Integrando Microservices a LLM com o Dapr

Walter Silvestre Coan

Walter Silvestre Coan

www.linkedin.com/in/waltercoan/

- Microsoft MVP na categoria Internet das Coisas
- Dapr Meteors 2025
- Mestre em Sistemas Distribuídos e Redes de sensores sem fio PUCPR
- Instrutor autorizado Microsoft, AWS, NVIDIA na Ka Solution
- Professor na UNIVILLE

Distributed

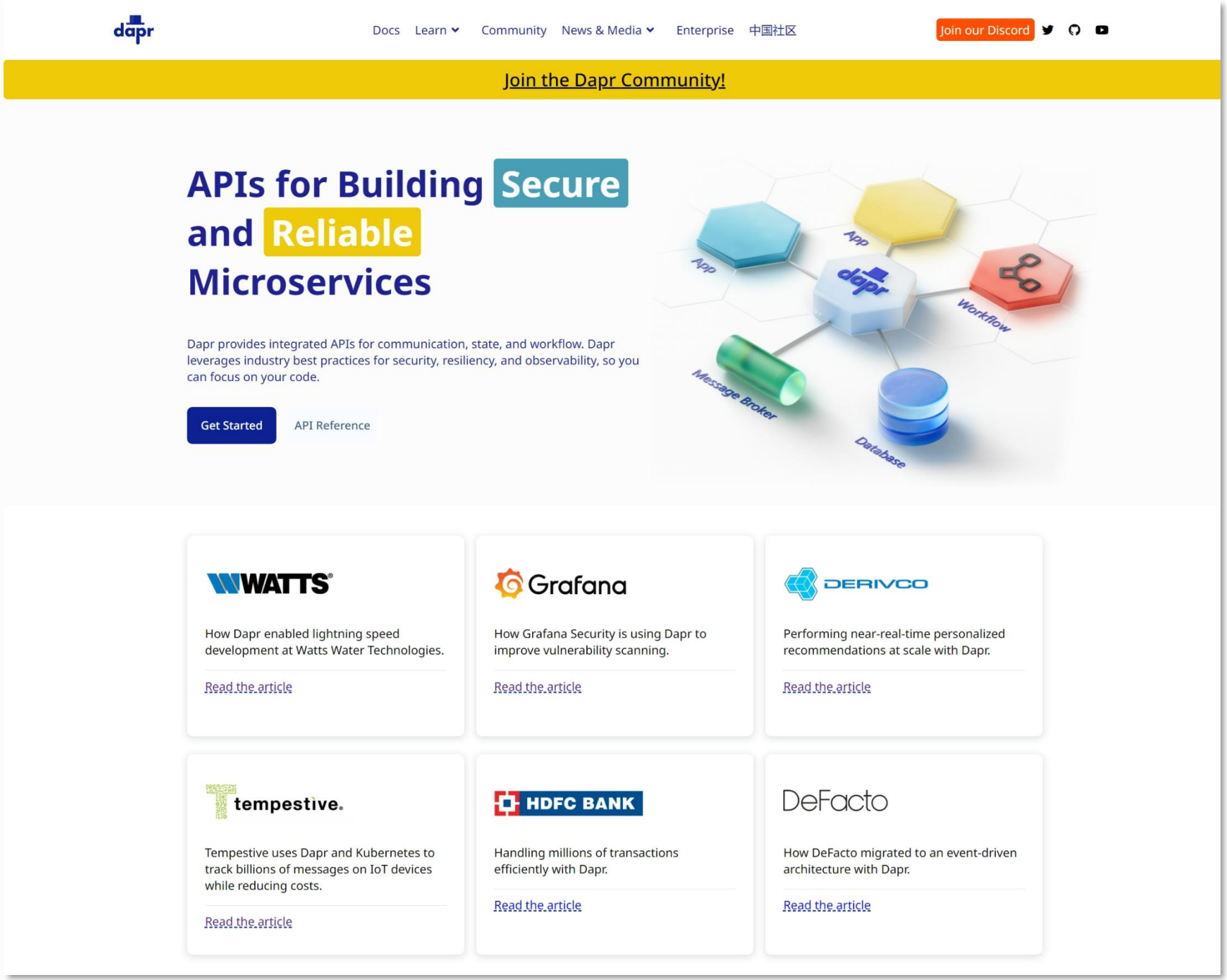
Application

Runtime

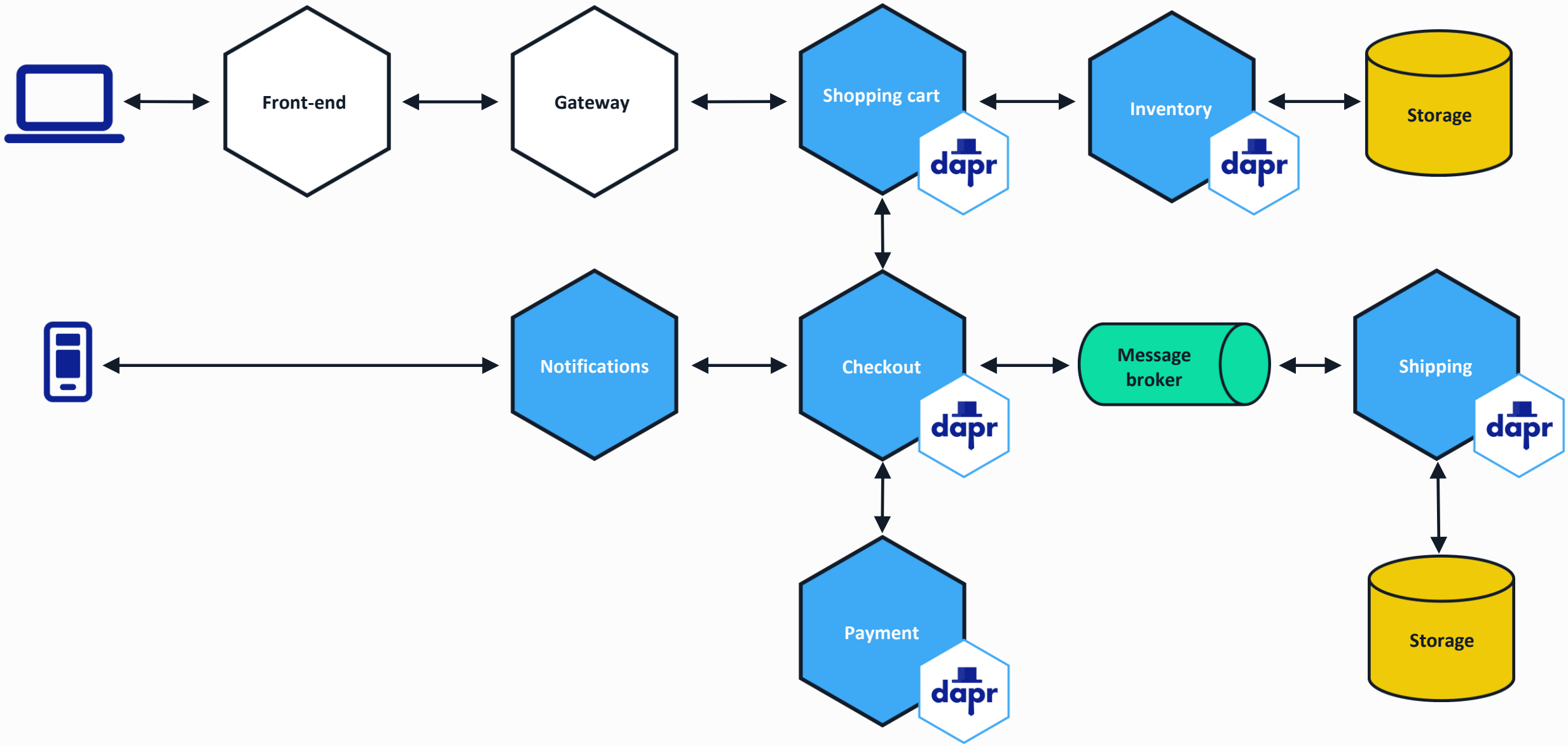
dapr.io

CLOUD NATIVE
COMPUTING FOUNDATION

Graduated project



Dapr uses a sidecar pattern



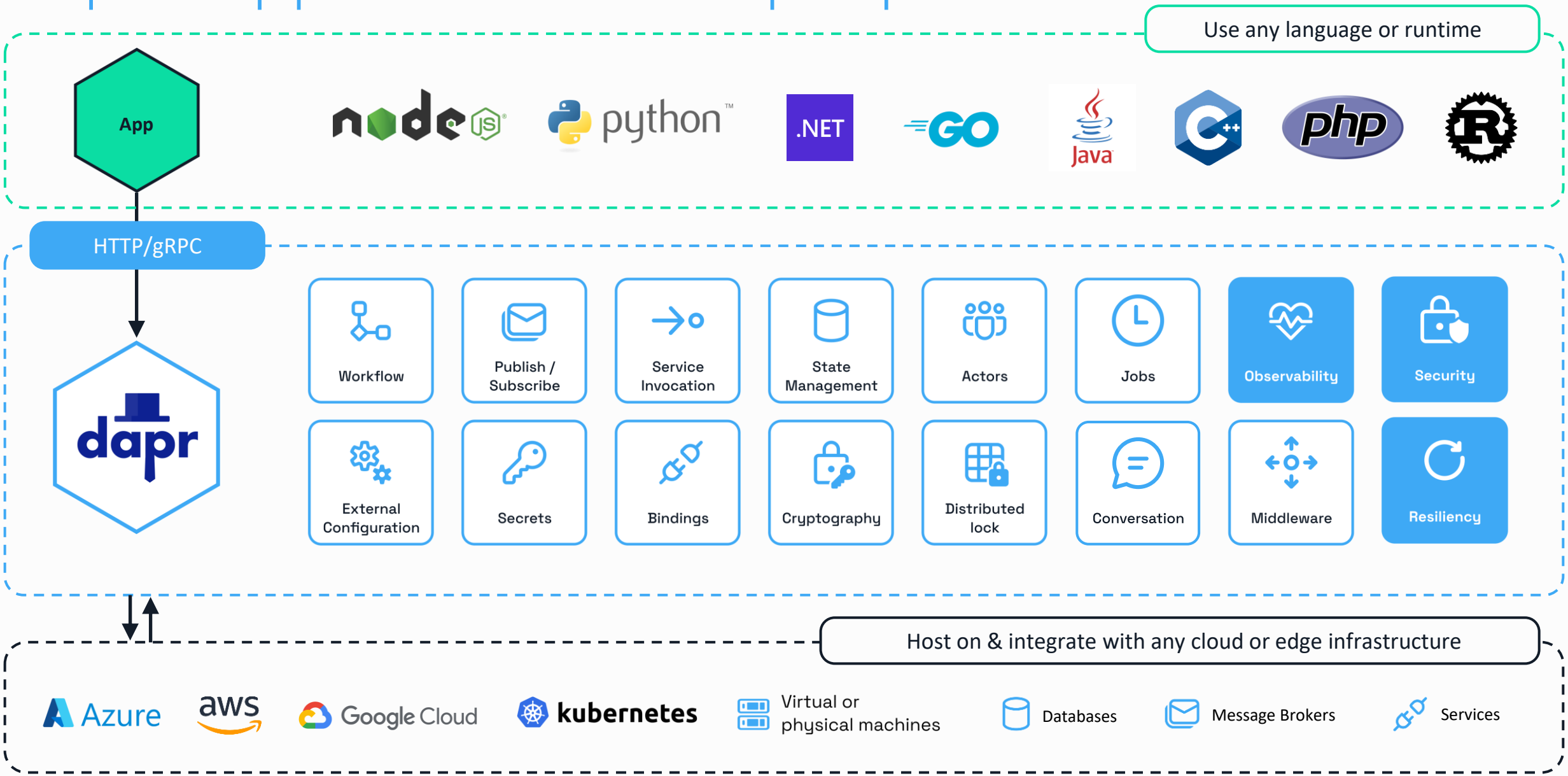
The Dapr sidecar provides built-in security, resiliency and observability capabilities.

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

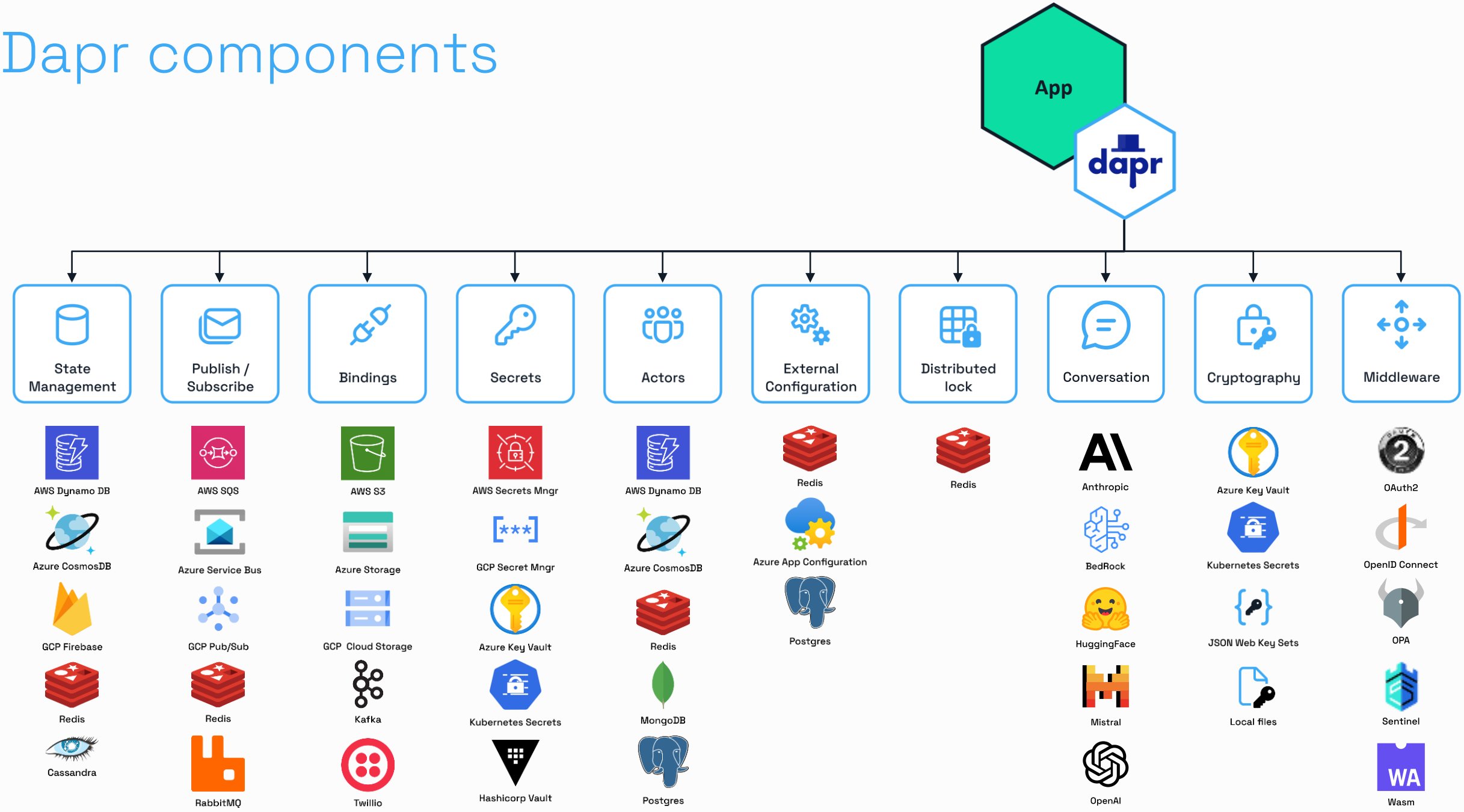
Dapr Goals

- Provide an integrated set of APIs
- Any language or framework
- Includes best practices & standards
- Platform agnostic
- Extensible and pluggable
- Community driven, vendor neutral

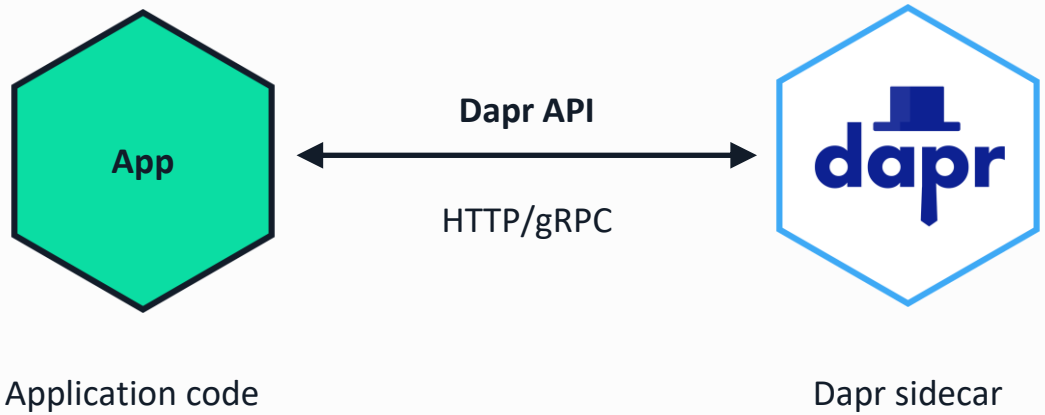
Dapr – Application developer platform



Dapr components



Sidecar pattern and the Dapr API



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

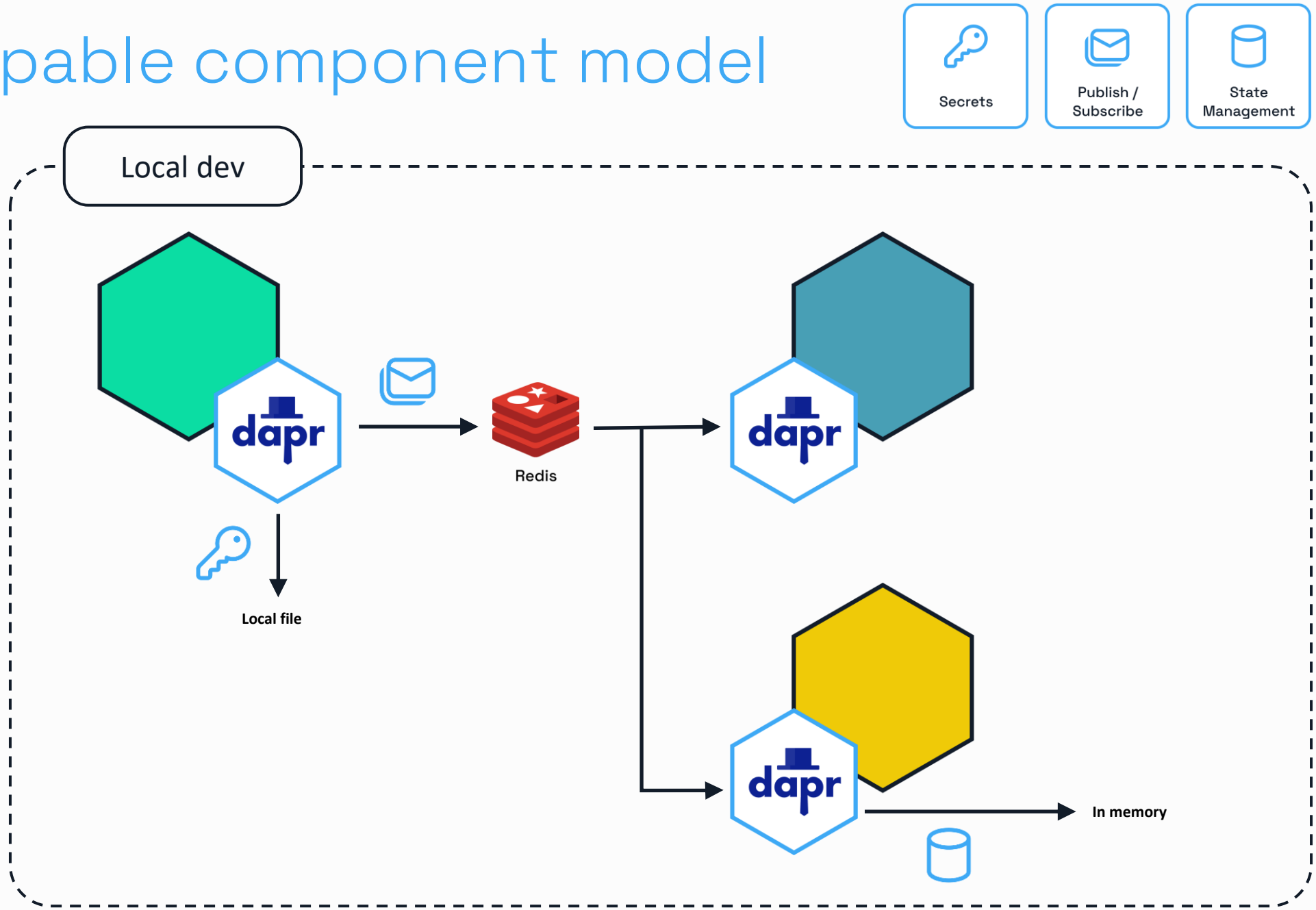
GET http://localhost:3500/v1.0/**state**/inventory/item50

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

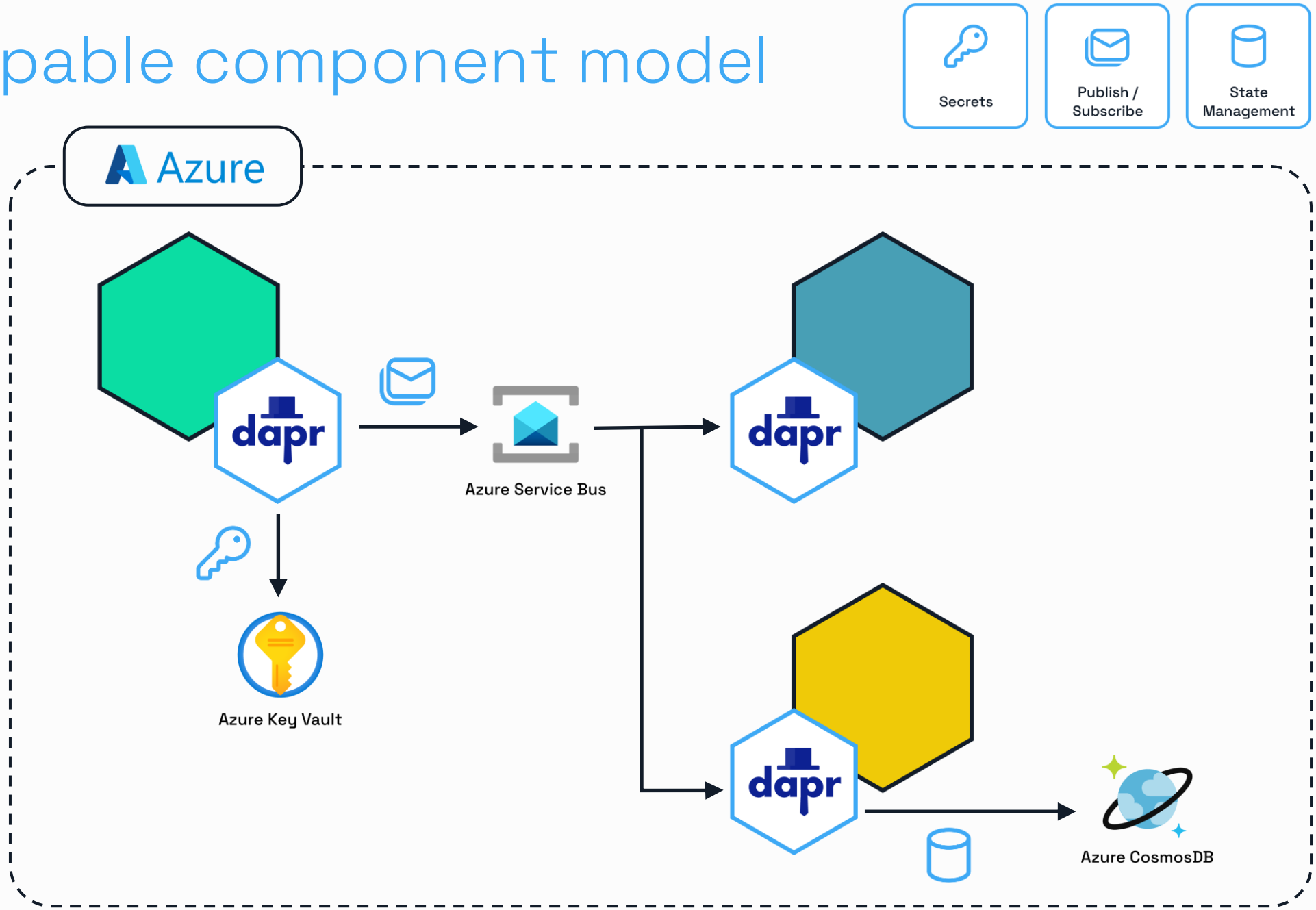
GET http://localhost:3500/v1.0/**secrets**/vault/dbaccess

POST http://localhost:3500/v1.0/**workflows**/dapr/businessprocess/start

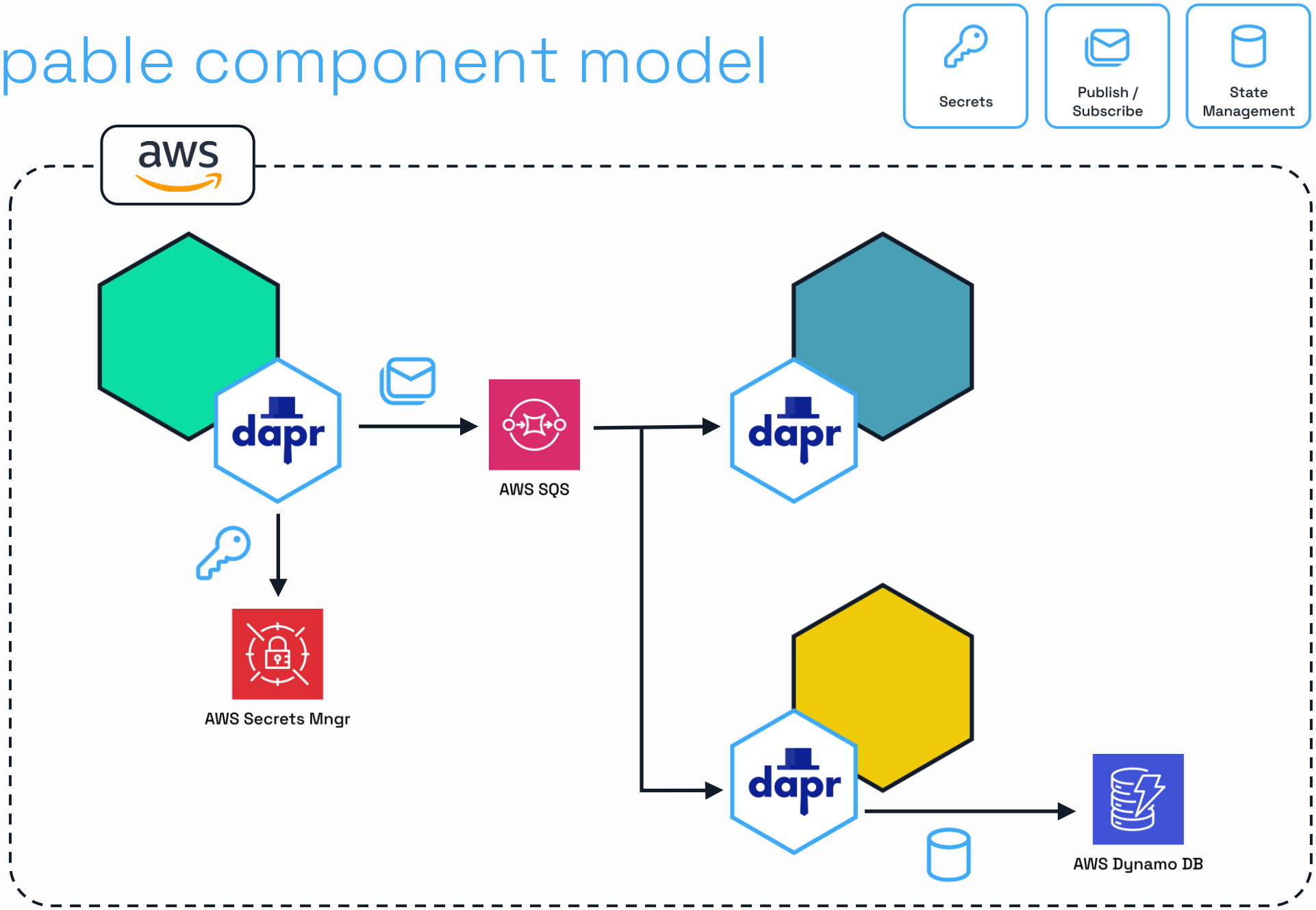
Swappable component model



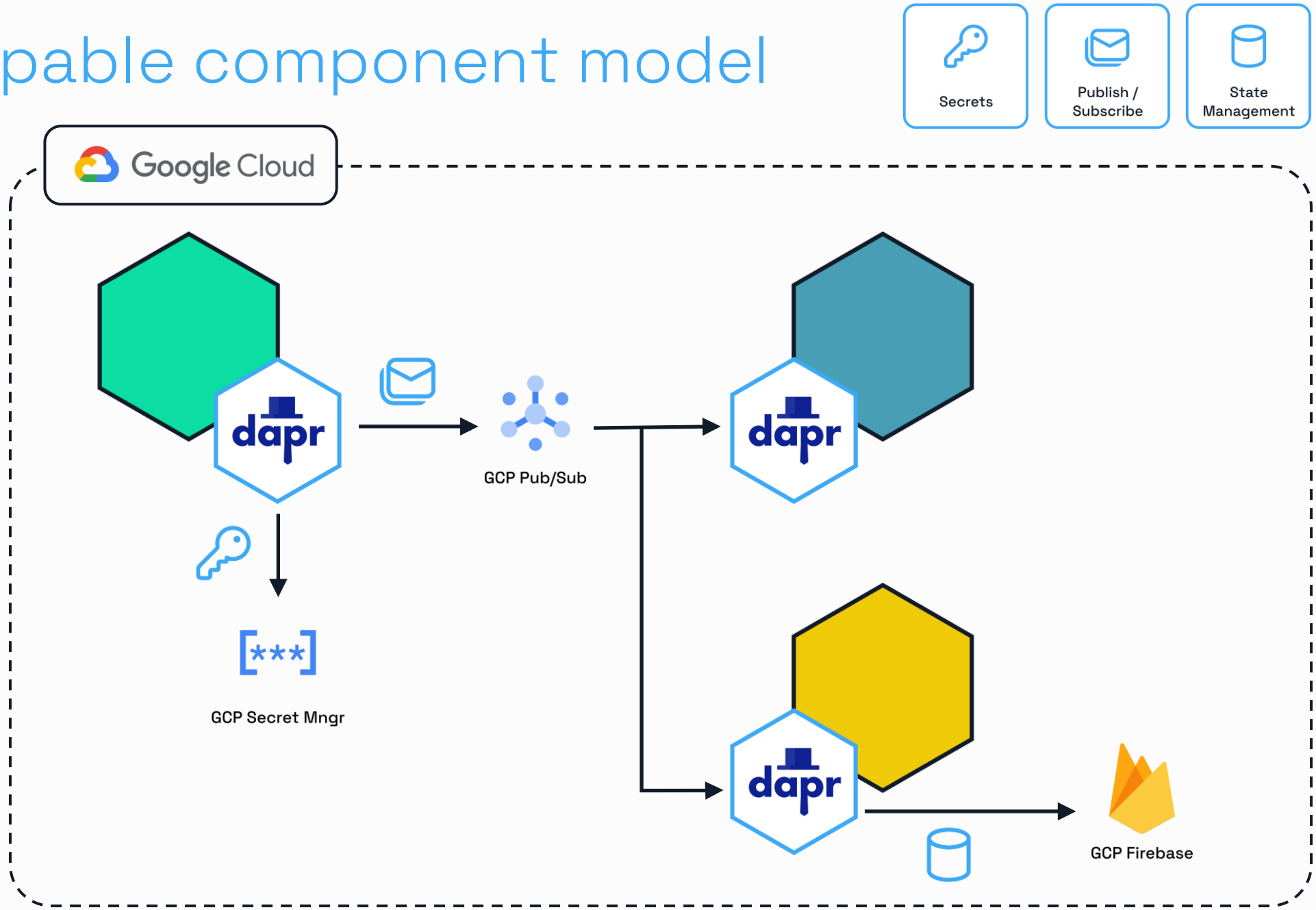
Swappable component model



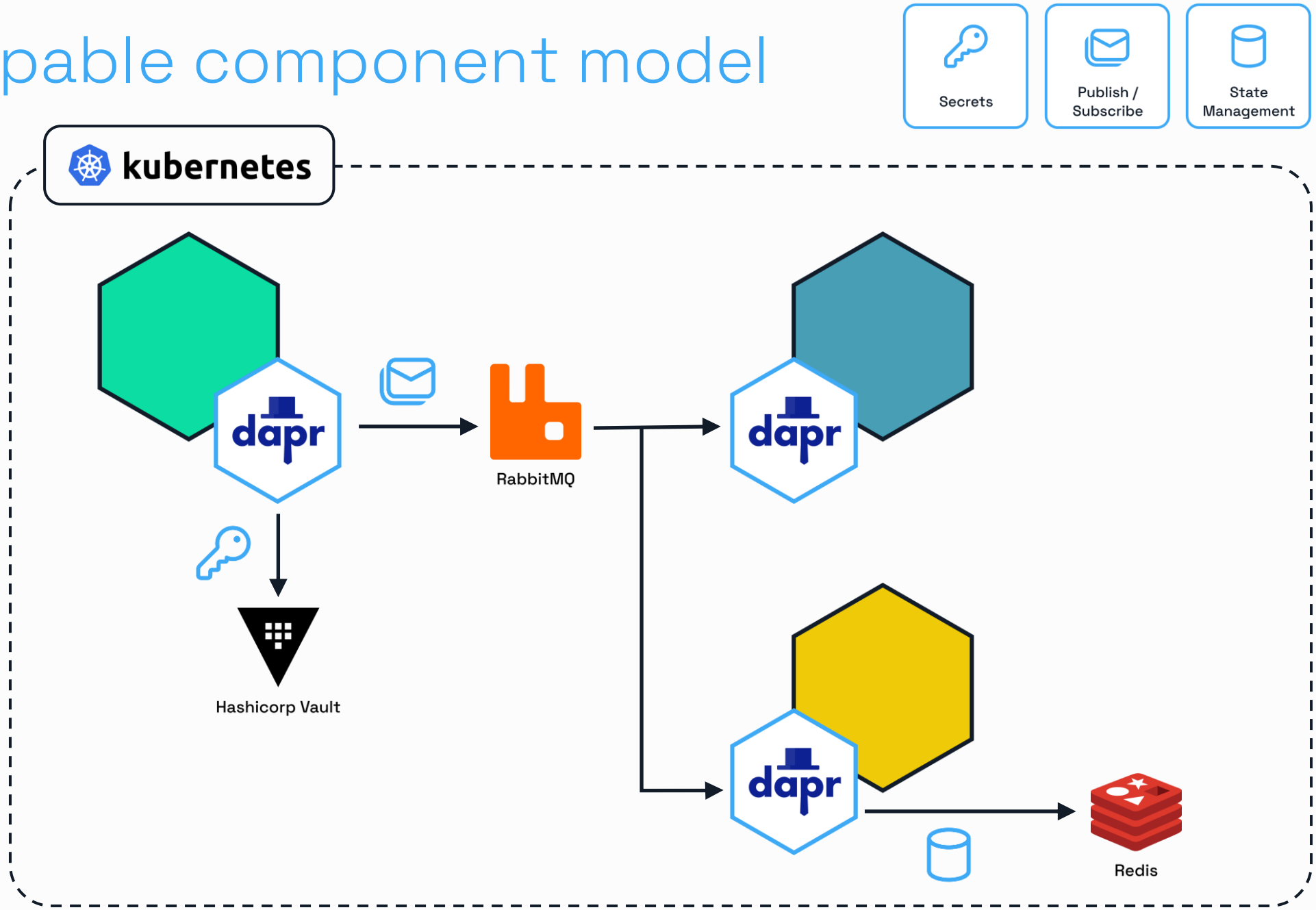
Swappable component model



Swappable component model



Swappable component model



Use Dapr anywhere

Azure Container Apps

Diagrid Catalyst

Microsoft Azure

Google Cloud

aws

Alibaba Cloud

kubernetes

Virtual or
physical machines

Dapr community

14/157
Largest CNCF project

CLOUD NATIVE COMPUTING FOUNDATION

120
Community components

700k+
Docker Hub pulls/month

300k
Unique doc views/month

31k
GitHub stars

3.9k
Contributors

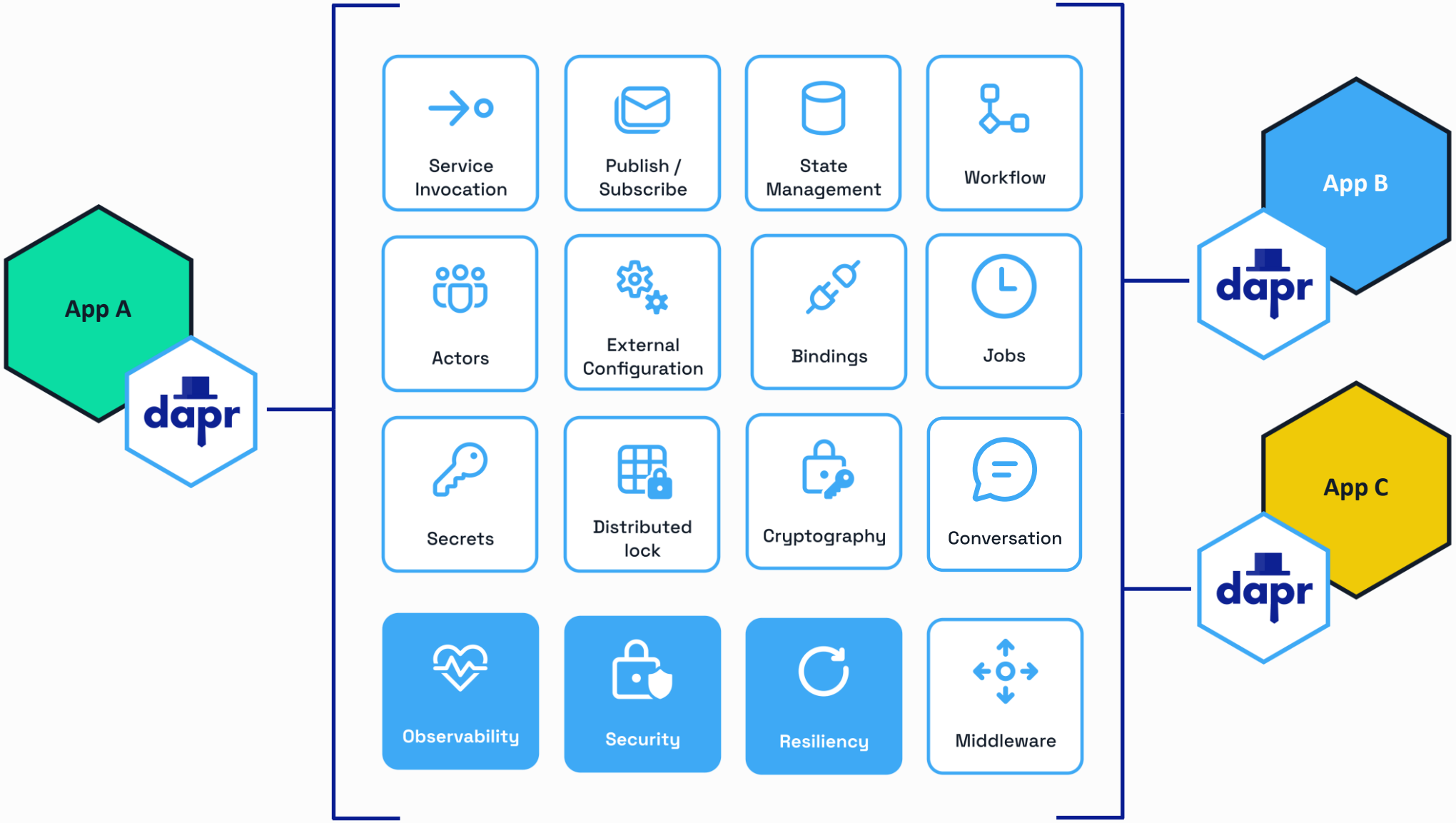
8k
Discord members

Dapr contributors

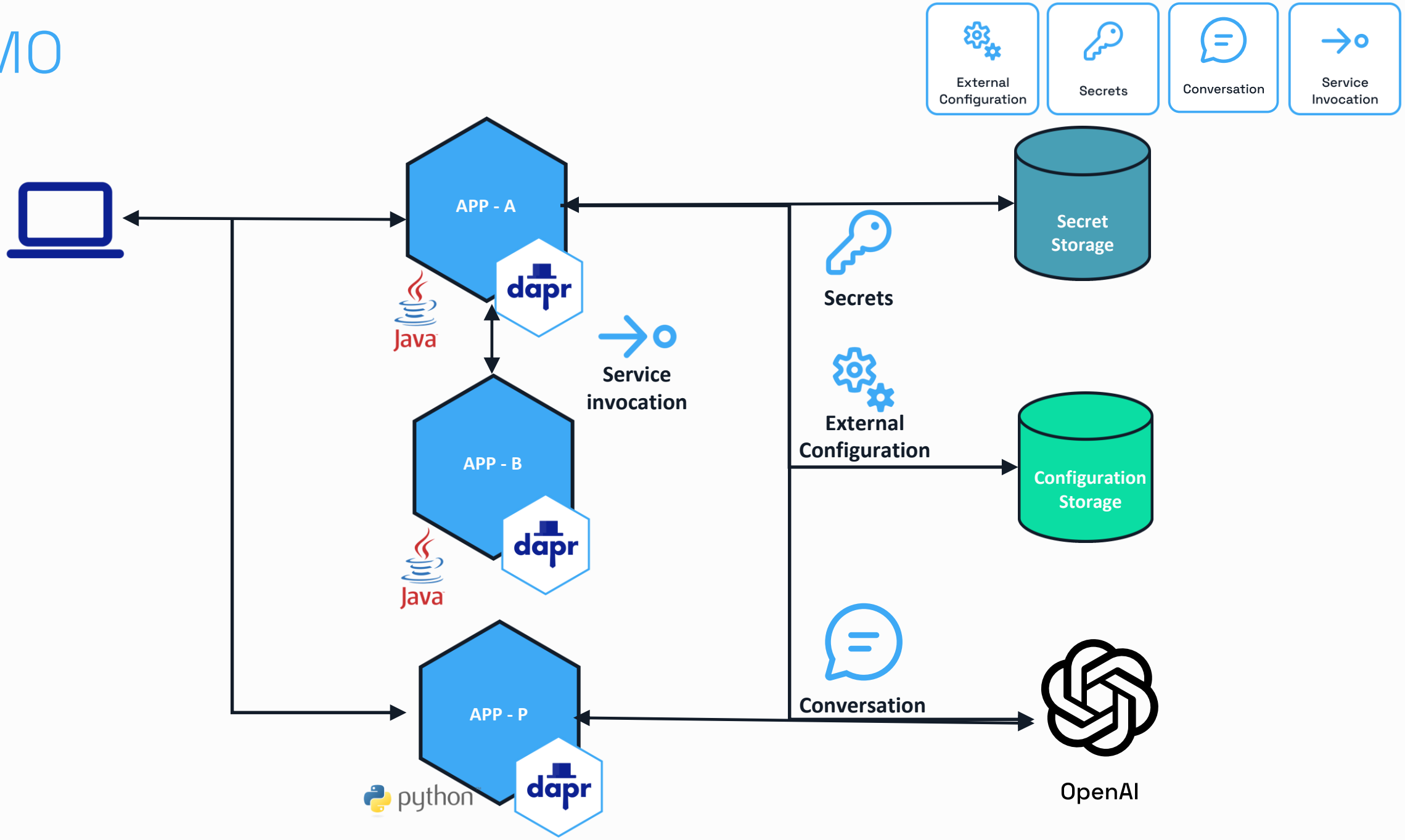
Dapr users


IBM
pwc
Microsoft
intel
Alibaba Cloud
ZEISS
NASA
HDFC BANK
CISCO
Rakuten
Lufthansa
FUJITSU
Grafana
zscaler
BOSCH
DeFacto
VONAGE
at bay
Dotmatics
wortell
IGNITION GROUP
Composabl
DERIVCO
WATTS
tempestive.

Dapr APIs & Cross cutting concerns



DEMO


External Configuration
Secrets
Conversation
Service Invocation
APP - A
dapr
Java
APP - B
dapr
Java
APP - P
dapr
python
Service invocation
Secrets
Secret Storage
External Configuration
Configuration Storage
Conversation
OpenAI

Service Invocation API

https://docs.dapr.io/developing-applications/building-blocks/service-invocation/

Service Invocation

The service invocation API allows synchronous communication between services.

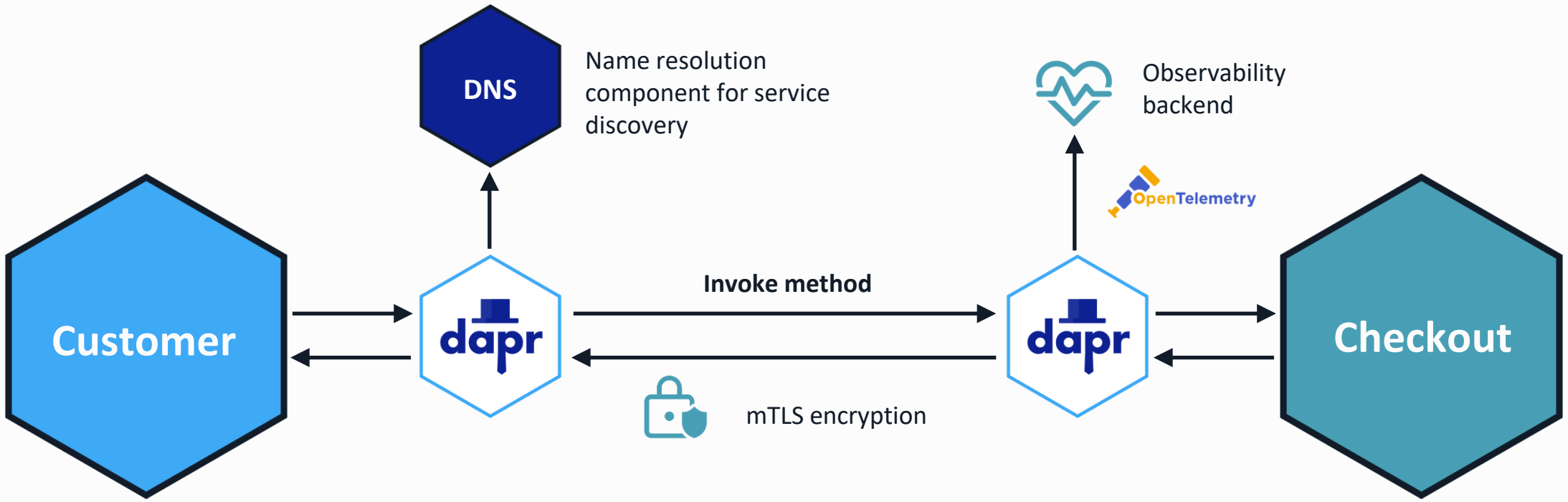
- Service discovery via name resolution components
- Invoke HTTP and gRPC services consistently
- Configurable resiliency policies
- Built-in distributed tracing & metrics
- Access control policies & mTLS
- Chain pluggable middleware components

Service Invocation

POST
http://localhost:3500/v1.0/**invoke**/checkout/method/order

POST
http://localhost:5100/order

Service Invocation



POST
http://localhost:3500/v1.0/**invoke**/checkout/method/order

POST
http://localhost:5100/order

Service Invocation in .NET


```
var order = new Order(orderId);

var client = DaprClient.CreateInvokeHttpClient(appId: "order-processor");

var response = await client.PostAsJsonAsync("/orders", order);
```

Service Invocation in Python


```
base_url = os.getenv('BASE_URL', 'http://localhost') + ':' +
           os.getenv('DAPR_HTTP_PORT', '3500')

headers = {'dapr-app-id': 'order-processor', 'content-type': 'application/json'}

order = {'orderId': orderId}

result = requests.post(
        url='%s/orders' % (base_url),
        data=json.dumps(order),
        headers=headers
```

Service Invocation Demo

https://docs.dapr.io/getting-started/quickstarts/serviceinvocation-quickstart/

Secrets Management API

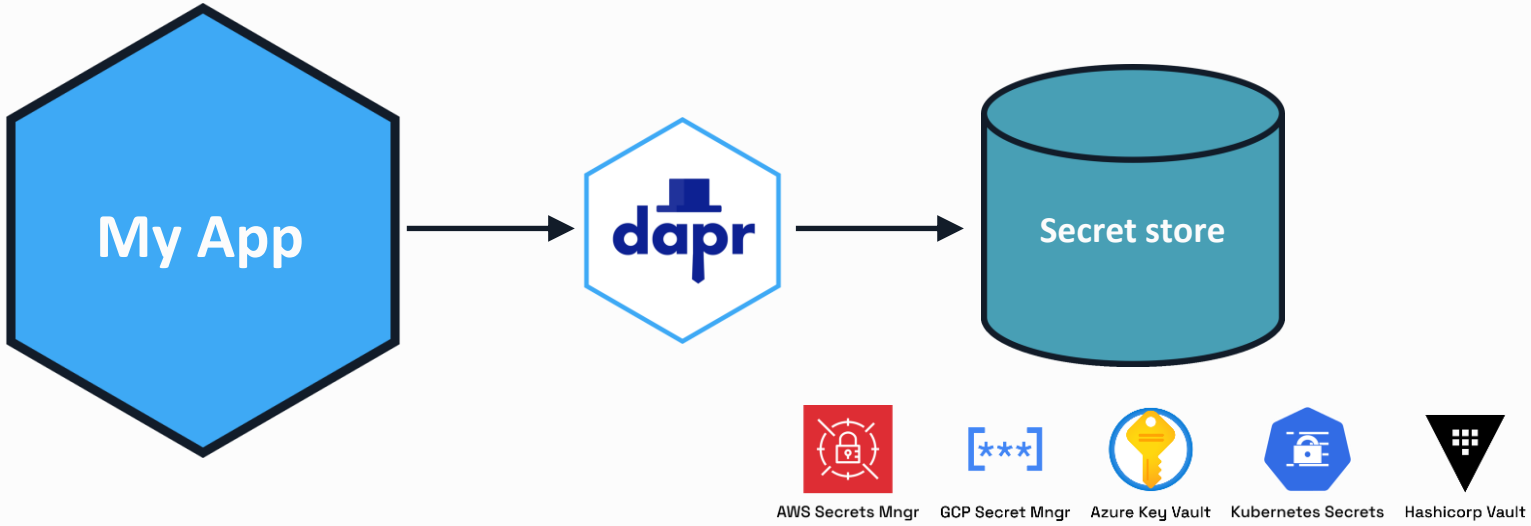
https://docs.dapr.io/developing-applications/building-blocks/secrets/secrets-overview/

Secrets Management

The secrets management API enables access to sensitive information in secret stores.

- Integrates with public cloud and cloud-native secret stores
- Safely access secrets in your applications
- Reference secrets in Dapr components
- Use scopes to limit access

Secrets Management



GET http://localhost:3500/v1.0/**secrets**/myvault/mysecret

RESPONSE

```
{
  "mysecret": "secretvalue"
}
```

Secrets Management with .NET SDK


```
const string DAPR_SECRET_STORE = "localsecretstore";
const string SECRET_NAME = "secret";

var client = new DaprClientBuilder().Build();

var secret = await client.GetSecretAsync(DAPR_SECRET_STORE, SECRET_NAME);
```

Secrets Management with Python SDK


```
DAPR_SECRET_STORE = 'localsecretstore'
SECRET_NAME = 'secret’

with DaprClient() as client:
        secret = client.get_secret(store_name=DAPR_SECRET_STORE, key=SECRET_NAME)
```

Secrets Management Demo

https://docs.dapr.io/getting-started/quickstarts/secrets-quickstart/

External Configuration API

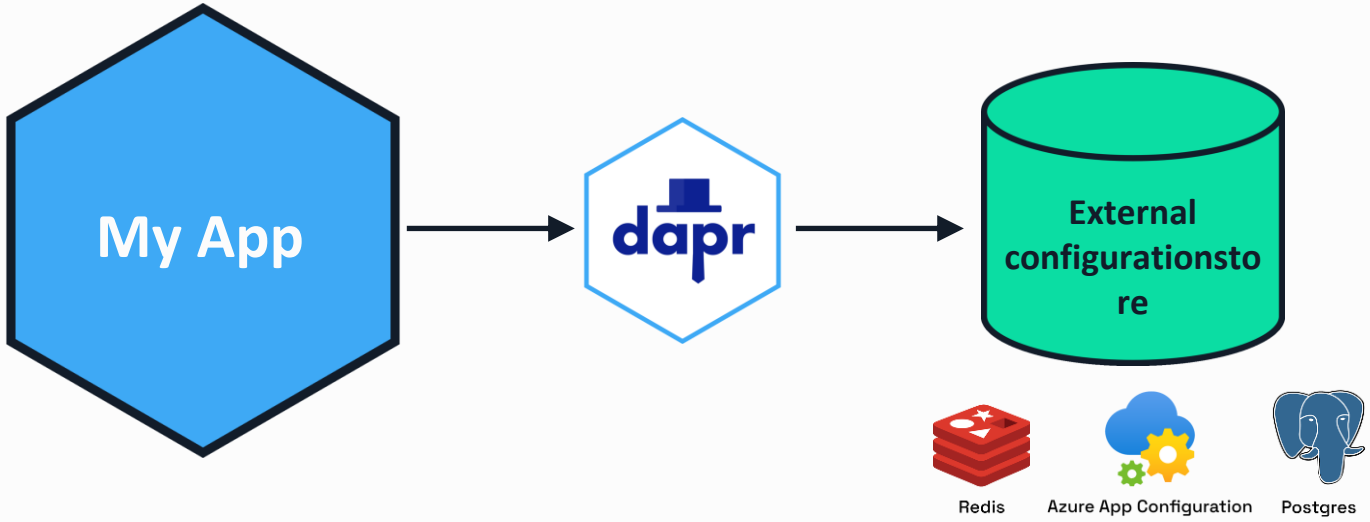
https://docs.dapr.io/developing-applications/building-blocks/configuration/

External configuration

The external configuration API enables read access to configuration data.

- Integrates with many configuration stores
- Subscribe to configuration changes
- Use scopes to limit access

External Configuration



GET http://localhost:3500/v1.0/**configuration**/myconfig?key=config1

RESPONSE

```
{
  "config1": {
    "value" : "configvalue"
  }
}
```

External Configuration with .NET SDK


```
const string DAPR_CONFIGURATION_STORE = "configstore";
var CONFIGURATION_ITEMS = new List<string> { "orderId1", "orderId2" };

var client = new DaprClientBuilder().Build();

var config = await client.GetConfiguration(
        DAPR_CONFIGURATION_STORE,
        CONFIGURATION_ITEMS);

foreach (var item in config.Items)
{
        var configItem = System.Text.Json.JsonSerializer.Serialize(item.Value);
}
```

External Configuration with Python SDK


```python
DAPR_CONFIGURATION_STORE = 'configstore'
CONFIGURATION_KEYS = ['orderId1', 'orderId2']

with DaprClient() as client:
    for key in CONFIGURATION_KEYS:
        resp = client.get_configuration(store_name=DAPR_CONFIGURATION_STORE,
               keys=[key], config_metadata={})
        print(f"Configuration for {key} : {resp.items[key].value}", flush=True)
```

External Configuration Demo

https://docs.dapr.io/getting-started/quickstarts/configuration-quickstart/

Conversation API

https://docs.dapr.io/developing-applications/building-blocks/conversation/

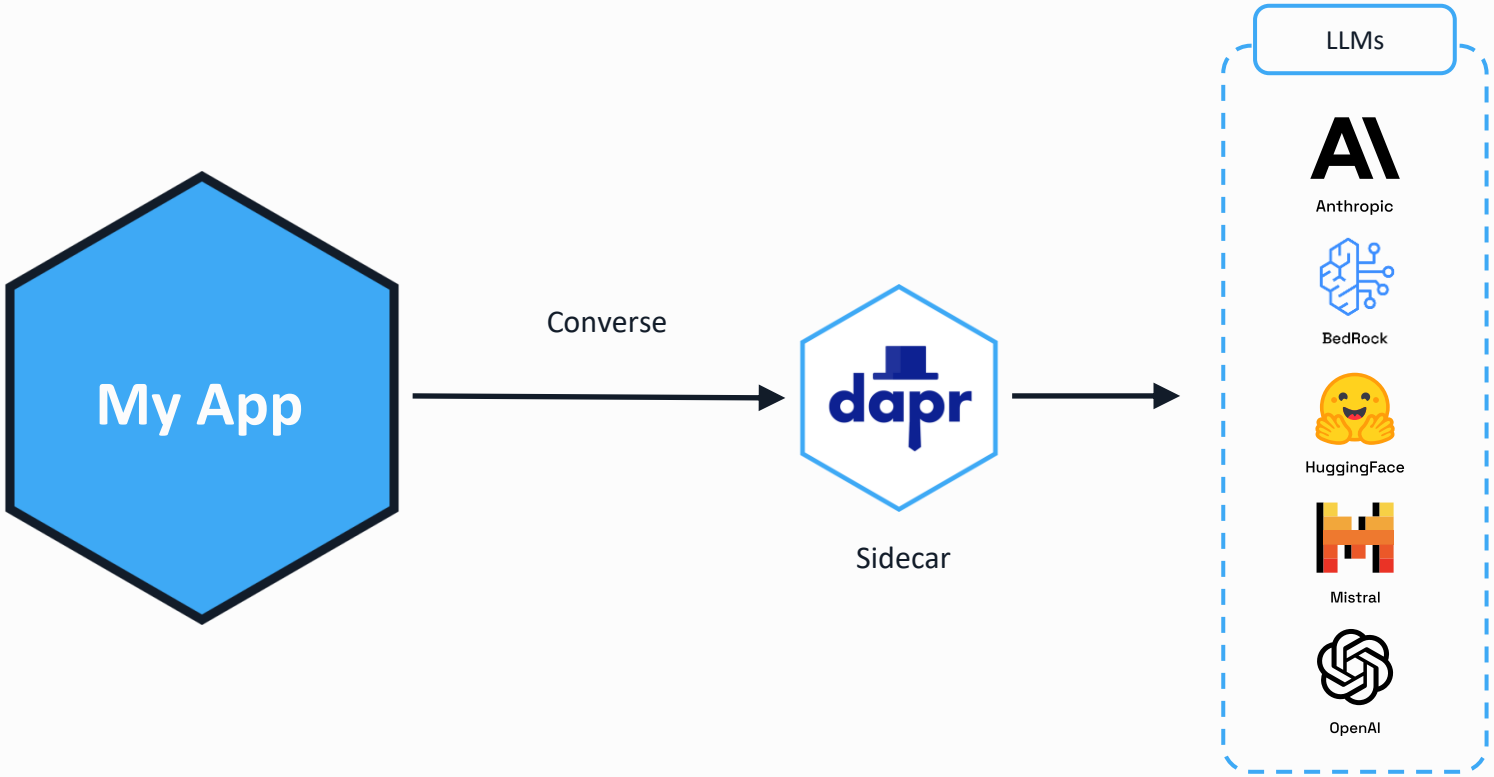
LLM Conversation

ALPHA

The Conversation API reduces the complexity of securely and reliably interacting with Large Language Models (LLM) at scale.

- One consistent API to talk to underlying LLM providers
- Use prompt caching to reduce latency and cost
- Use PII obfuscation to remove sensitive user input

LLM Conversation



POST http://localhost:3500/v1.0-alpha1/**conversation**/llm-name/converse

LLM Conversation with .NET SDK


```
const string prompt = "What is dapr?";

var conversationClient = app.Services.GetRequiredService<DaprConversationClient>();

var response = await conversationClient.ConverseAsync(ConversationComponentName, [new(prompt,
DaprConversationRole.Generic)]);

if (response != null)
{
        Console.Write("Output response:");
        foreach (var resp in response.Outputs)
        {
                Console.WriteLine($" {resp.Result}");
        }
}
```

LLM Conversation with Python SDK


```python
PROMPT = 'What is dapr?'

with DaprClient() as d:
    inputs = [
        ConversationInput(content=PROMPT, role='user', scrub_pii=True),
    ]

    metadata = {
        'model': 'modelname',
        'key': 'authKey',
        'cacheTTL': '10m',
    }


    response = d.converse_alpha1(
        name=CONVERSATION_COMPONENT, inputs=inputs, temperature=0.7,
       context_id='chat-123', metadata=metadata
    )

    for output in response.outputs:
        print(f'Output response: {output.result}')
```

Conversation Demo

https://docs.dapr.io/getting-started/quickstarts/conversation-quickstart/

Hosting modes

Hosting modes

Self-hosted

Run **`dapr init`** to install Docker images.

Run any app with a Dapr side car using dapr run.

Virtual/Physical machines

Self-deploy Dapr control plane and Hashicorp Consul per machine.

Use the Dapr Installer Bundle for airgapped environments.

Run any app with a Dapr side car using **`dapr run`**.

Kubernetes

Run **`dapr init -k`** to install Dapr (or use Helm). Integrated Dapr control plane.

Deploys placement, operator, sentry and injector pods.

Automatically injects a Dapr sidecar into all annotated pods.

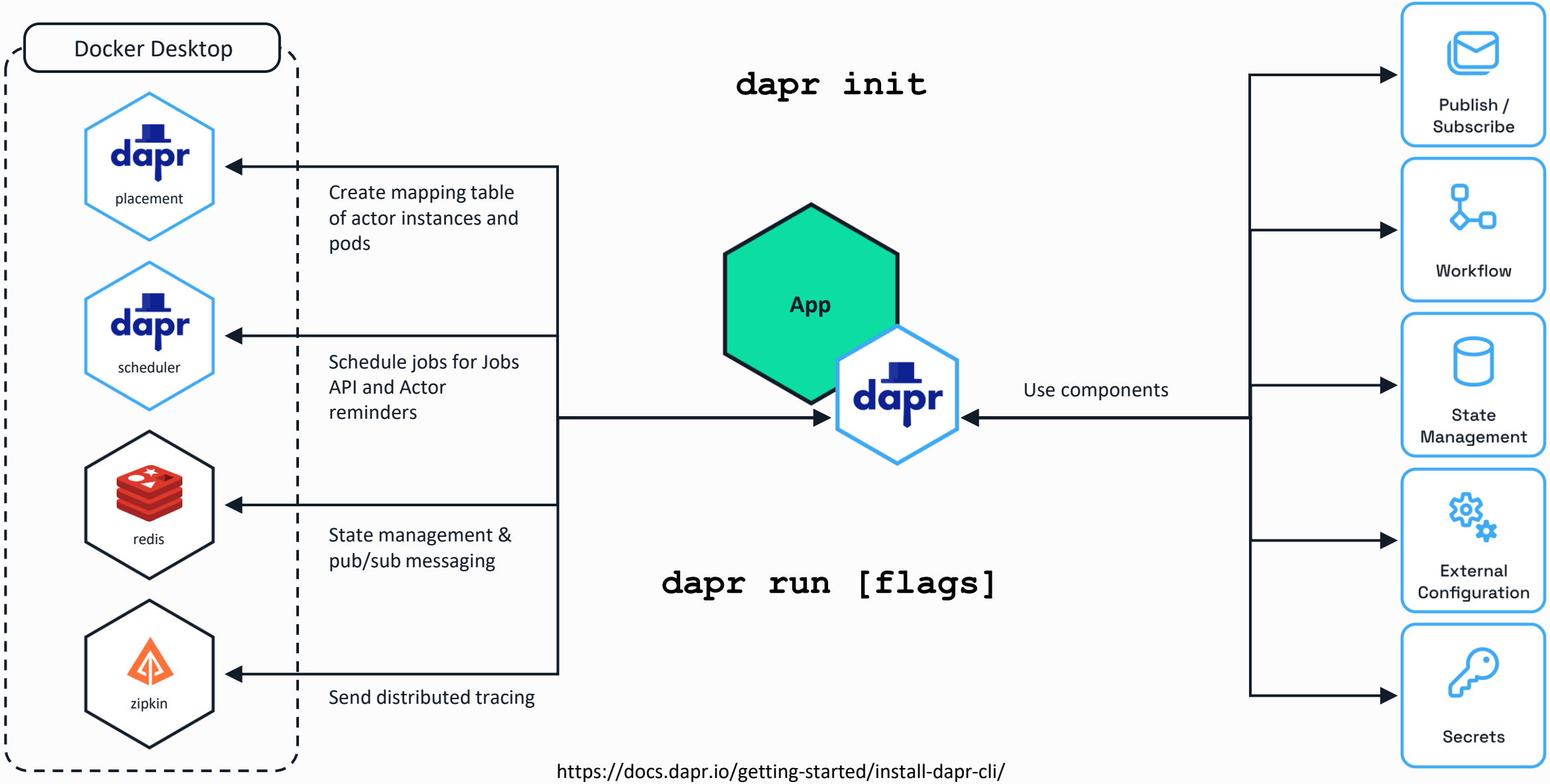
Hosting modes

Serverless

The Dapr side car is hosted by a provider.

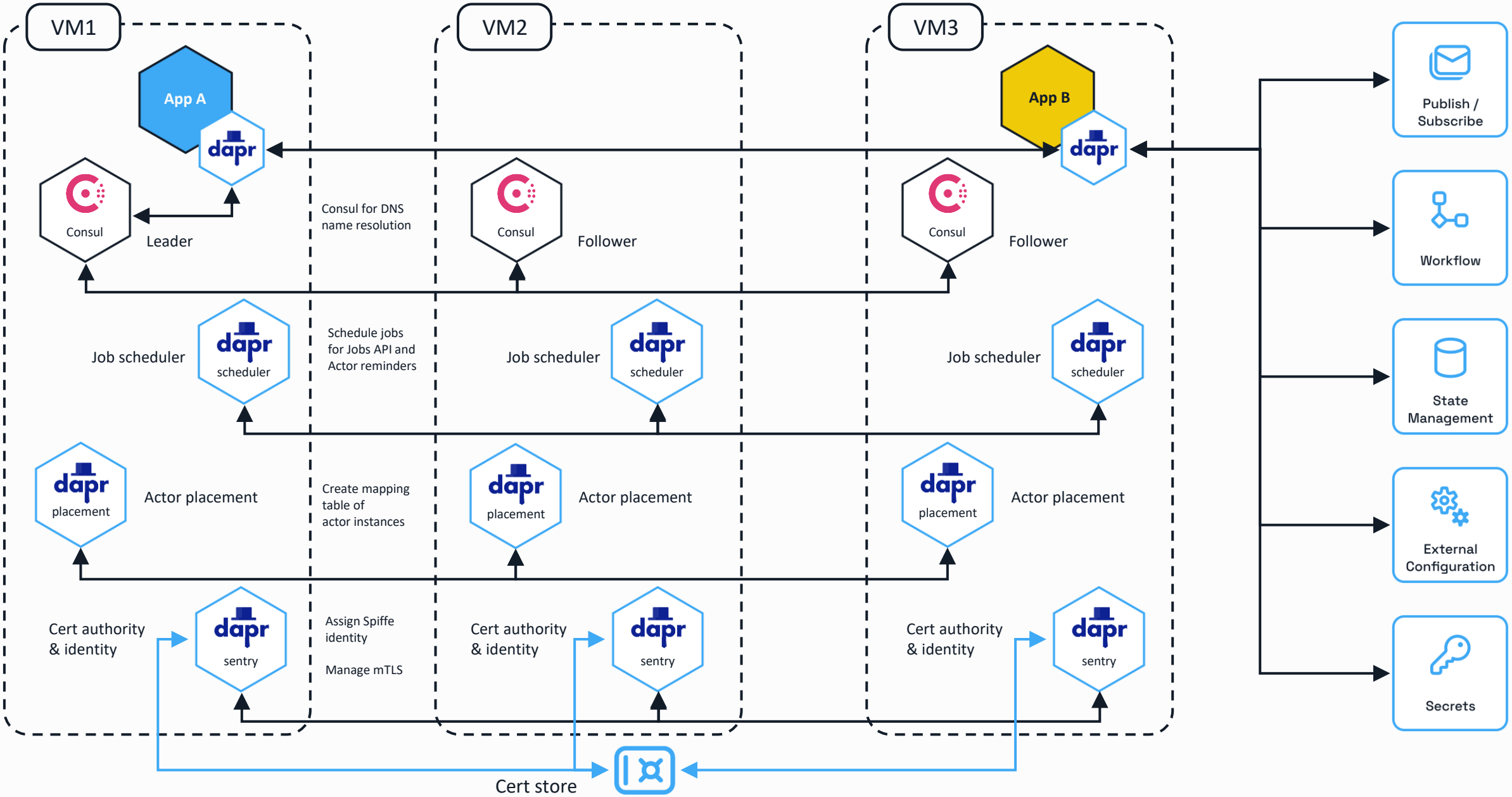
You only manage your applications.

Local development with the Dapr CLI

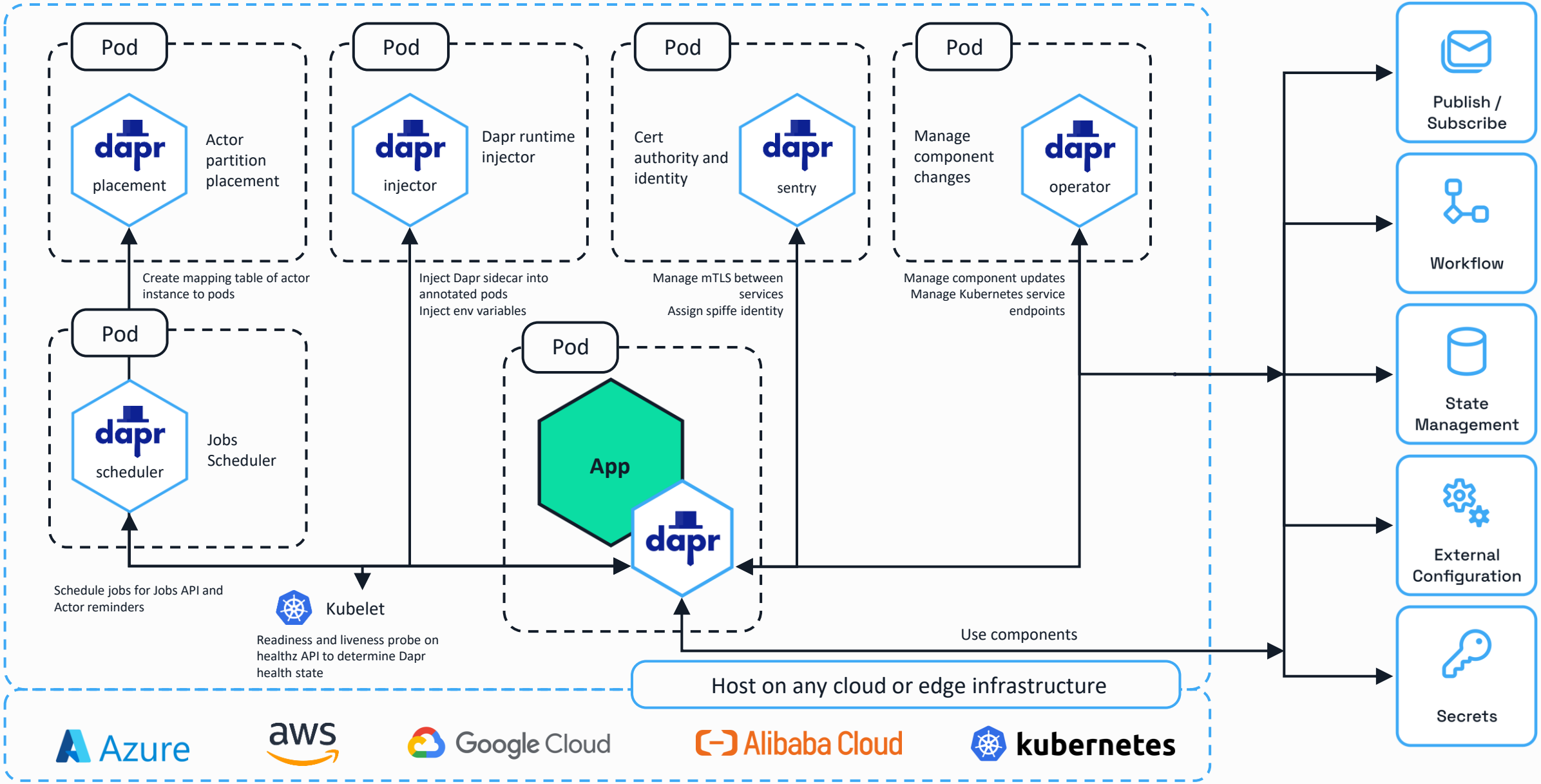


https://docs.dapr.io/getting-started/install-dapr-cli/

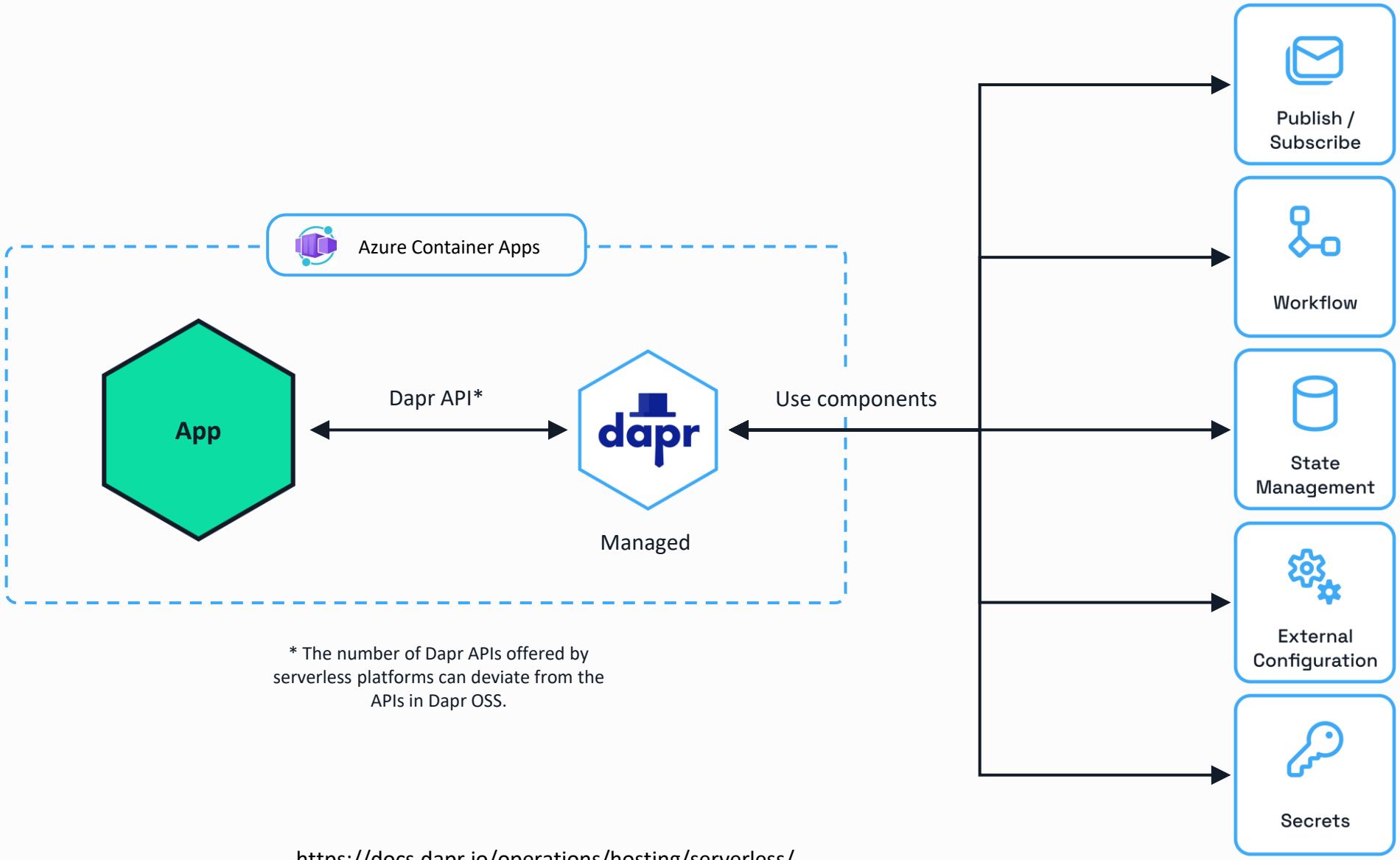
Dapr in self-hosted mode on VMs



Dapr on Kubernetes



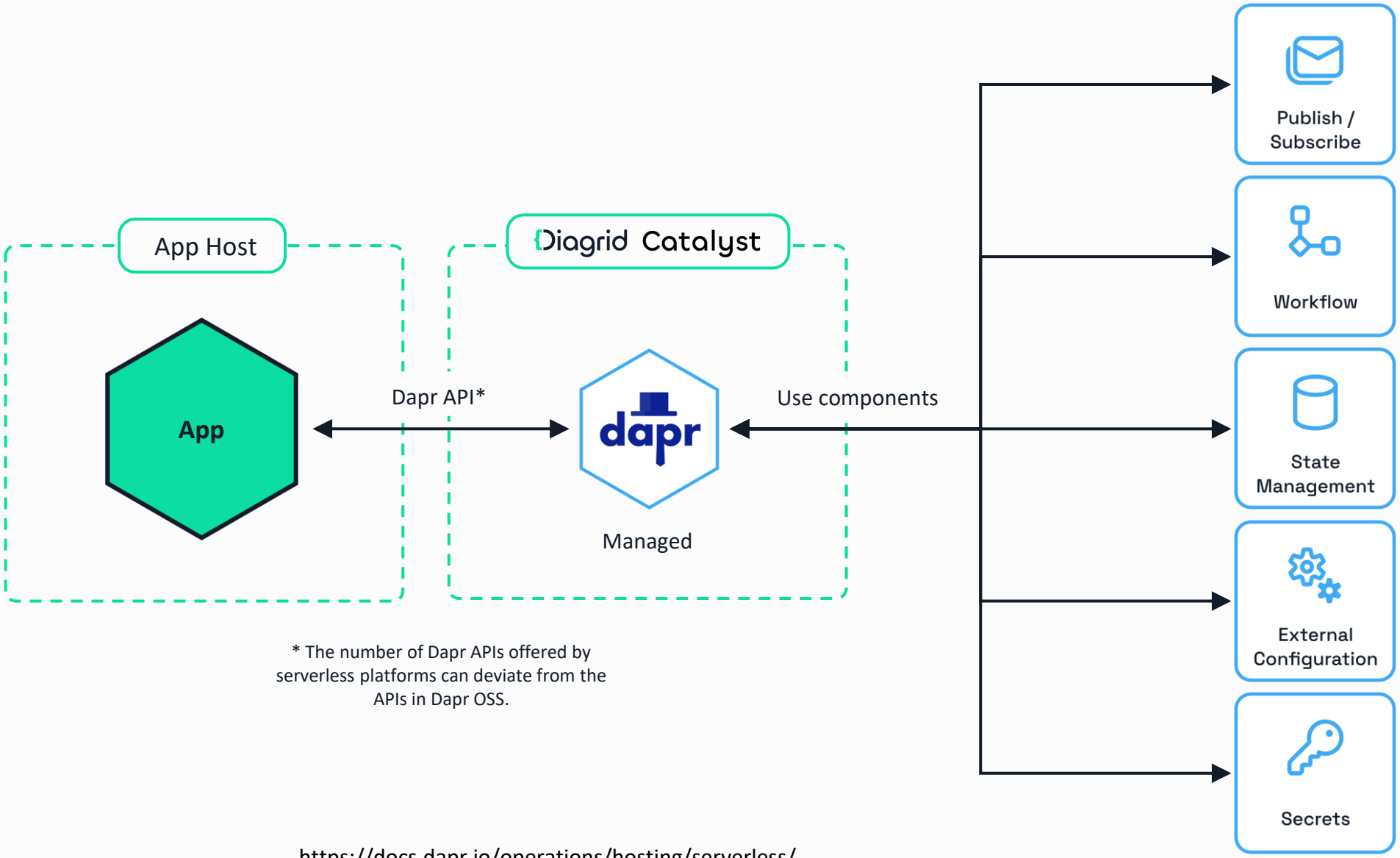
Serverless



* The number of Dapr APIs offered by serverless platforms can deviate from the APIs in Dapr OSS.

https://docs.dapr.io/operations/hosting/serverless/

Serverless



* The number of Dapr APIs offered by serverless platforms can deviate from the APIs in Dapr OSS.

https://docs.dapr.io/operations/hosting/serverless/

Dapr Resources

dapr.io

bit.ly/dapr-youtube

bit.ly/dapr-quickstarts

bit.ly/dapr-discord

@daprdev

@daprdev.bsky.social

dapr.io

Claim the Dapr Community Supporter badge!

bit.ly/dapr-supporter