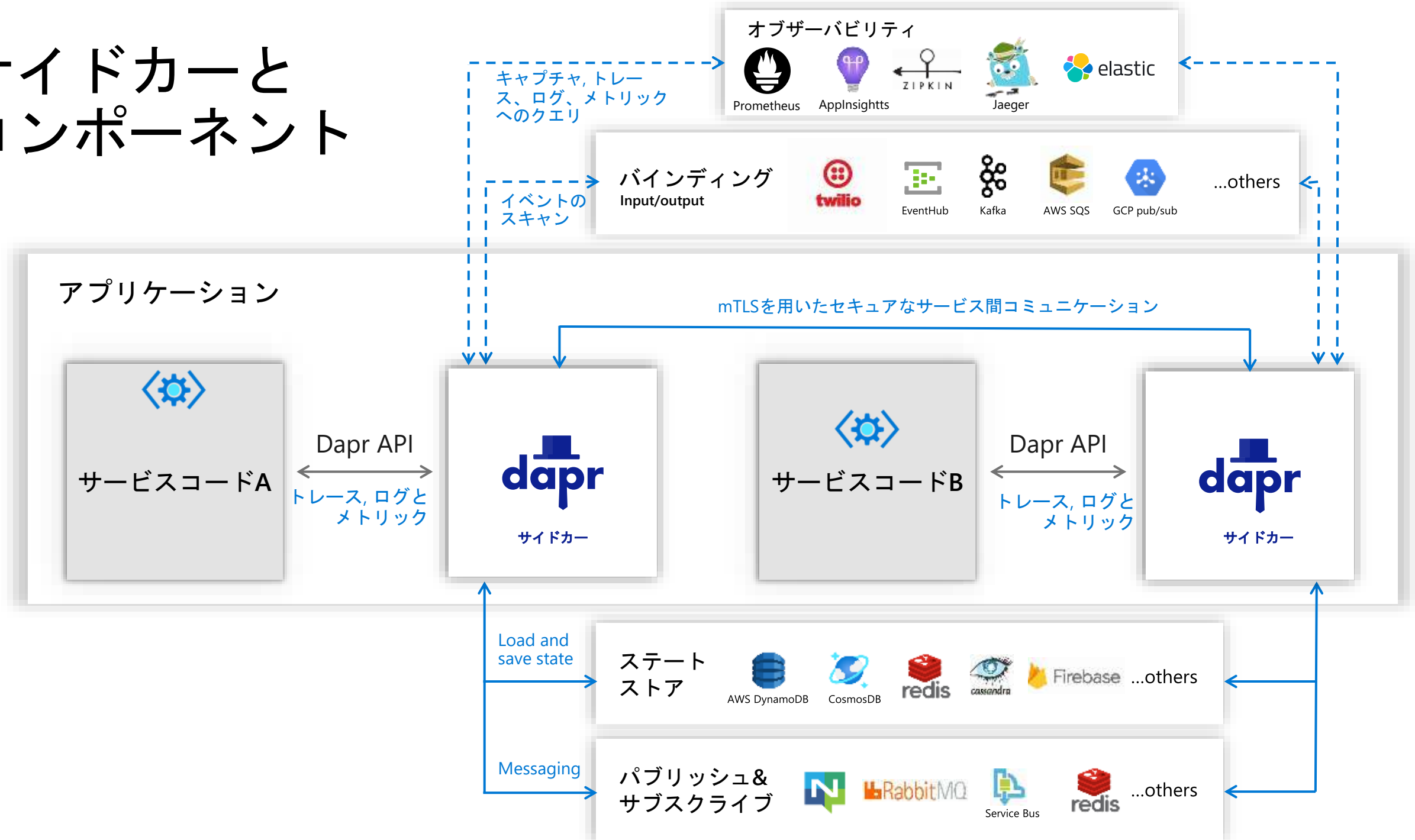
https://dapr.io



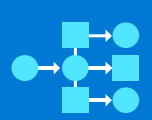
Microservice ビルディングブロック

- ユーザのサービスコードから http/gRPC で標準APIにアクセスできます
  http://localhost:3500/v1.0/invoke/cart/method/neworder
  http://localhost:3500/v1.0/state/inventory/item67
- 各サービスが実行する際に、動的にロードされる“サイドカーライブラリ”として動作します。

サイドカーとコンポーネント



Microservice building blocks



サービス間呼び出し

Perform direct, secure, service-to-service method calls

ステート管理

Create long running, stateless and stateful services

パブリッシュ/サブスクライブ

Secure, scalable messaging between services

バインディング (input/output)

Trigger code through events from a large array of inputs

Input and output バインディング to external resources including databases and queues

アクターサービスモデル

Encapsulate code and data in reusable actor objects as a common microservices design pattern

オブザーバビリティ(観測可能性)

See and measure the message calls across components and networked services

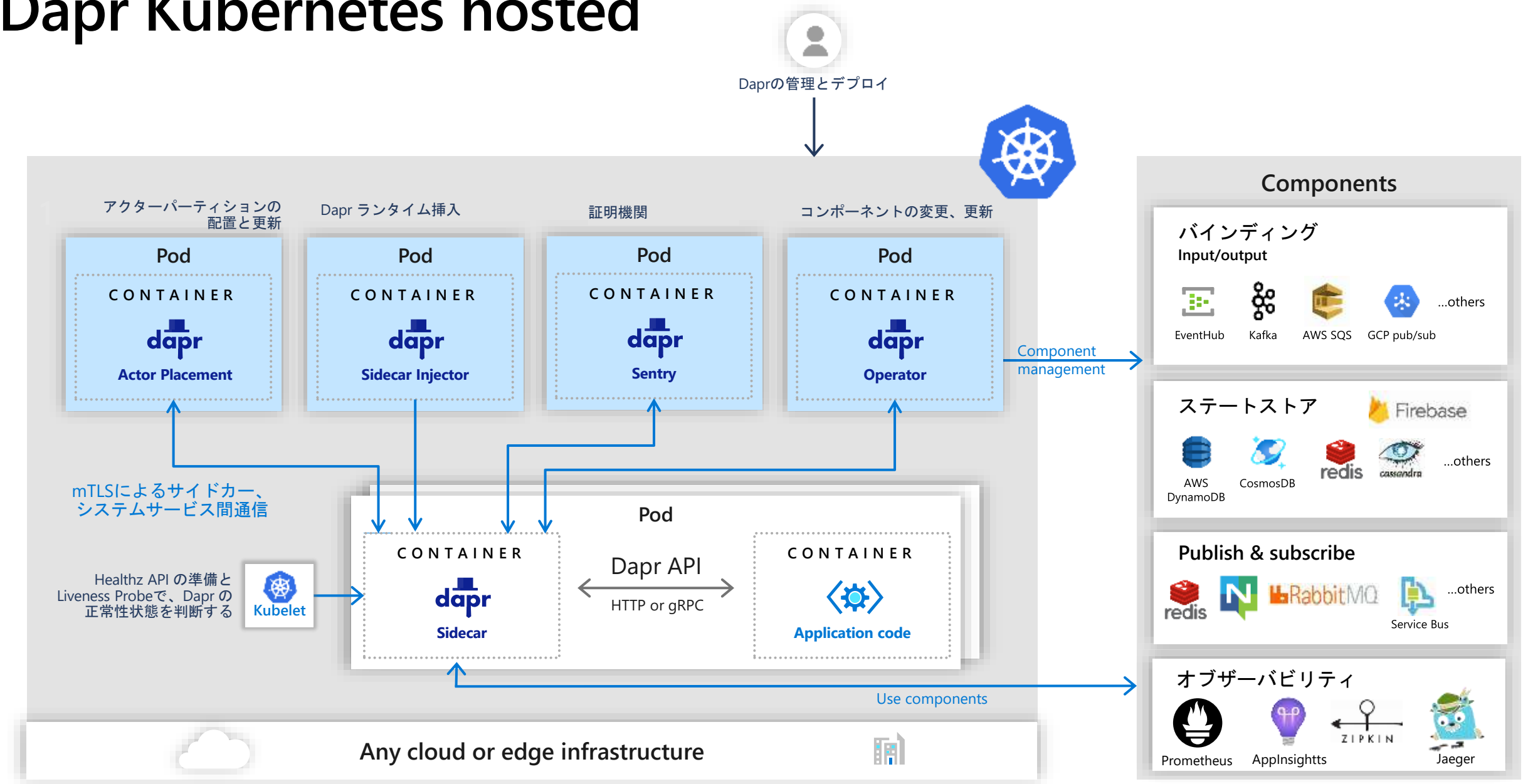
Secrets

Securely access secrets from your application

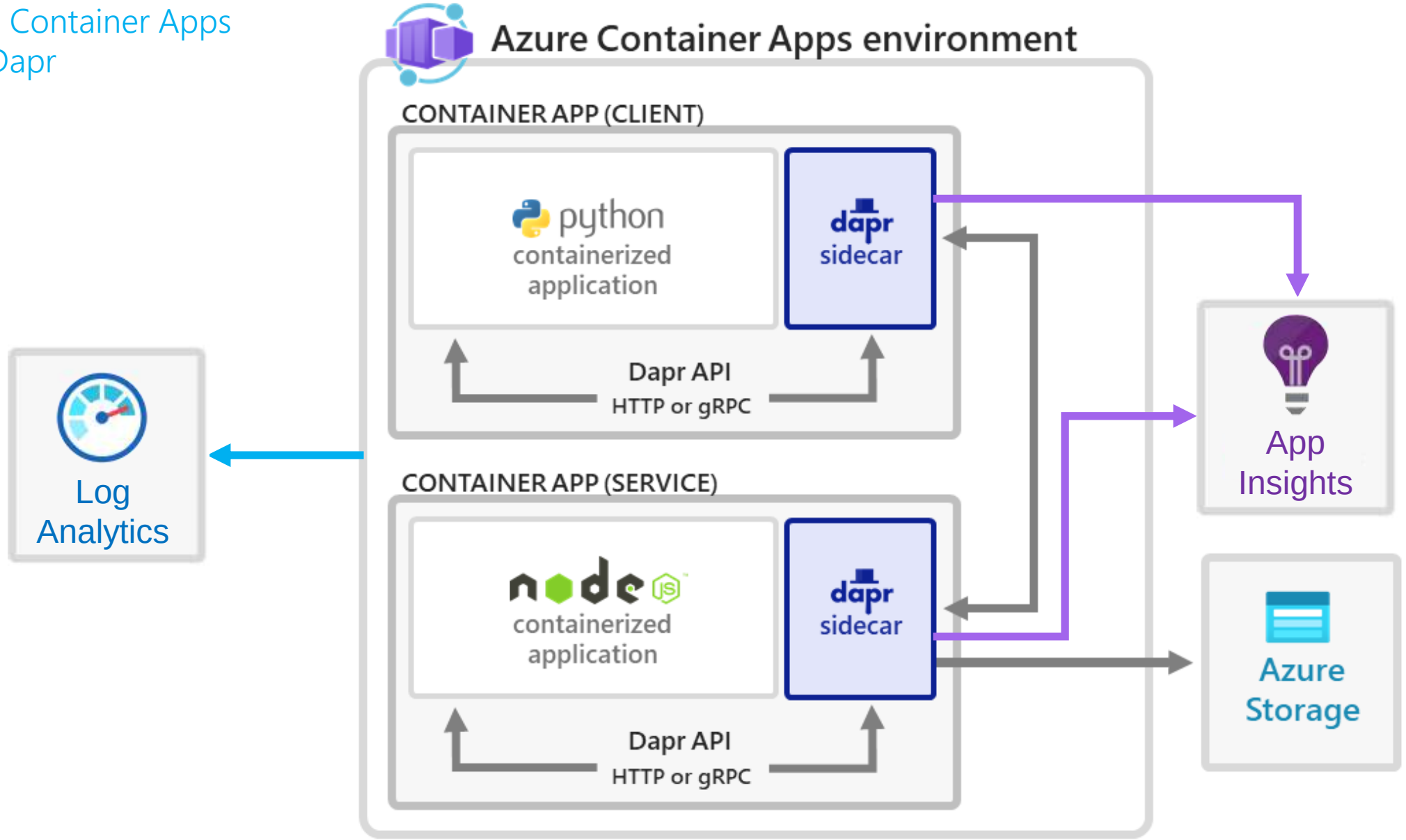
Dapr: 分散アプリケーションランタイム

どの言語、どのフレームワークの組み合わせでもアプリケーションの開発が可能

Dapr Kubernetes hosted



Azure Container Apps with Dapr



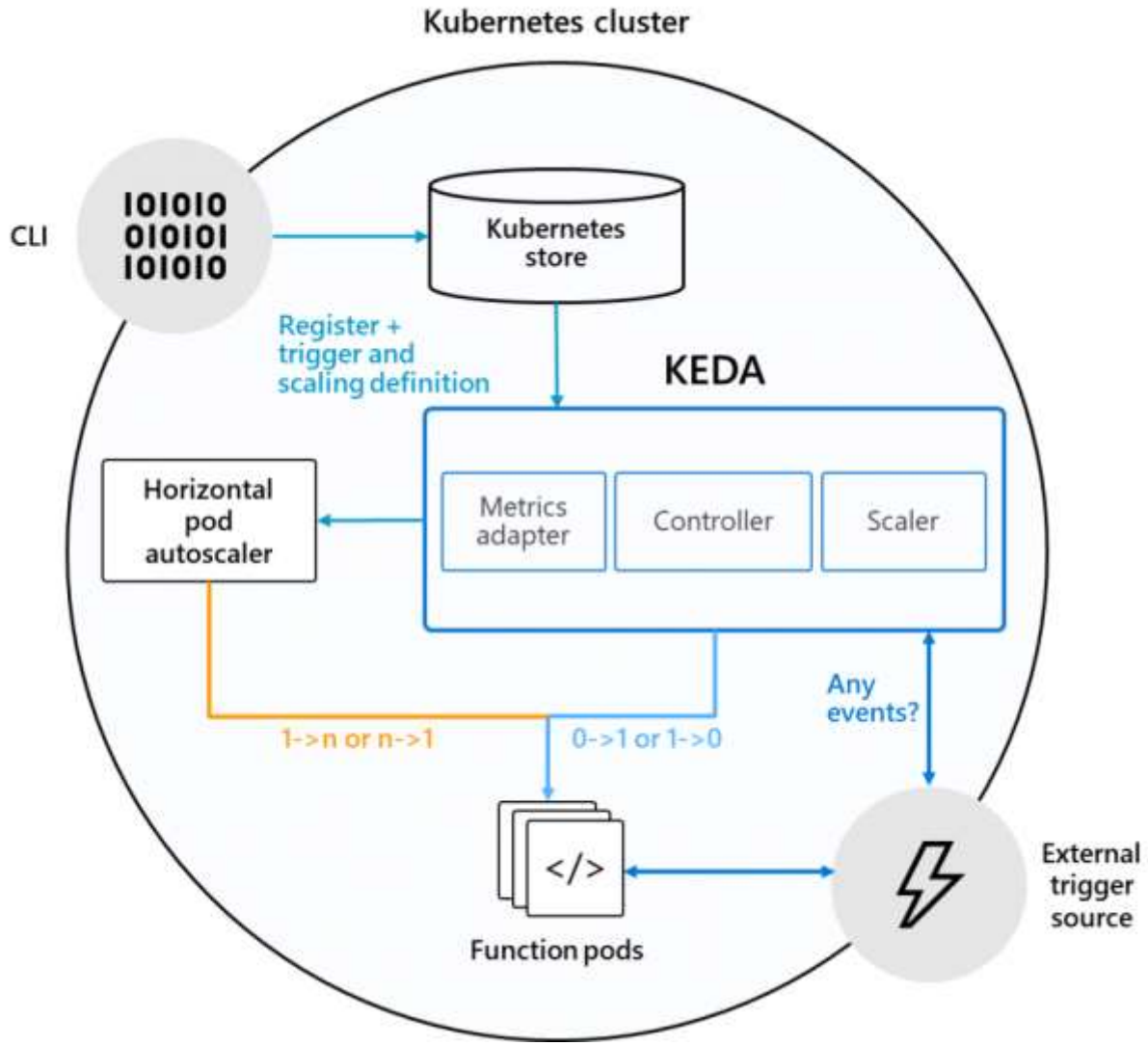
https://github.com/clarenceb/tutorial-dapr-cli

KEDA - Kubernetes Event-driven Autoscaling

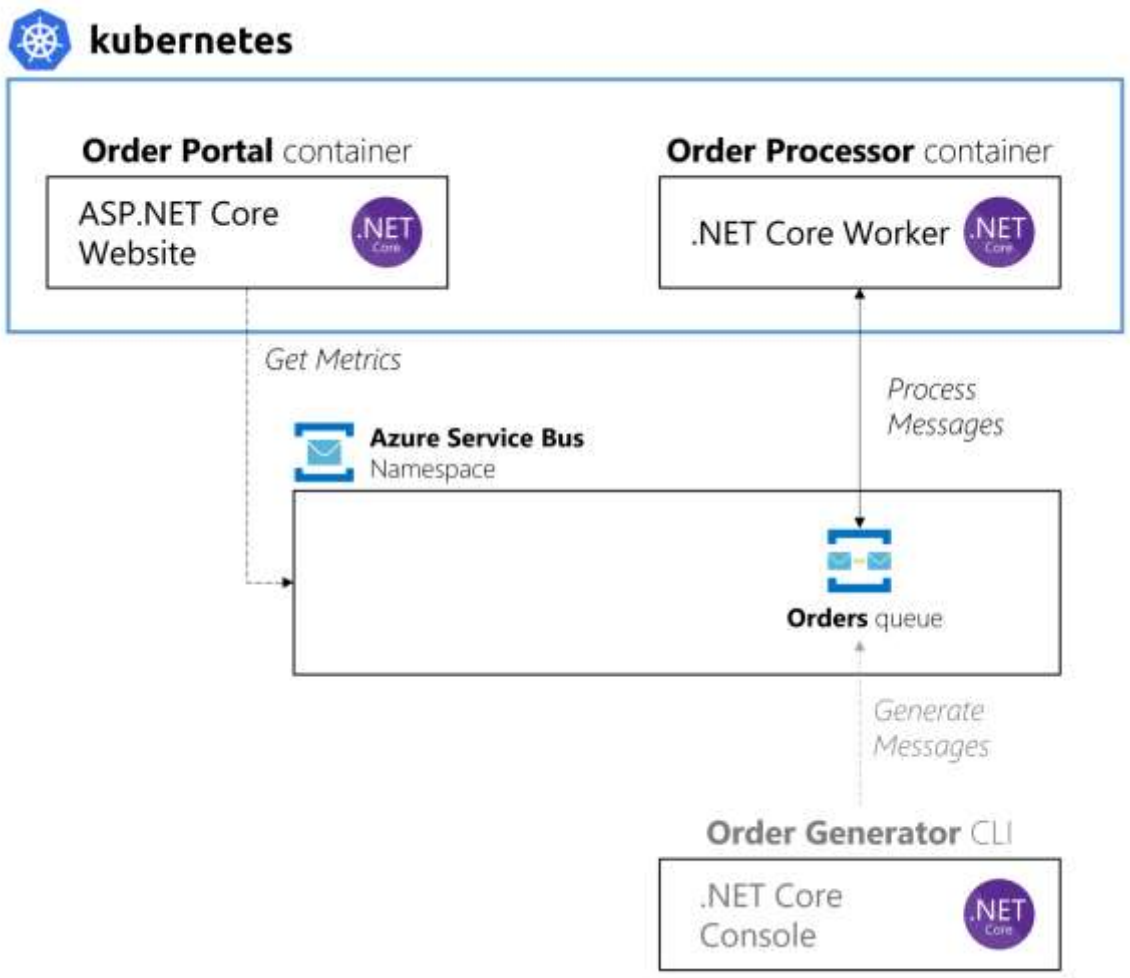
https://keda.sh/
https://www.cncf.io/projects/keda/

- Supports building event-driven applications in Kubernetes
- Fine grained autoscaling off of event sources for *any* container in Kubernetes
- Runs *anywhere* Kubernetes/OpenShift runs
- Native integration with Horizontal Pod Autoscaler (HPA)
- Supports scaling via Jobs (1 event -> 1 job)
- Pods get direct access to event sources
- New hosting option for Azure Functions via containers in Kubernetes
- Built in conjunction with Red Hat
- CNCF incubating project

Architecture



Example




```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: order-processor-scaler
  labels:
    app: order-processor
    name: order-processor
spec:
  scaleTargetRef:
    name: order-processor
  # minReplicaCount: 0 Change to define how many minimum replicas you want
  maxReplicaCount: 10
  triggers:
  - type: azure-servicebus
    metadata:
      queueName: orders
      queueLength: '5'
    authenticationRef:
      name: trigger-auth-service-bus-orders
```

```yaml
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: trigger-auth-service-bus-orders
spec:
  secretTargetRef:
  - parameter: connection
    name: secrets-order-management
    key: servicebus-order-management-connectionstring
```

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

ScaledObject CRD – Deployment, StatefulSets, Custom Resources

```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: {scaled-object-name}
spec:
  scaleTargetRef:
    apiVersion:    {api-version-of-target-resource}  # Optional. Default: apps/v1
    kind:          {kind-of-target-resource}         # Optional. Default: Deployment
    name:          {name-of-target-resource}         # Mandatory. Must be in the same na
    envSourceContainerName: {container-name}         # Optional. Default: .spec.templat
  pollingInterval:  30                               # Optional. Default: 30 seconds
  cooldownPeriod:   300                              # Optional. Default: 300 seconds
  idleReplicaCount: 0                                # Optional. Must be less than minR
  minReplicaCount:  1                                # Optional. Default: 0
  maxReplicaCount:  100                              # Optional. Default: 100
  fallback:                                          # Optional. Section to specify fal
    failureThreshold: 3                              # Mandatory if fallback section is
    replicas: 6                                      # Mandatory if fallback section is
  advanced:                                          # Optional. Section to specify adv
    restoreToOriginalReplicaCount: true/false        # Optional. Default: false
    horizontalPodAutoscalerConfig:                   # Optional. Section to specify HPA
      behavior:                                      # Optional. Use to modify HPA's sc
        scaleDown:
          stabilizationWindowSeconds: 300
          policies:
          - type: Percent
            value: 100
            periodSeconds: 15
  triggers:
  # {list of triggers to activate scaling of the target resource}
```

ScaledObject CRD – Job

```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledJob
metadata:
  name: {scaled-job-name}
spec:
  jobTargetRef:
    parallelism: 1                          # [max number of desired pods](https://ku
    completions: 1                          # [desired number of successfully finishe
    activeDeadlineSeconds: 600              #  Specifies the duration in seconds rela
    backoffLimit: 6                         # Specifies the number of retries before
    template:
      # describes the [job template](https://kubernetes.io/docs/concepts/workloads/cont
  pollingInterval: 30                       # Optional. Default: 30 seconds
  successfulJobsHistoryLimit: 5             # Optional. Default: 100. How many comple
  failedJobsHistoryLimit: 5                 # Optional. Default: 100. How many failed
  envSourceContainerName: {container-name}  # Optional. Default: .spec.JobTargetRef.te
  maxReplicaCount: 100                      # Optional. Default: 100
  scalingStrategy:
    strategy: "custom"                      # Optional. Default: default. Which Scali
    customScalingQueueLengthDeduction: 1    # Optional. A parameter to optimize custo
    customScalingRunningJobPercentage: "0.5"  # Optional. A parameter to optimize custo
    pendingPodConditions:                   # Optional. A parameter to calculate pend
      - "Ready"
      - "PodScheduled"
      - "AnyOtherCustomPodCondition"
  triggers:
  # {list of triggers to create jobs}
```

Triggers

Service Bus Trigger

```yaml
triggers:
- type: azure-servicebus
  metadata:
    # Required: queueName OR topicName and subscriptionName
    queueName: functions-sbqueue
    # or
    topicName: functions-sbtopic
    subscriptionName: sbtopic-sub1
    # Optional, required when pod identity is used
    namespace: service-bus-namespace
    # Optional, can use TriggerAuthentication as well
    connectionFromEnv: SERVICEBUS_CONNECTIONSTRING_ENV_NAME # This must be a connection
    # Optional
    messageCount: "5" # Optional. Count of messages to trigger scaling on. Default: 5 m
    cloud: Private # Optional. Default: AzurePublicCloud
    endpointSuffix: servicebus.airgap.example # Required when cloud=Private
```

Kafka Trigger

```yaml
triggers:
- type: kafka
  metadata:
    bootstrapServers: kafka.svc:9092
    consumerGroup: my-group
    topic: test-topic
    lagThreshold: '5'
    offsetResetPolicy: latest
    allowIdleConsumers: false
    version: 1.0.0
```

Prometheus Trigger

```yaml
triggers:
- type: prometheus
  metadata:
    # Required
    serverAddress: http://<prometheus-host>:9090
    metricName: http_requests_total
    query: sum(rate(http_requests_total{deployment="my-deployment"}[2m])) # Note: query
    threshold: '100'
```

Trigger Authentication (Env Var, Secret, Pod Identity, Vault)

Pod Identity Auth

```yaml
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: azure-servicebus-auth
spec:
  podIdentity:
    provider: azure
```

Secret Auth (connection string)

```yaml
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: mongodb-trigger
spec:
  secretTargetRef:
    - parameter: connectionString
      name: mongodb-secret
      key: connect
```

Secret Auth (bearer token)

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: keda-prom-secret
  namespace: default
data:
  bearerToken: "BEARER_TOKEN"
  ca: "CUSTOM_CA_CERT"
---
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: keda-prom-creds
  namespace: default
spec:
  secretTargetRef:
    - parameter: bearerToken
      name: keda-prom-secret
      key: bearerToken
      # might be required if you're using a custom CA
    - parameter: ca
      name: keda-prom-secret
      key: ca
```

KEDA – Event Sources and Scalers

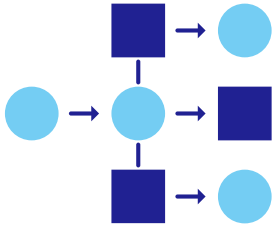
Currently available scalers for KEDA

- ActiveMQ Artemis
- Apache Kafka
- AWS CloudWatch
- AWS Kinesis Stream
- AWS SQS Queue
- Azure Blob Storage
- Azure Event Hubs
- Azure Log Analytics
- Azure Monitor
- Azure Pipelines
- Azure Service Bus
- Azure Storage Queue
- CPU
- Cron
- External
- External Push
- Google Cloud Platform Pub/Sub
- Huawei Cloudeye
- IBM MQ
- InfluxDB
- Kubernetes Workload
- Liiklus Topic
- Memory
- Metrics API
- MongoDB
- MSSQL
- MySQL
- NATS Streaming
- OpenStack Metric
- OpenStack Swift
- PostgreSQL
- Prometheus
- RabbitMQ Queue
- Redis Lists
- Redis Lists (supports Redis Cluster)
- Redis Streams
- Redis Streams (supports Redis Cluster)
- Selenium Grid Scaler
- Solace PubSub+ Event Broker

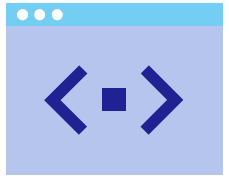
Dapr primer

State of enterprise developers

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

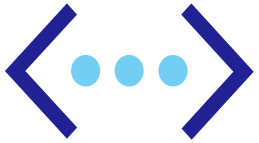


Developing resilient, scalable, microservice-based apps that interact with services



Focusing on building applications, not infrastructure

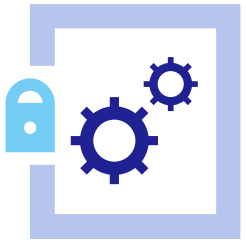
Trending toward serverless platforms with simple code to cloud pipelines



Using multiple languages and frameworks during development

What is holding back microservice development?

Limited tools and runtimes to build distributed applications

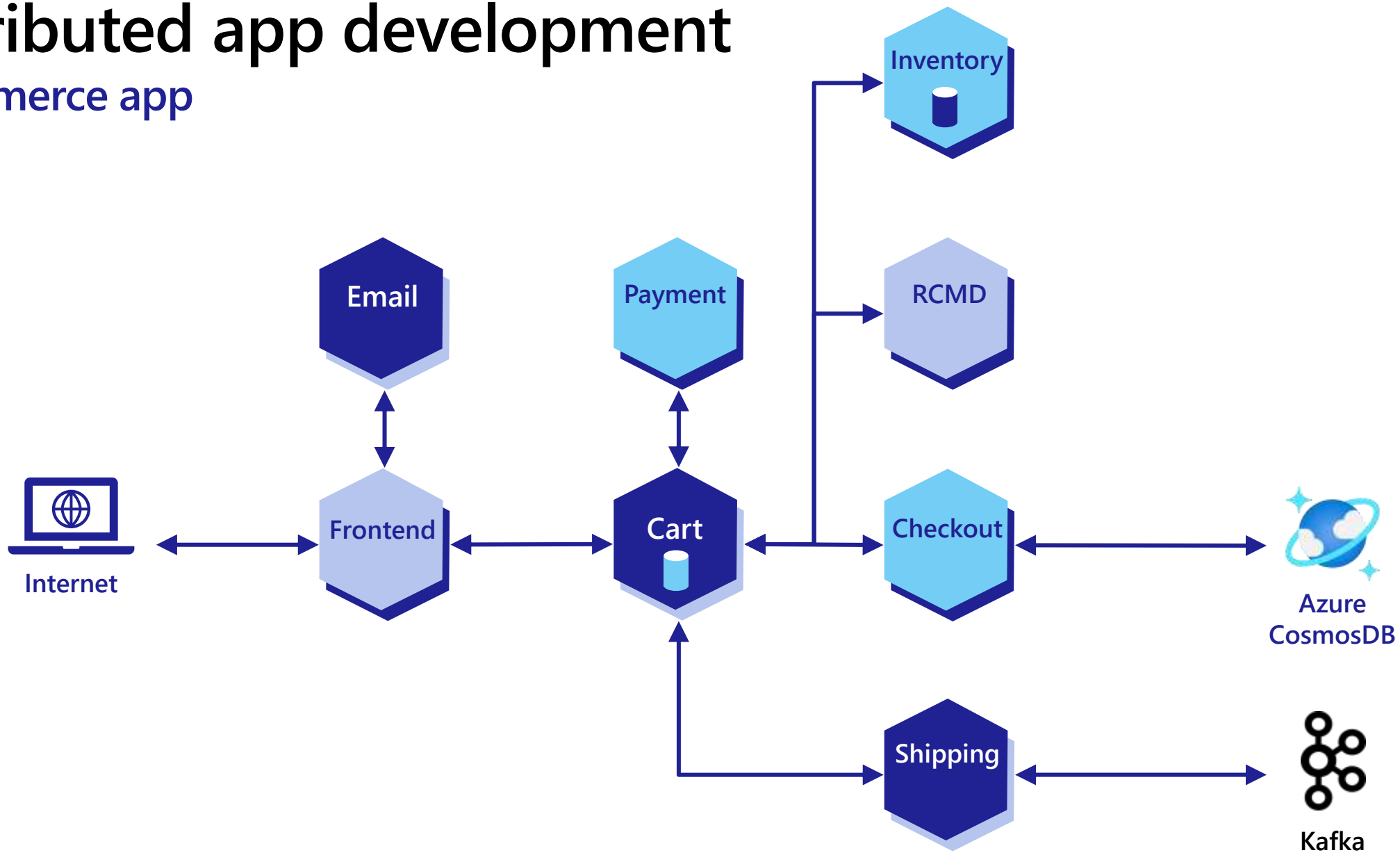


Runtimes have limited language support and tightly controlled feature sets

Runtimes only target specific infrastructure platforms with limited portability

Distributed app development

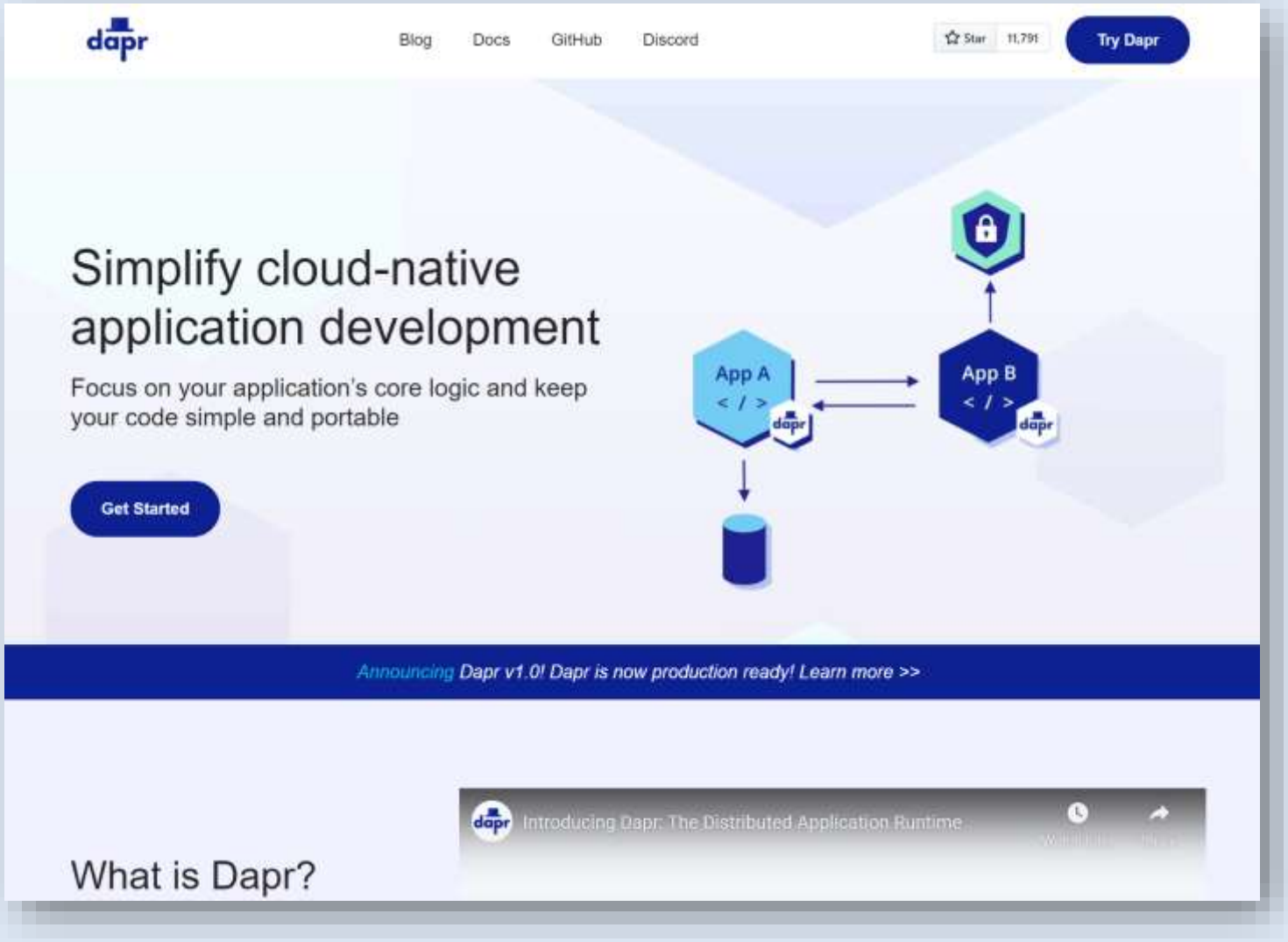
E-commerce app



Distributed Application Runtime

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

dapr.io



Dapr Goals

Best-practices building blocks

Any language or framework

Consistent, portable, open APIs

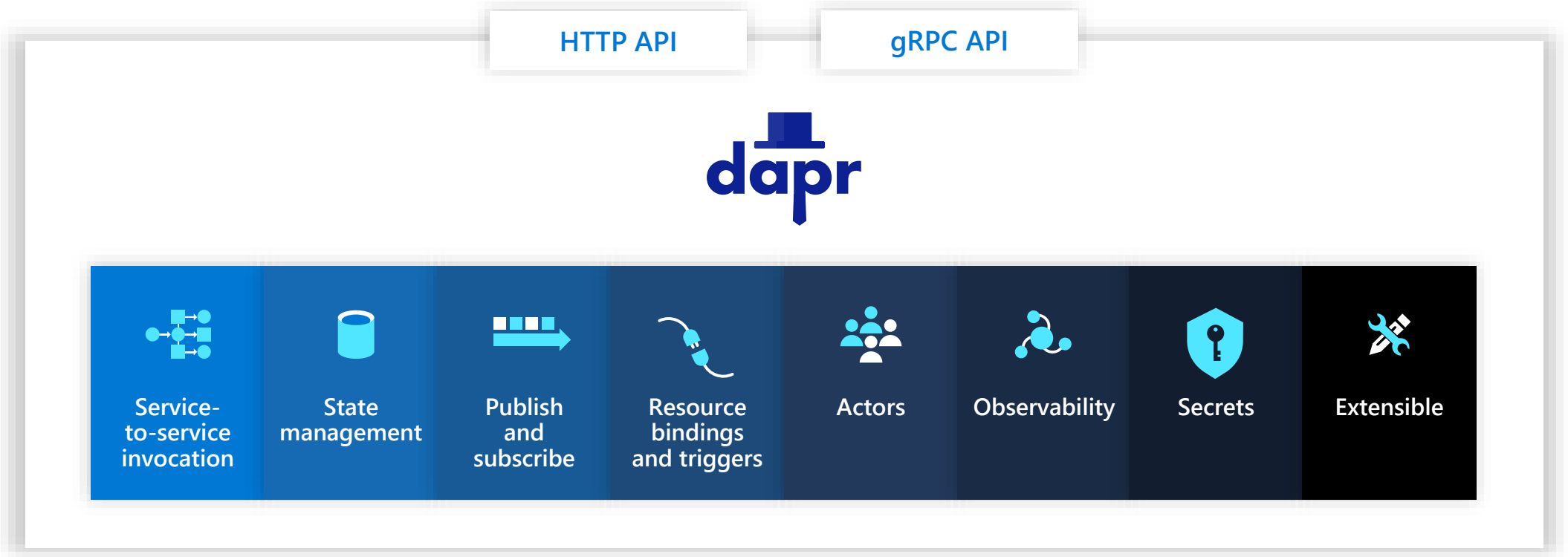
Adopt standards

Extensible and pluggable components

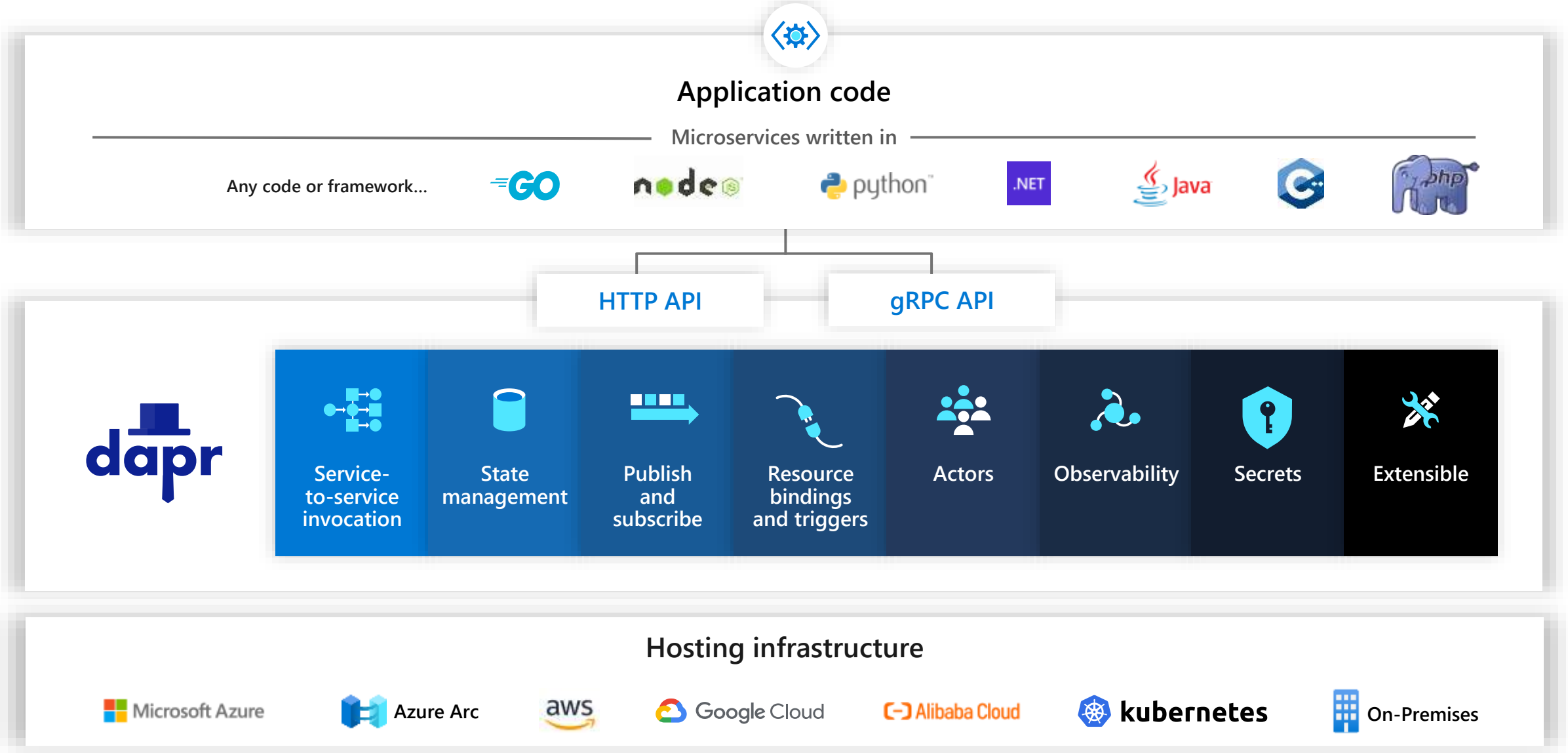
Platform agnostic cloud + edge

Community driven, vendor neutral

Microservice building blocks



Any cloud or edge infrastructure



Dapr components

Swappable YAML files with resource connection details

Over 70 components available

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