Lights, camera, action! Building distributed applications with Dapr Actors

@marcduiker

Marc Duiker

Sr Dev Advocate

Diagrid

Azure MVP

Dapr Community Manager

❤️ pixel art

@marcduiker

≠

dapr

@marcduiker

Distributed
application
runtime

@marcduiker

Dapr

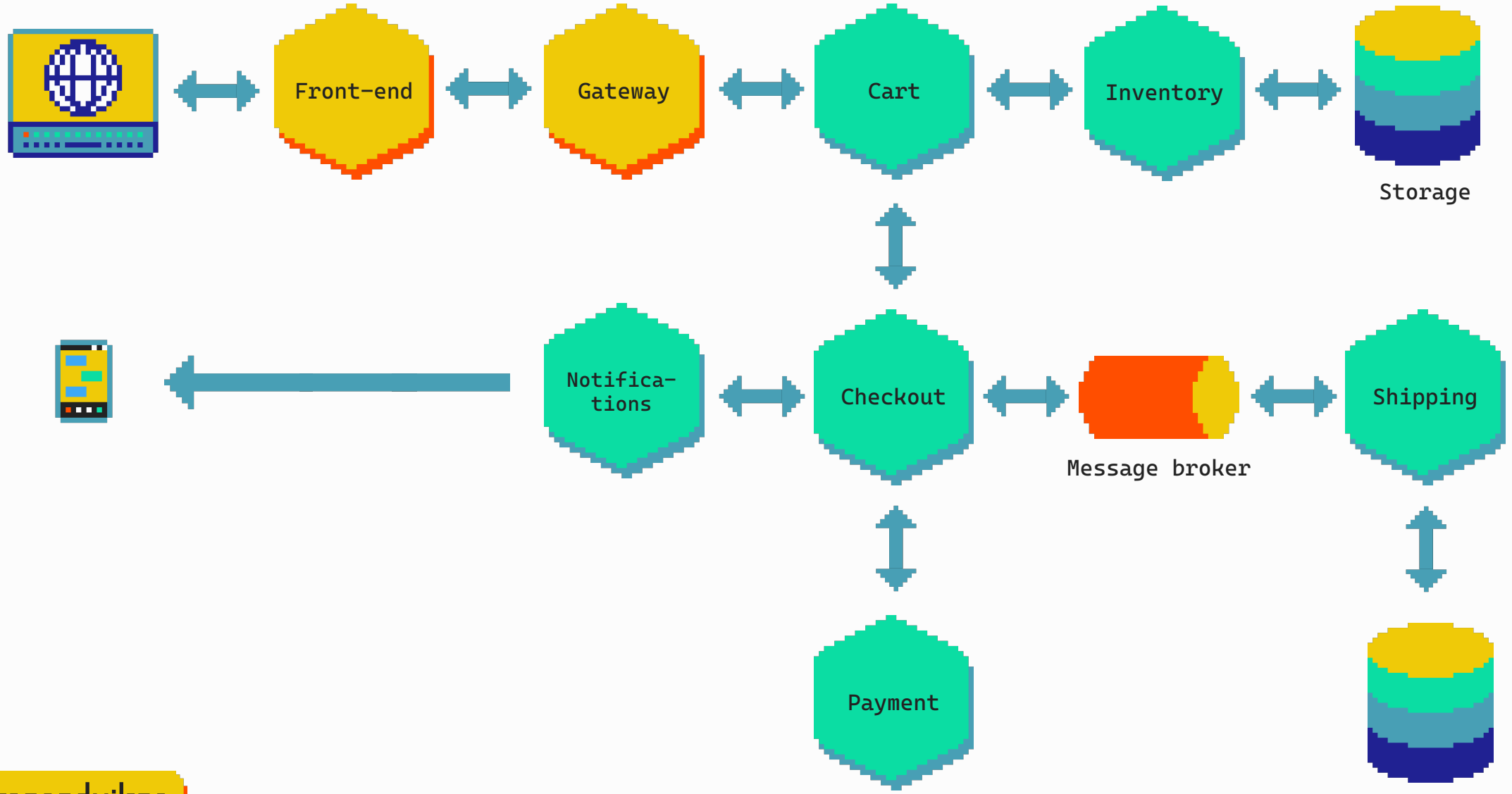
Actor Model

Dapr Actor API

Demos!

Dapr supporter badge

Distributed apps

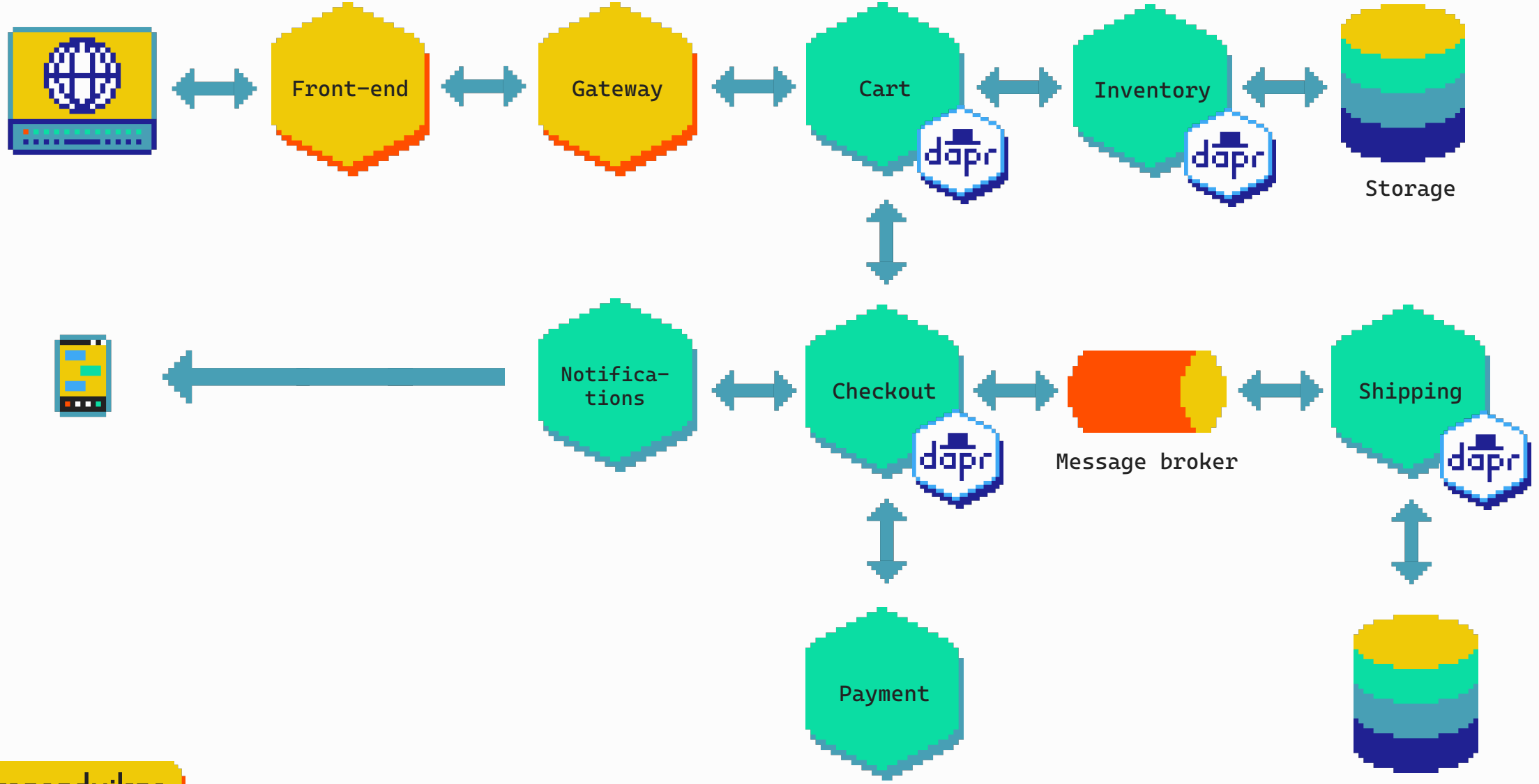


@marcduiker

6

dapr

Distributed apps with Dapr

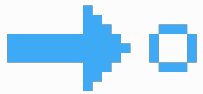


@marcduiker

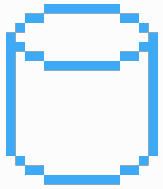
Built-in security,
resiliency and observability
capabilities.

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

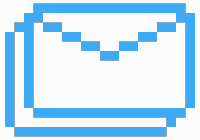
Dapr APIs



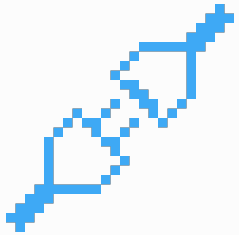
Service invocation



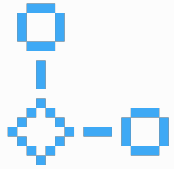
State Management



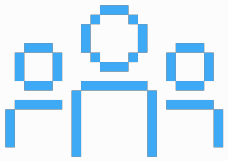
Publish & subscribe



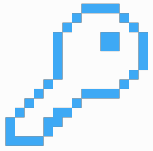
Bindings (input & output)



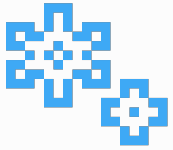
Workflow



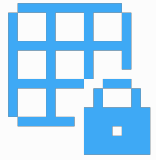
Actors



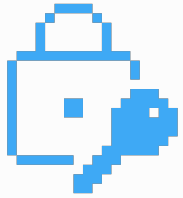
Secret Stores



External Configuration



Distributed Lock

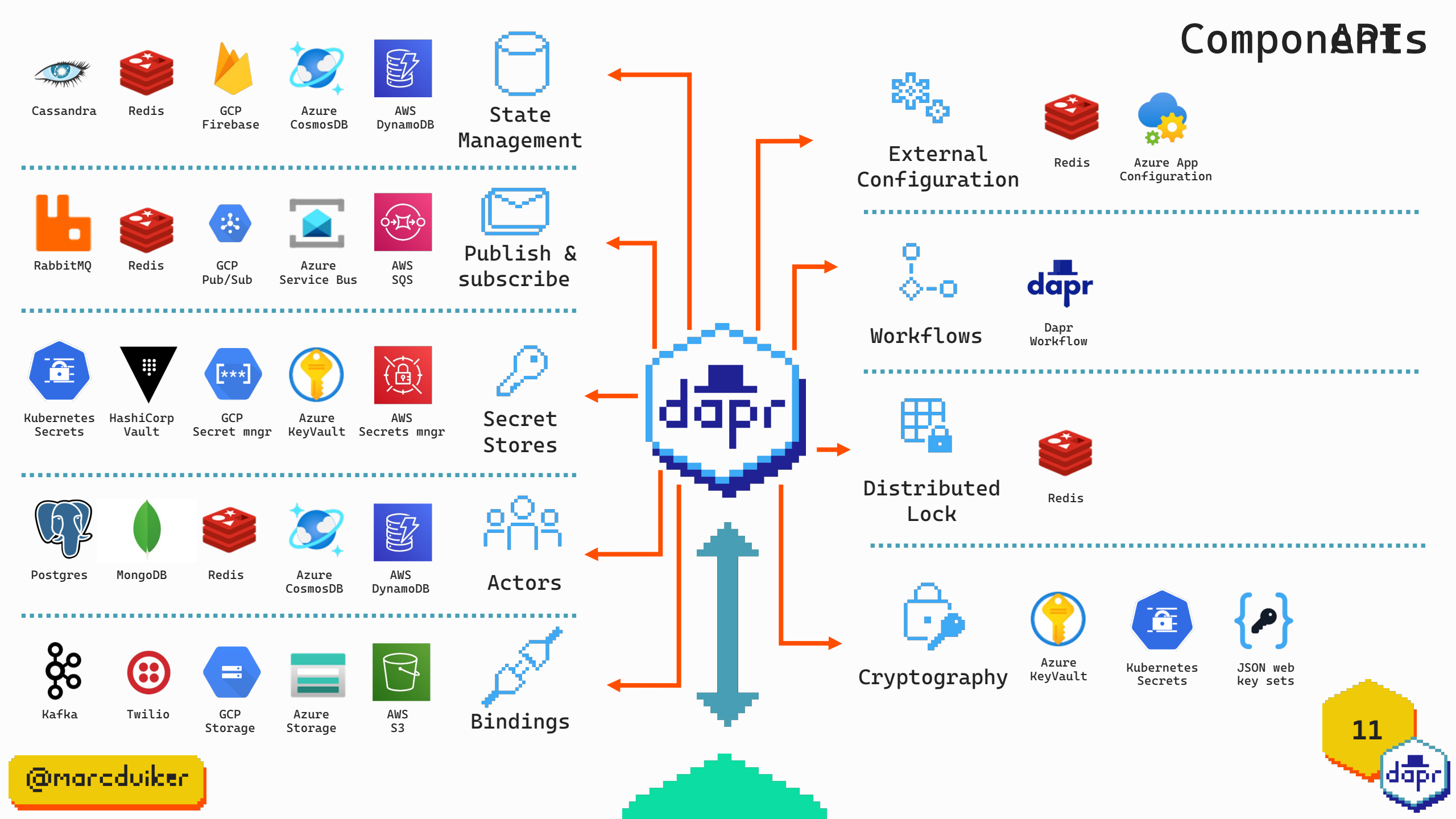


Cryptography

@marcduiker

dapr

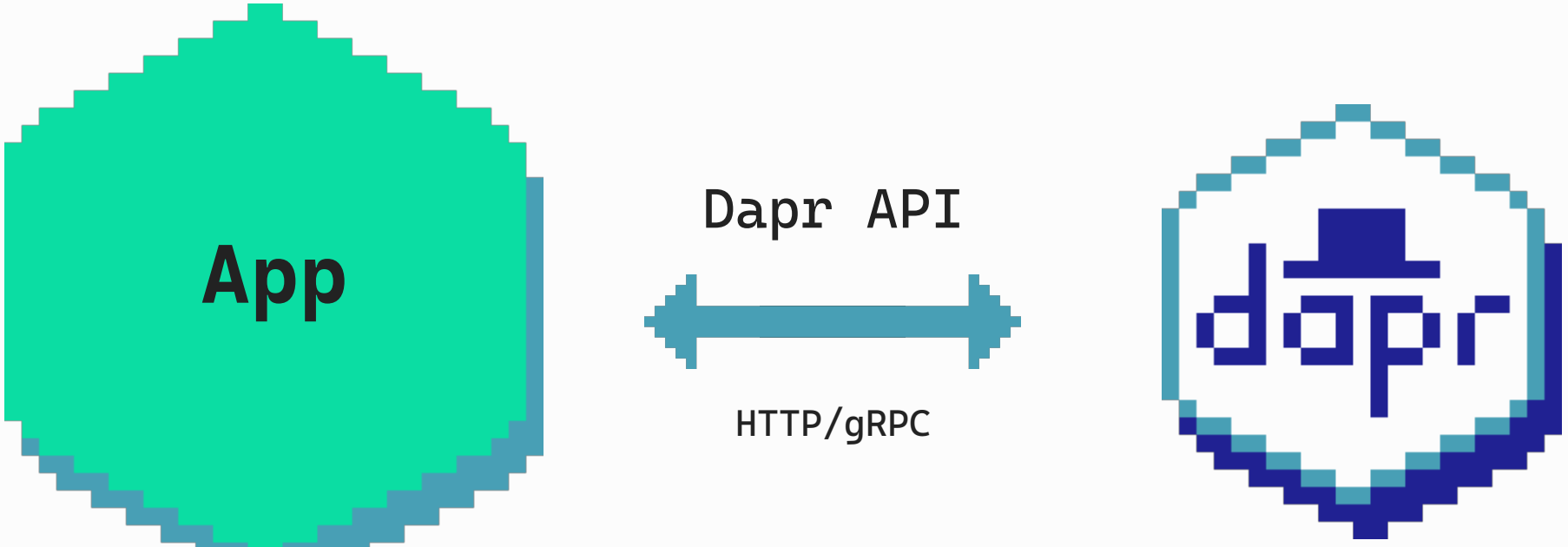
ComponAPIs

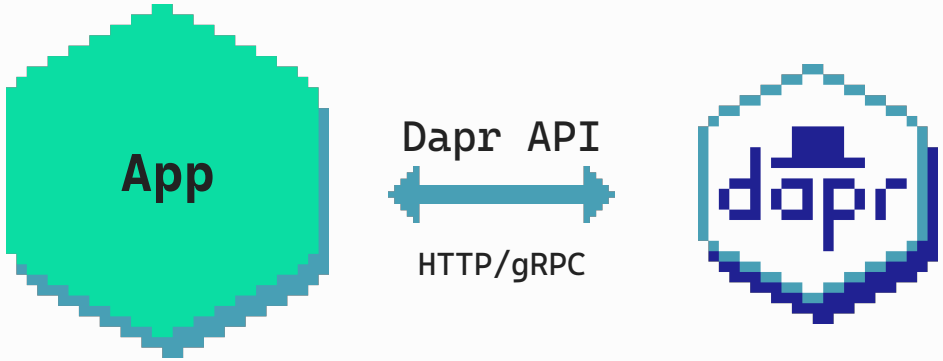


@marcduiker

Azure Container Apps

virtual or
physical machines


App
Dapr API
HTTP/gRPC
Application
Dapr sidecar




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

GET http://localhost:3500/v1.0/state/inventory/item50

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

GET http://localhost:3500/v1.0/secrets/vault/password42

POST http://localhost:3500/v1.0/actors/MyActor/A/method/Update
```

Actor Model

A model of concurrent computation where the actor is the basic building block.

A Universal Modular Actor Formalism for Artificial Intelligence (1973)

Carl Hewitt, Peter Bishop & Richard Steiger

www.ijcai.org/Proceedings/73/Papers/027B.pdf

en.wikipedia.org/wiki/Actor_model

Actor = a unit of computation

With these capabilities:

- processing
- storage
- communication

One actor is no actor

Actor has

- identity
- behavior
- state

When to use the Actor Model?

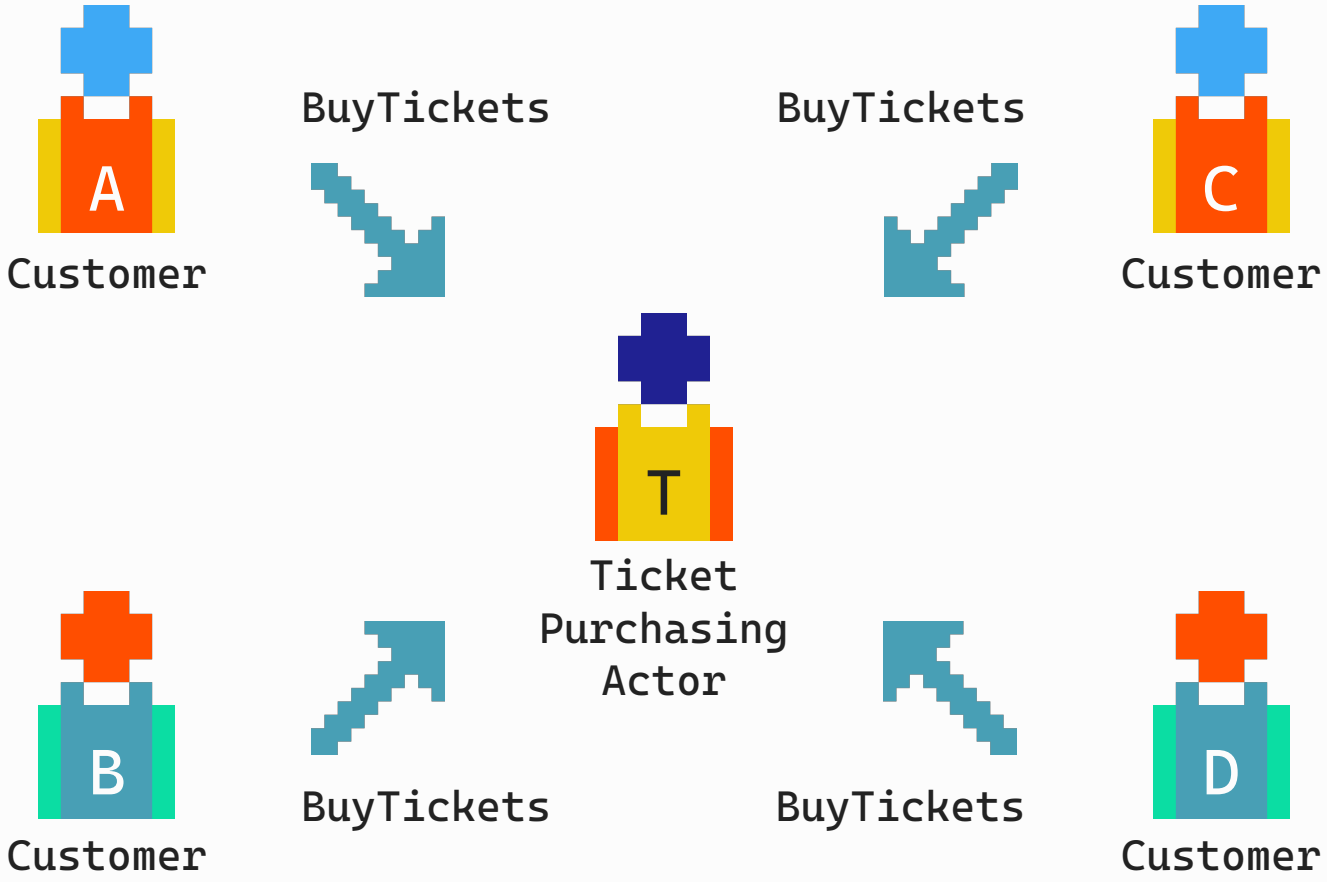
- Your problem space involves many small and independent units of state and logic.
- You need to handle concurrency and processing speed is important.
- Examples: gaming, simulations, trading systems, transaction processing, IoT

Actor Model vs Workflow

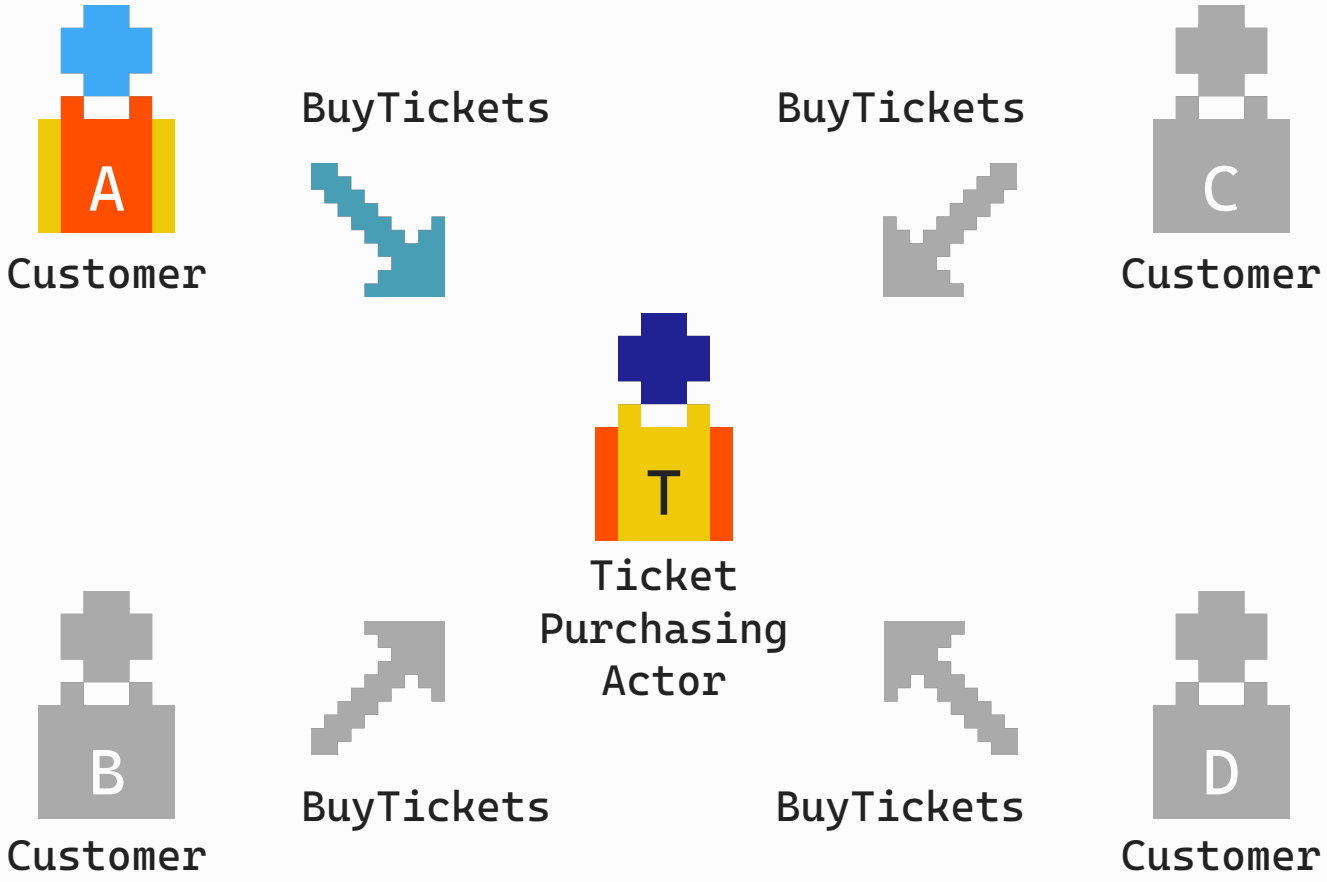
Actors → Processing needs to be quick

Workflow → Processing can take a long time

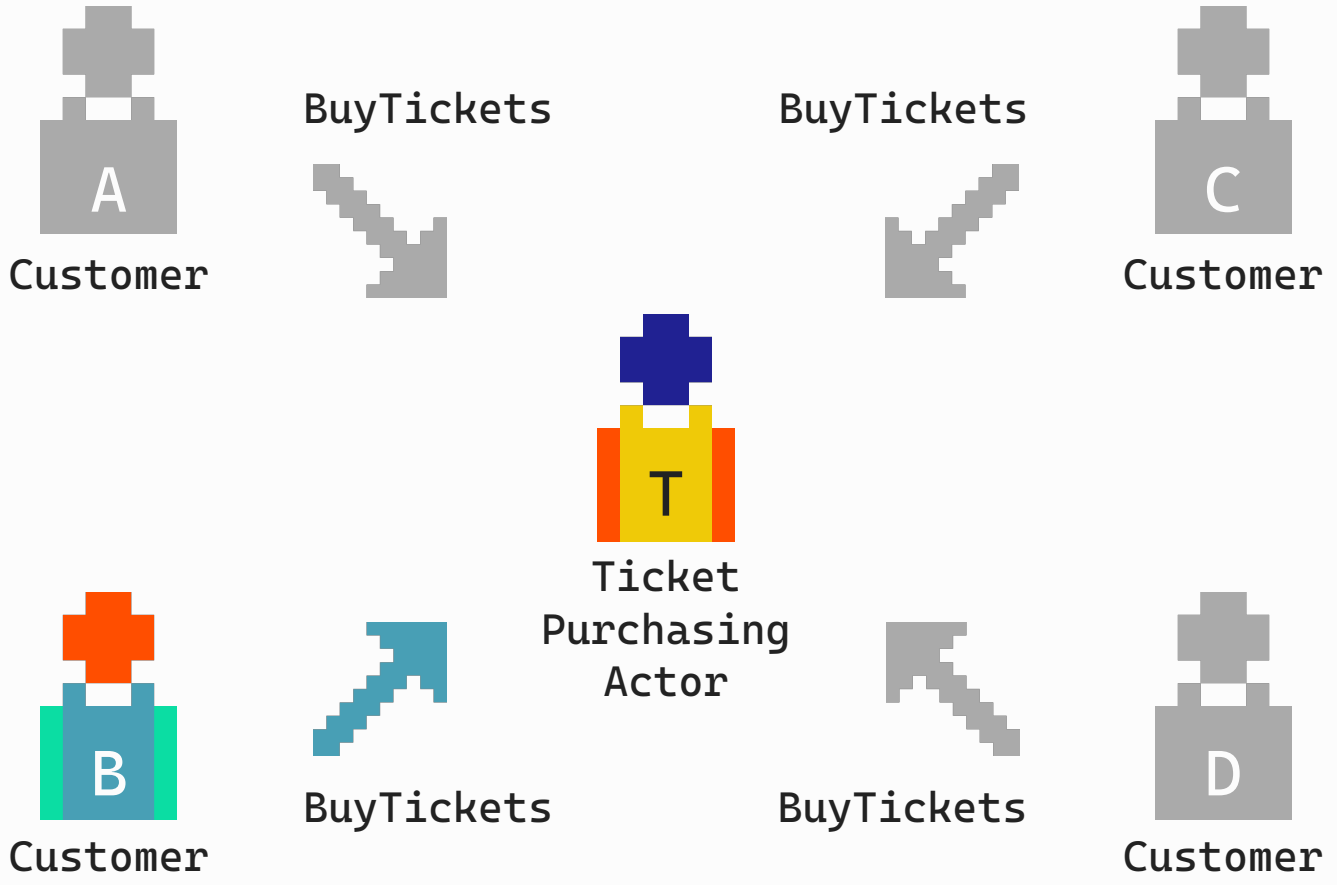
Turn based concurrent systems



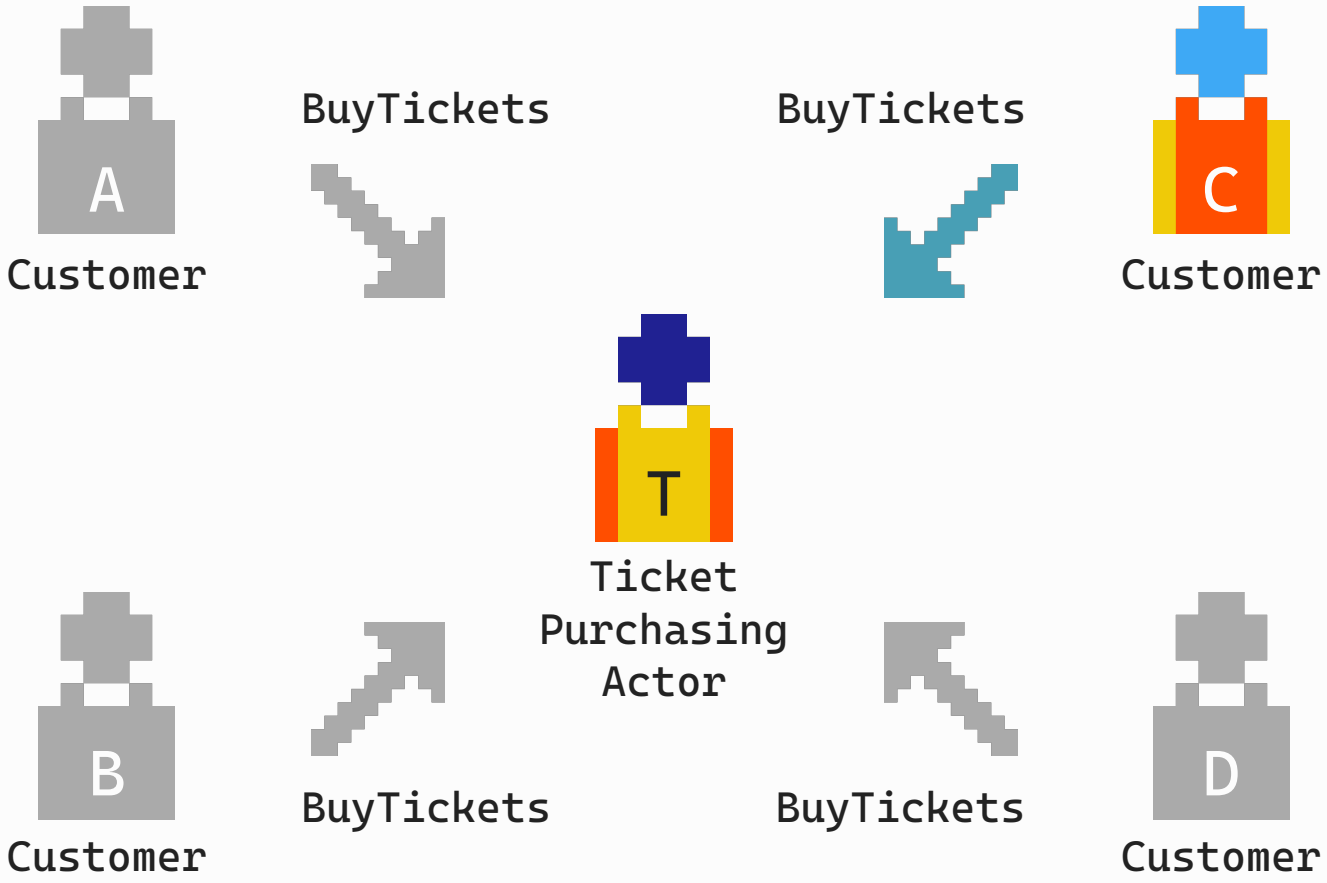
Turn based concurrent systems



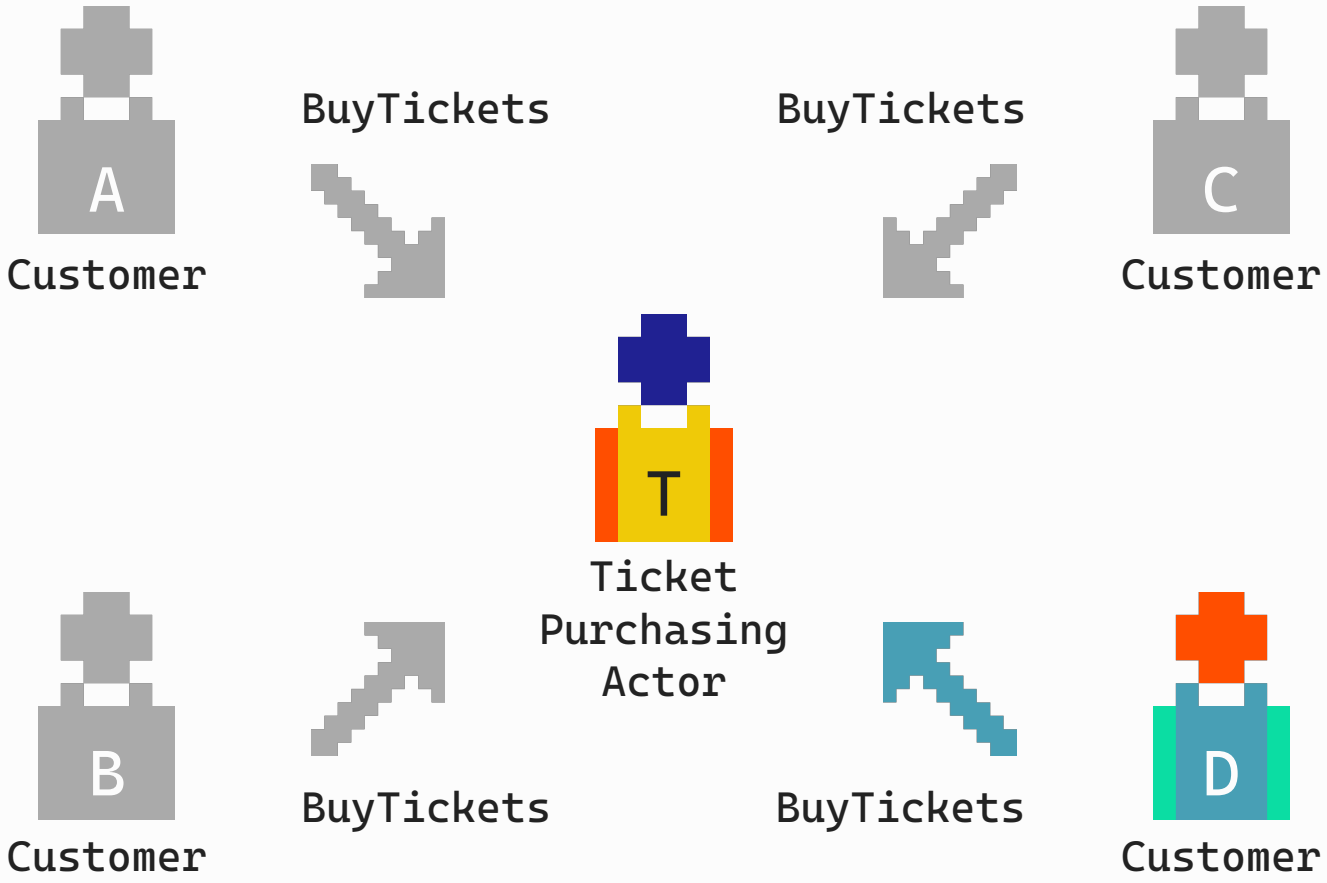
Turn based concurrent systems



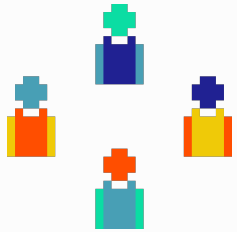
Turn based concurrent systems



Turn based concurrent systems



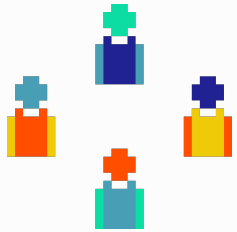
Dapr Actors



Actors

Virtual actor model

Actor lifetime is not tied to their in-memory representation. No need to explicitly create or destroy an actor.



Actors

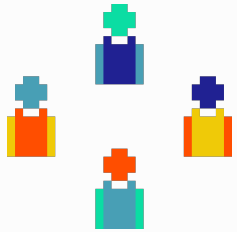
Orleans 2014

Service Fabric Reliable Actors 2016

Dapr Actors 2019

@marcduiker

dapr



Actors

Dapr Actors can be written in:

- C#
- Java
- JavaScript
- Python
- Go
- Rust (alpha)

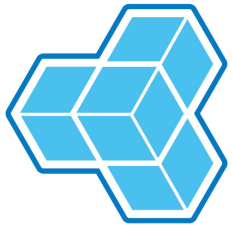
Interact with Dapr Actors using any language!

Dapr Actor users

Schréder
Experts in lightability™

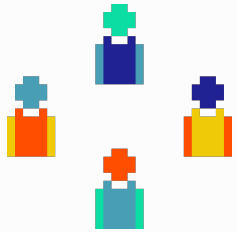
IGNITION
GROUP

ZEISS

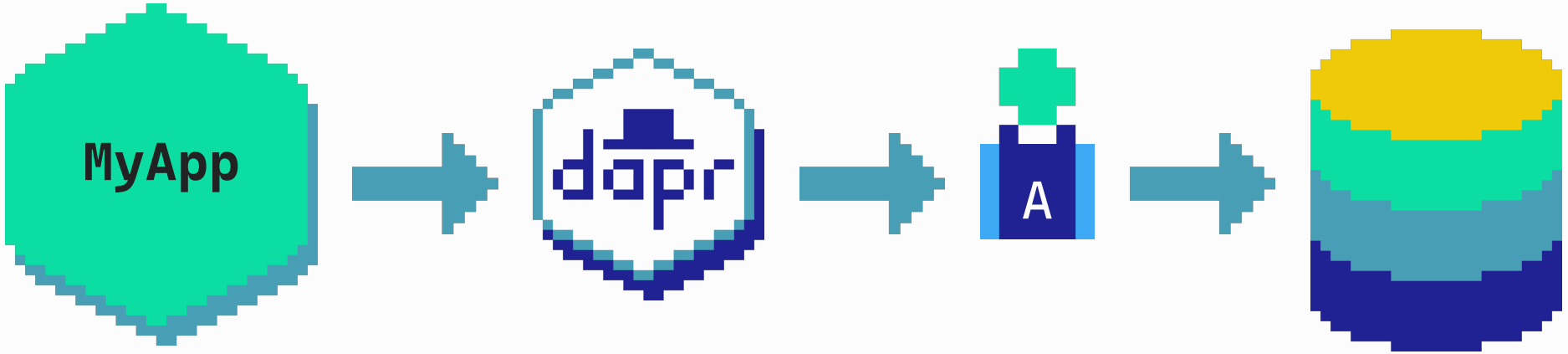


DERIVCO

https://headleysj.medium.com/building-event-driven-systems-at-scale-in-kubernetes-with-dapr-part-iii-what-does-at-scale-7c15dfa64338



Actors



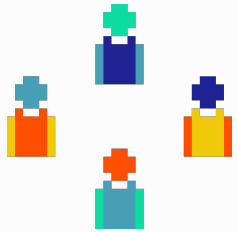
Actor definitions + Actor client code

Runtime

Actor instance

State

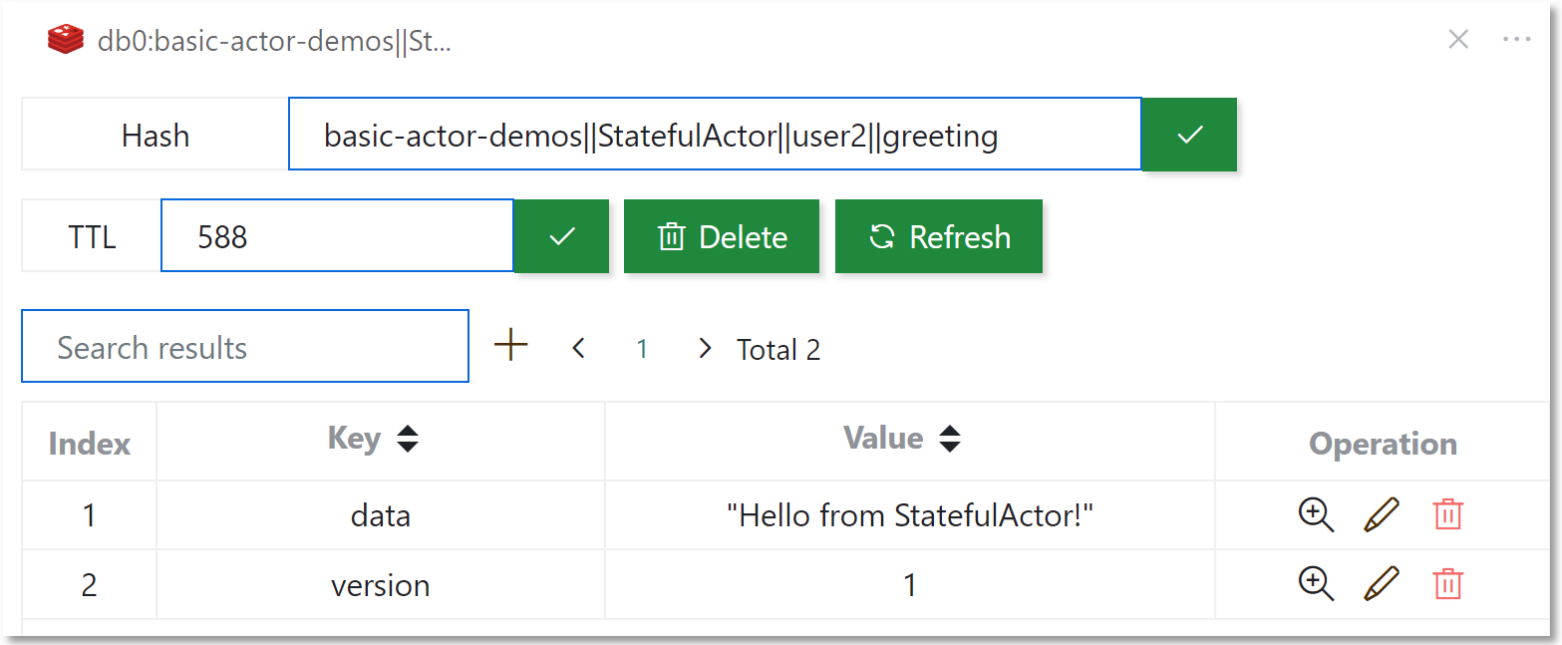
@marcduiker

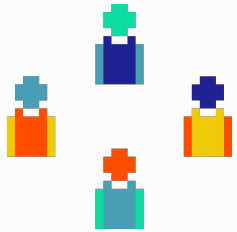


Actors

State management (key/value)

Combined key = AppID||ActorType||ActorID||key

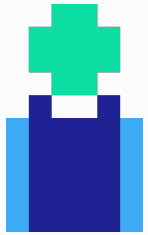




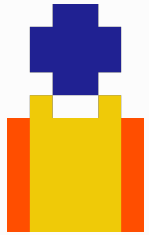
Actors

Timers & Reminders

Actor can schedule periodic work on itself.

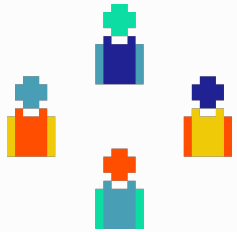


Timers are stateless
(lost after actor deactivation)

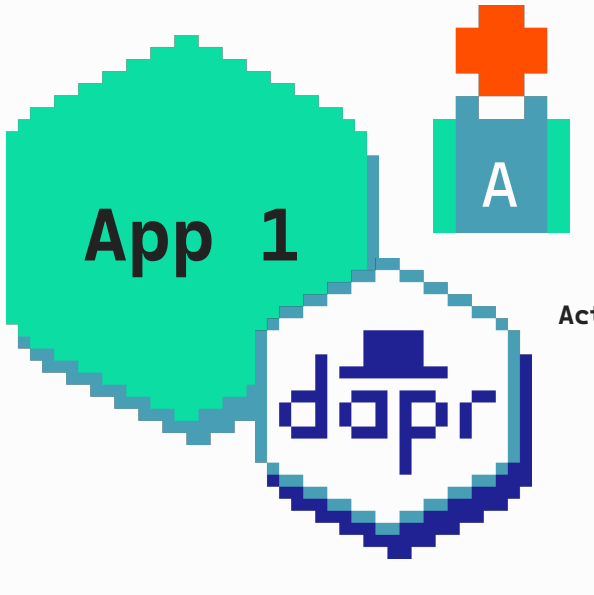


Reminders are stateful
(persists after deactivation)

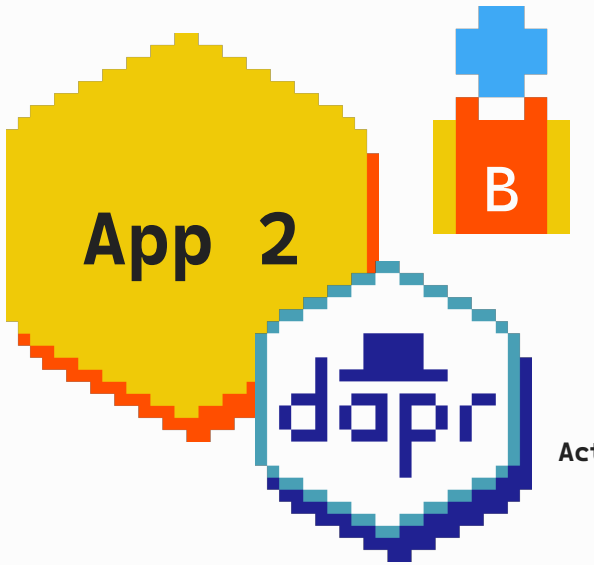
Placement



Actors

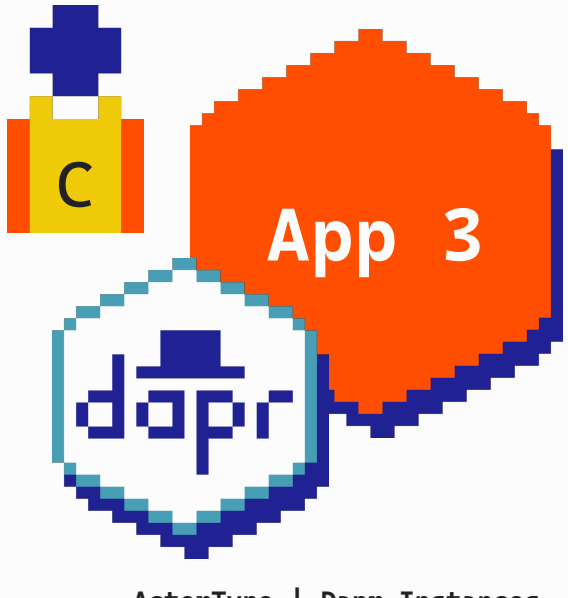


ActorType	Dapr Instances
A	partition 1
B	partition 2
C	partition 3



ActorType	Dapr Instances
A	partition 1
B	partition 2
C	partition 3

Placement service



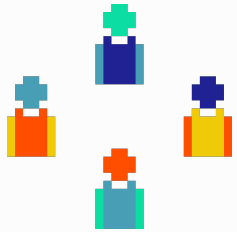
ActorType	Dapr Instances
A	partition 1
B	partition 2
C	partition 3

ActorType	Dapr Instances
A	partition 1
B	partition 2
C	partition 3

@marcduiker

Dapr Actors API

https://docs.dapr.io/reference/api/actors_api/



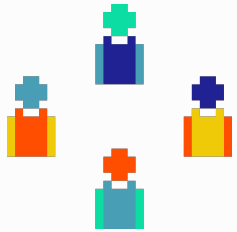
Actors

Invoke a method

```
POST http://localhost:3500/v1.0/actors/MyActor/A/method/SayHelloWorld
```

```
POST http://localhost:3500 /v1.0/actors/MyActor/A/method/SayHello
Content-Type: application/json

"Rene"
```



Actors

Set/get state

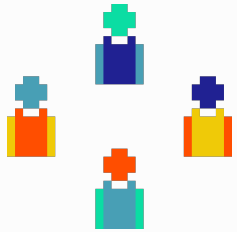
```
POST http://localhost:3500/v1.0/actors/MyActor/A/state
Content-Type: application/json

[
    {
        "operation": "upsert",
        "request": {
            "key": "greeting",
            "value": "Hello World!"
        }
    }
]
```

```
GET http://localhost:3500 /v1.0/actors/MyActor/A/state/greeting
```

@marcduiker

dapr



Actors

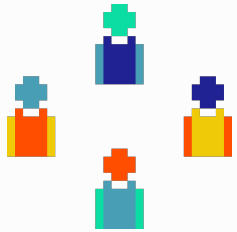
Set a reminder

```
POST http://localhost:3500/v1.0/actors/MyActor/A/reminders/snooze
Content-Type: application/json

{
    "dueTime" : "0h10m0s0ms",
    "period" : "R3/P0Y0M0W0DT0H0M30S"
}
```


Dapr Actors .NET SDK

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

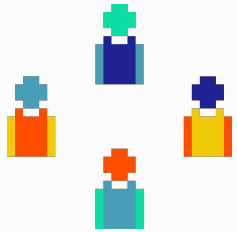


Actors

Actor Definition

```
public interface IHelloWorld : IActor
{
    Task<string> SayHelloWorld();

    Task<string> SayHello(string name);
}
```



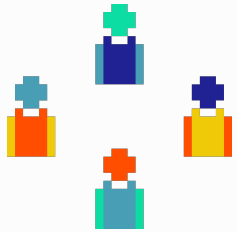
Actors

Actor Definition

```
public class HelloWorldActor : Actor, IHelloWorld
{
    public HelloWorldActor(ActorHost host) : base(host)
    {
    }

    public Task<string> SayHelloWorld()
    {
        return Task.FromResult("Hello World!");
    }

    public Task<string> SayHello(string name)
    {
        return Task.FromResult($"Hello {name}!");
    }
}
```

Actors

Using a strongly typed client

```
var helloWorldProxy = ProxyFactory.CreateActorProxy<IHelloWorld>(
    new ActorId(“helloworld1”),
    nameof(HelloWorldActor));
```

```
var result = await helloWorldProxy.SayHelloWorld();
```

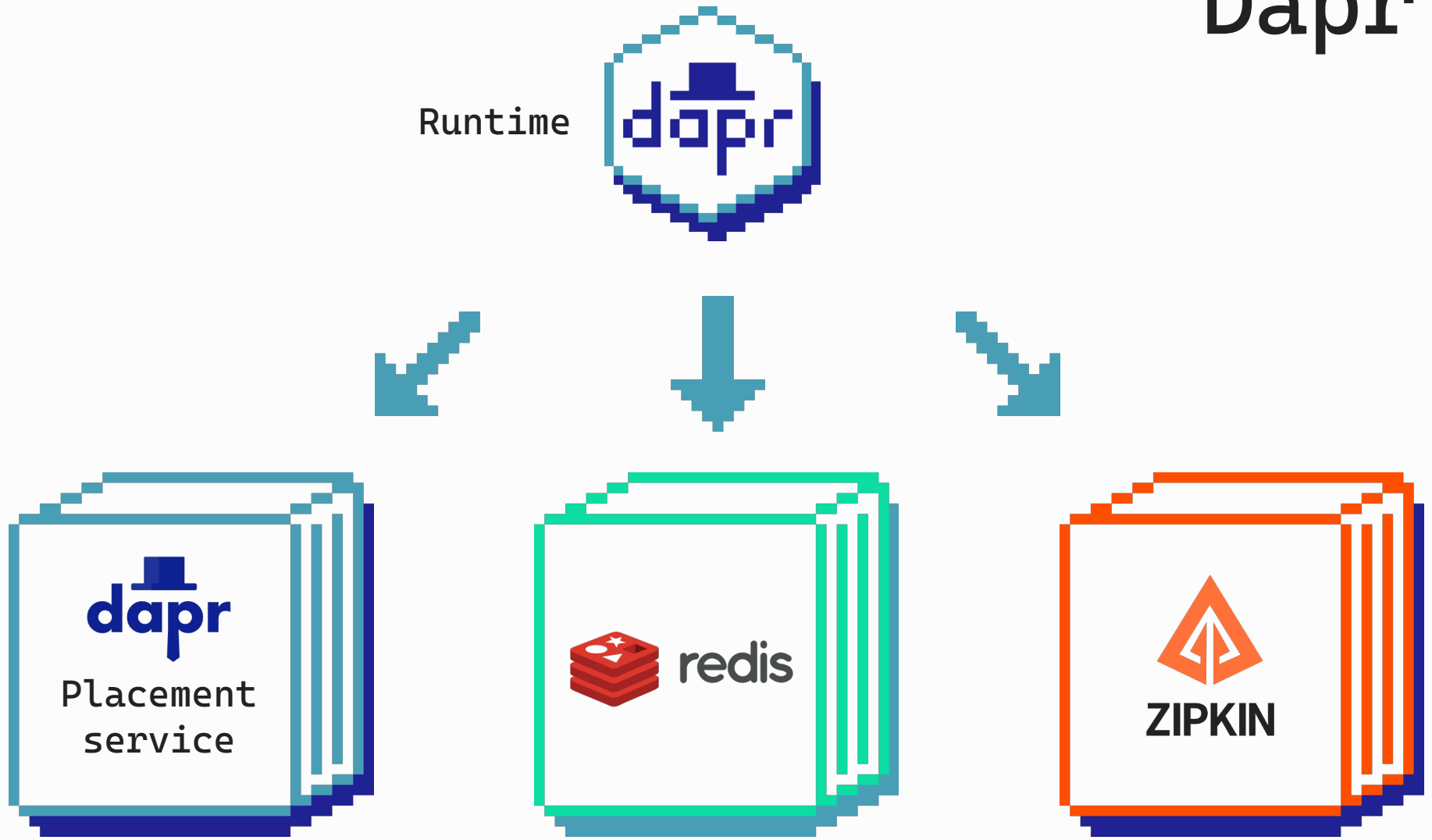
@marcduiker

dapr

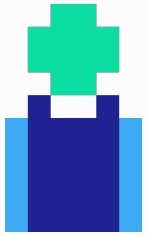
Actor Demos

https://github.com/diagrid-labs/dapr-actor-demos

Dapr CLI

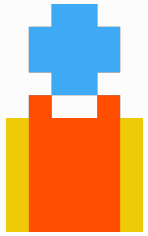


Basic Actor Samples

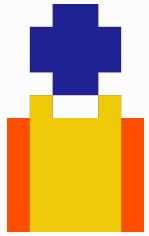


HelloWorld

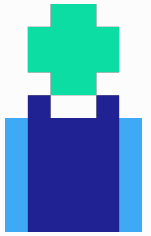
StatefulActor



TimerActor



ReminderActor



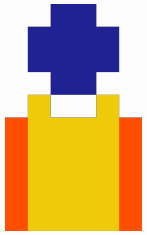
ActorToActor

Evil Corp 😈 Demo

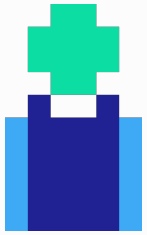
EvilCorp 😈 wants their employees to be more productive and have decided to implement a system with smart alarm clocks that will wake up their employees at 7am.

If the employees have not acknowledged the alarm within 3 snoozes, the alarm will send a message to the headquarters to lay off the employee 😱.

Evil Corp 😈 Demo

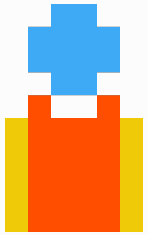


Simulation

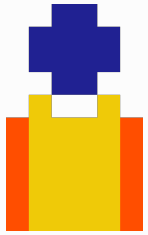


HeadQuarters

RegionalOffice



AlarmClock



Employee

Actor Demos

https://github.com/diagrid-labs/dapr-actor-demos

Dapr

Actor Model

Dapr Actor API

Demos!

Dapr supporter badge

Congratulations, you survived this presentation!

Claim this digital badge as your reward!

@marcduiker

Running Dapr on K8s? Try Conductor Free

diagrid.io/conductor

@marcduiker