



Façonner les architectures du futur avec Dapr

— **Christophe Gigax**
Cloud Solution Architect @ Hager Group
MVP Microsoft



@ChristopheGigax



THANK YOU !

GAMING¹

 ONIRYX i'groove

PROUDLY PART OF BLACKSHEEP TRIBES

 GENESIS CONSULT
IT Consultancy Services

 BuSI

 nsi
A CEGEKA COMPANY



 delaware

[sf=ir]

 devday
GET INSPIRED

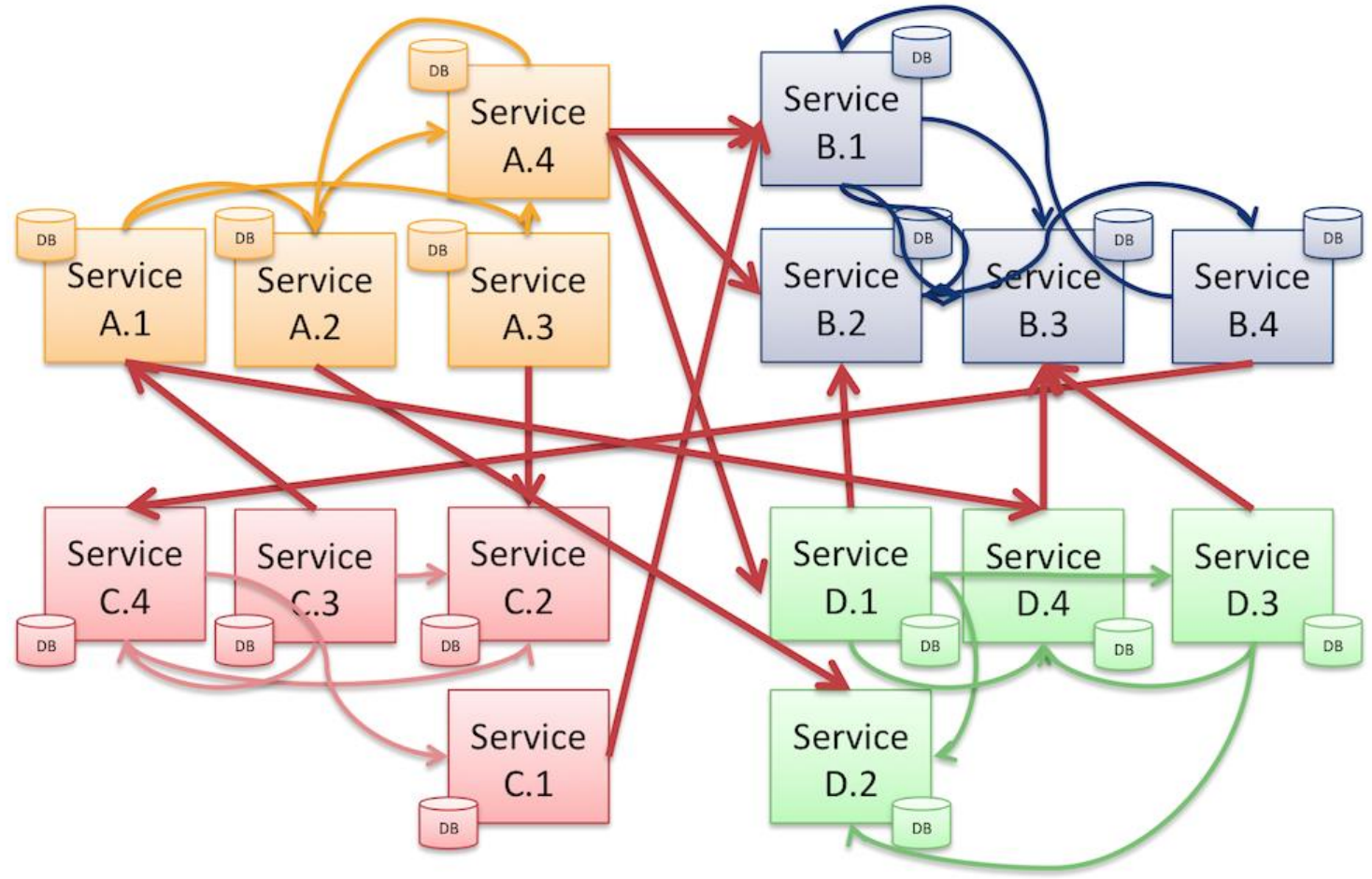
Les architectures microservices

Pourquoi les microservices ?

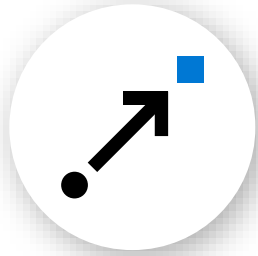
- Focus sur un aspect métier de l'application
- Autonomie
- Flexibilité
- Scalabilité
- Développement indépendant
- Déploiement indépendant
- Plus petit, mais plus performant

Les difficultés

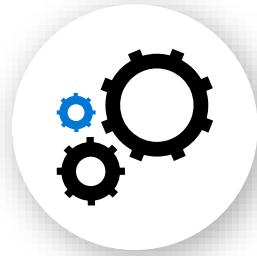
- Le debug
- Testabilité
- Complexité accrue de l'architecture
- Interdépendance des services
- Maîtrise des flux dans l'application



Les architectures microservices



Difficulté de migration d'un bloc monolithique vers une architecture microservice



Runtime non adapté et support des langages limités non-orienté pour le microservice



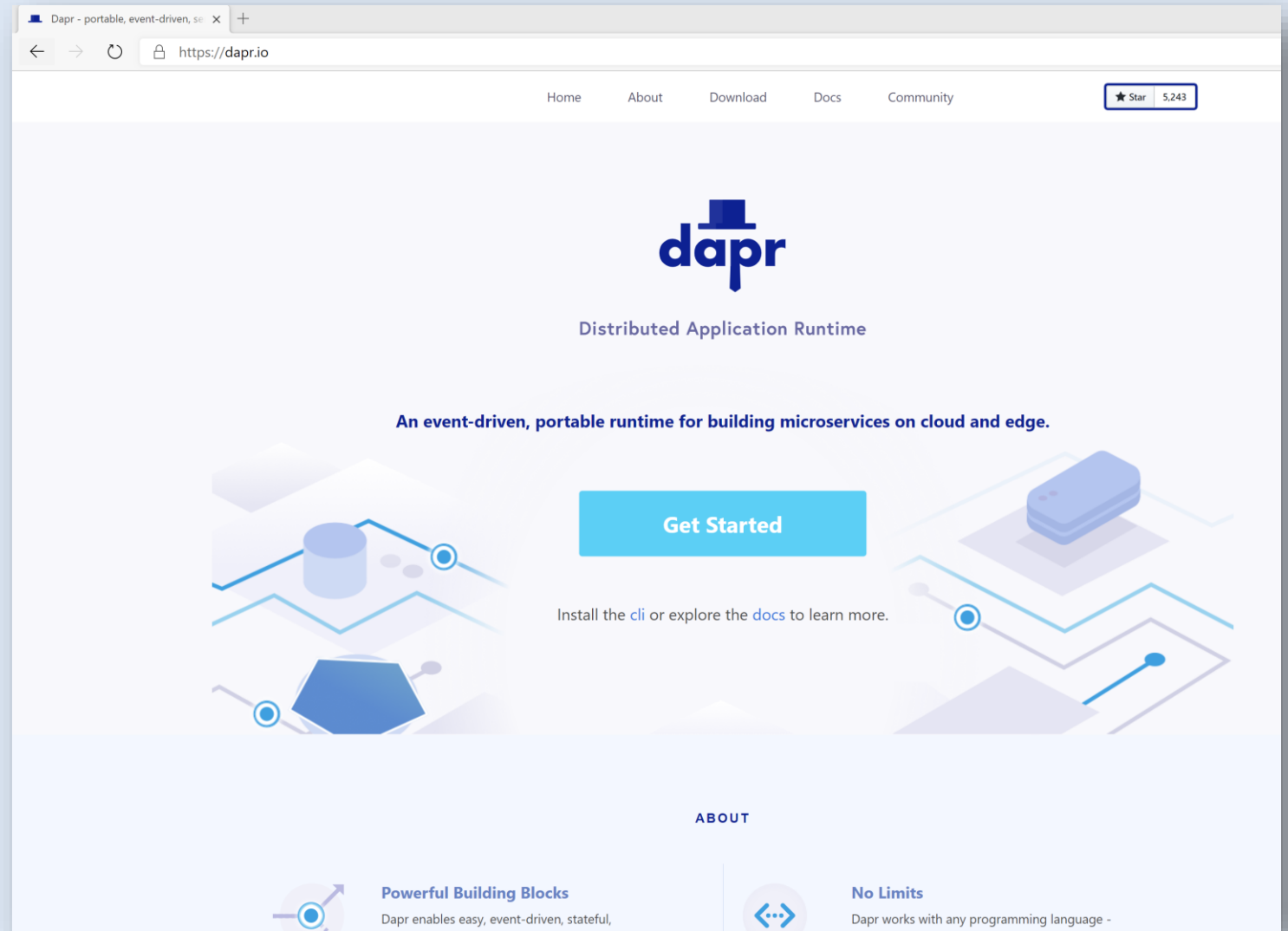
Application spécifique à une infrastructure / plateforme avec portage limité vers le Cloud ou l'Edge



Distributed Application Runtime

Portable, event-driven, *runtime* pour la création d'applications distribués au sein du Cloud et dans le Edge Computing

<https://dapr.io>



L'objectif de Dapr



Bonnes pratiques
Blocs préconstruits



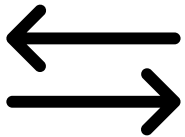
Any Language
ou Framework



Community Driven
Non commercial



Adoption des standards



Consistent, Portable,
APIs ouverts



Agnostique à la plateforme
Cloud + Edge



Extensible avec
composants pluggable

Les bases de Dapr

- ✓ API standard accessible via HTTP ou gRPC depuis le code du microservice
<http://localhost:3500/v1.0/state/inventory/orderkey>
<http://localhost:3500/v1.0/invoke/myapp/method/neworder>
- ✓ Fonctionnant en local sous forme d'un composant "sidecar" chargé au runtime pour chaque service

Application code

Microservices written in

Any code or framework...



HTTP API

gRPC API





Service-
to- service
invocation


State
management


Publish
and
subscribe


Resource
bindings
and triggers

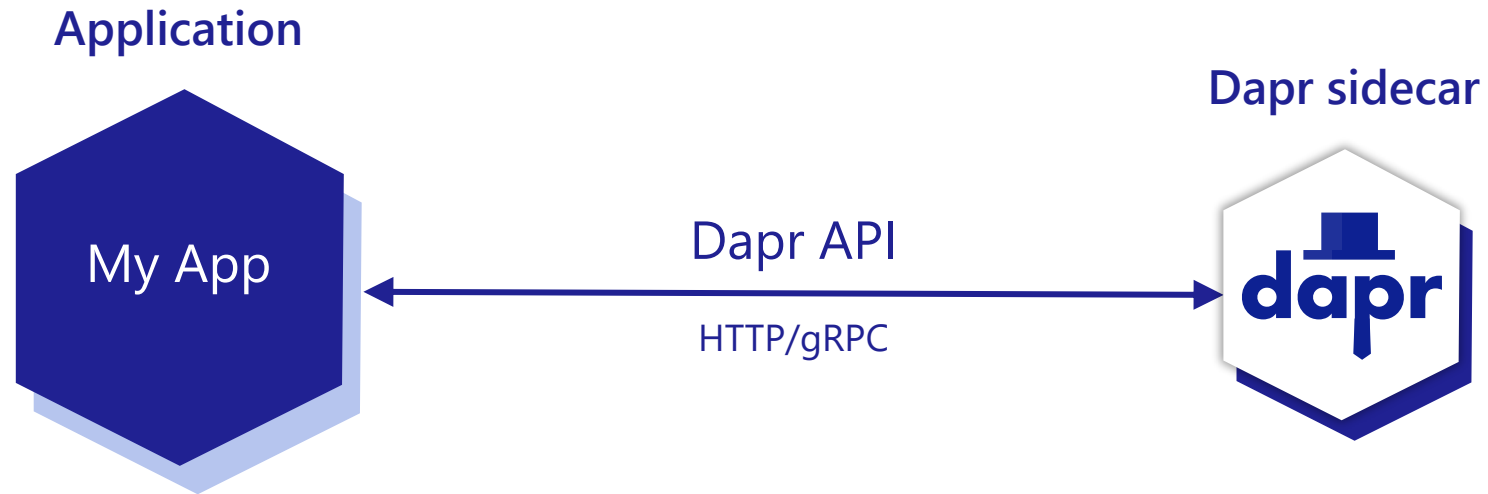

Actors


Distributed
tracing


Secrets


Extensible

Sidecar model



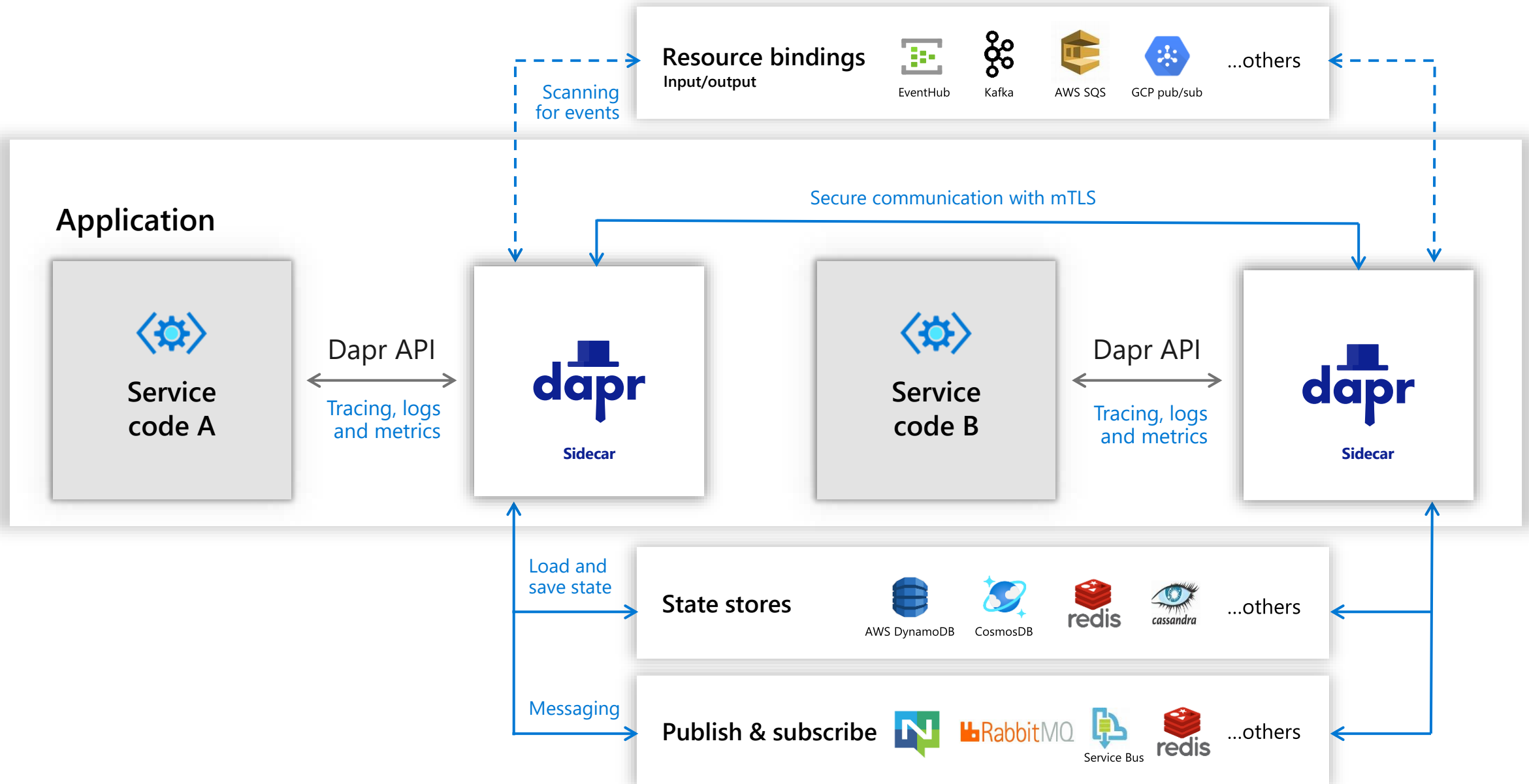
POST http://localhost:3500/v1.0/**invoke**/cart/method/neworder

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

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

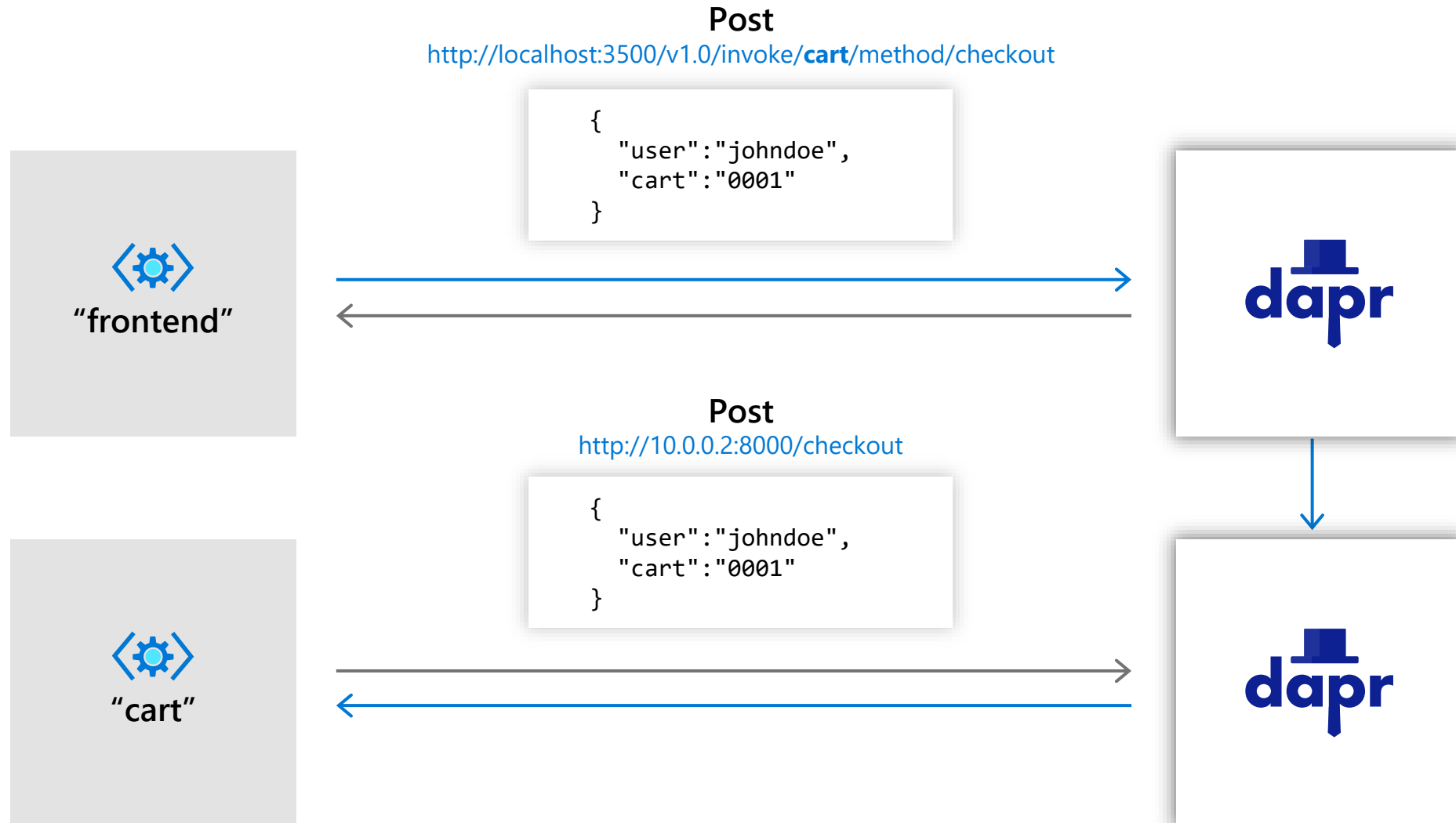
GET http://localhost:3500/v1.0/**secrets**/keyvault/password

Exemple en local

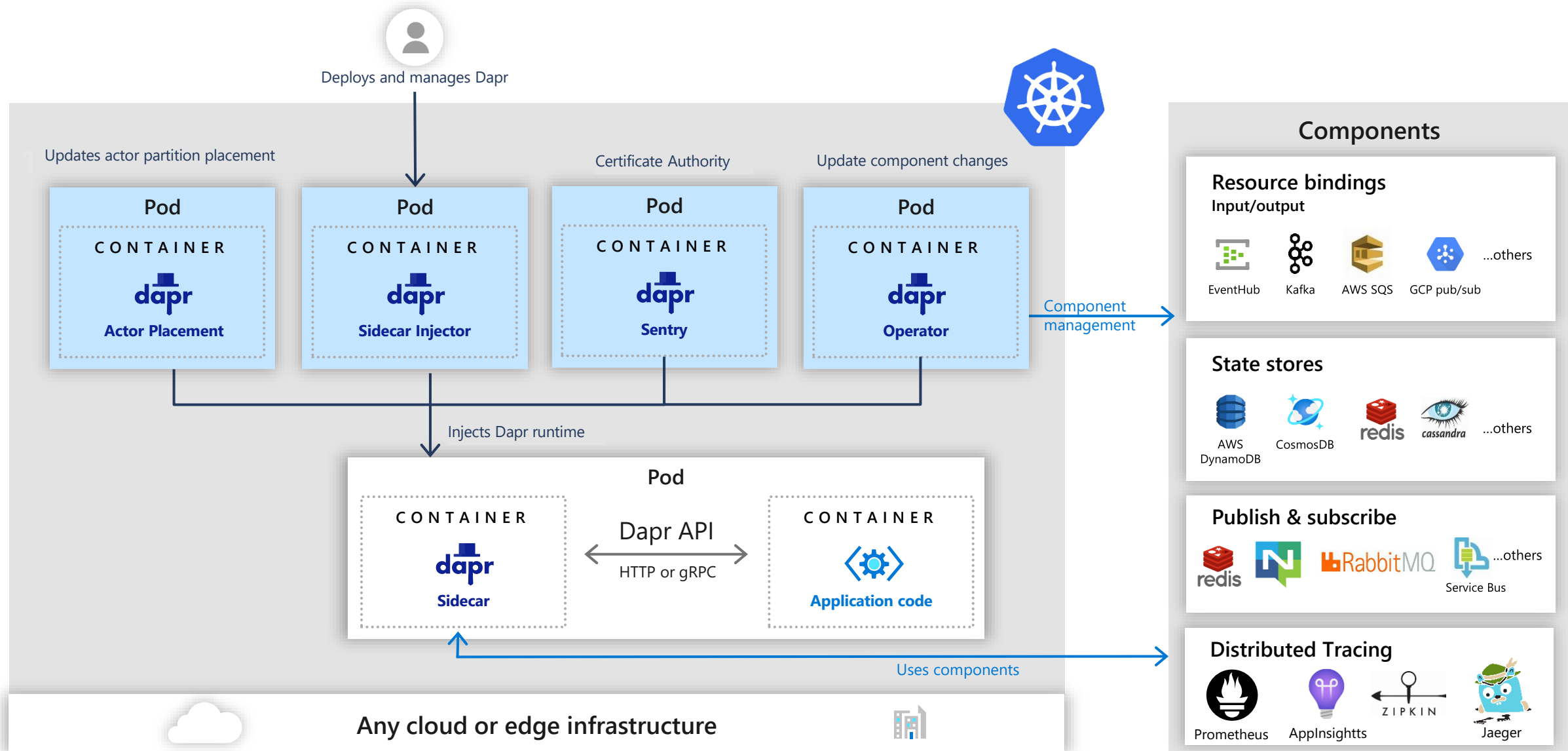


Dapr en local
DEMO

Service invocation



Exemple sur Kubernetes



Dapr sur Kubernetes

DEMO

Composants Dapr



Fichiers YAML avec les details de connexion à chaque service

Plus de 70 composants disponibles

State Stores



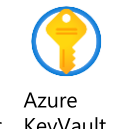
PubSub Brokers



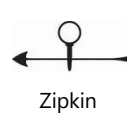
Bindings & Triggers



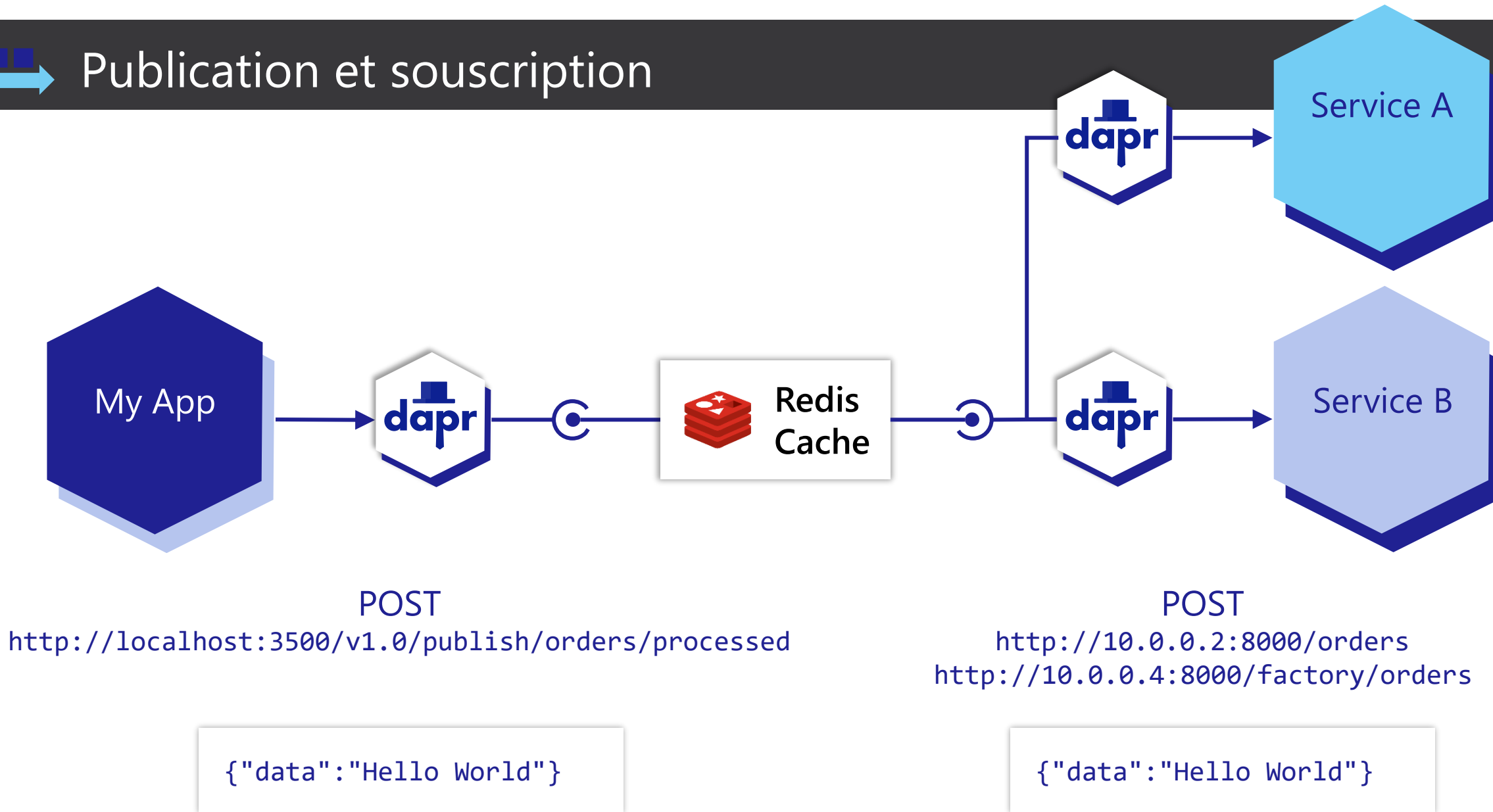
Secret Stores



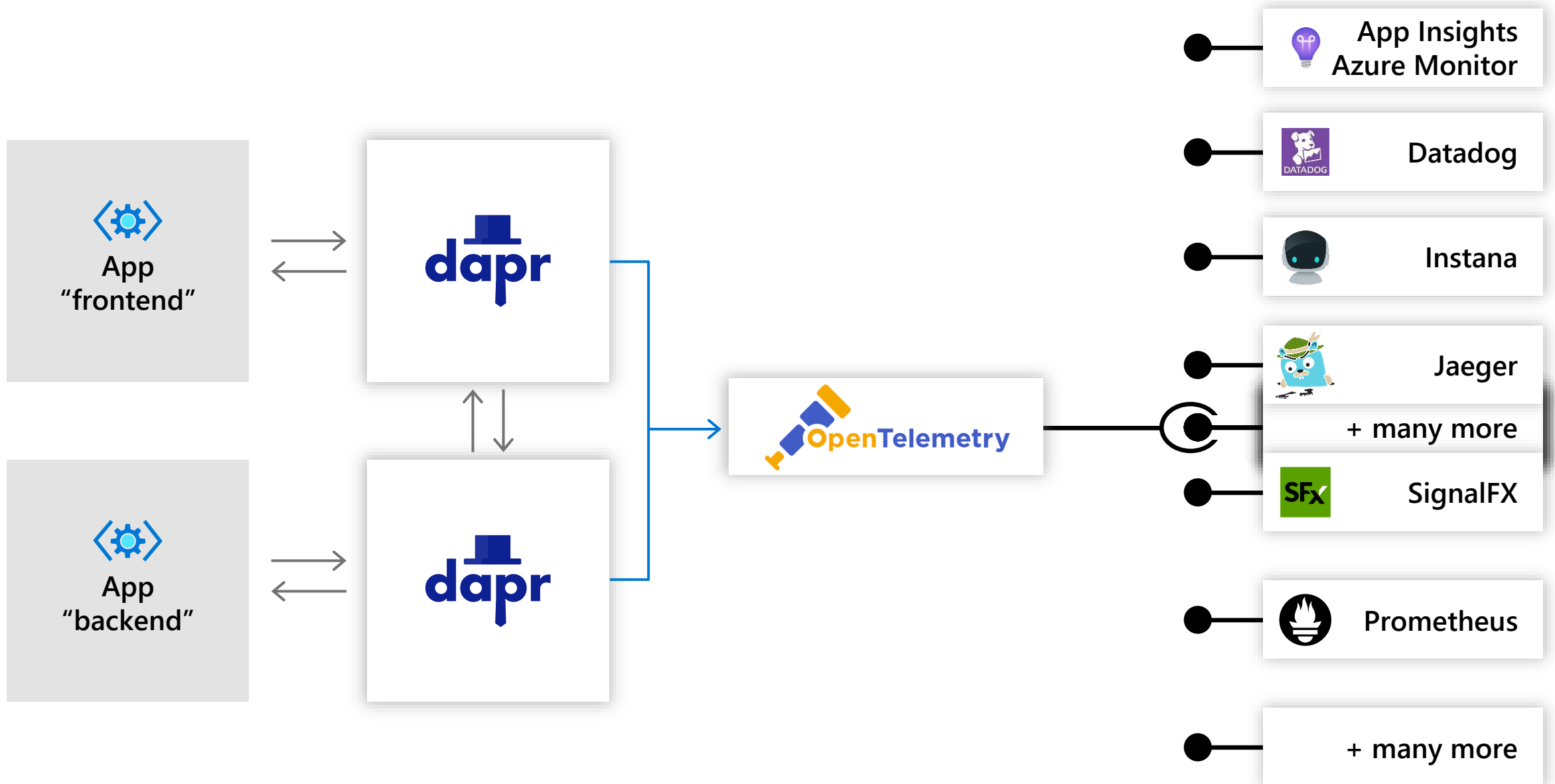
Observability



Publication et souscription



La collecte d'*insights* via un tronc commun



CNCF & Community



CLOUD NATIVE
COMPUTING FOUNDATION

Nov 2021: Accepté en tant que Project Incubé CNCF

17.9k

GitHub stars

1810

Contributors

97

Community
Components

+1M

Docker Hub
monthly pulls

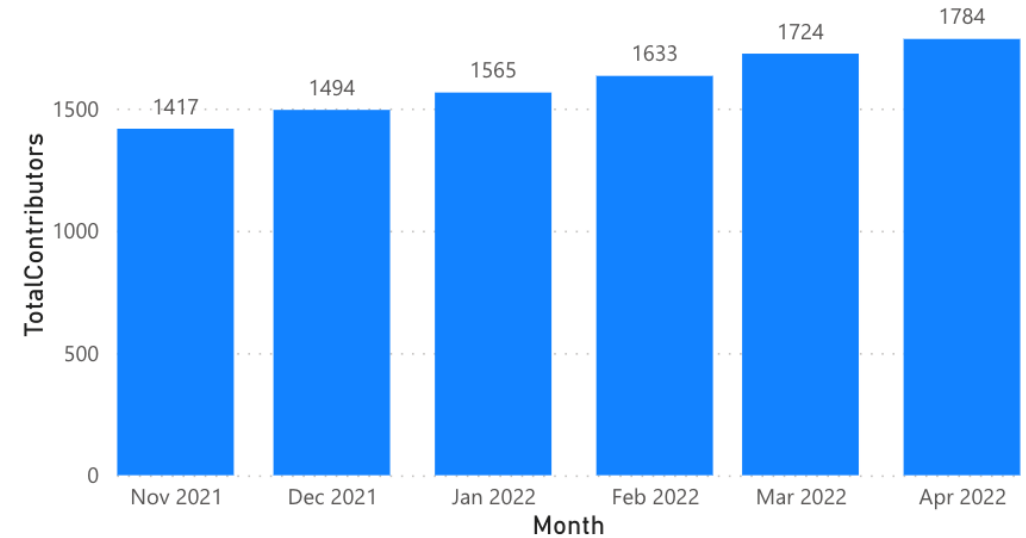
4k

Discord
members

+10k

Monthly Docs
views

Contributor growth



Contributing organizations include:



Microsoft

vmware®



HUAWEI



Alibaba Cloud



BOSCH

Tencent 腾讯

HashiCorp



Diagrid

Get involved



<https://github.com/dapr/dapr#community>
aka.ms/DaprDotnet



Questions ?



<https://careers.hagergroup.com/>



mtgstrasbourg



@MTG_Strasbourg



MTG Strasbourg

