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

Marc Müller
Principal Consultant

MVP
Microsoft Most Valuable Professional

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

4tecture©
empower your software solutions

About me:

Marc Müller
Principal Consultant
@muellermarc

Microsoft® Most Valuable Professional

MVP

4tecture©

empower your software solutions

Our Products:

Multi-Tenant OpenID Connect Identity Provider

ProAuth

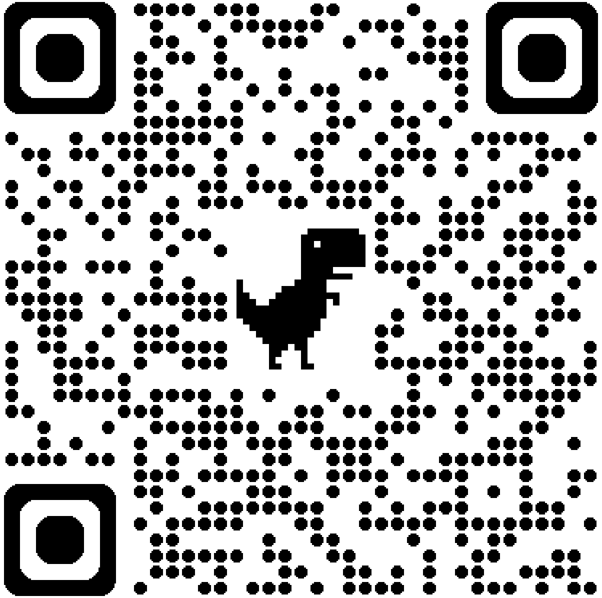
www.proauth.net

Enterprise Application Framework for .NET

ReaFx

www.reafx.net

Slide Download



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

Agenda

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

Agenda

Dapr

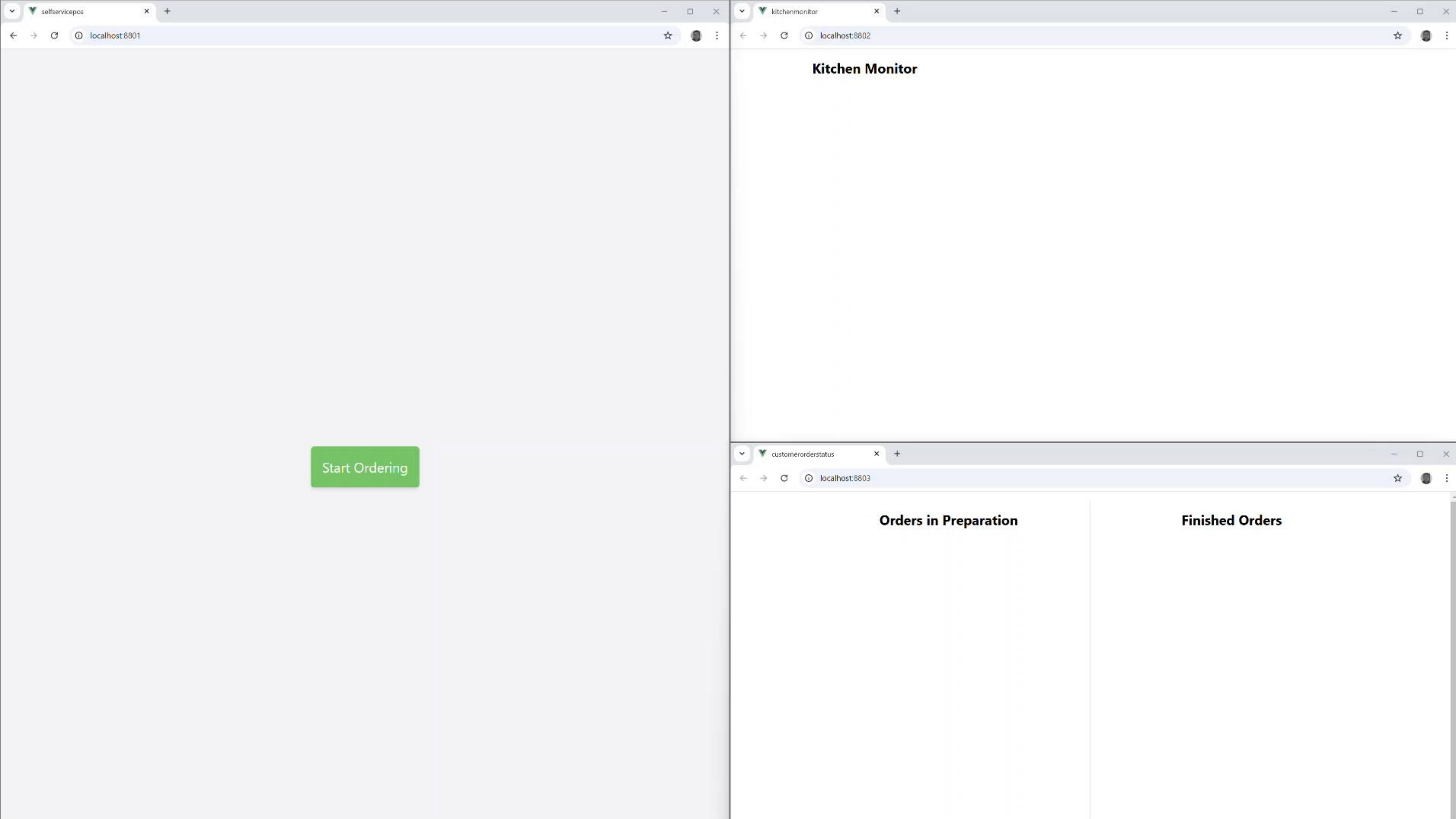
Intro

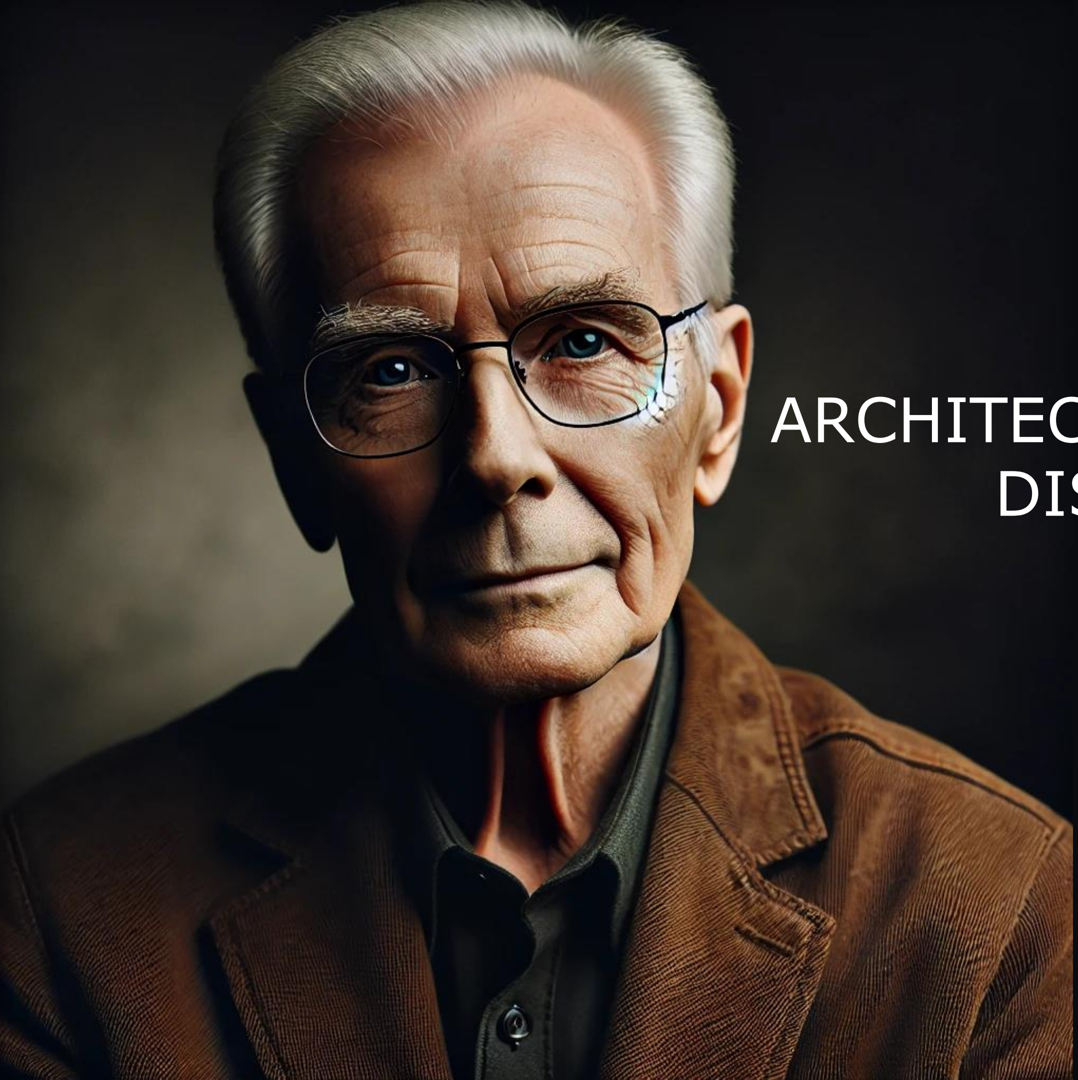
4tecture

empower your software solutions

DEMO

Fast Food Application


selfservicepos
localhost:8801
Start Ordering
kitchenmonitor
localhost:8802
Kitchen Monitor
customerorderstatus
localhost:8803
Orders in Preparation
Finished Orders



ARCHITECTING AND IMPLEMENTING
DISTRIBUTED APPLICATIONS
IS NOT STRESFUL AT ALL

John, 25 years old

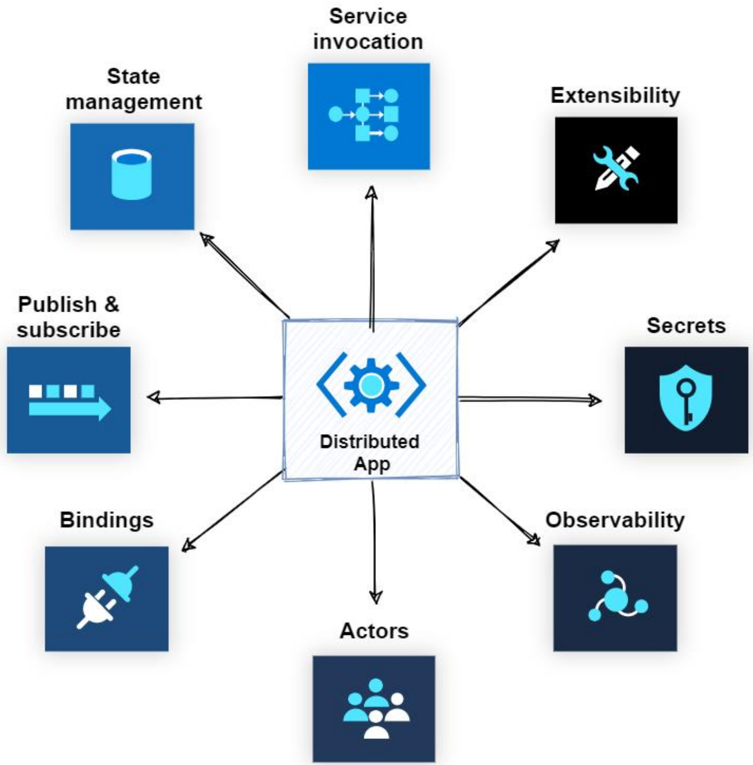
Challenges with Distributed Apps

- 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

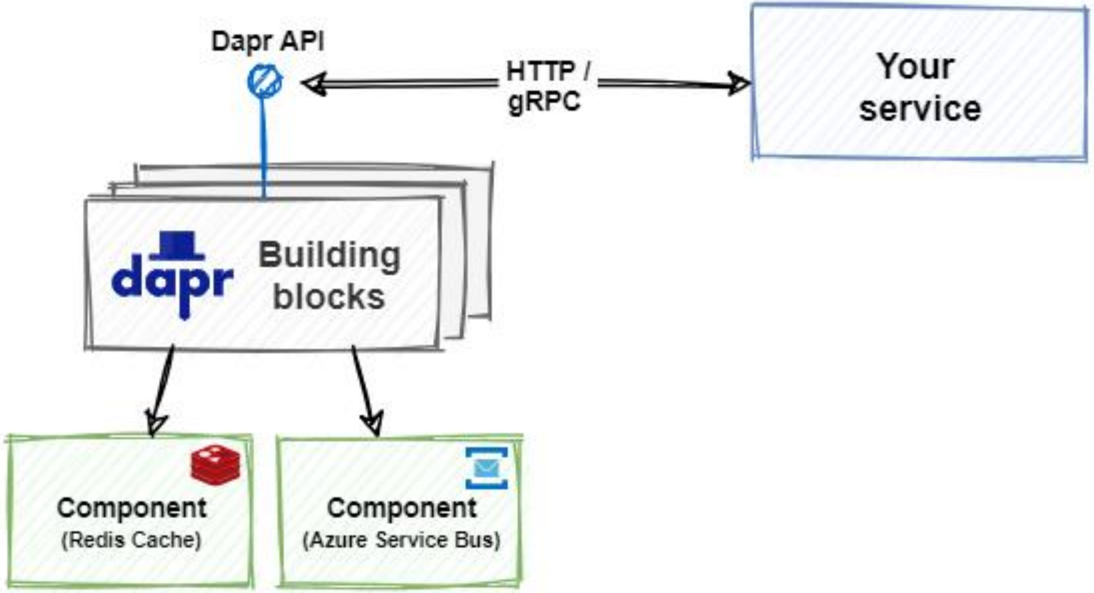
- Portable, event-driven runtime
- Build connected distributed applications faster
- APIs for solving distributed application challenges
- Cloud and Edge

Dapr Building Blocks



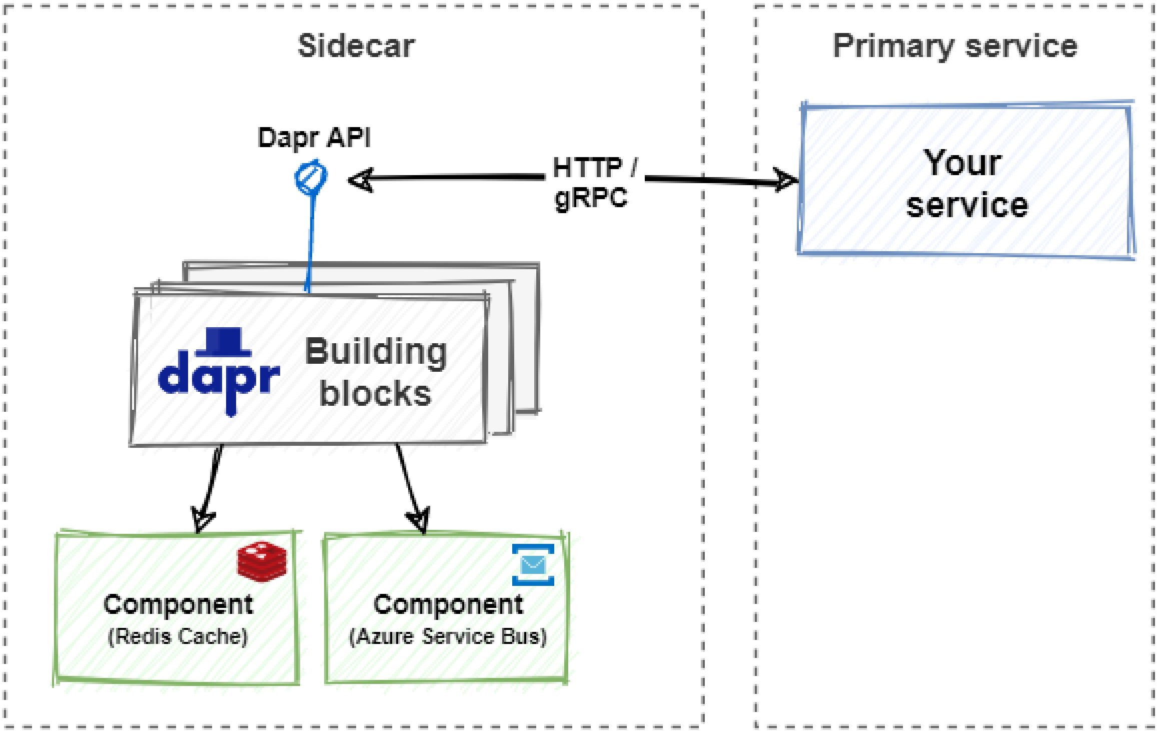
https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

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

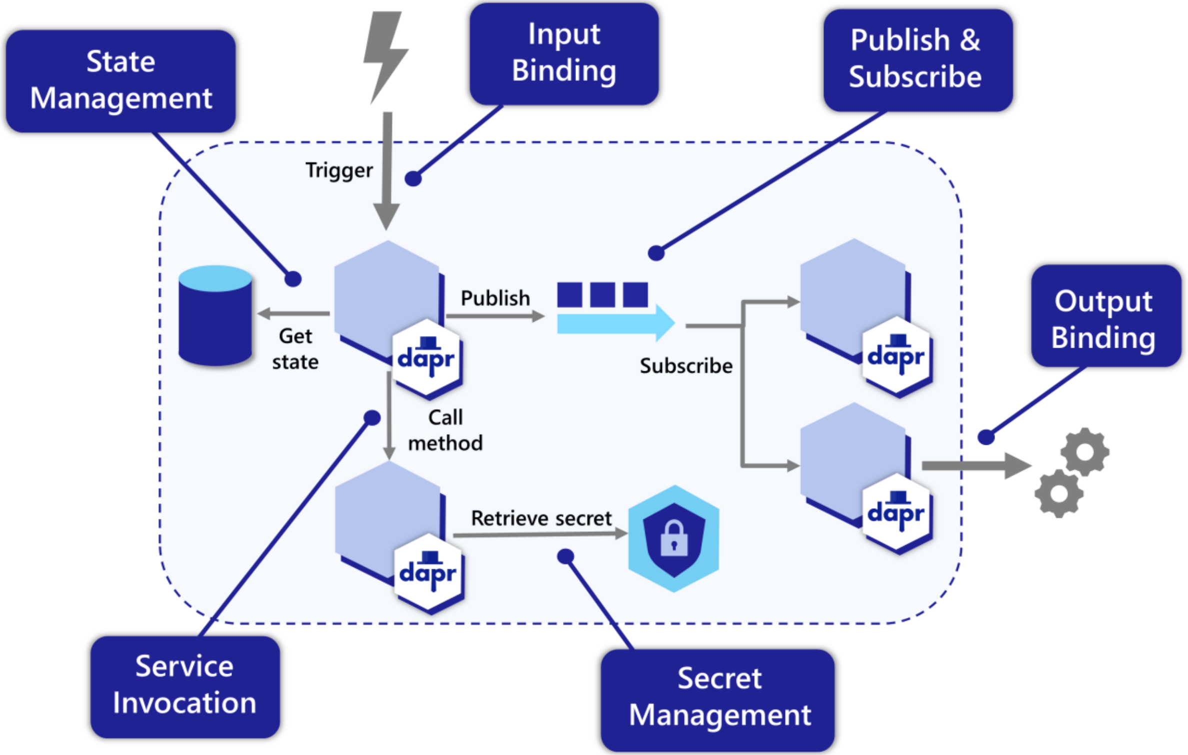
Dapr

- 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);
```

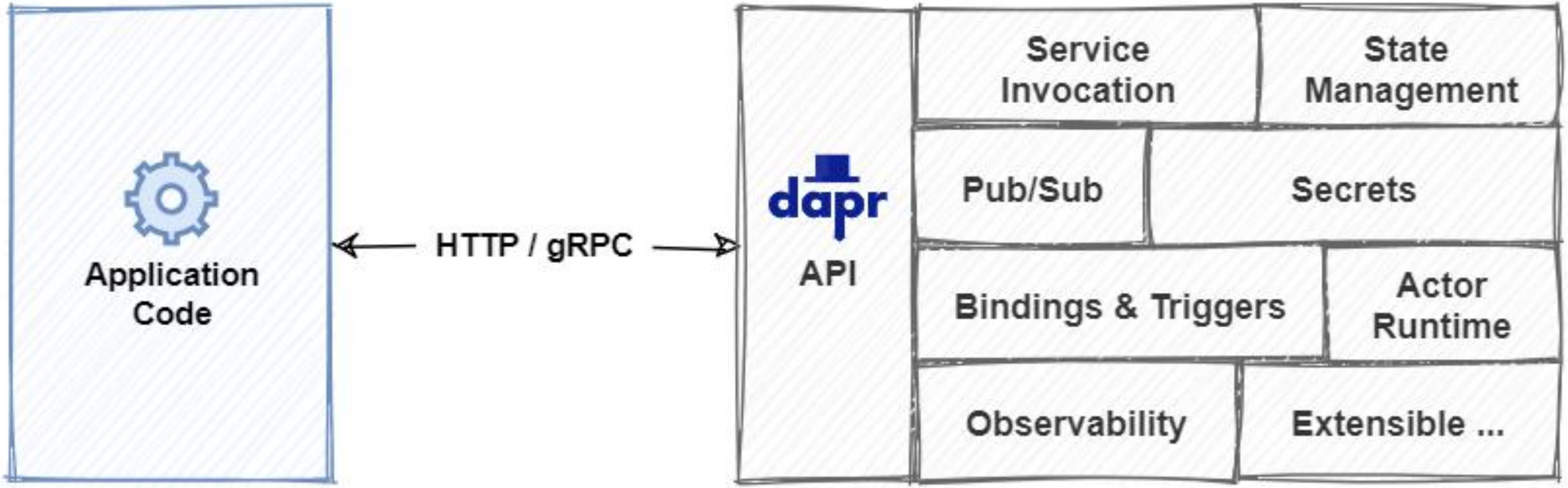
- AWS DynamoDB
- Aerospike
- Azure Blob Storage
- Azure CosmosDB
- Azure Table Storage
- Cassandra
- Cloud Firestore (Datastore mode)
- CloudState
- Couchbase
- Etcd
- HashiCorp Consul
- Hazelcast
- Memcached
- MongoDB
- PostgreSQL
- Redis
- RethinkDB
- SQL Server
- Zookeeper

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.

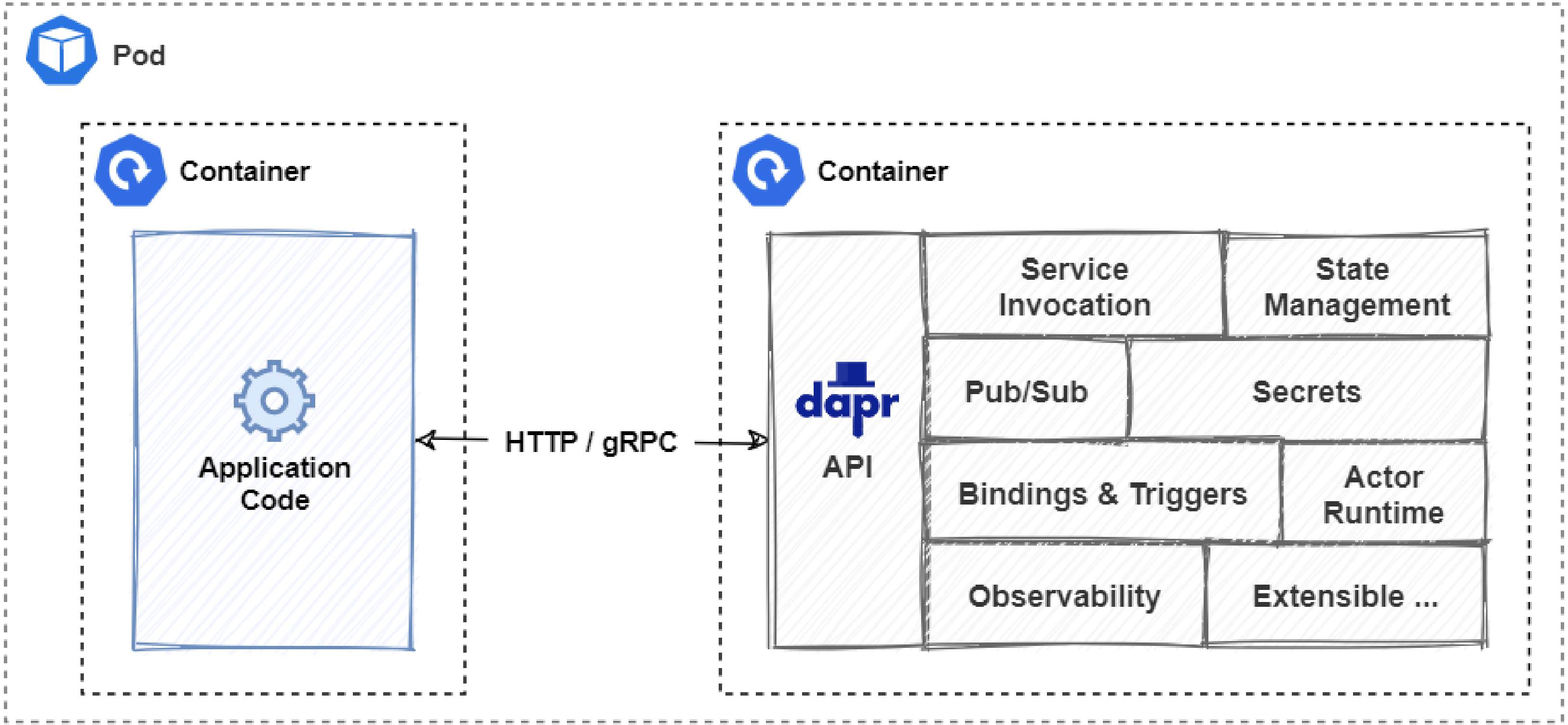
Speeds up application development by providing an integrated set of APIs for communication, state, and workflow.

Self-hosted Sidecar



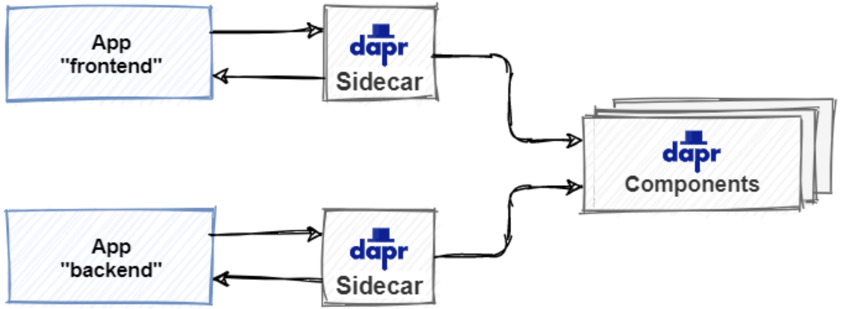
https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/dapr-at-20000-feet

Kubernetes-hosted Sidecar



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

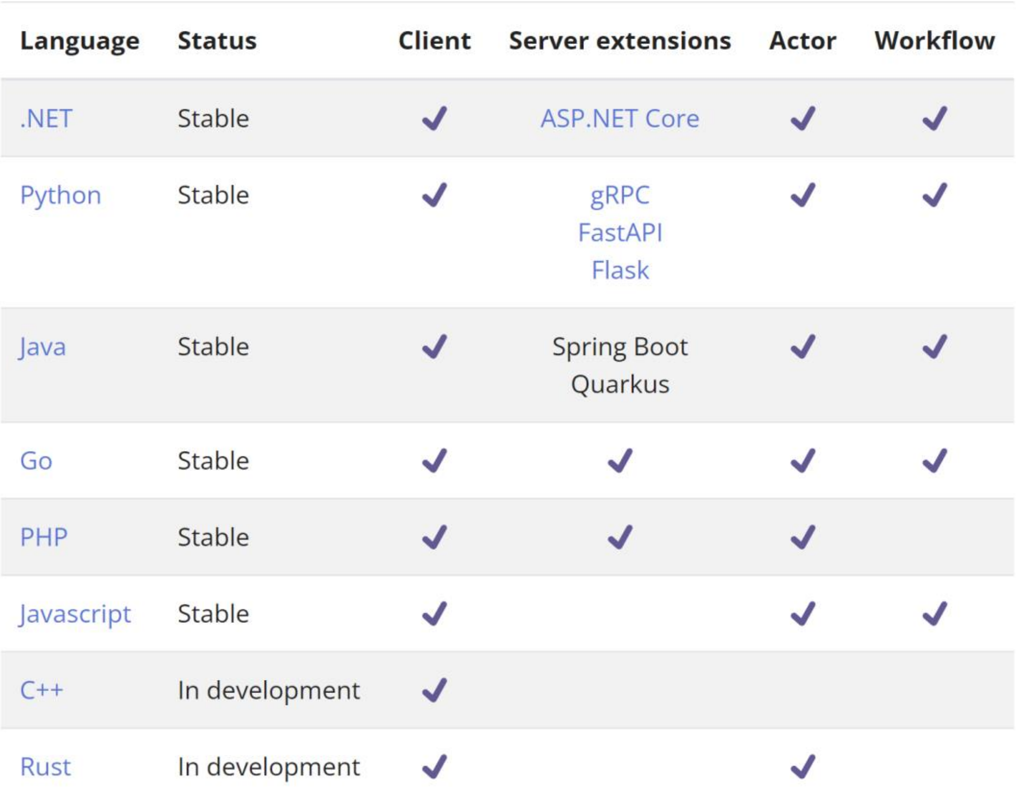
Sidecar Performance Considerations



- 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

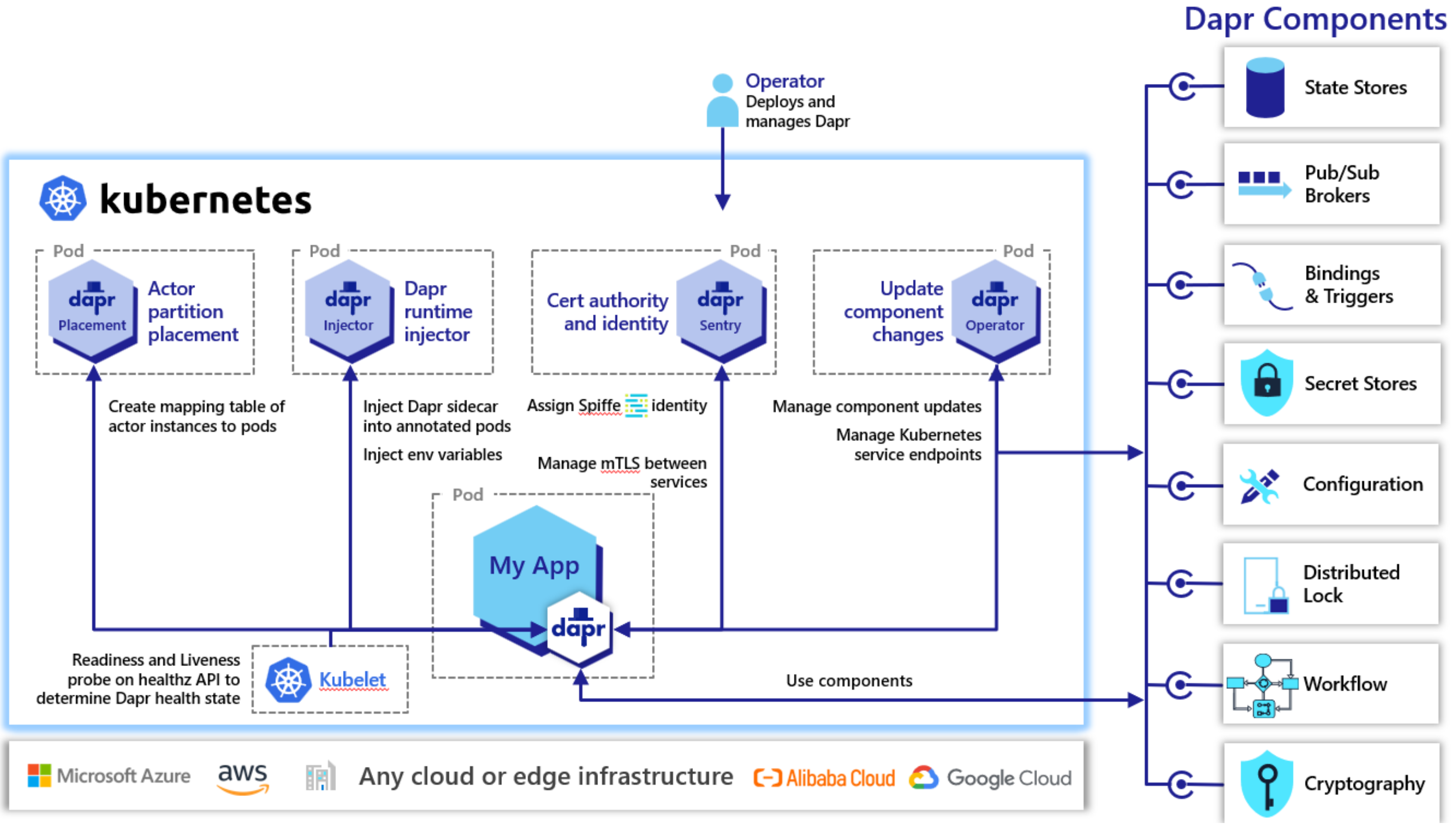
Kubernetes

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

Serverless

- Integrated in Azure
- i.e. Container Apps

Dapr in Kubernetes



Dapr Components

Operator
Deploys and manages Dapr

kubernetes

Pod: Placement (Actor partition placement)

Pod: Injector (Dapr runtime injector)

Pod: Sentry (Cert authority and identity)

Pod: Operator (Update component changes)

Create mapping table of actor instances to pods

Inject Dapr sidecar into annotated pods

Inject env variables

Assign Spiffe identity

Manage mTLS between services

Manage component updates

Manage Kubernetes service endpoints

Pod: My App (dapr)

Readiness and Liveness probe on healthz API to determine Dapr health state

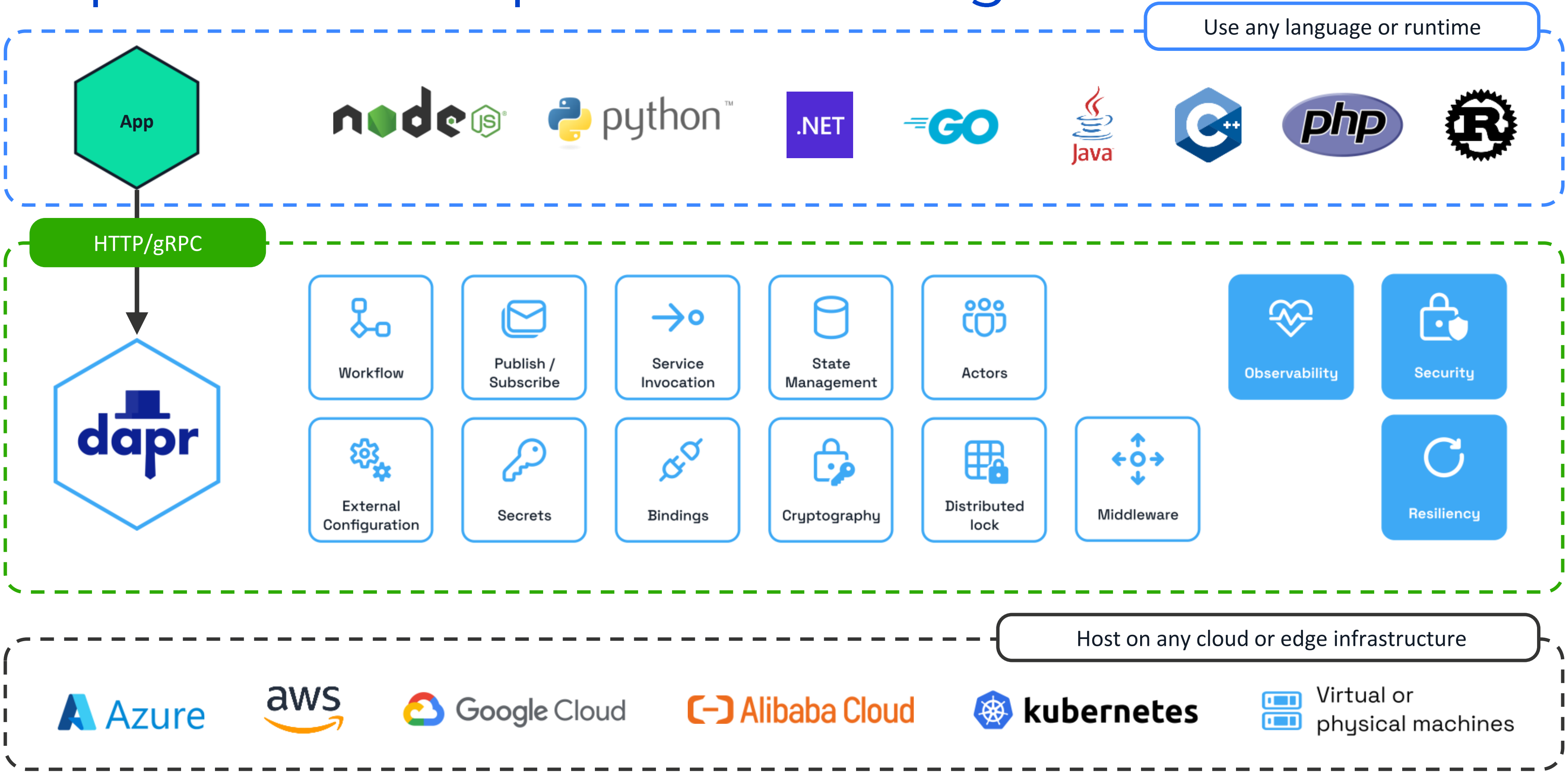
Kubelet

Use components

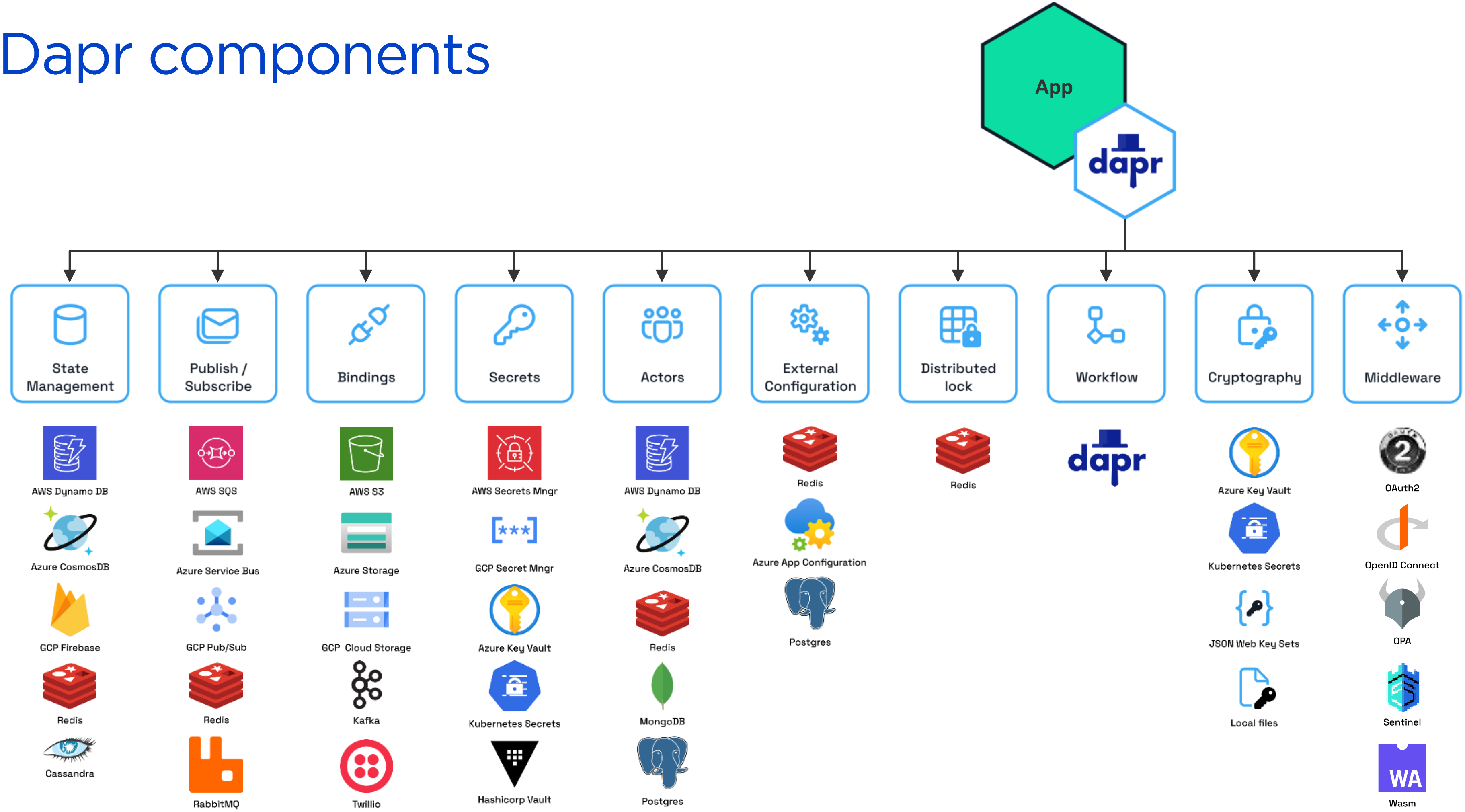
Any cloud or edge infrastructure

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

Dapr from development to hosting



Dapr components



App

dapr

- **State Management**: AWS Dynamo DB, Azure CosmosDB, GCP Firebase, Redis, Cassandra
- **Publish / Subscribe**: AWS SQS, Azure Service Bus, GCP Pub/Sub, Redis, RabbitMQ
- **Bindings**: AWS S3, Azure Storage, GCP Cloud Storage, Kafka, Twillio
- **Secrets**: AWS Secrets Mngr, GCP Secret Mngr, Azure Key Vault, Kubernetes Secrets, Hashicorp Vault
- **Actors**: AWS Dynamo DB, Azure CosmosDB, Redis, MongoDB, Postgres
- **External Configuration**: Redis, Azure App Configuration, Postgres
- **Distributed lock**: Redis
- **Workflow**: dapr
- **Cryptography**: Azure Key Vault, Kubernetes Secrets, JSON Web Key Sets, Local files
- **Middleware**: OAuth2, OpenID Connect, OPA, Sentinel, Wasm

Hands-on!

4tecture

empower your software solutions

DEMO

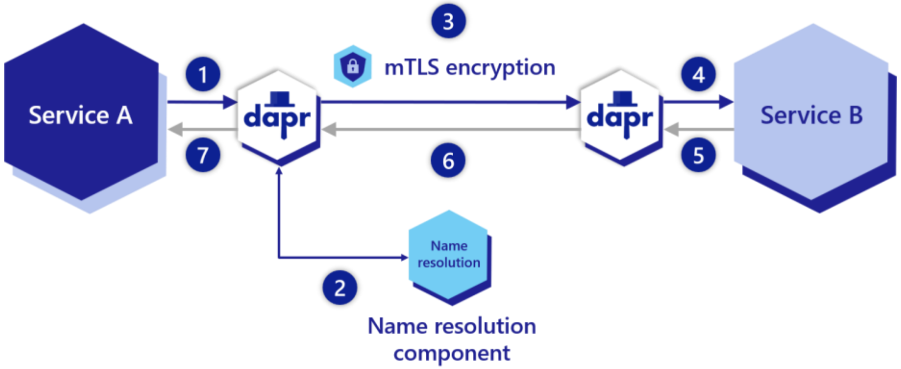
Dapr Building Blocks

Service Invocation and Configuration

4tecture

empower your software solutions

Service Invocation

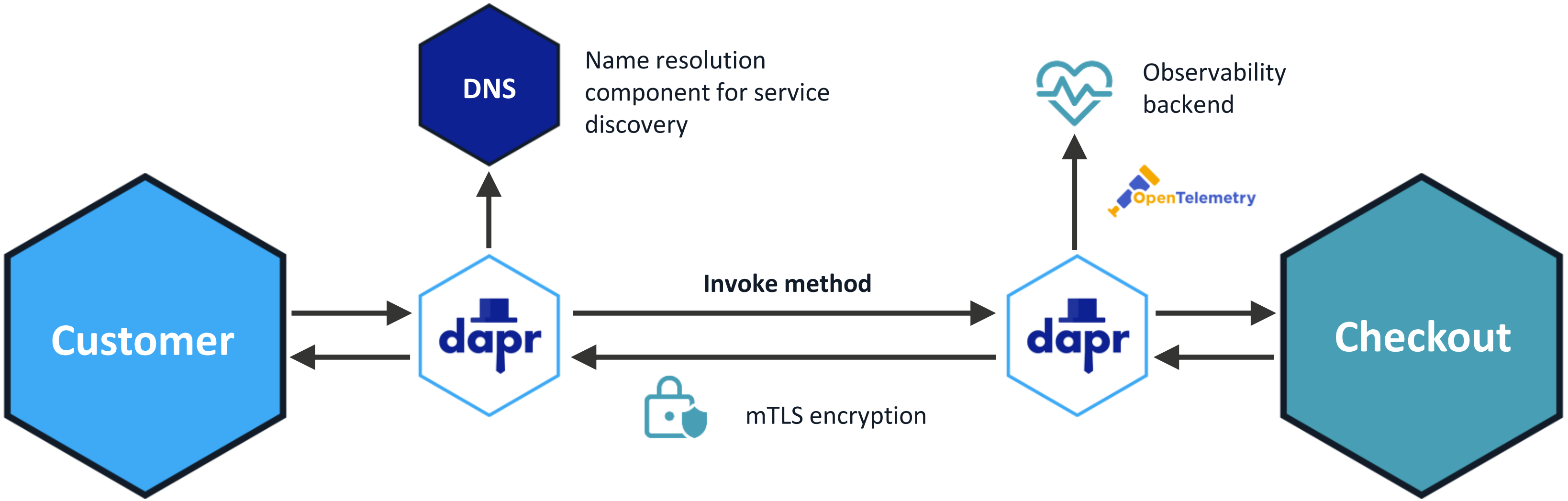


- HTTP and gRPC
- mTLS (with Dapr Sentry)
- Resiliency including retries
- Tracing and metrics with observability
- Access control (policies)
- Namespace scoping
- Load balancing (round robin with mDNS)
- Pluggable service discovery

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

Service Invocation

Service
Invocation



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

POST
http://localhost:5100/order

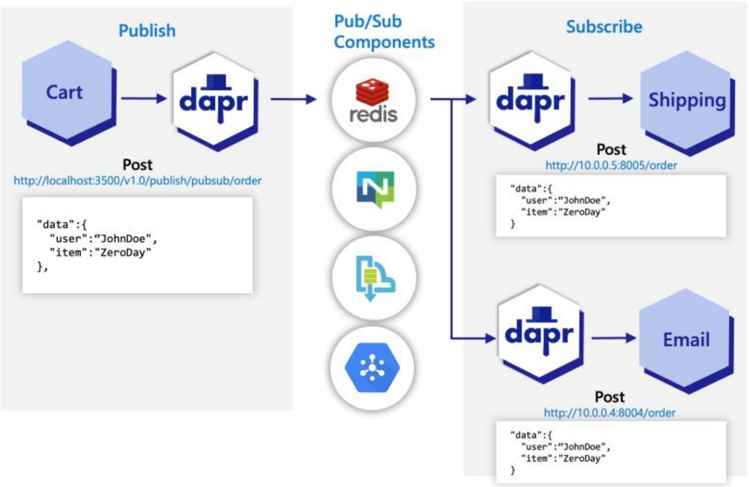
Dapr Building Blocks

Publish & Subscribe

4tecture©

empower your software solutions

Publish & Subscribe


Publish
Cart
dapr
Post
http://localhost:3500/v1.0/publish/pubsub/order
Pub/Sub
Components
redis
Subscribe
dapr
Shipping
Post
http://10.0.0.5:8005/order
dapr
Email
Post
http://10.0.0.4:8004/order

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

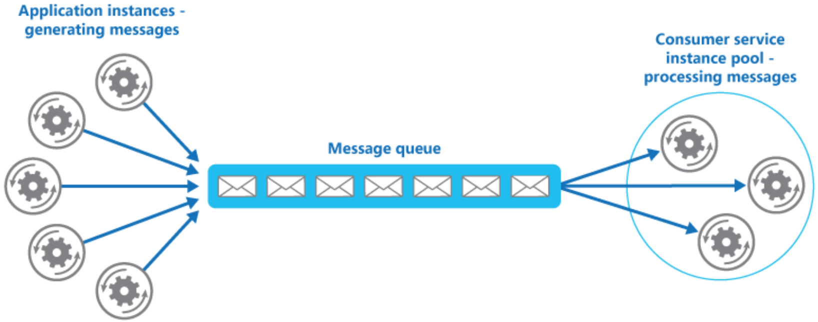
```
"data":{
  "user":"JohnDoe",
  "item":"ZeroDay"
}
```

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

- 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

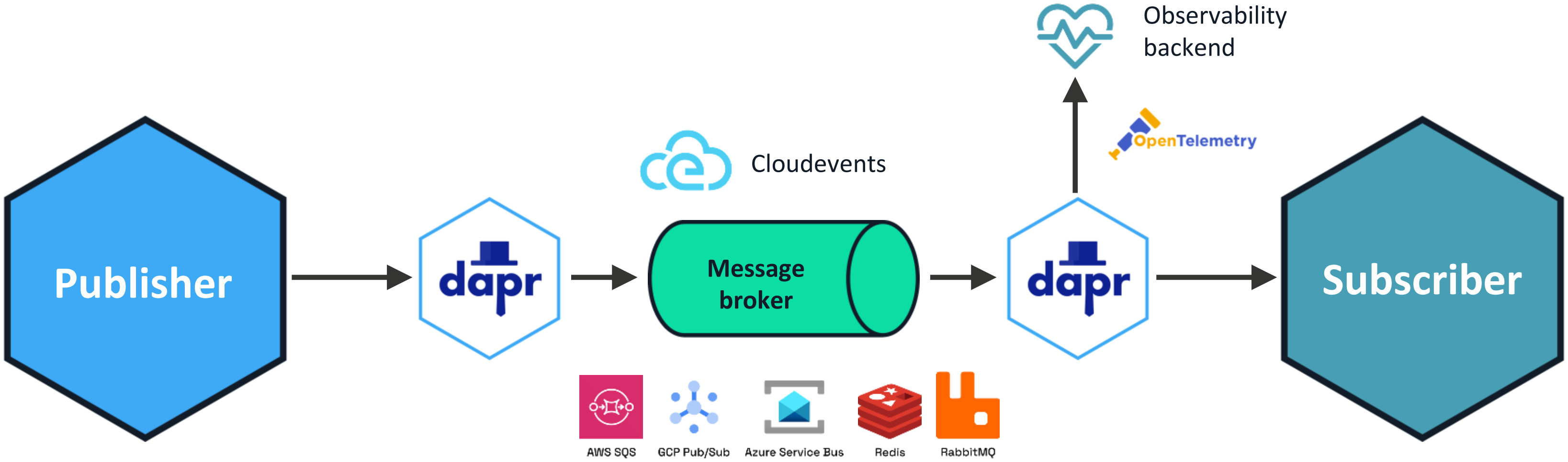

Application instances - generating messages
Message queue
Consumer service instance pool - processing messages

- 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

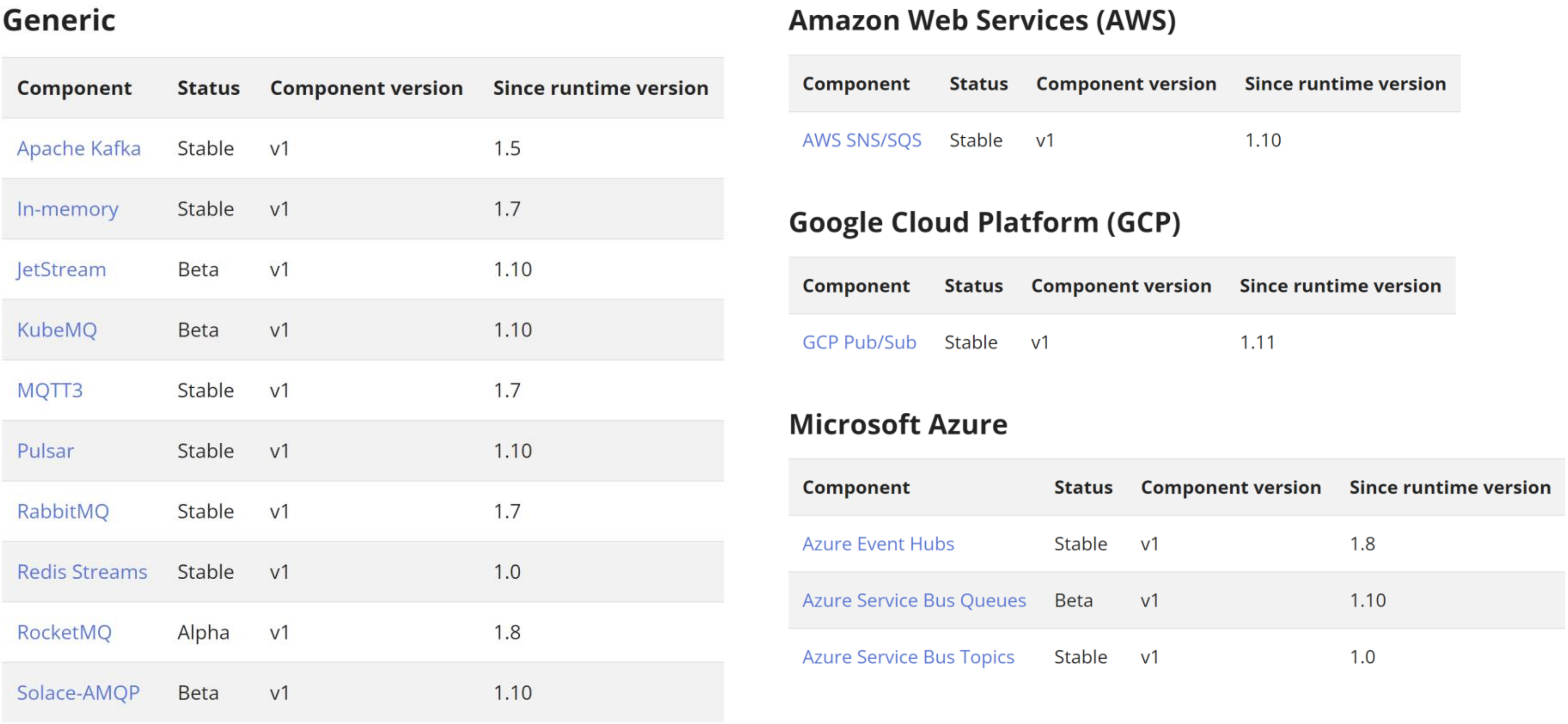
Publish / Subscribe



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

Sidecars, mTLS, PubSub, StateStore, Actors, Workflows

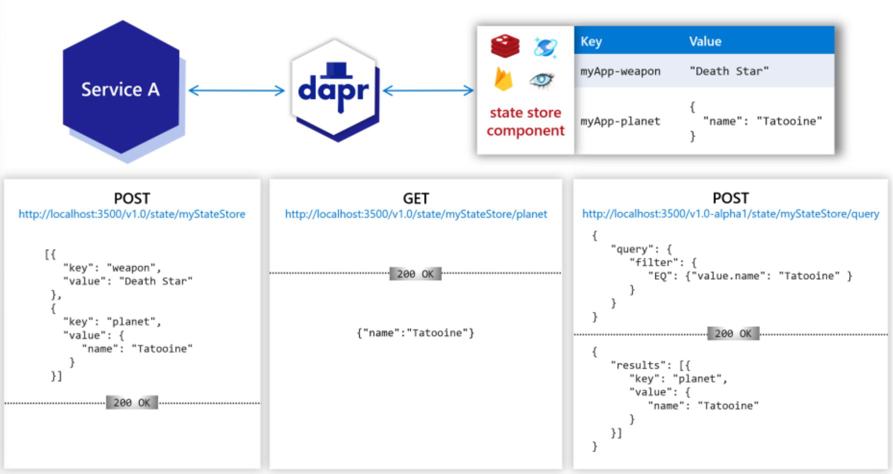
Dapr Building Blocks

State Management

4tecture

empower your software solutions

State Management



Key	Value
myApp-weapon	"Death Star"
myApp-planet	{ "name": "Tatooine" }

POST
http://localhost:3500/v1.0/state/myStateStore

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

200 OK

GET
http://localhost:3500/v1.0/state/myStateStore/planet

200 OK

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

POST
http://localhost:3500/v1.0-alpha1/state/myStateStore/query

```
{
   "query": {
     "filter": {
       "EQ": {"value.name": "Tatooine" }
     }
   }
}
```

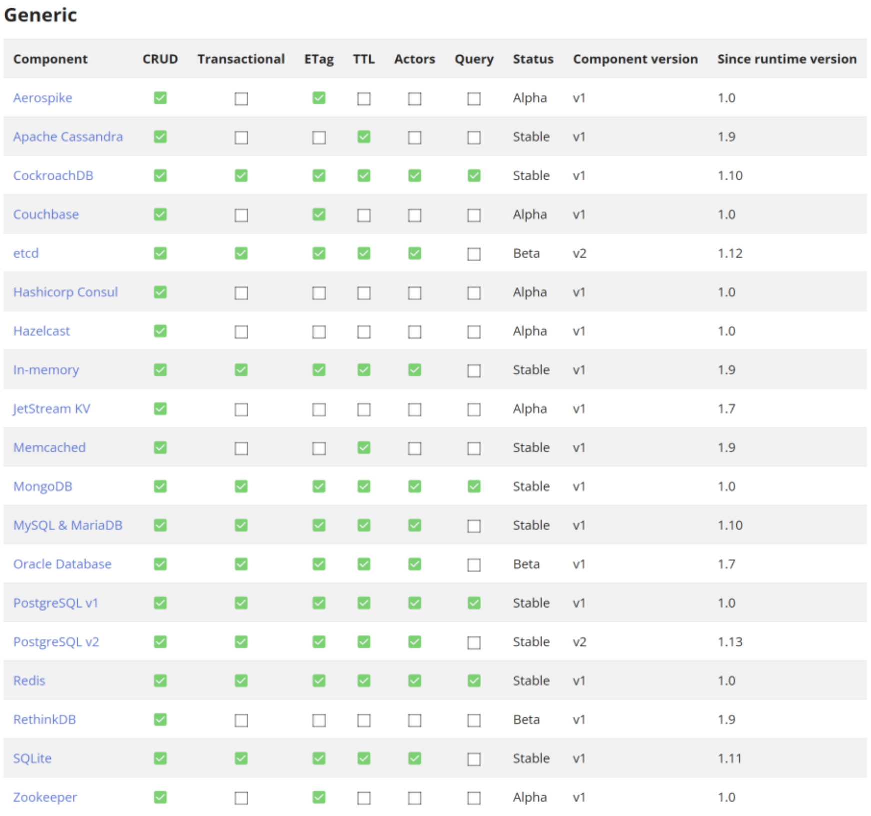
200 OK

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

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

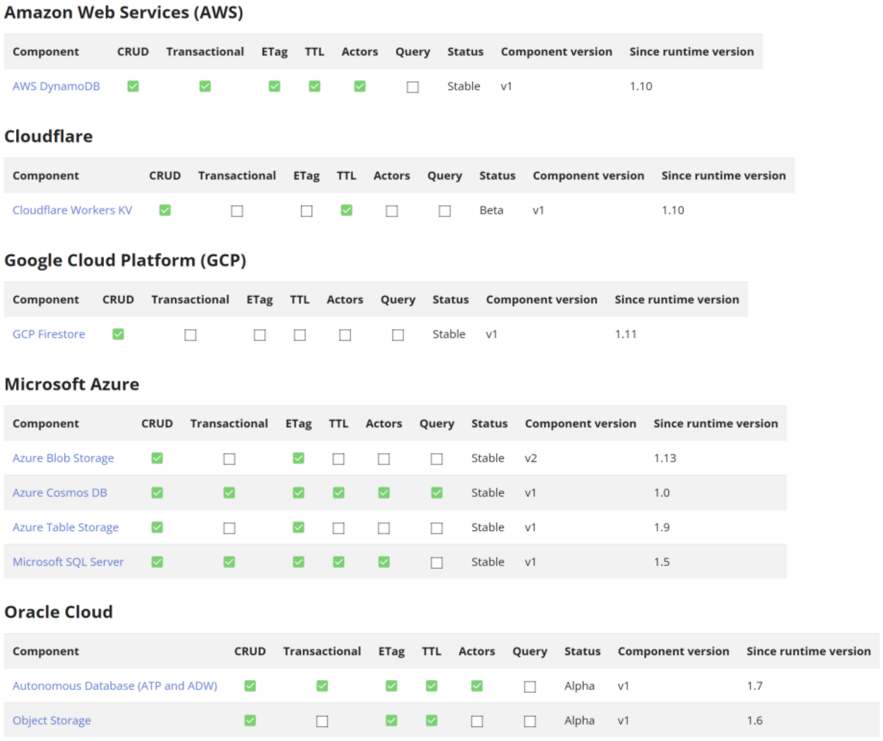
https://docs.dapr.io/developing-applications/building-blocks/state-management/state-management-overview/

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/

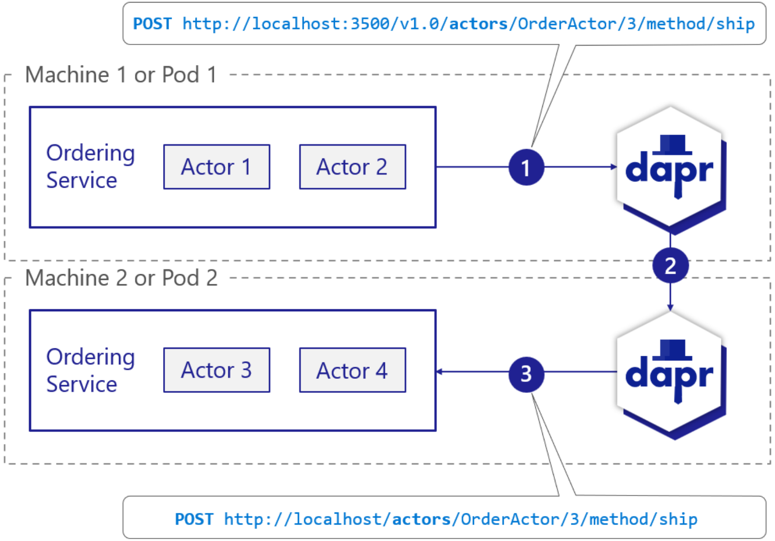
Dapr Building Blocks

Virtual Actors

4tecture©

empower your software solutions

Actors

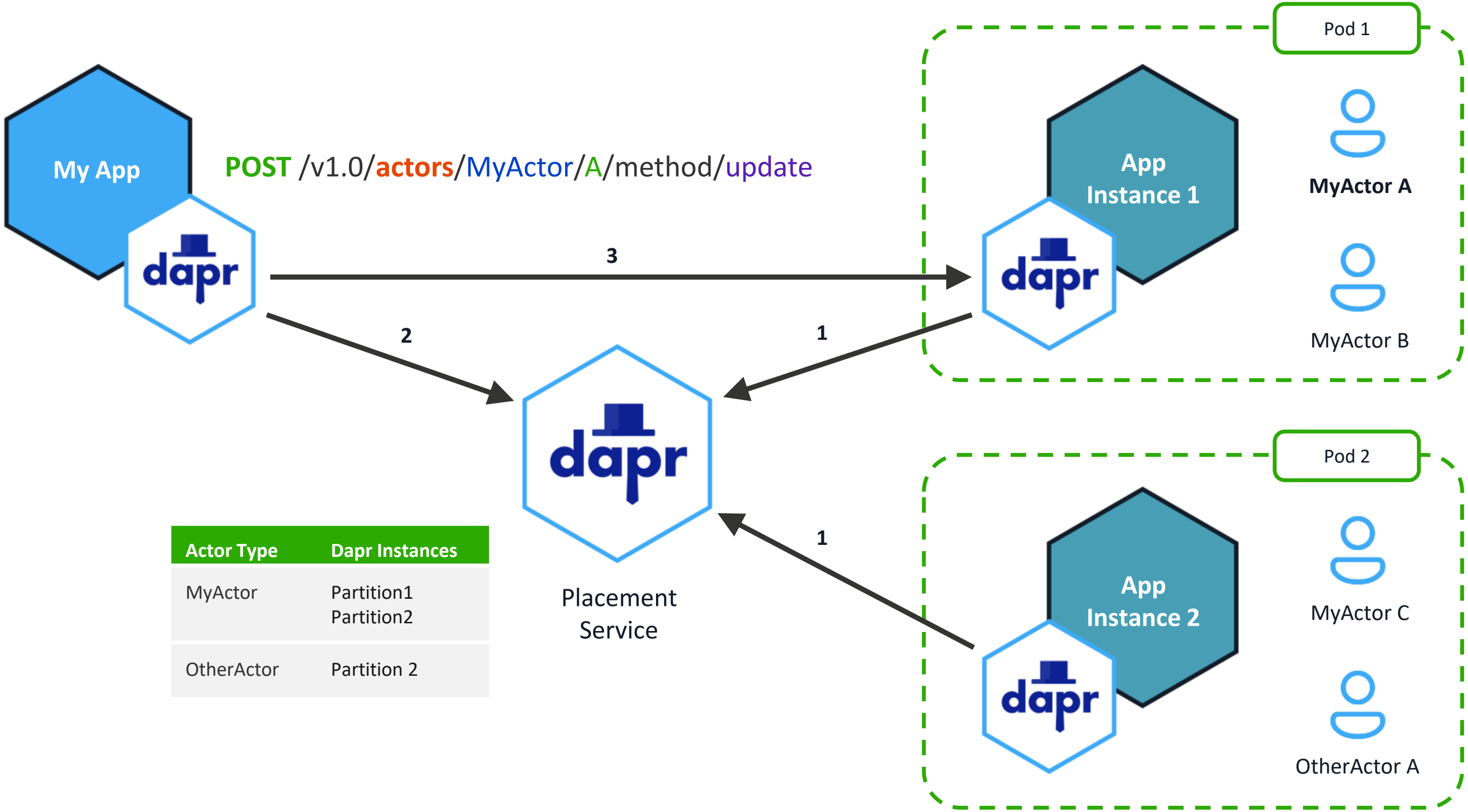

POST http://localhost:3500/v1.0/actors/OrderActor/3/method/ship
Machine 1 or Pod 1
Ordering Service
Actor 1
Actor 2
1
dapr
2
Machine 2 or Pod 2
Ordering Service
Actor 3
Actor 4
3
dapr
POST http://localhost/actors/OrderActor/3/method/ship

- 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

Actors



POST /v1.0/actors/MyActor/A/method/update

Actor Type	Dapr Instances
MyActor	Partition1 Partition2
OtherActor	Partition 2

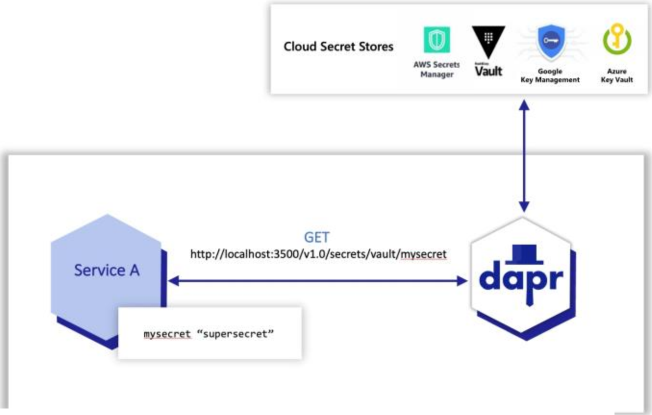
Dapr Building Blocks

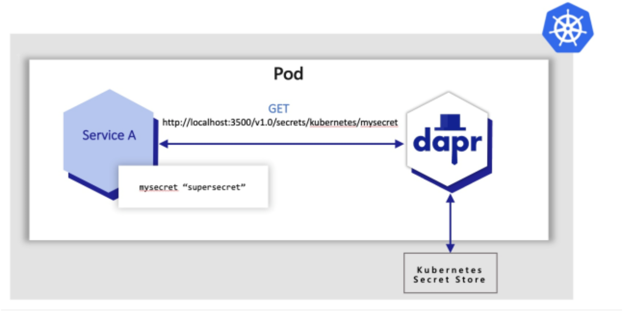
Secret Management

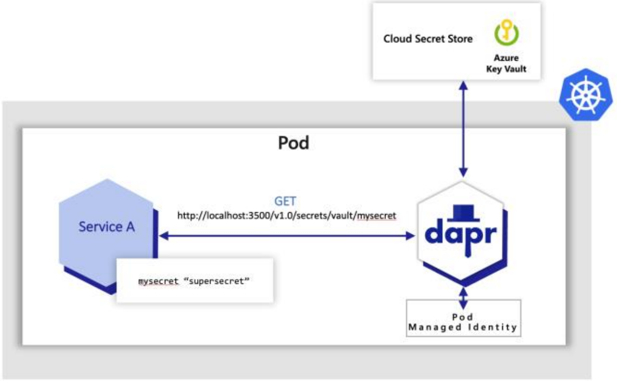
4tecture©
empower your software solutions

Secret Management

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







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

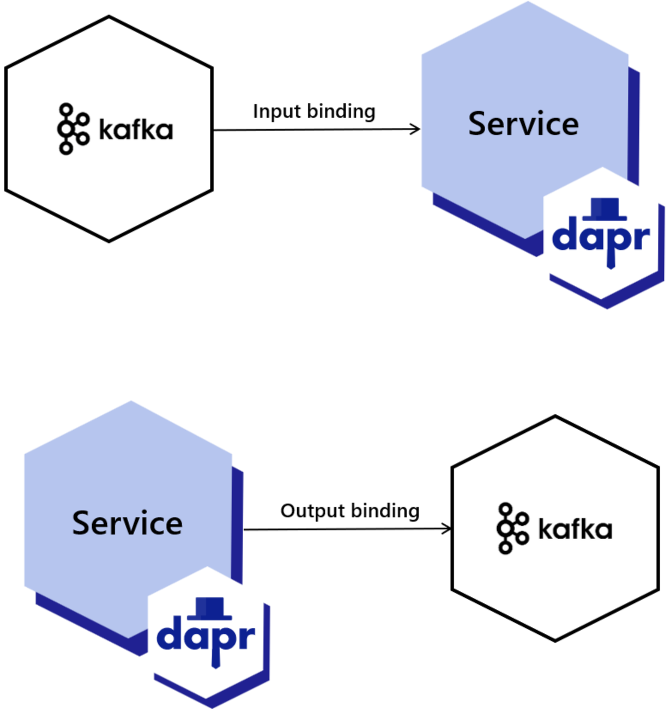
Dapr Building Blocks

Input/Output Bindings

4tecture

empower your software solutions

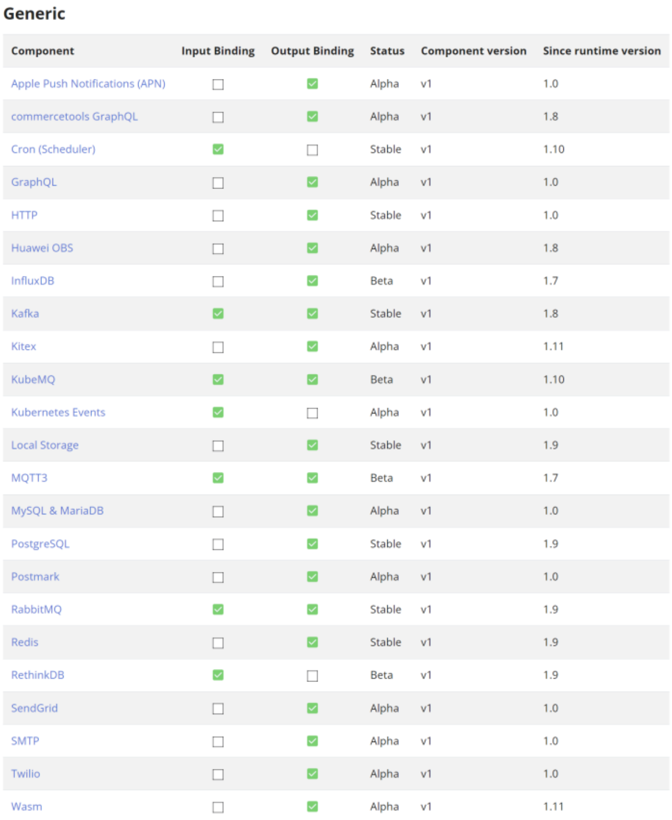
Input/Output Binding



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

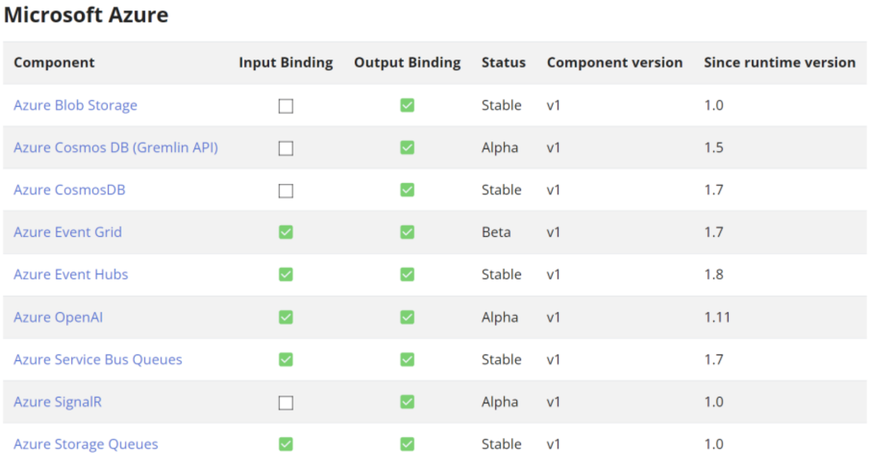
https://docs.dapr.io/developing-applications/building-blocks/bindings/bindings-overview/

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

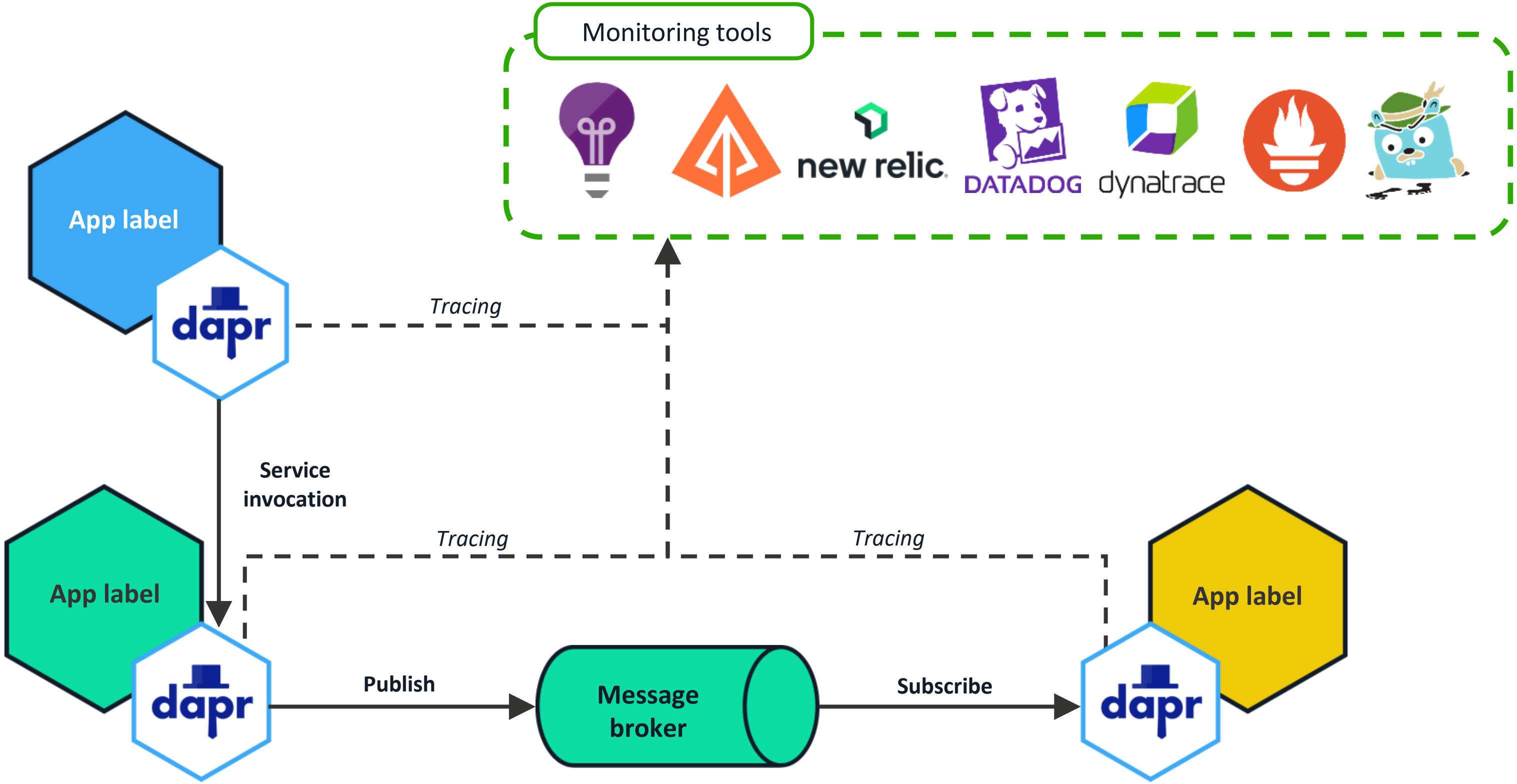
Observability

4tecture

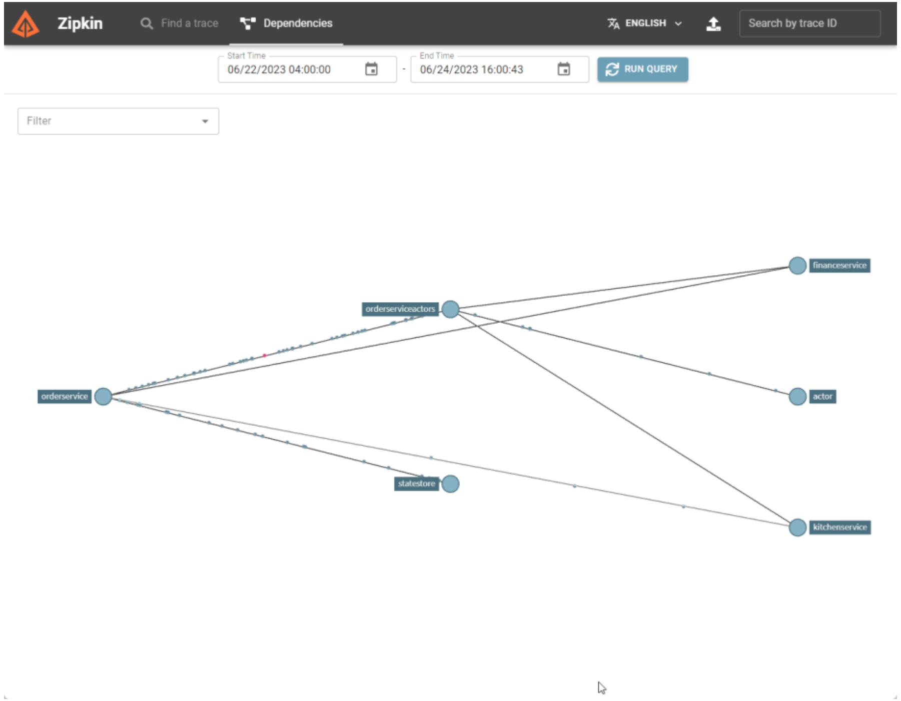
empower your software solutions

Distributed tracing

Observability



Observability



Start Time 06/22/2023 04:00:00 - End Time 06/24/2023 16:00:43

- 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
```

.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