.NET Microservices with Dapr

Marc Müller
Principal Consultant

marc.mueller@4tecture.ch
@muellermarc
www.4tecture.ch

MVP Microsoft Most Valuable Professional

4tecture
empower your software solutions

About me:

Marc Müller
Principal Consultant
@muellermarc

Microsoft Most Valuable Professional

4tecture©
empower your software solutions

Our Products:

Multi-Tenant OpenID Connect Identity Provider

ProAuth

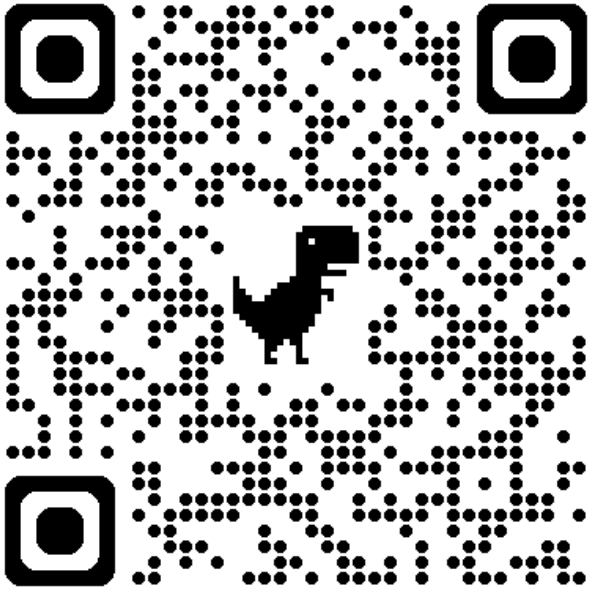
www.proauth.net

Enterprise Application Framework for .NET

ReaFx

www.reafx.net

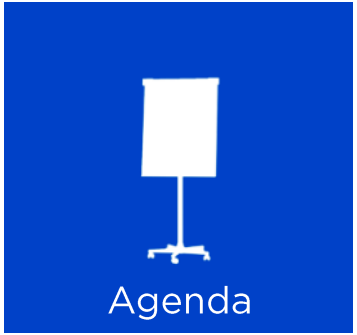
Slide Download



https://www.4tecture.ch/events/dwx24dapr

Agenda

- Intro
- Dapr Basics
  - BB: Service Invocation
  - BB: Pub/Sub
  - BB: State Management
  - BB: Virtual Actors
  - BB: Secret Management
  - BB: Observability
  - BB: Resiliency
- Conclusion



Dapr

Intro

4tecture
empower your software solutions

Cloud computing characteristics by NIST

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

Source: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

CNCF cloud native definition

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

Source: https://github.com/cncf/toc/blob/main/DEFINITION.md

Cloud native applications

Cloud-native applications should be designed and implemented to use and take advantage of the cloud computing characteristics.

Cloud native applications characteristics

- Immutable Packaging & Execution
- Decoupled Configurations and Secrets
- Statelessness and Statefulness
- Modular Applications / Microservices
- Polyglot Paradigm (multi lang / tech)
- Centralized Logging and Monitoring
- DevOps and SRE processes
- Automation
- API-Centric

Commonly used with cloud native apps

- Microservices
- Cloud platforms
- Containers
- Kubernetes
- Immutable infrastructures
- Declarative APIs
- Continuous Delivery technologies

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

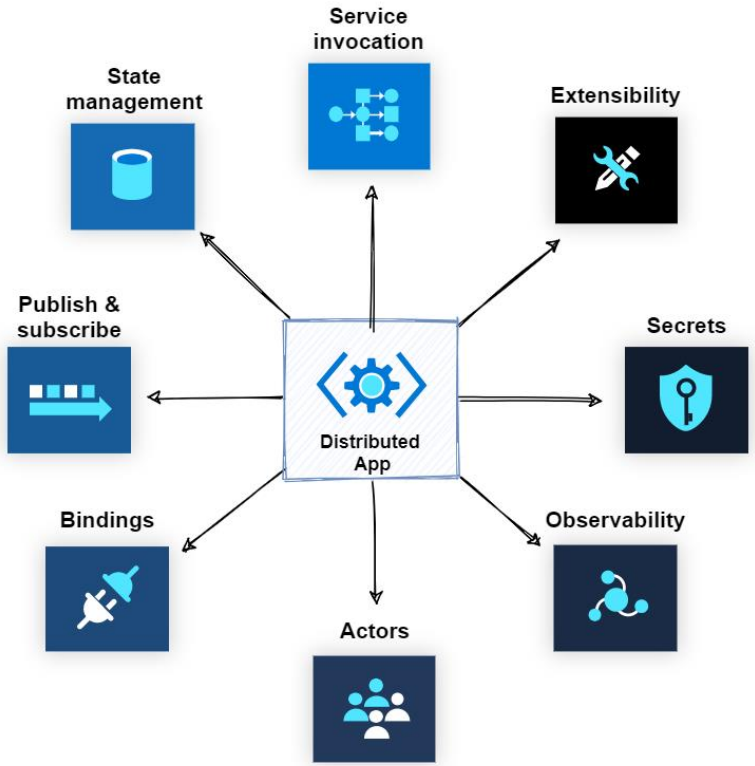
Portable, event-driven runtime

Build connected distributed applications faster

APIs for solving distributed application challenges

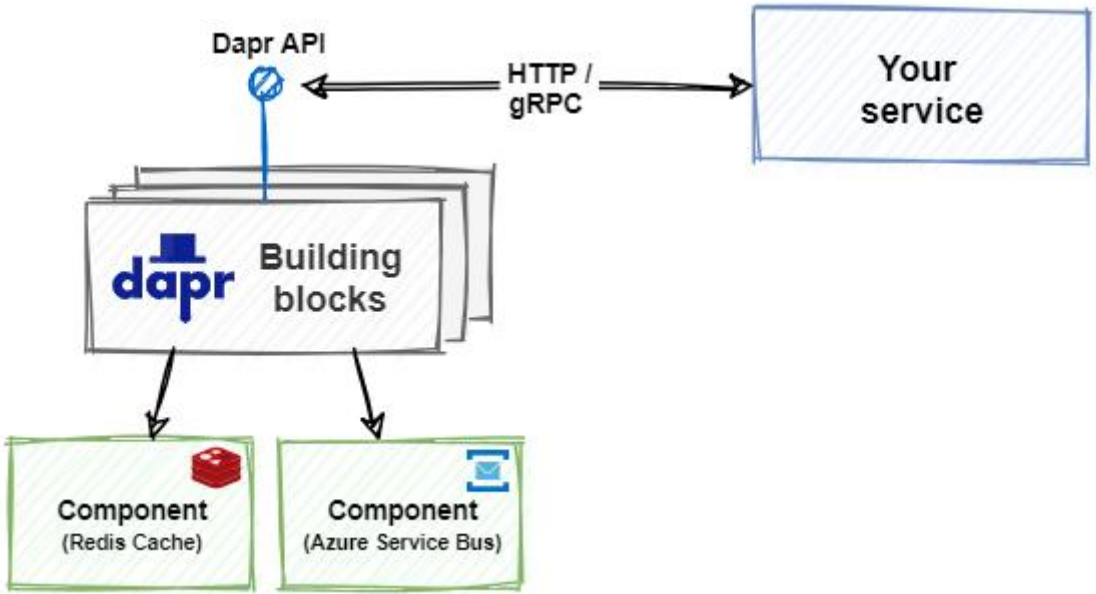
Cloud and Edge

Dapr Building Blocks



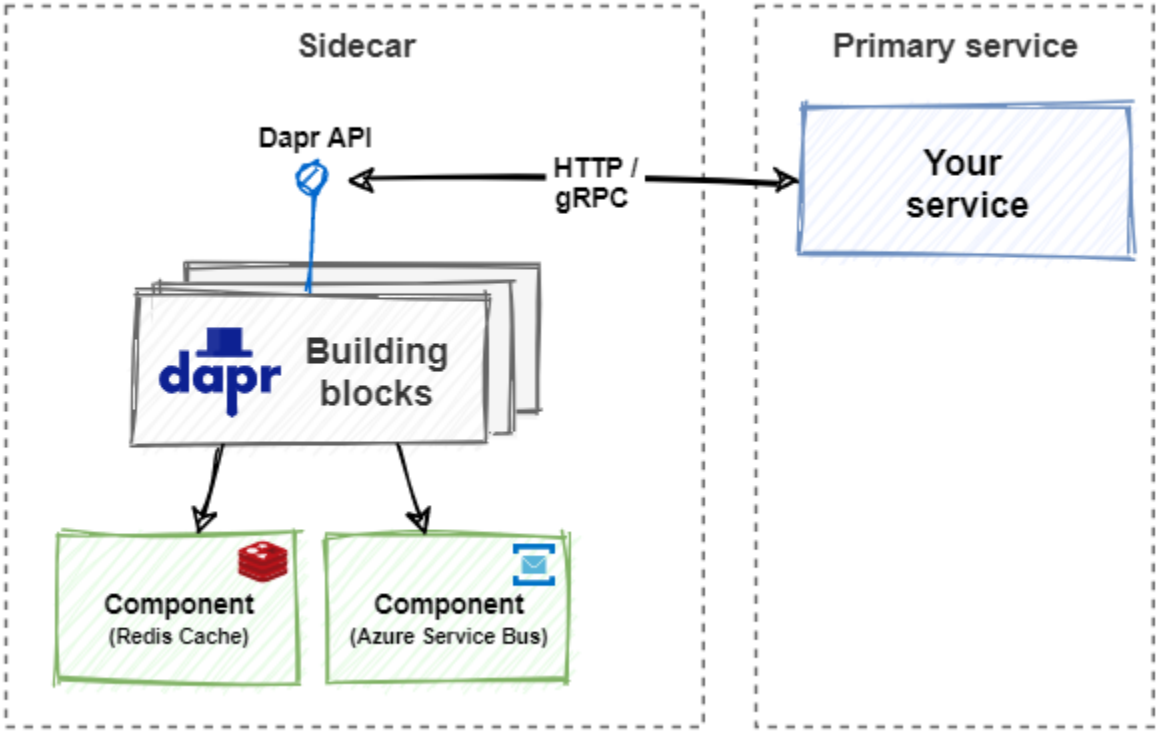
https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

Dapr Building Blocks Abstraction



https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

Dapr Sidecar Architecture



https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

Dapr Basics

4tecture
empower your software solutions

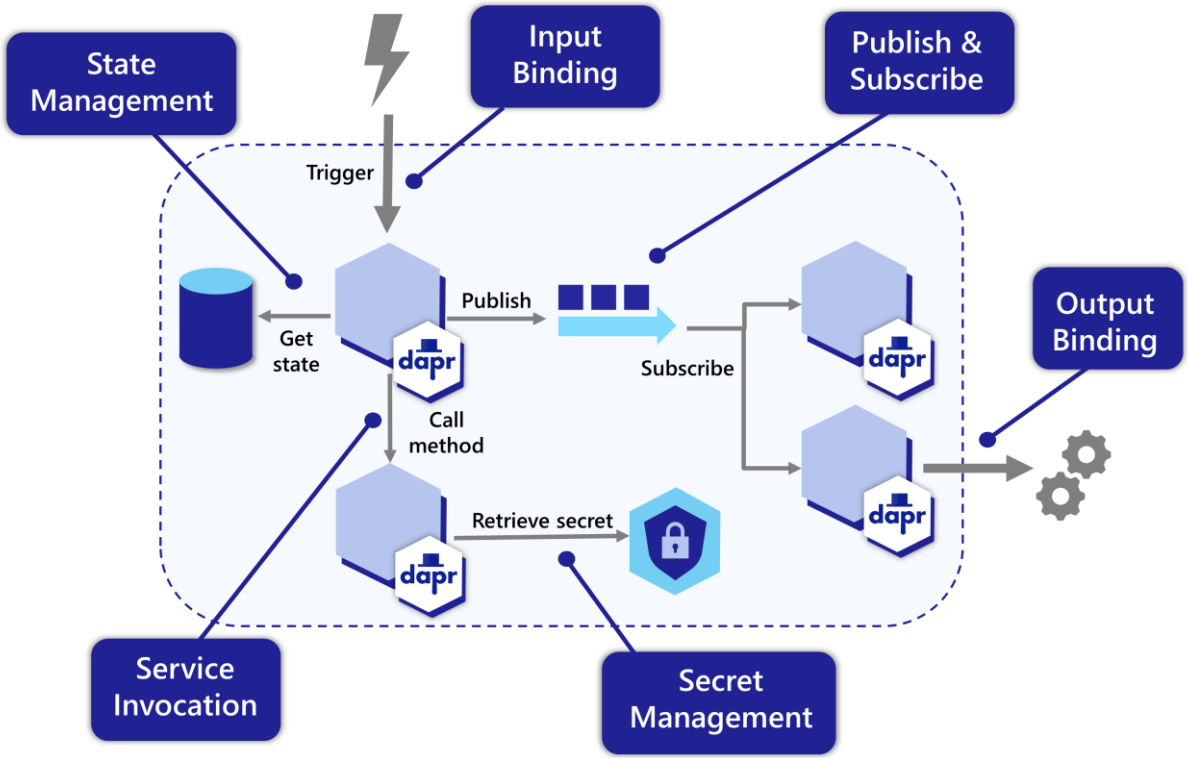
Dapr

- dapr.io
- Open source
- Originated at Microsoft
- Cloud Native Computing Foundation (CNCF) – incubating 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



https://dapr.io/

Single API - Multiple Components

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

daprClient.SaveStateAsync("statestore", "AMS", weatherForecast);
```

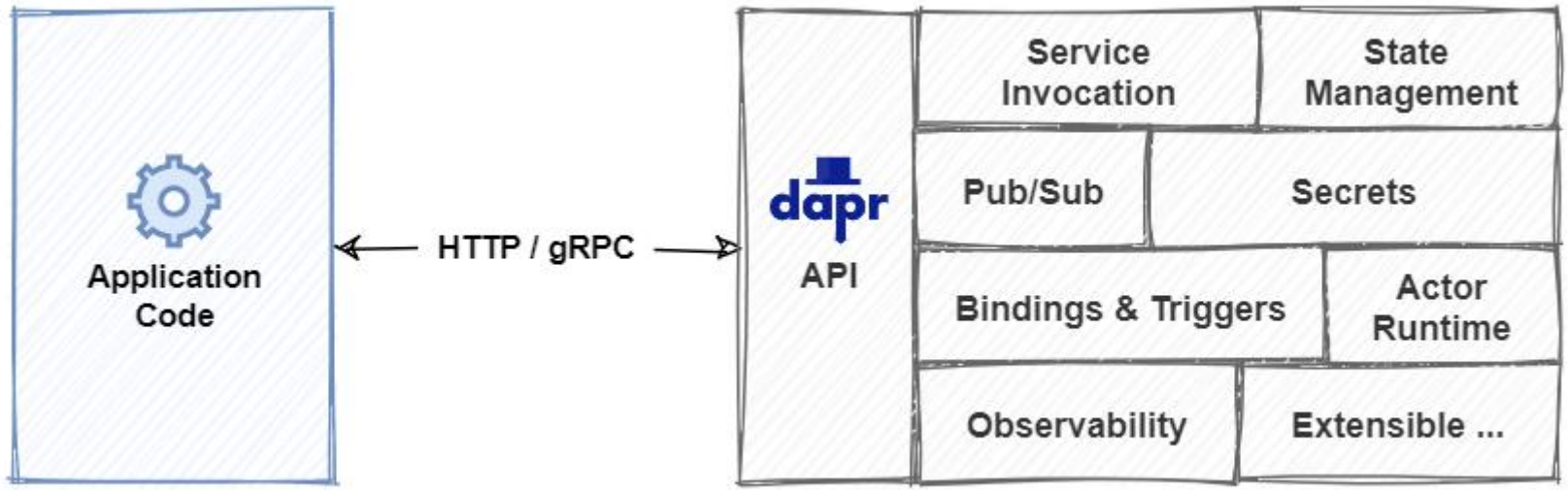
- AWS DynamoDB
- Aerospike
- Azure Blob Storage
- Azure CosmosDB
- Azure Table Storage
- Cassandra
- Cloud Firestore (Datastore mode)
- CloudState
- Couchbase
- Etcd
- HashiCorp Consul
- Hazelcast
- Memcached
- MongoDB
- PostgreSQL
- Redis
- RethinkDB
- SQL Server
- Zookeeper

https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

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

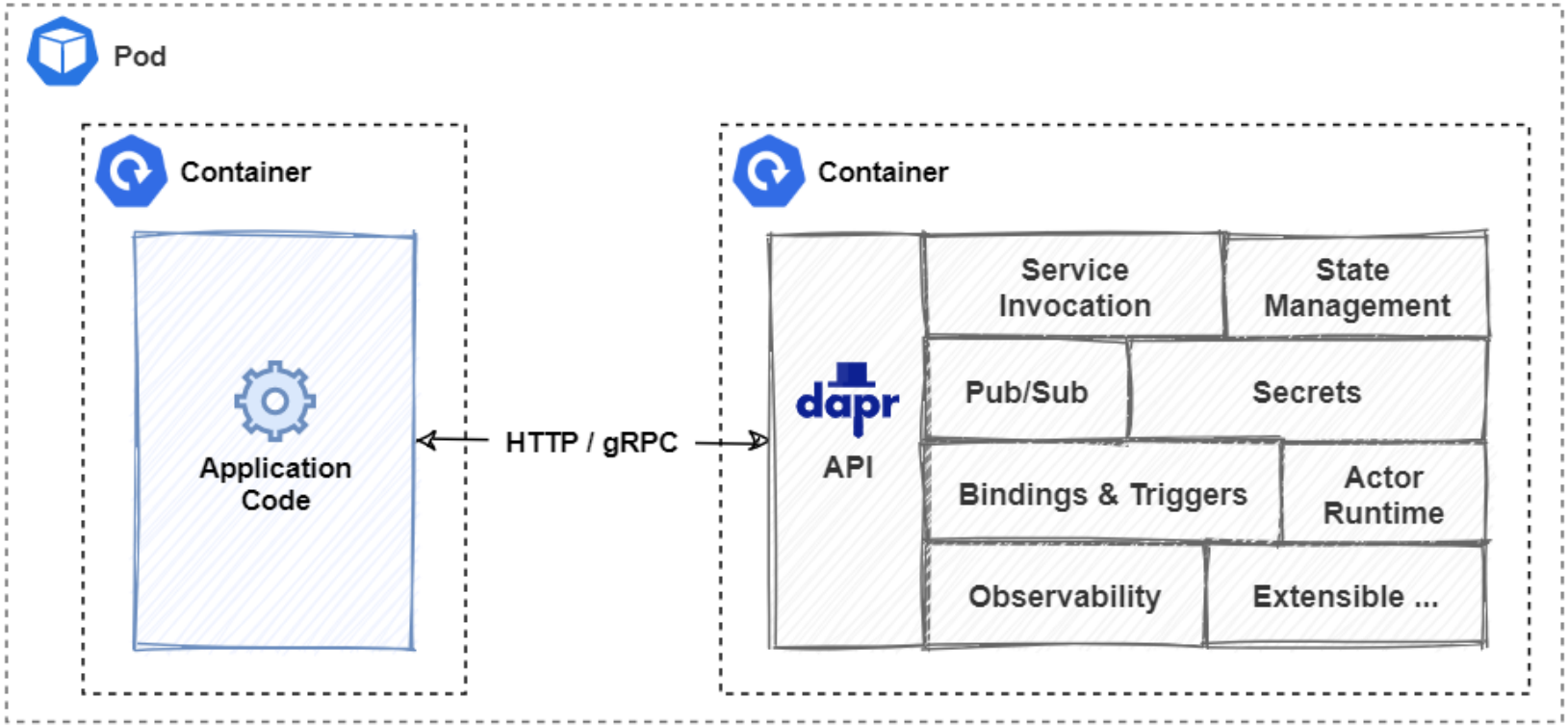
Speeds up application development by providing an integrated set of APIs for communication, state, and workflow.

Self-hosted Sidecar



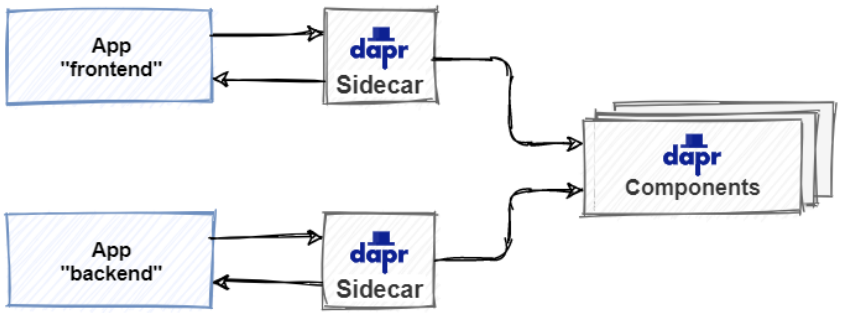
https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

Kubernetes-hosted Sidecar



https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

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

https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

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	✓		✓	

https://docs.dapr.io/developing-applications/sdks/

Runtimes

Self-Hosted

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

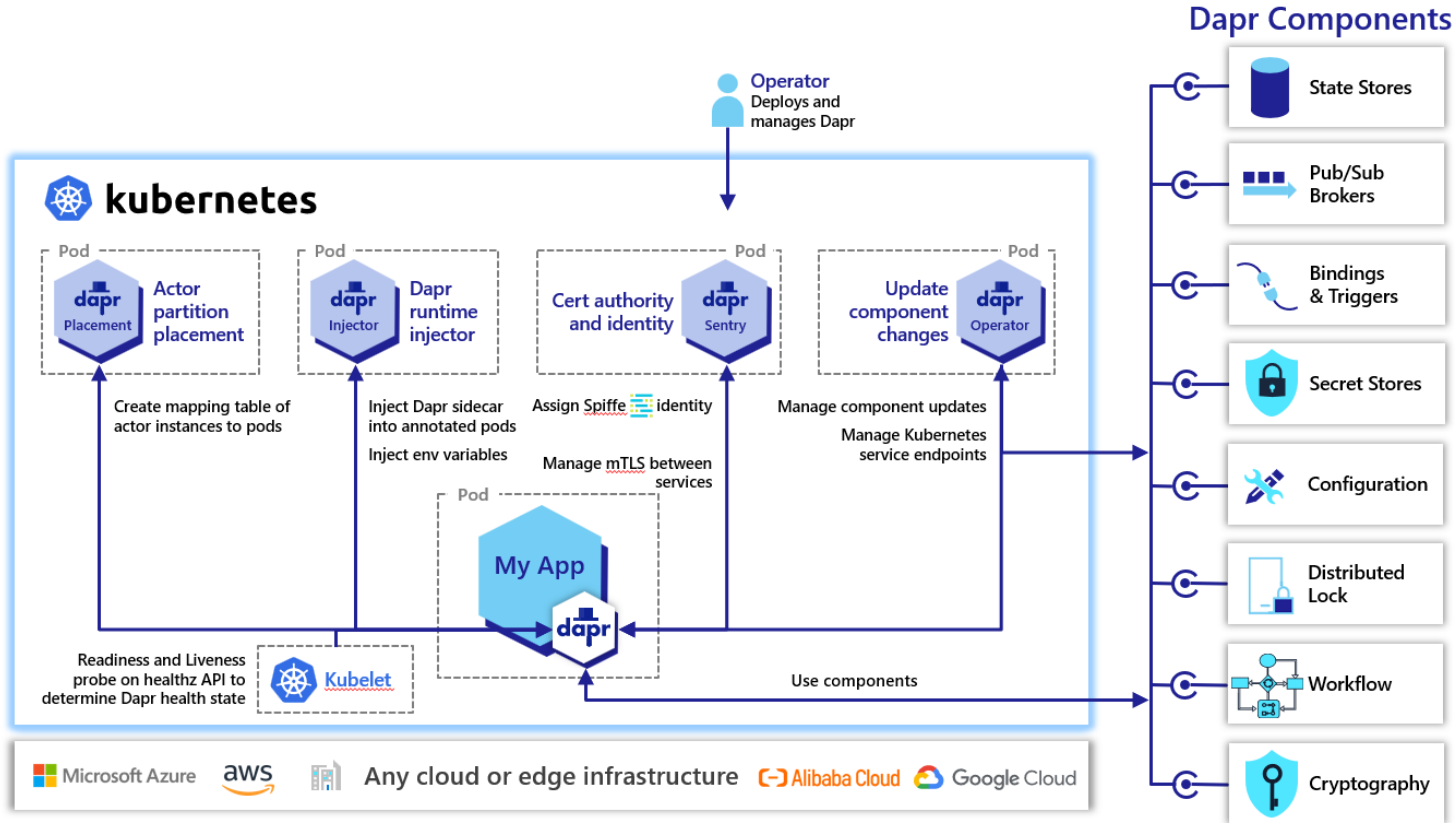
Kubernetes

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

Serverless

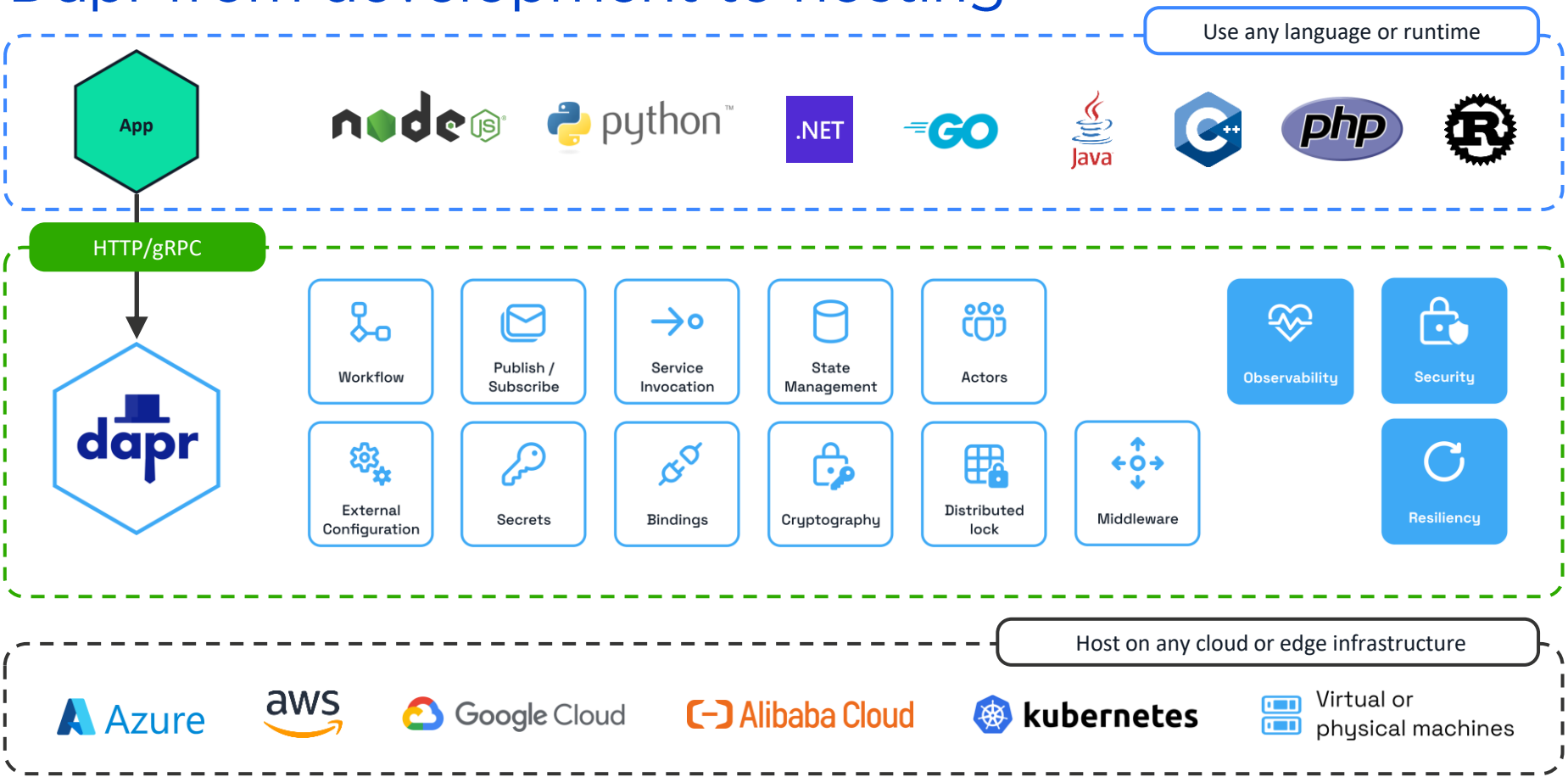
- Integrated in Azure
- i.e. Container Apps

Dapr in Kubernetes

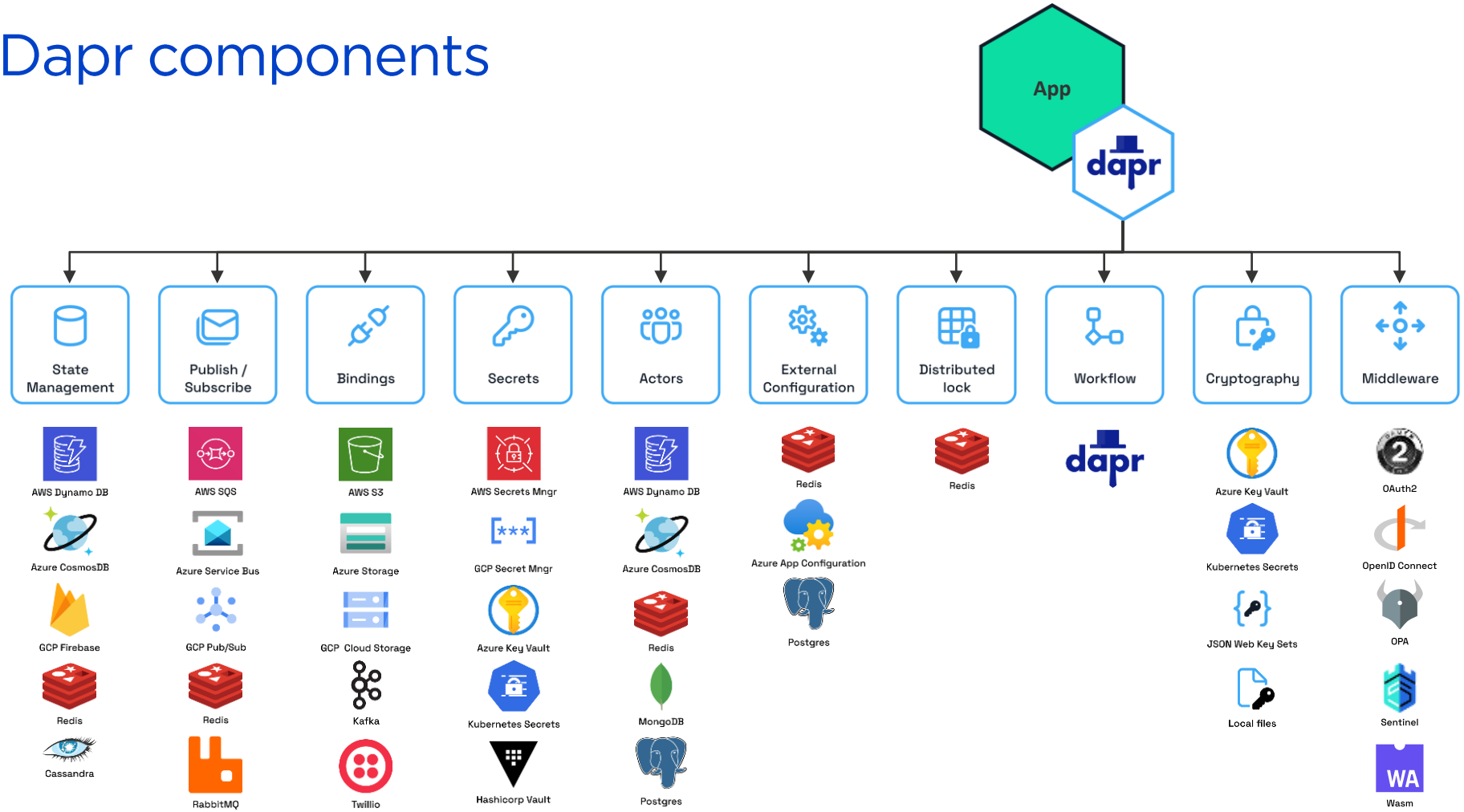


https://docs.dapr.io/operations/hosting/kubernetes/kubernetes-overview/

Dapr from development to hosting

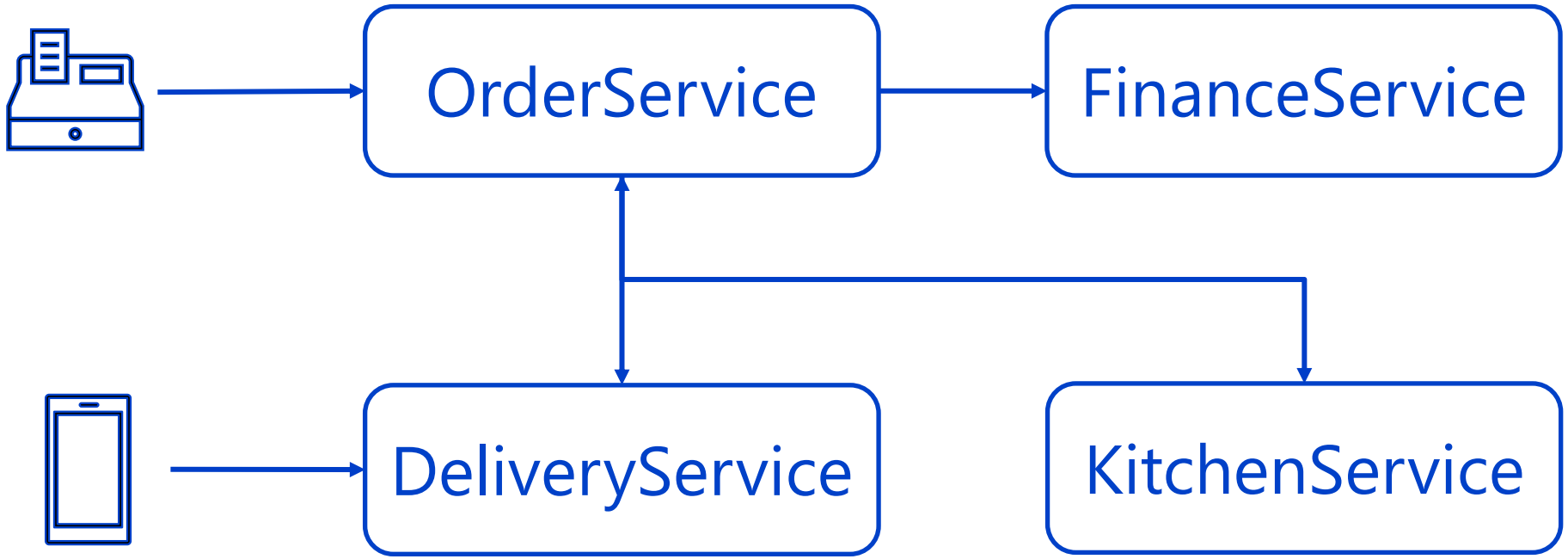


Dapr components




Demo first
4tecture
empower your software solutions

Fast Food Company



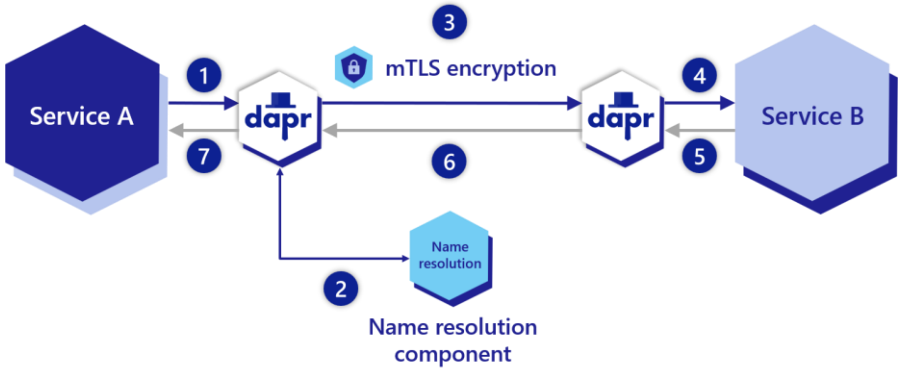

DEMO

Dapr Building Blocks

Service Invocation and Configuration

4tecture
empower your software solutions

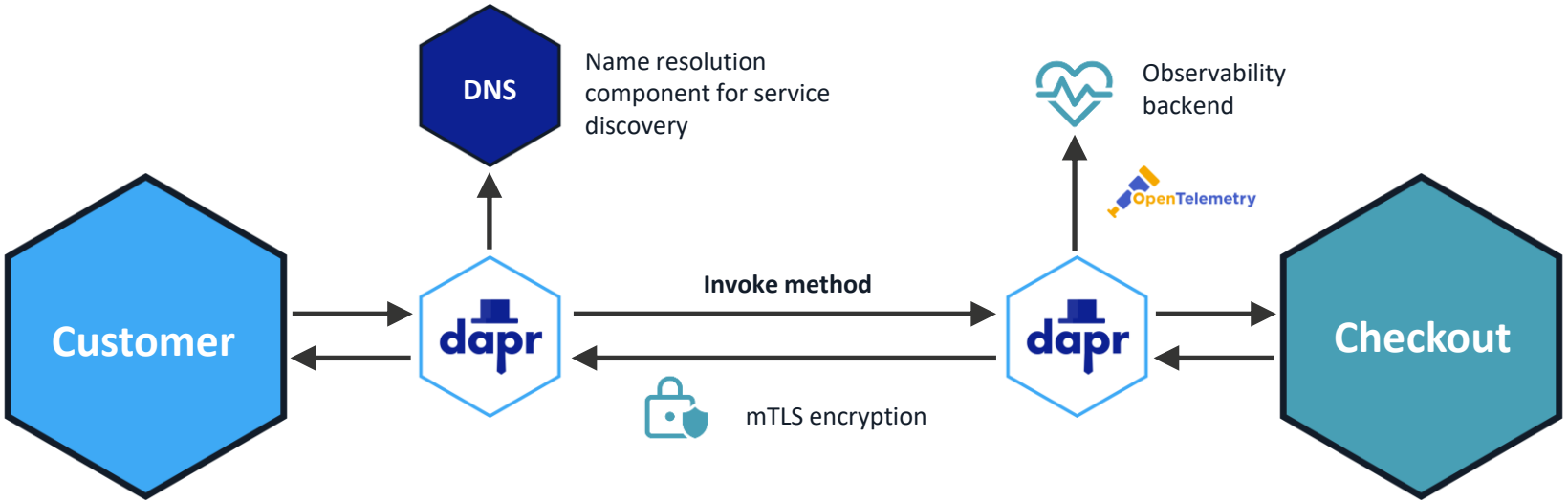
Service Invocation



- 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

https://docs.dapr.io/developing-applications/building-blocks/service-invocation/service-invocation-overview/

Service Invocation



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

POST
http://localhost:5100/order

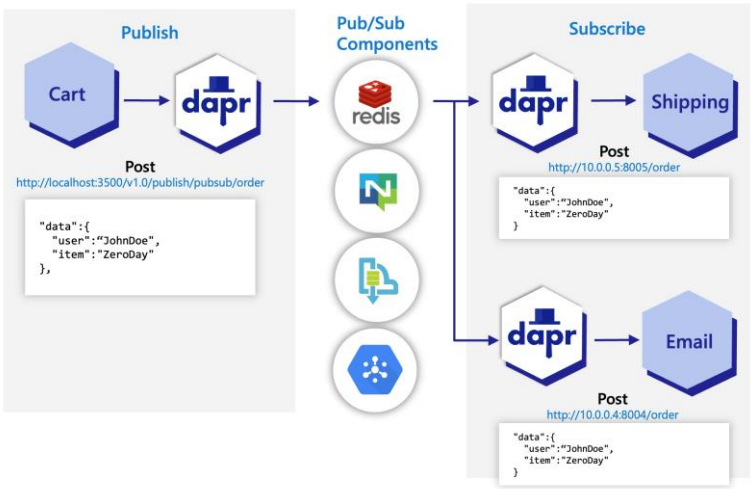

DEMO

Dapr Building Blocks

Publish & Subscribe

4tecture
empower your software solutions

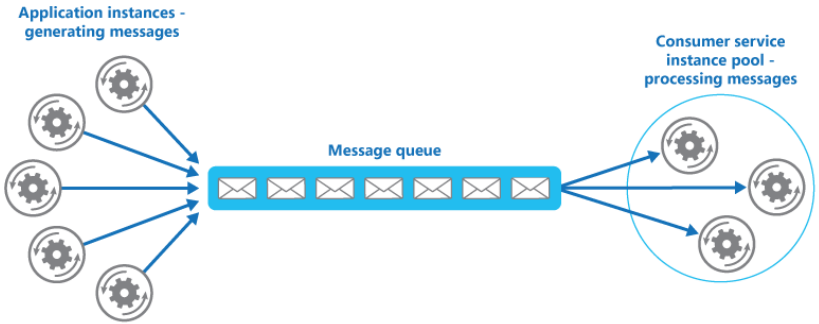
Publish & Subscribe



- 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

https://docs.dapr.io/developing-applications/building-blocks/pubsub/pubsub-overview/

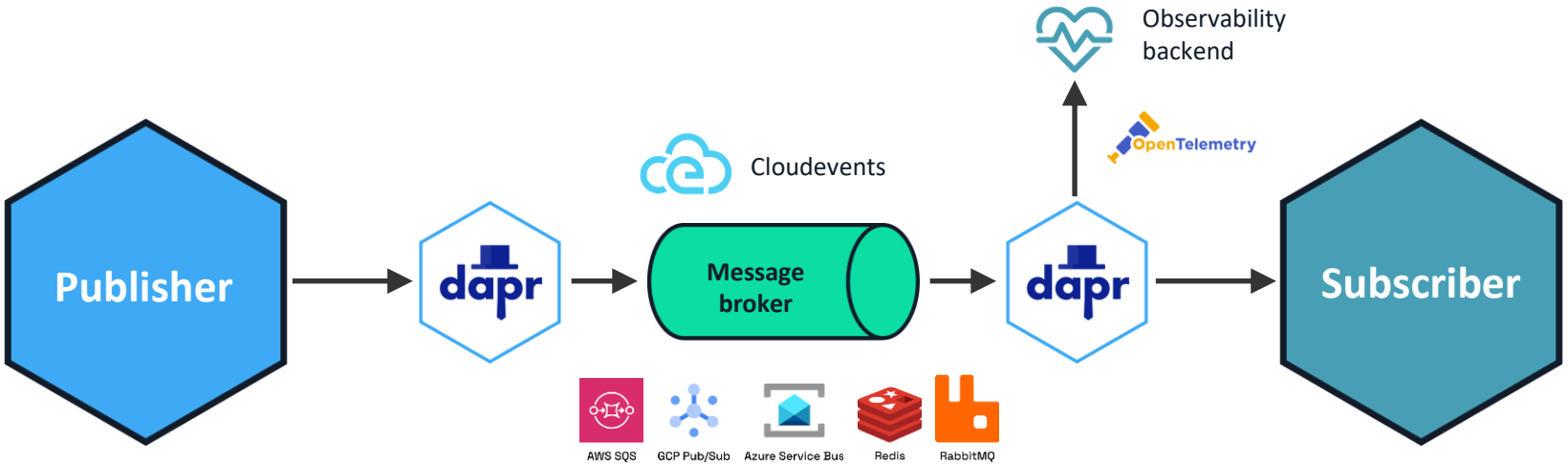
Competing consumers pattern



- 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

https://docs.dapr.io/developing-applications/building-blocks/pubsub/pubsub-overview/

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
Apache Kafka	Stable	v1	1.5
In-memory	Stable	v1	1.7
JetStream	Beta	v1	1.10
KubeMQ	Beta	v1	1.10
MQTT3	Stable	v1	1.7
Pulsar	Stable	v1	1.10
RabbitMQ	Stable	v1	1.7
Redis Streams	Stable	v1	1.0
RocketMQ	Alpha	v1	1.8
Solace-AMQP	Beta	v1	1.10

Amazon Web Services (AWS)

Component	Status	Component version	Since runtime version
AWS SNS/SQS	Stable	v1	1.10

Google Cloud Platform (GCP)

Component	Status	Component version	Since runtime version
GCP Pub/Sub	Stable	v1	1.11

Microsoft Azure

Component	Status	Component version	Since runtime version
Azure Event Hubs	Stable	v1	1.8
Azure Service Bus Queues	Beta	v1	1.10
Azure Service Bus Topics	Stable	v1	1.0

https://docs.dapr.io/reference/components-reference/supported-pubsub/

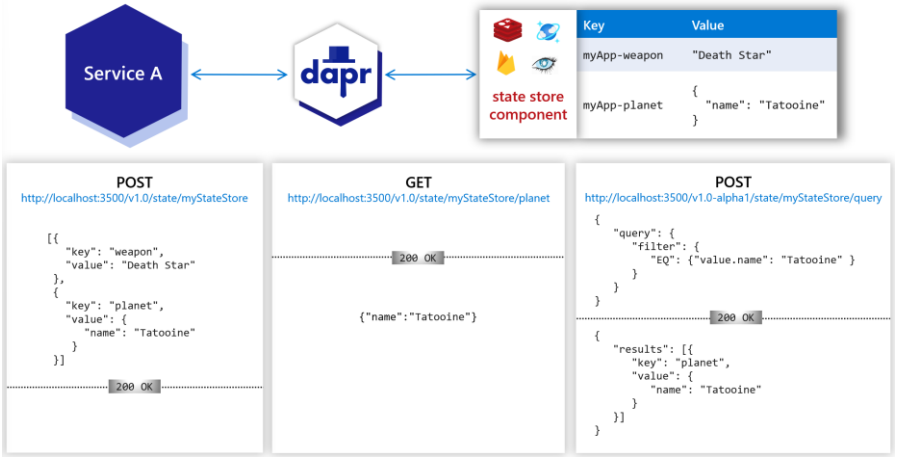

DEMO

Dapr Building Blocks

State Management

4tecture
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

https://docs.dapr.io/developing-applications/building-blocks/state-management/state-management-overview/

State Stores

Generic

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

Amazon Web Services (AWS)

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
AWS DynamoDB	✓	✓	✓	✓	✓		Stable	v1	1.10

Cloudflare

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
Cloudflare Workers KV	✓			✓			Beta	v1	1.10

Google Cloud Platform (GCP)

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
GCP Firestore	✓						Stable	v1	1.11

Microsoft Azure

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
Azure Blob Storage	✓		✓				Stable	v2	1.13
Azure Cosmos DB	✓	✓	✓	✓	✓	✓	Stable	v1	1.0
Azure Table Storage	✓		✓				Stable	v1	1.9
Microsoft SQL Server	✓	✓	✓	✓	✓		Stable	v1	1.5

Oracle Cloud

Component	CRUD	Transactional	ETag	TTL	Actors	Query	Status	Component version	Since runtime version
Autonomous Database (ATP and ADW)	✓	✓	✓	✓	✓		Alpha	v1	1.7
Object Storage	✓		✓	✓			Alpha	v1	1.6

https://docs.dapr.io/reference/components-reference/supported-state-stores/

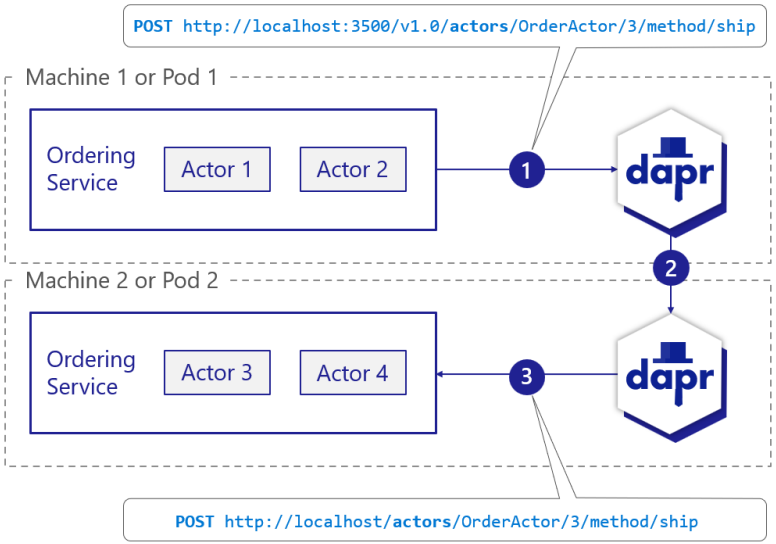

DEMO

Dapr Building Blocks

Virtual Actors

4tecture
empower your software solutions

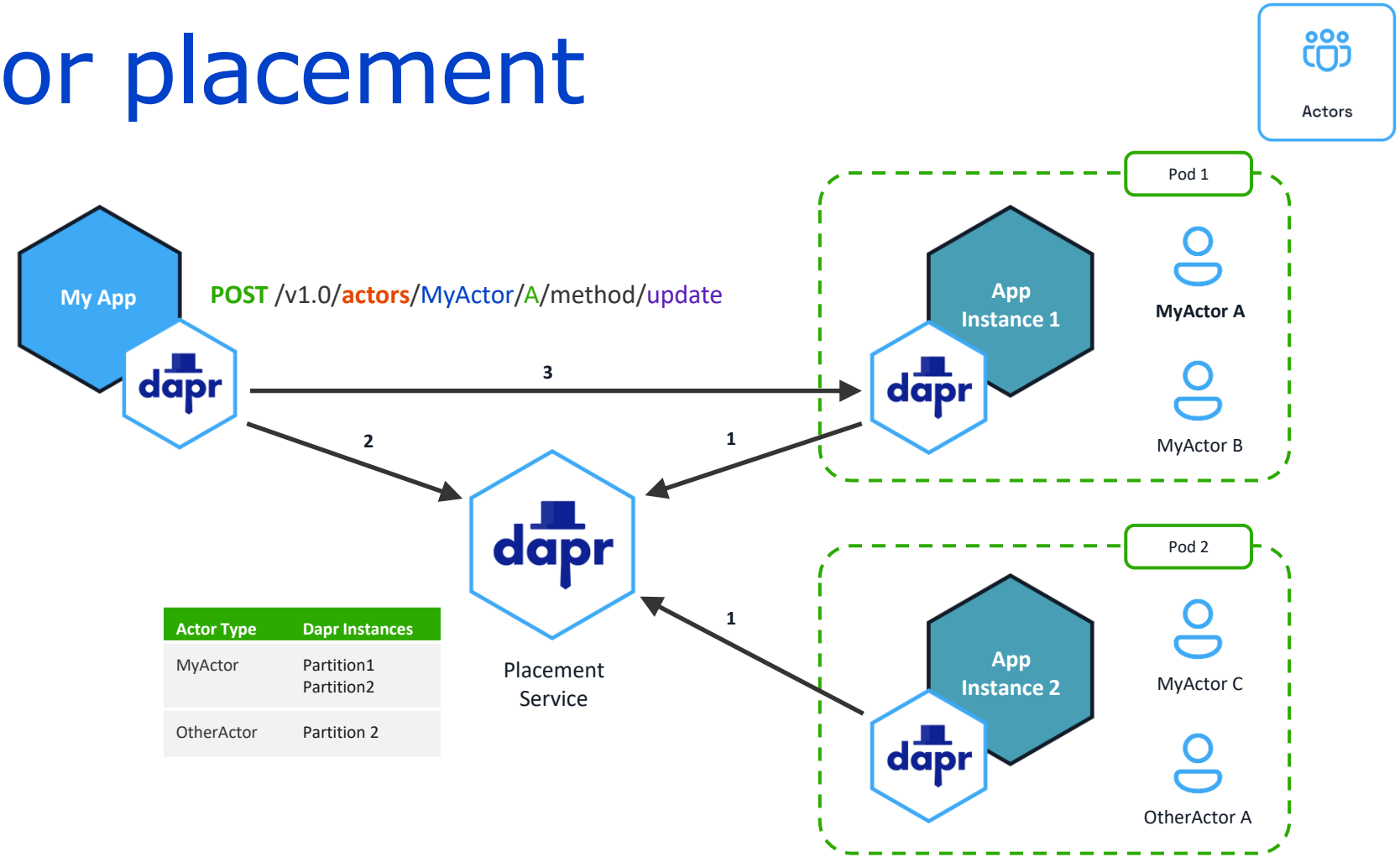
Actors



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

https://docs.dapr.io/developing-applications/building-blocks/actors/actors-overview/

Actor placement



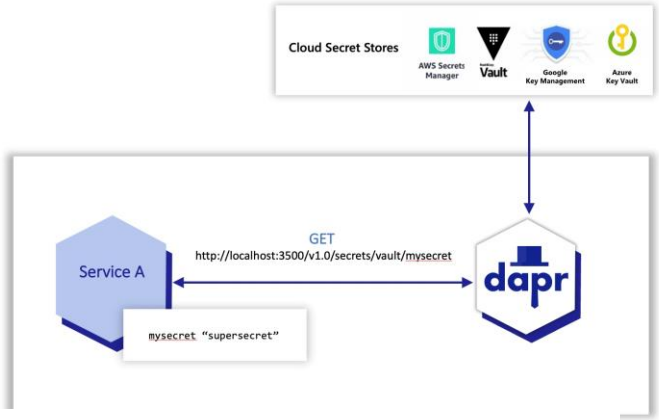

DEMO

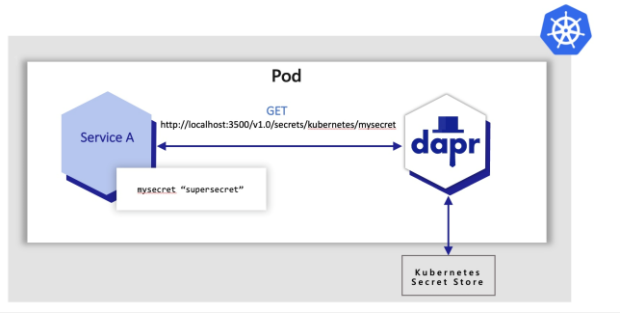
Dapr Building Blocks

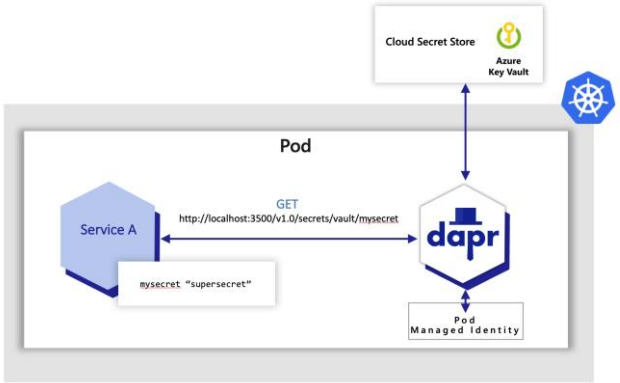
Secret Management

4tecture
empower your software solutions

Secret Management







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

https://docs.dapr.io/developing-applications/building-blocks/secrets/secrets-overview/

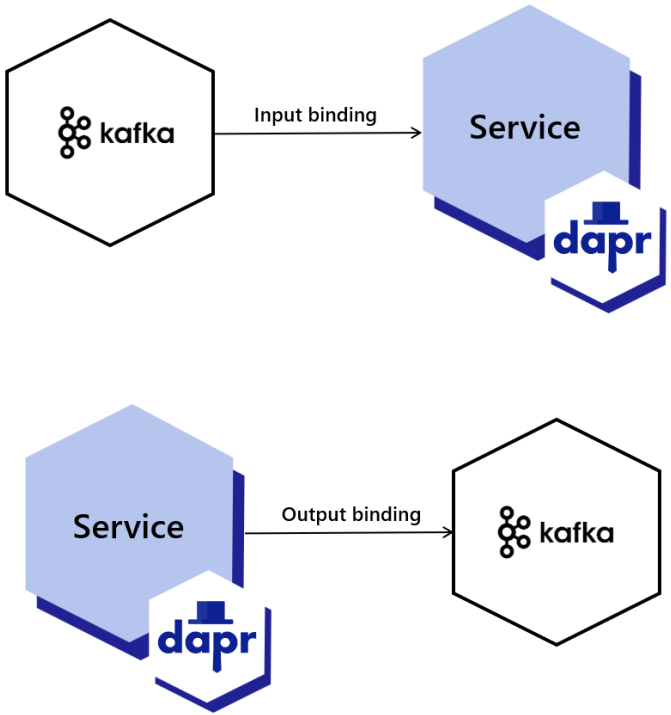

DEMO

Dapr Building Blocks

Input/Output Bindings

4tecture
empower your software solutions

Input/Output Binding



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

https://docs.dapr.io/developing-applications/building-blocks/bindings/bindings-overview/

Bindings

Generic

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

Microsoft Azure

Component	Input Binding	Output Binding	Status	Component version	Since runtime version
Azure Blob Storage	☐	☑	Stable	v1	1.0
Azure Cosmos DB (Gremlin API)	☐	☑	Alpha	v1	1.5
Azure CosmosDB	☐	☑	Stable	v1	1.7
Azure Event Grid	☑	☑	Beta	v1	1.7
Azure Event Hubs	☑	☑	Stable	v1	1.8
Azure OpenAI	☑	☑	Alpha	v1	1.11
Azure Service Bus Queues	☑	☑	Stable	v1	1.7
Azure SignalR	☐	☑	Alpha	v1	1.0
Azure Storage Queues	☑	☑	Stable	v1	1.0

https://docs.dapr.io/reference/components-reference/supported-bindings/

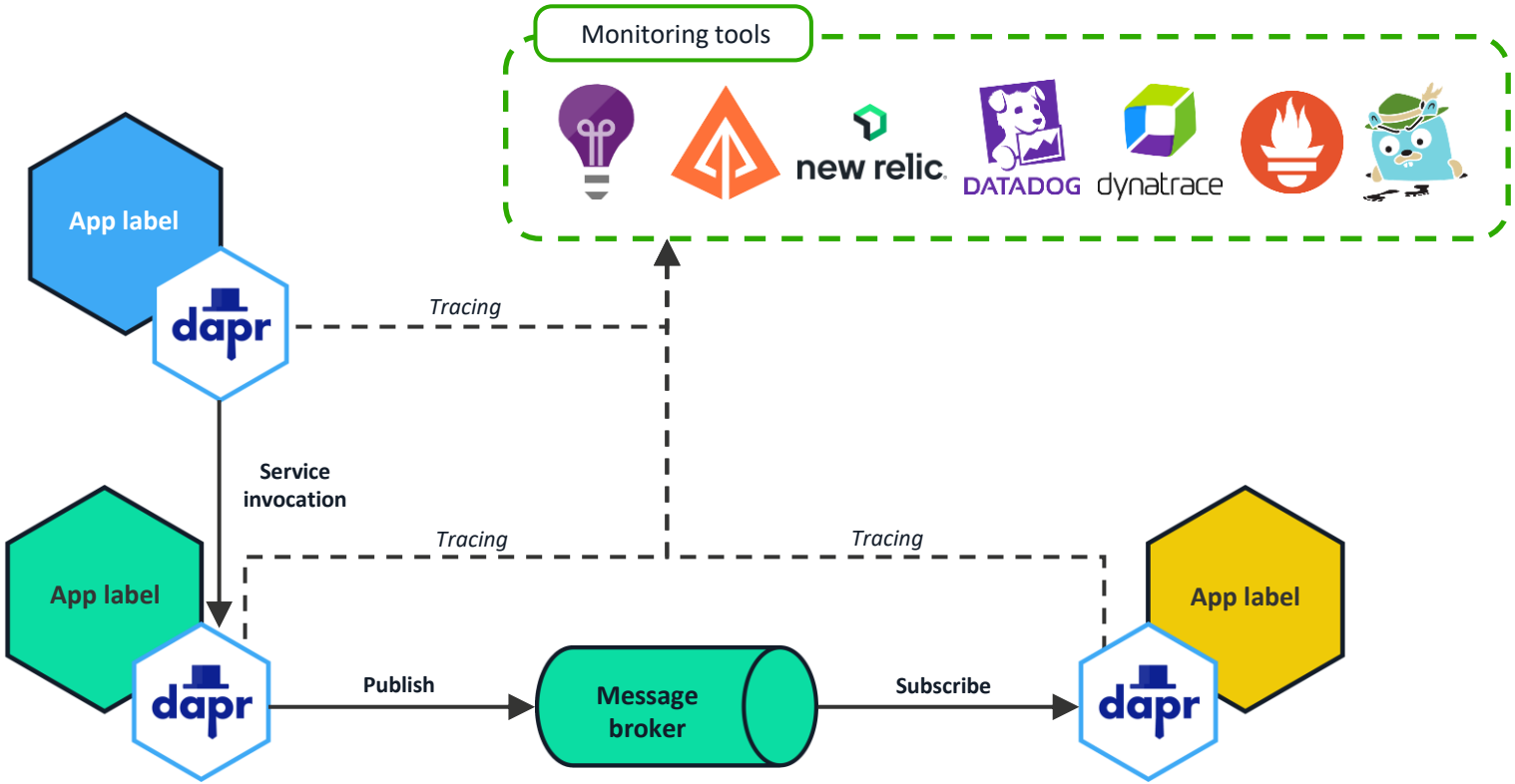
Dapr Building Blocks

Observability

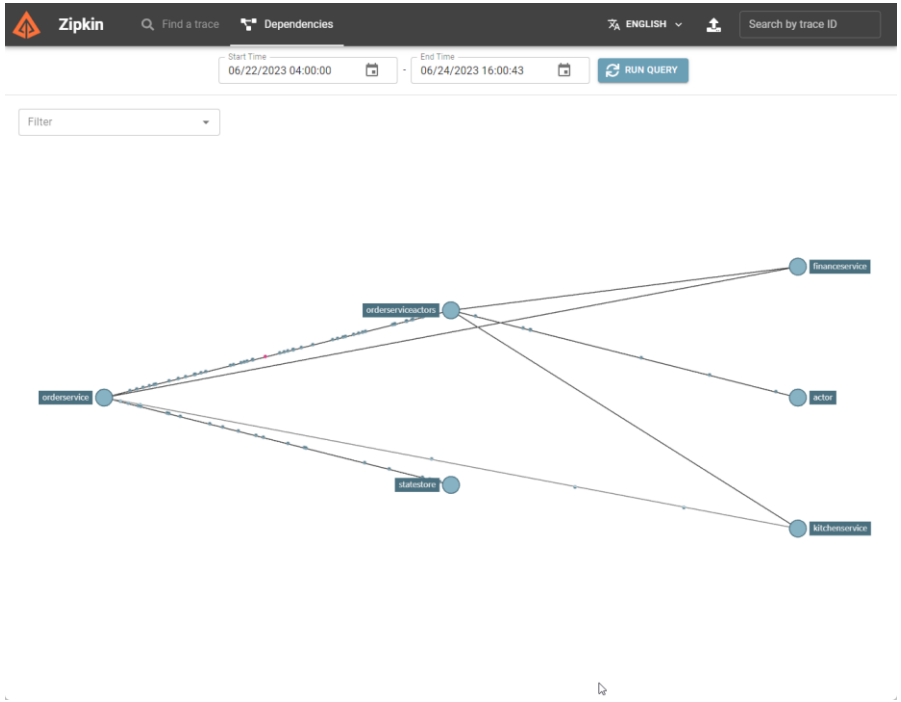
4tecture
empower your software solutions

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 binings
  - Short-circuiting all service-invocation requests

https://docs.dapr.io/developing-applications/building-blocks/observability/tracing-overview/

Dapr Building Blocks

Resiliency

4tecture
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.

```yaml
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
```


DEMO

.NET Microservices with Dapr

Conclusion

4tecture
empower your software solutions

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.)


.NET Microservices with Dapr
Q & A
4tecture
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

Marc Müller
Principal Consultant

www.powerofdevops.com

4tecture©
empower your software solutions

ProAuth

ReaFx


4tecture©
empower your software solutions