

Microservices Made Easy!

Jessica Deen @jldeen

Donovan Brown @DonovanBrown



Agenda



Getting Started

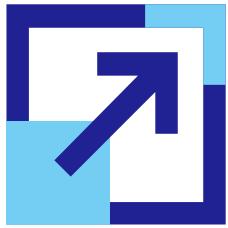


Extending the app

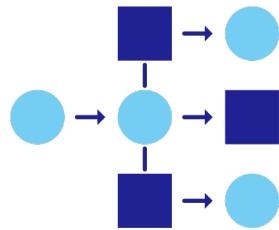


DevOps

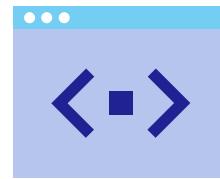
State of enterprise developers



Deploying scale-out apps for flexibility, cost, and efficiency



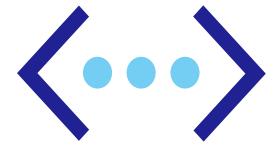
Developing resilient, scalable, microservice-based apps that interact with services



Focusing on building applications, not infrastructure



Trending toward serverless platforms with simple code to cloud pipelines

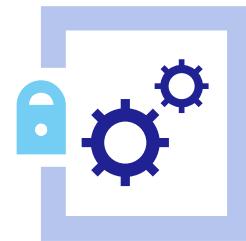


Using multiple languages and frameworks during development

What is holding back microservice development?



Limited tools and runtimes to build distributed applications



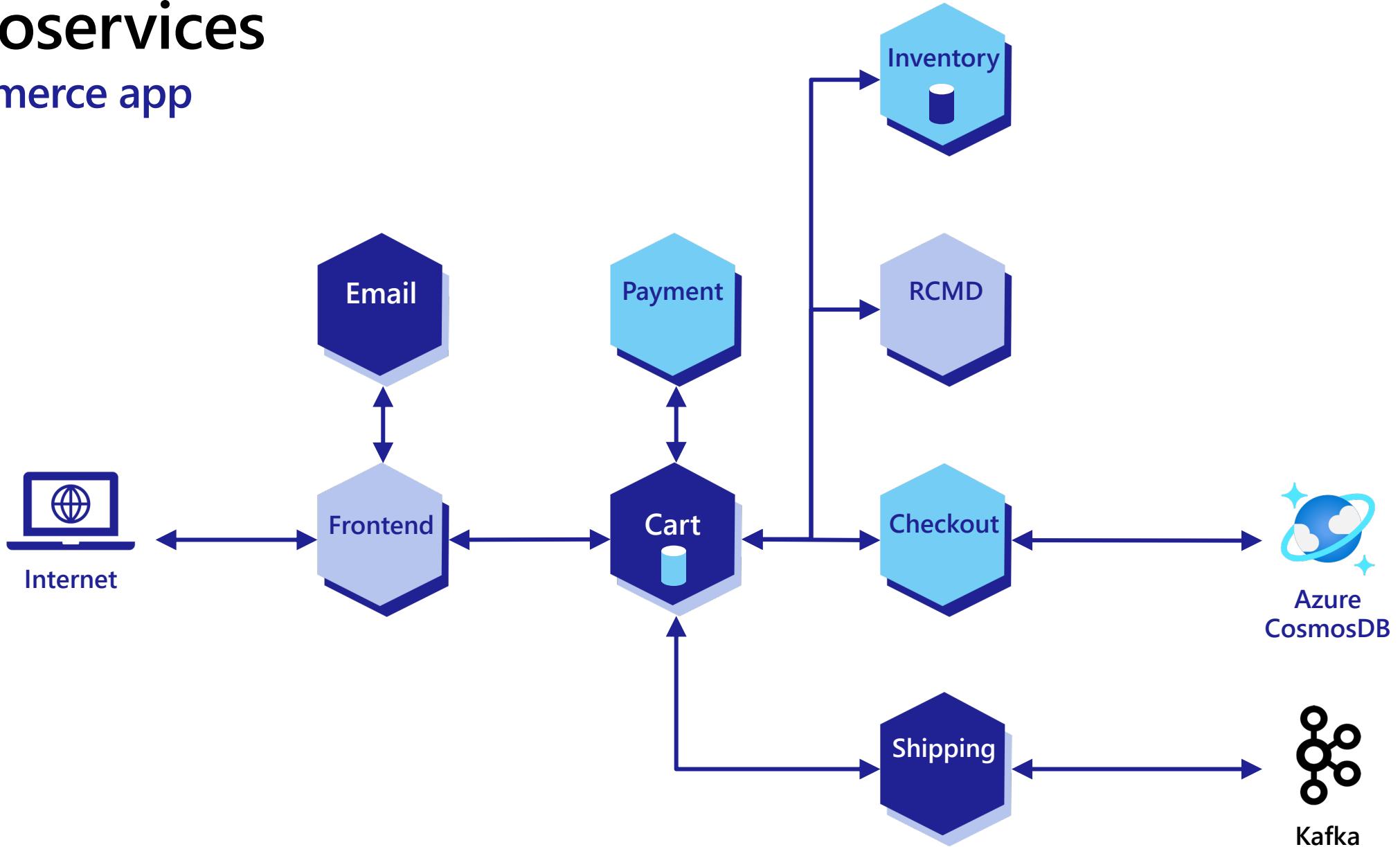
Runtimes have limited language support and tightly controlled feature sets



Runtimes only target specific infrastructure platforms with limited portability

Microservices

E-commerce app



Dapr Goals



Best-practices building blocks



Any language or framework



Consistent, portable, open APIs



Adopt standards



Extensible and pluggable components

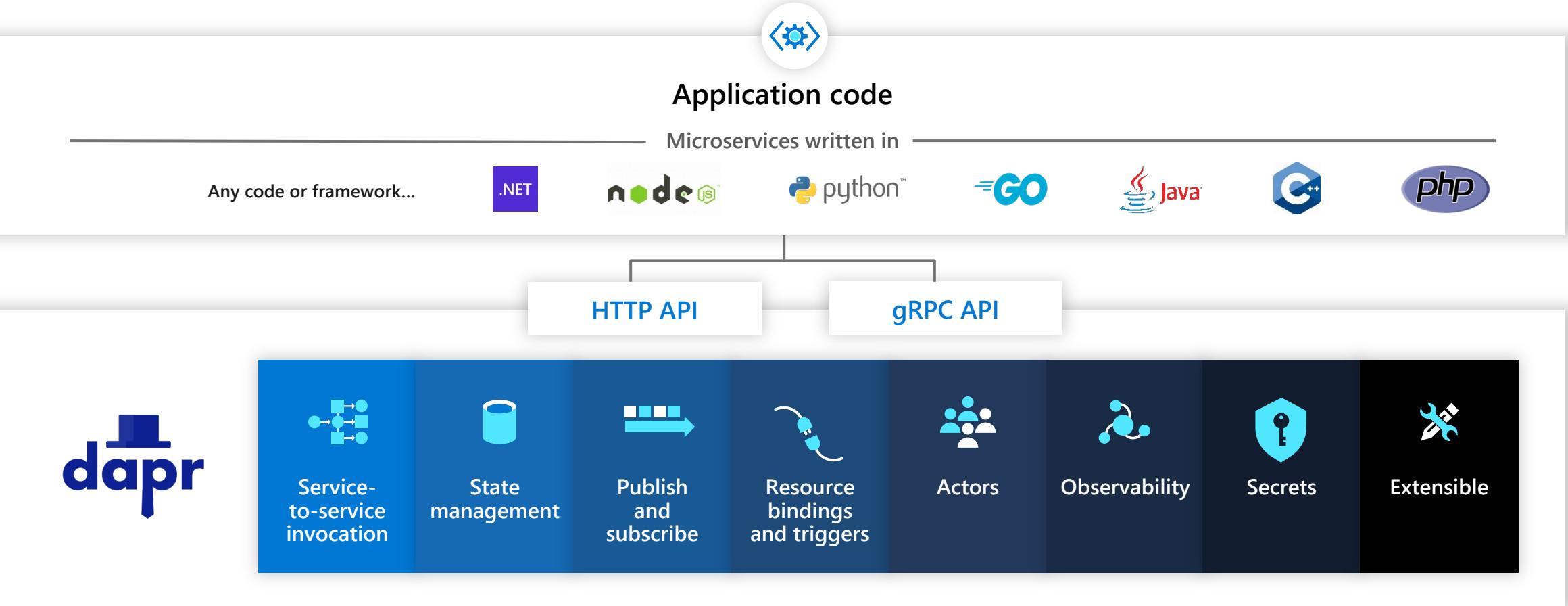


Platform agnostic cloud + edge



Community driven, vendor neutral

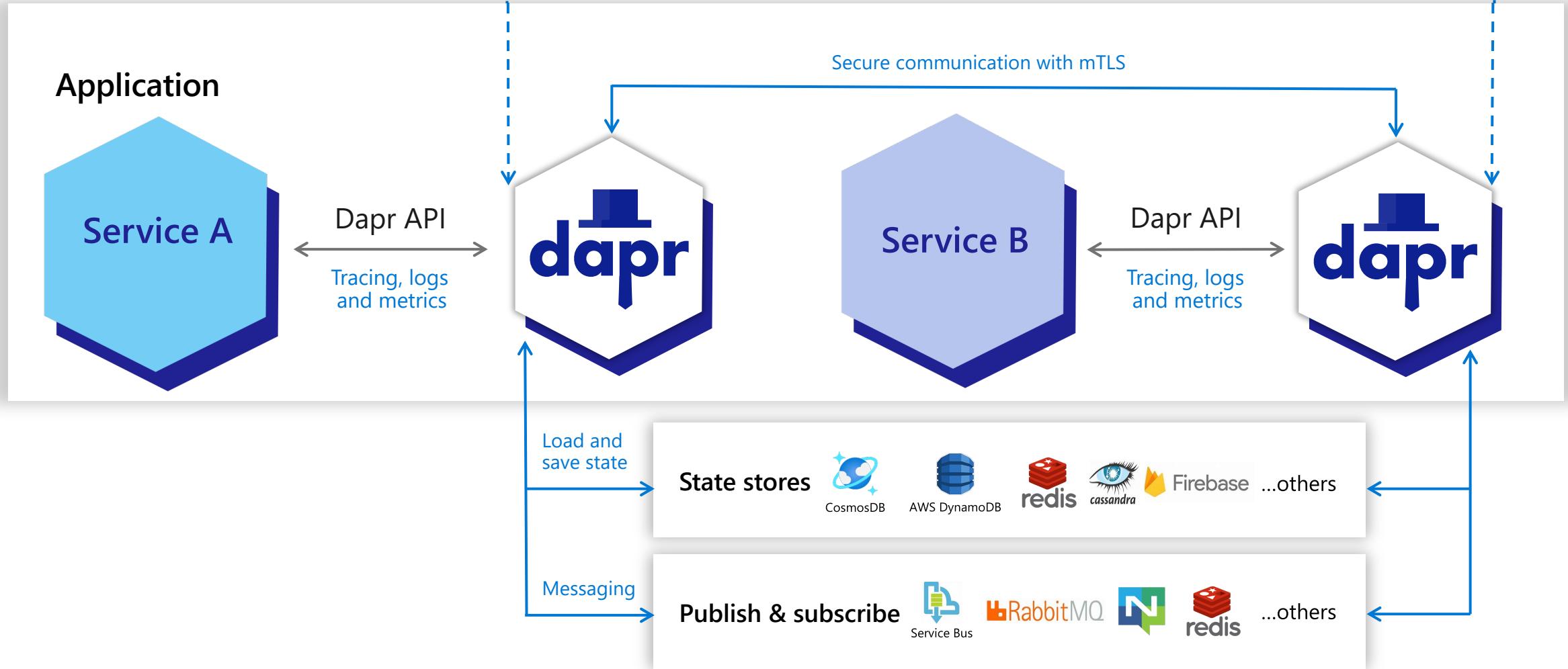
Any cloud or edge infrastructure

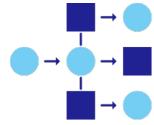


Hosting infrastructure

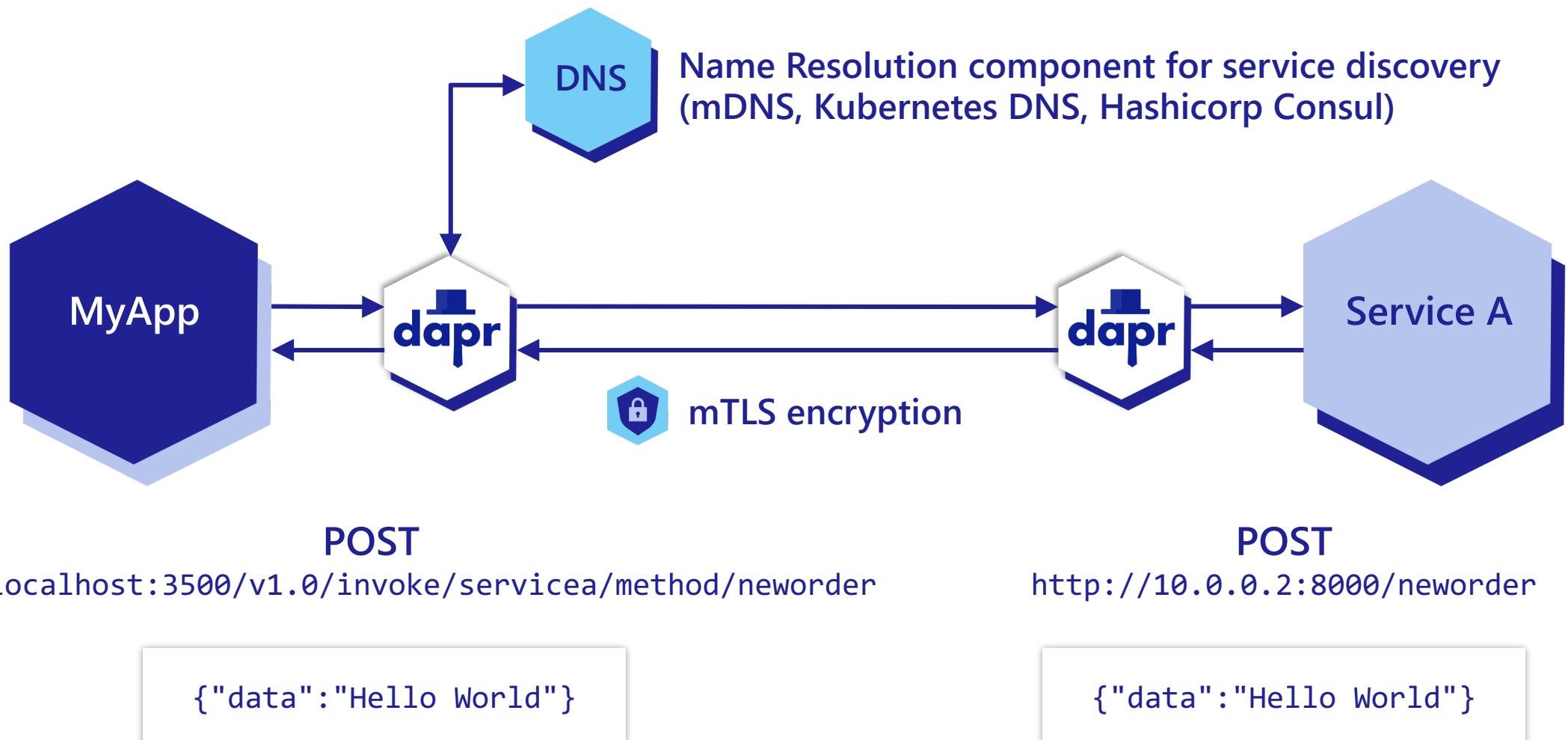


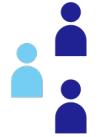
Sidecars and components





Service invocation



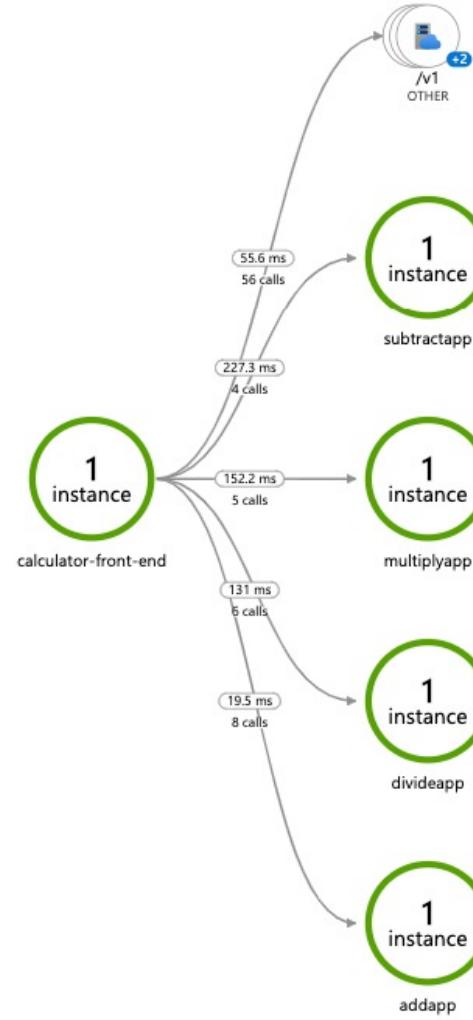


Observability Distributed tracing

Emit tracing data from calls to/from Dapr sidecars and system services for easy application-level instrumentation

Dapr distributed tracing features:

- ✓ Create call graphs using Trace IDs and Trace scopes
- ✓ Capture call metrics
- ✓ Configurable sampling rates
- ✓ ZipKin or Open Telemetry protocol support



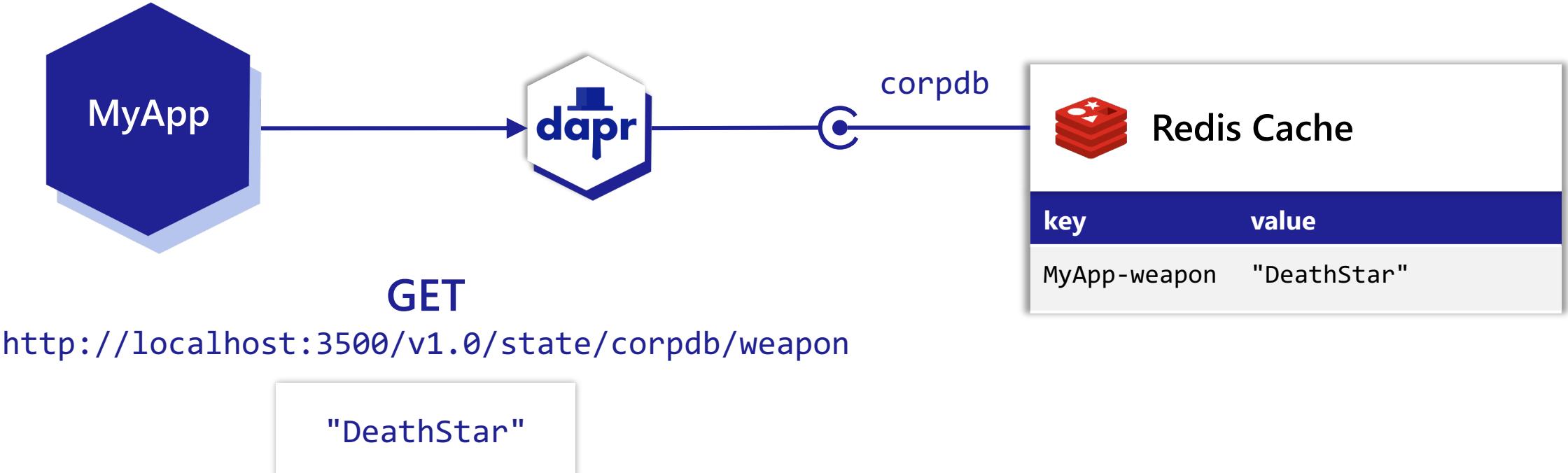
DEMO

Developer experience

Dapr and .NET Core

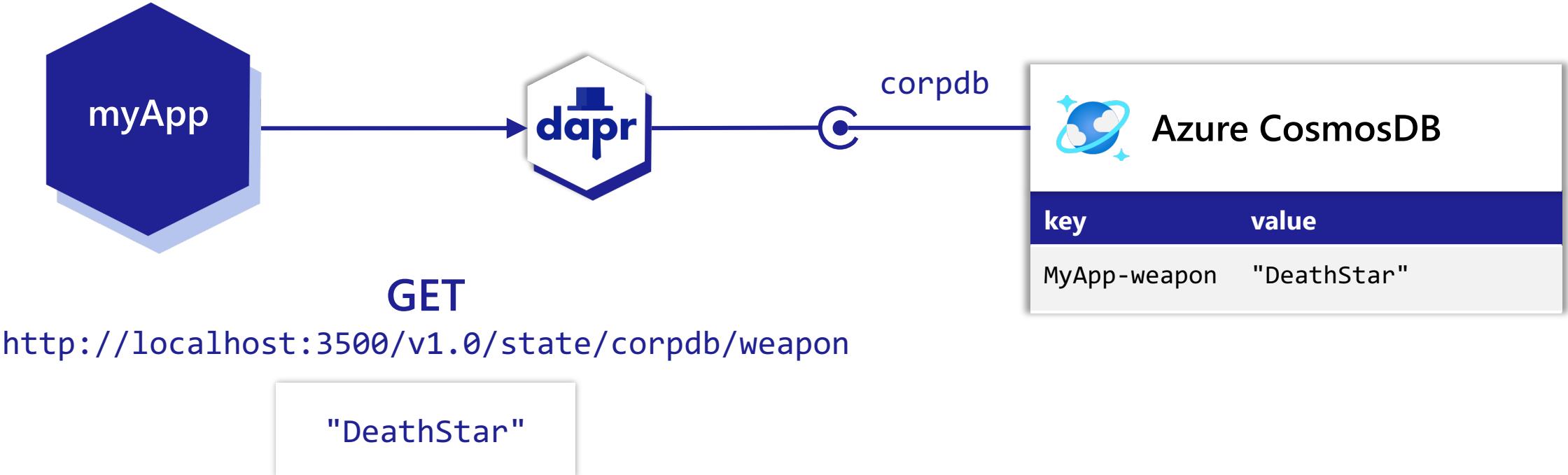


State management



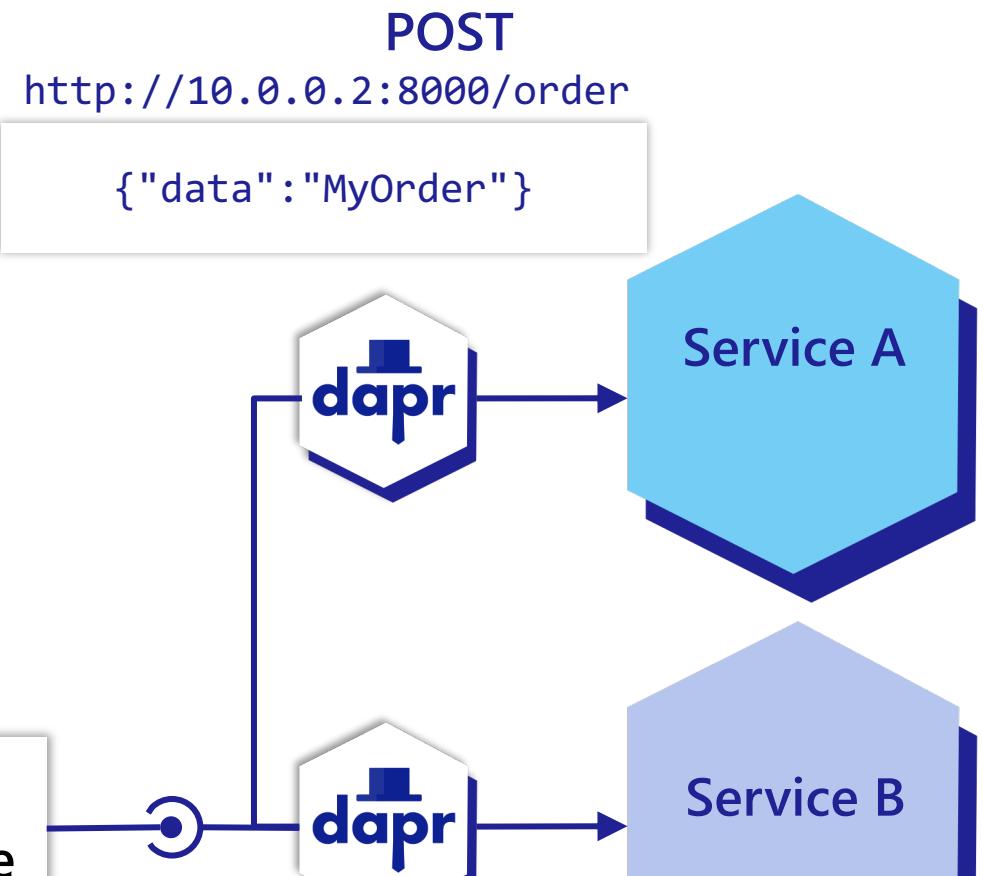
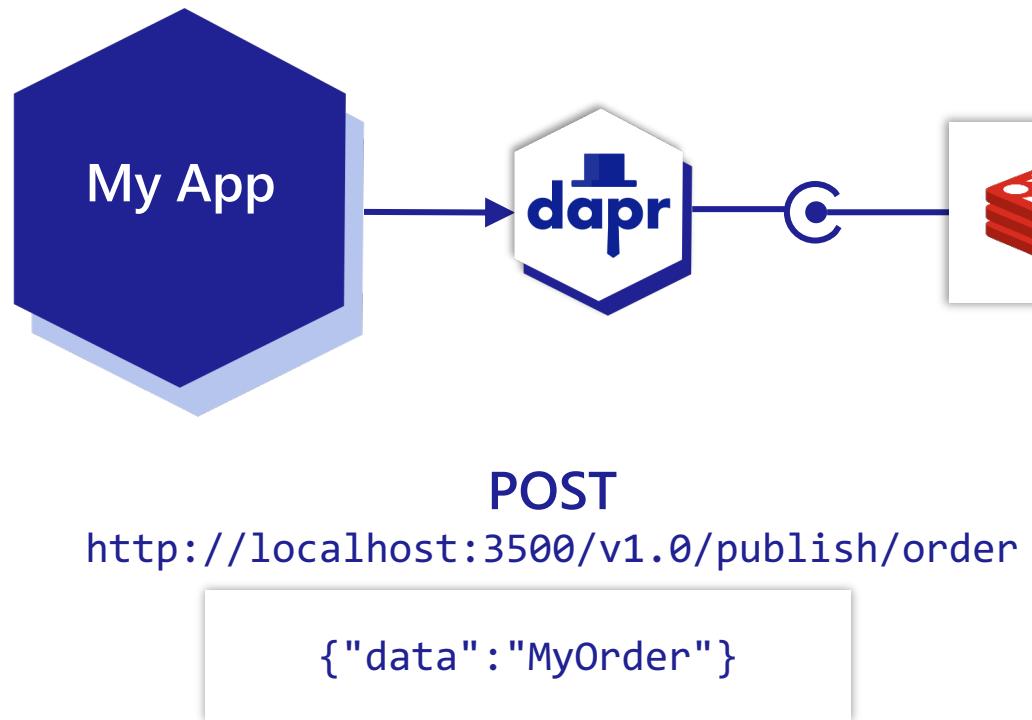


State management

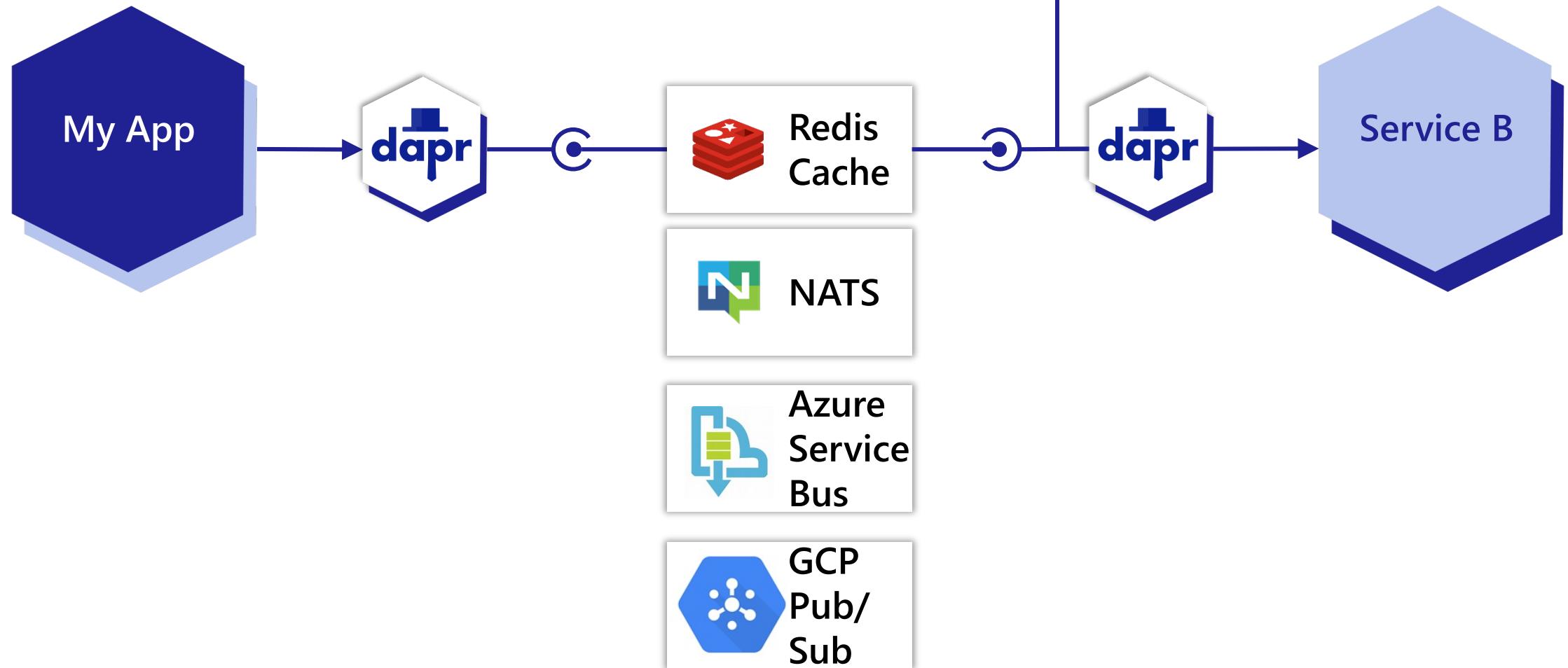




Publish and subscribe

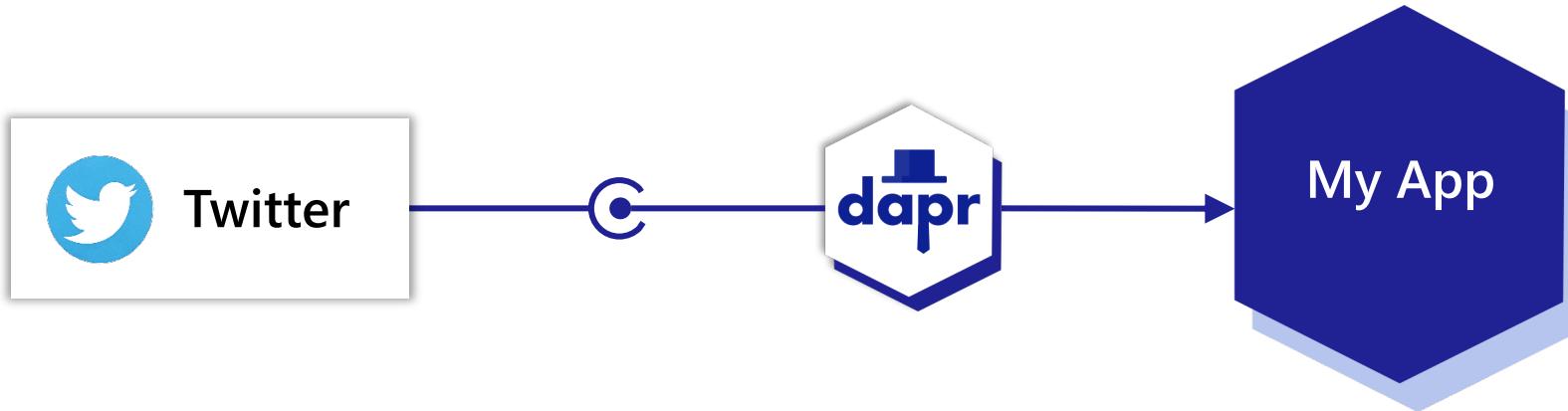


→ Publish and subscribe



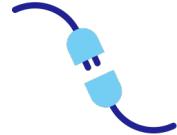


Input triggers



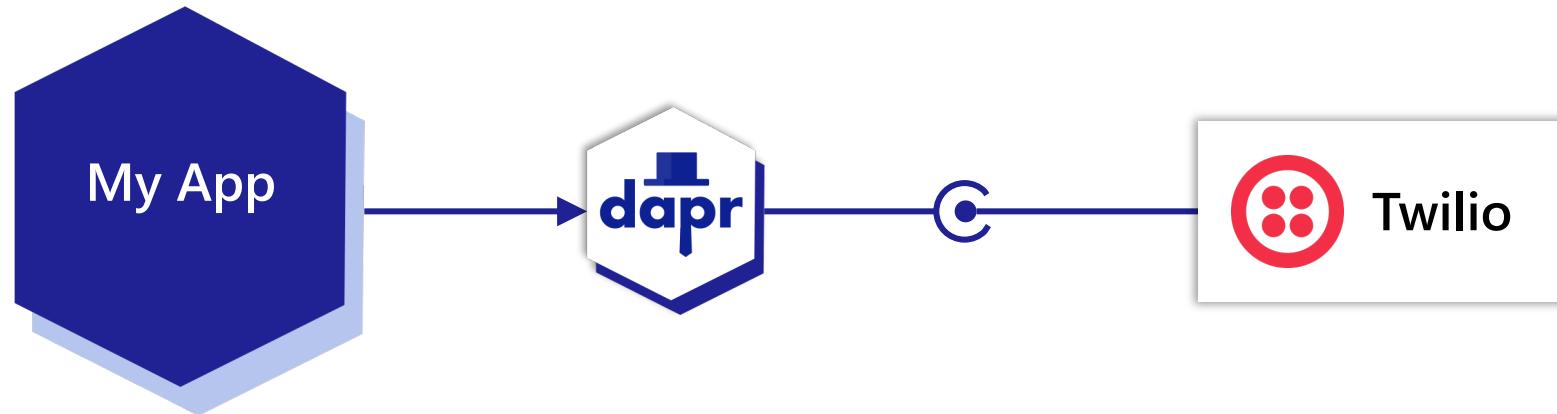
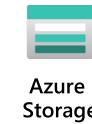
POST
`http://10.0.0.2:8000/newtweet`

```
{"data": ">We are excited  
to announce the ..."}
```



Output bindings

Bindings
& Triggers



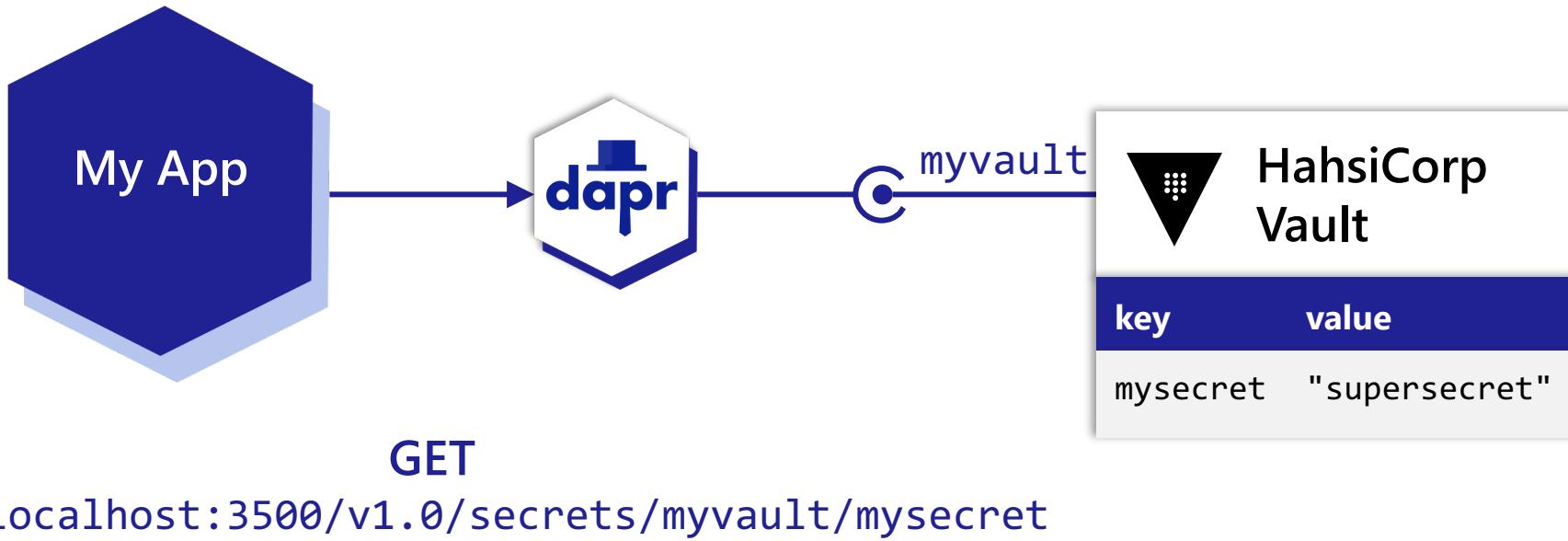
POST
<http://localhost:3500/v1.0/bindings/twilio>

```
{"data":"Hello World"}
```

Hello World



Secrets



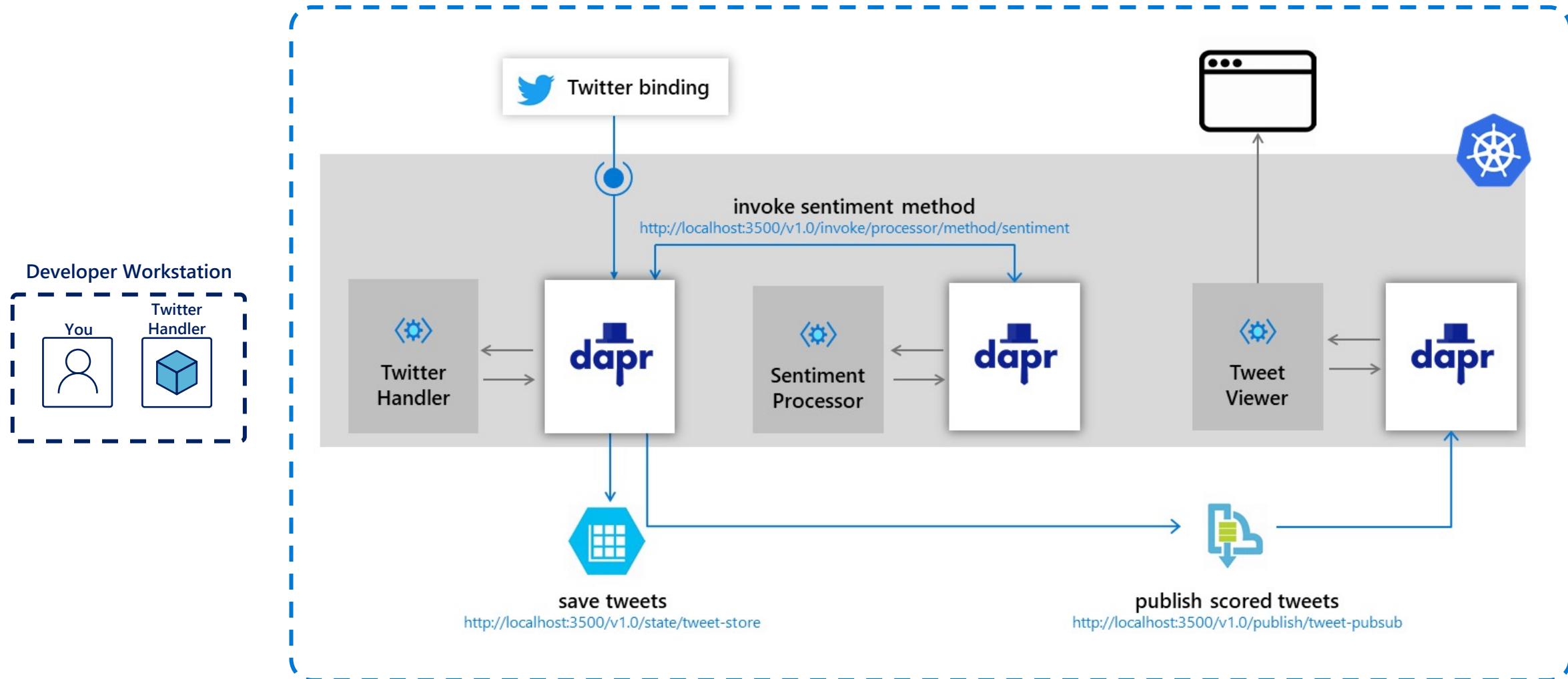
"supersecret"

DEMO

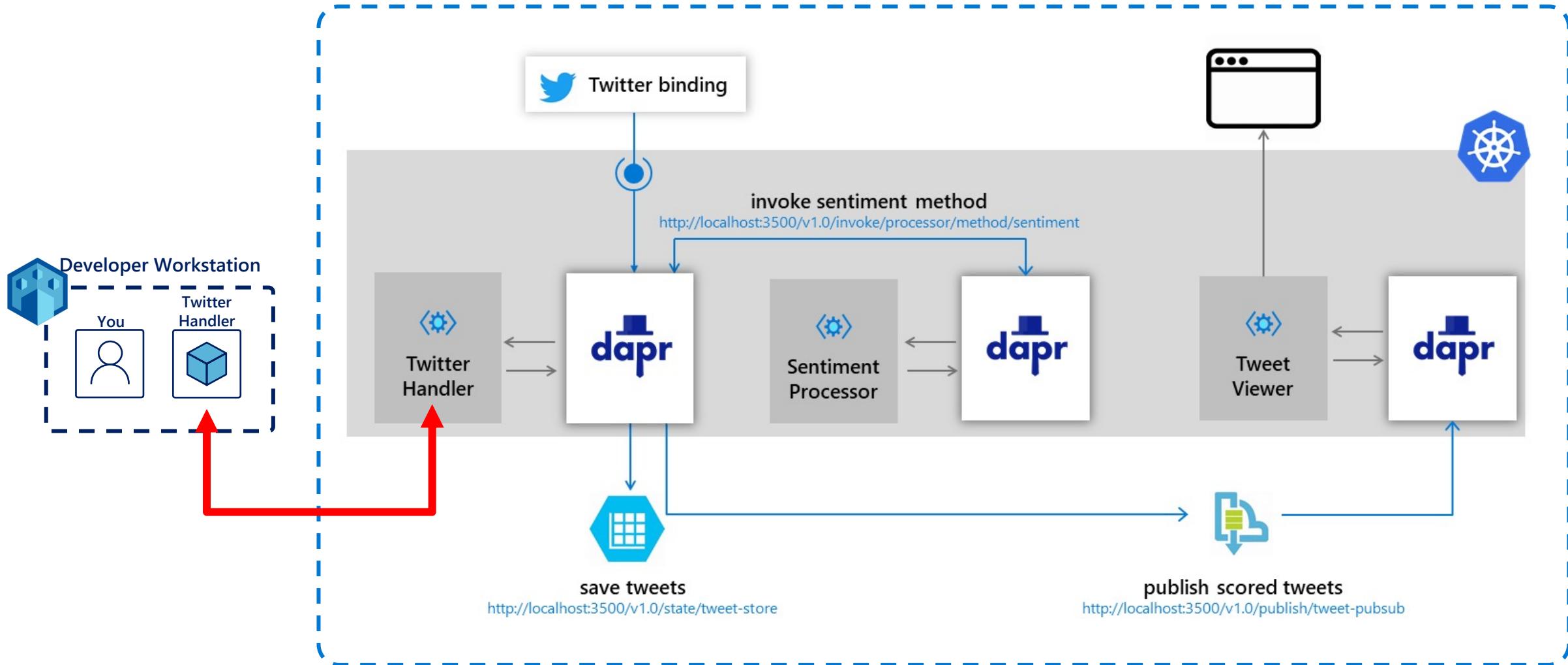
Developer experience

Dapr State management and Pub/Sub

Bridge to Kubernetes, Dapr, and DevOps



Bridge to Kubernetes, Dapr, and DevOps



DEMO

Debug & DevOps experience Bridge and Dapr GitHub Action

Dapr Community Momentum

Feb 2021: [Announced Dapr v1.0](#)

13.8k

GitHub stars

1087

Contributors

83

Community Components

41

Community calls

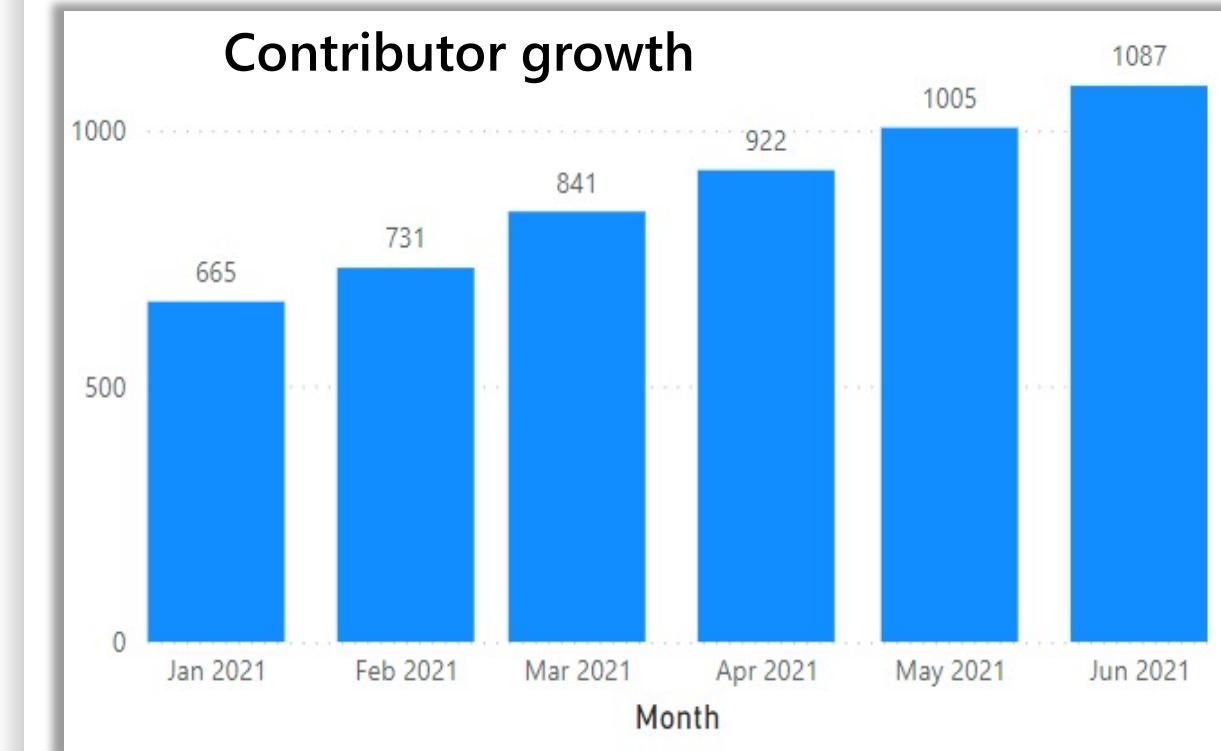
2k

Discord members

+200k

Monthly Docs views

March 2021 : [Dapr submitted to CNCF](#)



Contributing organizations include:



Tencent 腾讯



BOSCH

M Man

Join the community



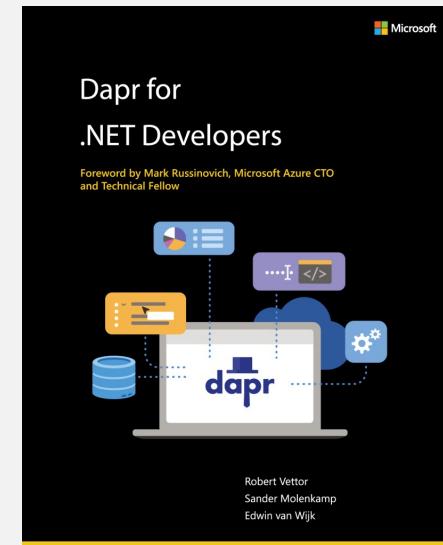
github.com/dapr

docs.dapr.io

aka.ms/dapr-community

aka.ms/dapr-discord

@daprdev



<https://aka.ms/dapr-ebook>

Dapr
Distributed Application Runtime. An event-driven, portable runtime for building microservices on cloud and edge.
<https://dapr.io>

Repositories 22 Packages People 98 Teams 4 Projects 13 Insights Settings Customize pinned repositories

Pinned repositories

- dapr Dapr is a portable, event-driven, runtime for building distributed applications across cloud and edge. Go ★ 6.1k 351
- docs User documentation for Dapr. ★ 536 137
- components-contrib Community driven, reusable components for distributed apps. Go ★ 53 51
- samples Dapr Samples Repository. JavaScript ★ 206 75

Search or jump to... Pull requests Issues Marketplace Explore

dapr / docs

Code Issues 36 Pull requests 4 Actions Wiki Security Insights Settings

Branch: master docs / getting-started / Create new file Upload files Find file History

AaronCrawfis Updated overview and concepts Latest commit 8faae3 13 days ago

cluster Fixes as per Microsoft Doc Authoring Pack ext. last month

README.md Updated overview and concepts 13 days ago

environment-setup.md Update environment-setup.md (#358) last month

Getting Started

Dapr is a portable, event-driven runtime that makes it easy for enterprise developers to build resilient, microservice stateless and stateful applications that run on the cloud and edge and embraces the diversity of languages and developer frameworks.

Core Concepts

runtime Discussion on the runtime repo https://github.com/dapr/runtime Seeing duplicated actions jobs on my PRs...

nowhere I'm currently trying to get a few jobs fail with internal errors. I guess WH is getting to the computers too.

wesleyly I'm currently trying to do this

brianwilkerson If I initialized Dapr in slim mode, will it still look to \$HOME/.dapr/components or will I always need to supply a --components-path arg to the cli?

willitlookthere Is there a default

wesleyly So my gitlab actions are running again, but the older duplicated failed actions are still blocking me from merging my PR. Anybody else noticed around this today? Thinking of maybe pushing a no-op change to this branch to see if that clears things up?

That seems to have worked...

wesleyly I'm unable to reproduce the failure to see key error. But so I understand the issue, what do you expect to happen in this case. You are applying 2 operations (put) to an existing key (creation of that item was part of this transaction). Do you expect the first update to fail because of the first update or the second?

willitlookthere I'm currently trying to include both a new key and the previously existing key. Still seeing a partial commit in Windows, OSX, and Linux.

wesleyly I've updated the example to include both a new key and the previously existing key. Still seeing a partial commit in Windows, OSX, and Linux.

willitlookthere I've updated the example to include both a new key and the previously existing key. Still seeing a partial commit in Windows, OSX, and Linux.

wesleyly Do you expect the first update to fail because of the first write-concurrency and strong consistency? Yes, I expect any failed transaction to be rolled back.

willitlookthere I can replicate this with Redis now too. I'm looking into the state component implementation to make sure the versions returned by Redis which we parse into an stag in Dapr plays role in uppercase. Should it even err when the version is diff. I know this works as intended in Cosmos and other transaction-aware DBs.