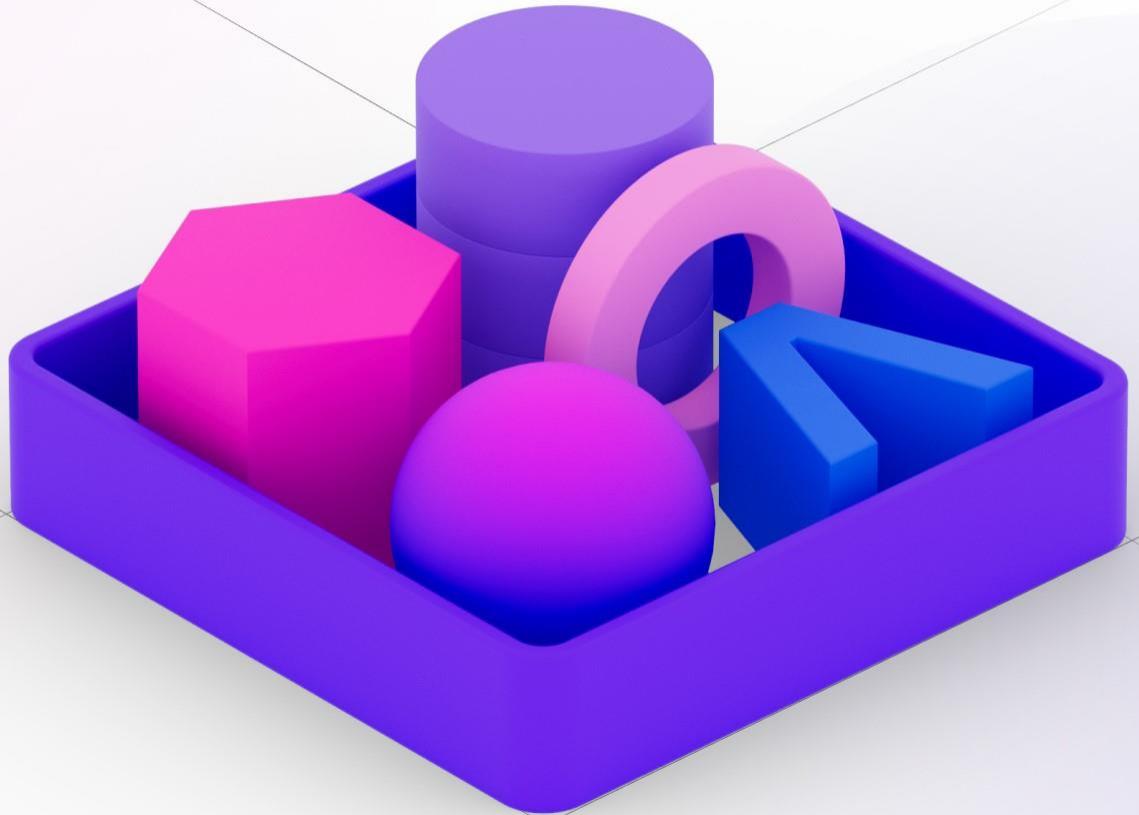


# .NET Conf



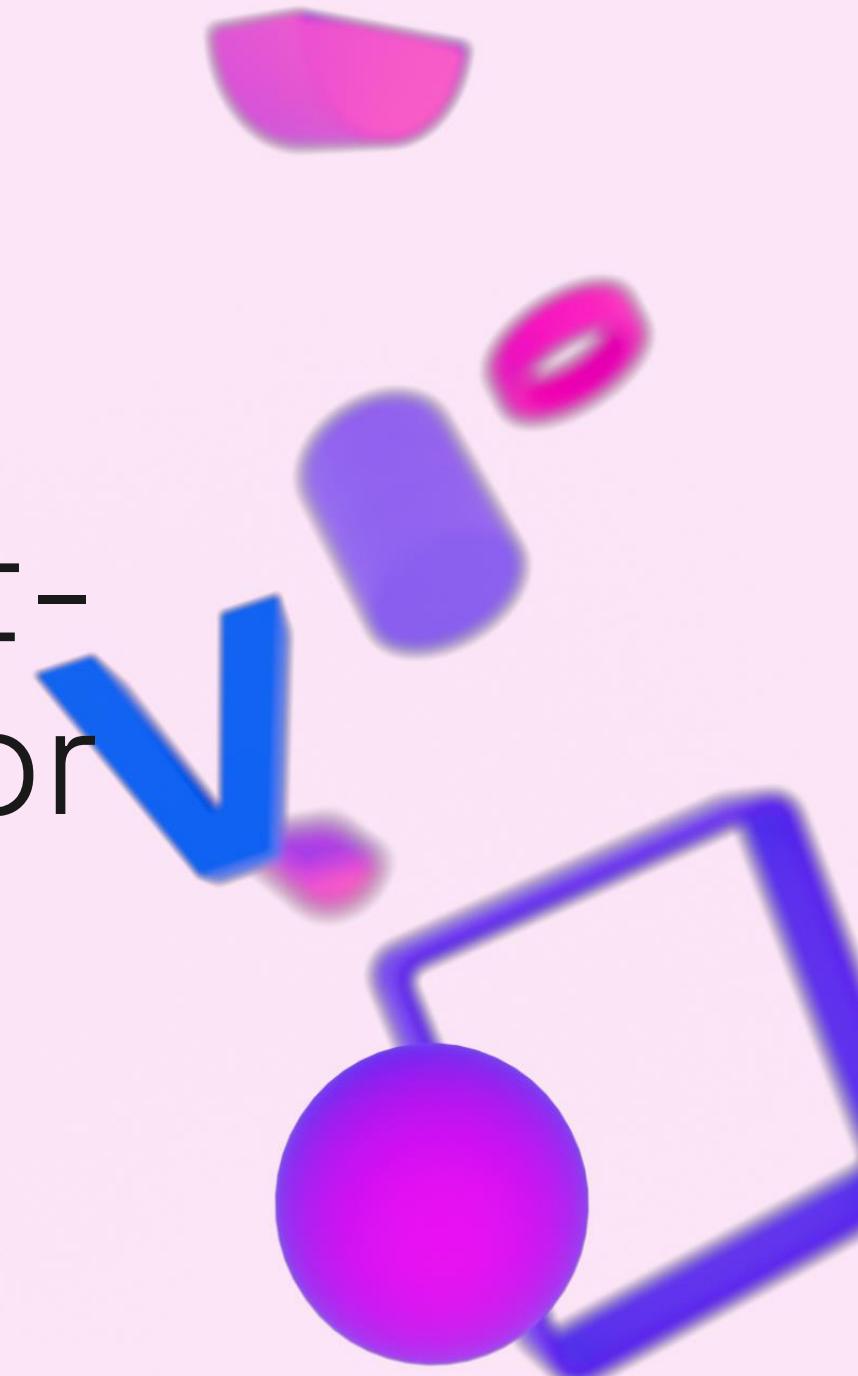


# Building .NET 8 Event-Driven Apps with Dapr

Thang Chung

November 2023

#dotnetconf2023



# Agenda

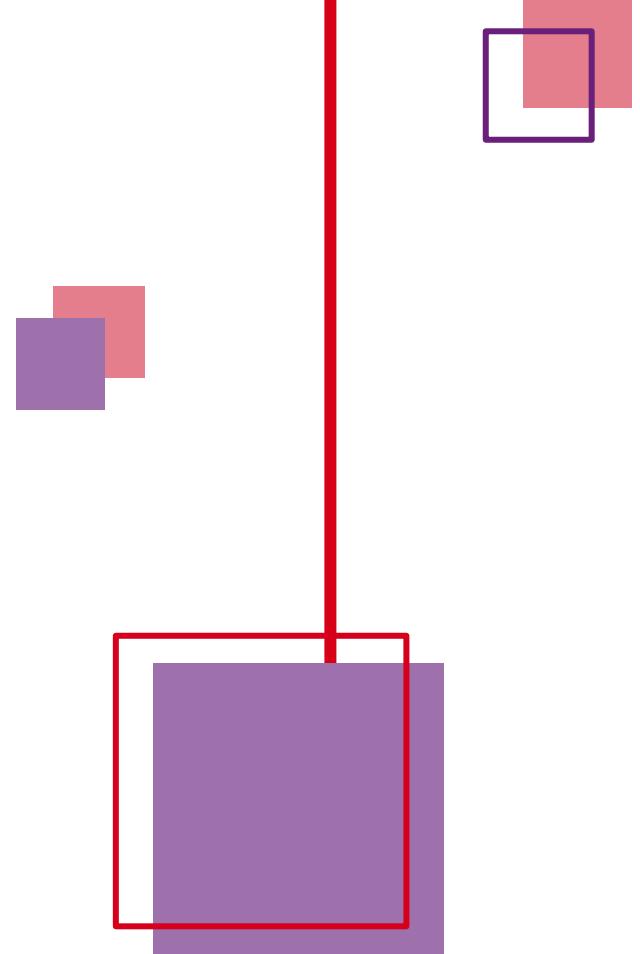
- Problems of distributed application development
  - What is EDA and its benefits.
- DAPR (Distributed Application Runtime) – what is it? And how does it help?
- Demo.
- Common Pitfalls & Solutions.



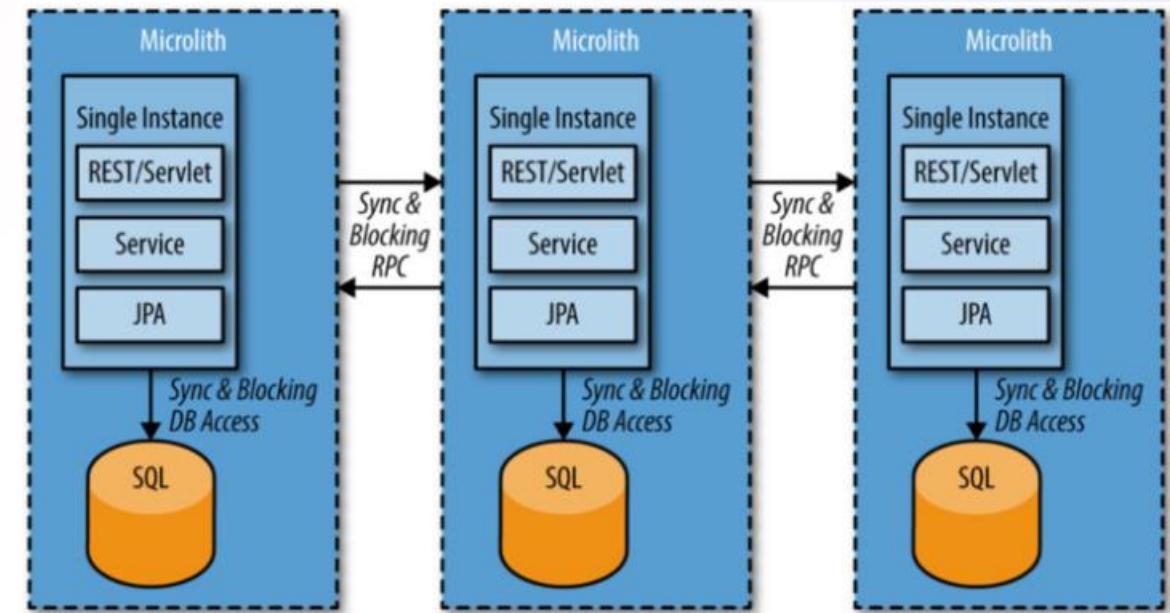
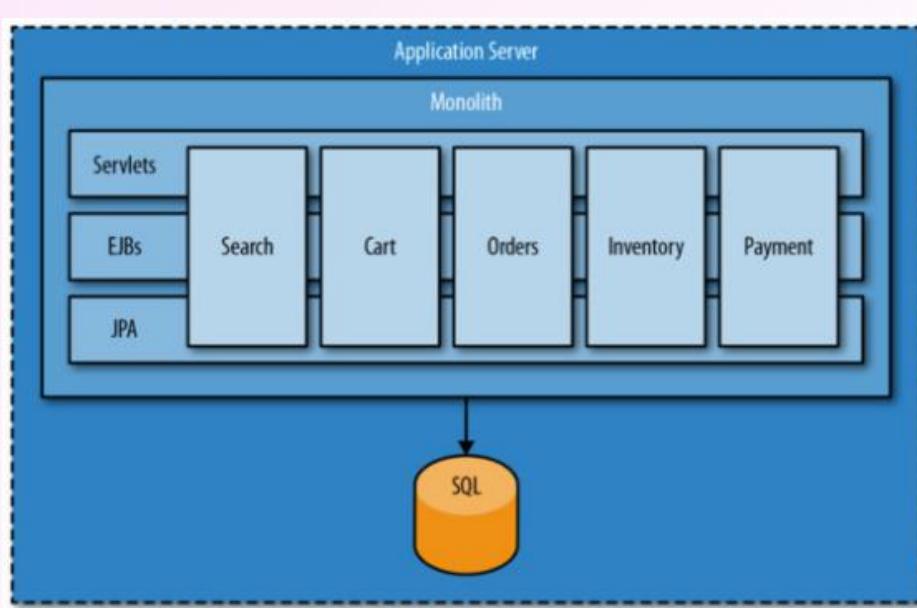
## Thang Chung

Technical Manager, NashTech VN  
Microsoft Azure MVP

- Creator of Vietnam Microservices Group on Facebook (>16k members).
  - <https://www.facebook.com/groups/645391349250568>
- Experience: >16 years in software consult, design, development, and deployment software for outsourcing, product, and startup companies.
- Expertise in cloud computing, cloud-native platform, serverless, and WebAssembly/WASI.
- Blog: <https://dev.to/thangchung>
- GitHub: <https://github.com/thangchung>
- LinkedIn: <https://www.linkedin.com/in/thang-chung-2b475614/>
- Twitter: @thangchung

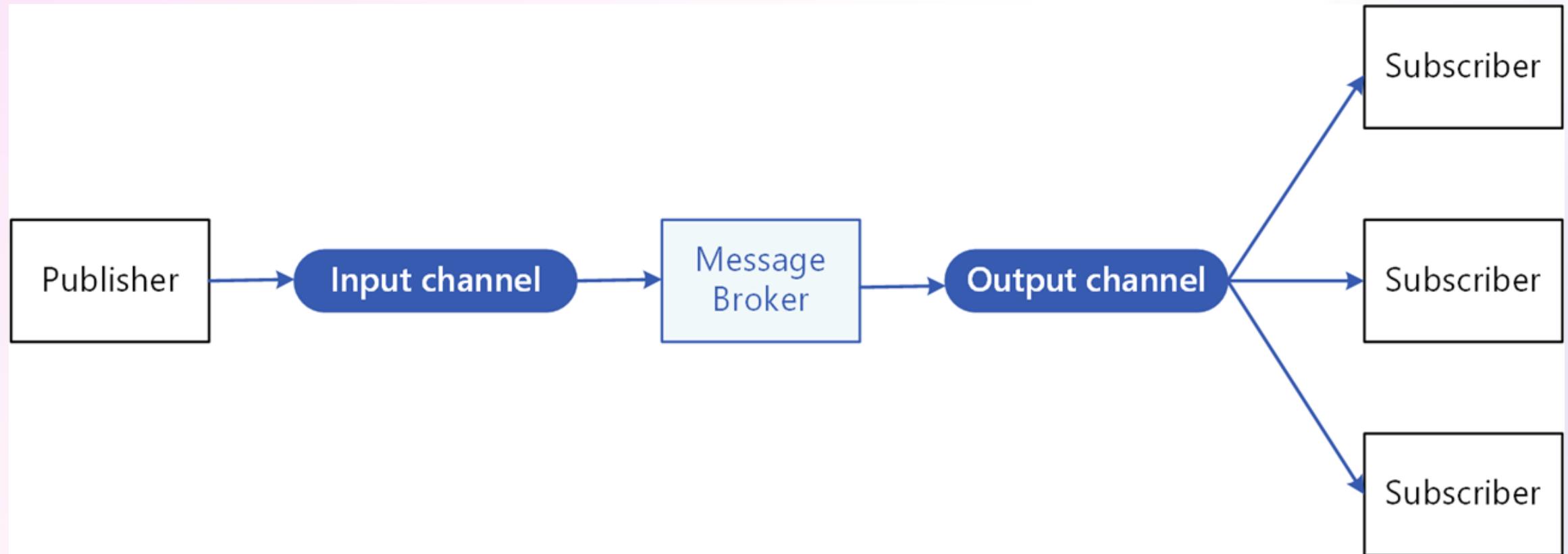


# Problems of building distributed apps?



*Distributed Monolith anti-pattern*

# What is Event-driven Architecture?

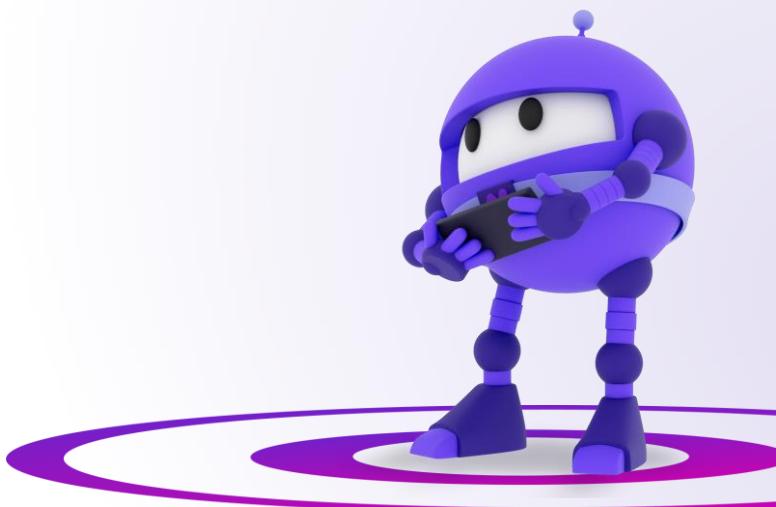


# Benefits of Event-driven Architecture

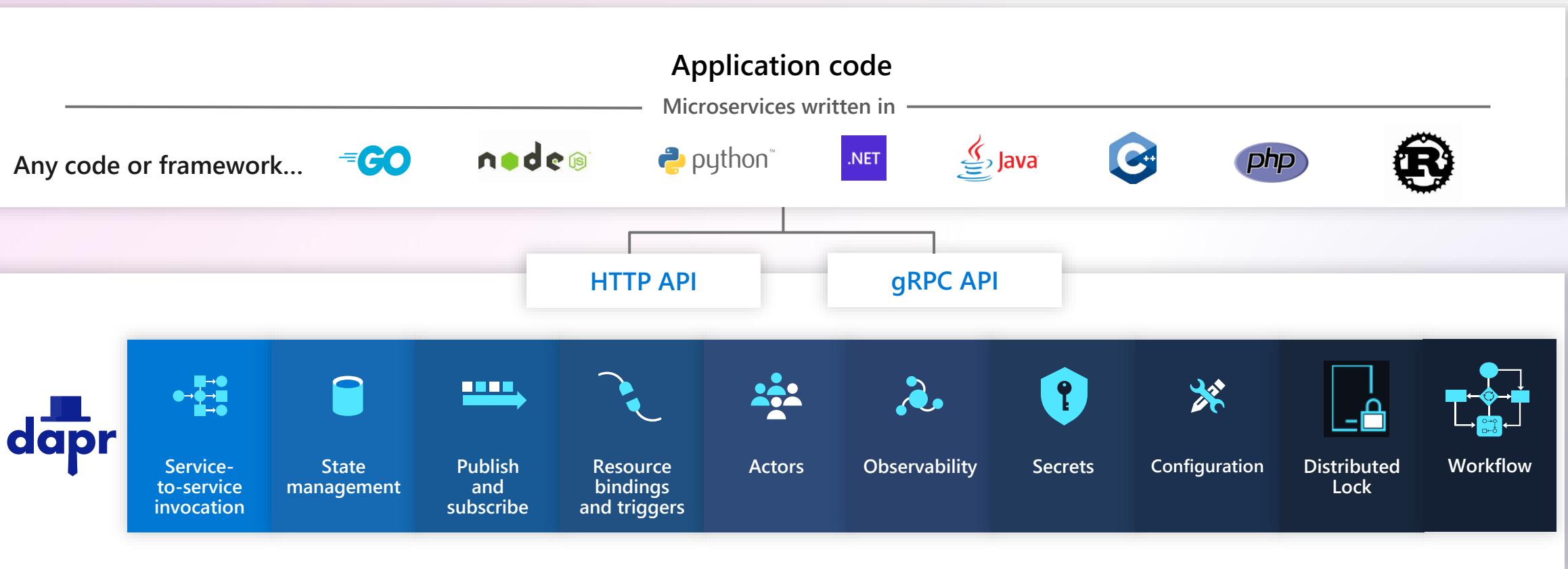
- Scalability
- Resilience
- Flexibility
- Responsiveness

# Dapr (Distributed Application Runtime)

APIs for Building Secure and Reliable Microservices



# Dapr APIs



Any cloud or edge infrastructure



virtual or  
physical machines

# Sidecar model



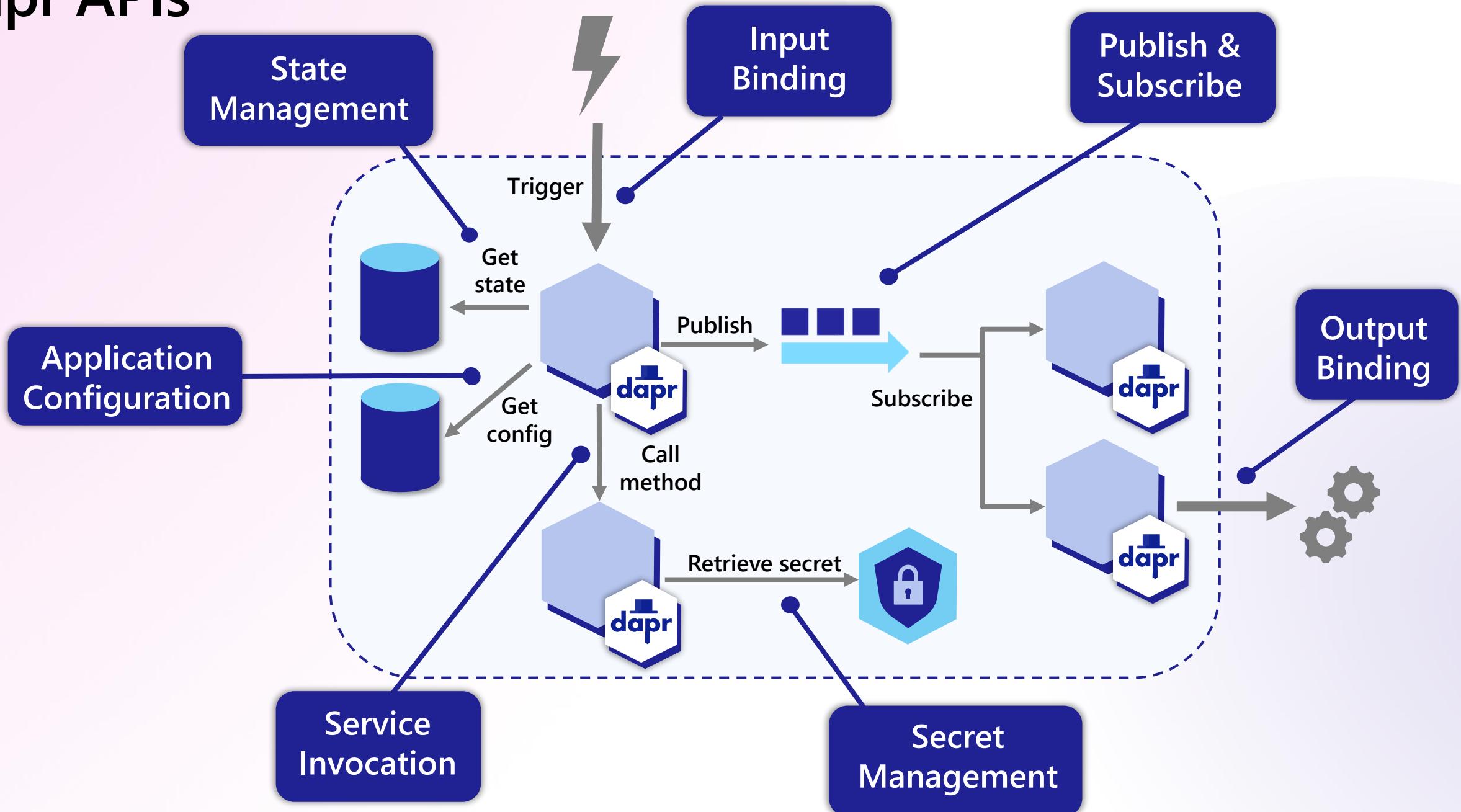
`POST http://localhost:3500/v1.0/invoke/cart/method/neworder`

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

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

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

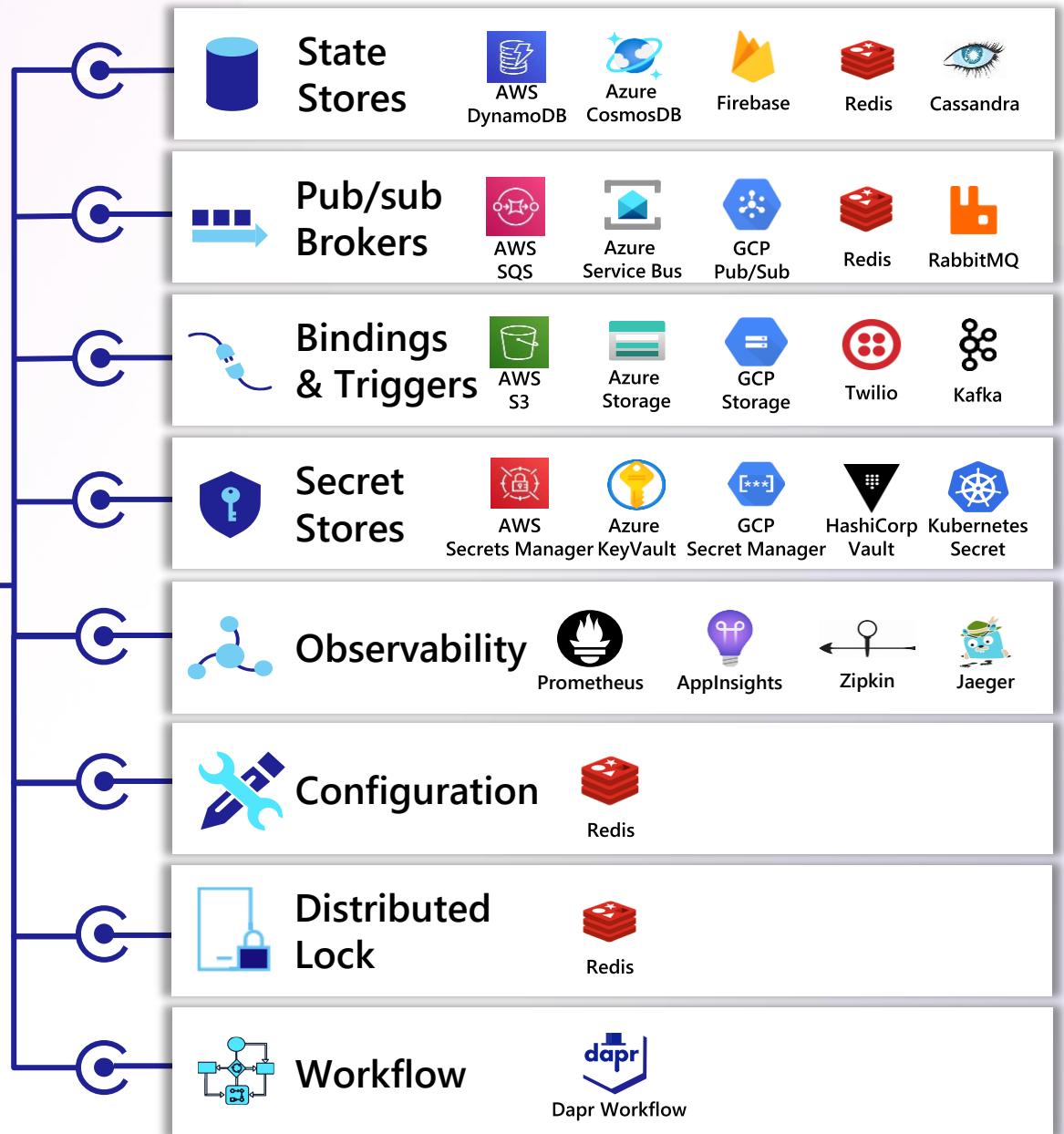
# Dapr APIs



# Dapr components

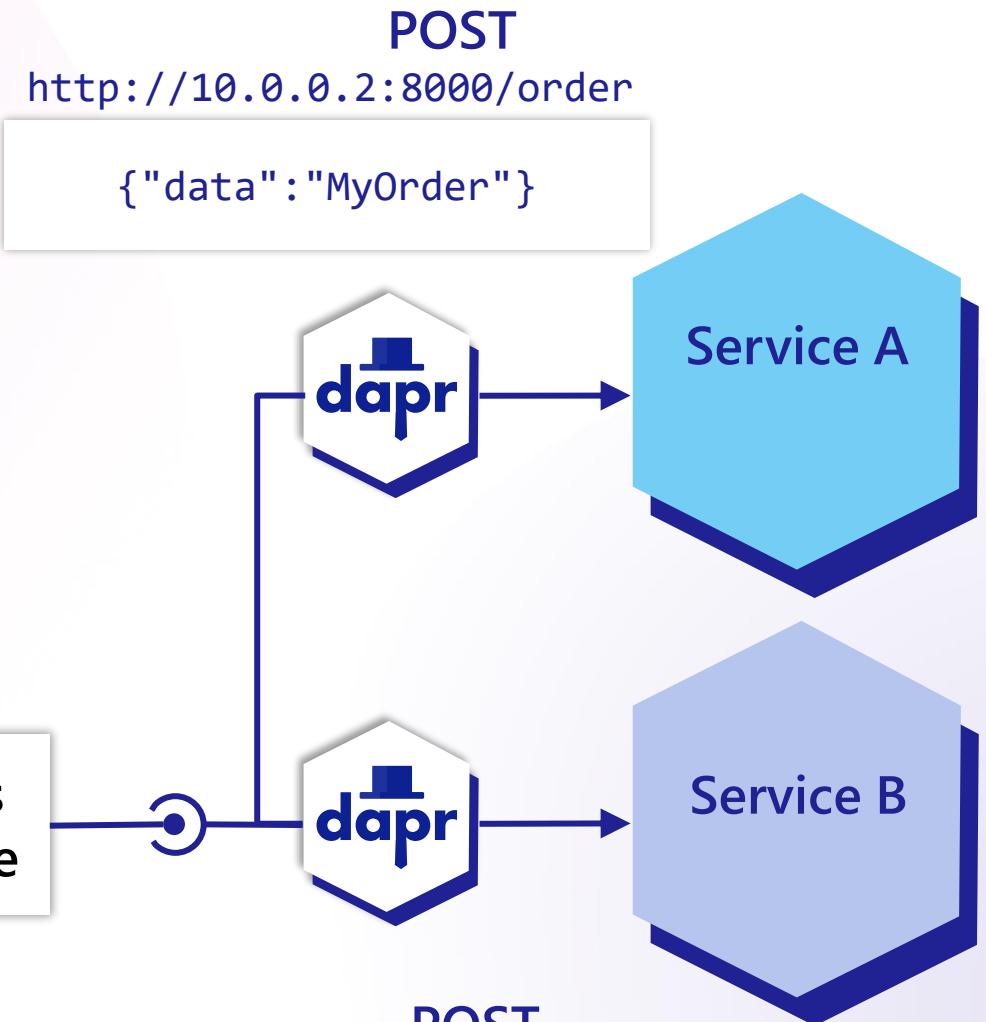
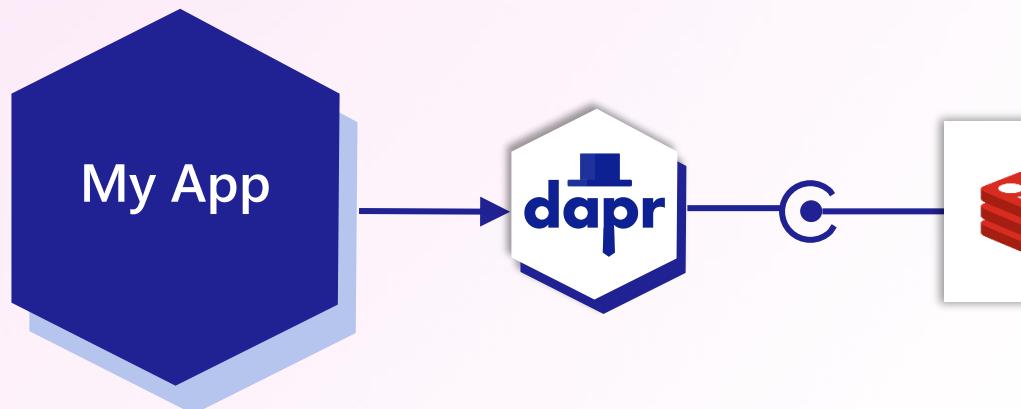


Swappable YAML files with  
resource connection metadata  
Over 100 components available



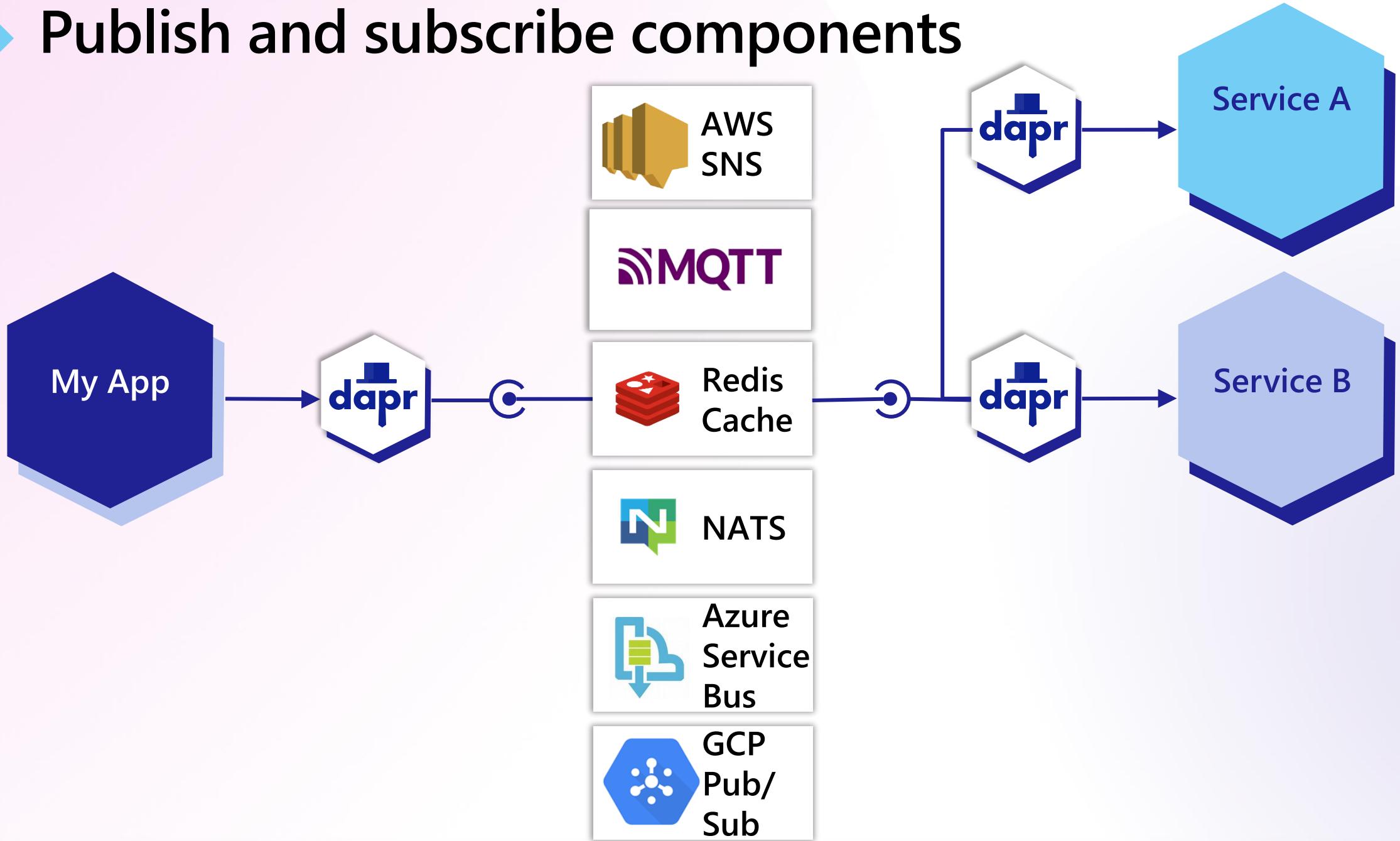


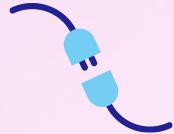
# Publish and subscribe



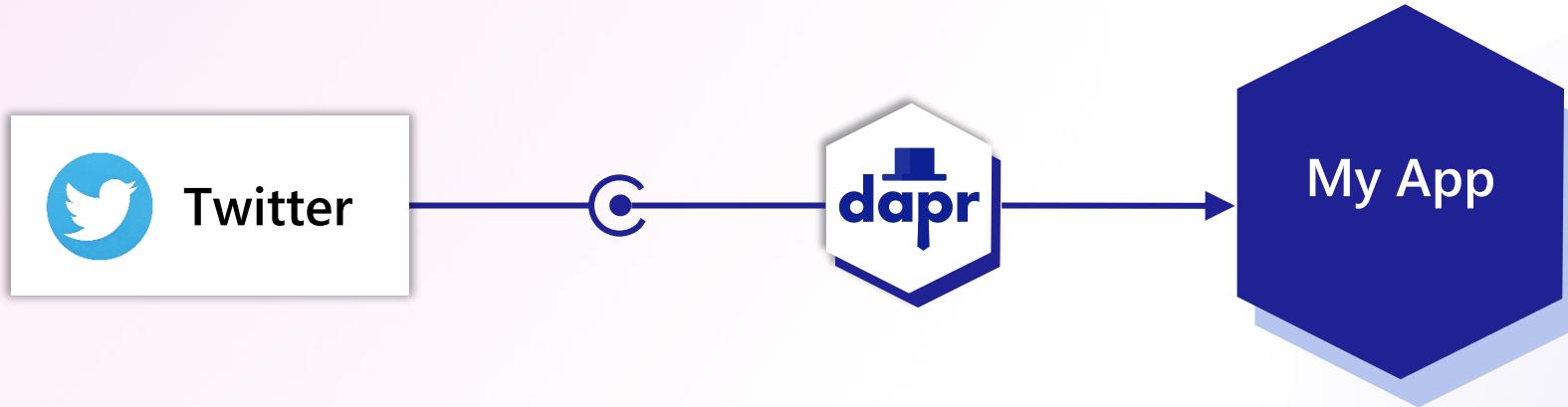


# Publish and subscribe components





# Input bindings



**POST**  
**<http://10.0.0.2:8000/newtweet>**

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

## twitter.yaml

# Dapr bindings API

App-to-sidecar

Invoke an output binding

POST/PUT /v1.0/bindings/twitter

Sidecar-to-app

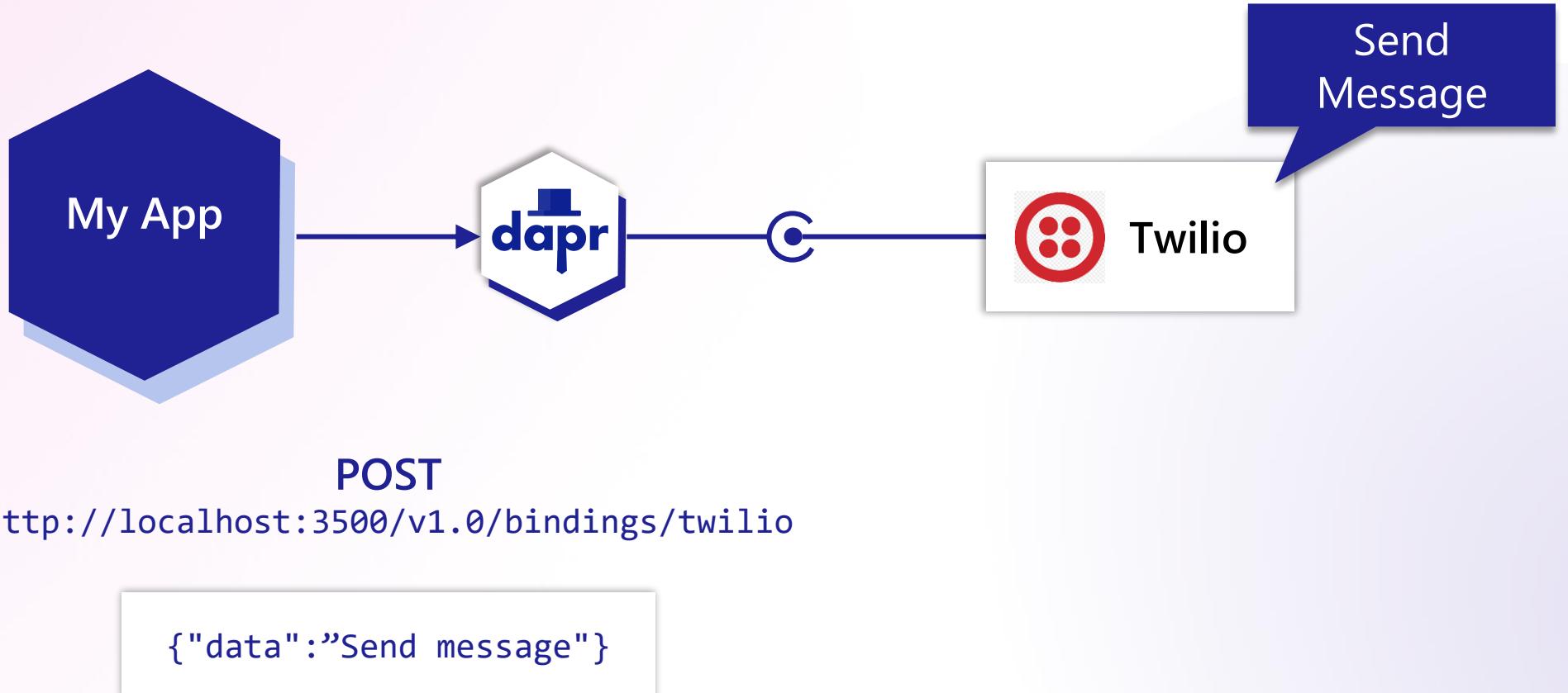
Trigger an app

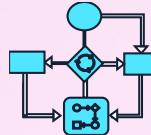
OPTIONS/POST /new-tweet

```
apiVersion: dapr.io/v1alpha1
kind: Component
metadata:
  name: twitter
spec:
  type: bindings.twitter
  version: v1
  metadata:
    - name: consumerKey
      secretKeyRef:
        name: twitter-secret
        key: consumerKeys
    - name: consumerSecret
      secretKeyRef:
        name: twitter-secret
        key: consumerSecret
    - name: accessToken
      secretKeyRef:
        name: twitter-secret
        key: accessToken
    - name: accessSecret
      secretKeyRef:
        name: twitter-secret
        key: accessSecret
```



# Output bindings

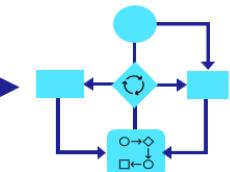




# Workflow APIs



Dapr API  
HTTP/gRPC



Dapr Workflow

POST  
<http://localhost:3500/v1.0-alpha1/workflows/{workflowComponent}/{workflowType}/{instanceID}/start>

```
[  
  {  
    "workflow_options" :  
      {  
        "key" : "value"  
      },  
    "input" : User-Specified value  
  }]
```

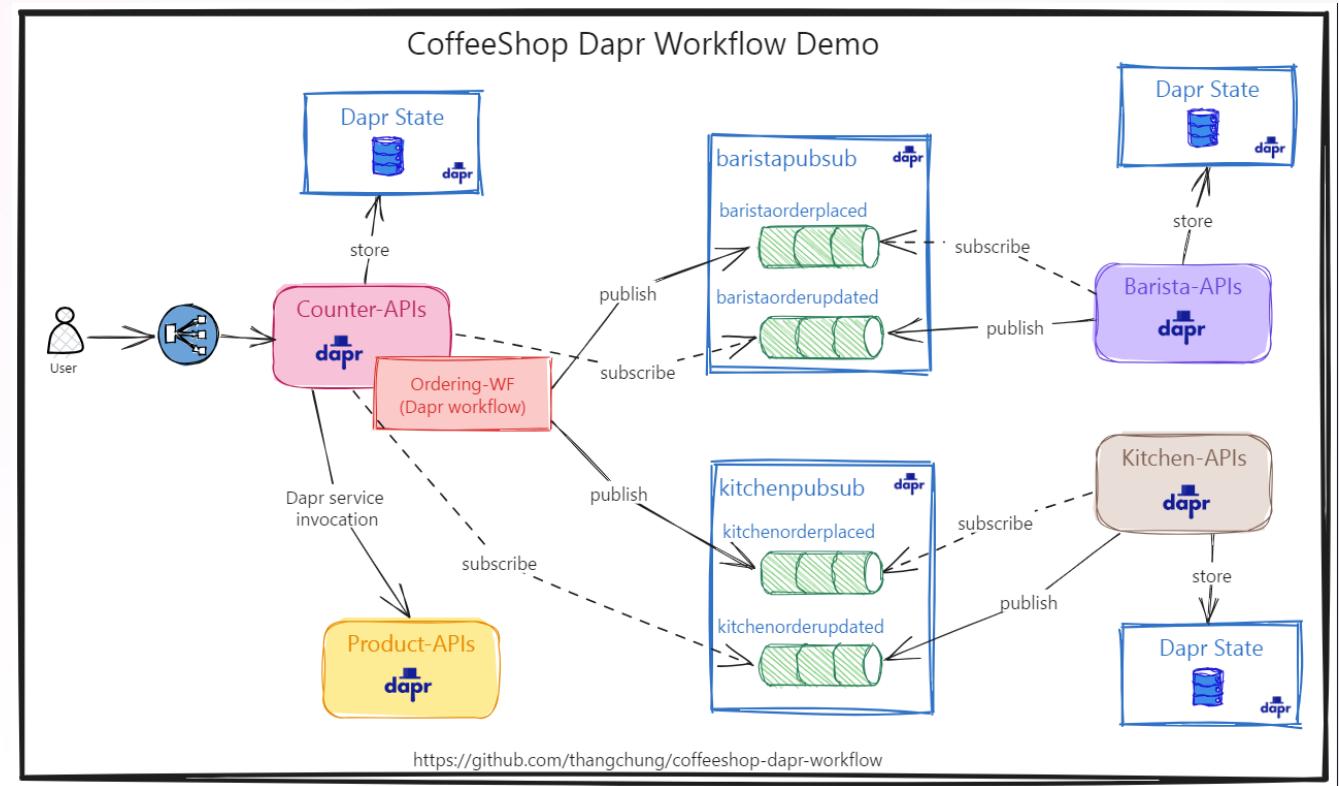
POST  
<http://localhost:3500/v1.0-alpha1/workflows/{workflowComponent}/{workflowType}/{instanceID}/terminate>

GET  
<http://localhost:3500/v1.0-alpha1/workflows/{workflowComponent}/{workflowType}/{instanceID}>

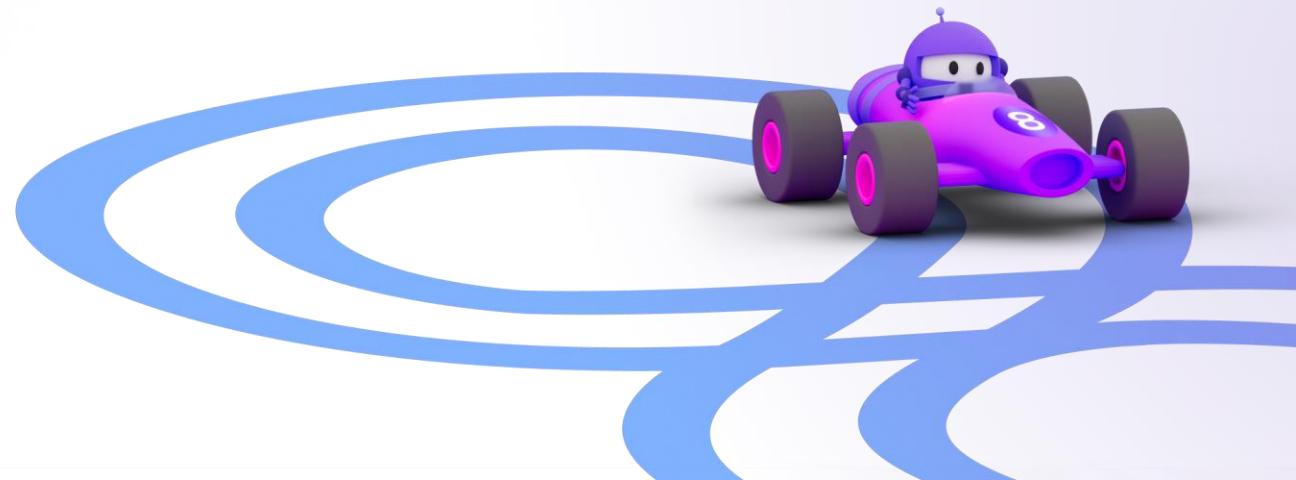
```
{  
  "WFIInfo": {  
    "instance_id": "{instanceID}"  
  },  
  "start_time": "2023-01-12T21:31:13Z",  
  "metadata": {  
    "status": "Running",  
    "key": "value"  
  }  
}
```

# Demo

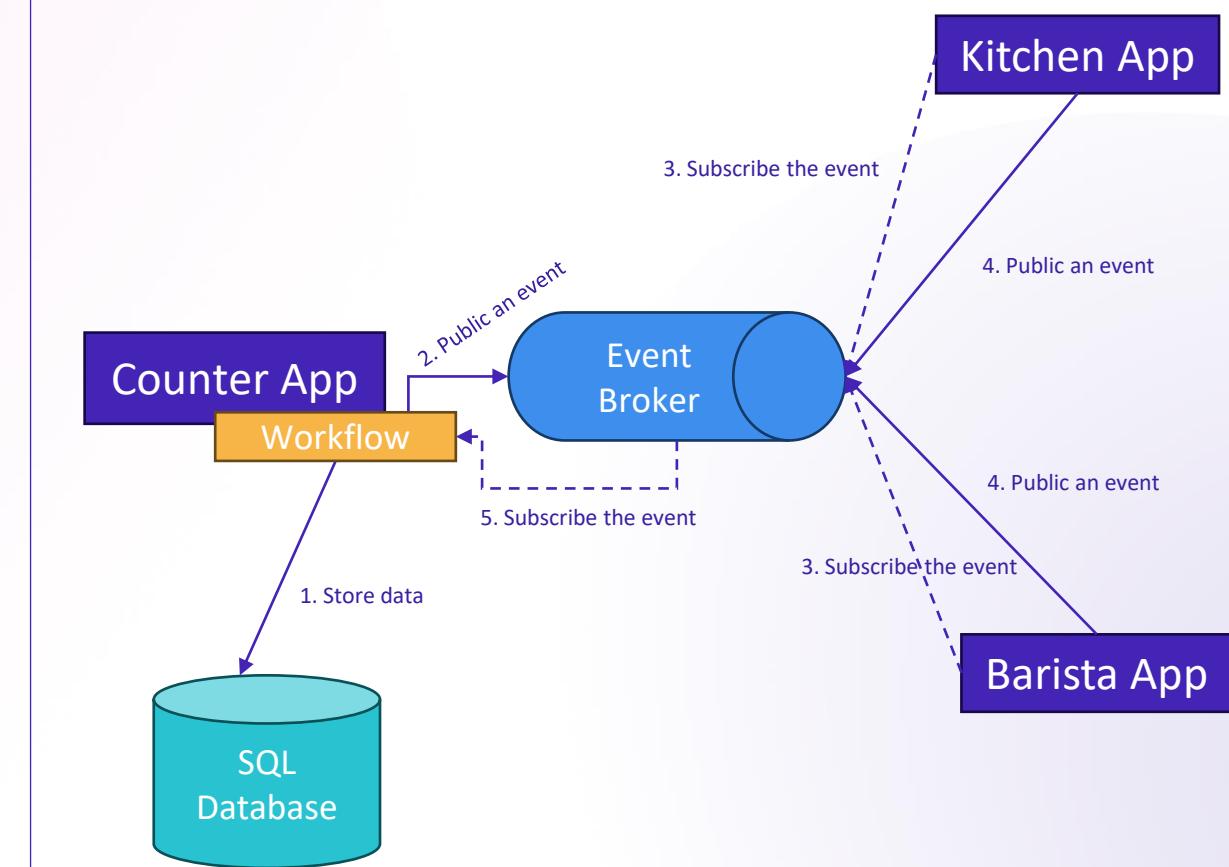
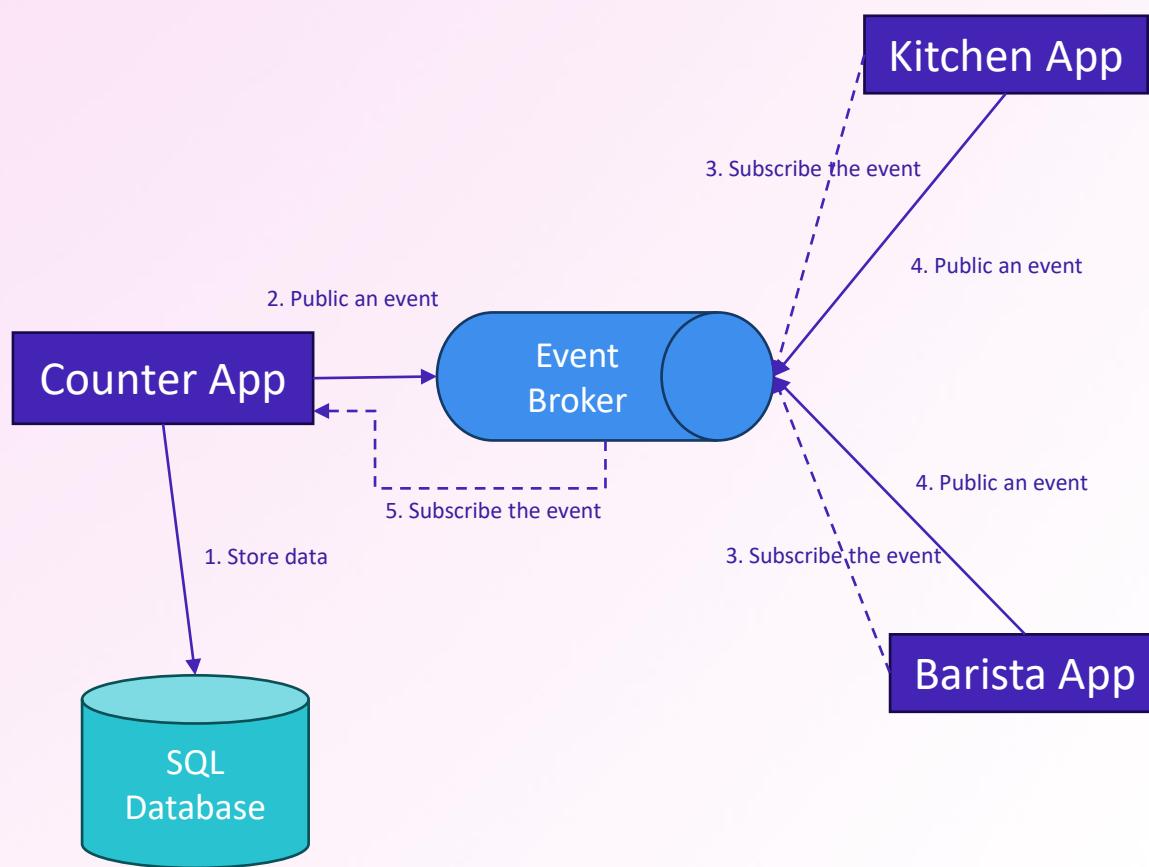
The .NET 8 Event-driven Apps  
with Dapr



# Common Pitfalls & Solutions

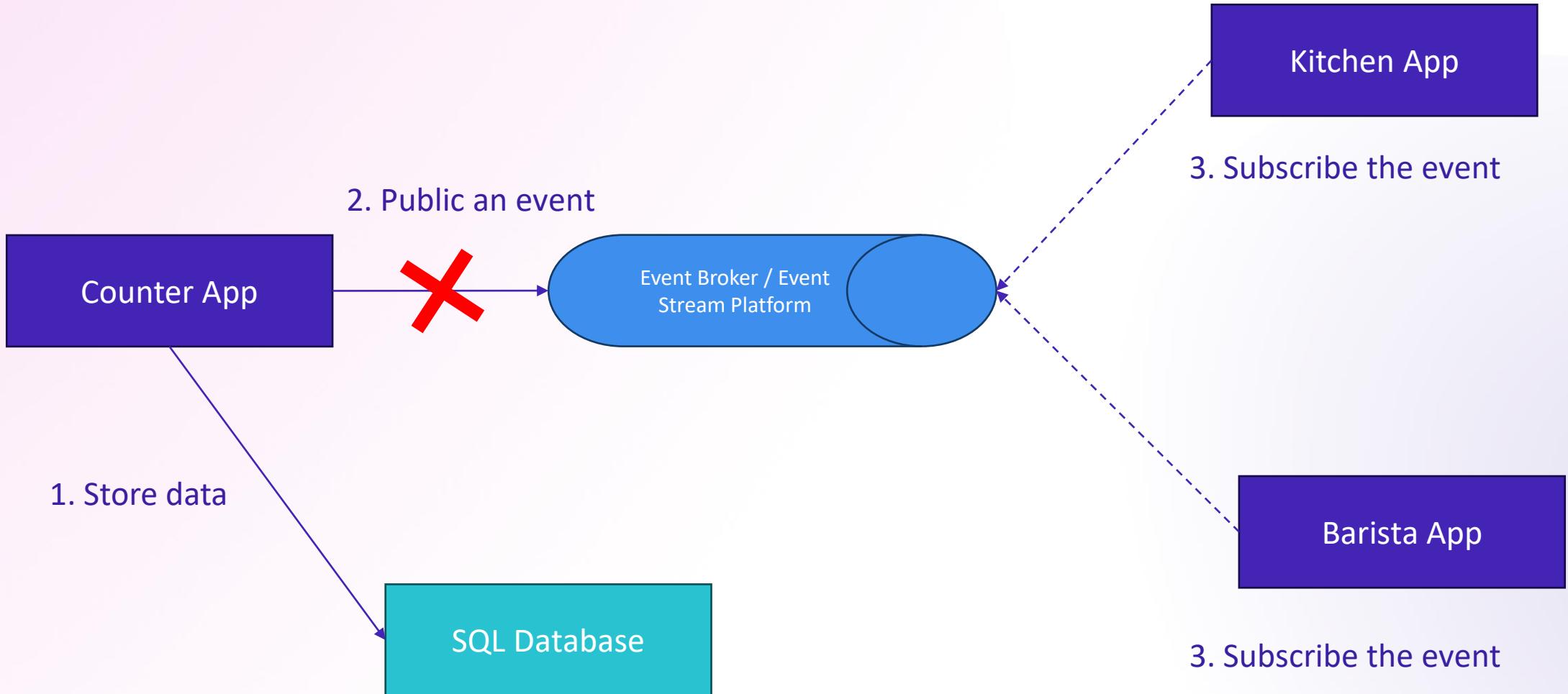


# 1 Choreography vs Orchestration Sagas?

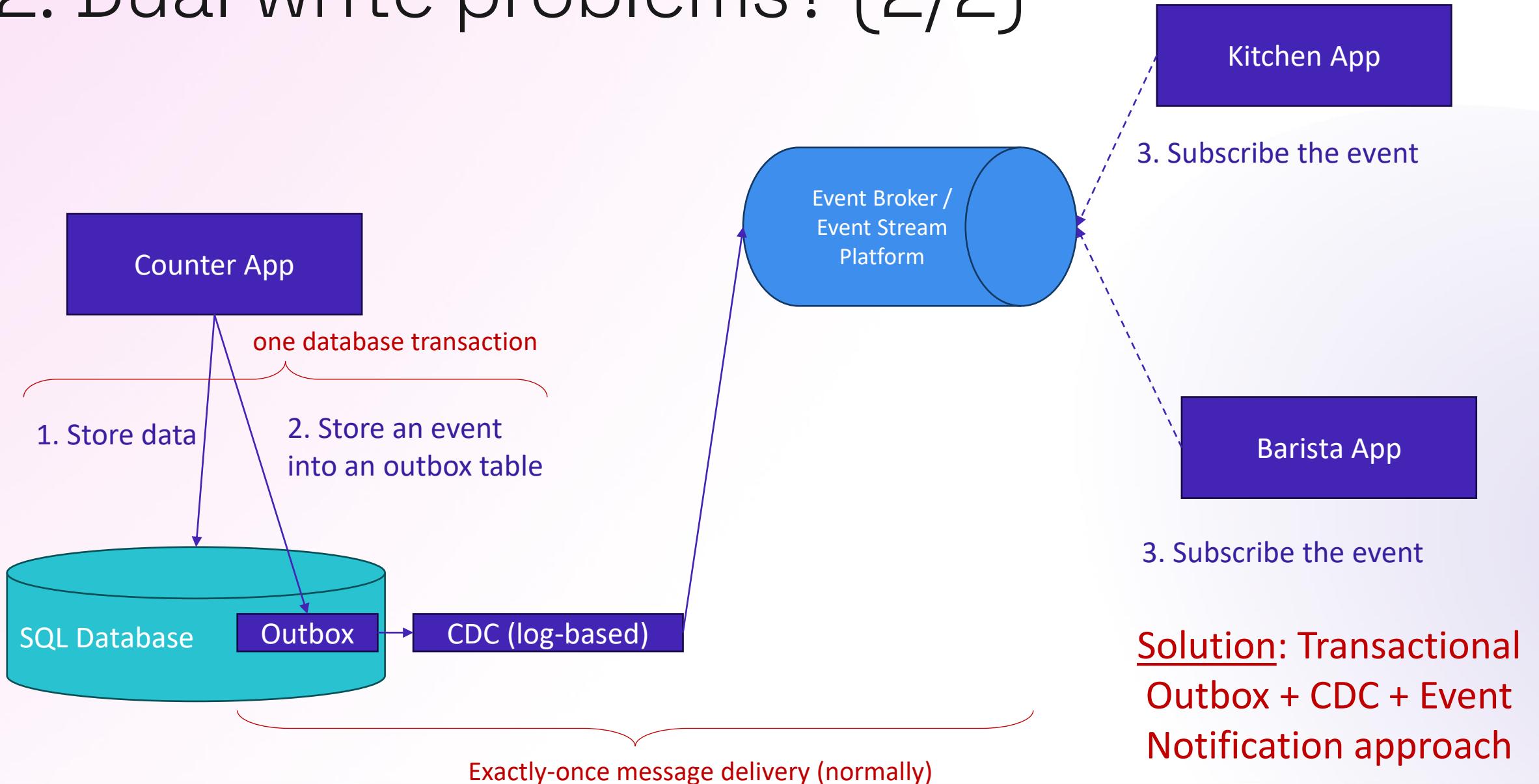


Rule of thumb: “Use orchestration within the bounded context of a microservice, but use choreography between bounded-contexts”

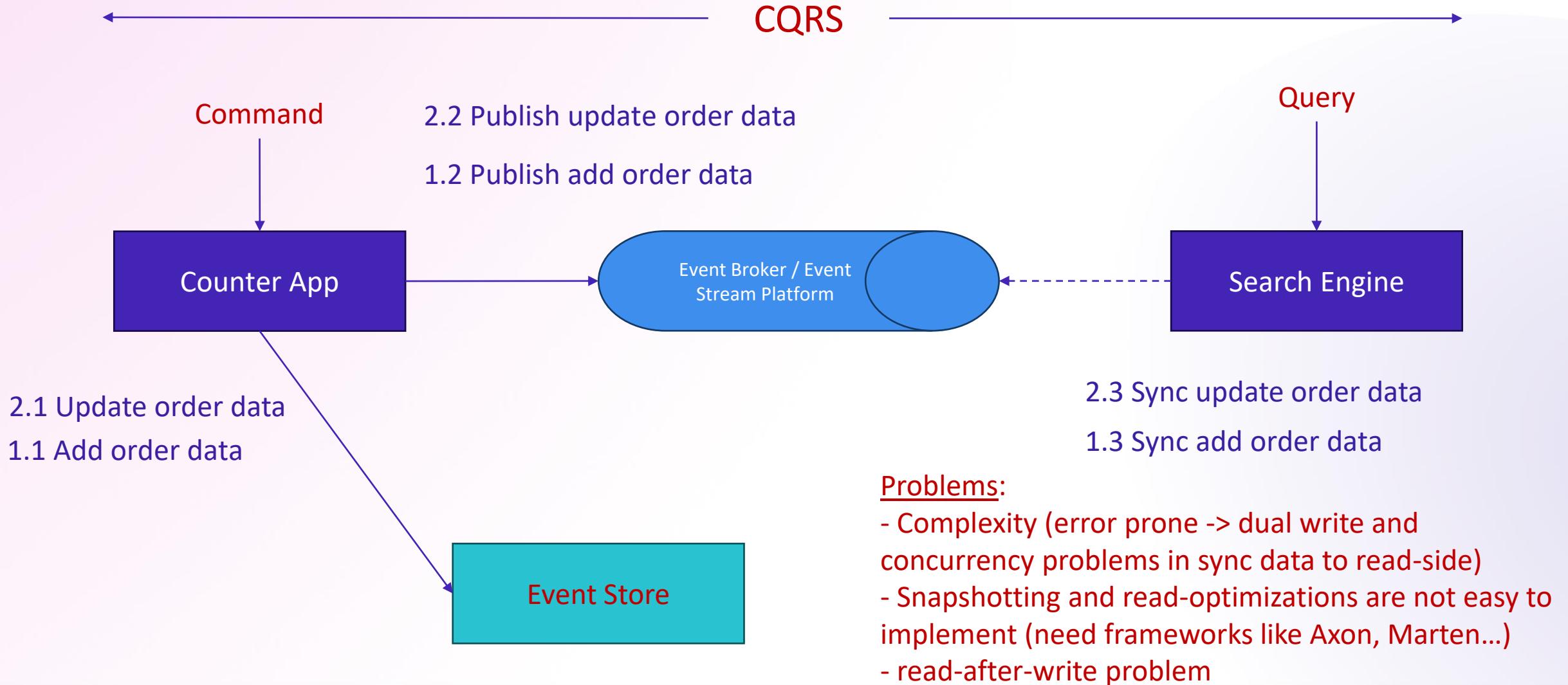
## 2. Dual write problems? (1/2)



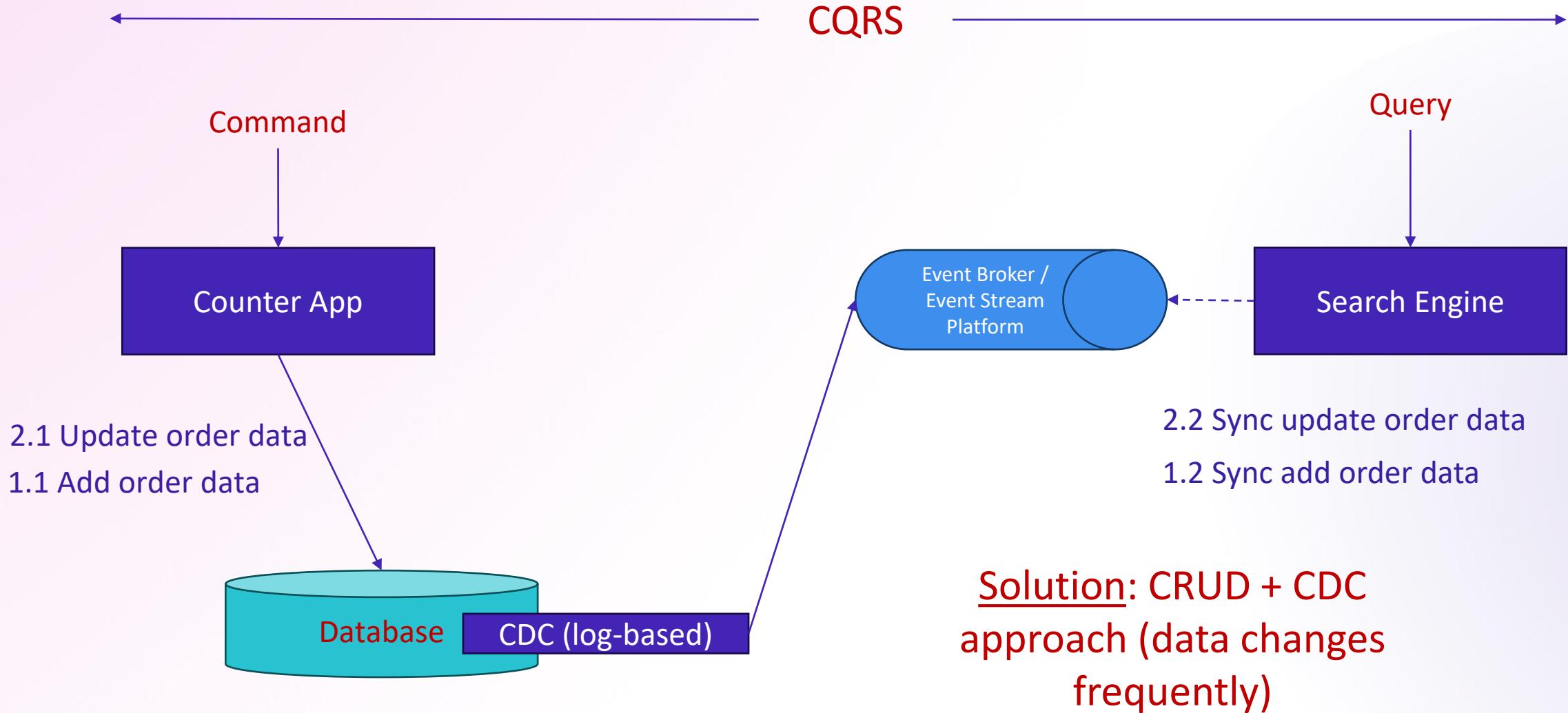
## 2. Dual write problems? (2/2)



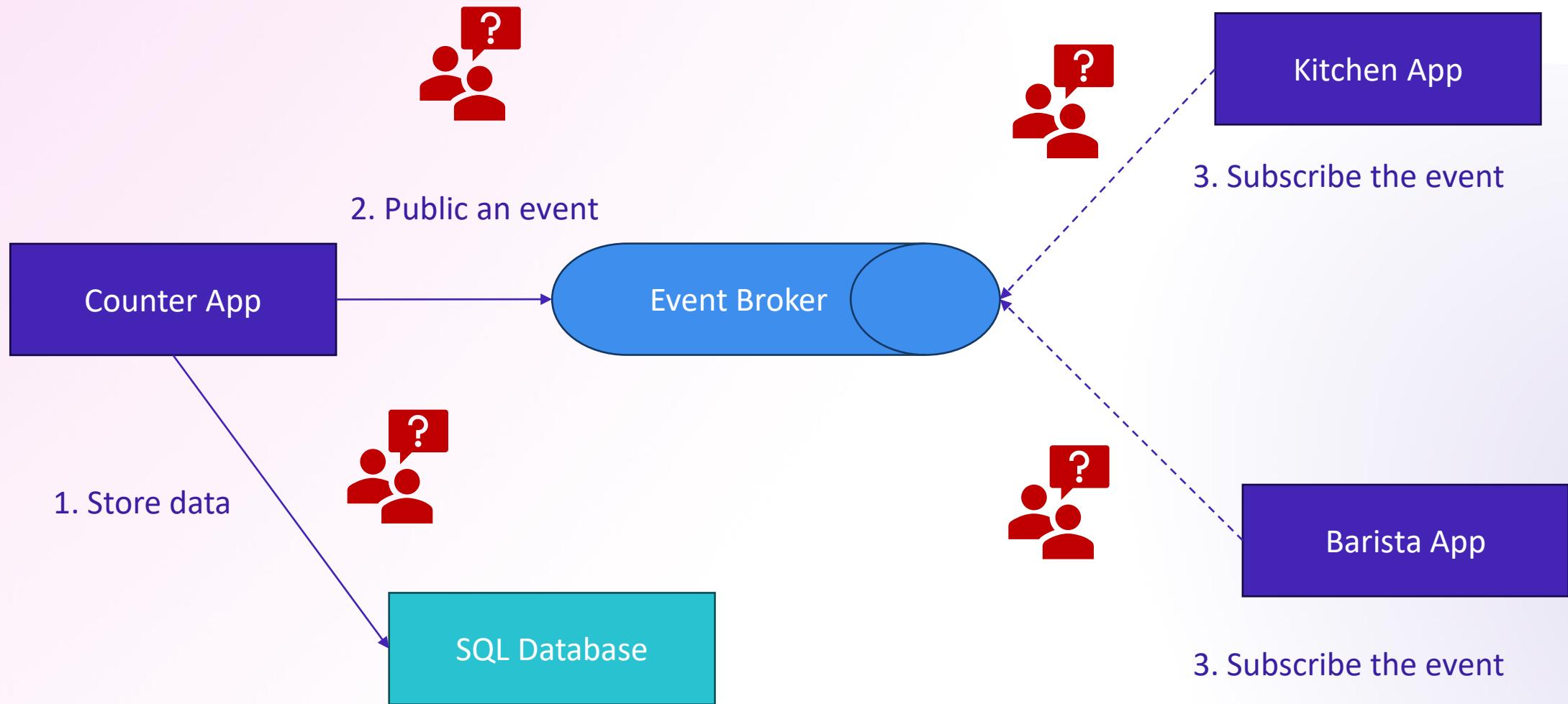
# 3. Event Sourcing everywhere (1/2)



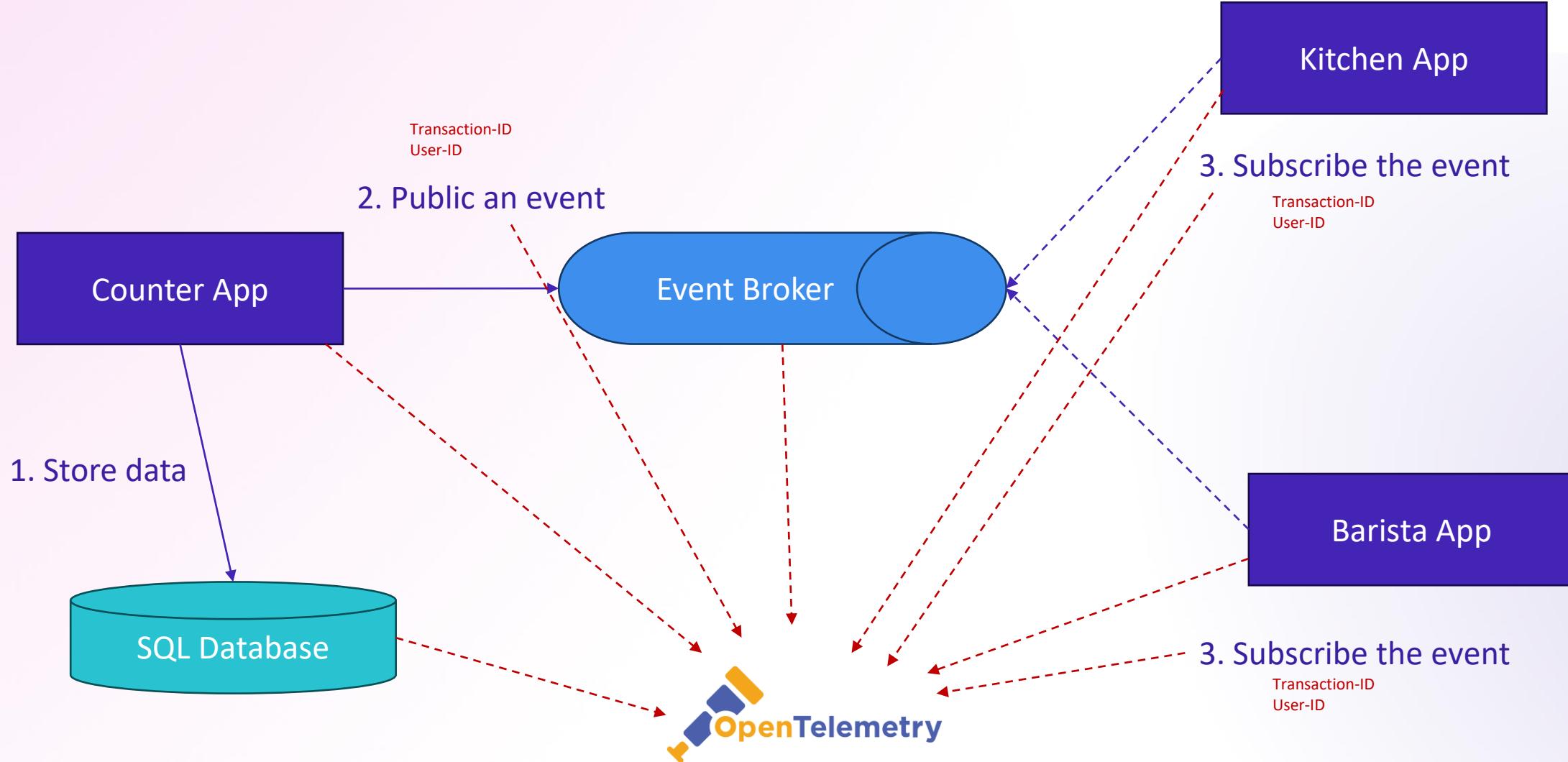
# 3. Event Sourcing everywhere (2/2)



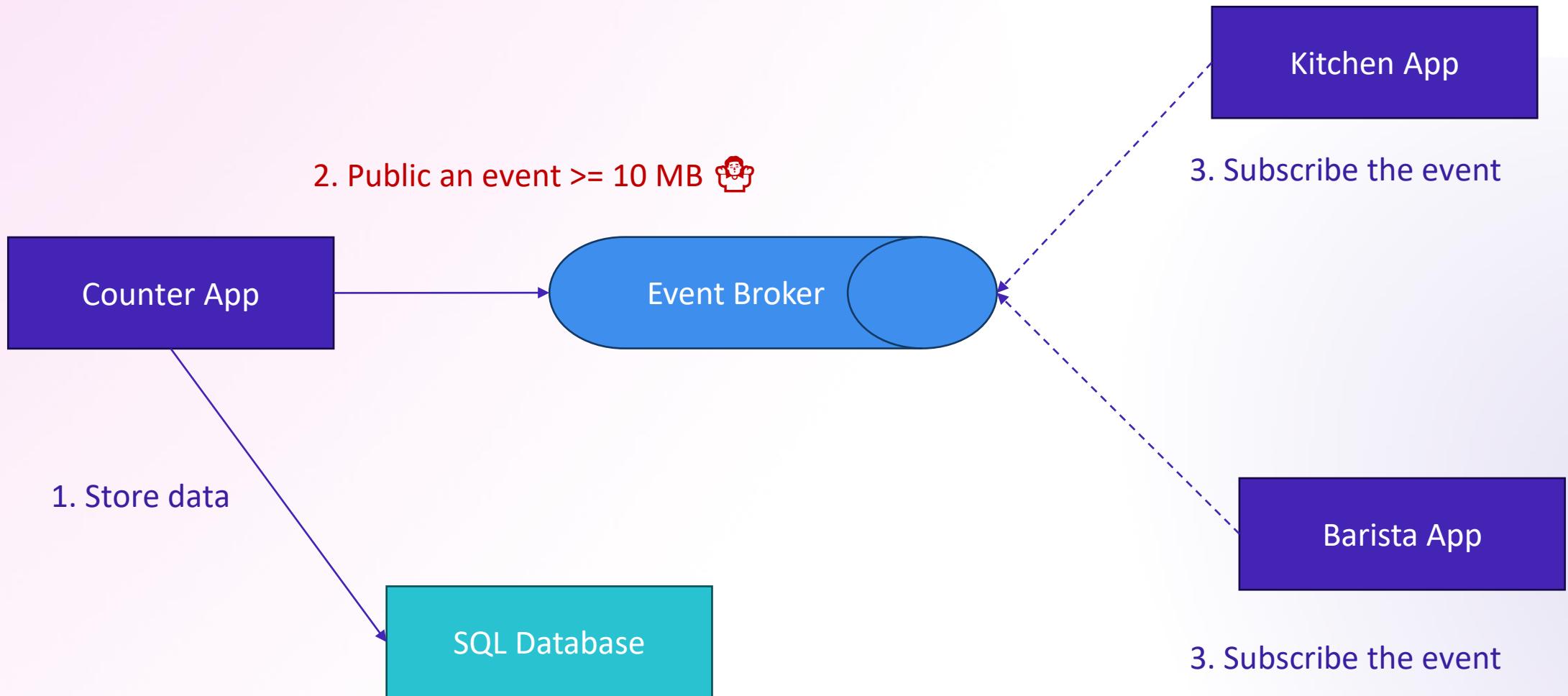
# 4. No context propagation (Transaction-ID) (1/2)



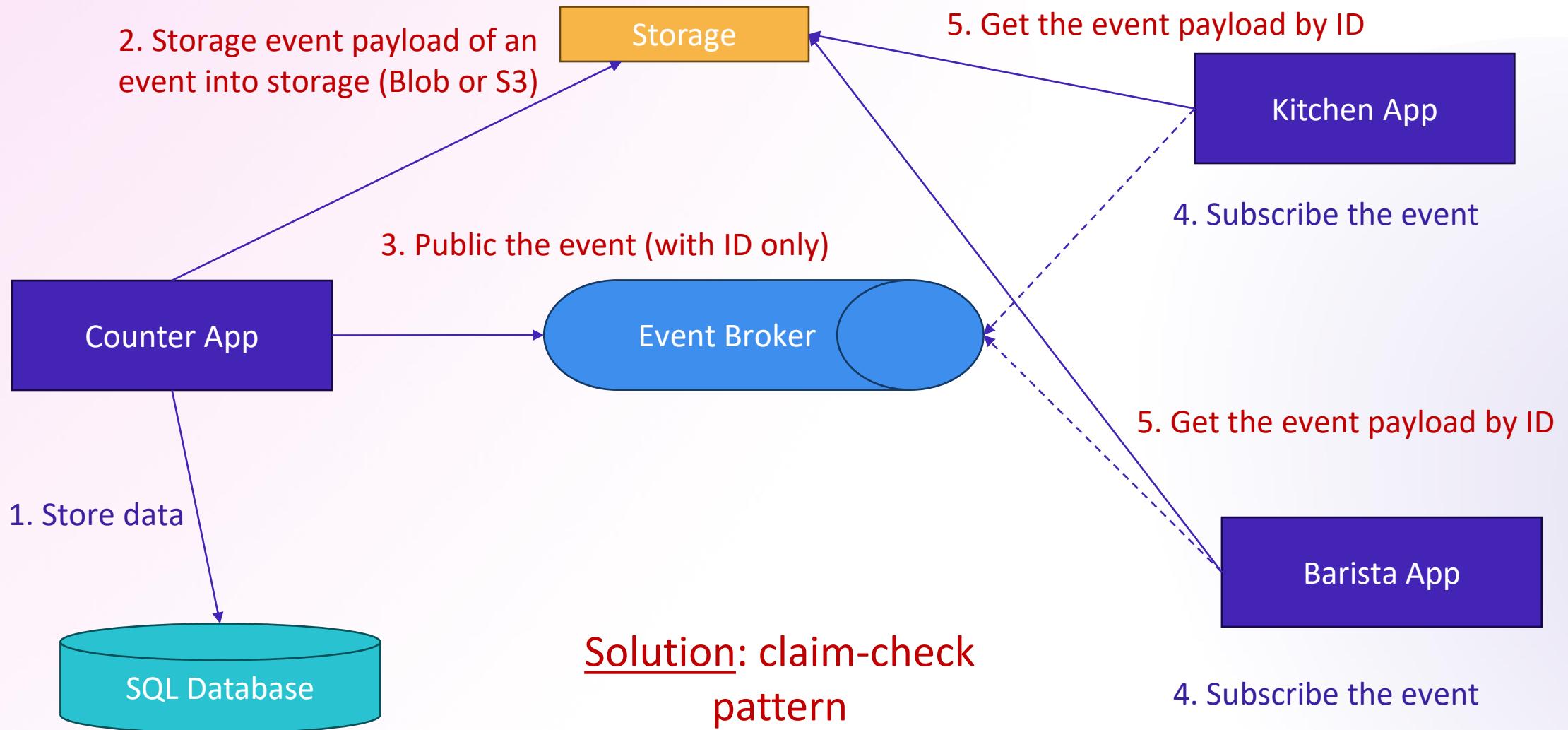
# 4. No context propagation (Transaction-ID) (2/2)



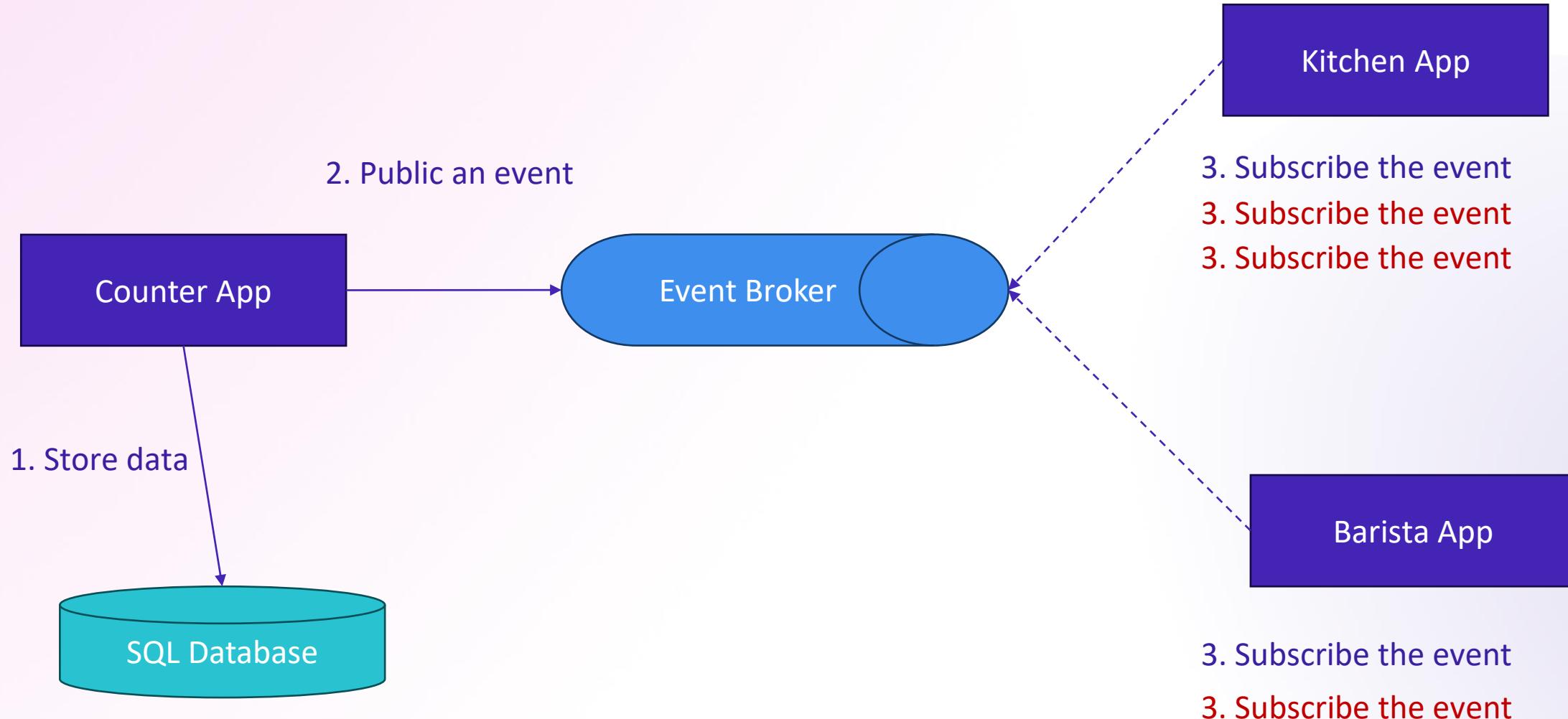
# 5. Publish events with large payloads (1/2)



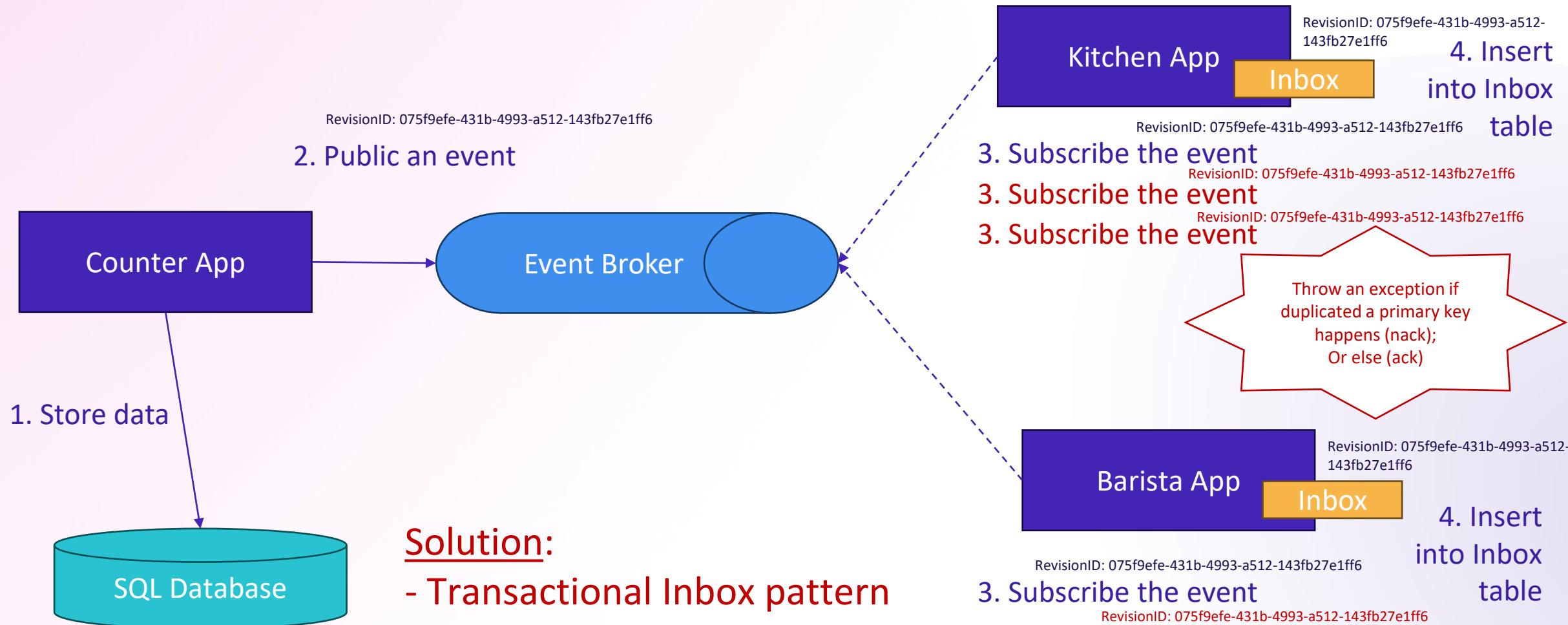
# 5. Publish events with large payloads (2/2)



# 6. Idempotent (duplicated events) (1/2)



# 6. Idempotent (duplicated events) (2/2)



# Appendix.

# .NET Aspire



.NET Aspire

```
// GitHub      🎉: @IEvangelist
// X          🚧: @davidpine7
// .NET Aspire 📁: aka.ms/dotnet/aspire/docs

// Create a distributed app builder.
var builder = DistributedApplication.CreateBuilder(args);

// Add a Redis container resource to the app model.
var cache = builder.AddRedisContainer("cache");

// Add an ASP.NET Core Minimal API project resource.
var apiservice = builder.AddProject<Projects.AspireApp_ApiService>("apiservice");

// Add an ASP.NET Core Blazor web app that references and depends on both the
// cache and API service.
builder.AddProject<Projects.AspireApp_Web>("webfrontend")
    .WithReference(cache)
    .WithReference(apiservice);

builder.Build().Run();
```

A cloud ready stack for building observable,  
production ready, distributed applications

- **Orchestration:** .NET Aspire provides features for running and connecting multi-project applications and their dependencies.
  - **Components:** .NET Aspire components are NuGet packages for commonly used services, such as Redis or Postgres, with standardized interfaces ensuring they connect consistently and seamlessly with your app.
  - **Tooling:** .NET Aspire comes with project templates and tooling experiences for Visual Studio and the dotnet CLI help you create and interact with .NET Aspire apps.
- 
- <https://github.com/dotnet/aspire>
  - <https://github.com/dotnet/aspire-samples>

# Project Radius

The screenshot shows the Project Radius website. At the top, there's a navigation bar with links for Docs, Blog, GitHub, and Community, and a prominent red "Get Started" button. Below the navigation is a large heading: "Cloud-native applications are more than just Kubernetes". A subtext below it reads: "Radius is an open-source, cloud-native, application platform that enables developers and the operators that support them to define, deploy, and collaborate on cloud-native applications across public clouds and private infrastructure". At the bottom of the page, there's a grid of icons representing various cloud-native technologies and services, including Kubernetes, Docker, AWS, and others.

- Developer + Operator Collaboration
  - Application Graph
  - Infrastructure Recipes
  - Cloud Neutral
  - Leverage Existing Tools
- 
- <https://radapp.io/>
  - <https://github.com/radius-project>

# Better observability, resiliency, scalability, manageability, and more...



Observability



Resilience



Health Checks



Testing/Fakes

`Extensions.Resilience`

`Extensions.Compliance.Redaction`

`Extensions.Http.Resilience`

`Extensions.Http.Telemetry`

`Extensions.Diagnostics.HealthChecks.Common`

`AspNetCore.Testing`

`Extensions.Diagnostics.Probes`

`Hosting.Testing`

`Extensions.Telemetry`

`Extensions.TimeProvider.Testing`

# .NET 8 Container Enhancements – More secure, compact, and productive



**Hardened**



**Smaller**



**More Productive**

**Non-Root Base Images**

**USER "app"**

**Default port - 8080**

**Mariner distroless**

**AOT base images**

**"Composite" base images**

**"extra" base images**

**Distroless / Chiseled**

**Publish with .NET SDK**

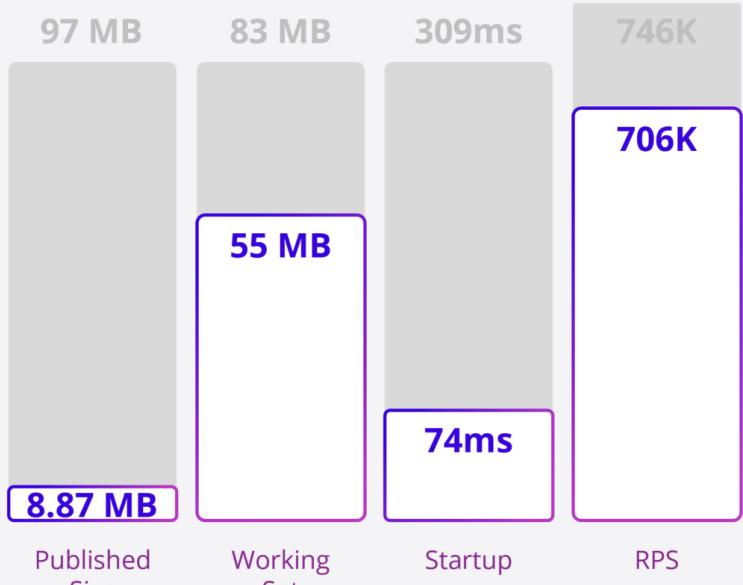
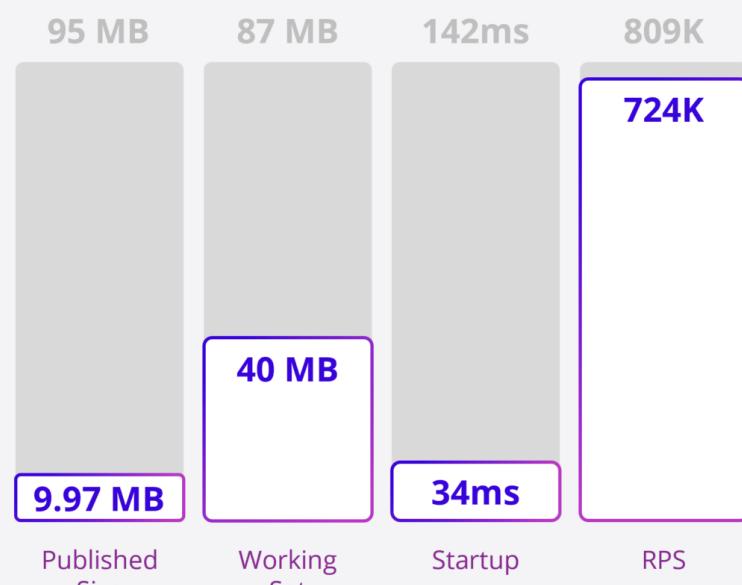
**Cross compilation**

**Non-root by default**

**Supports all Azure auth**

# Native AoT – Journey towards higher density sustainable compute

## .NET 8 – Native AOT Performance



# Q&A

# References

- <https://dot.net>
- <https://dapr.io>
- <https://github.com/dotnet/aspire>
- <https://radapp.io/>
- <https://devblogs.microsoft.com/dotnet/announcing-dotnet-8/>
- <https://medium.com/wix-engineering/event-driven-architecture-5-pitfalls-to-avoid-b3ebf885bdb1>
- <https://amanagrawal.blog/2019/08/27/building-an-event-driven-architecture-lessons-learned/>
- <https://martinfowler.com/articles/201701-event-driven.html>
- <https://theburningmonk.com/2020/08/choreography-vs-orchestration-in-the-land-of-serverless>
- <https://learn.microsoft.com/en-us/azure/architecture/patterns/claim-check>
- <https://exactly-once.github.io/posts/exactly-once-delivery/>
- <https://www.slideshare.net/chris.e.richardson/developing-eventdriven-microservices-with-event-sourcing-cqrs-gotoams>

# Download .NET 8

<https://aka.ms/get-dotnet-8>

