Simplifying Microservices with Dapr

matt sheehan
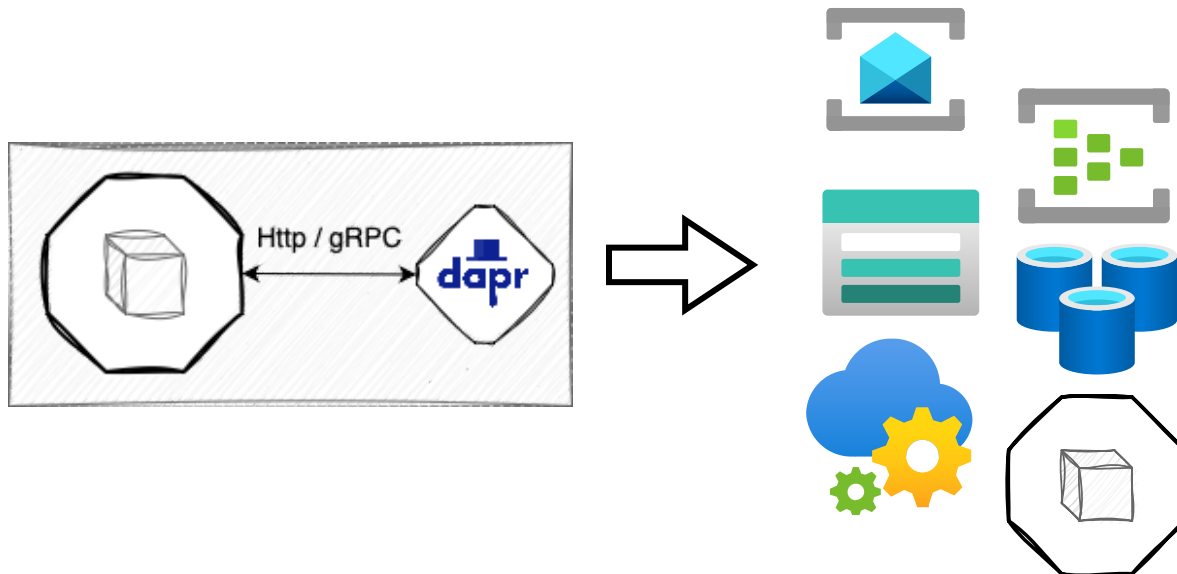
Github Code
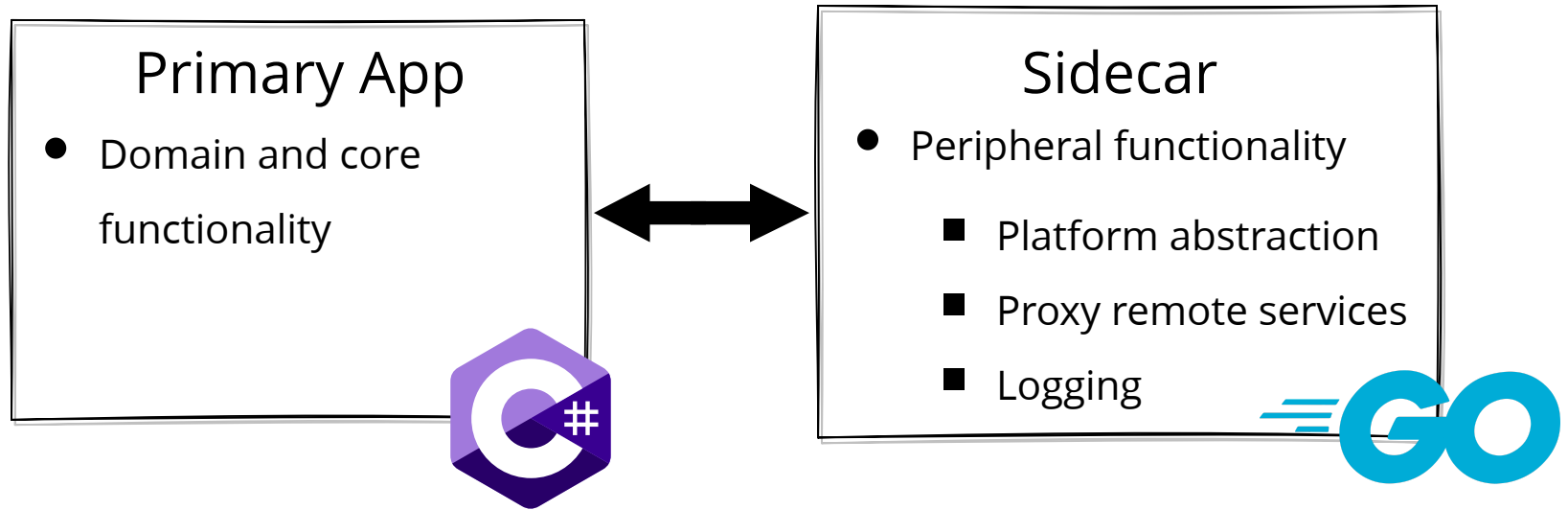

https://github.com/MattSheehanDev/SimplifyingMicroservicesWithDapr

Introducing dapr

- **D**istributed **Ap**plication **R**untime
- Introduced in Oct. 2019 as a Microsoft open-source project
- Joined Cloud Native Computing Foundation (CNCF) in Nov. 2021 as an incubating project
- Codifies best patterns and practices
- Sidecar to enable highly distributed apps

Sidecar Pattern

Primary App
- Domain and core functionality

Sidecar
- Peripheral functionality
  - Platform abstraction
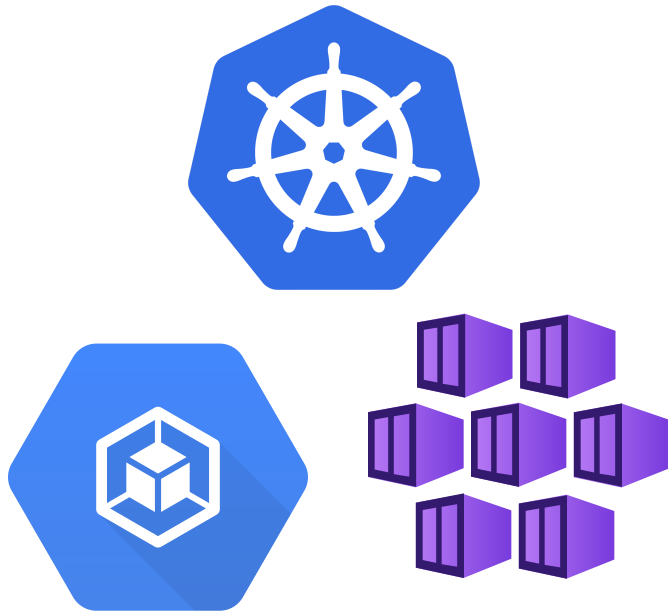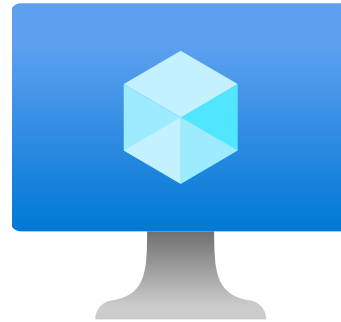 - Proxy remote services
  - Logging

Advantages

- Provides a homogenous interface for platform services
- Simplifies the plumbing and glue
- Independent from primary application runtime environment and language
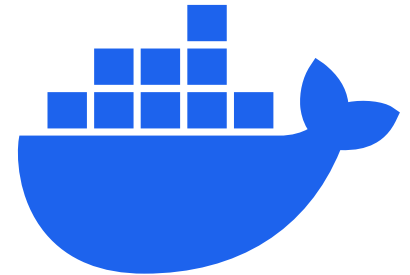
Hosting

Kubernetes

(GKE, AKS, EKS)

Virtual Machine

Docker container

dapr benefits

Language neutral

Provides flexibility and simplicity

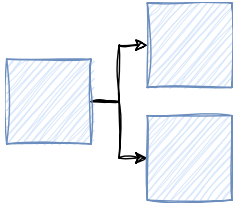Decreases your API surface area

Codifies best patterns and practices
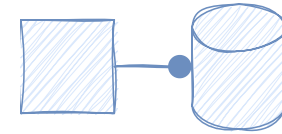
Provides a standard and declarative configuration

Moves design decisions downstream
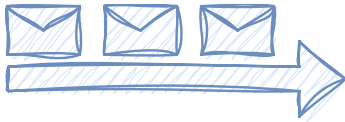
dapr building blocks

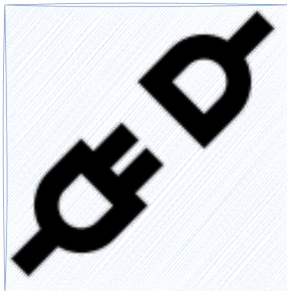Service-to-Service Invocation

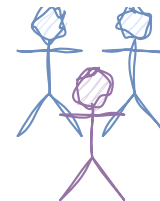State Management

Publish and Subscribe

Configuration & Secrets

Input and Output Bindings

Actors

dapr configuration

- An implementation of a **Building Block** is a *Component*
- **Components** are configured with *Component Specs*

```yaml
1  apiVersion: dapr.io/v1alpha1
2  kind: Component
3  metadata:
4    name: statestore
5  spec:
6    type: state.redis
7    version: v1
8    metadata:
9    - name: redisHost
10     value: localhost:6379
11   - name: redisPassword
12     value: ""
13   - name: actorStateStore
14     value: "true"
```

```yaml
1  apiVersion: dapr.io/v1alpha1
2  kind: Component
3  metadata:
4    name: pubsub
5  spec:
6    type: pubsub.redis
7    version: v1
8    metadata:
9    - name: redisHost
10     value: localhost:6379
11   - name: redisPassword
12     value: ""
```

https://github.com/dapr/components-contrib

dapr configuration secrets

```
 1  apiVersion: dapr.io/v1alpha1
 2  kind: Component
 3  metadata:
 4    name: secretstore-file
 5  spec:
 6    type: secretstores.local.file
 7    version: v1
 8    metadata:
 9    - name: secretsFile
10      value: /configuration/secrets.json
11    - name: namedSeparator
12      value: ":"
13    - name: multiValued
14      value: "false"
```

```
 1  apiVersion: dapr.io/v1alpha1
 2  kind: Component
 3  metadata:
 4    name: statestore
 5  auth:
 6    secretStore: secretstore-file
 7  spec:
 8    type: state.redis
 9    version: v1
10    metadata:
11    - name: redisHost
12      value: localhost:6379
13    - name: redisPassword
14      secretRef:
15        name: secretName
16        value: secretKey
17    - name: actorStateStore
18      value: "true"
```

http://localhost:3500/v1.0/secrets/<secretstore>/<secretname>

dapr service invocation

http://localhost:3500/v1.0/invoke/<service>/method/<path>
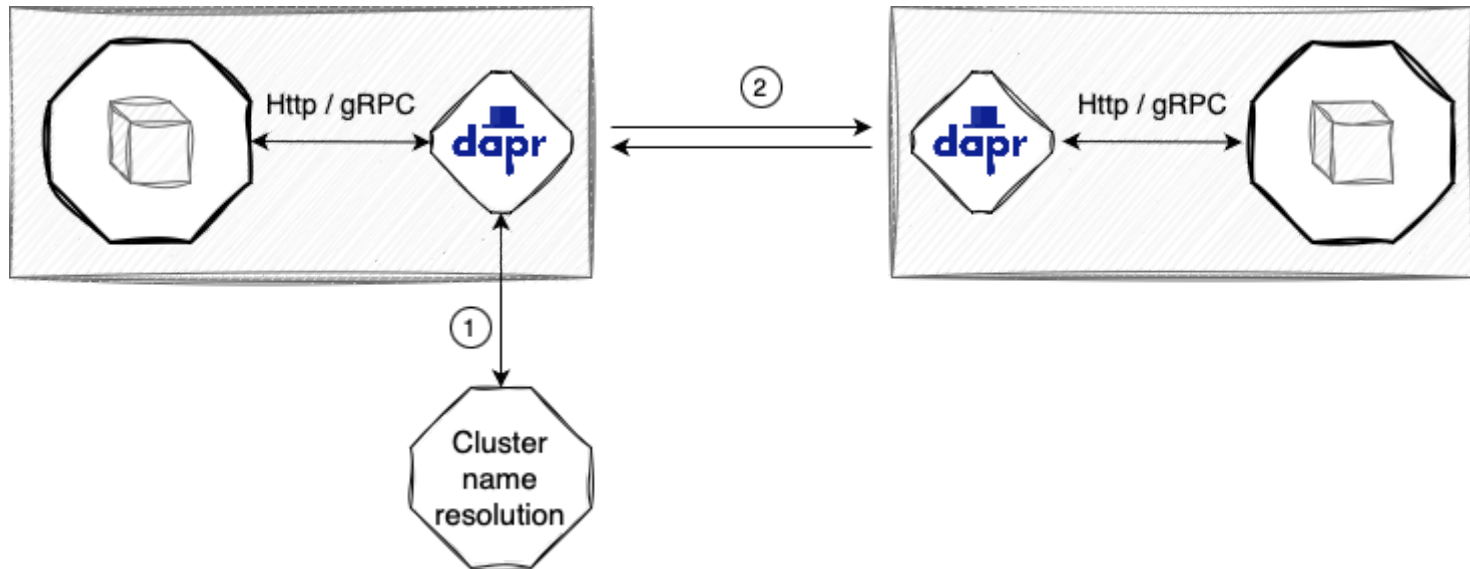
- Service discovery
- HTTP or gRPC
- Supports retries and circuit breaker patterns

CLOUD NATIVE
COMPUTING FOUNDATION

Retry & Circuit Breaker Pattern

- Retry Pattern - Retry an operation with the expectation that it will succeed
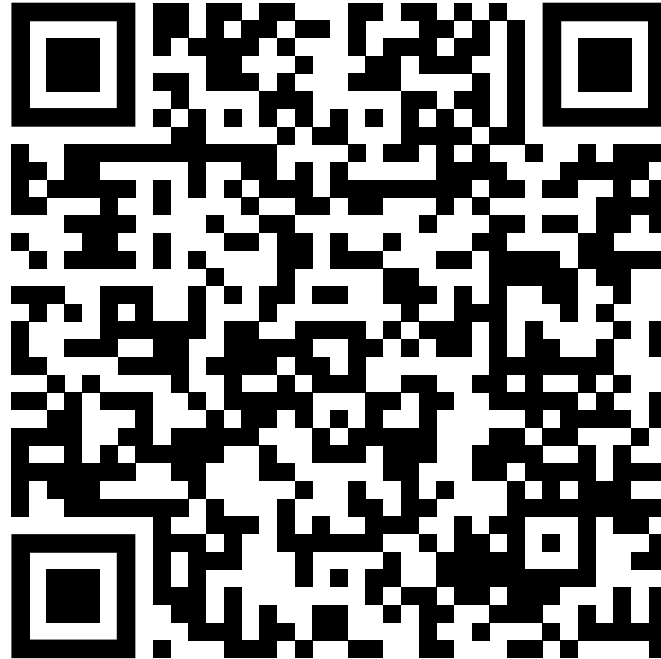- Circuit Breaker Pattern - Prevent an operation that is likely to fail

Circuit Closed

① 502

② 429

③ 408

Circuit Open
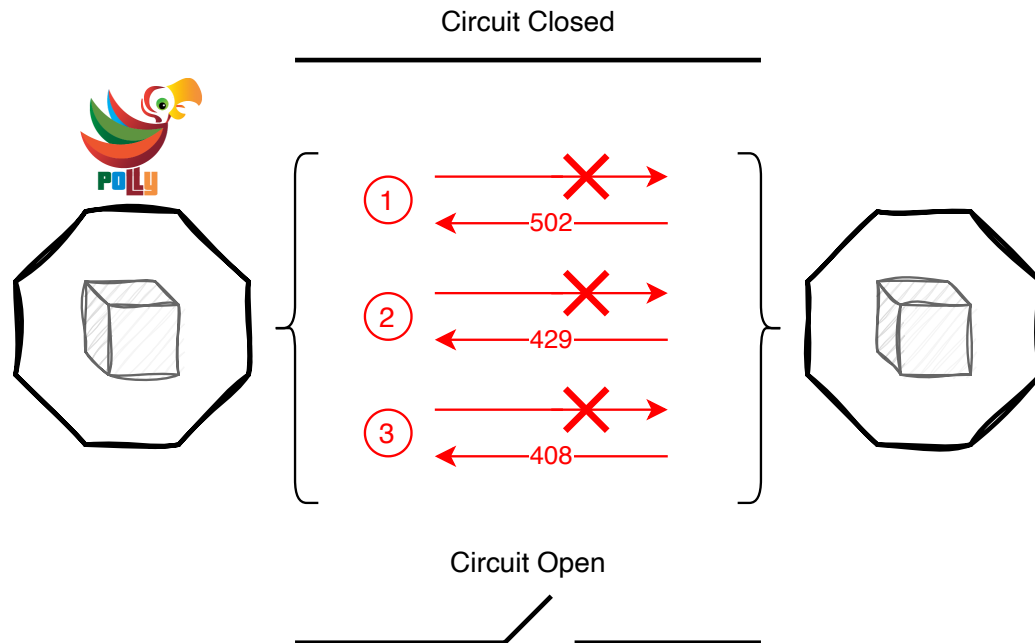
service invocation resiliency

```yaml
1  apiVersion: dapr.io/v1alpha1
2  kind: Resiliency
3  metadata:
4    name: generalresiliencypolicy
5  version: v1alpha1
6  scopes:
7    - appA
8    - appB
9  spec:
10   policies:
11     timeouts:
12       general: 10s
13       longResponse: 60s
14     retries:
15       general:
16         policy: constant
17         duration: 5s
18         maxRetries: 10
19     circuitBreakers:
20       serviceCB:
21         maxRequests: 1
22         timeout: 30s
23         trip: consecutiveFailures >= 5
24 targets:
25   apps:
26     appA:
27       timeout: general
28       retry: general
29       circuitBreaker: serviceCB
```
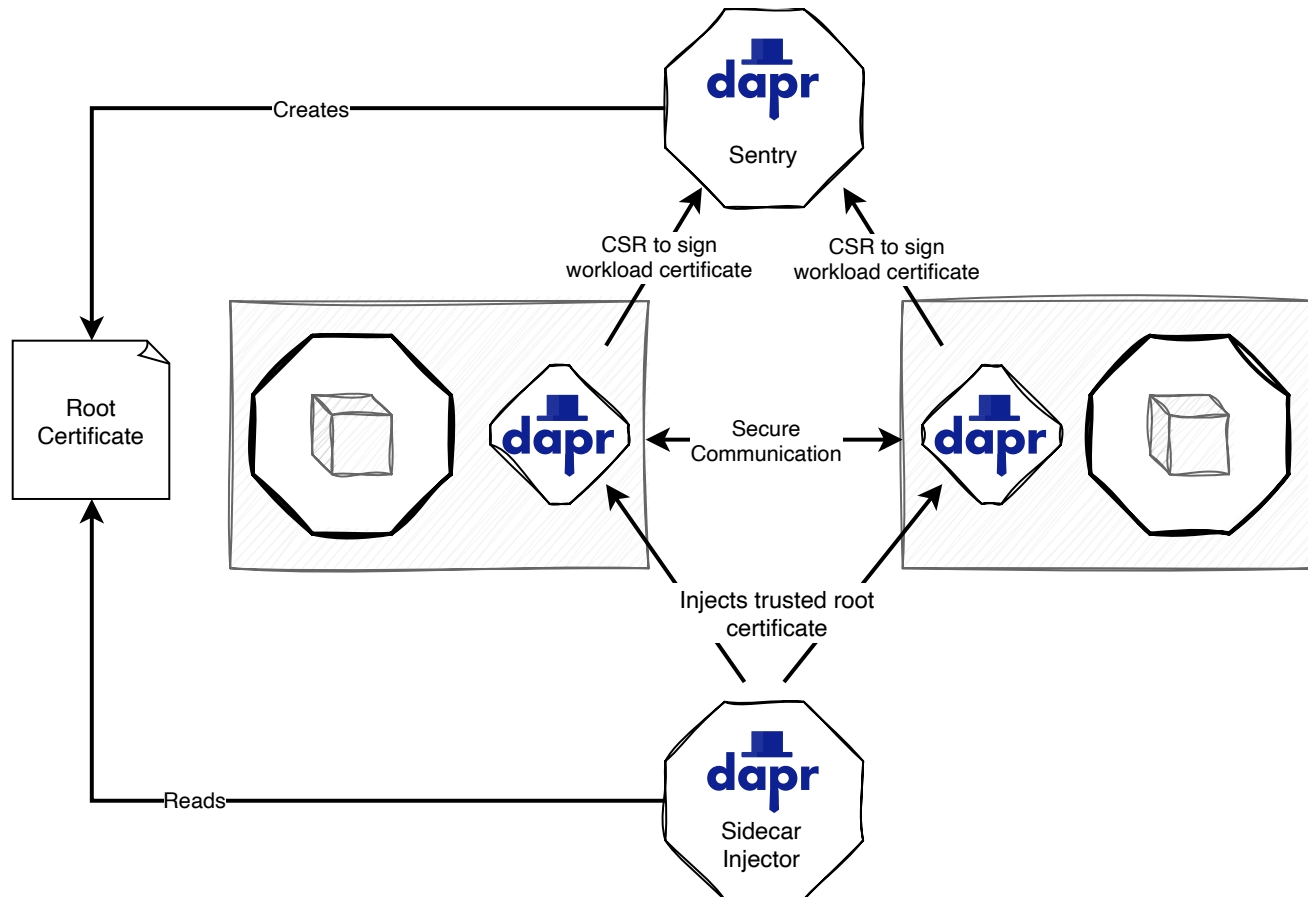
dapr service invocation

https://github.com/MattSheehanDev/SimplifyingMicroservicesWithDapr/tree/main/01.service-invocation

dapr mTLS

- Certificates to prove both the client and server identity
- Dapr Sentry acts as a Certificate Authority
- Dapr Sidecar Injector adds CA cert as trusted root cert

dapr state management

http://localhost:3500/v1.0/state/<storename>/<key>

Http / gRPC

./components

metadata:
name: statestore
spec:
type: state.redis
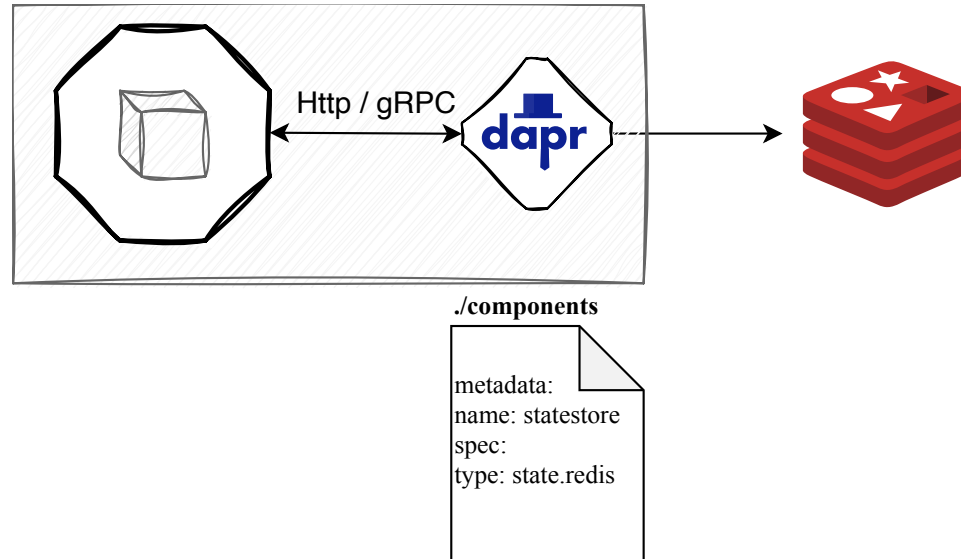
- Consistency
  - Strong vs Eventual
- Concurrency
  - Last-write wins *Default*
  - First-write wins *Optimistic Concurrency Control*

Optimistic Concurrency Control
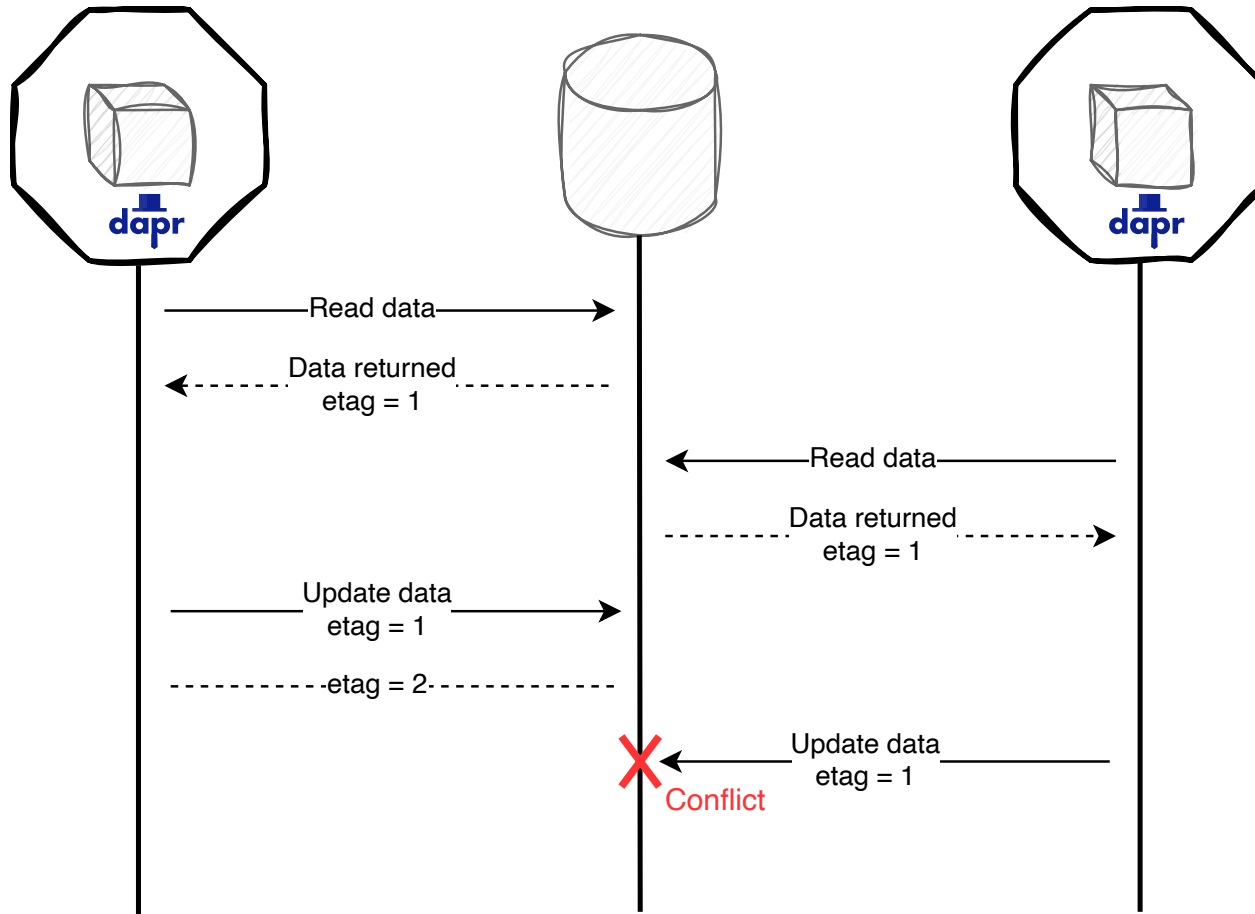
state management **resiliency**

```yaml
 1  apiVersion: dapr.io/v1alpha1
 2  kind: Resiliency
 3  metadata:
 4    name: statestoreresiliencypolicy
 5  version: v1alpha1
 6  scopes:
 7    - appA
 8  spec:
 9    policies:
10      timeouts:
11        general: 5s
12      retries:
13        retryForever:
14          policy: exponential
15          maxInterval: 15s
16          maxRetries: -1
17      circuitBreakers:
18        componentCB:
19          maxRequests: 1
20          timeout: 30s
21          trip: consecutiveFailures >= 5
22  targets:
23      components:
24        statestore:
25          outbound:
26            timeout: general
27            retry: retryForever
28            circuitBreaker: componentCB
```

17

dapr state management

https://github.com/MattSheehanDev/SimplifyingMicroservicesWithDapr/tree/main/02.state-management
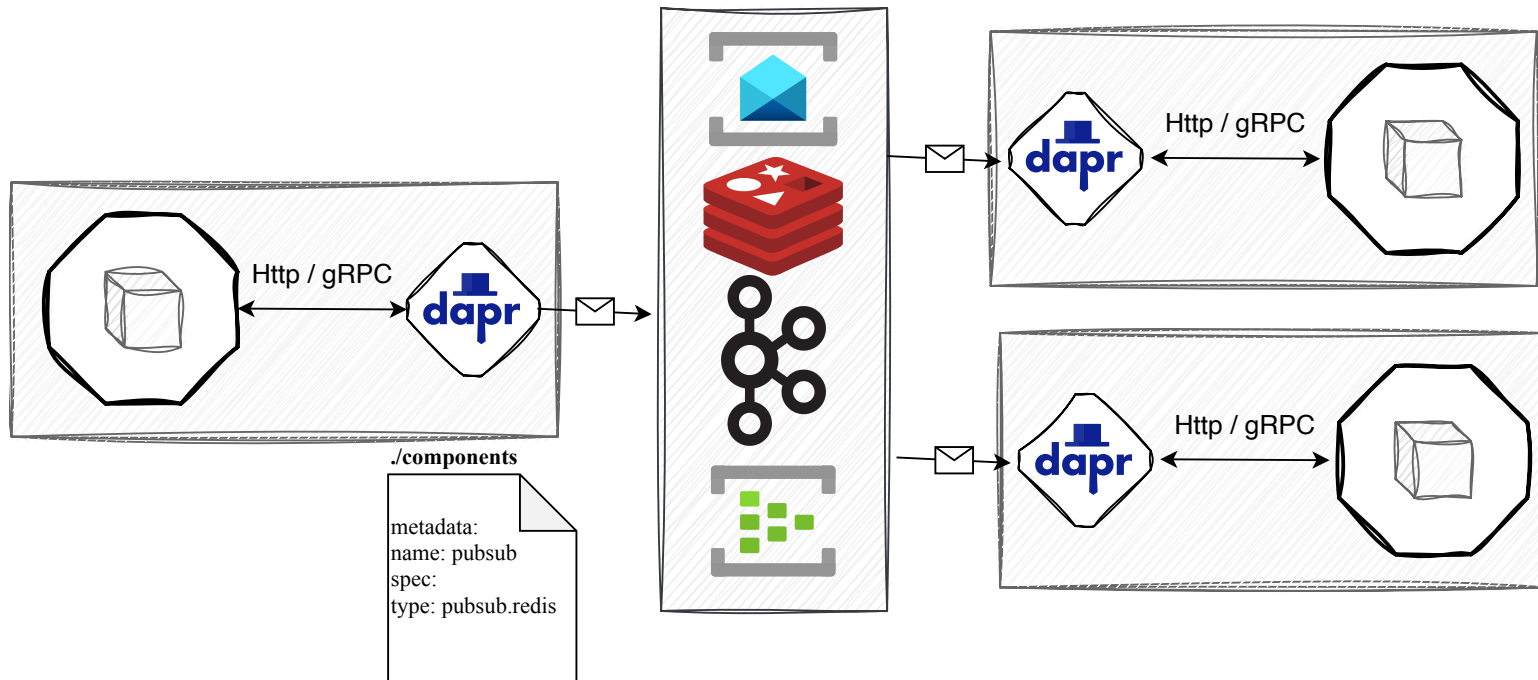
dapr publish/subscribe

`http://localhost:3500/v1.0/publish/<pubsubname>/<topic>`

./components

metadata:
name: pubsub
spec:
type: pubsub.redis

- At-least-once message delivery
- Content-based routing
- Cloud Events v1.0 spec

CLOUD NATIVE
COMPUTING FOUNDATION

dapr publish/subscribe

./components

metadata:
name: pubsub
spec:
type: pubsub.redis
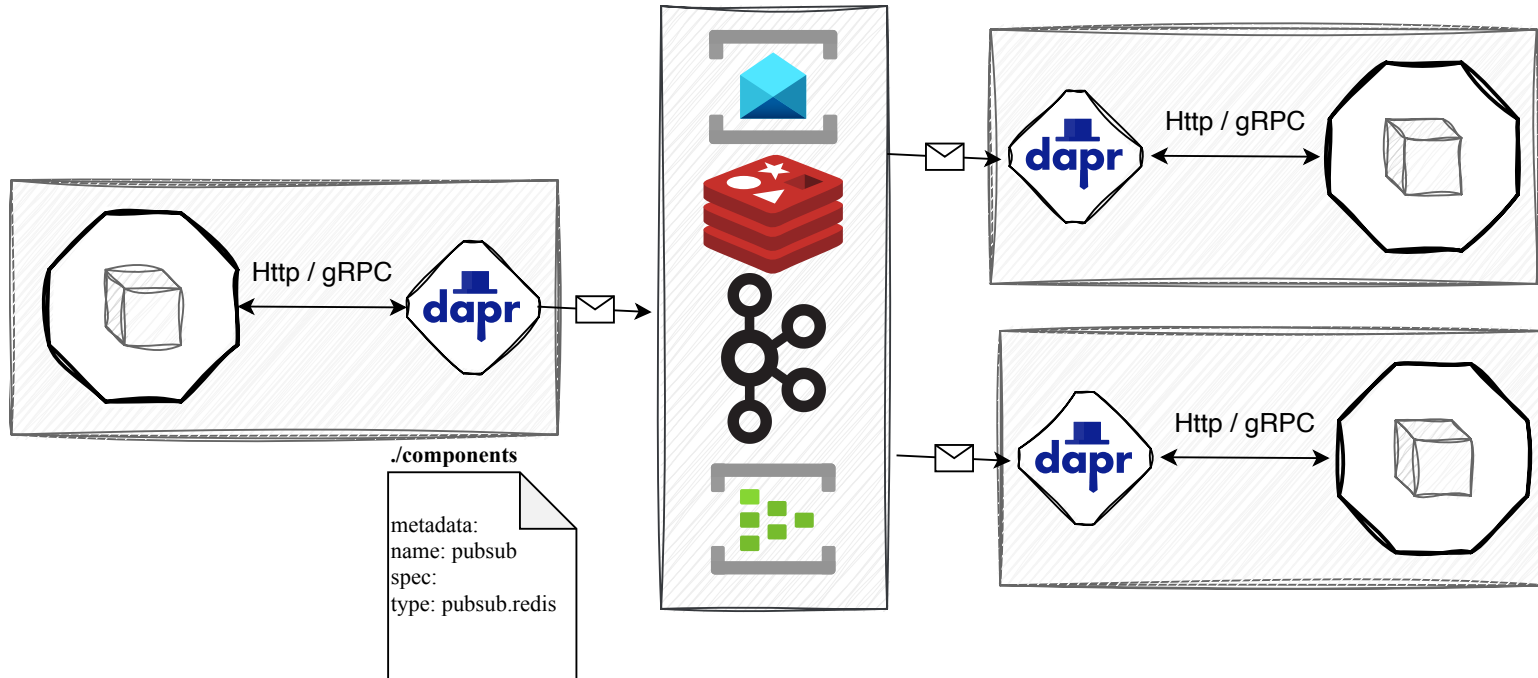
| 2xx	Message is processed
404	Message is dropped
5xx	Message is retried

Cloud Events

- Enables content-based routing
- Enables message de-duplication
- Improves interoperability

```json
{
  "id": "ade1a5ef-ba37-4a12-935f-6ca6c5c1254a",
  "source": "app_name",
  "specversion": "1.0",
  "type": "com.dapr.event.sent",
  "topic": "topic_name",
  "pubsubname": "pub_sub_name",
  "data": {
    // serialized message data
  },
  "datacontenttype": "application/json; charset=utf-8",
  "time": "2024-01-08T13:00:00Z",
  "traceid": "00-113ad9c4e42b27583ae98ba698d54255-e3743e35ff56f219-01",
  "tracestate": "",
  "traceparent": "00-113ad9c4e42b27583ae98ba698d54255-e3743e35ff56f219-01"
}
```

dapr resiliency

```yaml
 1 apiVersion: dapr.io/v1alpha1
 2 kind: Resiliency
 3 metadata:
 4   name: resiliencypolicy
 5 version: v1alpha1
 6 spec:
 7   policies:
 8     timeouts:
 9       general: 5s
10     retries:
11       constantRetry:
12         policy: constant
13         duration: 5s
14         maxRetries: 10
15     circuitBreakers:
16       pubsubCB:
17         maxRequests: 1
18         interval: 8s
19         timeout: 45s
20         trip: consecutiveFailures > 5
21 targets:
22     components:
23       pubsub:
24         outbound:
25           retry: constantRetry
26           circuitBreaker: pubsubCB
27         inbound:
28           timeout: general
29           retry: constantRetry
30           circuitBreaker: pubsubCB
```
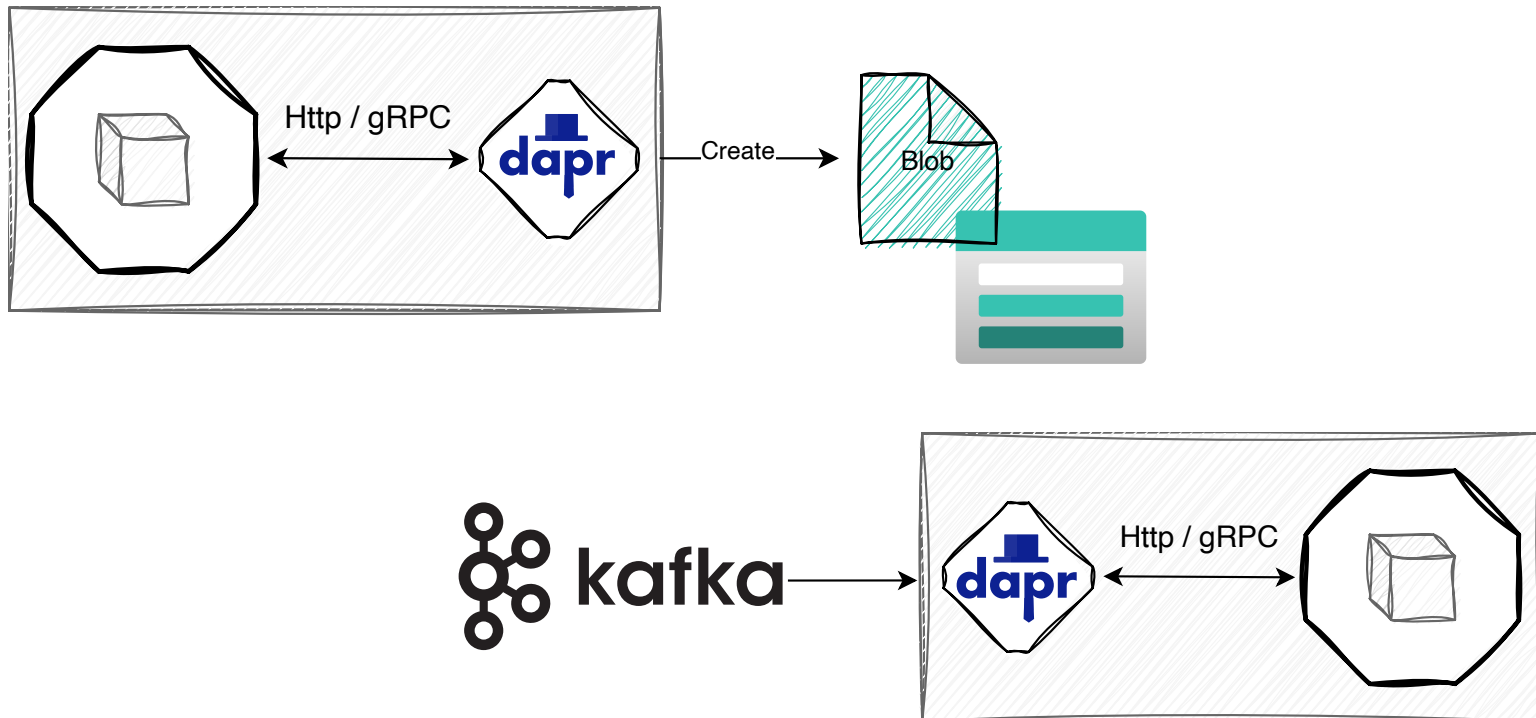
dapr publish/subscribe
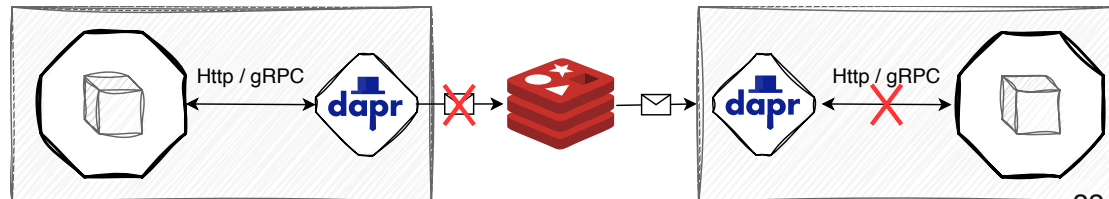
https://github.com/MattSheehanDev/SimplifyingMicroservicesWithDapr/tree/main/03.publish-subscribe

dapr bindings

- Outbound binding - invoke external resource
- Input binding - trigger based on an event
- Can depend on the runtime instead of a library

binding resiliency

```yaml
 1 apiVersion: dapr.io/v1alpha1
 2 kind: Resiliency
 3 metadata:
 4   name: resiliencypolicy
 5 version: v1alpha1
 6 spec:
 7   policies:
 8     timeouts:
 9       general: 5s
10     retries:
11       important:
12         policy: constant
13         duration: 5s
14         maxRetries: 30
15     circuitBreakers:
16       bindingCB:
17         maxRequests: 1
18         interval: 8s
19         timeout: 45s
20         trip: consecutiveFailures > 8
21 targets:
22     components:
23       blobStorageBinding:
24         outbound:
25           timeout: general
26           retry: important
27           circuitBreaker: bindingCB
28         inbound:
29           timeout: general
30           retry: important
```
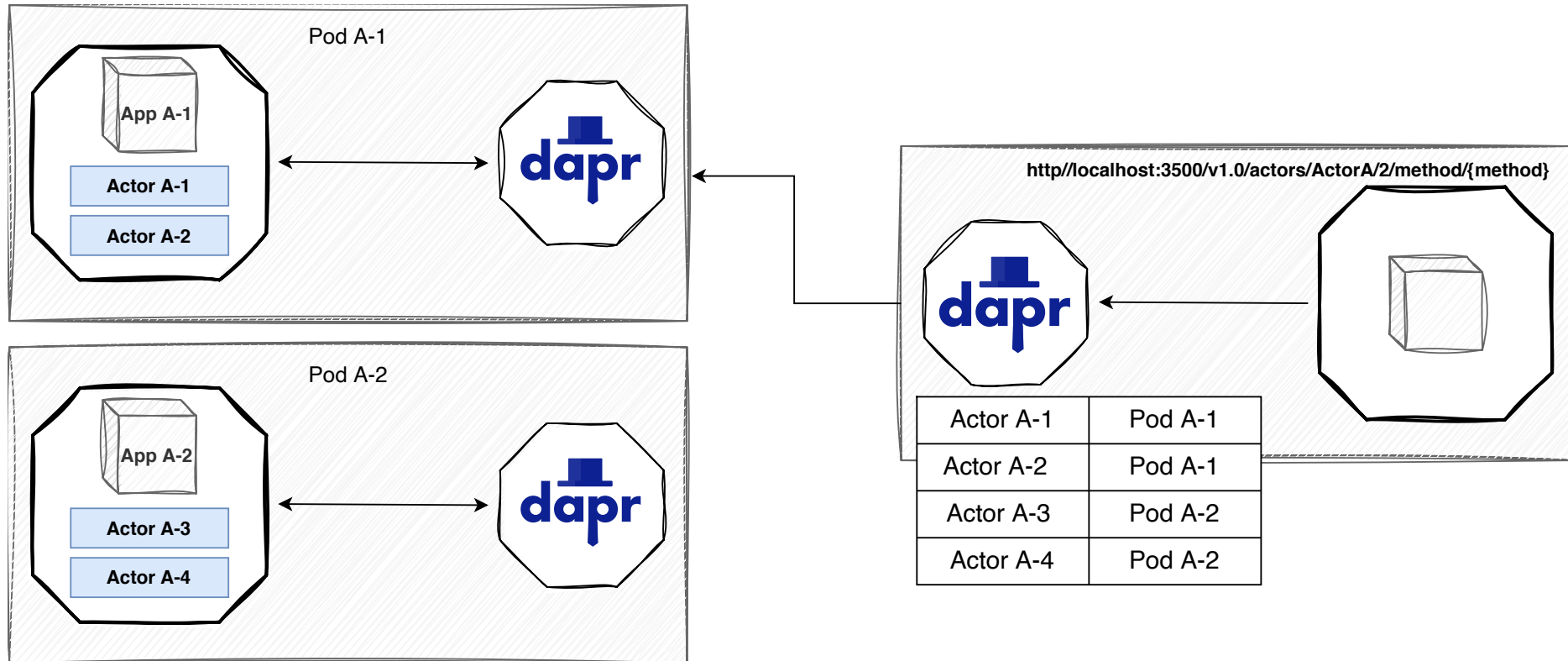
dapr bindings

https://github.com/MattSheehanDev/SimplifyingMicroservicesWithDapr/tree/main/04.bindings

dapr actors

- Manage concurrency
  - Turn-based concurrency access **per-actor**
- External state store
  - Require state store that supports transactions and etags
- Managed lifetimes
  - Timers & Reminders
- Placement and routing
  - Dapr placement service broadcasts current actors to all sidecars

dapr actors

Pod A-1

App A-1

Actor A-1

Actor A-2

Pod A-2

App A-2

Actor A-3

Actor A-4

http//localhost:3500/v1.0/actors/ActorA/2/method/{method}

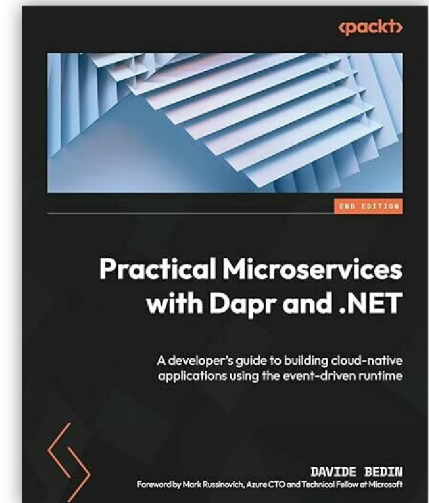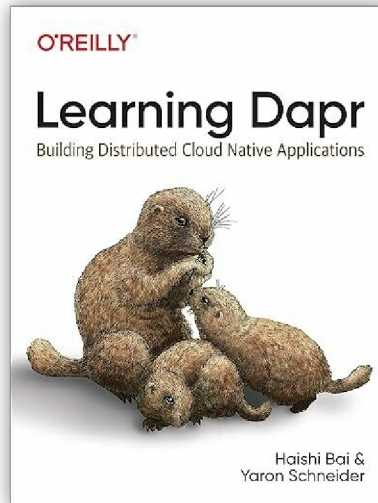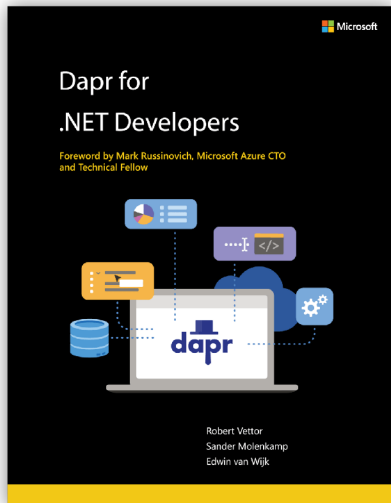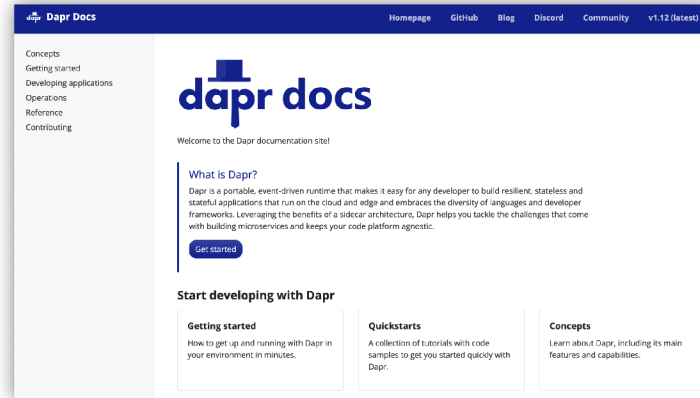| Actor A-1	Pod A-1
Actor A-2	Pod A-1
Actor A-3	Pod A-2
Actor A-4	Pod A-2

actor resiliency

```yaml
 1  apiVersion: dapr.io/v1alpha1
 2  kind: Resiliency
 3  metadata:
 4    name: resiliencypolicy
 5  version: v1alpha1
 6  scopes:
 7    - appB
 8  spec:
 9    policies:
10      timeouts:
11        general: 5s
12      retries:
13        retry10:
14          policy: exponential
15          maxInterval: 15s
16          maxRetries: 10
17      circuitBreakers:
18        serviceCB:
19          maxRequests: 1
20          timeout: 30s
21          trip: consecutiveFailures >= 5
22  targets:
23      actors:
24        actorA:
25          timeout: general
26          retry: retry10
27          circuitBreaker: serviceCB
```

29

dapr actors

https://github.com/MattSheehanDev/SimplifyingMicroservicesWithDapr/tree/main/04.actors

Further Reading

https://github.com/dapr/samples

https://github.com/dapr/quickstarts