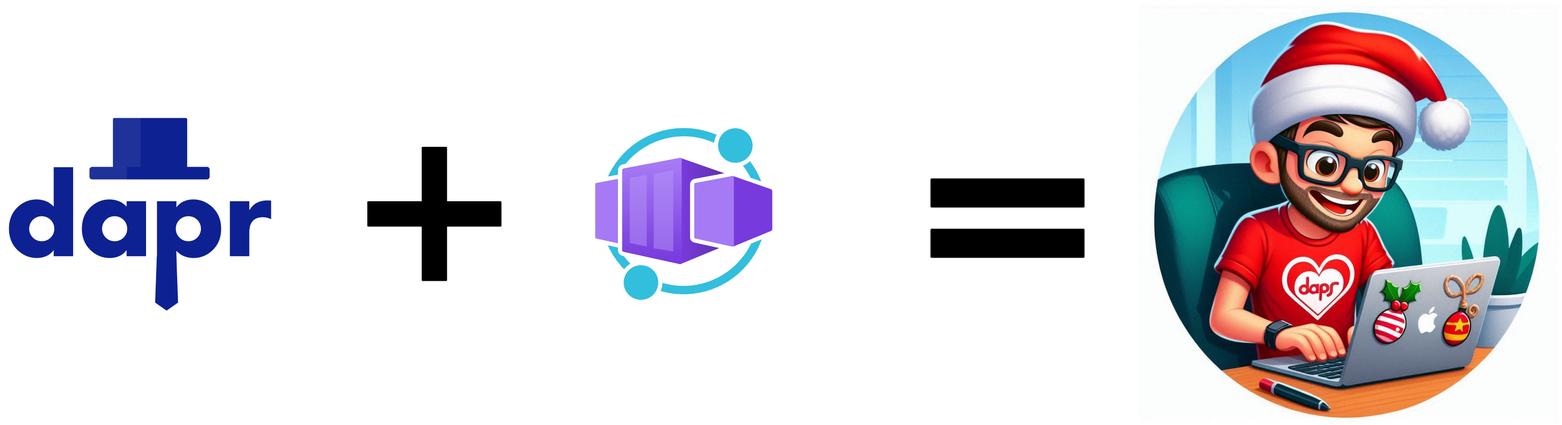
Cloud Native applications with Dapr and Azure Container Apps



Anssi Pannula

Agenda

- What is Dapr and how to use it?
- Azure Container Apps
- Demo - show me the code!

What is Dapr?

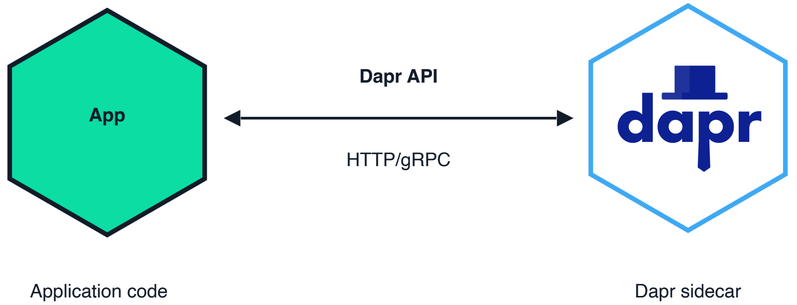
- Dapr stands for Distributed Application Runtime
- It was created at Microsoft in 2019 as part of an incubation project.
- Cloud Native Computing Foundation (CNCF) Incubating project since 2021
  - Dapr graduated to a top-level project in October 2024
- Definitions:

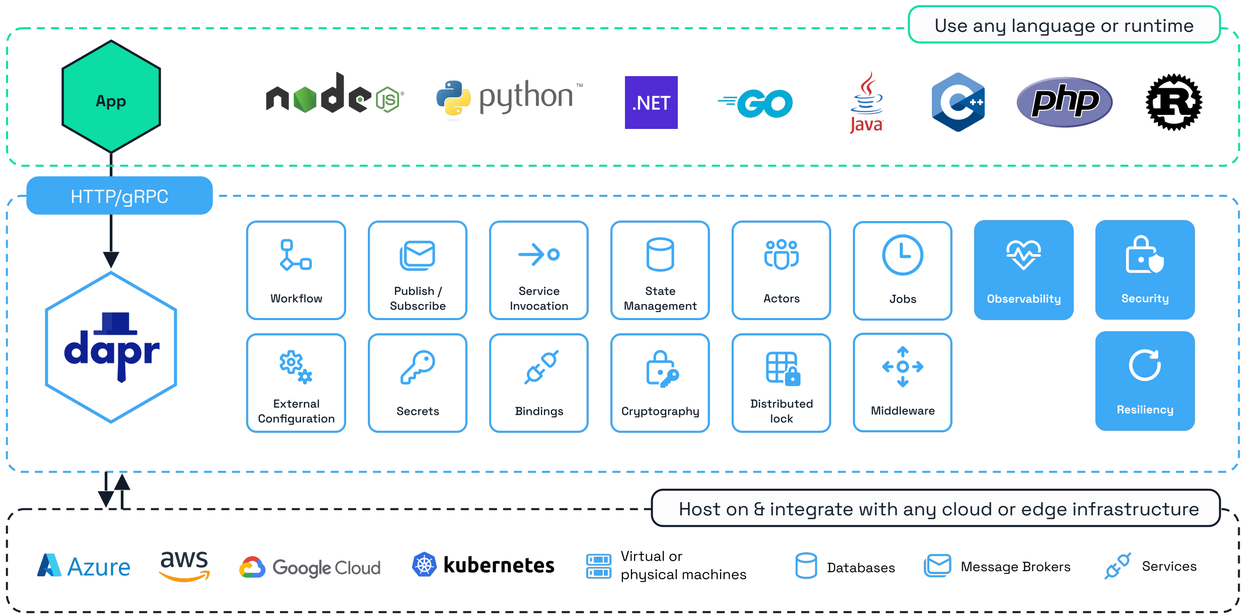
Dapr is a portable, event-driven **runtime** that makes it easy for any developer to **build resilient, stateless and stateful applications** that run on the cloud and edge and embraces the diversity of languages and developer frameworks. Leveraging the benefits of a **sidecar architecture**, Dapr helps you tackle the challenges that come with building **microservices** and keeps your code platform agnostic.

Dapr is a **toolkit** that helps different parts of an application communicate and work together smoothly, making development **easier** and more **efficient**. It handles tasks like service calls, state management, and messaging, so you can **focus on building features.**

What is it really?

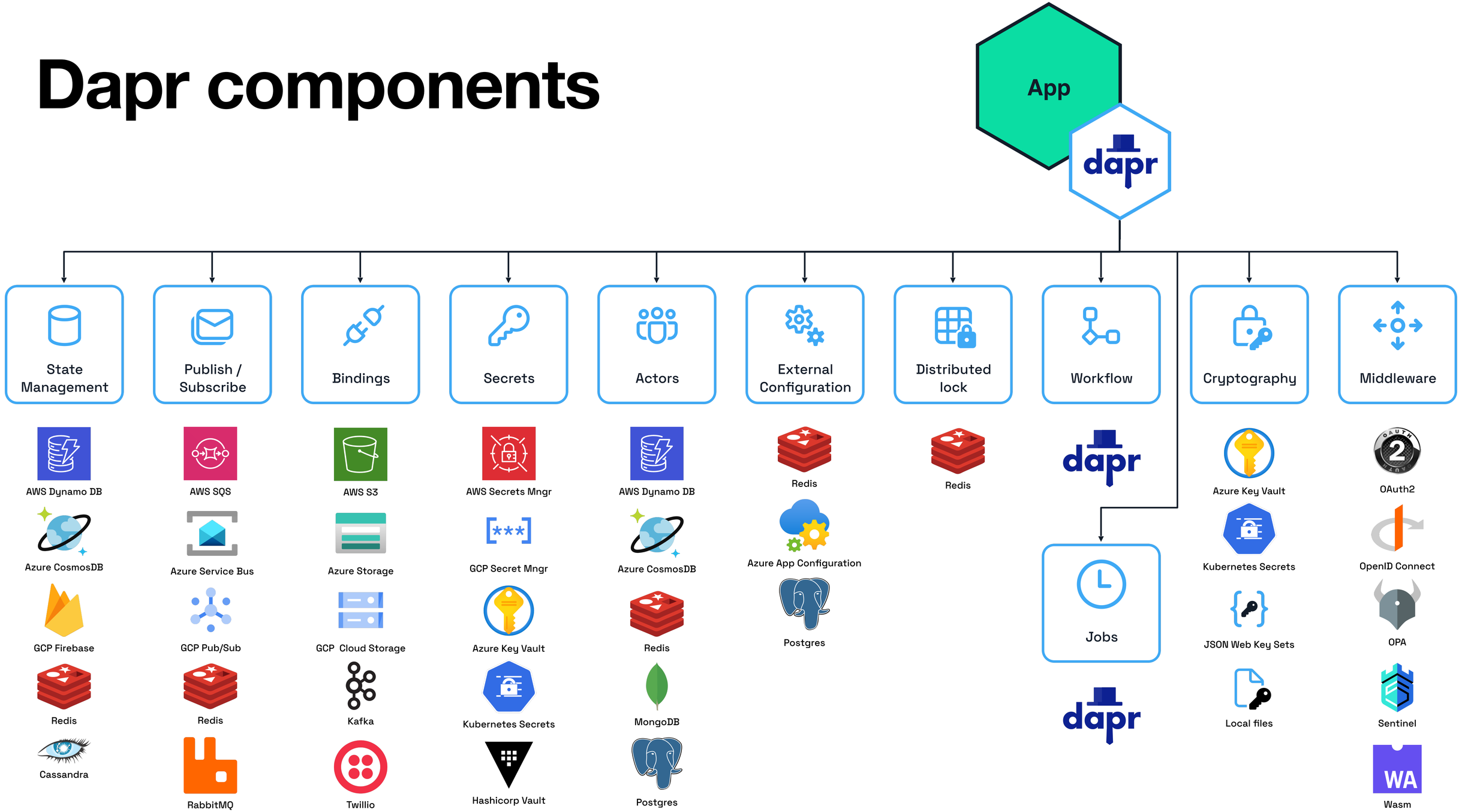
- Tool for building cloud native microservices
- A sidecard process
  - Sidecar pattern
- Use any language, code and framework
- Offers building blocks that enable us to focus on implementing business logic
- Speeds up application development by providing an single set of APIs for communication
- Attempts to address and resolve challenges related to distributed applications
  - Security (mTLS, access policy), service discovery, inter-service communication, resilience (timeout, retry, circuit breaker)





https://dapr.io/

Dapr components



https://dapr.io/

How to use Dapr API in your application?

- Using HTTP client
  - POST http://localhost:3500/v1.0/invoke/service-a/method/mymethod
  - POST http://localhost:3500/v1.0/publish/mybroker/mytopic
  - GET http://localhost:3500/v1.0/secrets/myvault/mytopsecret
- DaprClient from SDK
  - .NET, Python, Java, GO, PHP, JavaScript
- Use your existing gRPC service definition

Hosting modes

- Self-hosted
  - Ideal for local development and testing
  - Run Dapr on your local machine using the Dapr CLI
    - **dapr init** and **dapr run**
- Kubernetes
  - Suitable for production use
  - Deploy dapr on a Kubernetes using **dapr init -k** or use Helm charts
    - Deploys placements, operator, sentry and injector pods.
- Azure Container Apps
  - Suitable for production use
  - Enable Dapr with a single boolean flag in your container app configuration.

Why you should use Dapr?

- Makes it easy for any developer to build cloud-native microservices
  - Increases your developer productivity by 20-40%*
- Language, framework and cloud agnostic
- Cloud Native Computing Foundation (CNCF) project
    - Vendor-neutral
    - Reduces the risk of the project being abandoned
- Large and active community
  - 120 community components
  - 3k contributors
  - 12/157 Largest CNCF project
- Dapr is secure, audited, and enterprise-ready
  - Dapr completed a comprehensive security audit in 2023 and no critical or high-severity issues found.
    - https://blog.dapr.io/posts/2023/09/05/dapr-completes-2023-security-audit-increasing-enterprise-confidence/

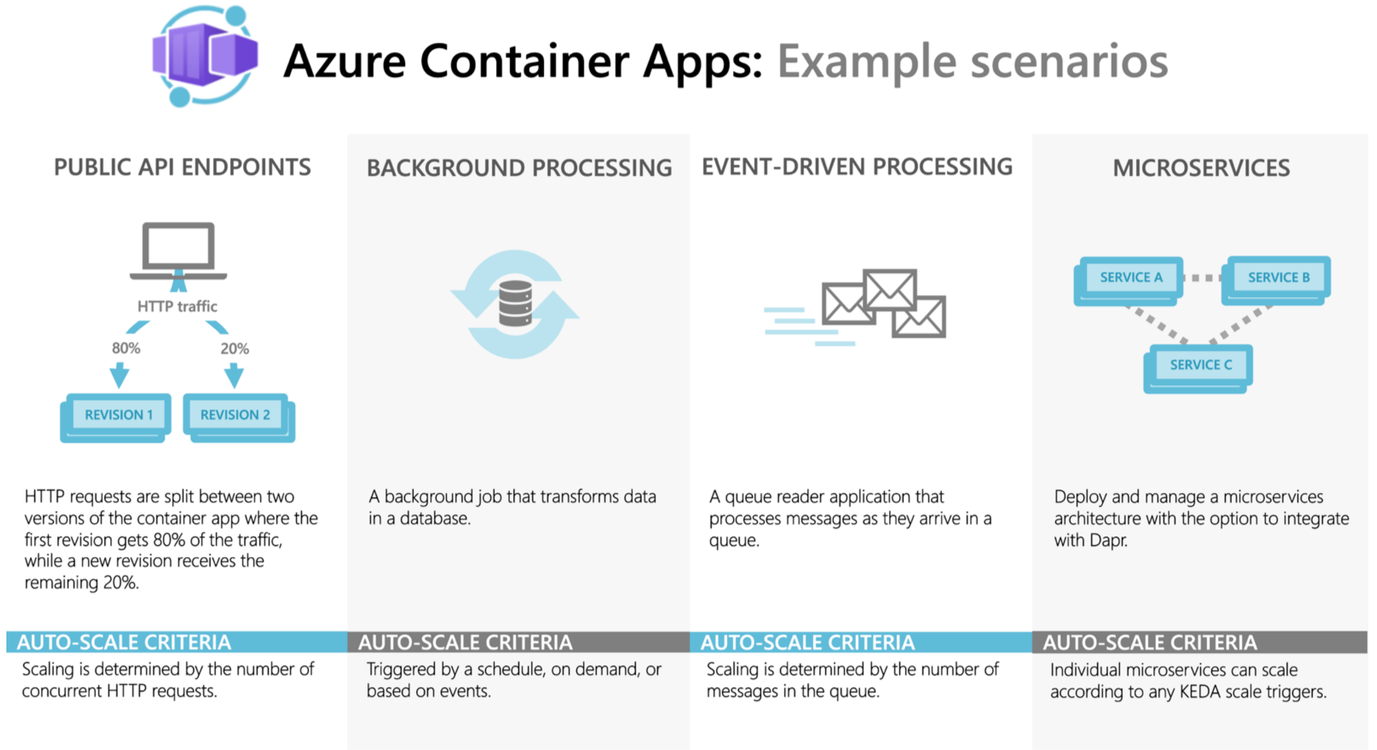
**https://dapr.io*

Case Studies

- HDFC Bank (https://www.cncf.io/case-studies/hdfc-bank/)
  - “Development began in early 2022 and the system went live after one year, becoming one of the **fastest applications to be developed and deployed** to a Tier One Bank.”
- Vonage (https://www.cncf.io/case-studies/vonage/)
  - “Dapr proved to be a phenomenal fit for both our new systems and our existing stacks; whether its written in Java, Dotnet, Golang or Javascript and running on metal, VMs, Nomad or Kubernetes we were able to **standardize tracing, AuthNZ and resilient service-to-service communication**.”
  - 1000+ of development hours saved
- Watts Water Technologies (https://www.cncf.io/case-studies/wattswatertechnologies/)
  - “Dapr allows our development to **focus on business logic and features** that deliver a powerful digital experience and it allows our platform team to focus on other areas that Dapr streamlines”
  - “**Debugging is a breeze with a local Dapr** install which gives developers the confidence they need when deploying new services as well as the hand-off between developers and the DevOps team”
- Derivco (https://www.cncf.io/case-studies/derivco/)
  - “Dapr has also given us the freedom to **rewrite legacy systems** that were previously infeasible to rewrite due to the time it would take to complete.”

Azure Container Apps - Focus on apps, not infrastructure

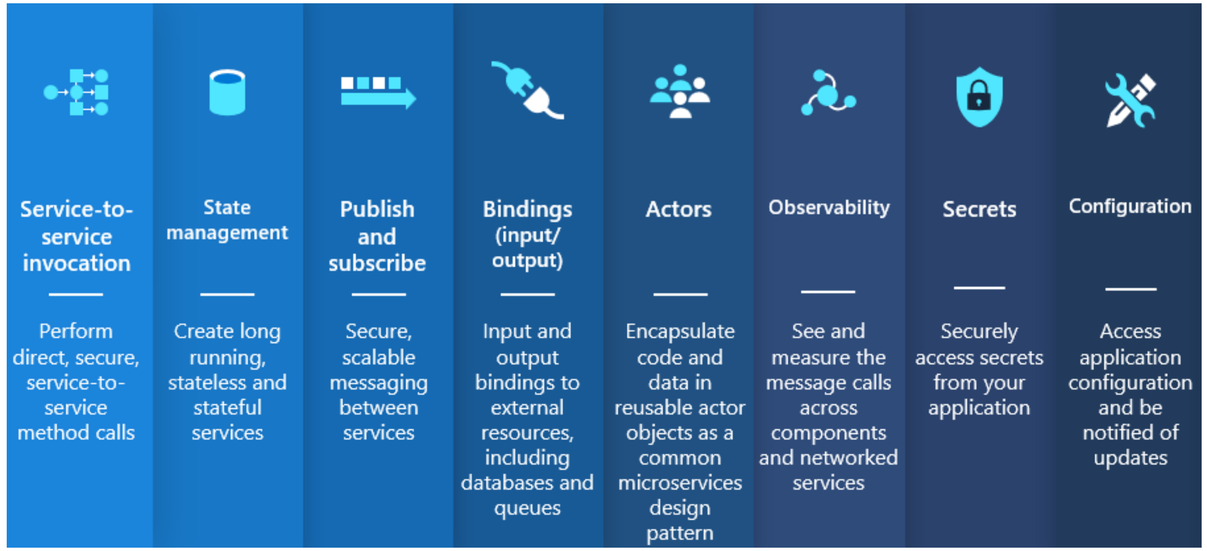
- Fully managed service that allows you to run microservices and containerized applications on a serverless platform
  - KEDA (Kubernetes Event-driven Autoscaling)
    - Provides horizontal scaling (scale out)
  - Envoy proxy
    - Network proxy for all HTTP requests
  - Dapr is “first-class citizen”
    - Version 1.12.15
- “Abstraction layer” on top of Kubernetes
  - Hides the Kubernetes complexities
  - No need for kubectl
- Core Azure resources
  - Container Apps Environment:
    - Secure boundary around one or more container apps
    - Provides infrastructure (CPU,GPU, memory, network etc.) resources
  - Container App
    - Runs and manage containerized applications



https://learn.microsoft.com/en-us/azure/container-apps

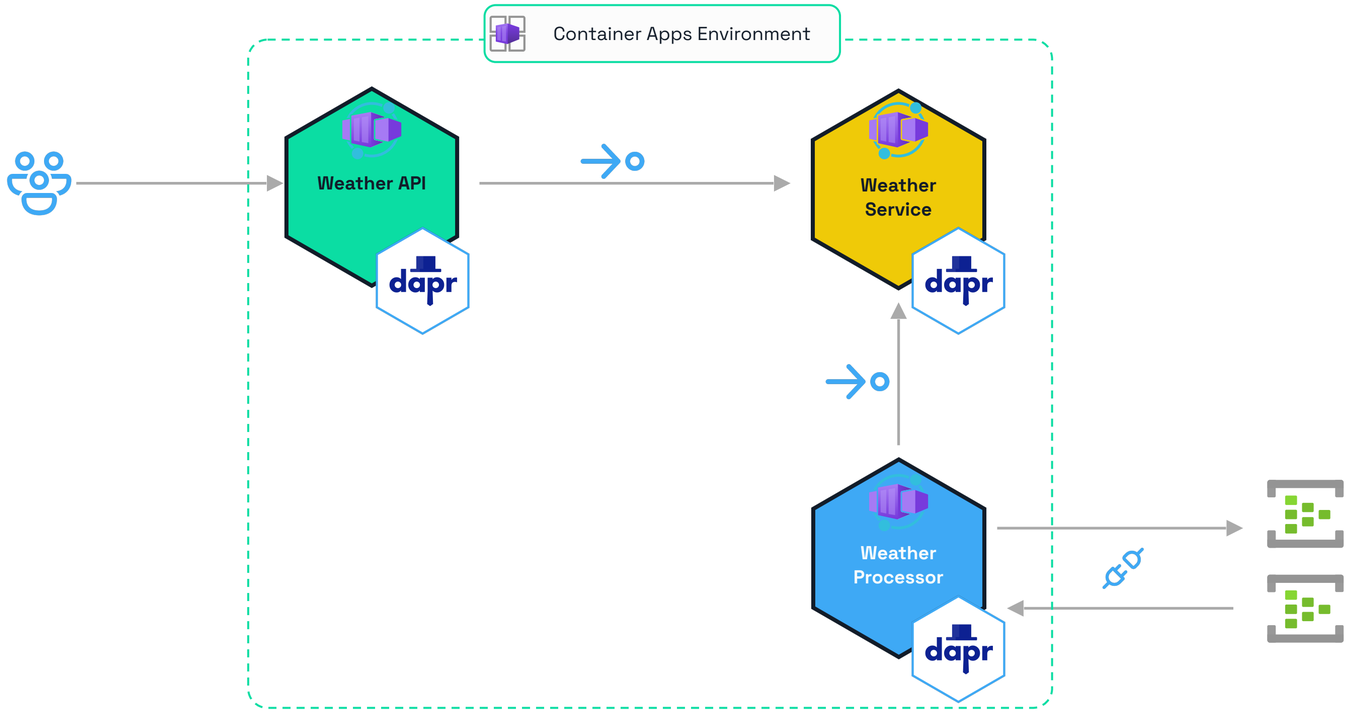
Supported Dapr components

- All the common components are supported
  - Actors
  - Bindings
  - Configuration
  - Publish and subscribe
  - Secrets
  - Service Invocation
  - State management



https://learn.microsoft.com/

Demo


Container Apps Environment
Weather API
dapr
Weather Service
dapr
Weather Processor
dapr

Questions?

Thank you!