



# Pluggable Components in Dapr

# Presenters



**Yaron Schneider**

@yaron2 

Maintainer and steering committee member

Diagrid



**Artur Souza**

@artursouza 

Maintainer and steering committee member

Microsoft

# Thanks to community contributors



**Marcos Candeia**  
@mcandeia



**John Ewart**  
@johnewart

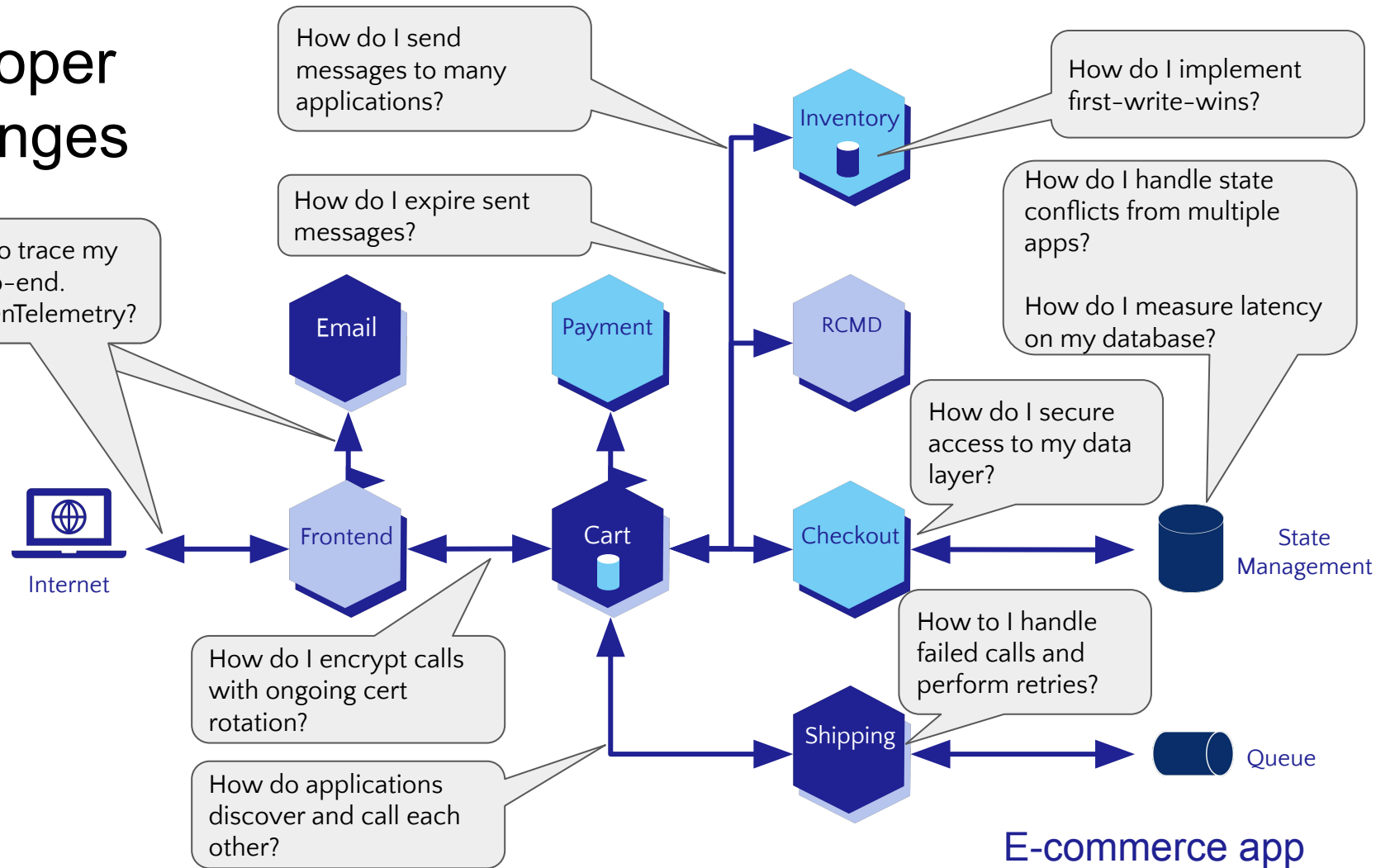


**Patrick Huber**  
@patrickhuber

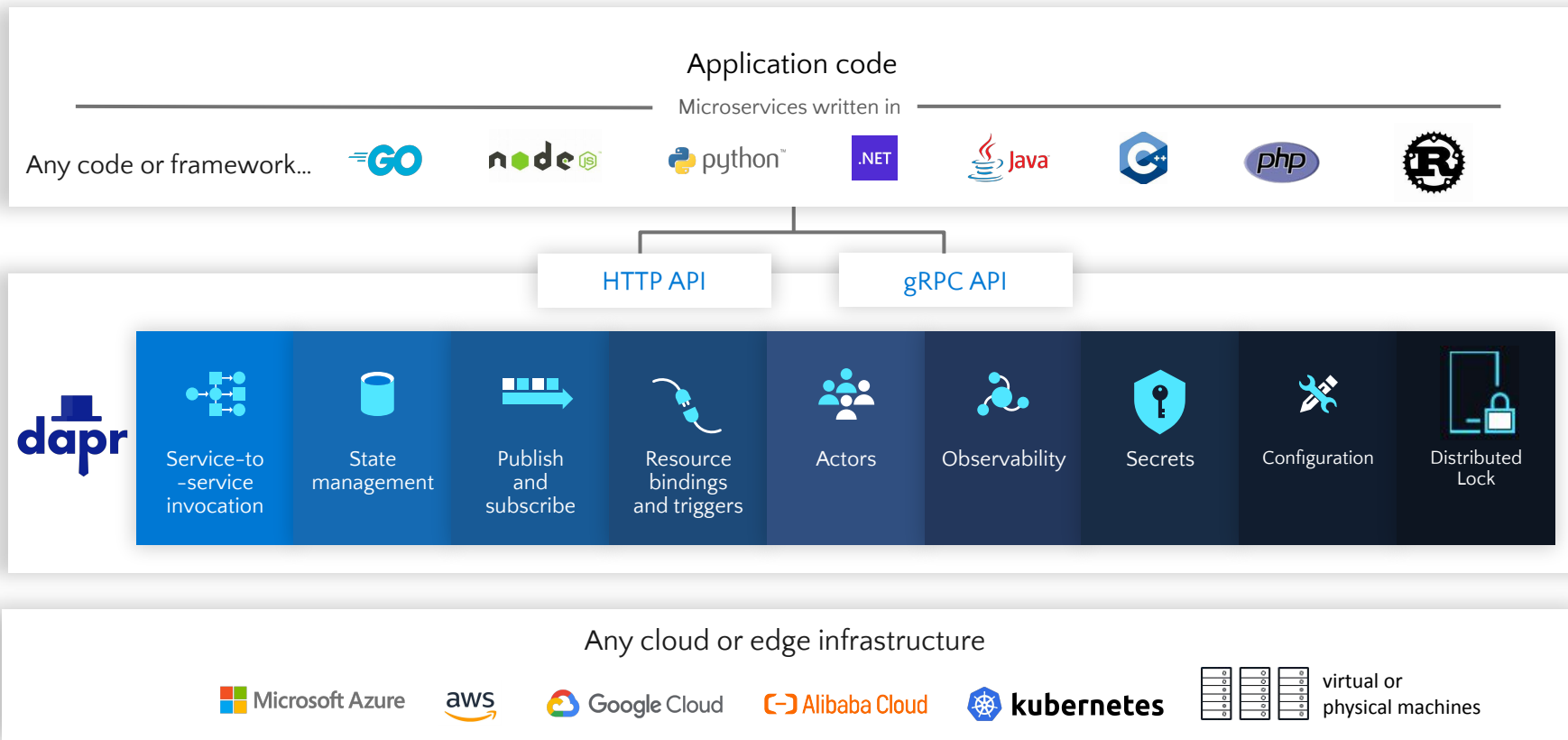


**Joni Collinge**  
@jjcollinge

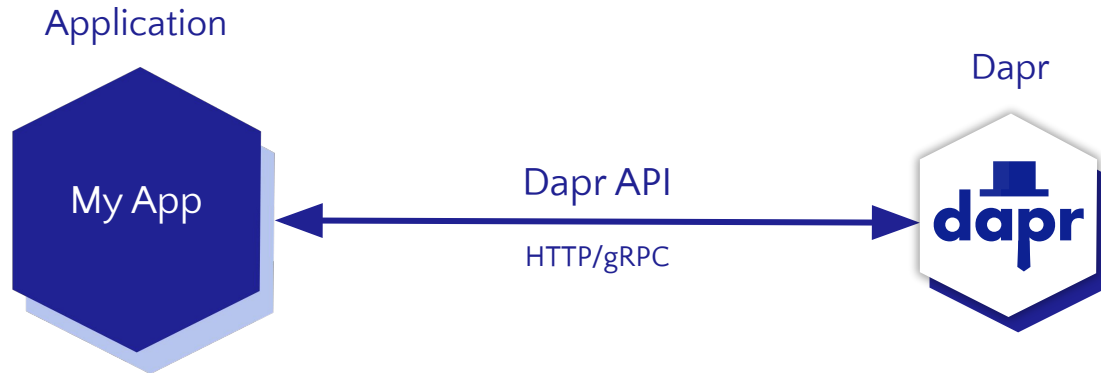
# Developer challenges



# Enter Dapr - Distributed Systems APIs



# Dapr does the heavy lifting



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

**GET** `http://localhost:3500/v1.0/state/inventory/item67`

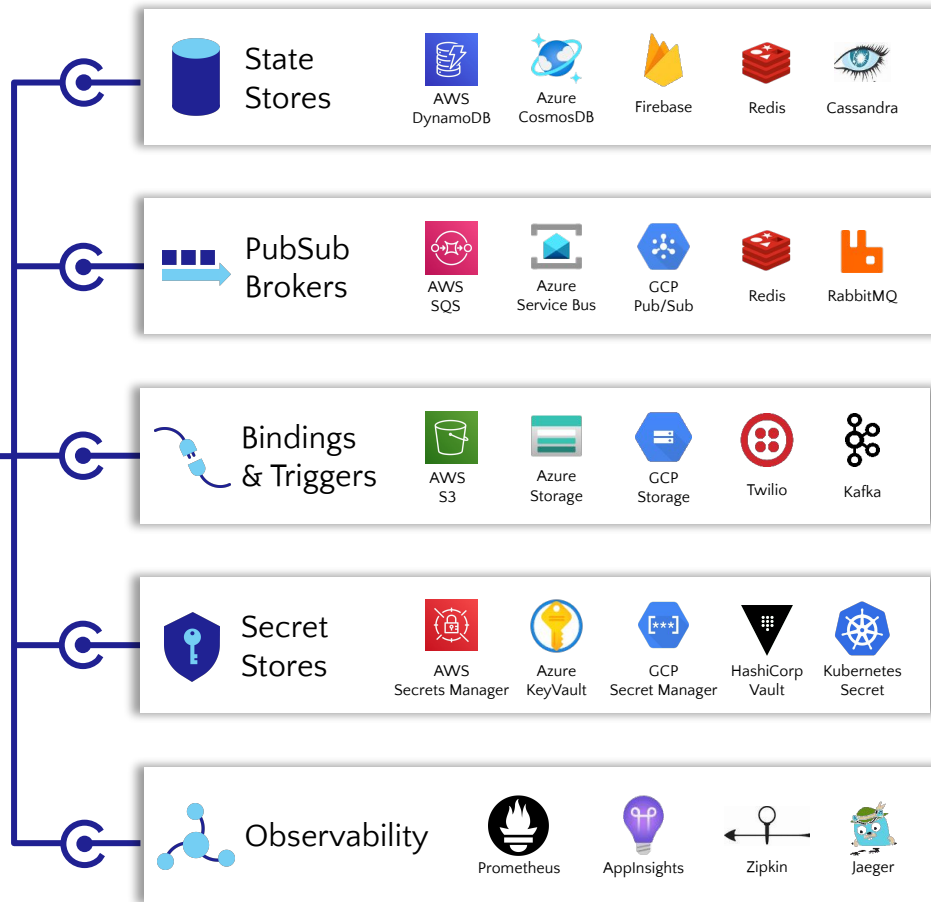
**POST** `http://localhost:3500/v1.0/publish/shipping/orders`

**GET** `http://localhost:3500/v1.0/secrets/appvault/password`

# Dapr integrates with existing infrastructure



Over 105 community contributed components



# Dapr community update

19.5k

GitHub stars

2.2k

Contributors

108

Community components

+1.0M

Docker Hub monthly pulls

4.7k

Discord members

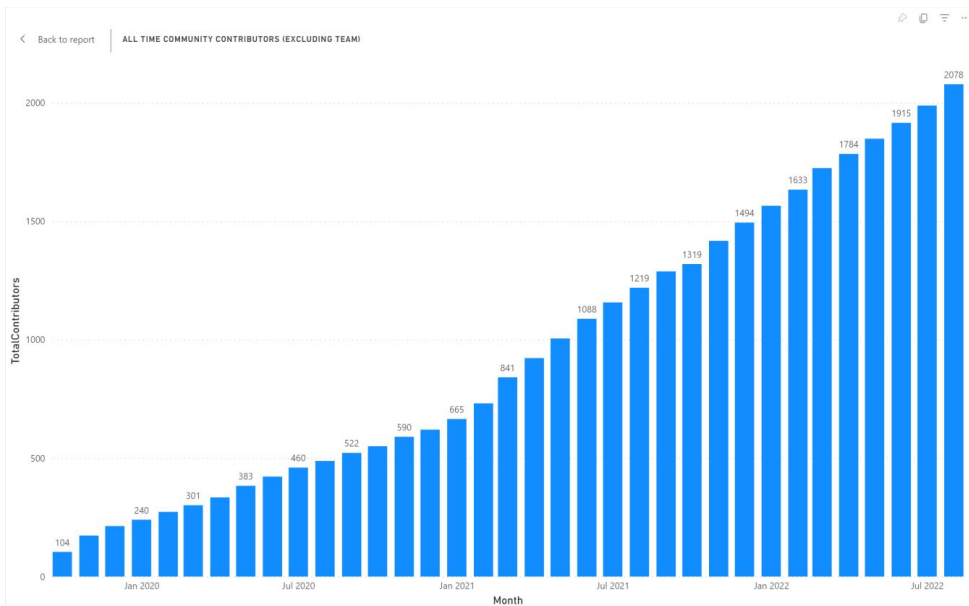
210k

Unique docs views/mo

Nov 2021: Dapr accepted into CNCF as incubating project

9th largest CNCF project (of 124)

DaprCon + KubeCon Oct 2022



Contributing organizations include:





# Dapr components

1. Compiled as part of the Dapr binary
2. Must be written in Go

HTTP API  
Server

gRPC API  
Server

 State  
Stores

  
AWS  
DynamoDB

  
Azure  
CosmosDB

  
Firebase

  
Redis

  
Cassandra

 PubSub  
Brokers


  
AWS  
SQS

  
Azure  
Service Bus

  
GCP  
Pub/Sub

  
Redis

  
RabbitMQ

 Bindings  
& Triggers

  
AWS  
S3

  
Azure  
Storage


  
GCP  
Storage

  
Twilio

  
Kafka

 Secret  
Stores

  
AWS  
Secrets Manager

  
Azure  
KeyVault

  
GCP  
Secret Manager

  
HashiCorp  
Vault

  
Kubernetes  
Secret

 Observability

  
Prometheus

  
Applinsights

  
Zipkin

  
Jaeger

# Adding private components before v1.9

Fork dapr/dapr repository from GitHub

Fork dapr/components-contrib repository from GitHub

Write custom component in Go

Build and package custom daprd binary with new component

Configure environment to run with custom daprd binary

# Adding private components in v1.9

Write custom component in any language as a gRPC service

Package

Deploy

# Core design tenants

1. Secure and performant by default - uses unix domain sockets
2. Leverages RPC standards - uses gRPC for communications
3. Low operational overhead - Leverages existing Dapr CRDs
4. Platform agnostic - supports both Kubernetes and self-hosted modes

# Go Plugins - ruled out

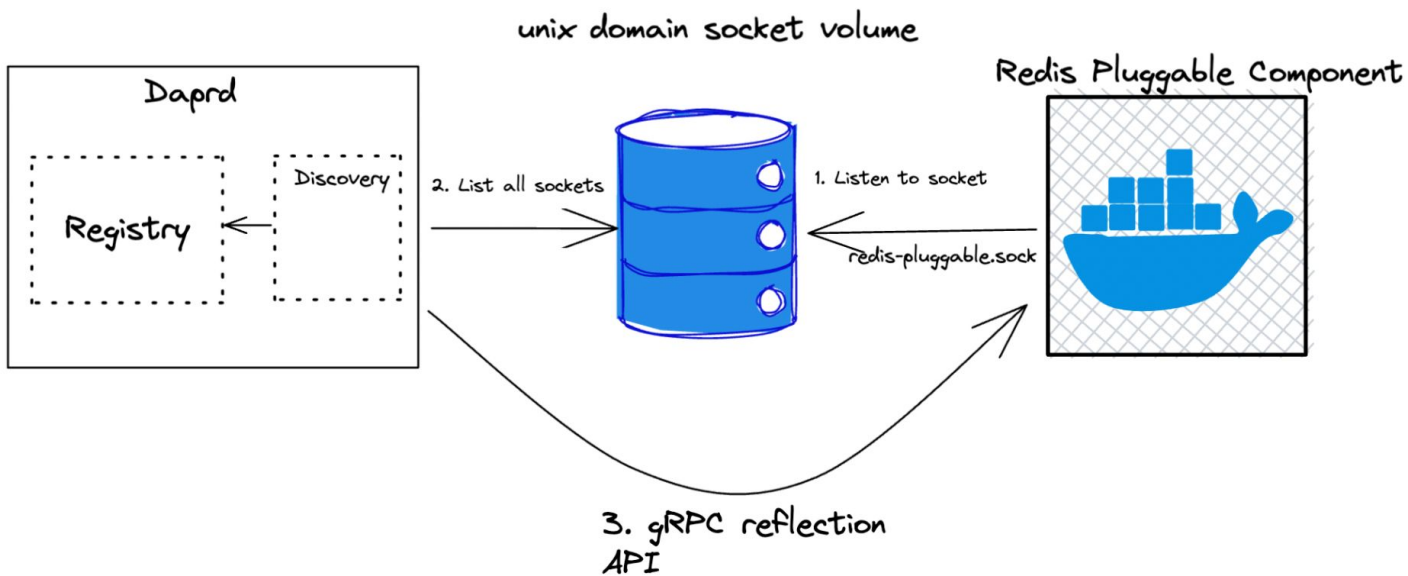
- Requires use of CGO (slow performance, cross compilation issues)
- Requires same dependencies version for plugin and host process
- Supports Linux and MacOS only

# User Experience

```
apiVersion: dapr.io/v1alpha1
kind: Component
metadata:
  name: customStateStore
spec:
  type: state.mydb
  version: v1
  metadata:
    - name: host
      value: http://my-address:8317/
```

/tmp/dapr-components-sockets/mydb.sock

# How it works



# Adding private components in v1.9 in standalone mode

```
$ dotnet run
```

```
Building...
```

```
Removing existing socket
```

```
warn: Microsoft.AspNetCore.Server.Kestrel[0]
```

```
Overriding address(es) 'http://localhost:5259, https://localhost:7089'. Binding to endpoints def.
```

```
info: Microsoft.Hosting.Lifetime[14]
```

```
Now listening on: http://unix:/tmp/dapr-components-sockets/memstore.sock
```

```
info: Microsoft.Hosting.Lifetime[0]
```

```
Application started. Press Ctrl+C to shut down.
```

```
info: Microsoft.Hosting.Lifetime[0]
```

```
Hosting environment: Development
```

```
info: Microsoft.Hosting.Lifetime[0]
```

```
Content root path: /Users/asouza/code/artursouza/kubecon-na-2022/pluggable-components/memstore/
```



# Adding private components in v1.9 in standalone mode

```
$ dapr run --app-id nodeapp --app-port 3000 --dapr-http-port 3500 node app.js
```

```
WARN[0000] A non-YAML component file statestore.yaml_ was detected, it will not be loaded instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime.components type=log ver=unknown
Starting Dapr with id nodeapp. HTTP Port: 3500. gRPC Port: 61680
INFO[0000] starting Dapr Runtime -- version 1.9.0-rc.3 -- commit adb3730123c6bf46d63660544680cbf41c94b4d7 app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] log level set to: info app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] metrics server started on :61681/ app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
WARN[0000] A non-YAML resiliency file statestore.yaml_ was detected, it will not be loaded app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] Resiliency configuration loaded. app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] standalone mode configured app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] app id: nodeapp app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] mTLS is disabled. Skipping certificate request and tls validation app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] local service entry announced: nodeapp -> 192.168.1.15:61682 app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.contrib type=log ver=1.9.0-rc.3
INFO[0000] Initialized name resolution to mdns app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
INFO[0000] pluggable component 'memstore' was successfully registered for 'dapr.proto.components.v1.StateStore' app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=pluggable-components-grpc-connector type=log ver=1.9.0-rc.3
INFO[0000] loading components app_id=nodeapp instance=DESKTOP_frlucas.europe.corp.microsoft.com scope=dapr.runtime type=log ver=1.9.0-rc.3
```

# Adding private components in v1.9 in Kubernetes mode

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nodeapp
  labels:
    app: node
spec:
  replicas: 1
  selector:
    matchLabels:
      app: node
  template:
    metadata:
      labels:
        app: node
      annotations:
        dapr.io/enabled: "true"
        dapr.io/app-id: "nodeapp"
        dapr.io/app-port: "3000"
        dapr.io/enable-api-logging: "true"
        dapr.io/unix-domain-socket-path: "/tmp/dapr-components-sockets"
```

# Adding private components in v1.9 in Kubernetes mode

```
spec:
  containers:
  - name: node
    image: ghcr.io/dapr/samples/hello-k8s-node:latest
    env:
    - name: APP_PORT
      value: "3000"
    ports:
    - containerPort: 3000
    imagePullPolicy: Always
  - name: memstore
    image: artursouza/memstore:latest
    imagePullPolicy: Always
    volumeMounts:
    - mountPath: /tmp/dapr-components-sockets
      name: dapr-unix-domain-socket
  volumes:
  - name: dapr-unix-domain-socket
    emptyDir: {}
```

DEMO

# What is next?

- More building blocks supported as pluggable components
- SDKs for easier implementation of Dapr components in multiple languages
- Tooling for easier packaging
- Tooling for easier deployment on Kubernetes and standalone mode
- WASM pluggable components as WASI matures

<https://sched.co/182P8>

