

# Entwicklung von ASP.NET Core Microservices mit Dapr: Eine praktische Anleitung

Marc Müller  
Principal Consultant



[marc.mueller@4tecture.ch](mailto:marc.mueller@4tecture.ch)  
[@muellermarc](https://twitter.com/muellermarc)  
[www.4tecture.ch](http://www.4tecture.ch)

4tecture<sup>®</sup>  
empower your software solutions

A black and white portrait of Marc Müller, a middle-aged man with short, dark hair, wearing glasses and a dark polo shirt under a light-colored jacket. He is smiling and looking towards the camera.

About me:

Marc Müller  
Principal Consultant  
@muellermarc



4tecture<sup>©</sup>  
empower your software solutions

Our Products:

Multi-Tenant OpenID  
Connect Identity Provider



ProAuth

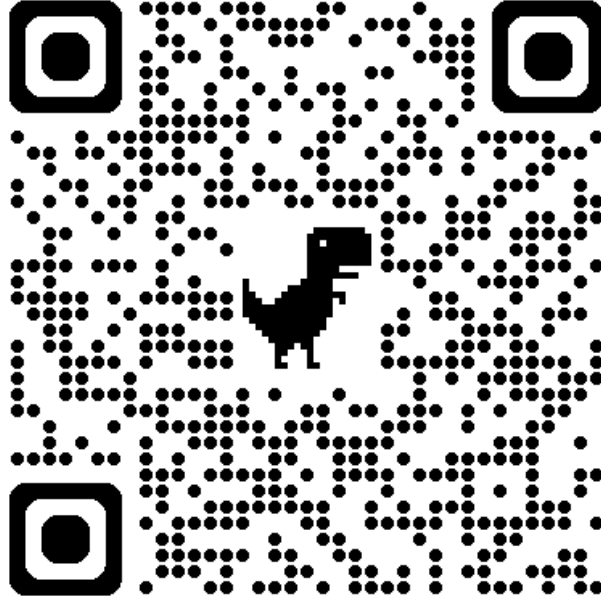
[www.proauth.net](http://www.proauth.net)

Enterprise Application  
Framework for .NET



[www.reafx.net](http://www.reafx.net)

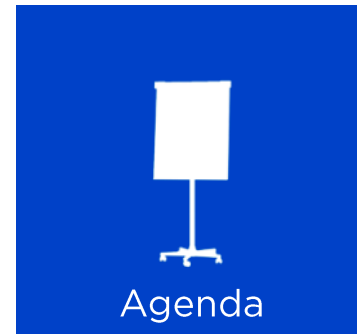
# Slide Download



<https://www.4tecture.ch/events/ddc24dapr>

# Agenda

- Intro
- Dapr Basics
- Hands-On!
- Building Blocks
- Conclusion



A close-up, low-angle shot of several rowers in a boat, wearing blue and red uniforms, pulling their oars. The oars are grey with yellow handles and are connected to a black metal frame. The background is a bright, slightly blurred body of water.

Dapr

# Intro

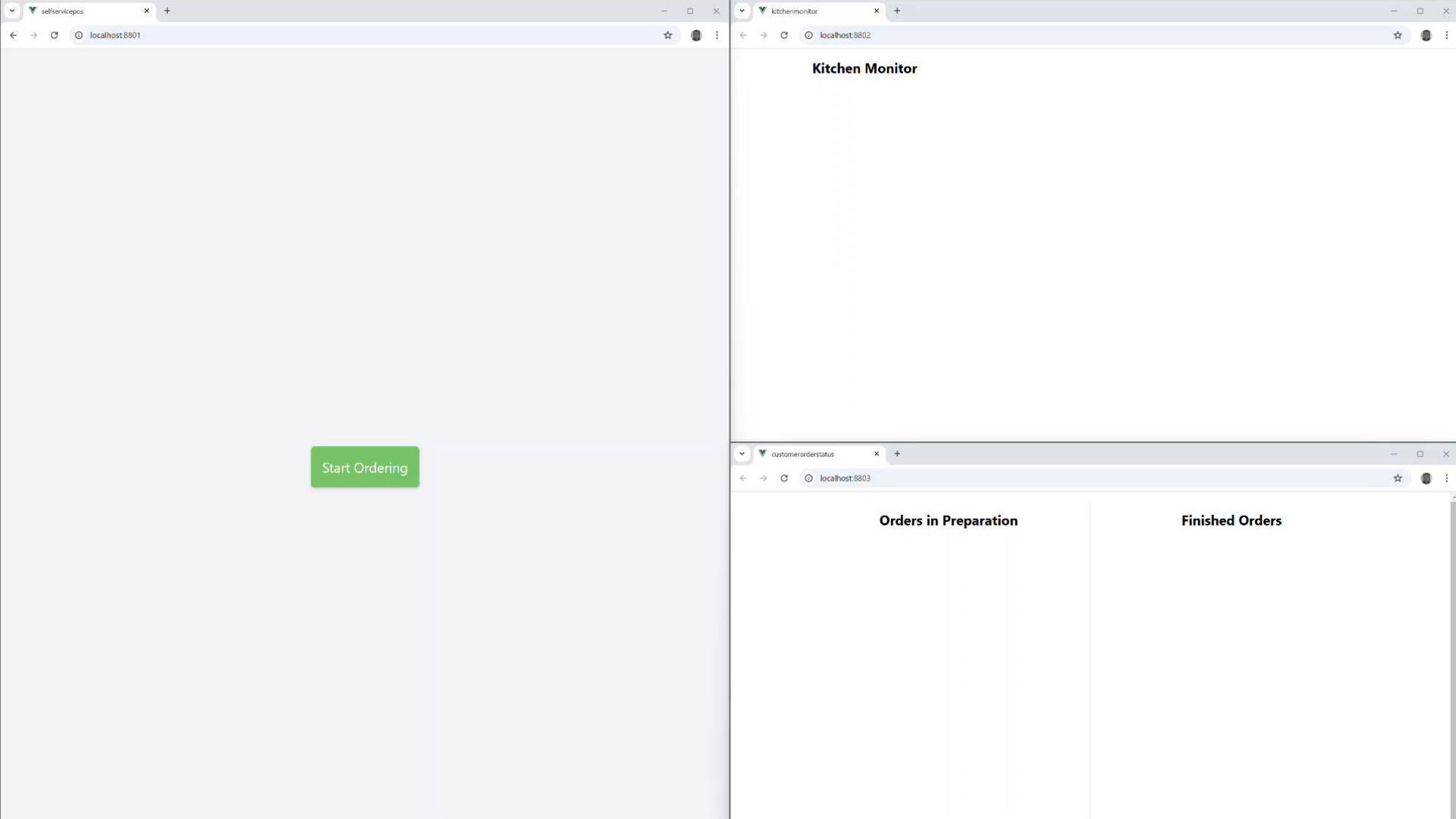
4tecture<sup>®</sup>  
empower your software solutions





# DEMO

Fast Food Application





ARCHITECTING AND IMPLEMENTING  
DISTRIBUTED APPLICATIONS  
IS NOT STRESSFUL AT ALL

John, 25 years old



# Challenges with Distributed Apps

- Lots of components / services
- Service-to-service communication
- Decoupling by using events
- Handling state across multiple instances
- Stateful services / actors
- Local dev environment vs. cloud environment

# dapr.io



Portable, event-driven runtime



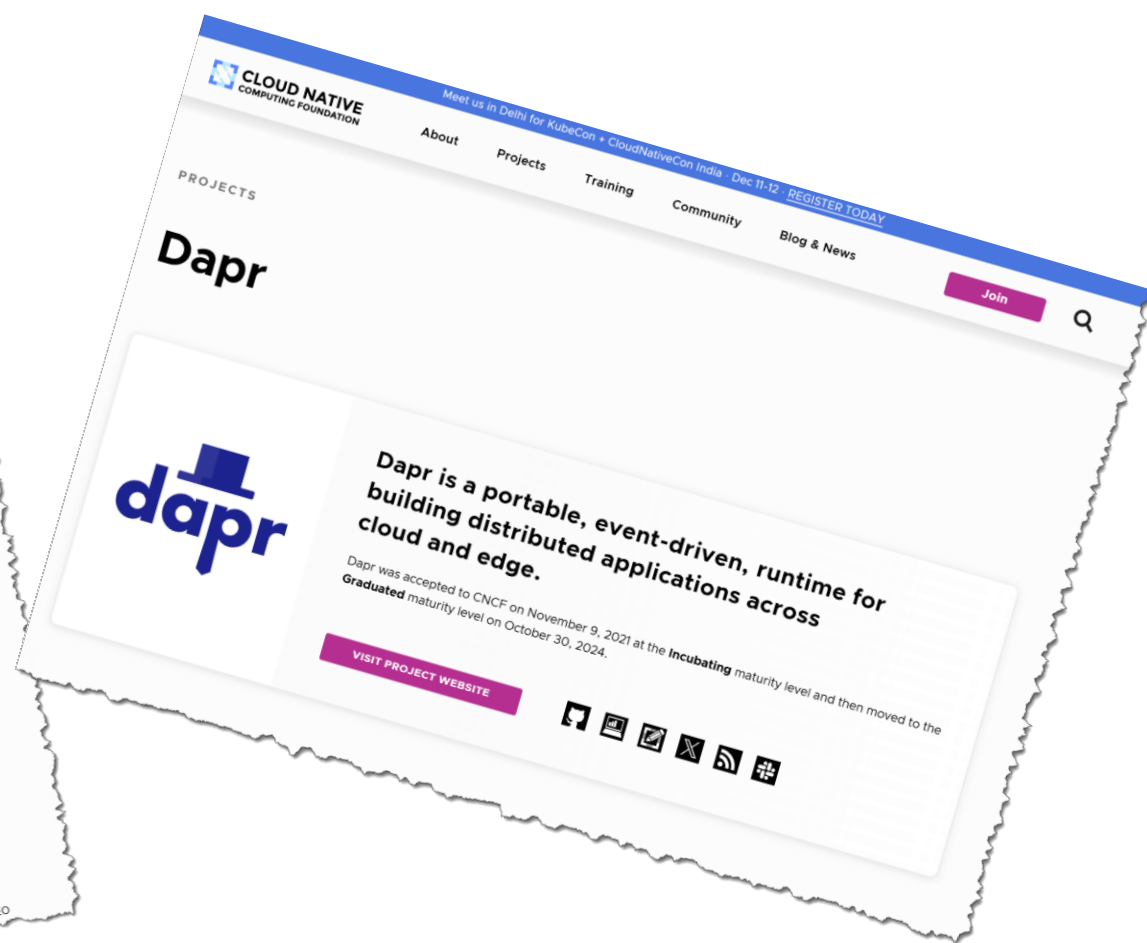
Build connected distributed applications faster



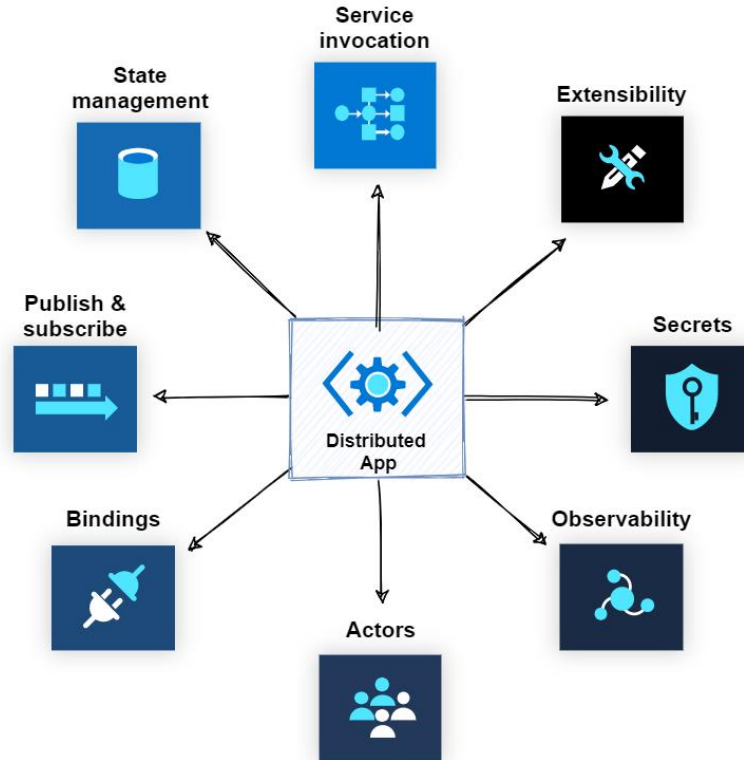
APIs for solving distributed application challenges



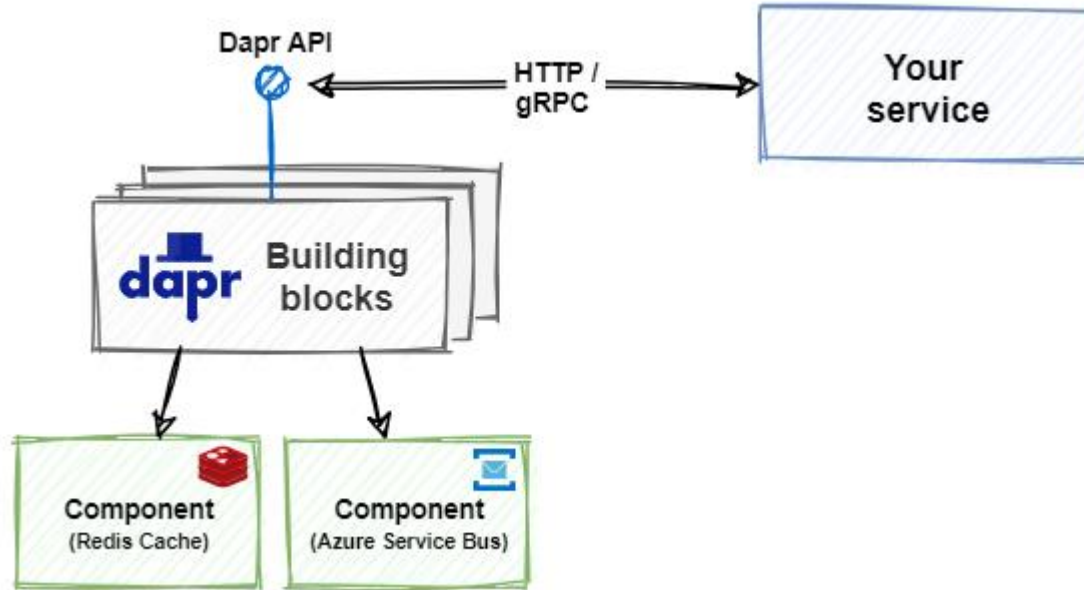
Cloud and Edge



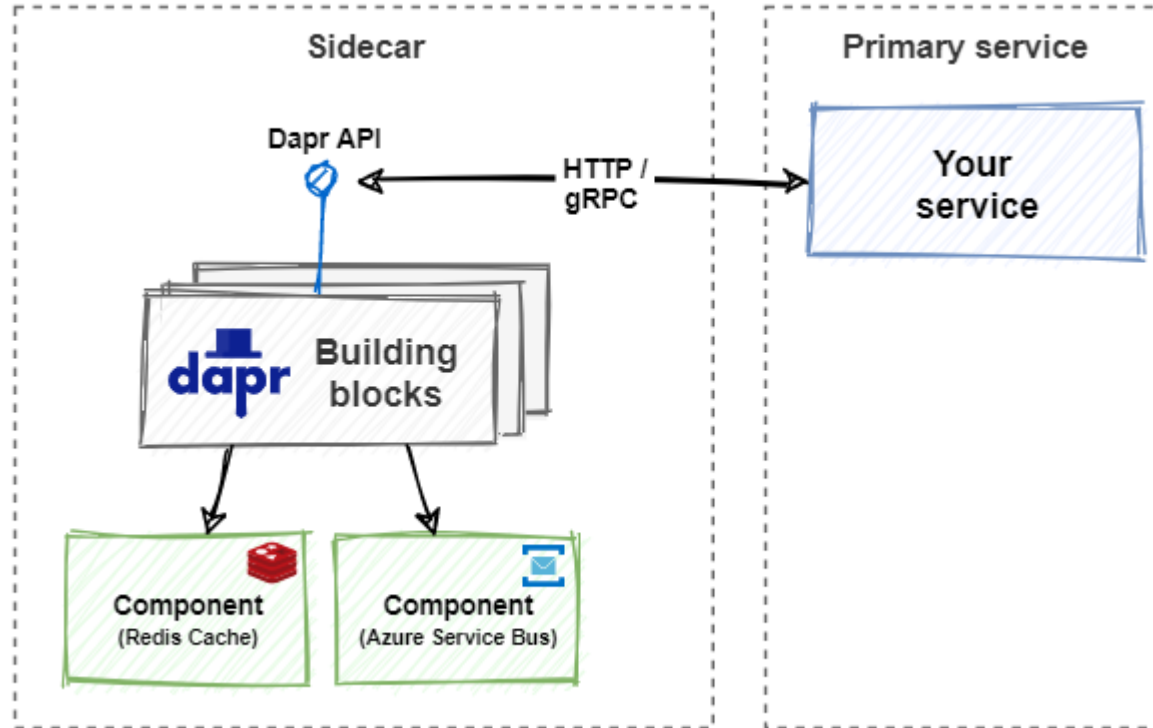
# Dapr Building Blocks



# Dapr Building Blocks Abstraction



# Dapr Sidecar Architecture







# Dapr Basics

4tecture<sup>®</sup>  
empower your software solutions

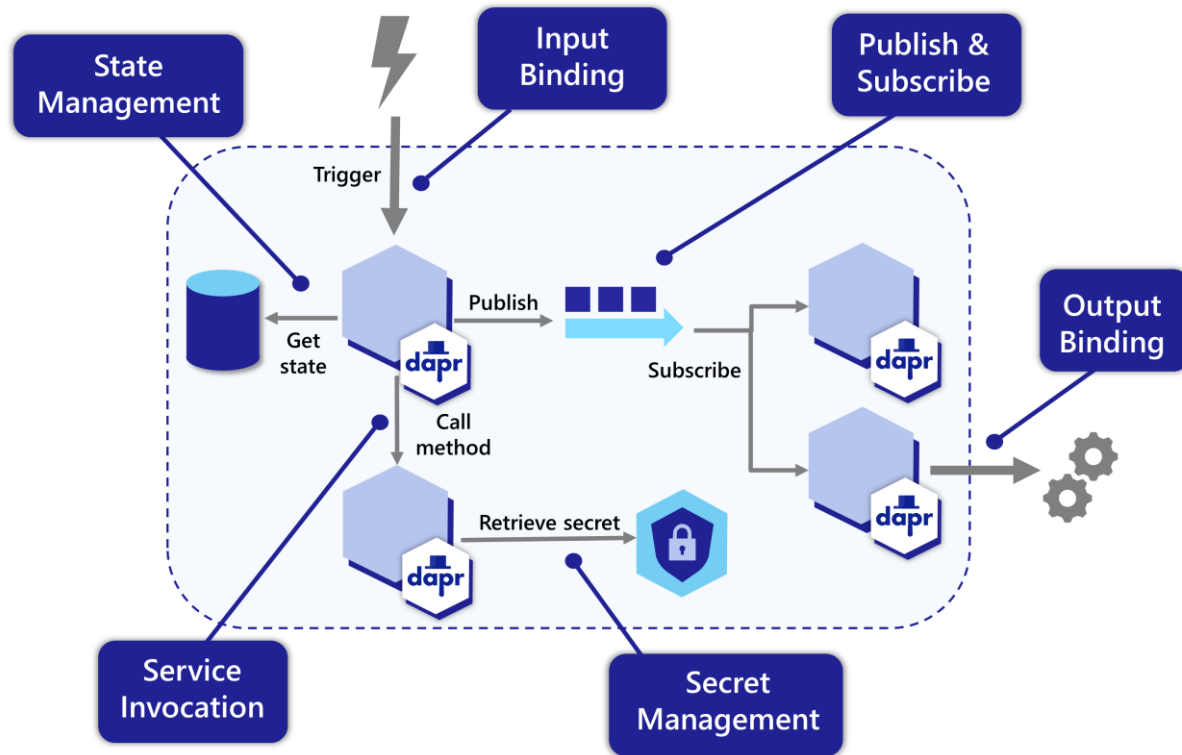
# Dapr

- [dapr.io](https://dapr.io)
- Open source
- Originated at Microsoft
- Cloud Native Computing Foundation (CNCF) – graduated maturity level

# Dapr – High Level Definition

- Any language or framework
- Portable APIs
- Building blocks applying best practices
  - Use the blocks you need
  - No big bang framework
- Platform agnostic
- Extensible and pluggable components

# Dapr Building Blocks Overview



# Single API - Multiple Components

```
var weatherForecast =  
await daprClient.GetStateAsync<WeatherForecast>("statestore", "AMS");  
  
daprClient.SaveStateAsync("statestore", "AMS", weatherForecast);
```

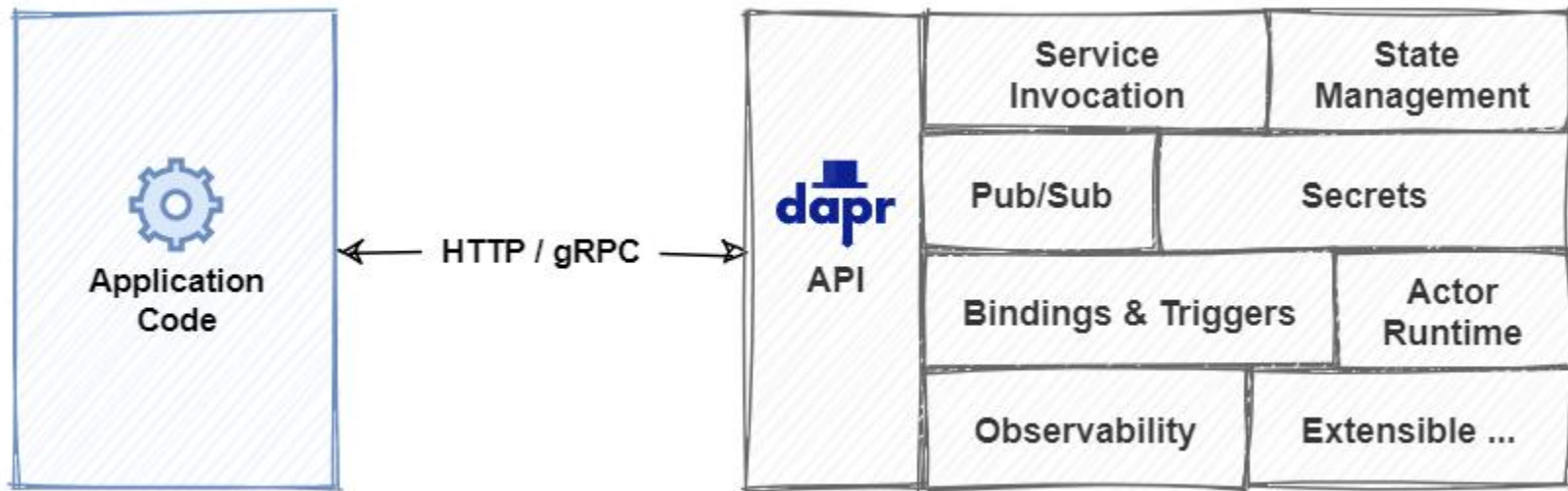
- AWS DynamoDB
- Aerospike
- Azure Blob Storage
- Azure CosmosDB
- Azure Table Storage
- Cassandra
- Cloud Firestore (Datastore mode)
- CloudState
- Couchbase
- Etcd
- HashiCorp Consul
- Hazelcast
- Memcached
- MongoDB
- PostgreSQL
- Redis
- RethinkDB
- SQL Server
- Zookeeper

The Dapr sidecar provides built-in  
**security**, **resiliency** and **observability**  
capabilities.

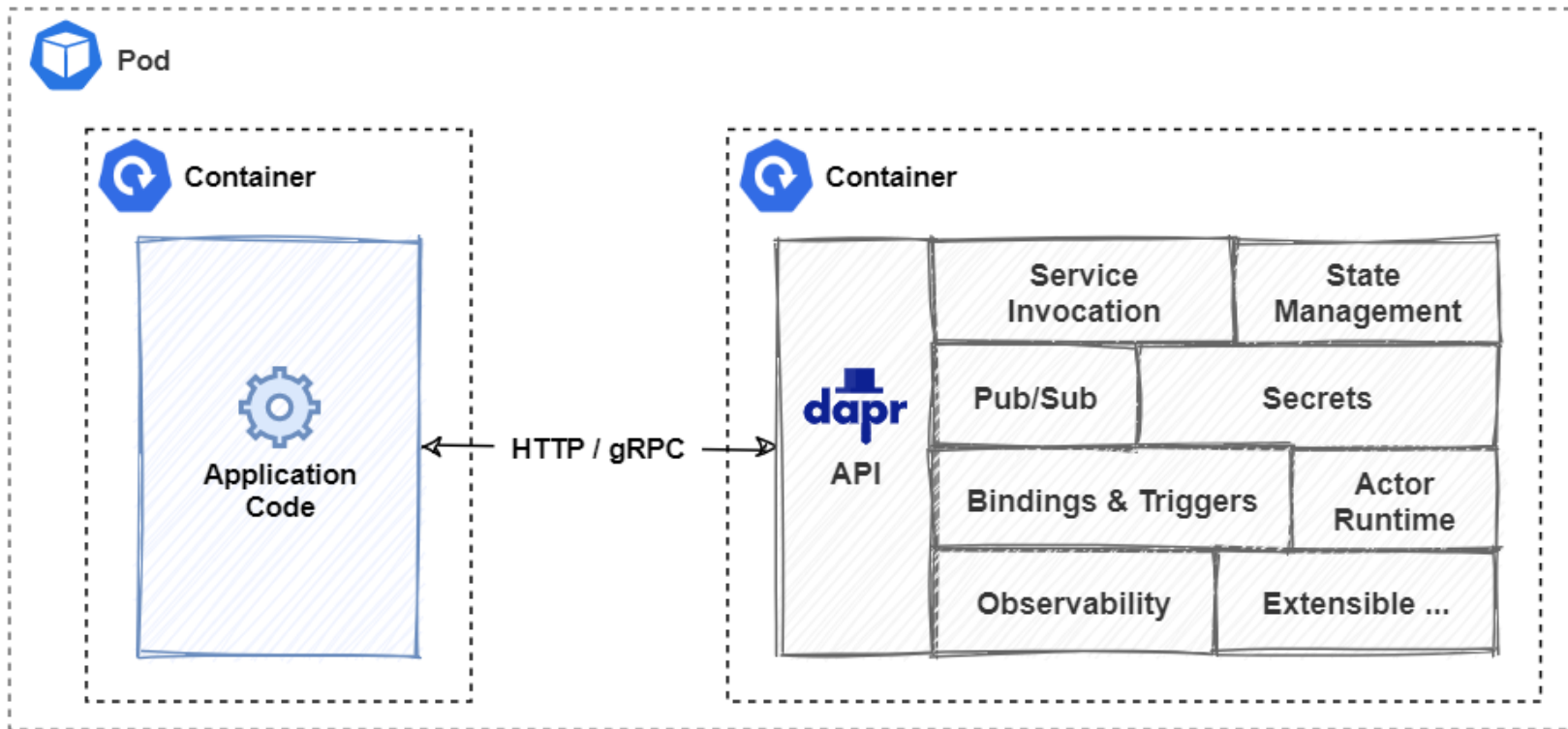


Speeds up application development by providing an integrated set of APIs for communication, state, and workflow.

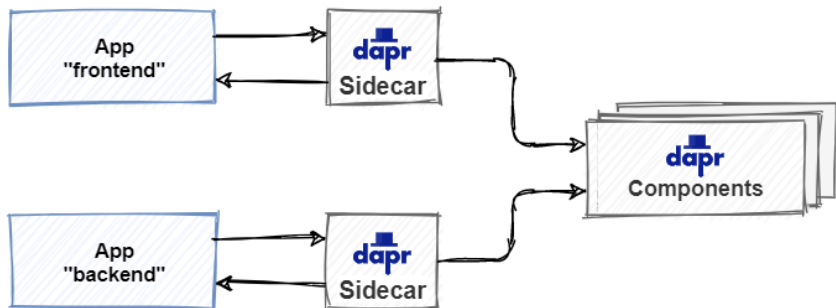
# Self-hosted Sidecar



# Kubernetes-hosted Sidecar



# Sidecar Performance Considerations



- Dapr operation:  
≥ 1 out-of-process network call
- Heavily optimized sidecar implementation
- gRPC with multiplexing, bidirectional full-duplex, streaming
- Overhead should be sub-millisecond

# SDK Overview

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

# Runtimes



## Self-Hosted

- Dedicated process next to your application process
- Executable or Docker Image

## Kubernetes

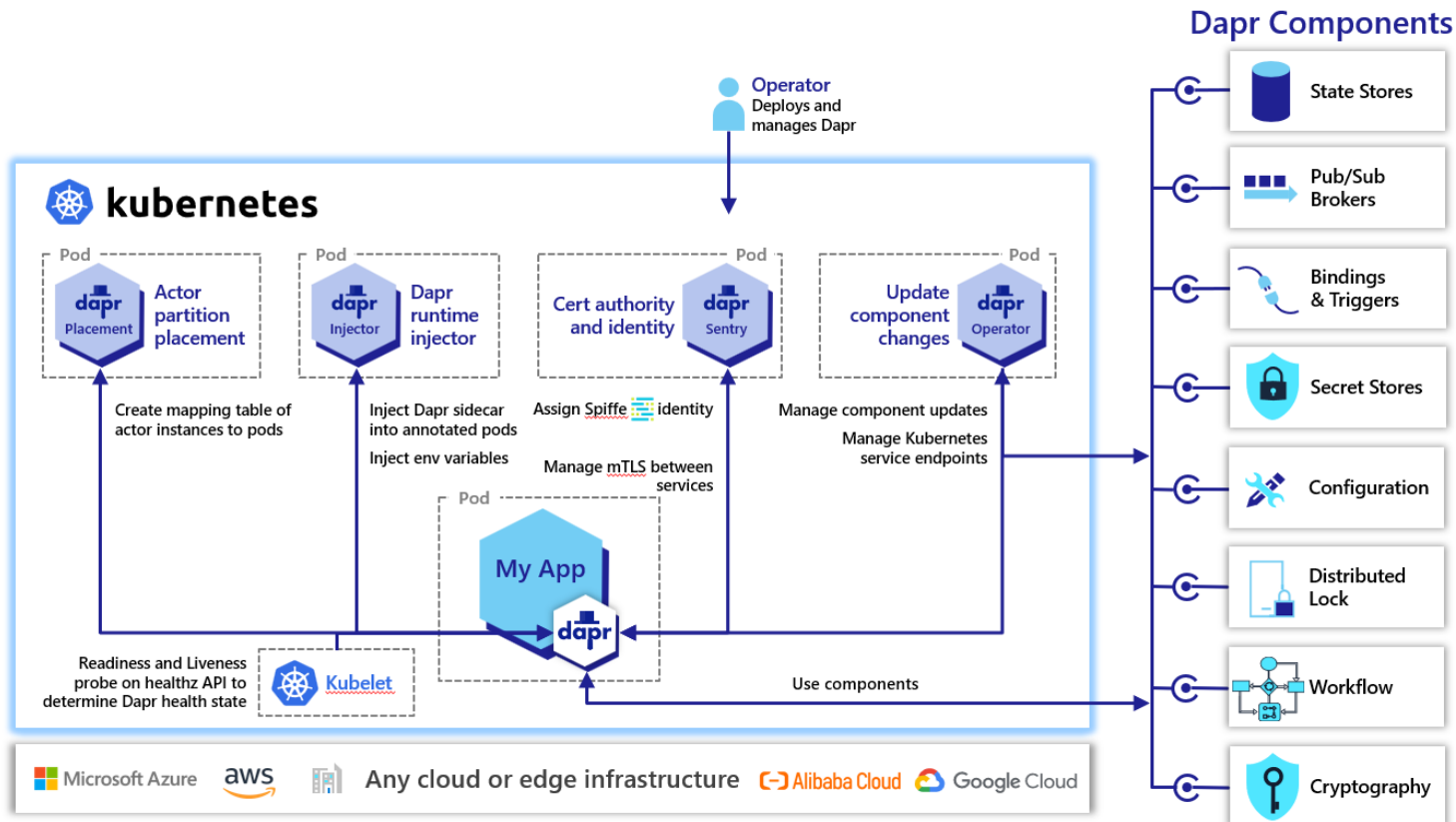
- Sidecar container in your pod, uses localhost interface
- Usually injected based on attributes

## Serverless

- Integrated in Azure
- i.e. Container Apps



# Dapr in Kubernetes



# Dapr from development to hosting

Use any language or runtime



HTTP/gRPC



Workflow



Publish /  
Subscribe



Service  
Invocation



State  
Management



Actors



Observability



Security



External  
Configuration



Secrets



Bindings



Cryptography



Distributed  
lock



Middleware



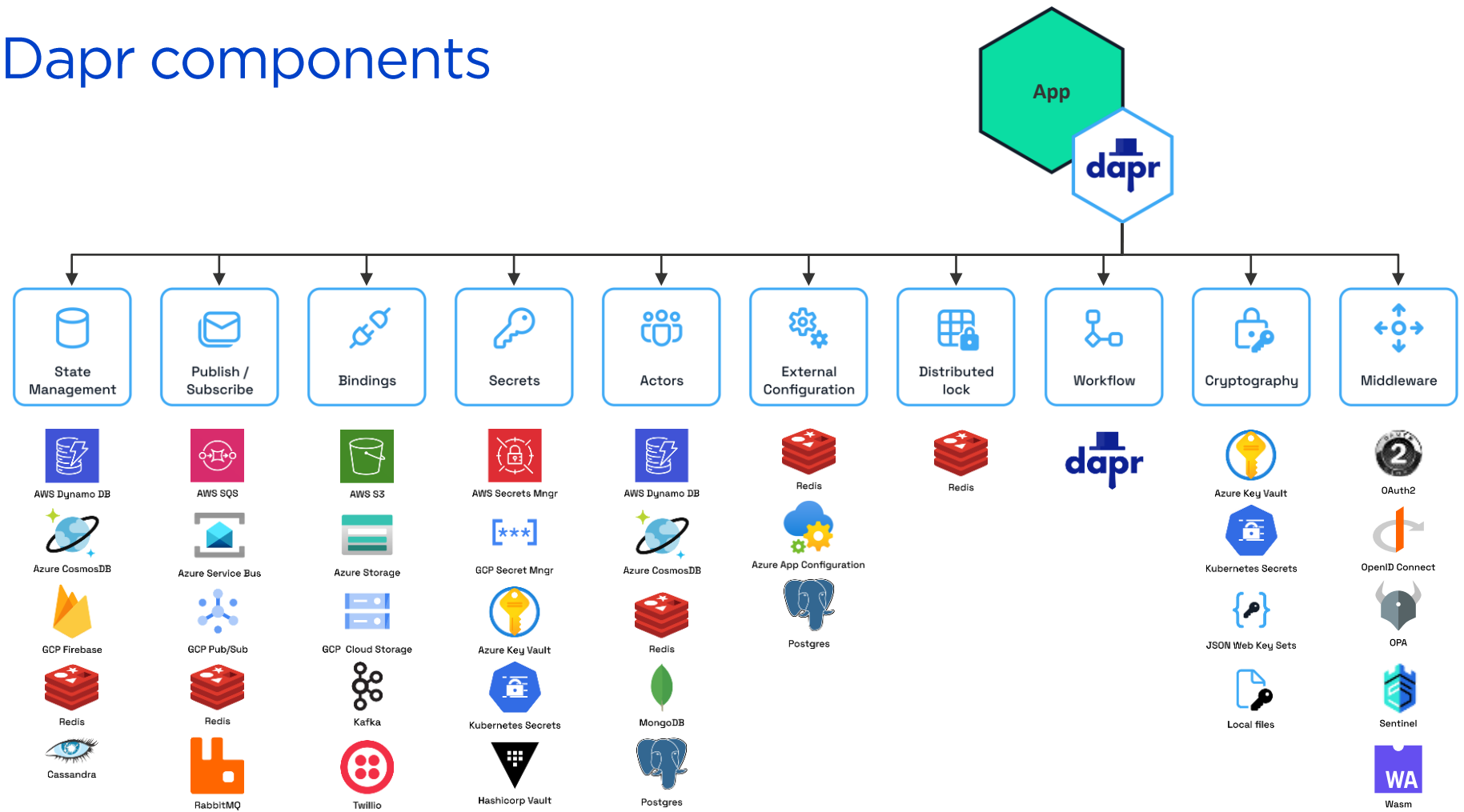
Resiliency

Host on any cloud or edge infrastructure



Virtual or  
physical machines

# Dapr components



A close-up, low-angle shot of several rowers in a boat, wearing blue and red uniforms, pulling their oars. The oars are black with yellow and blue handles. The background is a bright, slightly blurred body of water.

# Hands-on!

4tecture<sup>®</sup>  
empower your software solutions

A person is sitting at a desk in a dark room, illuminated by the blue light of two computer monitors. The person is seen in profile, looking at the left monitor which displays a web application with a sidebar and a main content area. The right monitor shows a code editor with syntax-highlighted code. On the desk, there is a red Coca-Cola can, a white mouse, and a pen holder with several pens. The overall atmosphere is focused and professional.

# DEMO



A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms and are pulling oars with yellow handles. The boat is on water, and the background is slightly blurred.

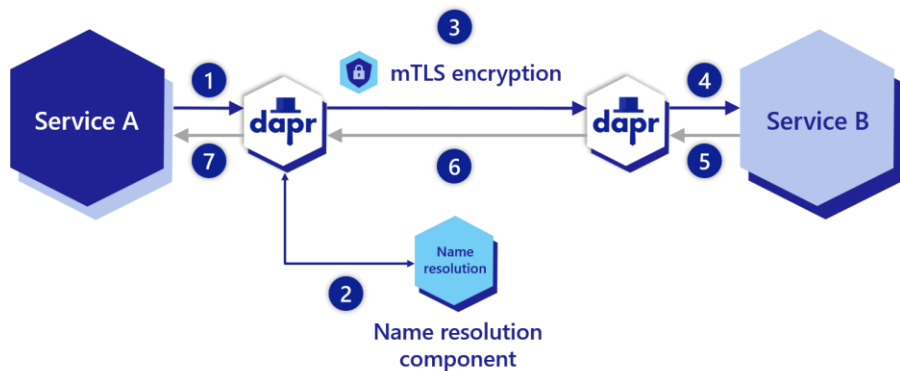
Dapr Building Blocks

# Service Invocation and Configuration

4tecture<sup>®</sup>  
empower your software solutions



# Service Invocation

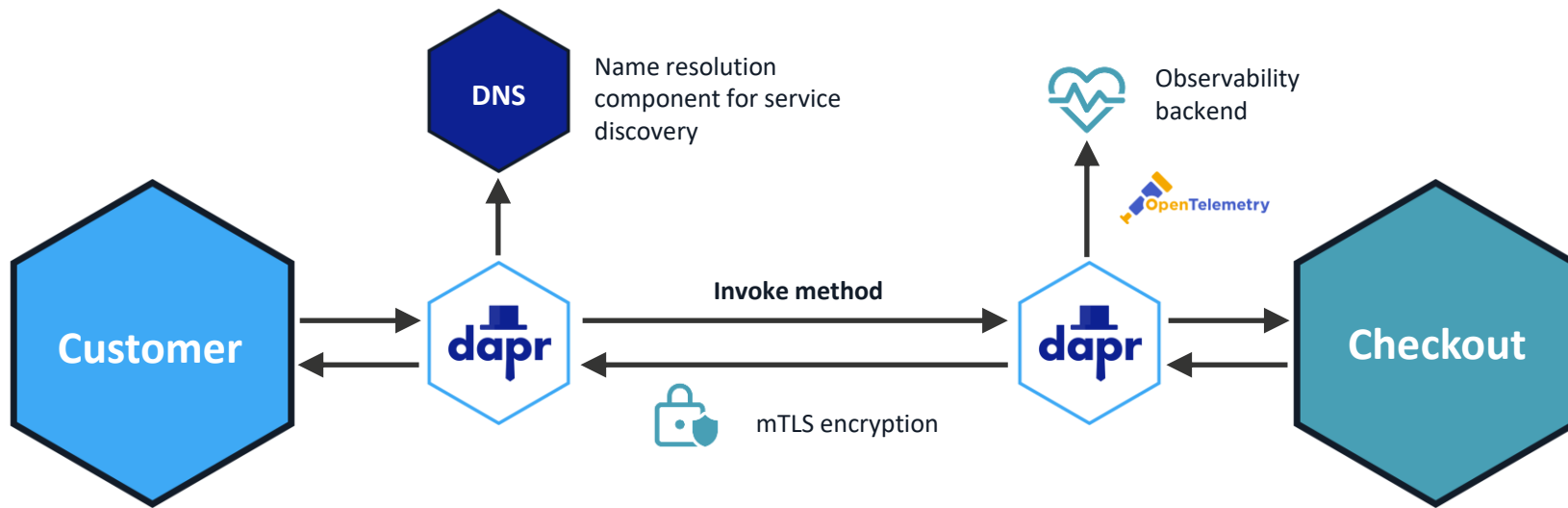


- HTTP and gRPC
- mTLS (with Dapr Sentry)
- Resiliency including retries
- Tracing and metrics with observability
- Access control (policies)
- Namespace scoping
- Load balancing (round robin with mDNS)
- Pluggable service discovery

# Service Invocation



Service  
Invocation



**POST**

`http://localhost:3500/v1.0/invoke/checkout/method/order`

**POST**

`http://localhost:5100/order`

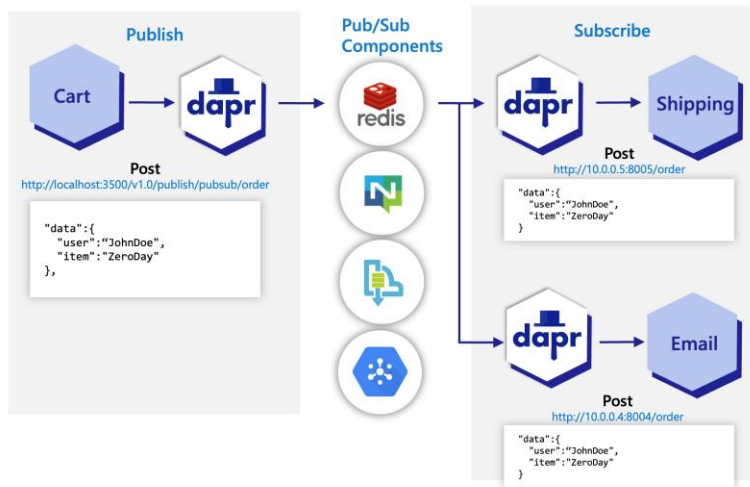
A background image showing a rowing team in blue and red uniforms, pulling oars with yellow handles. The text is overlaid on this image.

Dapr Building Blocks

# Publish & Subscribe

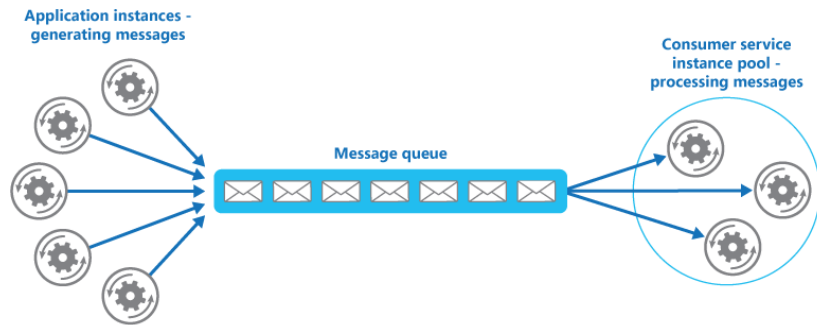
4tecture<sup>®</sup>  
empower your software solutions

# Publish & Subscribe



- Platform-agnostic API to send and receive messages
- At-least-once message delivery guarantee
- Integration with various message brokers
- CloudEvents 1.0 specification
- Message content type
- Content-based Routing
- Dead letter topics
- Namespace consumer groups
- Scoping topics

# Competing consumers pattern

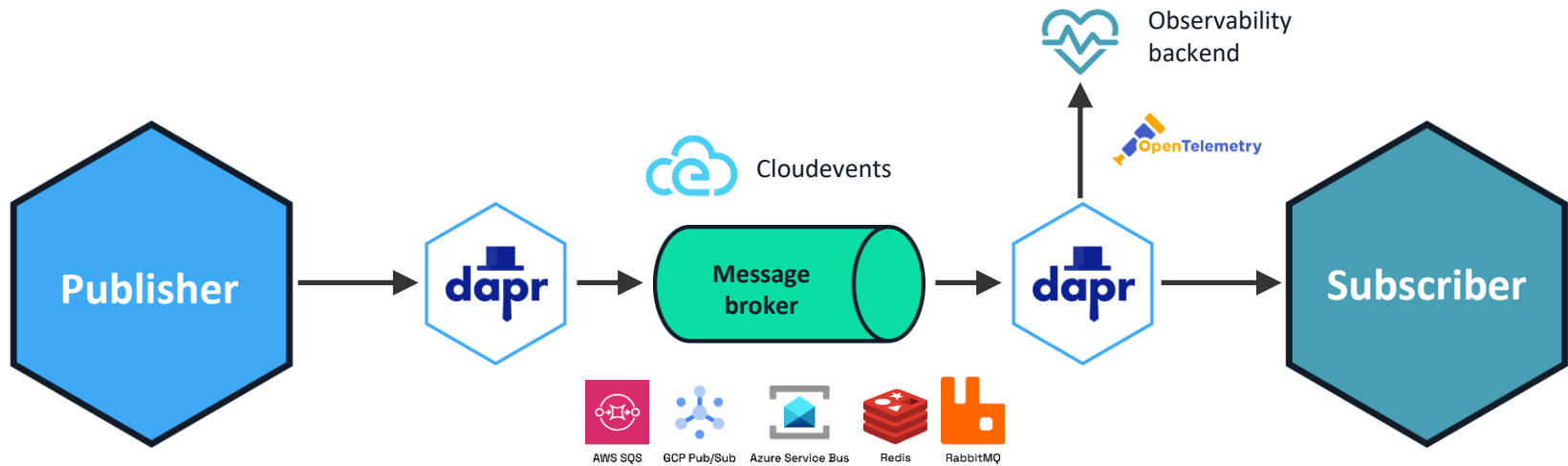


- Multiple application instances using a single consumer group
- Same app id = same consumer group
- Dapr delivers each message to only one instance of that application

# Publish / Subscribe



Publish /  
Subscribe



**POST**

<http://localhost:3500/v1.0/publish/mybroker/order-messages>

**POST**

<http://localhost:5100/orders>

# Pub/Sub Brokers

## Generic

Component	Status	Component version	Since runtime version
<a href="#">Apache Kafka</a>	Stable	v1	1.5
<a href="#">In-memory</a>	Stable	v1	1.7
<a href="#">JetStream</a>	Beta	v1	1.10
<a href="#">KubeMQ</a>	Beta	v1	1.10
<a href="#">MQTT3</a>	Stable	v1	1.7
<a href="#">Pulsar</a>	Stable	v1	1.10
<a href="#">RabbitMQ</a>	Stable	v1	1.7
<a href="#">Redis Streams</a>	Stable	v1	1.0
<a href="#">RocketMQ</a>	Alpha	v1	1.8
<a href="#">Solace-AMQP</a>	Beta	v1	1.10

## Amazon Web Services (AWS)

Component	Status	Component version	Since runtime version
<a href="#">AWS SNS/SQS</a>	Stable	v1	1.10

## Google Cloud Platform (GCP)

Component	Status	Component version	Since runtime version
<a href="#">GCP Pub/Sub</a>	Stable	v1	1.11

## Microsoft Azure

Component	Status	Component version	Since runtime version
<a href="#">Azure Event Hubs</a>	Stable	v1	1.8
<a href="#">Azure Service Bus Queues</a>	Beta	v1	1.10
<a href="#">Azure Service Bus Topics</a>	Stable	v1	1.0

A person is seen from the side, sitting at a desk in a dark room. They are looking at two computer monitors. The left monitor displays a web application with a sidebar and a main content area. The right monitor displays a code editor with syntax-highlighted code. A red Coca-Cola can is on the desk between the monitors. The person's hands are on a keyboard in front of them.

# DEMO

Sidcars, mTLS, PubSub, StateStore, Actors, Workflows



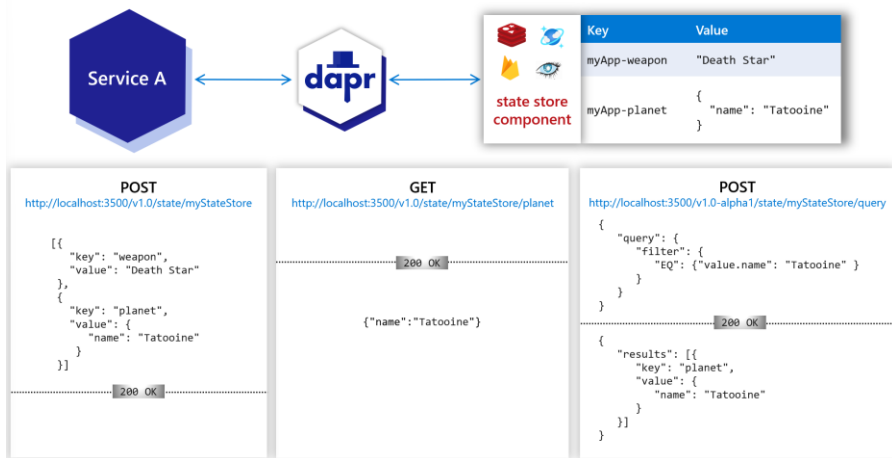
A background image showing a rowing team in blue and red uniforms, pulling oars with yellow handles. The image is slightly blurred to emphasize the text overlay.

Dapr Building Blocks

# State Management

4tecture<sup>®</sup>  
empower your software solutions

# State Management



- Configurable state store behavior (default eventually consistent, last-write-wins concurrency pattern)
- Optimistic concurrency with ETag
- Time to live (TTL)
- State encryption
- Querying state

# State Stores

## Generic

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
<a href="#">Aerospike</a>	✓	□	✓	□	□	□	Alpha	v1	1.0
<a href="#">Apache Cassandra</a>	✓	□	□	✓	□	□	Stable	v1	1.9
<a href="#">CockroachDB</a>	✓	✓	✓	✓	✓	✓	Stable	v1	1.10
<a href="#">Couchbase</a>	✓	□	✓	□	□	□	Alpha	v1	1.0
<a href="#">etcd</a>	✓	✓	✓	✓	✓	□	Beta	v2	1.12
<a href="#">Hashicorp Consul</a>	✓	□	□	□	□	□	Alpha	v1	1.0
<a href="#">Hazelcast</a>	✓	□	□	□	□	□	Alpha	v1	1.0
<a href="#">In-memory</a>	✓	✓	✓	✓	✓	□	Stable	v1	1.9
<a href="#">JetStream KV</a>	✓	□	□	□	□	□	Alpha	v1	1.7
<a href="#">Memcached</a>	✓	□	□	✓	□	□	Stable	v1	1.9
<a href="#">MongoDB</a>	✓	✓	✓	✓	✓	✓	Stable	v1	1.0
<a href="#">MySQL &amp; MariaDB</a>	✓	✓	✓	✓	✓	□	Stable	v1	1.10
<a href="#">Oracle Database</a>	✓	✓	✓	✓	✓	□	Beta	v1	1.7
<a href="#">PostgreSQL v1</a>	✓	✓	✓	✓	✓	✓	Stable	v1	1.0
<a href="#">PostgreSQL v2</a>	✓	✓	✓	✓	✓	□	Stable	v2	1.13
<a href="#">Redis</a>	✓	✓	✓	✓	✓	✓	Stable	v1	1.0
<a href="#">RethinkDB</a>	✓	□	□	□	□	□	Beta	v1	1.9
<a href="#">SQLite</a>	✓	✓	✓	✓	✓	□	Stable	v1	1.11
<a href="#">Zookeeper</a>	✓	□	✓	□	□	□	Alpha	v1	1.0

## Amazon Web Services (AWS)

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
<a href="#">AWS DynamoDB</a>	✓	✓	✓	✓	✓	□	Stable	v1	1.10

## Cloudflare

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
<a href="#">Cloudflare Workers KV</a>	✓	□	□	✓	□	□	Beta	v1	1.10

## Google Cloud Platform (GCP)

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
<a href="#">GCP Firestore</a>	✓	□	□	□	□	□	Stable	v1	1.11

## Microsoft Azure

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
<a href="#">Azure Blob Storage</a>	✓	□	✓	□	□	□	Stable	v2	1.13
<a href="#">Azure Cosmos DB</a>	✓	✓	✓	✓	✓	✓	Stable	v1	1.0
<a href="#">Azure Table Storage</a>	✓	□	✓	□	□	□	Stable	v1	1.9
<a href="#">Microsoft SQL Server</a>	✓	✓	✓	✓	✓	□	Stable	v1	1.5

## Oracle Cloud

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
<a href="#">Autonomous Database (ATP and ADW)</a>	✓	✓	✓	✓	✓	□	Alpha	v1	1.7
<a href="#">Object Storage</a>	✓	□	✓	✓	□	□	Alpha	v1	1.6

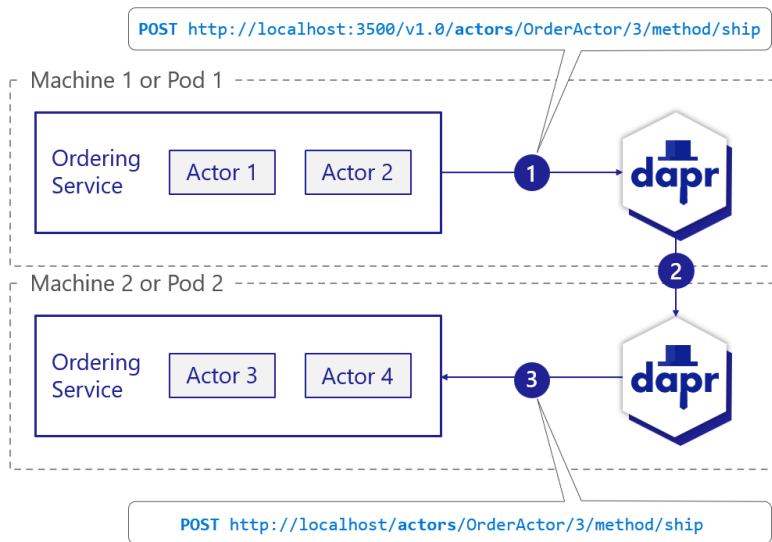
A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms. The focus is on the oars and the rowing mechanism, which includes yellow and blue components. The water is visible in the background.

Dapr Building Blocks

# Virtual Actors

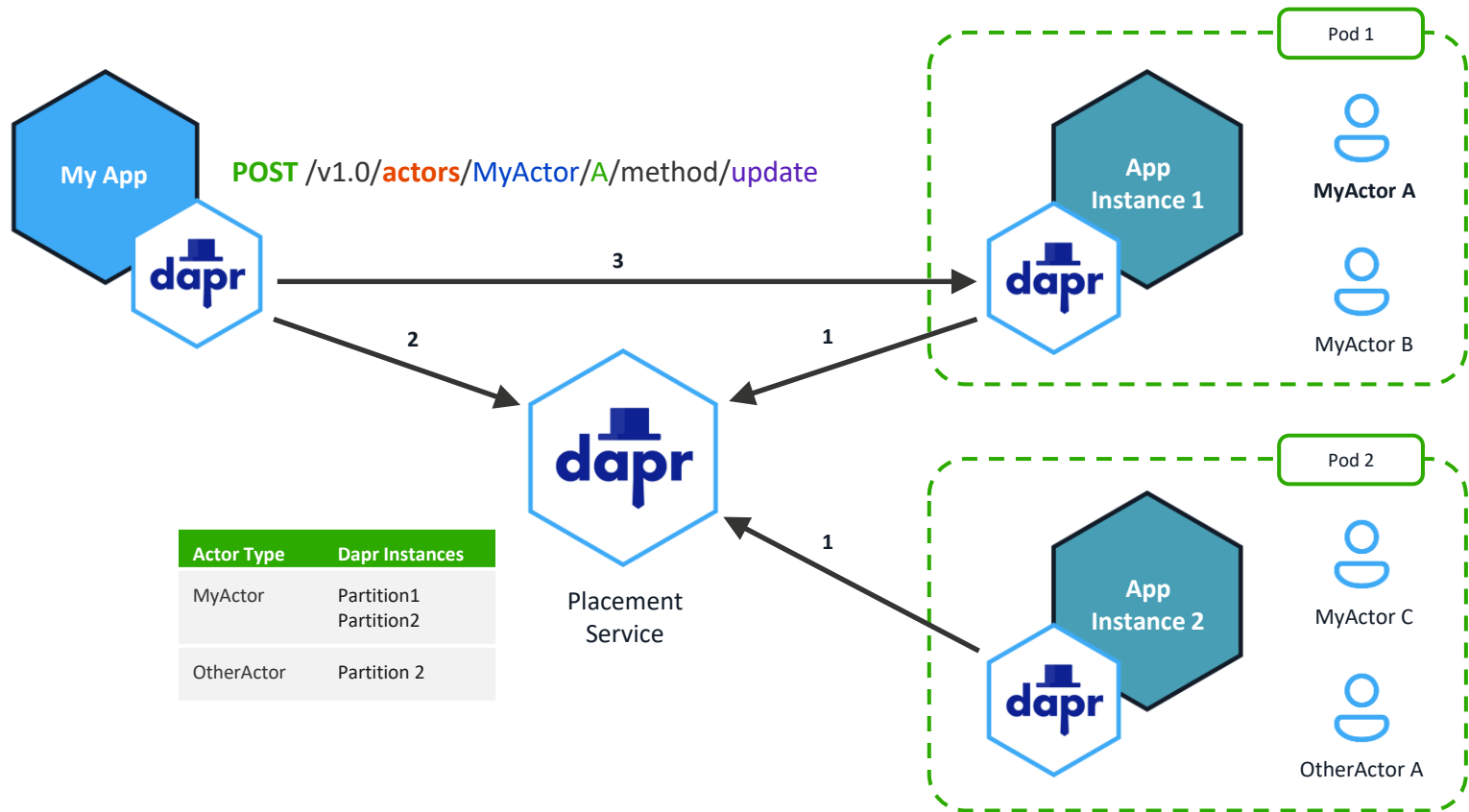
4tecture<sup>®</sup>  
empower your software solutions

# Actors



- Virtual Actor pattern
- Stateful, long running objects with identity
- Encapsulate state and behavior within a distributed system
- Actor state store
- Actor timers and reminders

# Actor placement





A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms. The focus is on the oars and the rowing mechanism, which includes yellow and blue components. The water is visible in the foreground.

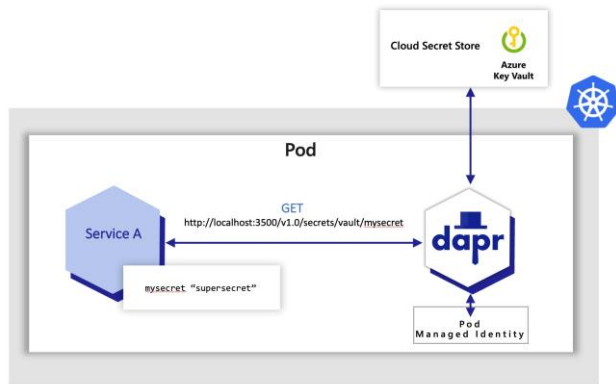
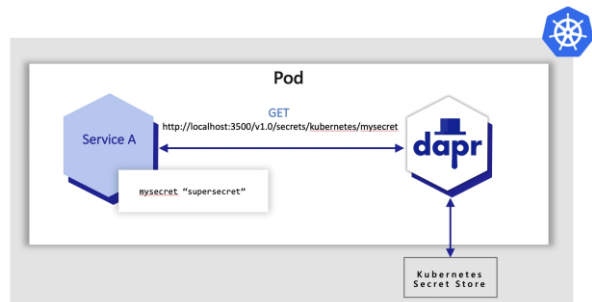
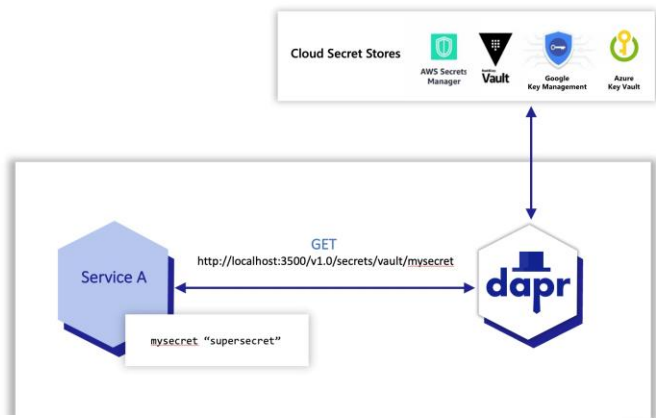
Dapr Building Blocks

# Secret Management

4tecture<sup>®</sup>  
empower your software solutions

# Secret Management

- Access secret stores through generic Dapr API
- Secret scoping (limit access)



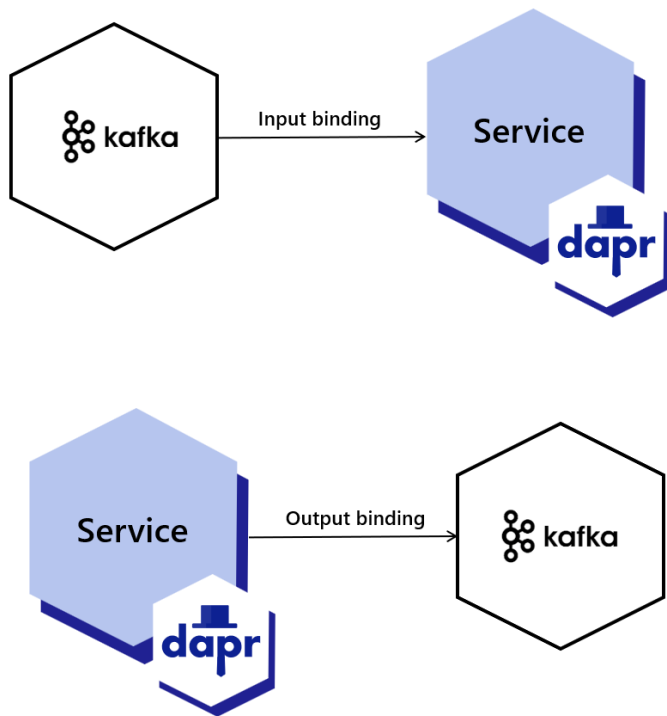


A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms and are captured in a synchronized rowing motion. The focus is on the oars and the rowers' hands, with the water visible in the background.

Dapr Building Blocks

# Input/Output Bindings

# Input/Output Binding



- Trigger your app with events coming from external systems
- Handle retries and failure recovery
- Portable app with environment-specific bindings

# Bindings

## Generic

Component	Input Binding	Output Binding	Status	Component version	Since runtime version
Apple Push Notifications (APN)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
commercetools GraphQL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.8
Cron (Scheduler)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Stable	v1	1.10
GraphQL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
HTTP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.0
Huawei OBS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.8
InfluxDB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Beta	v1	1.7
Kafka	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.8
Kitex	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.11
KubeMQ	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Beta	v1	1.10
Kubernetes Events	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Alpha	v1	1.0
Local Storage	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.9
MQTT3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Beta	v1	1.7
MySQL & MariaDB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
PostgreSQL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.9
Postmark	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
RabbitMQ	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.9
Redis	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.9
RethinkDB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Beta	v1	1.9
SendGrid	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
SMTP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
Twilio	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
Wasm	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.11

## Microsoft Azure

Component	Input Binding	Output Binding	Status	Component version	Since runtime version
Azure Blob Storage	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.0
Azure Cosmos DB (Gremlin API)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.5
Azure CosmosDB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.7
Azure Event Grid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Beta	v1	1.7
Azure Event Hubs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.8
Azure OpenAI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.11
Azure Service Bus Queues	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.7
Azure SignalR	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alpha	v1	1.0
Azure Storage Queues	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stable	v1	1.0

A background image showing a rowing team in blue uniforms and red accents, pulling oars with yellow handles. The oars are connected to a mechanical system with blue and yellow components. The scene is on a body of water.

Dapr Building Blocks

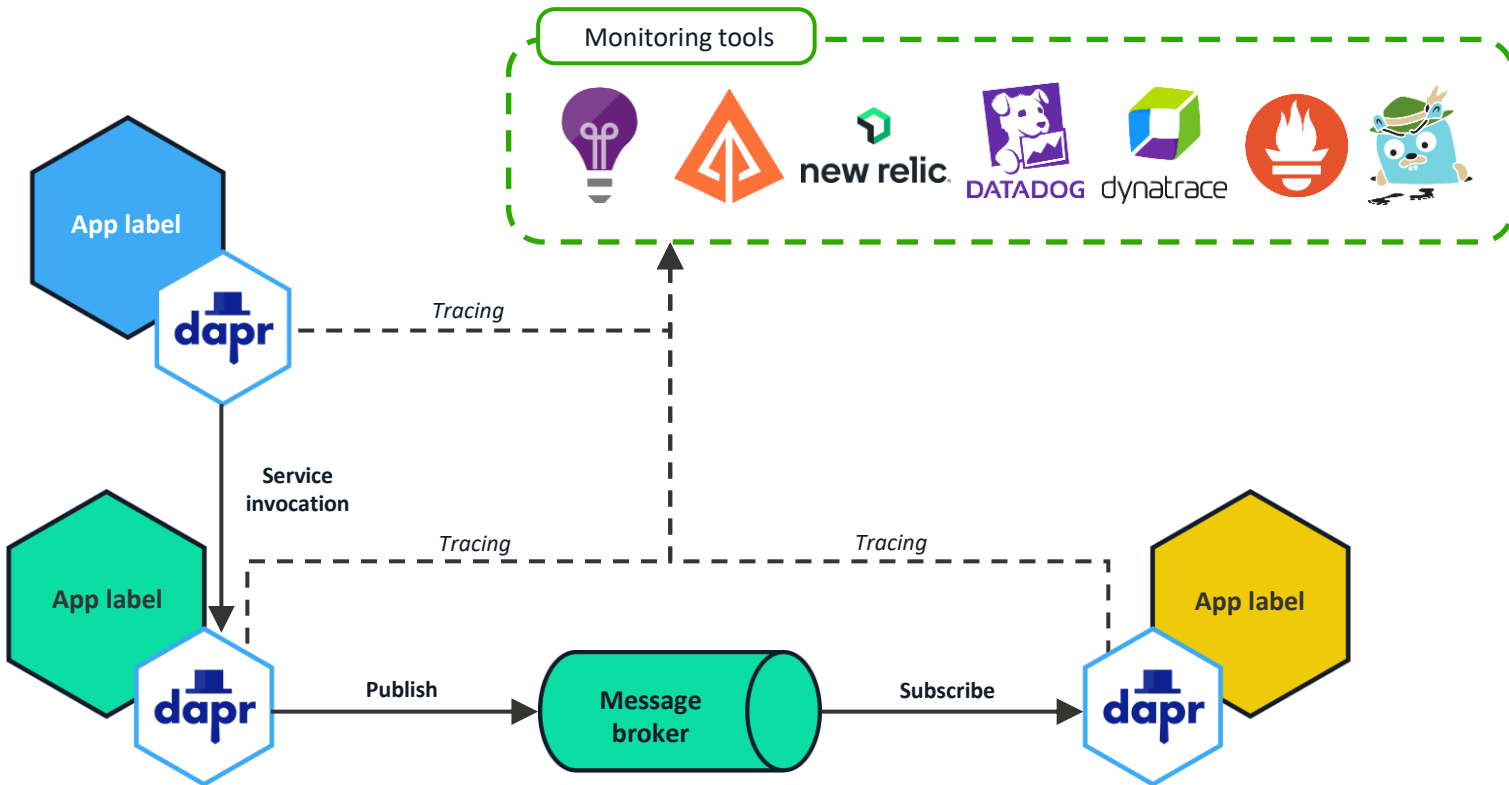
# Observability

4tecture<sup>®</sup>  
empower your software solutions

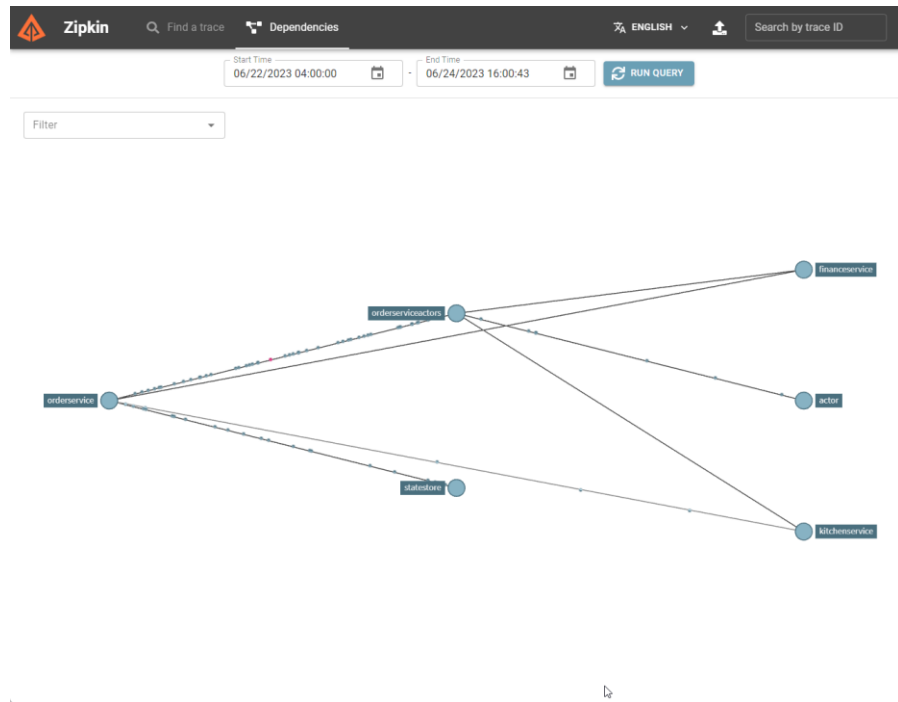
# Distributed tracing



Observability



# Observability



- Distributed tracing
- Open Telemetry (OTEL) and Zipkin protocols
- Used with service invocation and pub/sub APIs
- Sidecar health
- App health checks
  - Unsubscribing Pub/Sub
  - Stop input bindings
  - Short-circuiting all service-invocation requests



A close-up, low-angle shot of rowers in a boat, showing their legs and hands gripping the oars. The rowers are wearing blue long-sleeved shirts and red and white shorts. The oars are black with yellow and blue handles. The background is a bright, slightly blurred body of water.

Dapr Building Blocks

# Resiliency

4tecture<sup>®</sup>  
empower your software solutions

# Resiliency

Resiliency patterns can be applied across Dapr APIs:

- Retries
- Timeouts
- Circuit breakers

Declarative and decoupled from application code.

Available across all component types, service invocation, and actors.

```
apiVersion: dapr.io/v1alpha1
```

```
kind: Resiliency
```

```
metadata:
```

```
  name: myresiliency
```

```
scopes:
```

```
  - order-processor
```

```
spec:
```

```
  policies:
```

```
    retries:
```

```
      retryForever:
```

```
        policy: constant
```

```
        duration: 5s
```

```
        maxRetries: -1
```

```
  circuitBreakers:
```

```
    simpleCB:
```

```
      maxRequests: 1
```

```
      timeout: 5s
```

```
      trip: consecutiveFailures >= 5
```

```
targets:
```

```
  components:
```

```
    statestore:
```

```
      outbound:
```

```
        retry: retryForever
```

```
        circuitBreaker: simpleCB
```



A close-up, low-angle shot of rowers in a boat. The rowers are wearing blue long-sleeved shirts and red and white life vests. They are holding yellow handles of oars. The boat is on water, and the background is a bright, overcast sky. The image is used as a background for the text overlay.

.NET Microservices with Dapr

# Conclusion

# Pros / Cons

## Advantages

- Develop faster
- Best practices
- Portability
- Focus on your logic

## Disadvantages

- Additional hop / network overhead
- Common API – less features

# Conclusion

- Suitable for most teams and applications
- Base your development on proven best practices
- Ideal, if portability is key (different environments / clouds, local, etc.)

A background image showing a rowing team in a boat. The rowers are wearing blue and red uniforms and are pulling oars with yellow handles. The boat is on water, and the background is slightly blurred.

.NET Microservices with Dapr

Q & A

4tecture<sup>®</sup>  
empower your software solutions

# Thank you for your attention!

If you have any questions do not hesitate to contact us:

4tecture GmbH  
Industriestrasse 25  
CH-8604 Volketswil  
[www.4tecture.ch](http://www.4tecture.ch)

Marc Müller  
Principal Consultant  
[www.powerofdevops.com](http://www.powerofdevops.com)





A background image showing several hands of different skin tones reaching towards the center, where they are assembling a small cluster of four interlocking wooden puzzle pieces. The pieces are colored light brown, white, red, and green. The overall scene is softly blurred, focusing attention on the hands and the puzzle pieces.

4tecture<sup>©</sup>  
empower your software solutions