BASTA!

Dapr for .NET Developers

dapr

Ricardo Niepel
Cloud Solution Architect - Azure App Dev
ricardo.niepel@microsoft.com

@RicardoNiepel RicardoNiepel

Microsoft

What is holding microservice development back?

Hard to incrementally migrate from existing code to a microservices architecture

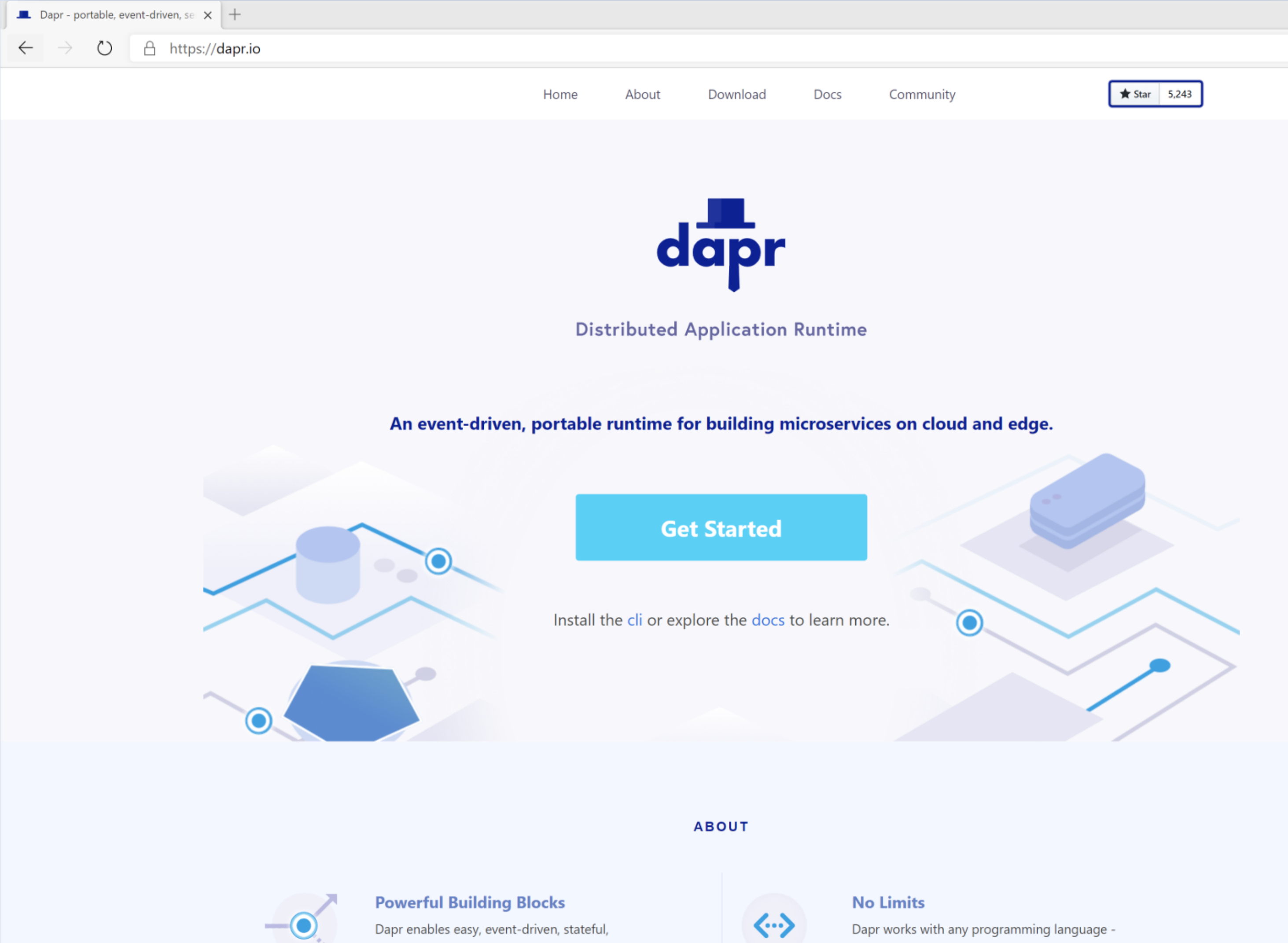
Programming model runtimes have narrow language support and tightly controlled feature sets

Runtimes only target specific infrastructure platforms with limited code portability across clouds and edge

Distributed Application Runtime

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

https://dapr.io



Dapr - portable, event-driven, se

https://dapr.io

dapr

Distributed Application Runtime

An event-driven, portable runtime for building microservices on cloud and edge.

Install the cli or explore the docs to learn more.

ABOUT

Powerful Building Blocks

Dapr enables easy, event-driven, stateful,

No Limits

Dapr works with any programming language -

Dapr Goals

Best-Practices
Building Blocks

Any Language
or Framework

Community Driven
Vendor Neutral

Adopt Standards

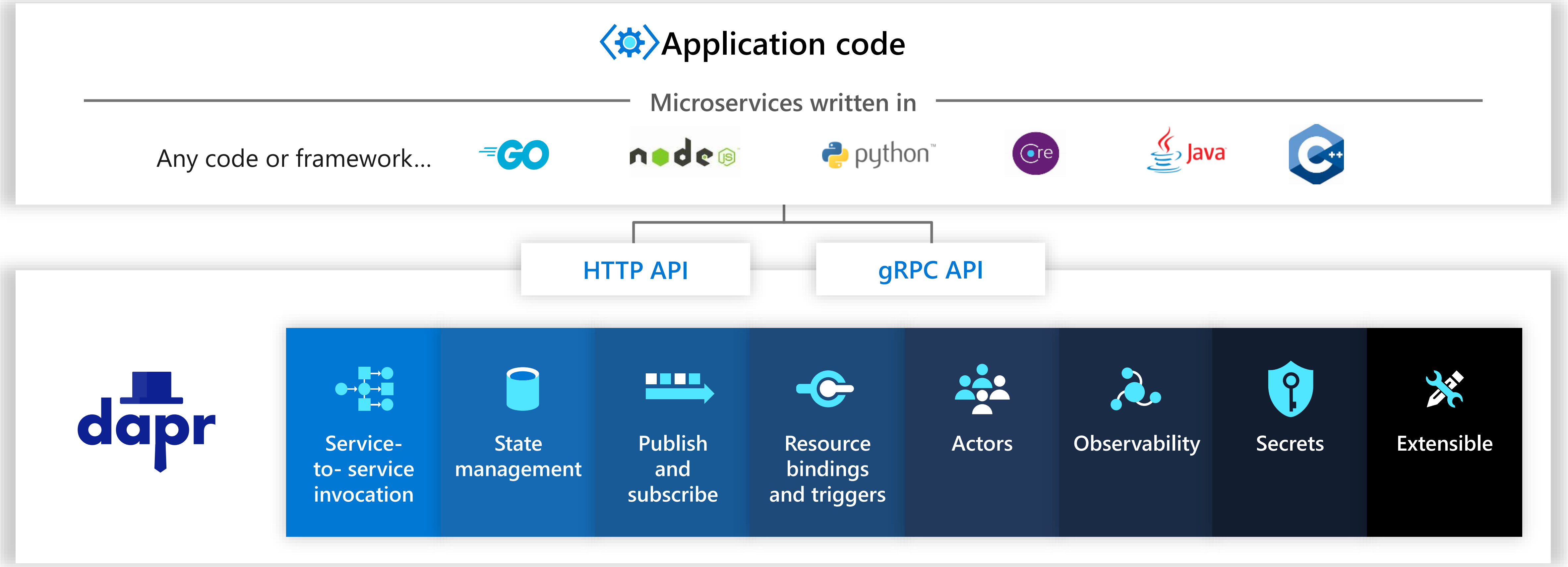
Consistent, Portable,
Open APIs

Platform Agnostic
Cloud + Edge

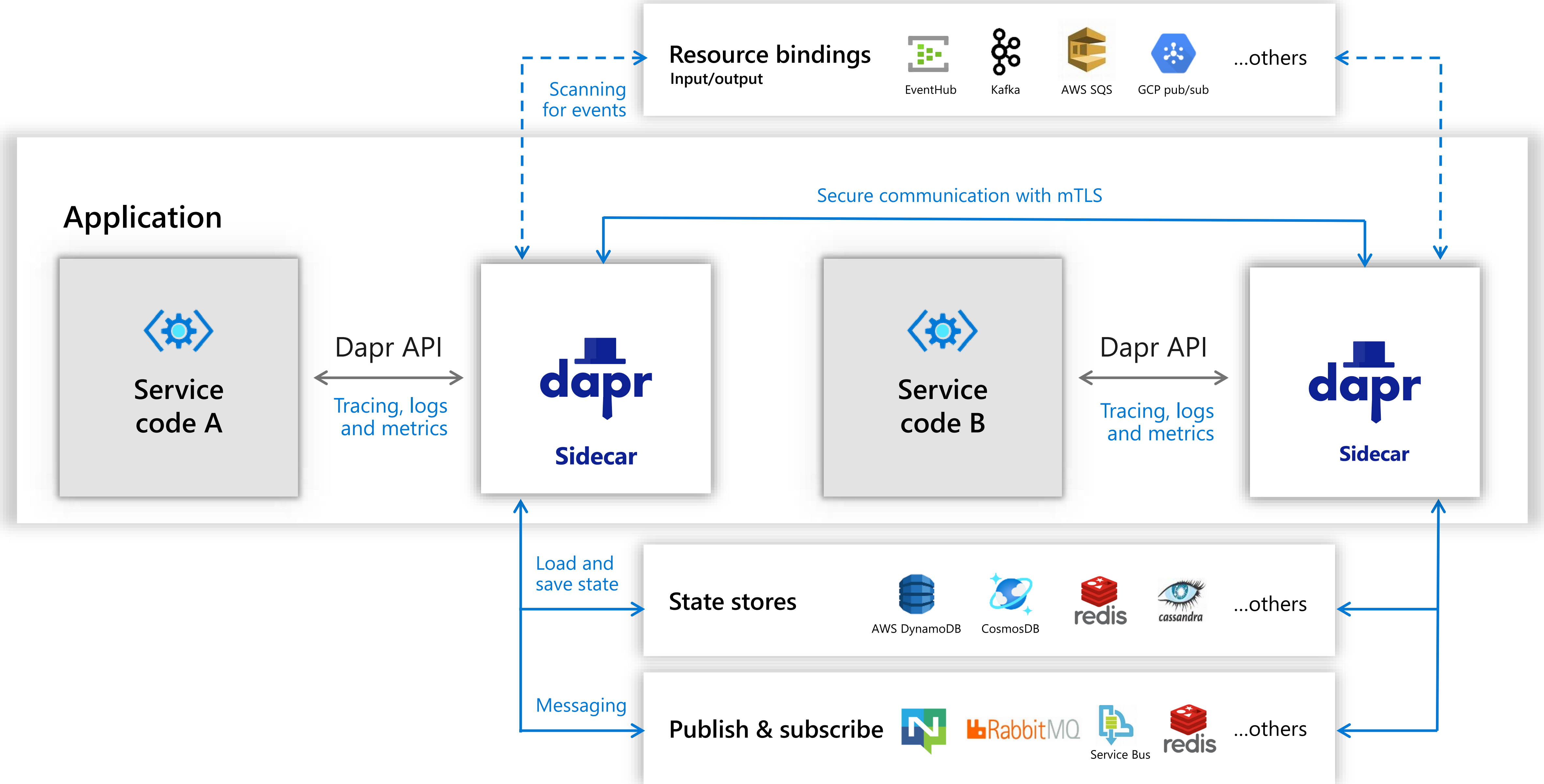
Extensible and
Pluggable Components

Concept of Dapr

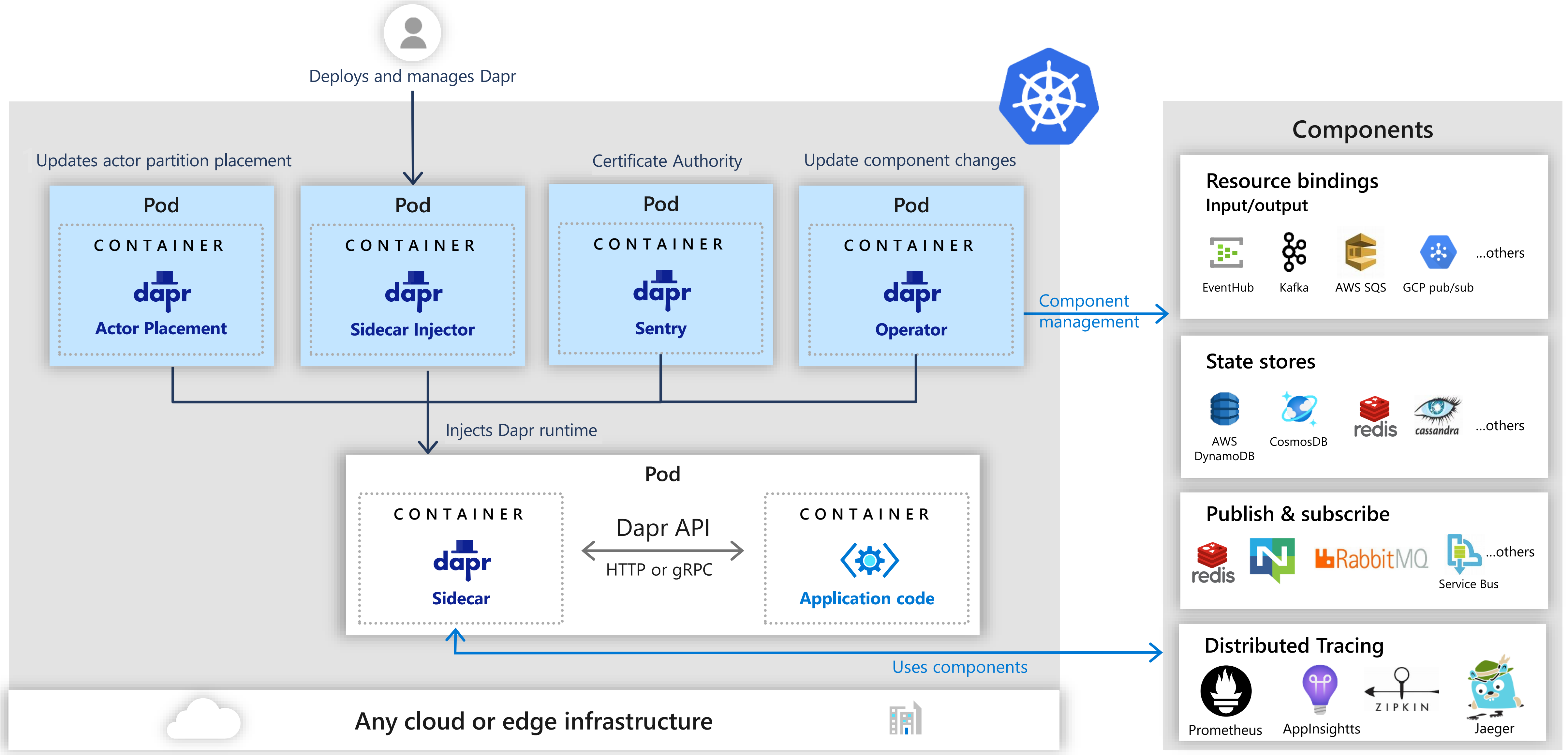
- ✓ Standard APIs accessed over http/gRPC protocols from user service code
  - http://localhost:3500/v1.0/state/inventory/orderkey
  - http://localhost:3500/v1.0/invoke/myapp/method/neworder
- ✓ Runs as local “side car library” dynamically loaded at runtime for each service



Sidecar and component architecture



Dapr Kubernetes hosted



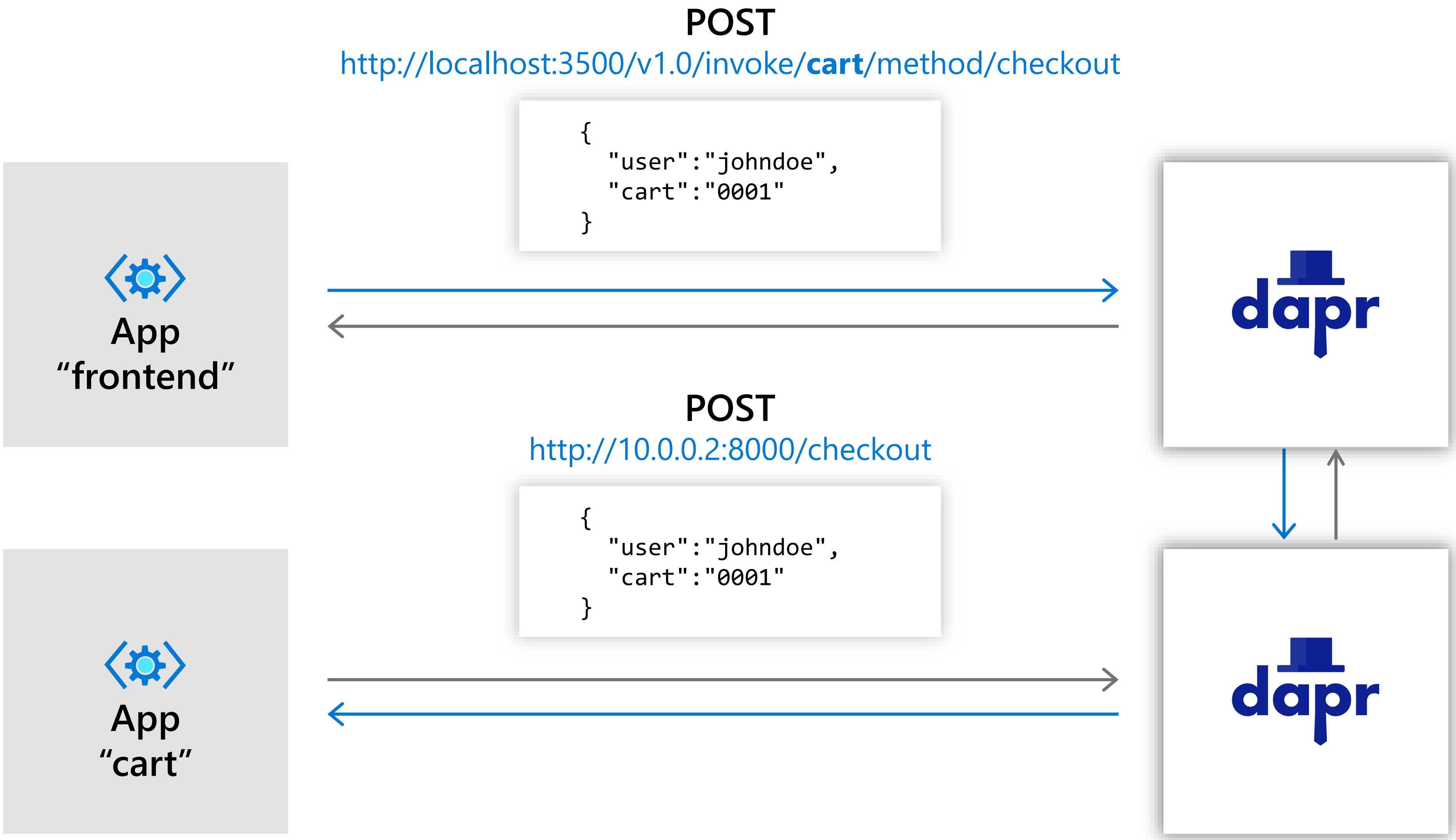
Microservice Building Blocks

Service-to-service invocation

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

Microservice Building Block

Service Invocation



POST

http://localhost:3500/v1.0/invoke/**cart**/method/checkout

```
{
  "user":"johndoe",
  "cart":"0001"
}
```

App
“frontend”

POST

http://10.0.0.2:8000/checkout

```
{
  "user":"johndoe",
  "cart":"0001"
}
```

App
“cart”

Microservice Building Blocks

Service-to-service invocation

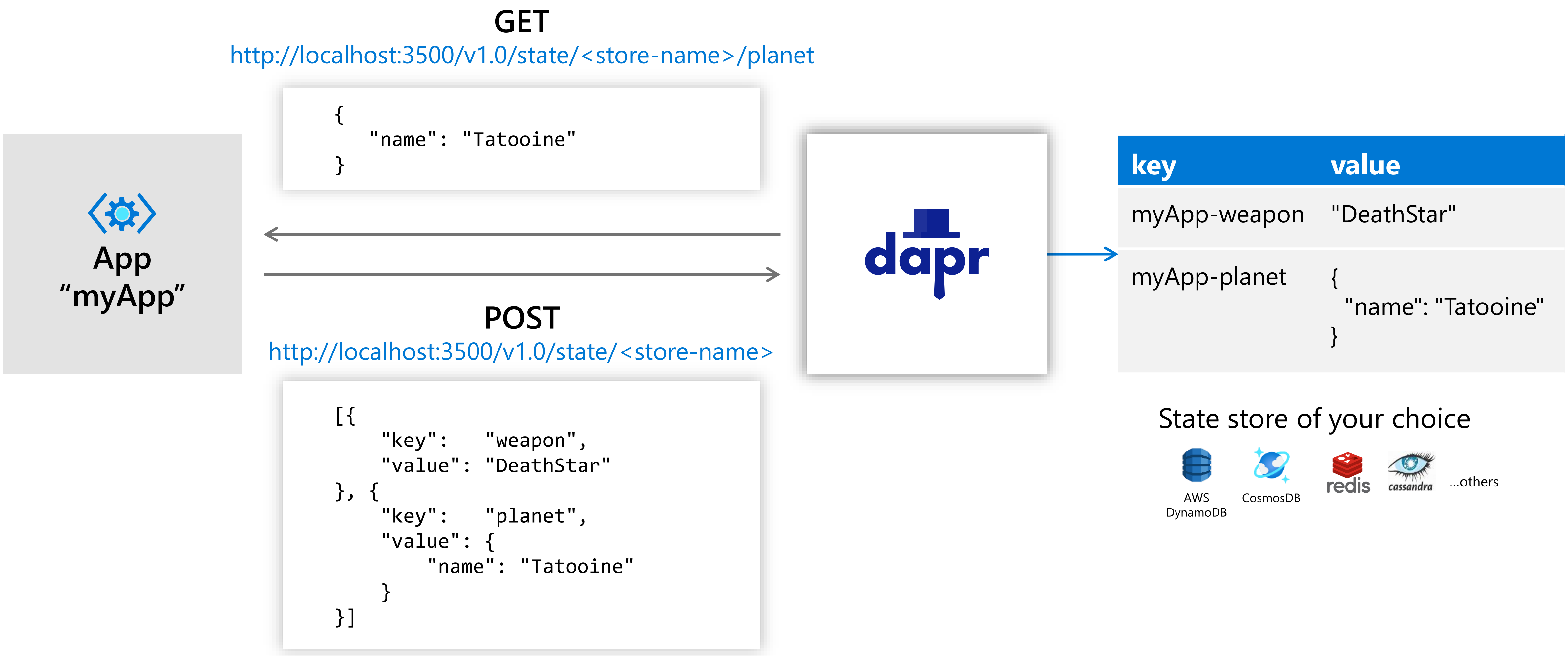
Perform direct, secure, service-to-service method calls

State management

Create long running, stateless and stateful services

Microservice Building Block

State Management: key/value



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"
    }
}]
```

key	value
myApp-weapon	"DeathStar"

myApp-planet	{
	"name": "Tatooine"
	}

State store of your choice

Microservice Building Blocks

Service-to-service invocation

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

State management

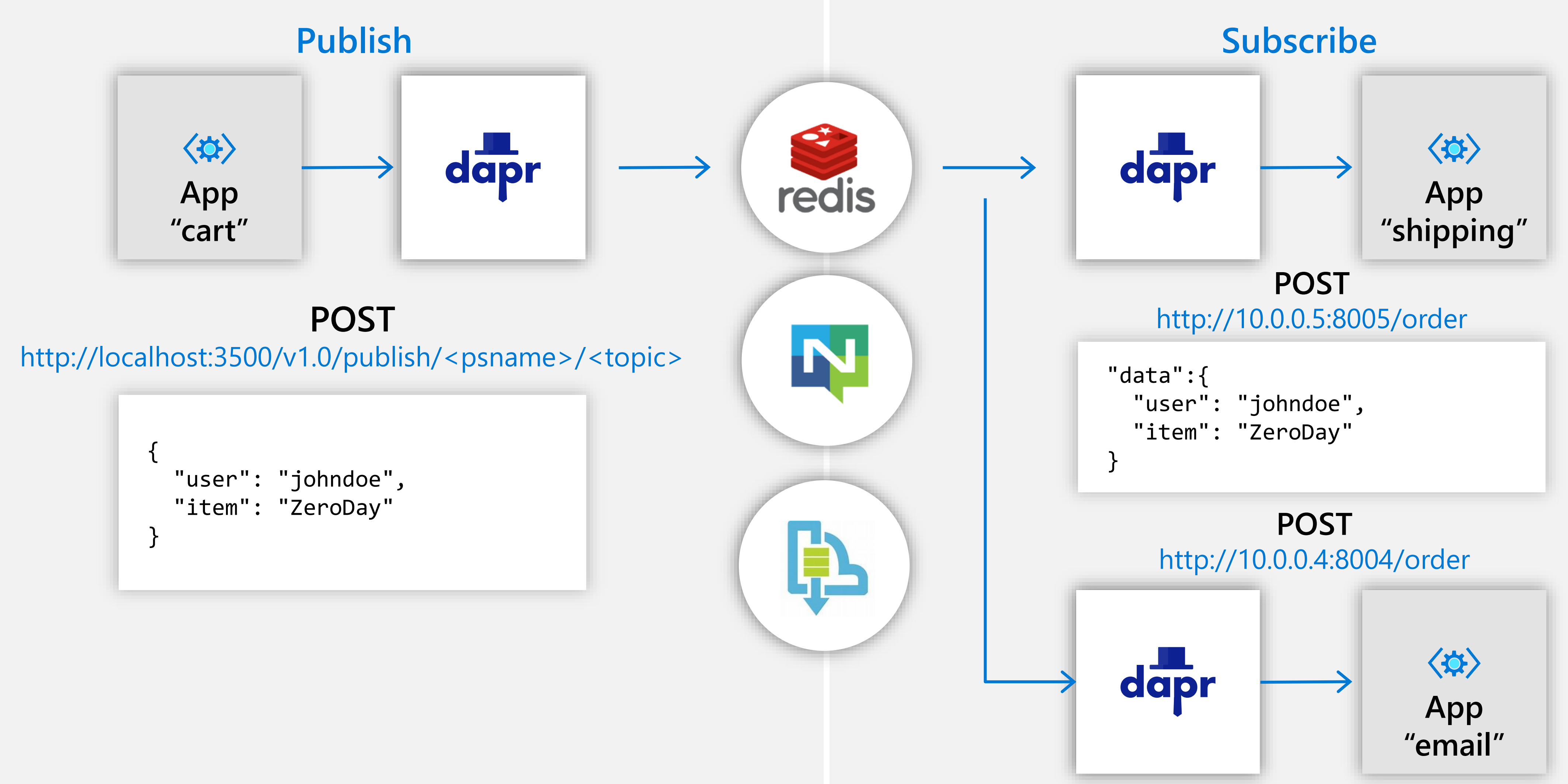
Create long running, stateless and stateful services

Publish and subscribe

Secure, scalable messaging between services

Microservice Building Block

Publish and Subscribe



POST
http://localhost:3500/v1.0/publish/<psname>/<topic>

```
{
  "user": "johndoe",
  "item": "ZeroDay"
}
```

POST
http://10.0.0.5:8005/order

```
"data":{
  "user": "johndoe",
  "item": "ZeroDay"
}
```

POST
http://10.0.0.4:8004/order

Dapr .NET SDK, Service Invocation, State Management & PubSub

Demo

Microservice Building Blocks

Service-to-service invocation

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

State management

Create long running, stateless and stateful services

Publish and subscribe

Secure, scalable messaging between services

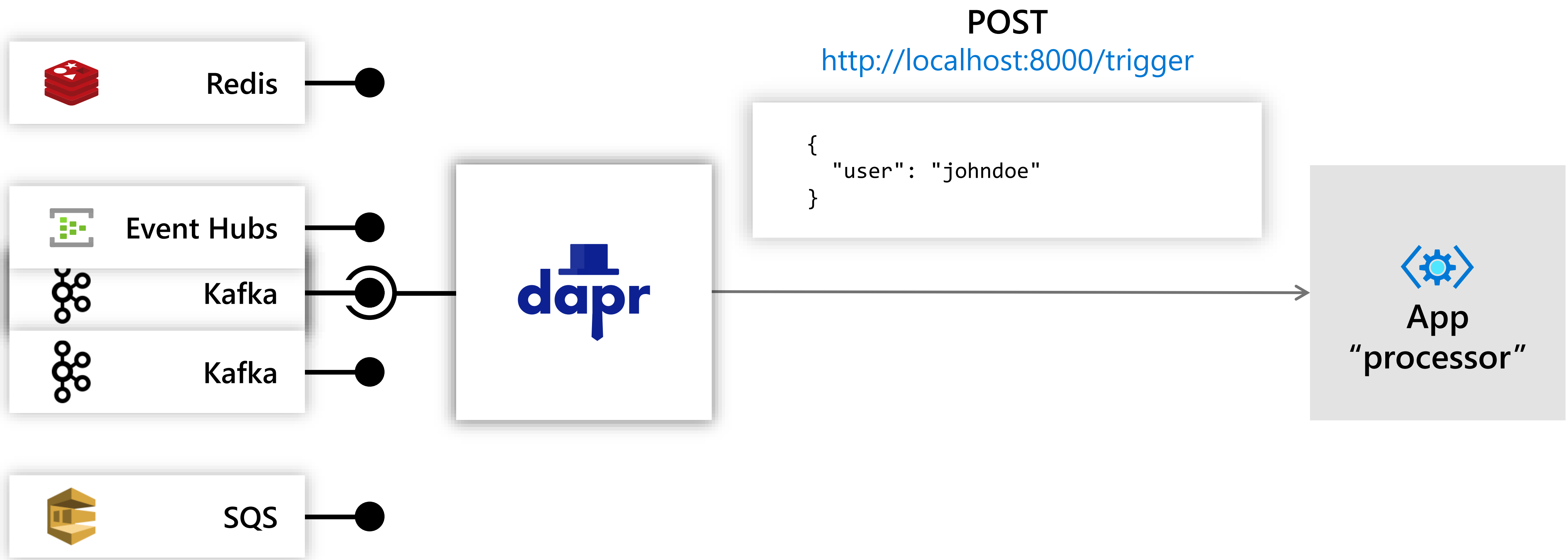
Resource bindings and triggers

Trigger code through events from a large array of inputs

Output bindings to external resources including databases and queues

Microservice Building Block

Resource Bindings: Input Trigger



Redis

Event Hubs

Kafka

Kafka

SQS

dapr

POST

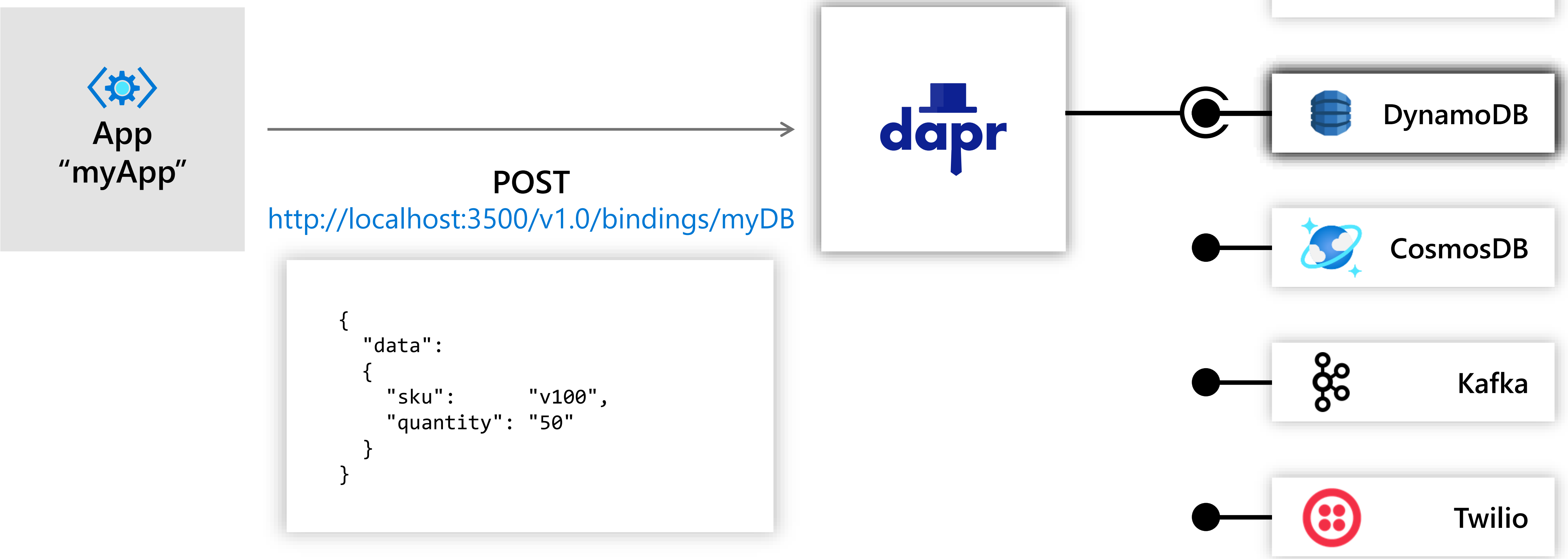
http://localhost:8000/trigger

```
{
  "user": "johndoe"
}
```

App
“processor”

Microservice Bausteine

Resource Bindings: Output



App
“myApp”

POST
http://localhost:3500/v1.0/bindings/myDB

```
{
  "data":
  {
    "sku":      "v100",
    "quantity": "50"
  }
}
```

Microservice Building Blocks

Service-to-service invocation

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

State management

Create long running, stateless and stateful services

Publish and subscribe

Secure, scalable messaging between services

Resource bindings and triggers

Trigger code through events from a large array of inputs

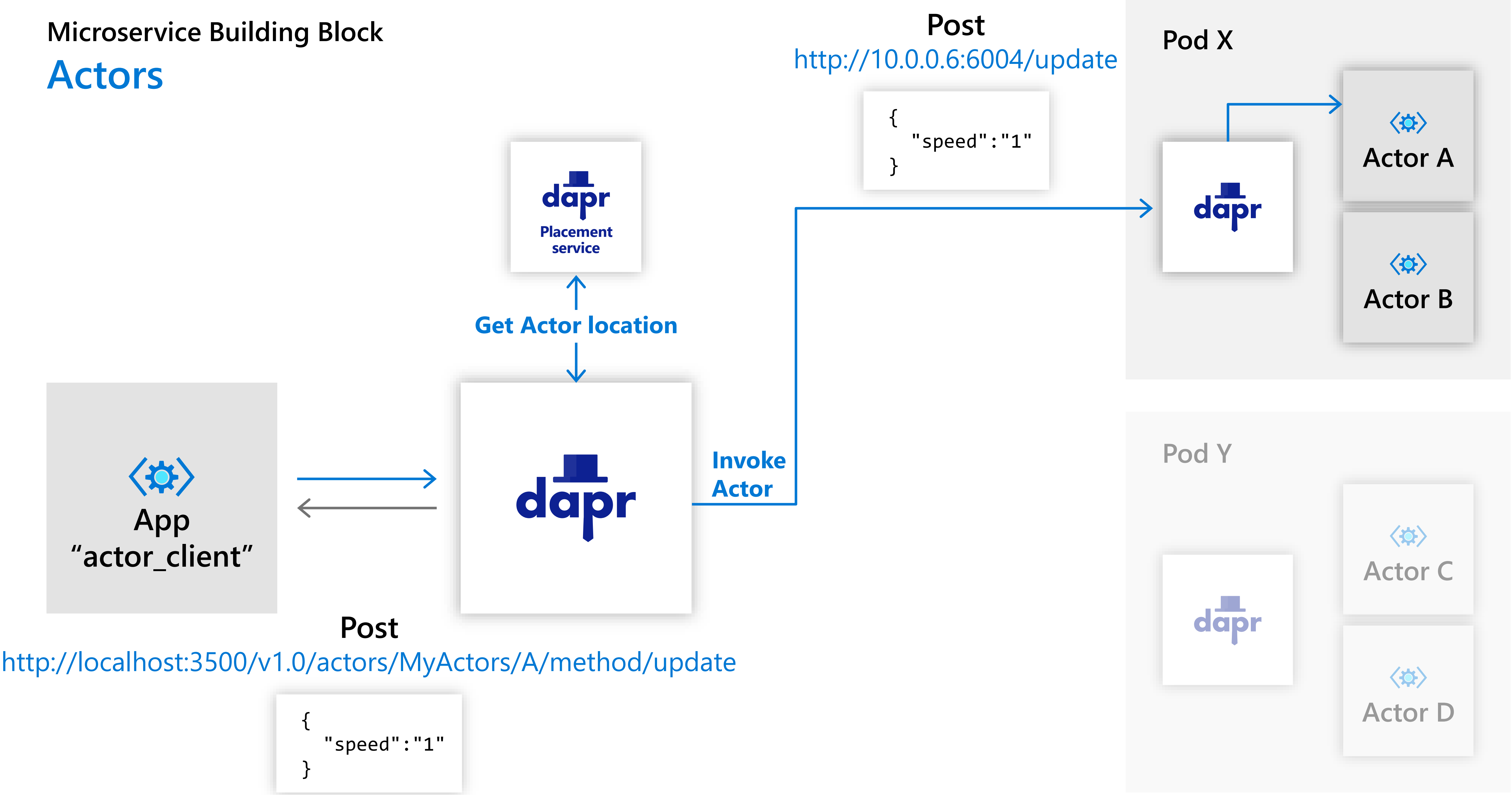
Output bindings to external resources including databases and queues

Actors

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

Microservice Building Block

Actors



Post
http://10.0.0.6:6004/update

```
{
  "speed":"1"
}
```

Pod X

Actor A

Actor B

dapr
Placement service

Get Actor location

App
“actor_client”

Invoke Actor

Pod Y

Actor C

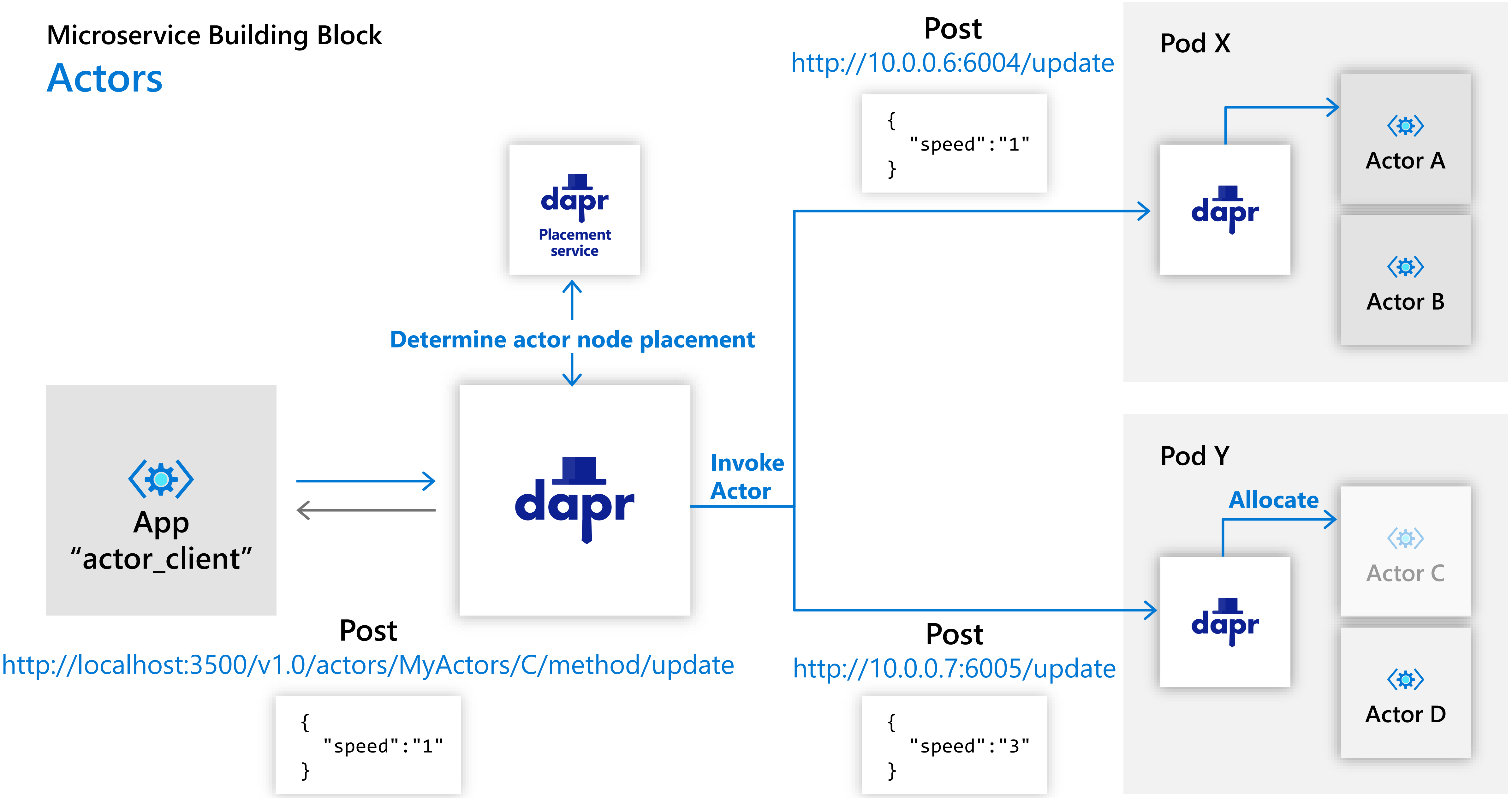
Actor D

Post
http://localhost:3500/v1.0/actors/MyActors/A/method/update

```
{
  "speed":"1"
}
```

Microservice Building Block

Actors



App
“actor_client”

Post
http://localhost:3500/v1.0/actors/MyActors/C/method/update

```
{
   "speed":"1"
}
```

Placement service

Determine actor node placement

Invoke
Actor

Post
http://10.0.0.6:6004/update

```
{
   "speed":"1"
}
```

Pod X

Actor A

Actor B

Pod Y

Allocate

Actor C

Actor D

Post
http://10.0.0.7:6005/update

```
{
   "speed":"3"
}
```

Dapr Actors

Demo

Microservice Building Blocks

Service-to-service invocation

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

State management

Create long running, stateless and stateful services

Publish and subscribe

Secure, scalable messaging between services

Resource bindings and triggers

Trigger code through events from a large array of inputs

Output bindings to external resources including databases and queues

Actors

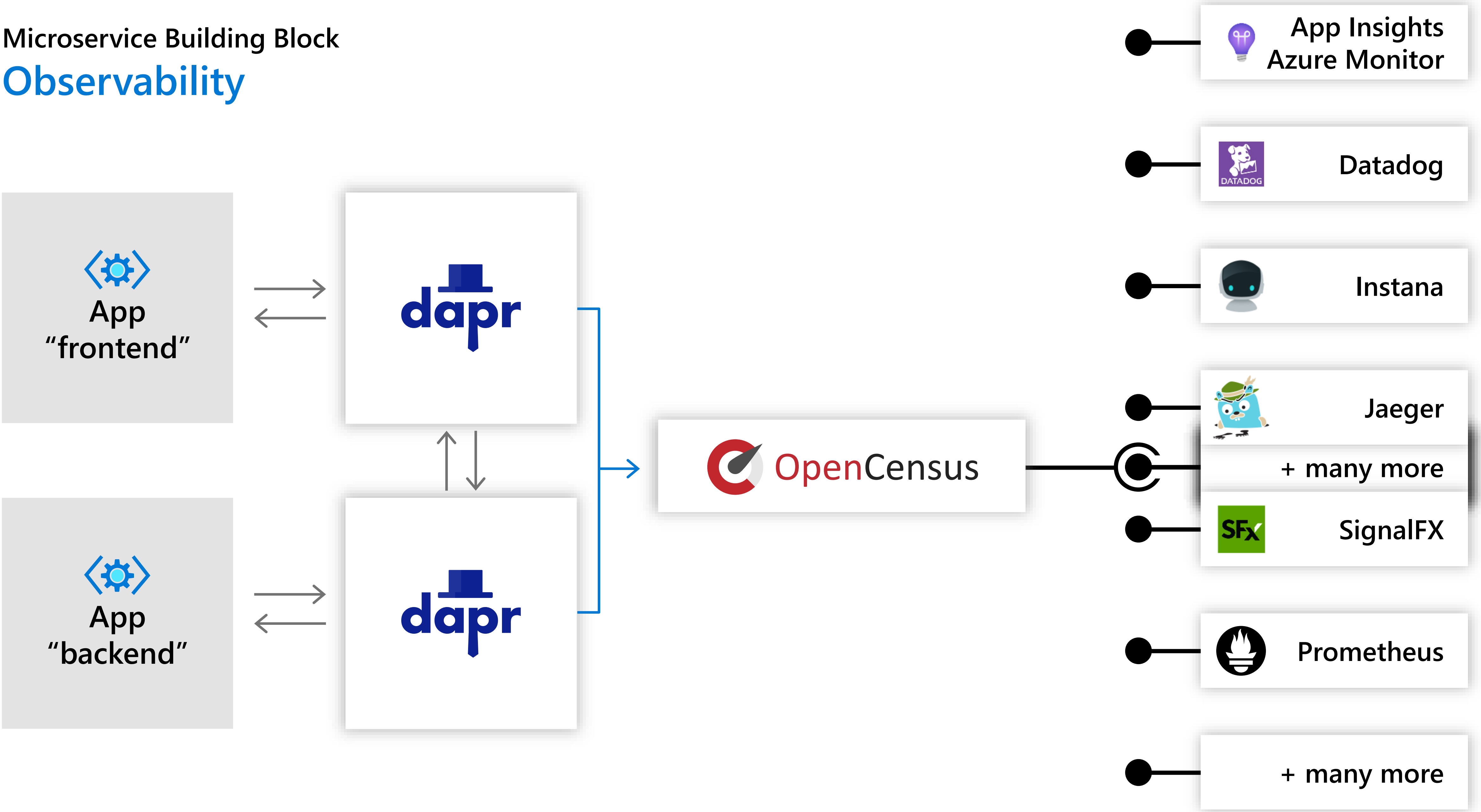
Encapsulate code and data in reusable actor objects as a common microservices design pattern

Observability

See and measure the message calls across components and networked services

Microservice Building Block

Observability



Microservice Building Blocks

Service-to-service invocation

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

State management

Create long running, stateless and stateful services

Publish and subscribe

Secure, scalable messaging between services

Resource bindings and triggers

Trigger code through events from a large array of inputs

Output bindings to external resources including databases and queues

Actors

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

Observability

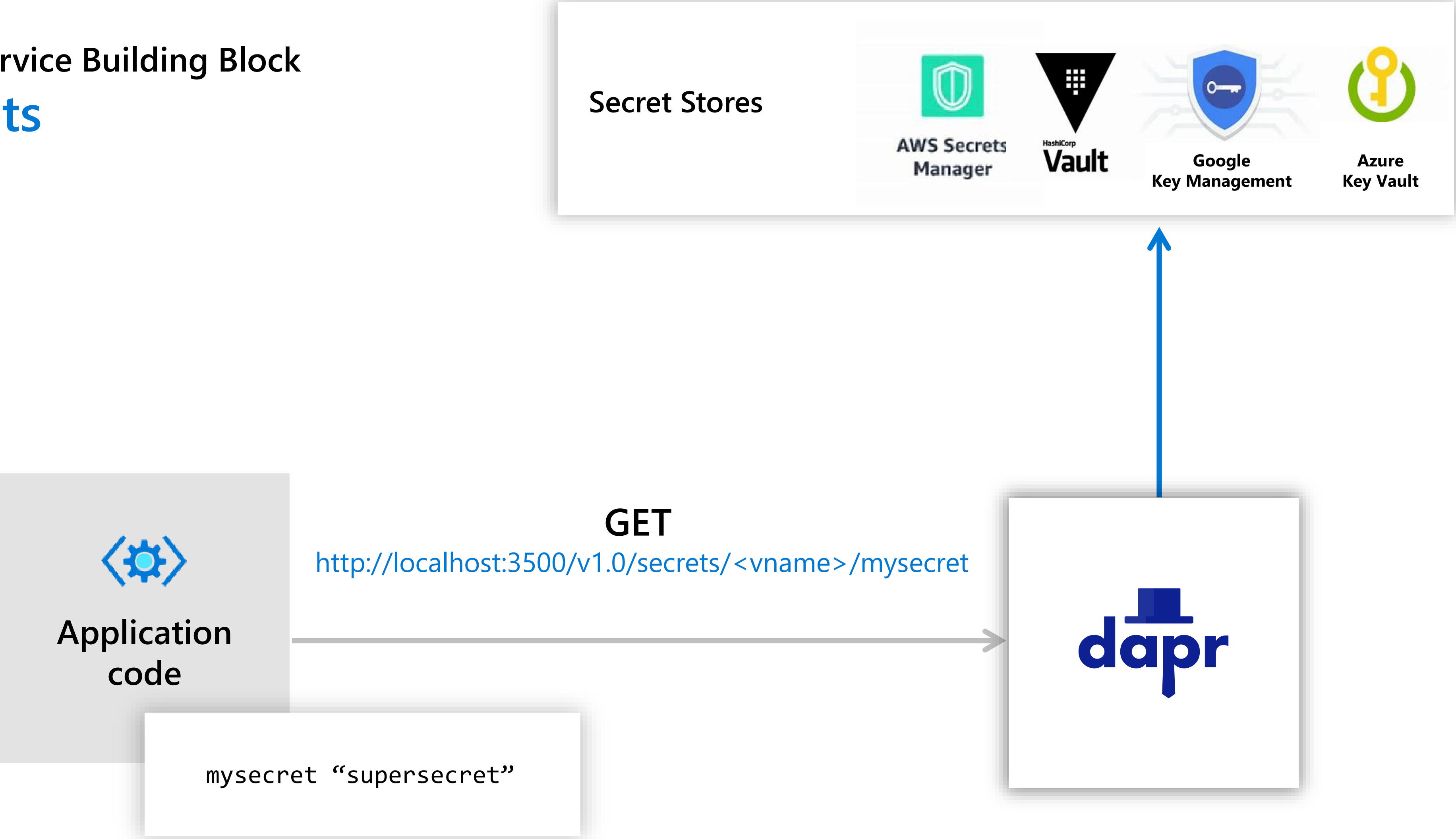
See and measure the message calls across components and networked services

Secrets

Securely access secrets from your application

Microservice Building Block

Secrets



Secret Stores

AWS Secrets Manager

HashiCorp Vault

Google Key Management

Azure Key Vault

GET

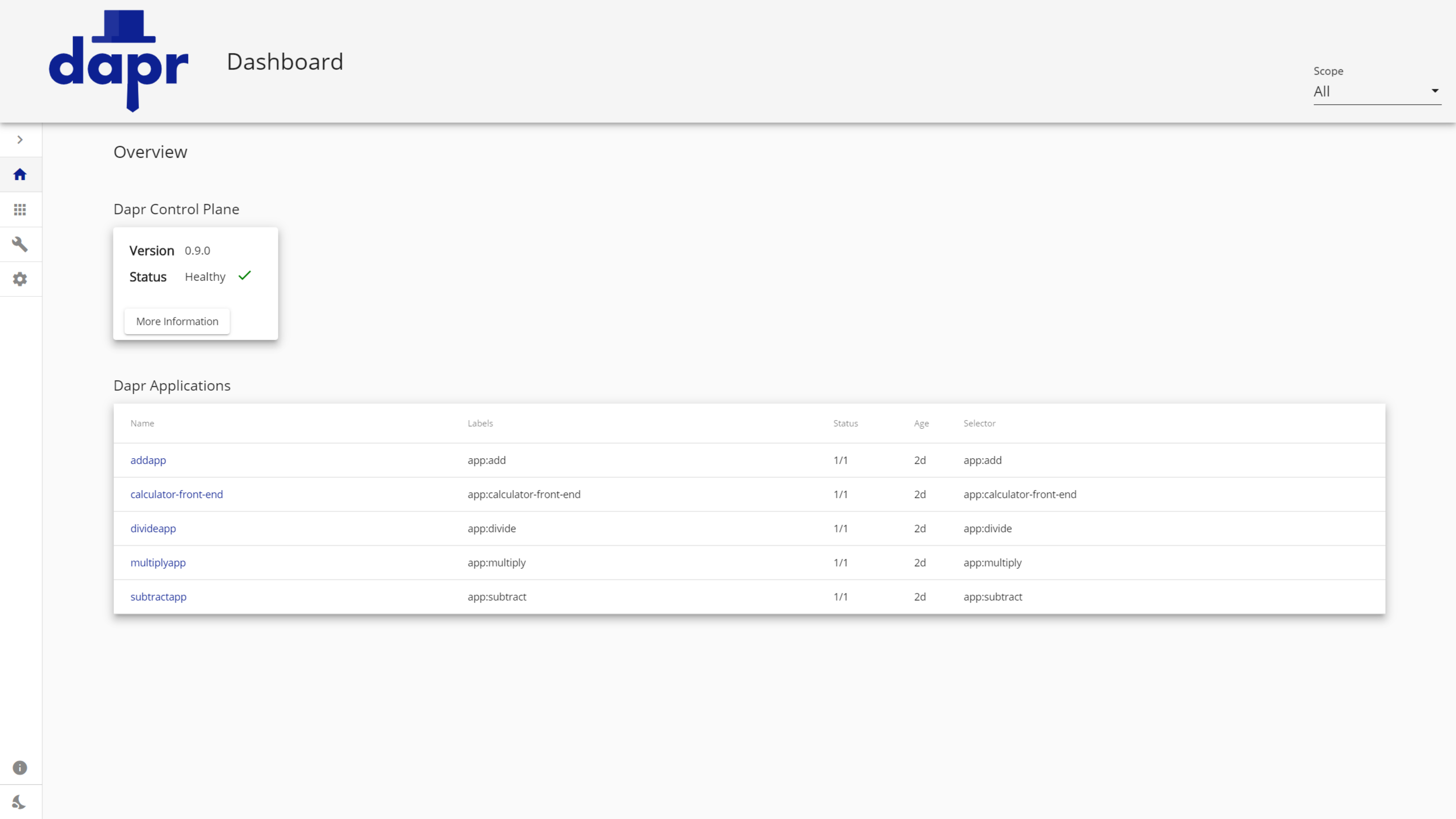
http://localhost:3500/v1.0/secrets/<vname>/mysecret

Application code

mysecret “supersecret”

dapr

Some other things…



Dashboard

Scope: All

Overview

Dapr Control Plane

Version 0.9.0

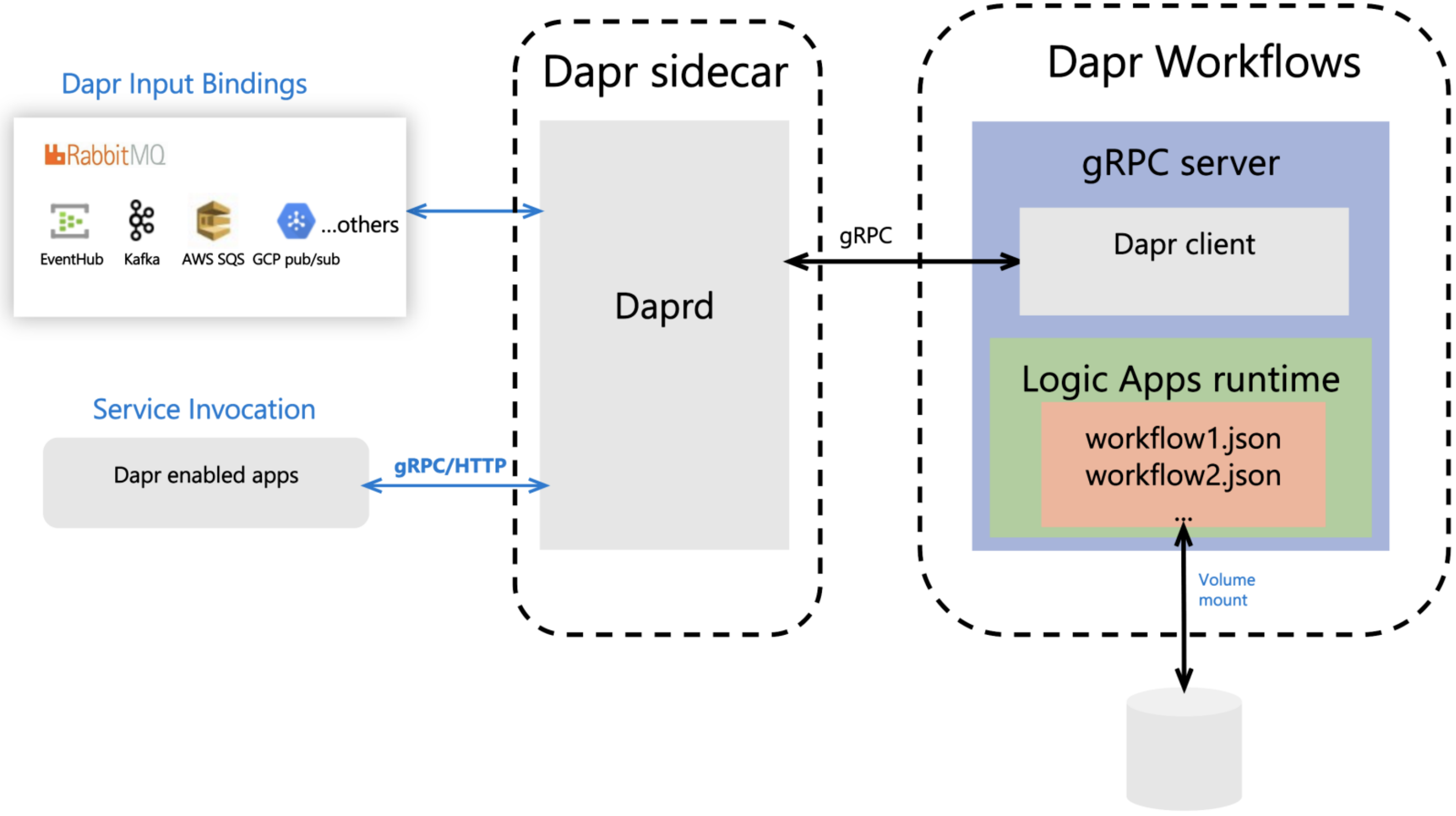
Status Healthy ✓

Dapr Applications

Name	Labels	Status	Age	Selector
addapp	app:add	1/1	2d	app:add
calculator-front-end	app:calculator-front-end	1/1	2d	app:calculator-front-end
divideapp	app:divide	1/1	2d	app:divide
multiplyapp	app:multiply	1/1	2d	app:multiply
subtractapp	app:subtract	1/1	2d	app:subtract

Advanced Topics

Dapr Workflows

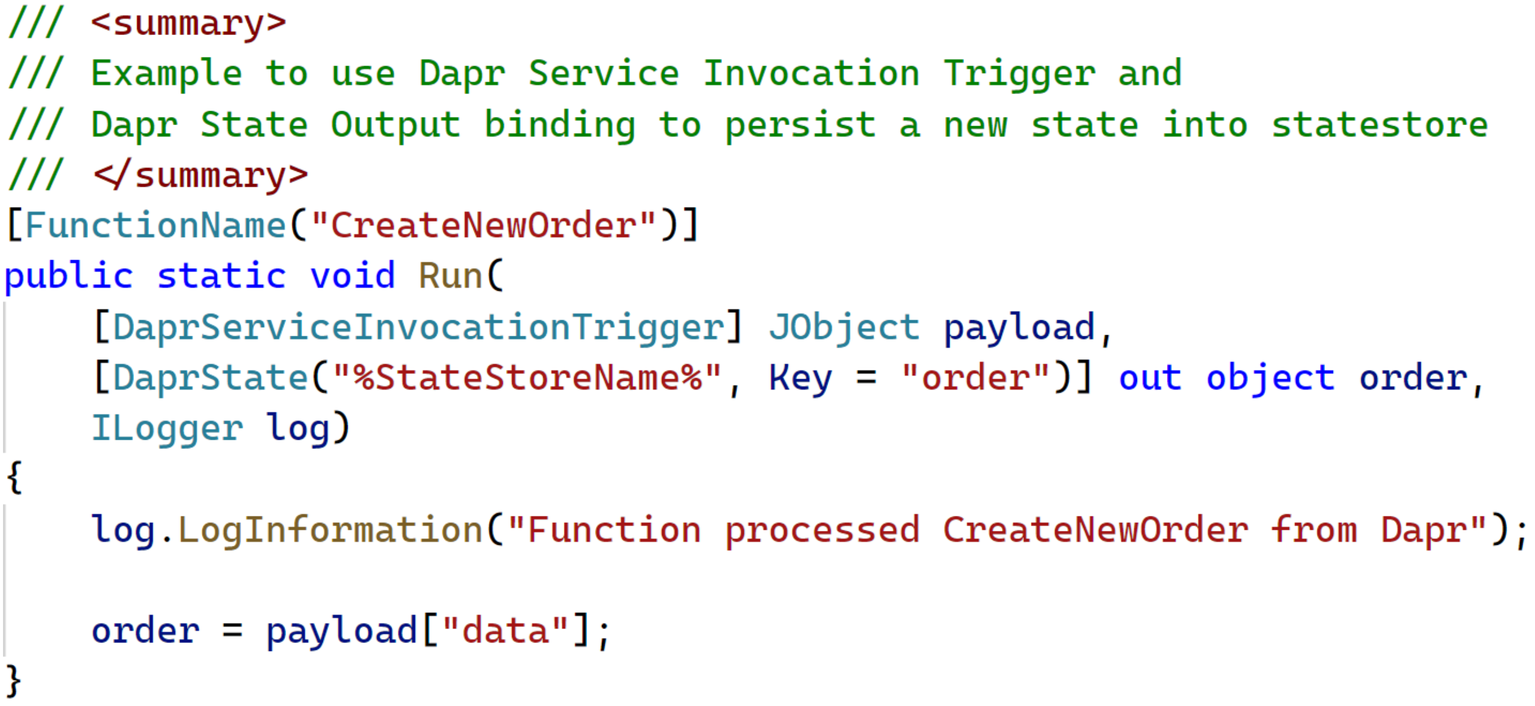


Invoking workflows using Dapr bindings

```
apiVersion: dapr.io/v1alpha1
 kind: Component
metadata:
 name: workflow1
spec:
 type: bindings.kafka
 metadata:
 - name: topics
   value: topic1
 - name: brokers
   value: localhost:9092
 - name: consumerGroup
   value: group1
 - name: authRequired
   value: "false"
```

Advanced Topics

Azure Functions Dapr Extensions



```
/// <summary>
/// Example to use Dapr Service Invocation Trigger and
/// Dapr State Output binding to persist a new state into statestore
/// </summary>
[FunctionName("CreateNewOrder")]
public static void Run(
    [DaprServiceInvocationTrigger] JObject payload,
    [DaprState("%StateStoreName%", Key = "order")] out object order,
    ILogger log)
{
    log.LogInformation("Function processed CreateNewOrder from Dapr");

    order = payload["data"];
}
```

Function Triggers:

- daprServiceInvocationTrigger
- daprTopicTrigger
- daprBindingTrigger

Function Bindings

- daprState (Input+Output)
- daprSecret (Input)
- daprInvoke (Output)
- daprPublish (Output)
- daprBinding (Output)

Advanced Topics

Benchmark

Service Invocation – Test Setup

- Sidecar resource constraints: 0.5 CPU, 256 MB RAM
- Dapr features enabled: TLS, Tracing, Metrics
- 1,000 RPS (16 client connections)
- Payload: 1 KB playoad
- Test Duration: 30 mins

Results

- End2End added latency (both sidecars): 4.1ms for the 90th percentile
- Memory utilization: 18MB
- CPU utilization: 0.3 (300 millicores)

Upcoming

- Actors
- Pub/Sub
- State

Roadmap

Dapr v1.0 runtime

- Retry Pattern in Dapr runtime for all building blocks
- Access lists for inter-service invocation
- Allow list for secrets
- IoT Edge module to deploy Dapr as a side-car
- More tests / more performance benchmarks / another security assessment

Azure Functions extensions v1.0

Combine Dapr and Functions on Kubernetes and IoT Edge

Log App workflows v1.0

Build event driven workflow apps on Kubernetes and IoT Edge

Dapr in Azure managed services

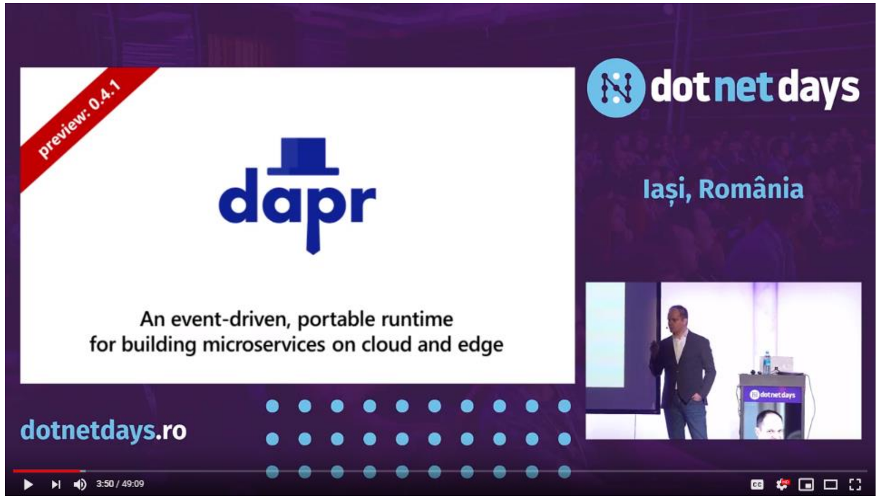
Integrated into Azure App Service, Azure Functions, AKS

Learn More about Dapr:

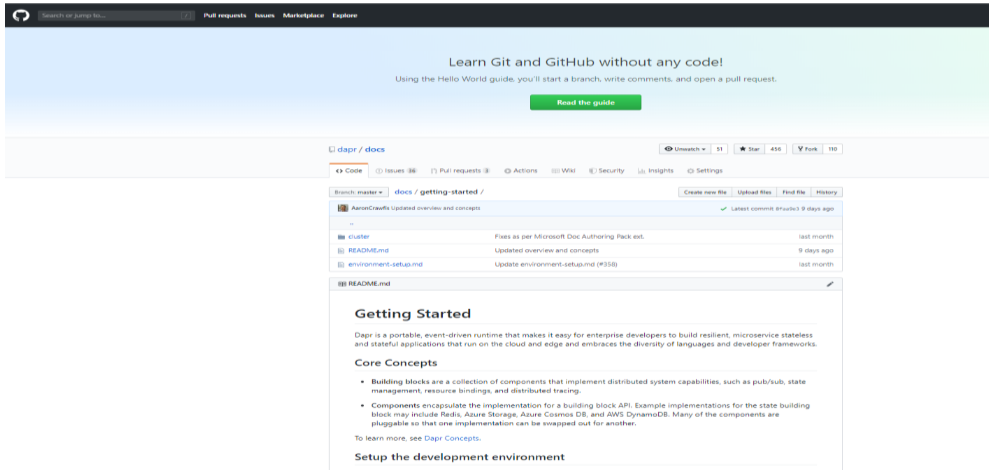
https://myignite.techcommunity.microsoft.com/sessions/82059

https://www.youtube.com/watch?v=CgqI7nen-Ng

https://www.youtube.com/watch?v=a2OZ0VI4JTg


preview: 0.4.1
dapr
An event-driven, portable runtime
for building microservices on cloud and edge
dotnetdays
Iași, România
dotnetdays.ro
3:50 / 49:09

https://github.com/dapr/docs/tree/master/getting-started



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

dapr / docs

docs / getting-started /

AaronCrawfis Updated overview and concepts ✓ Latest commit 8faa9e3 9 days ago

..

cluster	Fixes as per Microsoft Doc Authoring Pack ext.	last month
README.md	Updated overview and concepts	9 days ago
environment-setup.md	Update environment-setup.md (#358)	last month

README.md

Getting Started

Dapr is a portable, event-driven runtime that makes it easy for enterprise developers to build resilient, microservice stateless and stateful applications that run on the cloud and edge and embraces the diversity of languages and developer frameworks.

Core Concepts

- **Building blocks** are a collection of components that implement distributed system capabilities, such as pub/sub, state management, resource bindings, and distributed tracing.
- **Components** encapsulate the implementation for a building block API. Example implementations for the state building block may include Redis, Azure Storage, Azure Cosmos DB, and AWS DynamoDB. Many of the components are pluggable so that one implementation can be swapped out for another.

To learn more, see Dapr Concepts.

Setup the development environment

Bi-Weekly Dapr Community Calls

Learn more & Join:	https://aka.ms/dapr-community-call
Meeting Recordings:	http://aka.ms/dapr-recordings
Meeting Notes:	https://aka.ms/dapr-meeting-notes

Next:

- Tuesday September 29th 10am Pacific Time (PST)
  > also the release date of v0.11

Get involved

GitHub | https://github.com/dapr/dapr#community

Thank you!

Ricardo Niepel
Cloud Solution Architect - Azure App Dev
ricardo.niepel@microsoft.com

@RicardoNiepel RicardoNiepel