Microsoft

Dapr: Distributed Application Runtime

Chris Wiederspan
Azure App Dev Specialist

https://aka.ms/learn-dapr
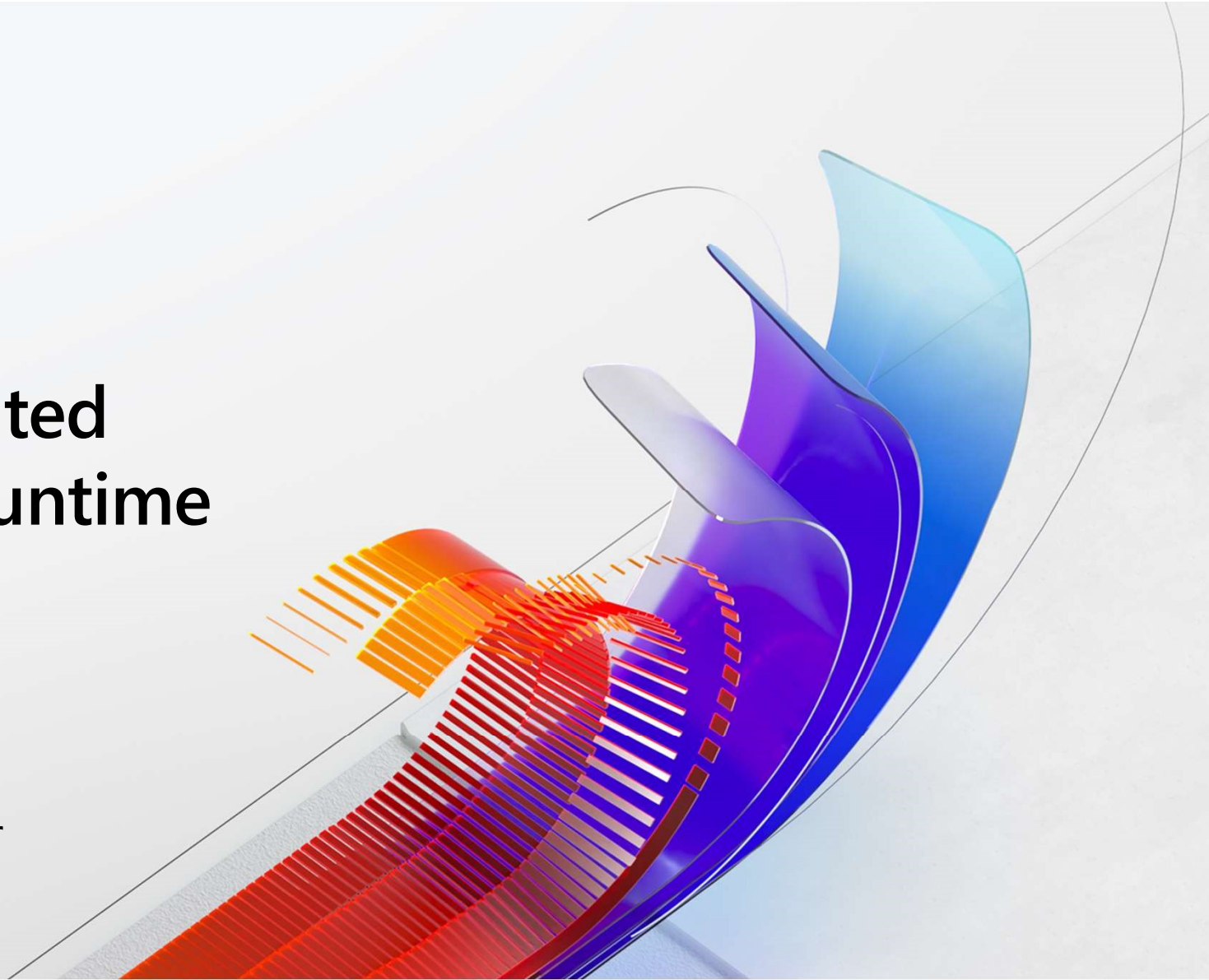
Software Innovation Collision

Containers
(Standardization of the application packaging)

DevOps
(Optimizing how applications are build, delivered and Managed)

Better Software

Microservices
(Optimizing how applications are composed)

Kubernetes Orchestration
(Efficient ways to deliver Infrastructure for Applications)
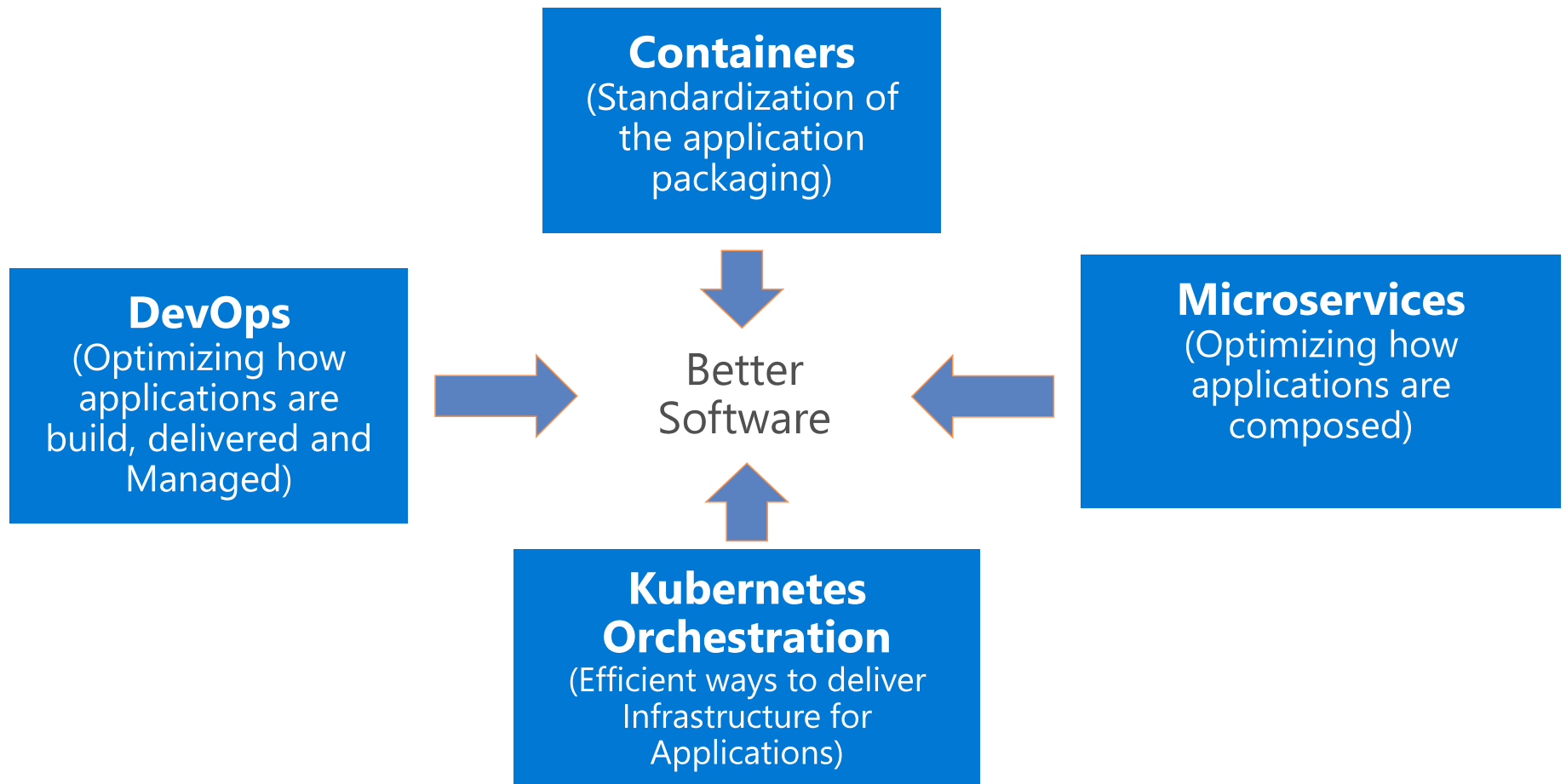
State of Enterprise Developers

Being asked to develop resilient, scalable, microservice-based apps that interact with services

Use multiple languages and frameworks during development

Focus on building apps not infrastructure

What is holding back microservice development?

Have limited tools and programming model runtimes to build distributed applications

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

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

Dapr is a portable, event-driven runtime that
makes it easy for developers to build
resilient, microservice applications that run
on the cloud and edge

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

Application code

Microservices written in

Any code or framework...   =GO   node JS   python   .NET   Java   C++

HTTP API   **gRPC API**

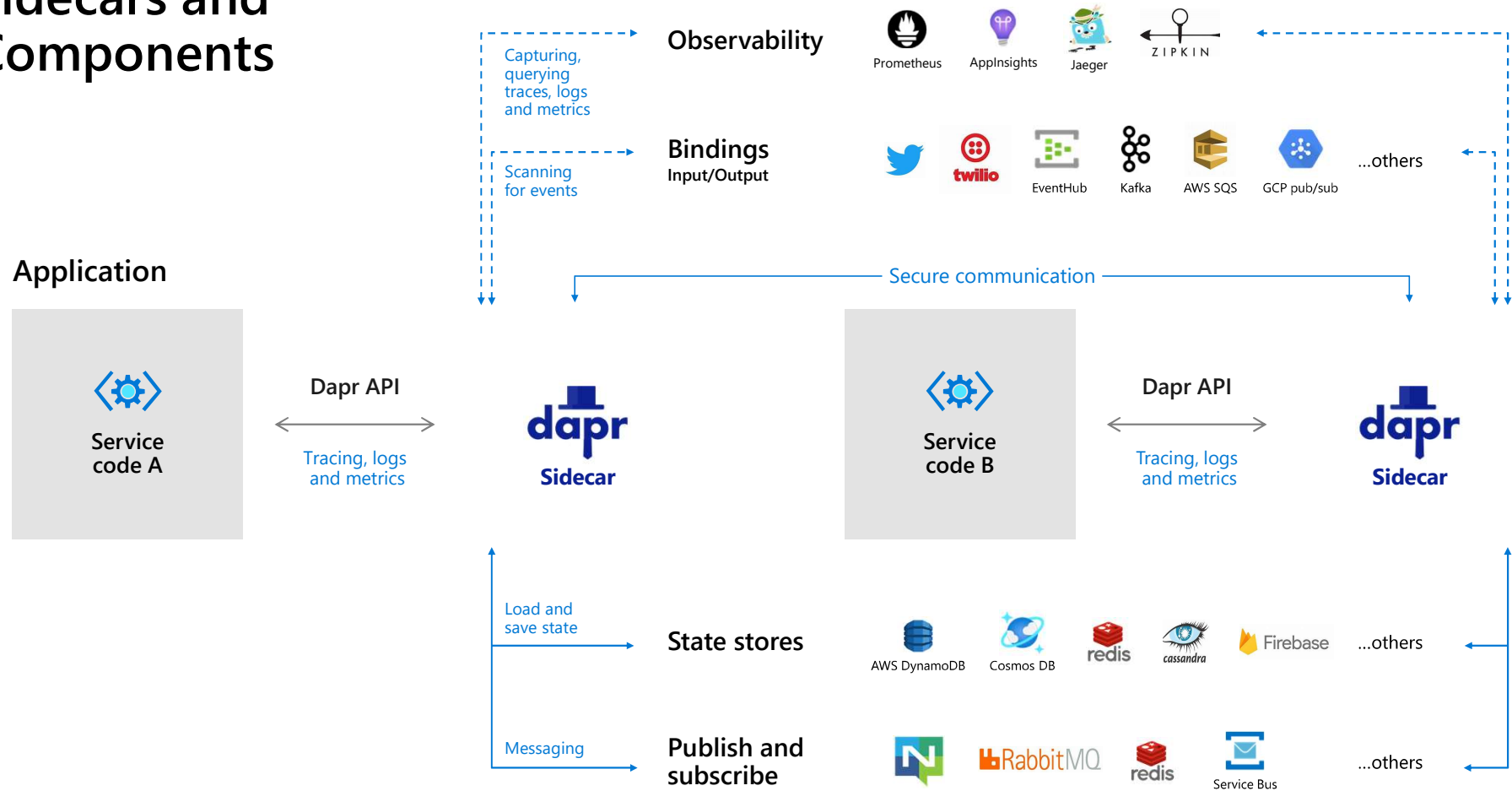dapr | Service-to-service invocation | State management | Publish and subscribe | Resource bindings and triggers | Actors | Observability | Secrets | Extensible
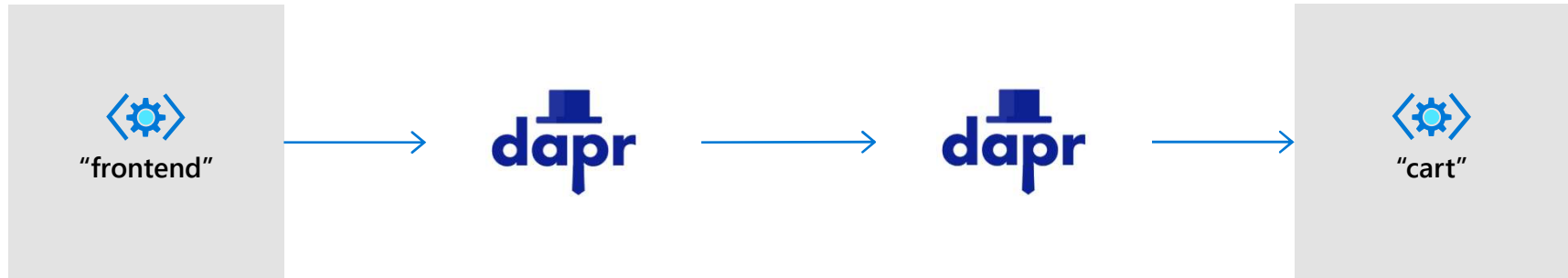
Sidecars and Components

Observability

Capturing, querying traces, logs and metrics

Prometheus AppInsights Jaeger ZIPKIN

Bindings
Input/Output

Scanning for events

Twilio EventHub Kafka AWS SQS GCP pub/sub ...others

Application

Secure communication

| Service code A | | Dapr Sidecar | | Service code B | | Dapr Sidecar |

Dapr API

Tracing, logs and metrics

Dapr API

Tracing, logs and metrics

State stores

Load and save state

AWS DynamoDB Cosmos DB redis cassandra Firebase ...others

Publish and subscribe

Messaging

RabbitMQ redis Service Bus ...others

Microservice building blocks
Service invocation

"frontend"

dapr

dapr

"cart"

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

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

Post
http://10.0.0.2:8000/checkout

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

Microservice building blocks
State management: key/value

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

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

App "myApp"

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 stores   AWS DynamoDB   Cosmos DB   redis   cassandra   Firebase   ...others
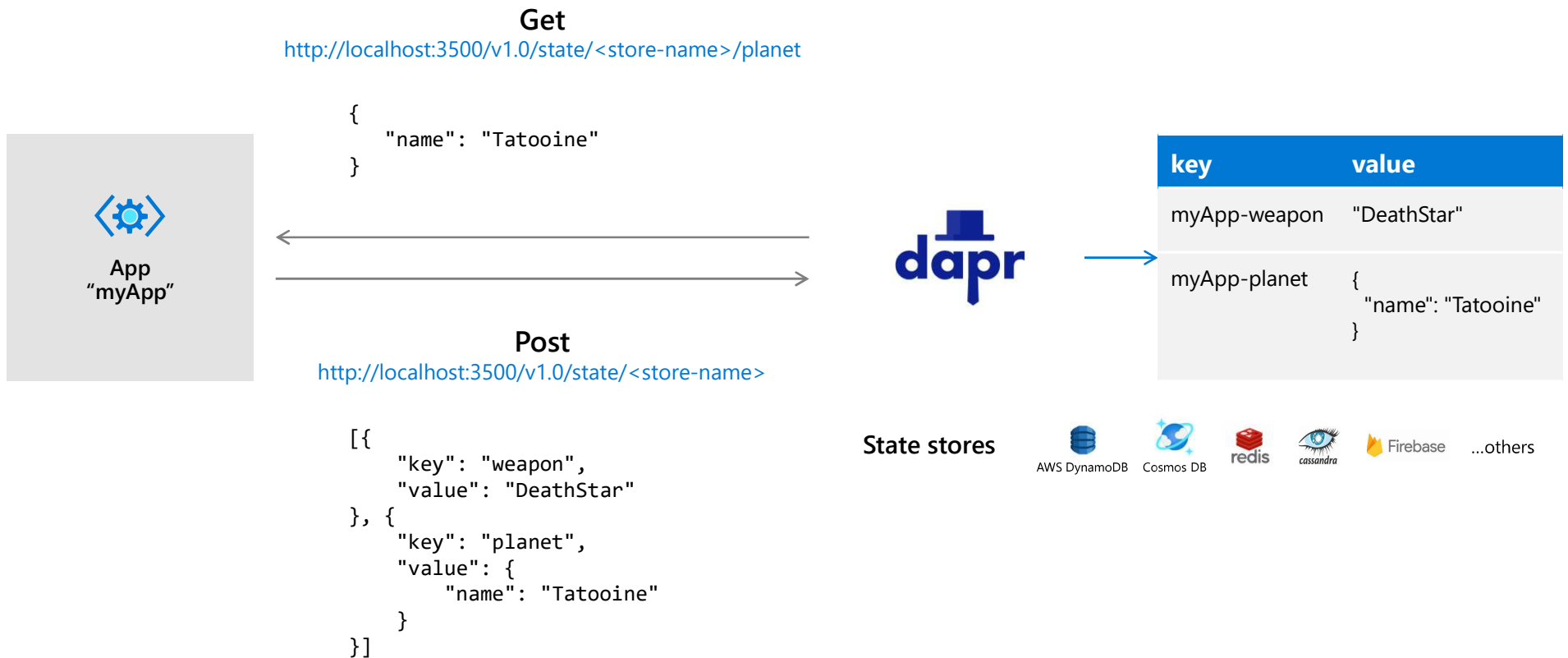
Microservice building blocks
Publish and Subscribe

Publish

cart

Post
http://localhost:3500/v1.0/publish/

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

Subscribe

shipping

Post
http://10.0.0.5:8005/order

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

Post
http://10.0.0.4:8004/order

email
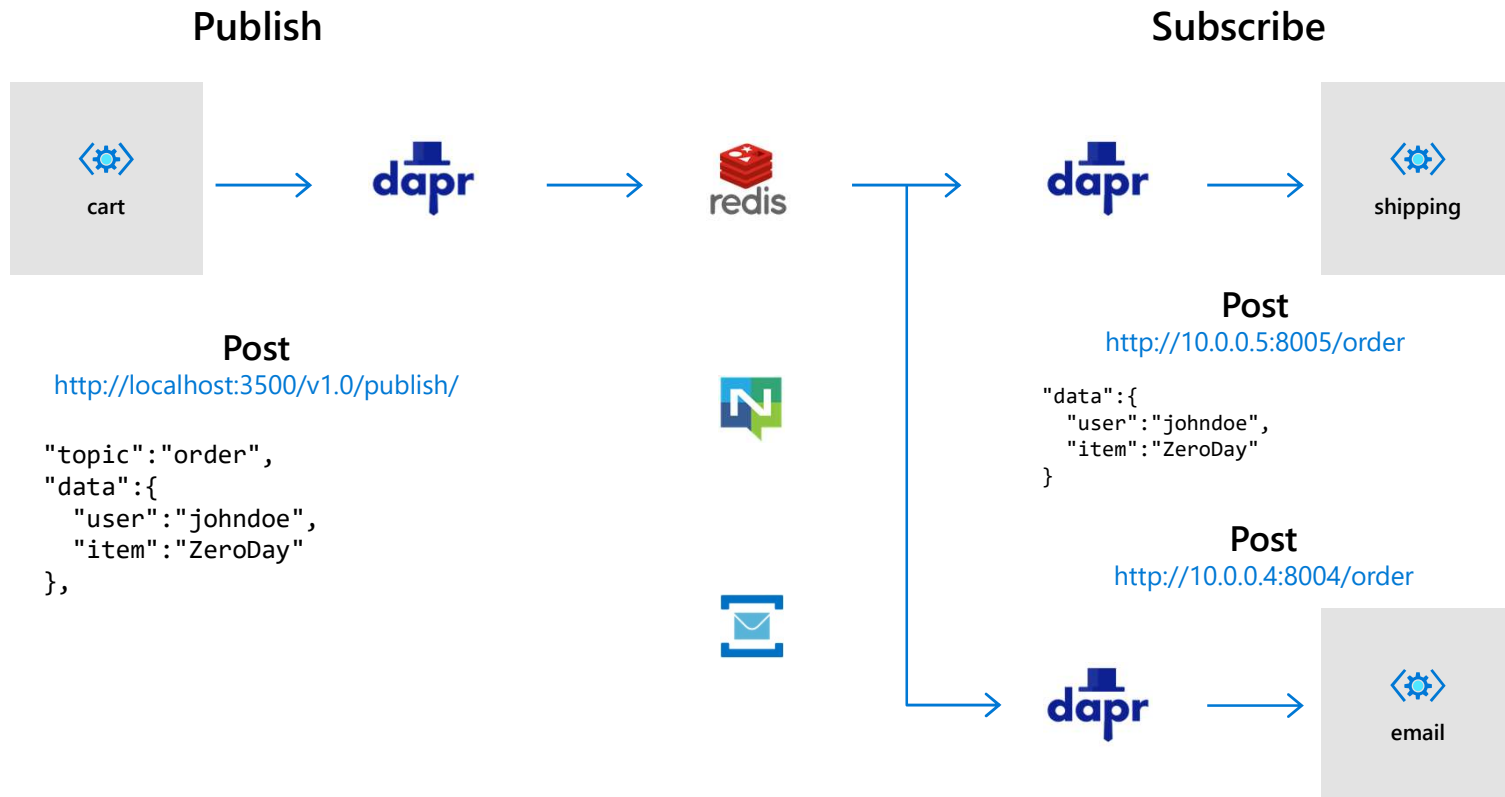
Join the community

github.com/dapr

aka.ms/dapr-community

@daprdev

Microsoft