Developing Microservices with Dapr

NDC Oslo 2021

@jakobehn

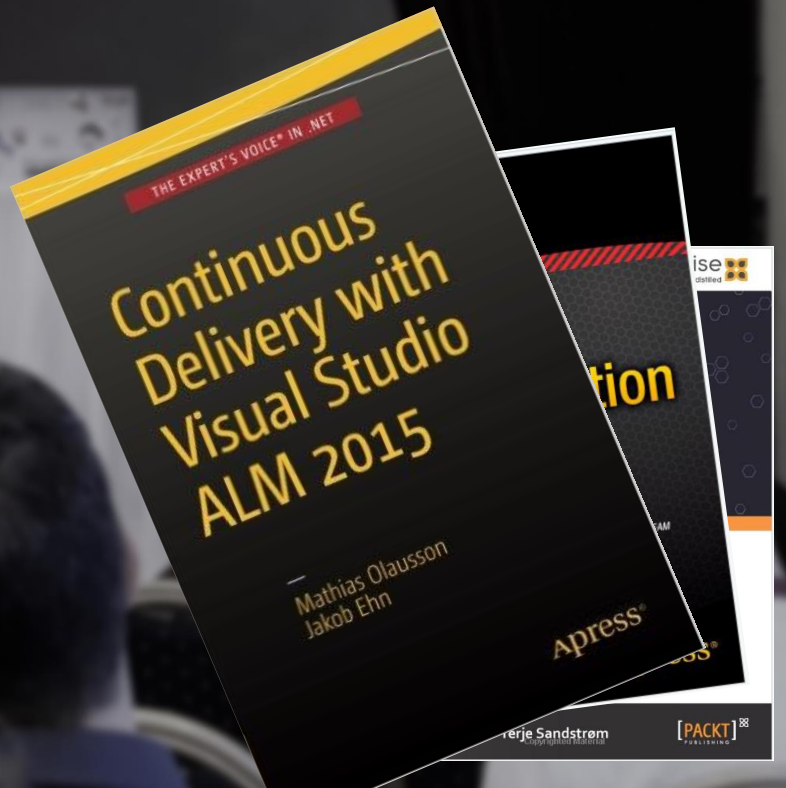
https://blog.ehn.nu

active SOLUTION

Jakob Ehn
Microsoft Azure MVP

https://blog.ehn.nu

@jakobehn

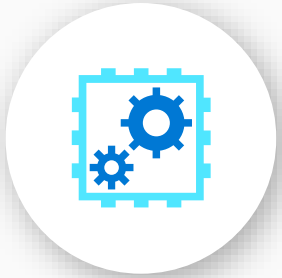


MVP Microsoft® Most Valuable Professional

What’s hard about microservices?

Need to support lots of different integration points

(cache, message queues, 3rd party APIs, secret stores)



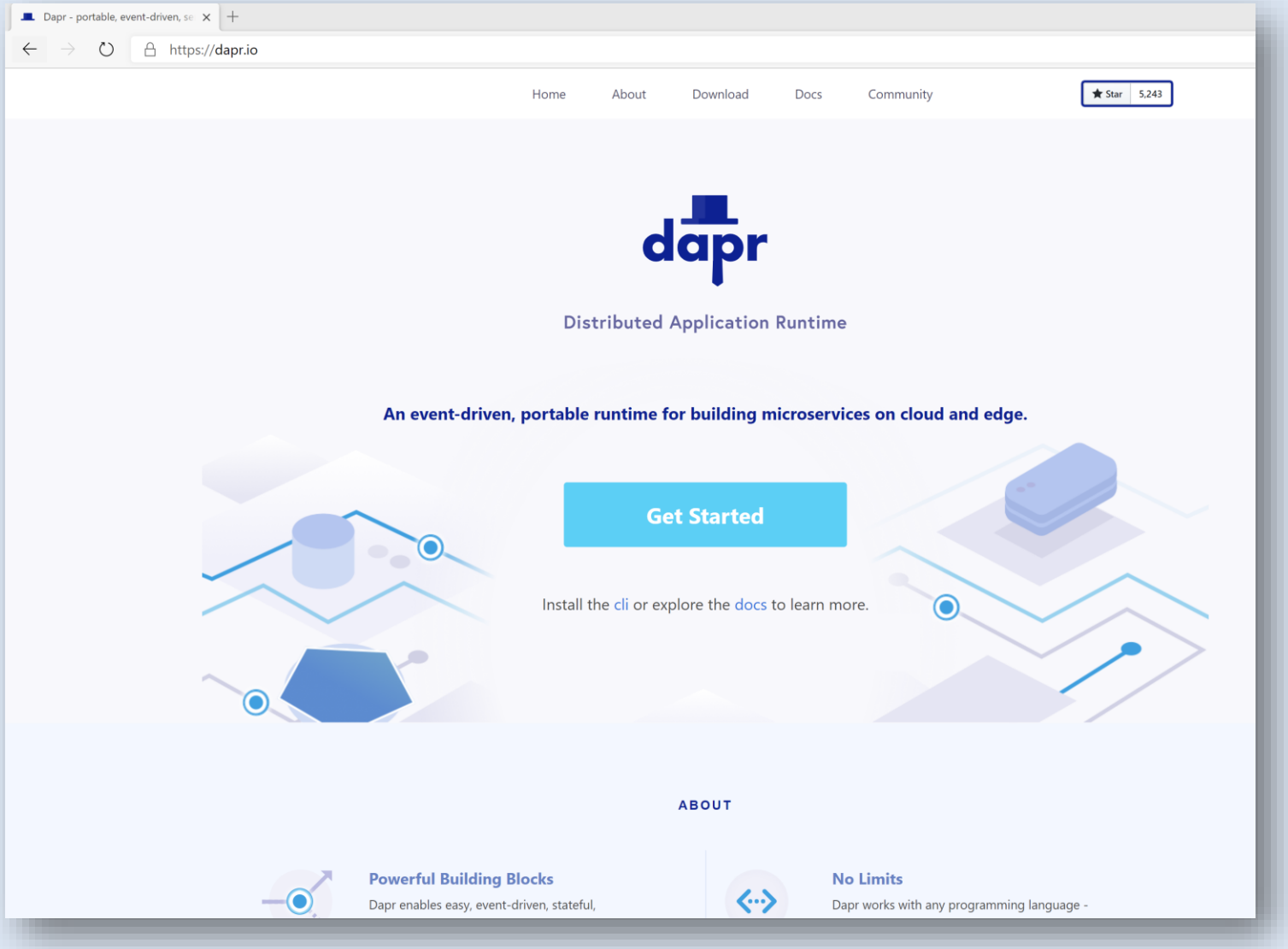
Lots of different targets to support **tracing**, **configuration**, and **secret management**

Need to handle things like **service discovery**, **transient failures** and **distributed tracing**

Distributed Application Runtime

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

https://dapr.io

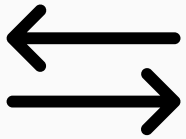


Dapr Goals

Best-Practices
Building Blocks

Any Language
or Framework

Community Driven
Vendor Neutral

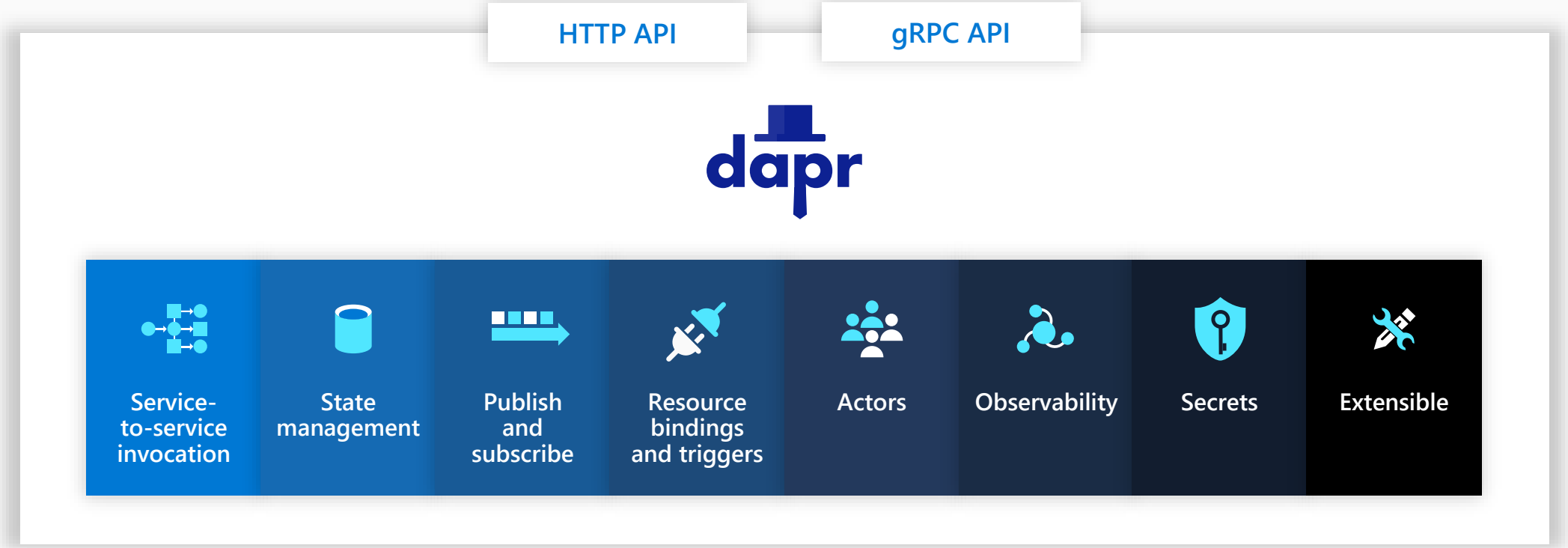


Consistent, Portable,
Open APIs

Platform Agnostic
Cloud + Edge

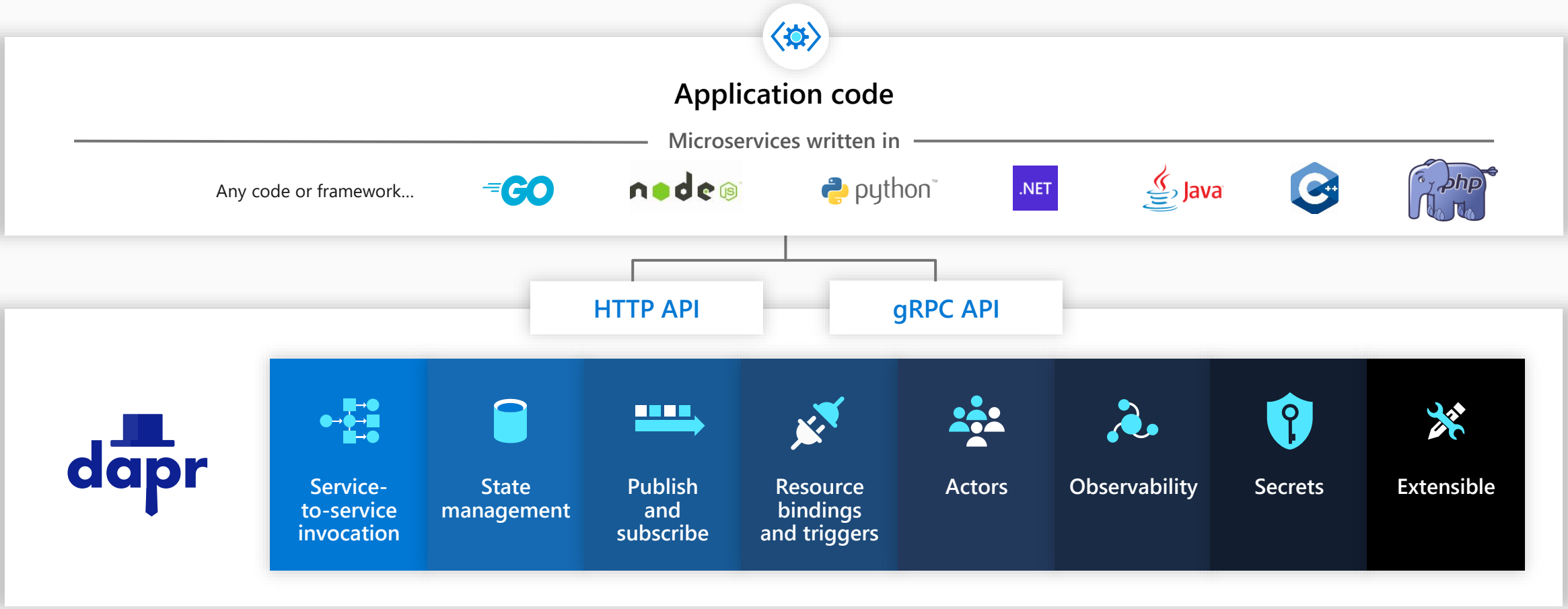
Extensible and
Pluggable Components

Microservice building blocks

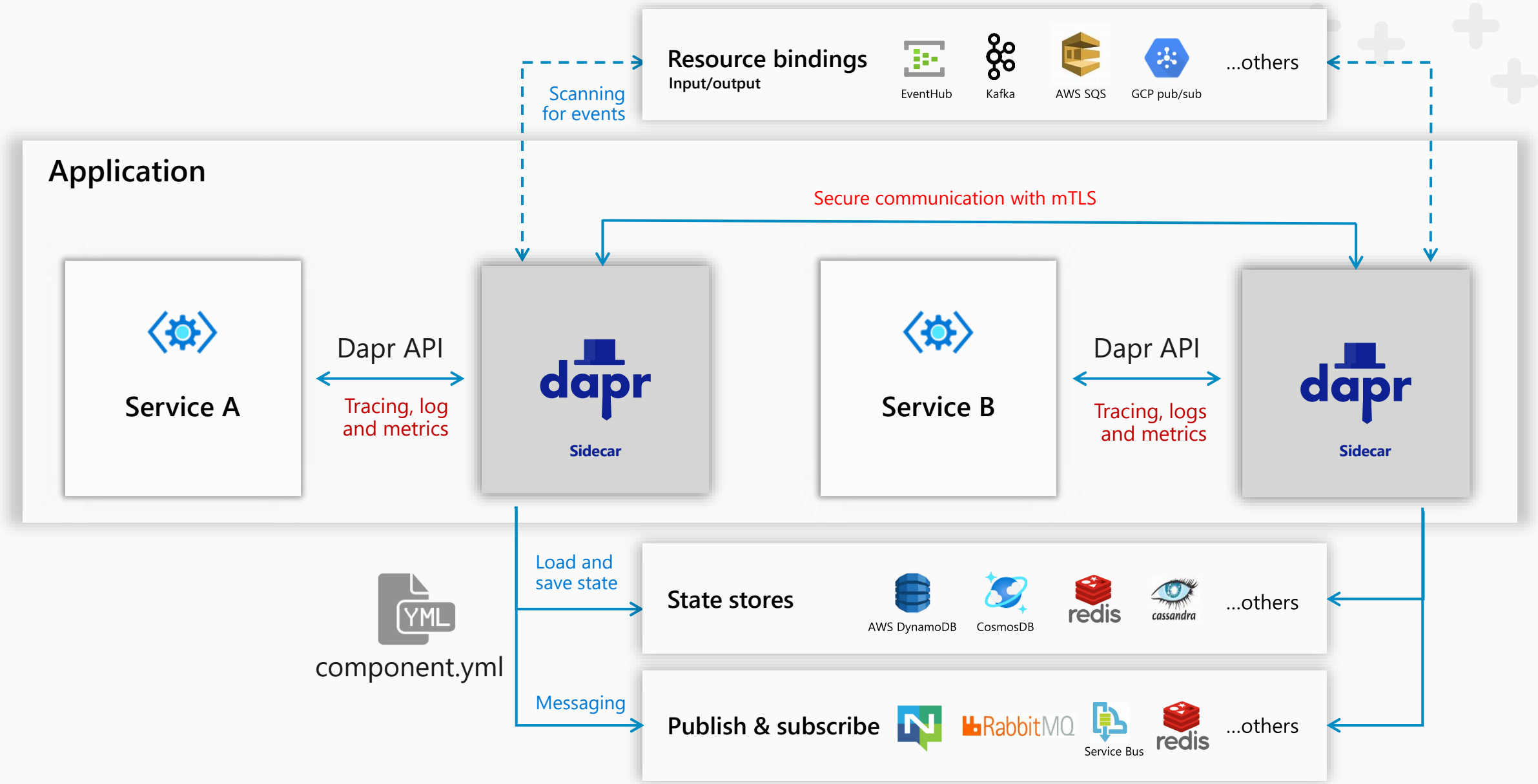


Microservice building blocks

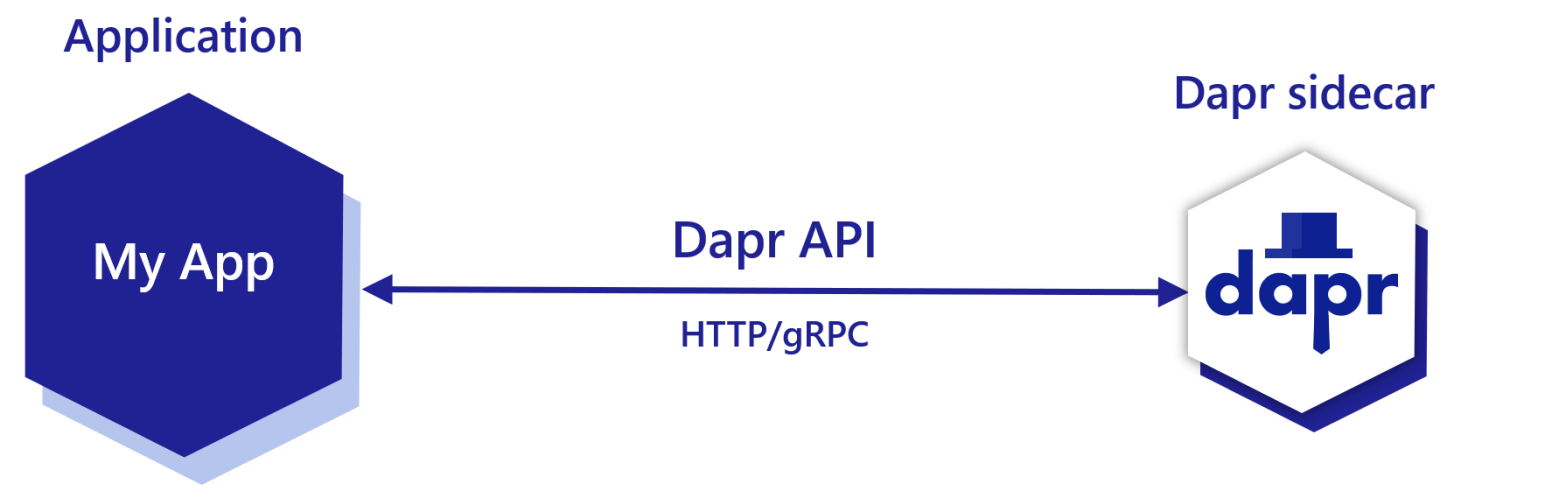
- Standard APIs accessed over HTTP/gRPC protocols from user service code
  http://localhost:3500/v1.0/invoke/cart/method/neworder
  http://localhost:3500/v1.0/state/inventory/item67
- Runs as local “side car library” dynamically loaded at runtime for each service



Sidecar and component architecture



Dapr Sidecar Model



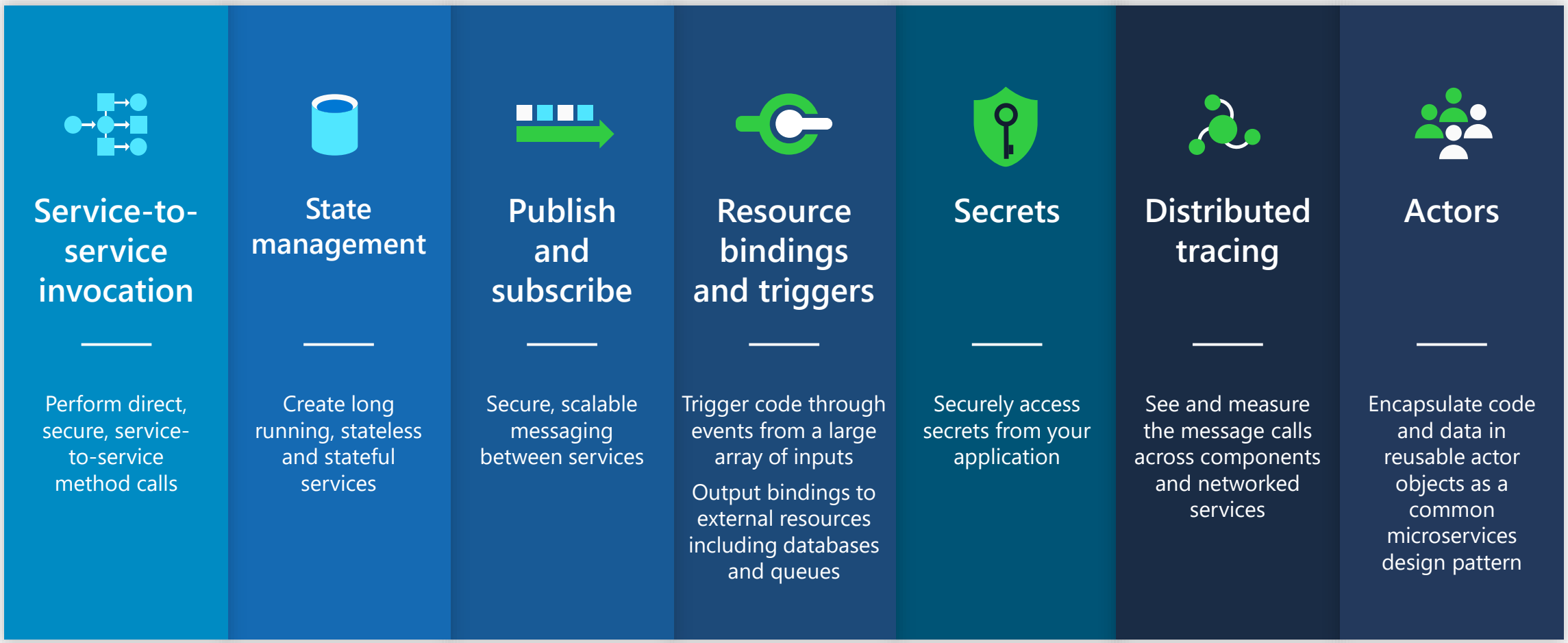

```
POST http://localhost:3500/v1.0/invoke/cart/method/neworder

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 Components

component.yaml

```yaml
apiVersion: dapr.io/v1alpha1
kind: Component
metadata:
  name: messagebus
  namespace: default
spec:
  type: pubsub.azure.servicebus
  version: v1
  metadata:
  - name: connectionString
    value: "Endpoint=sb://daprbus-demo.servicebus.windows……."
```

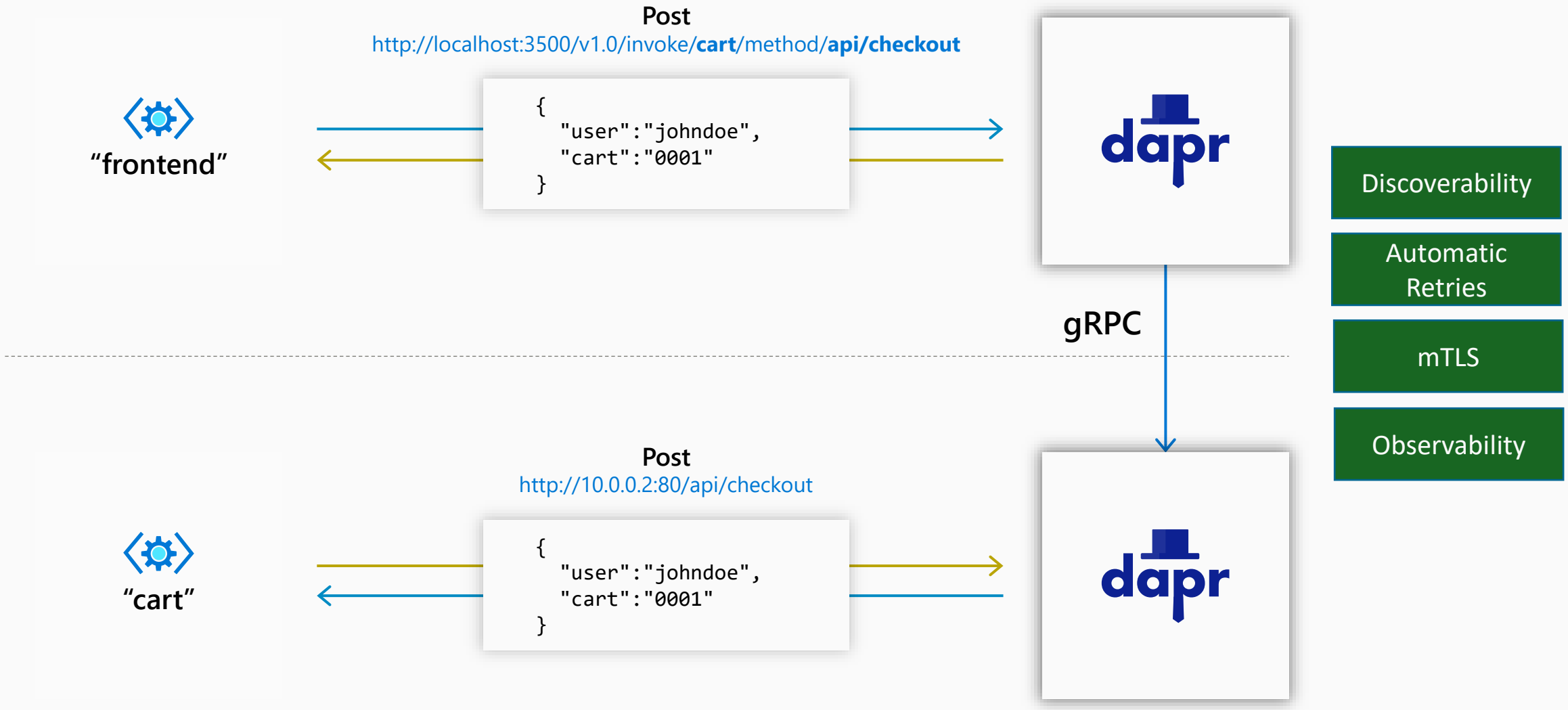
Microservice building blocks



Use Dapr components

Dapr building blocks

Service invocation



Dapr building blocks

State management: key/value

Optimistic concurrency

Automatic Retries

State transactions

App "myApp"

Get

http://localhost:3500/v1.0/state/<**store-name**>**/planet**

```
{
   "name": "Tatooine"
}
```

Post

http://localhost:3500/v1.0/state**/<store-name>**

```
[{
    "key": "weapon",
    "value": "DeathStar"
}, {
    "key": "planet",
    "value": {
        "name": "Tatooine"
    }
}]
```

dapr

key	value
myApp-weapon	"DeathStar"
myApp-planet	{ "name": "Tatooine" }

State store of your choice

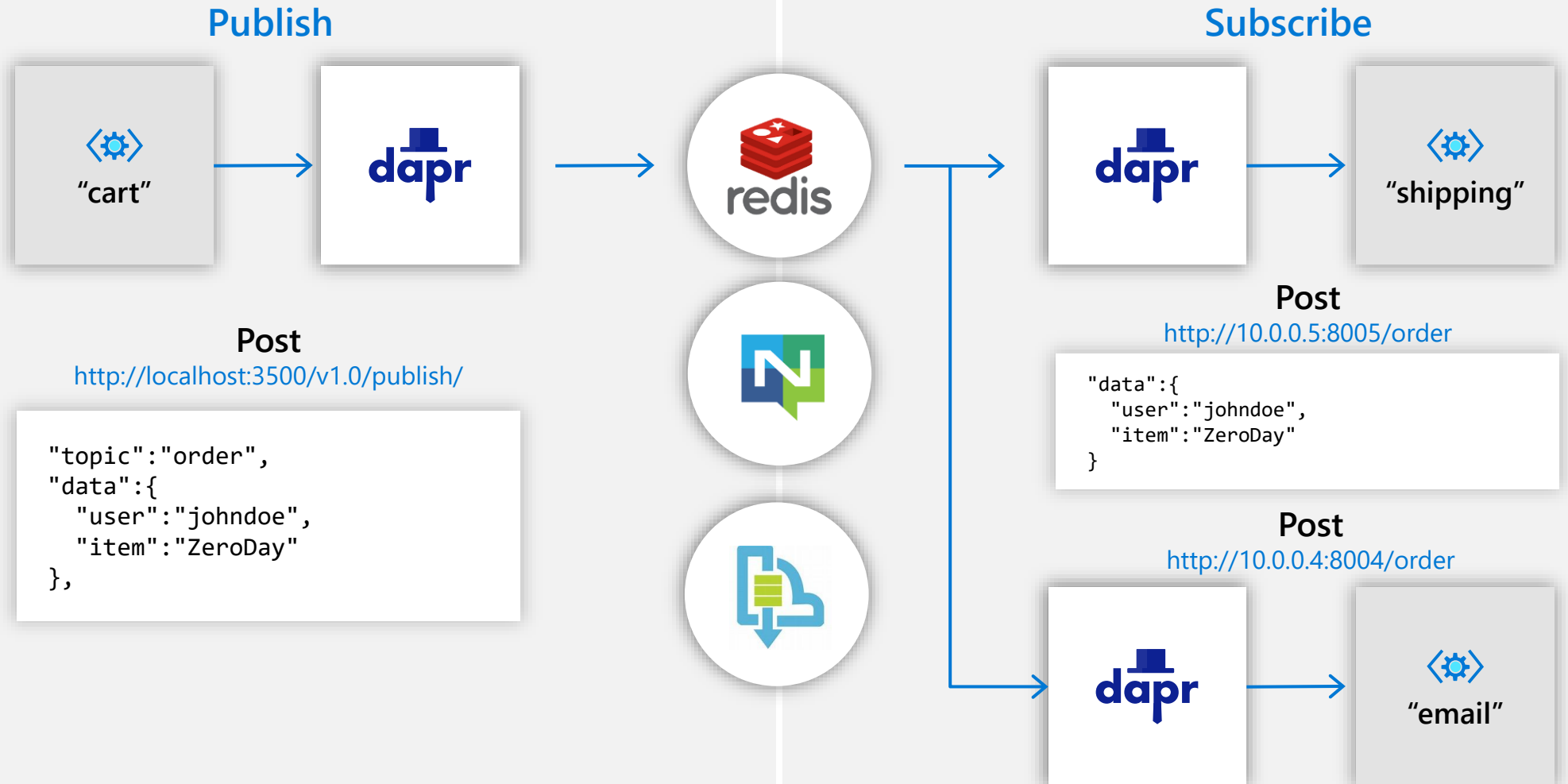
AWS DynamoDB, CosmosDB, redis, cassandra, ...others

Dapr building blocks

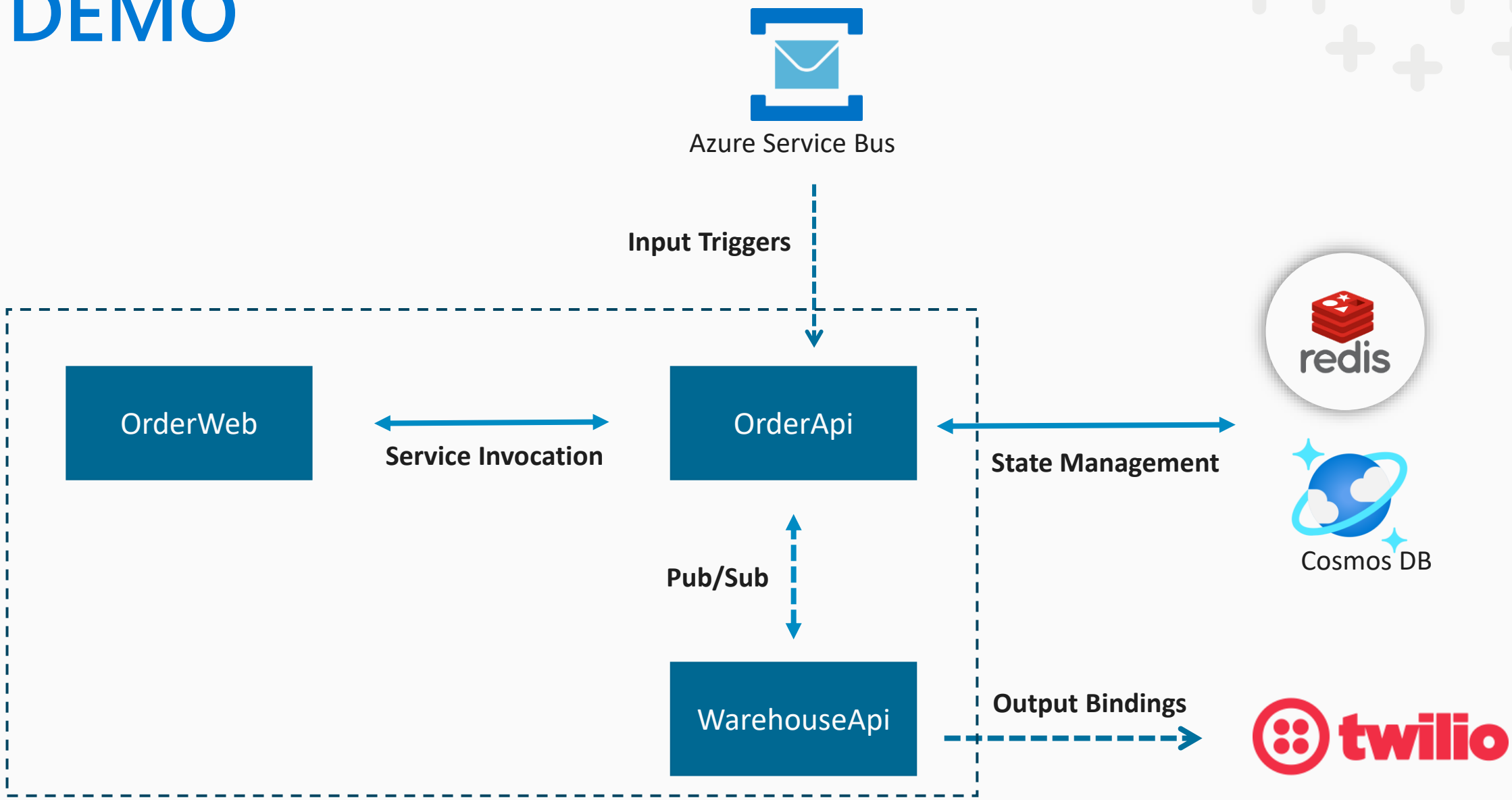
Publish and subscribe

At-least-once guarantee

Message Time-to-Live



DEMO


Azure Service Bus
Input Triggers
OrderWeb
Service Invocation
OrderApi
State Management
redis
Cosmos DB
Pub/Sub
WarehouseApi
Output Bindings
twilio

Demo

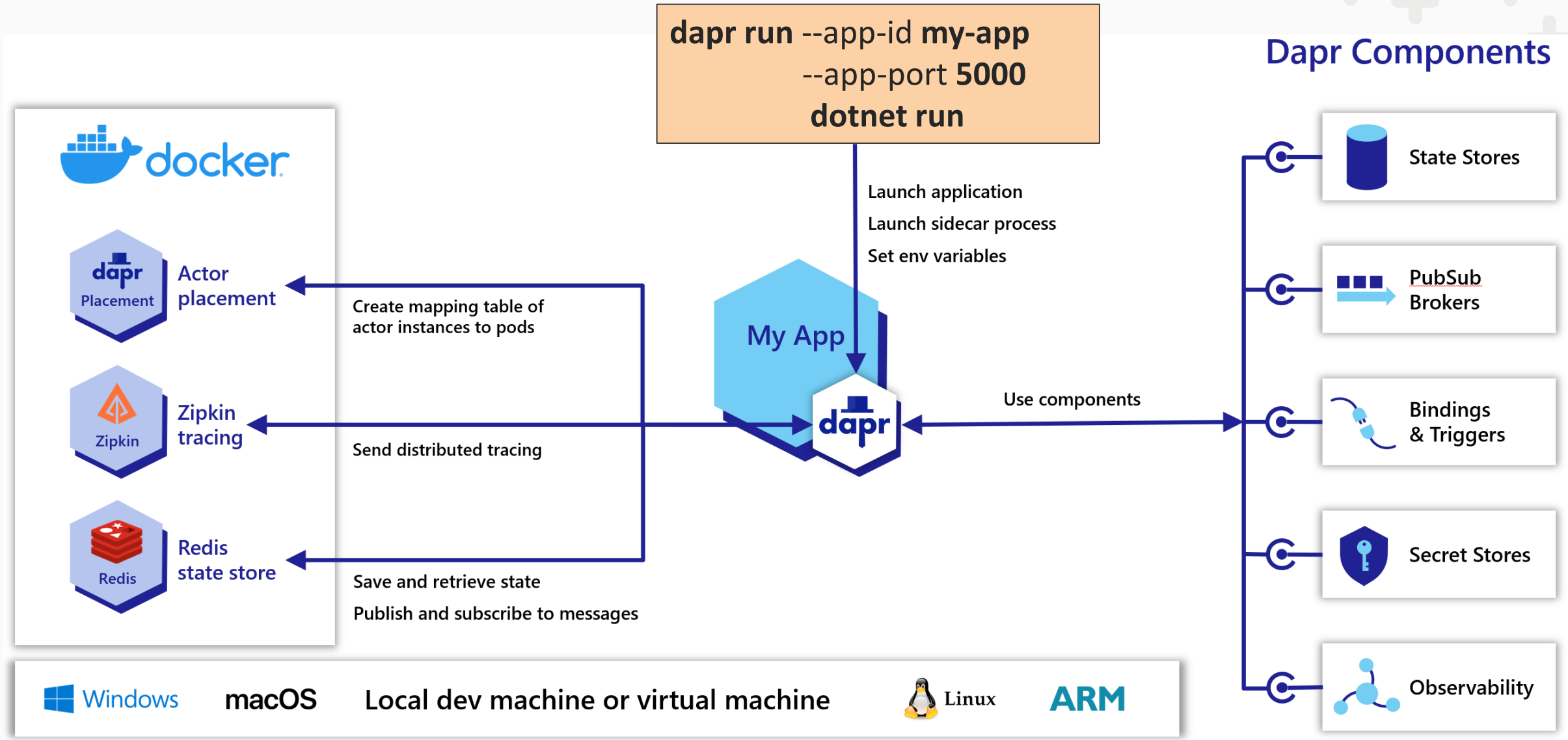
Dapr Hosting Options

Self-hosted

Kubernetes

Azure Container Apps (preview)

Running Dapr Self-hosted



Running Dapr on Kubernetes

Initialize Dapr on k8s cluster

```
dapr init -k
```

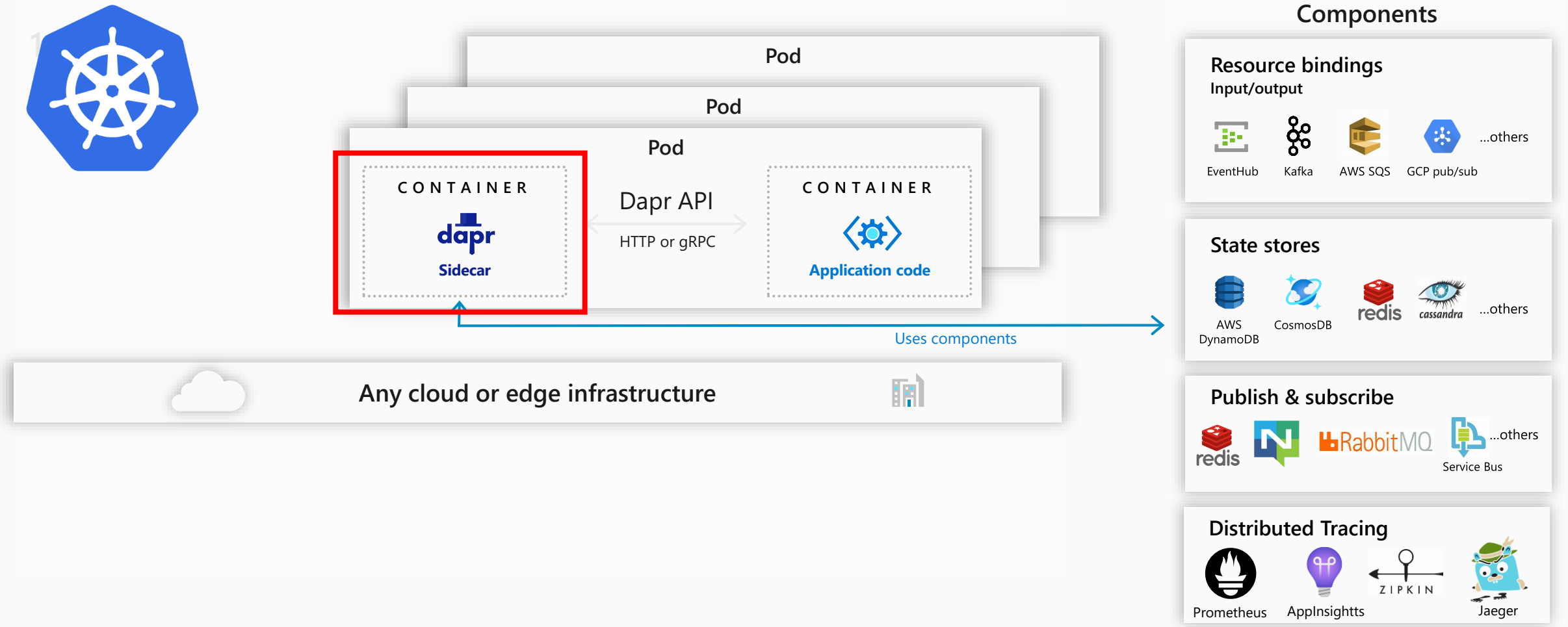
Deploy components

```
Kubectl apply –f components.yaml
```

Annotate k8s deployment

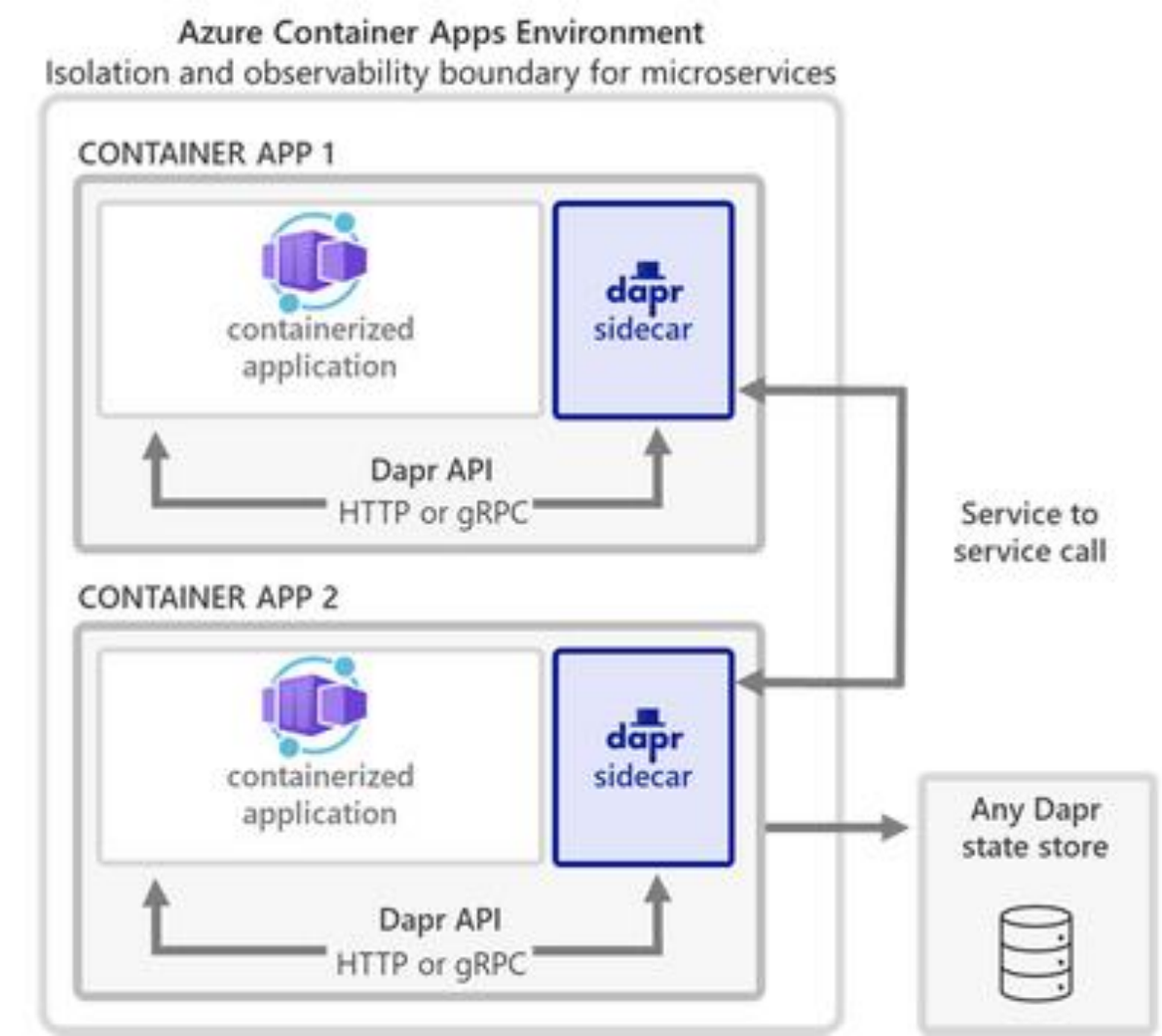
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
      annotations:
        dapr.io/enabled: "true"
        dapr.io/app-id: "my-app"
        dapr.io/app-protocol: "http"
        dapr.io/app-port: "8080"
…
```

Dapr Kubernetes hosted



Running Dapr on Azure Container Apps (preview)

```
az containerapp create
  --name myapp
  --resource-group group
  --environment myapp-env
  --image myImage:1.2
  --target-port 5000
  --ingress 'external'
  --min-replicas 1
  --max-replicas 1
  --enable-dapr
  --dapr-app-port 5000
  --dapr-app-id myapp
  --dapr-components
./components.yaml
```



Demo

- Hosting Dapr

SDK Languages

Language	Status	Client SDK	Server extensions	Actor SDK
.NET	Stable	✔	ASP.NET Core	✔
Python	Stable	✔	gRPC	FastAPI Flask
Java	Stable	✔	Spring Boot	✔
Go	Stable	✔	✔	
PHP	Stable	✔	✔	✔
C++	In development	✔		
Rust	In development	✔		
Javascript	In development	✔		

Client SDK:
The Dapr client allows you to invoke Dapr building block APIs and perform actions such as:

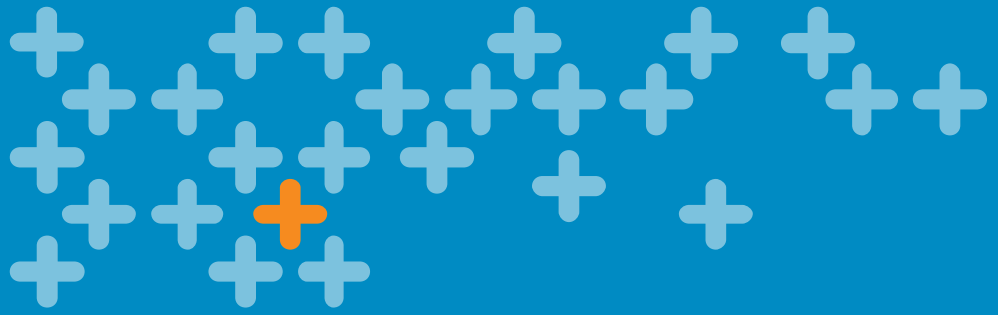
- Invoke methods on other services
- Store and get state
- Publish and subscribe to message topics
- Interact with external resources through input and output bindings
- Get secrets from secret stores

Server extensions:
The Dapr service extensions allow you to create services that can:

- Be invoked by other services
- Subscribe to topics

Actor SDK:
The Dapr Actor SDK allows you to build virtual actors with:

- Methods that can be invoked by other services
- State that can be stored and retrieved
- Timers with callbacks



Thank you!

Jakob Ehn

@jakobehn

https://blog.ehn.nu