



Einführung in die Entwicklung verteilter Anwendungen mit Dapr

Ricardo Niepel

Cloud Solution Architect - Azure App Dev

ricardo.niepel@microsoft.com



Herausforderungen von Enterprise- Entwicklern

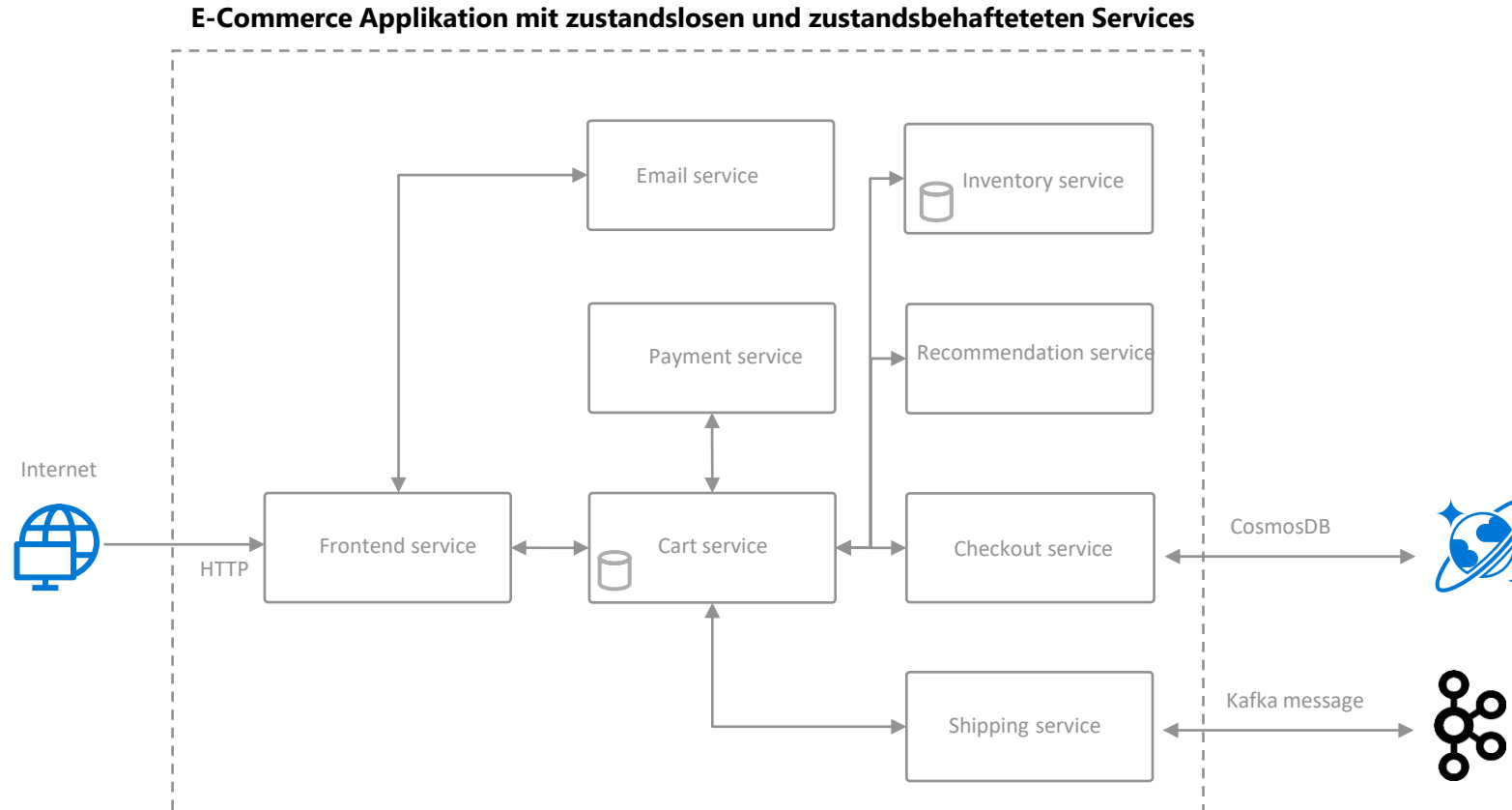
Entwicklung von zuverlässigen, skalierbaren, oft auf Microservices basierten, Systemen, welche mit einer Vielzahl von Diensten interagieren

Zunehmend polyglott, Nutzung verschiedener Programmiersprachen, Laufzeitumgebungen und Zielumgebungen

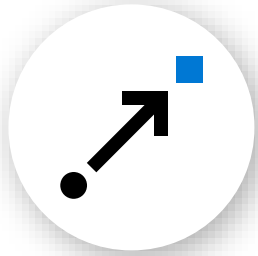
Bestehender Quellcode soll genutzt werden

Workflows, Aktoren und event-getriebene Funktionen sind leistungsfähige Programmiermodelle

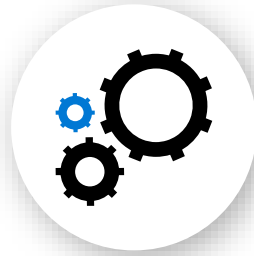
Entwicklung verteilter Anwendungen



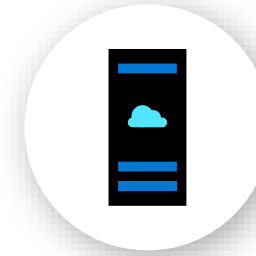
Was hemmt die Entwicklung von Microservices?



Inkrementelle Migration von vorhandenem Code zu einer Microservices Architektur ist schwierig



Umgebungen für Programmiermodelle haben eine eingeschränkte Sprachenunterstützung und einen engen Funktionsumfang



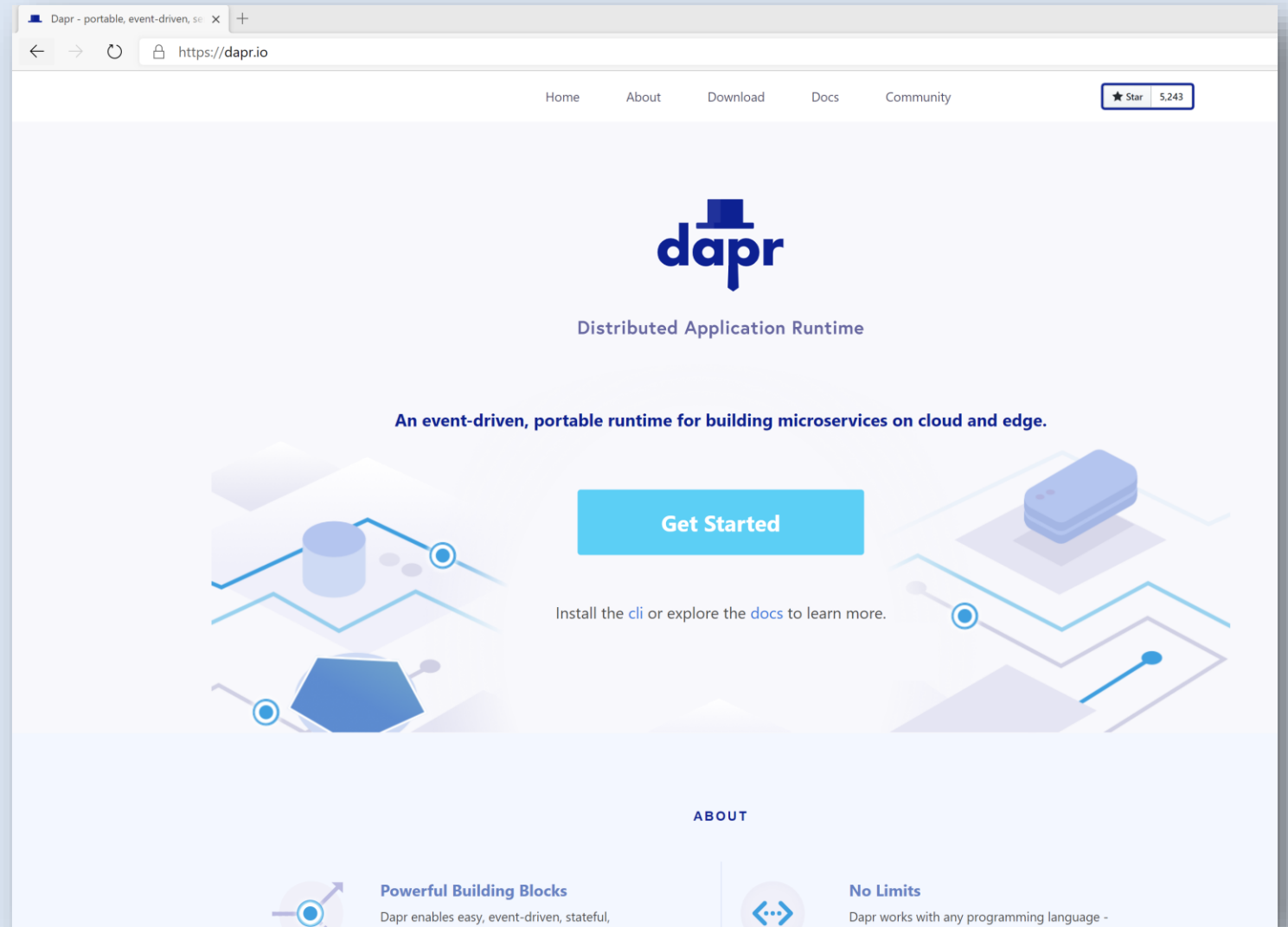
Laufzeitumgebungen fokussieren sich auf bestimmte Infrastrukturplattformen mit nur einer begrenzten Portabilität zwischen Cloud und Edge



Distributed Application Runtime

Portable und ereignisgesteuerte Laufzeitumgebung für die Entwicklung verteilter Anwendungen für Cloud und Edge

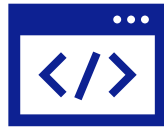
<https://dapr.io>



Dapr Ziele



Best-Practices
Bausteine



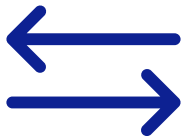
Jede Sprache oder
Framework



Community getrieben
Anbieter neutral



Nutzung von
Standards



Konsistent, Portabel,
Offene APIs



Plattform Agnostik
Cloud + Edge



Erweiterbar und
Pluggable Komponenten

Generelles Konzept von Dapr

- ✓ Standard APIs, welche über HTTP/gRPC angesprochen werden
<http://localhost:3500/v1.0/state/inventory/orderkey>
<http://localhost:3500/v1.0/invoke/myapp/method/neworder>
- ✓ Dapr läuft als "Sidecar Bibliothek", die dynamisch zur Laufzeit für jeden Dienst geladen wird

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



Observability

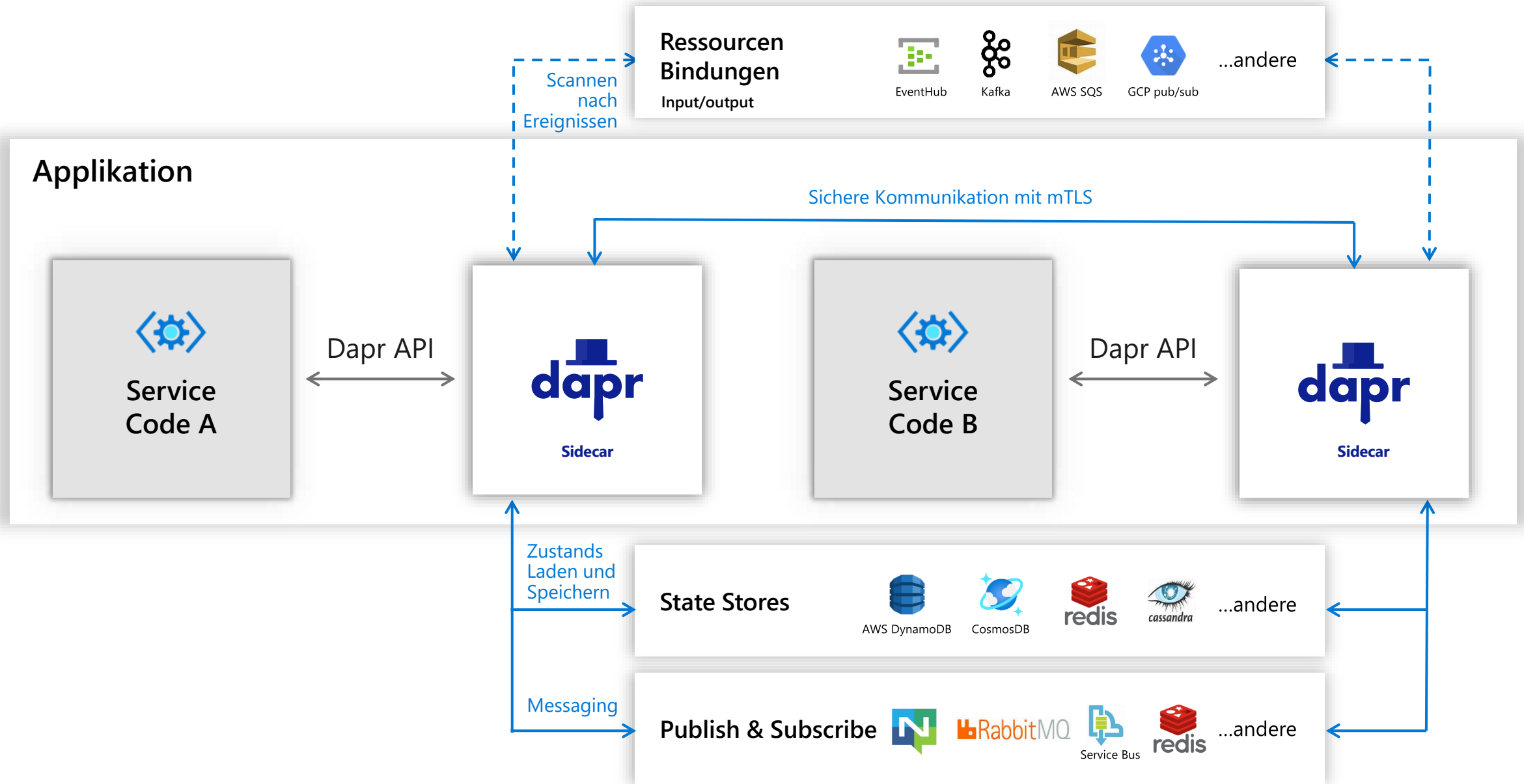


Secrets

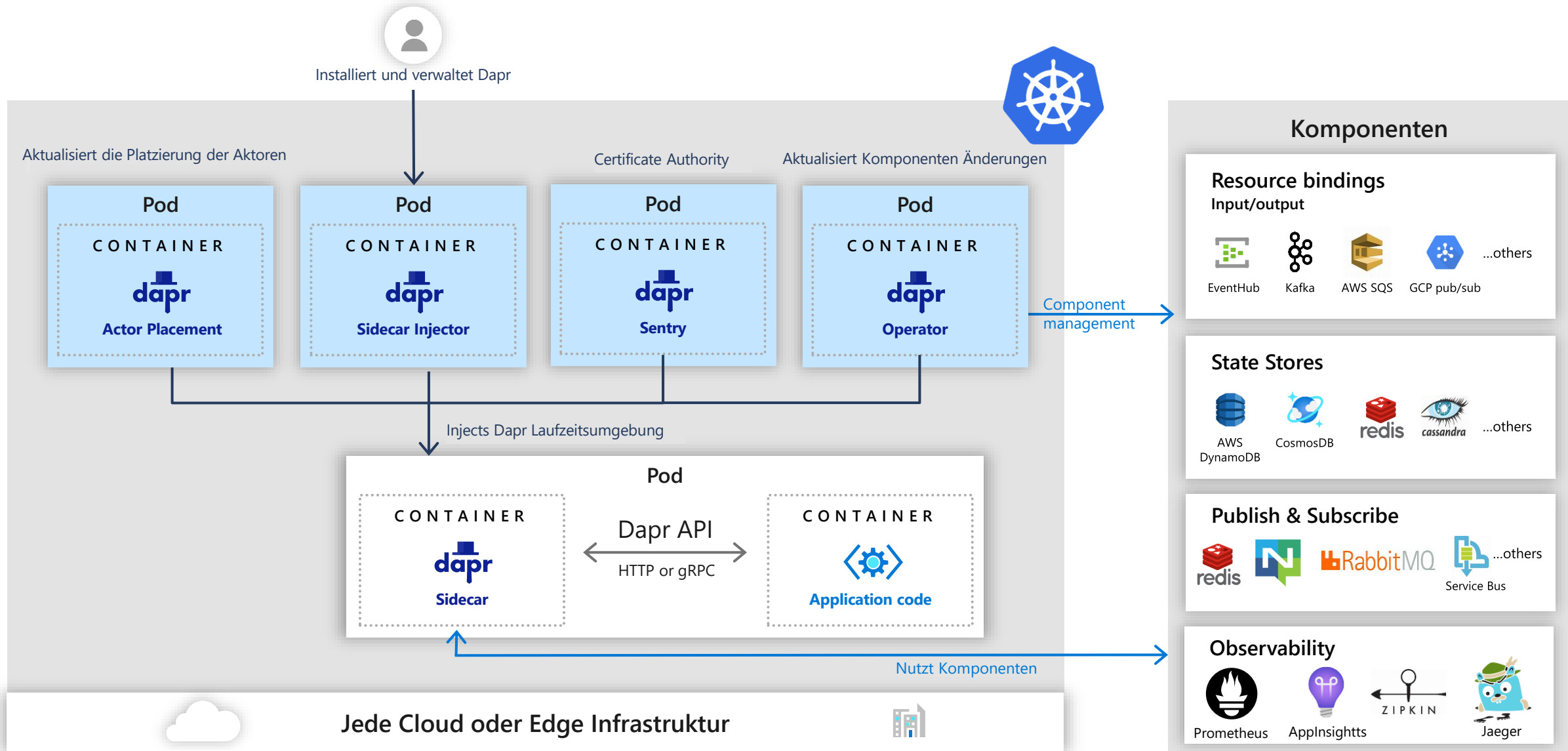


Extensible

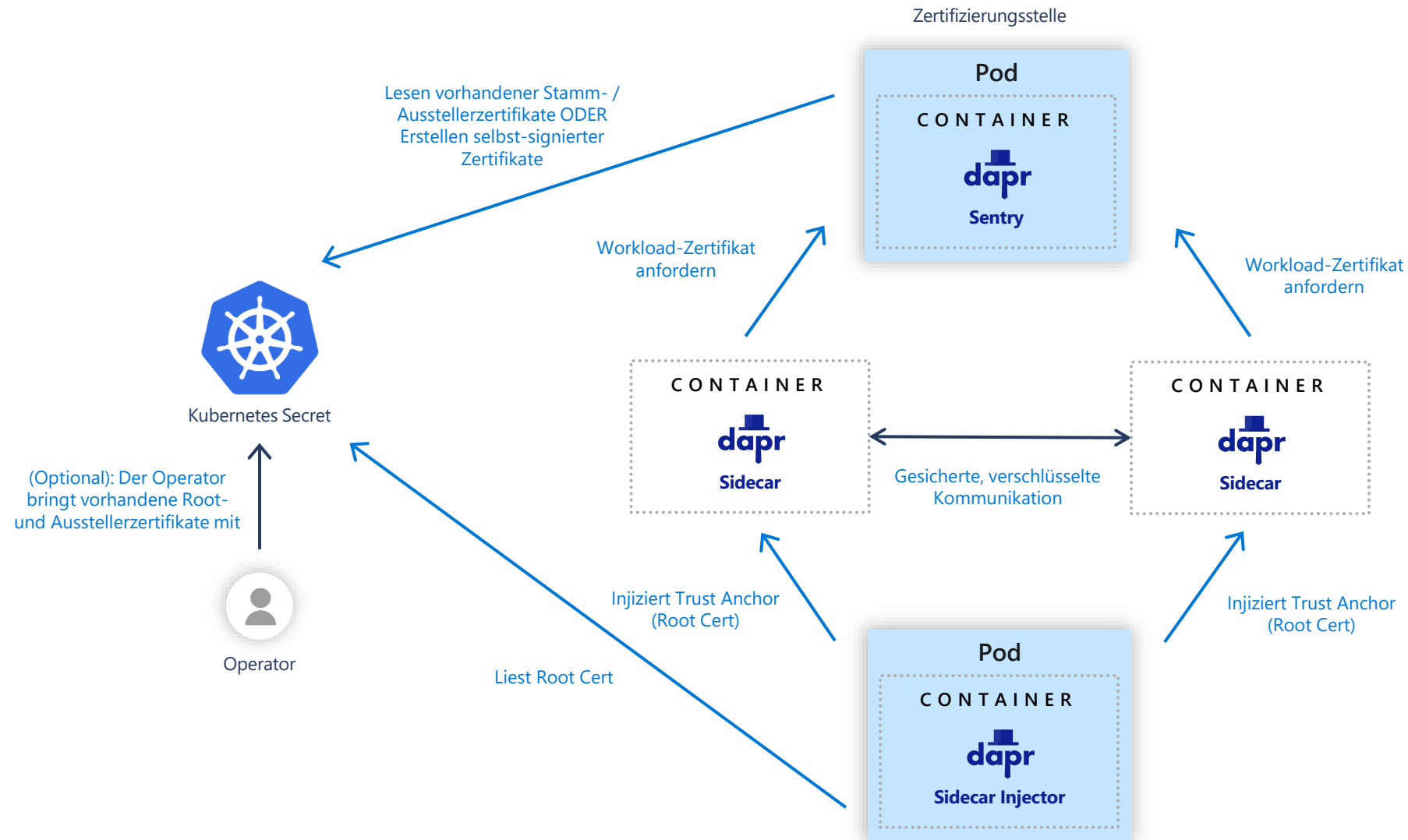
Sidecar und Komponenten Architektur



Dapr innerhalb von Kubernetes



Mutual TLS zwischen Dapr Instanzen

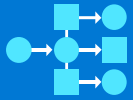


Microservice Bausteine



Verwenden Dapr Komponenten

Microservice Bausteine

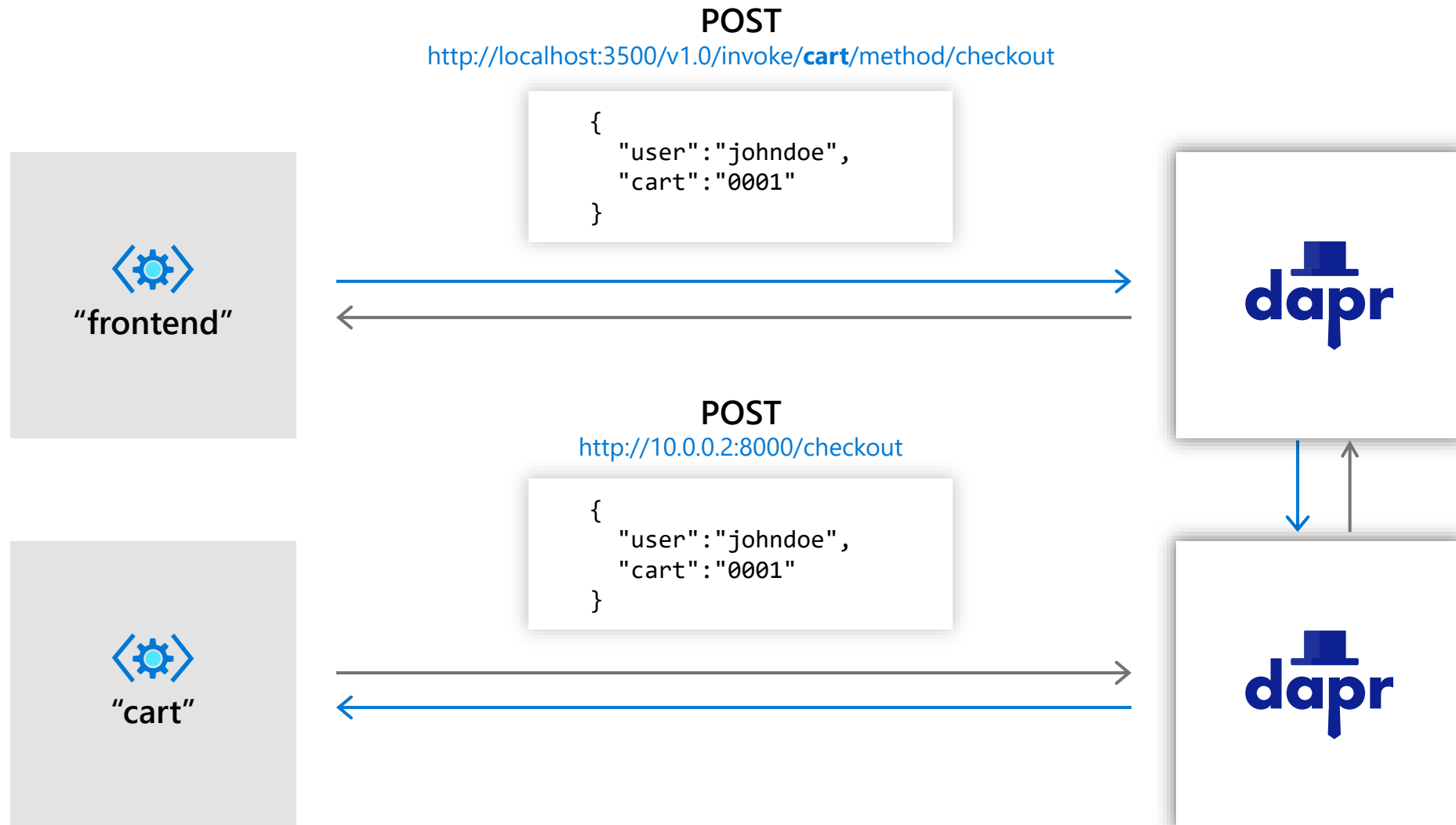


Service-to-
service
invocation

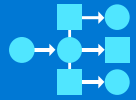
Direkte und
sichere Service zu
Service Aufrufe

Microservice Bausteine

Service Invocation



Microservice Bausteine



Service-to-service invocation

Direkte und sichere Service zu Service Aufrufe

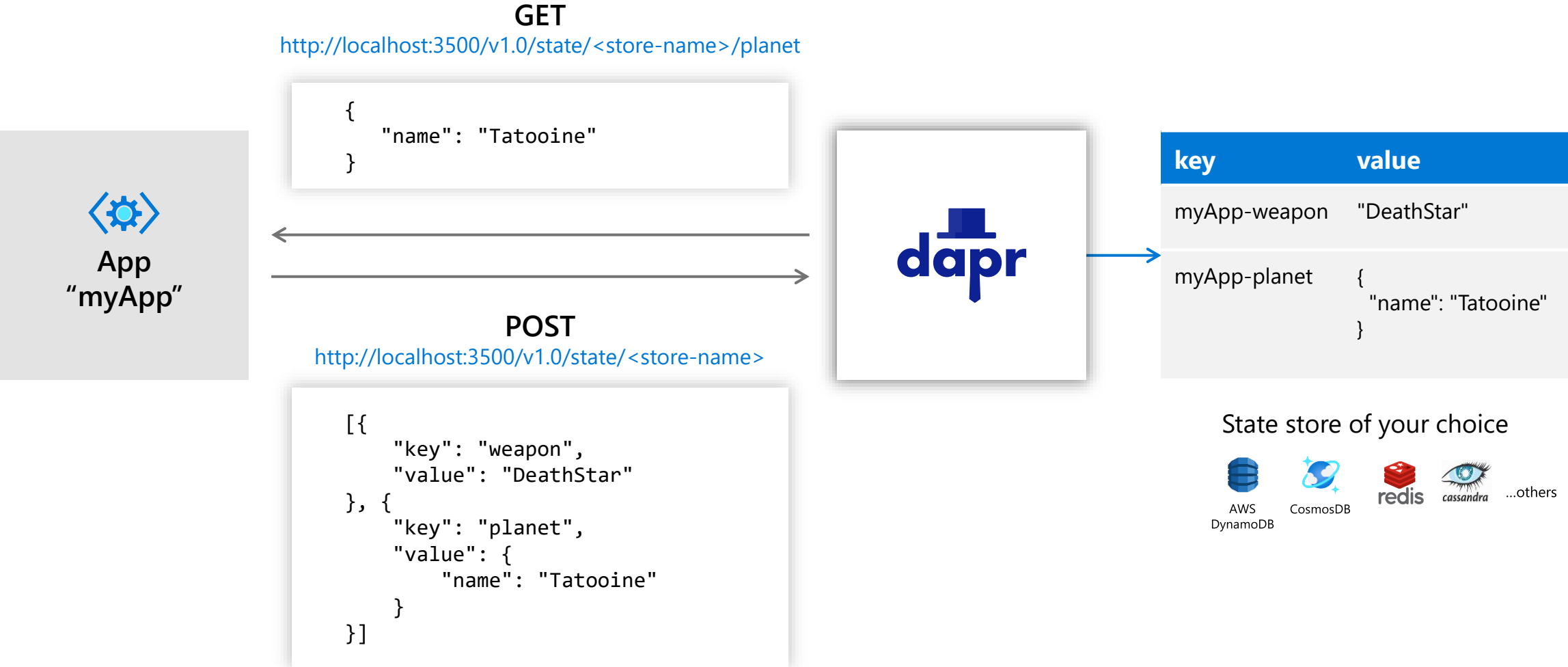


State management

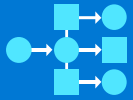
Lang laufende, zustandslose und zustands-behaftete Services

Microservice Bausteine

State Management: key/value



Microservice Bausteine



Service-to-service invocation

Direkte und sichere Service zu Service Aufrufe



State management

Lang laufende, zustandslose und zustands-behaftete Services

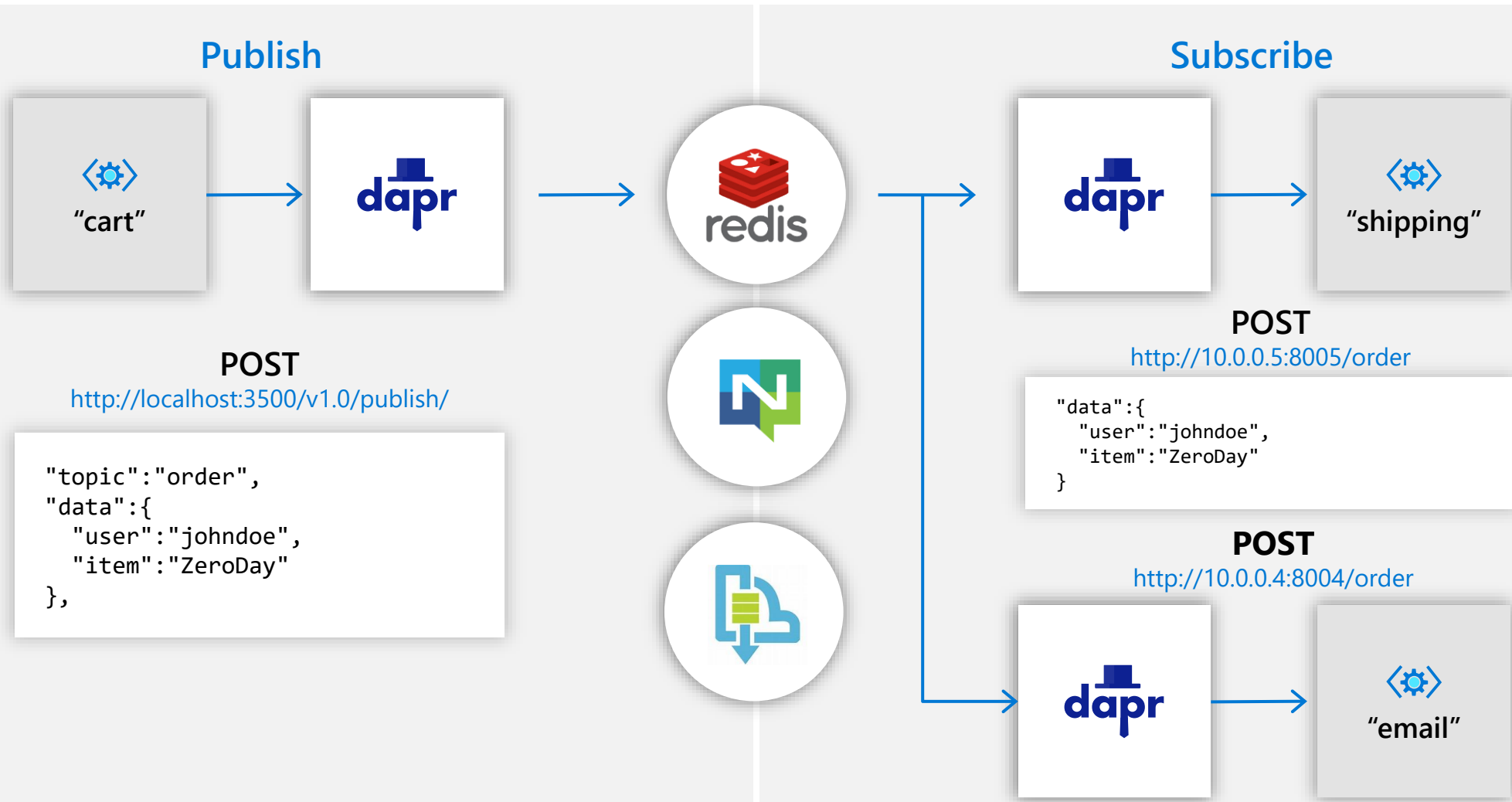


Publish and subscribe

Sicherer und skalierbarer Nachrichtenaustausch zwischen Services

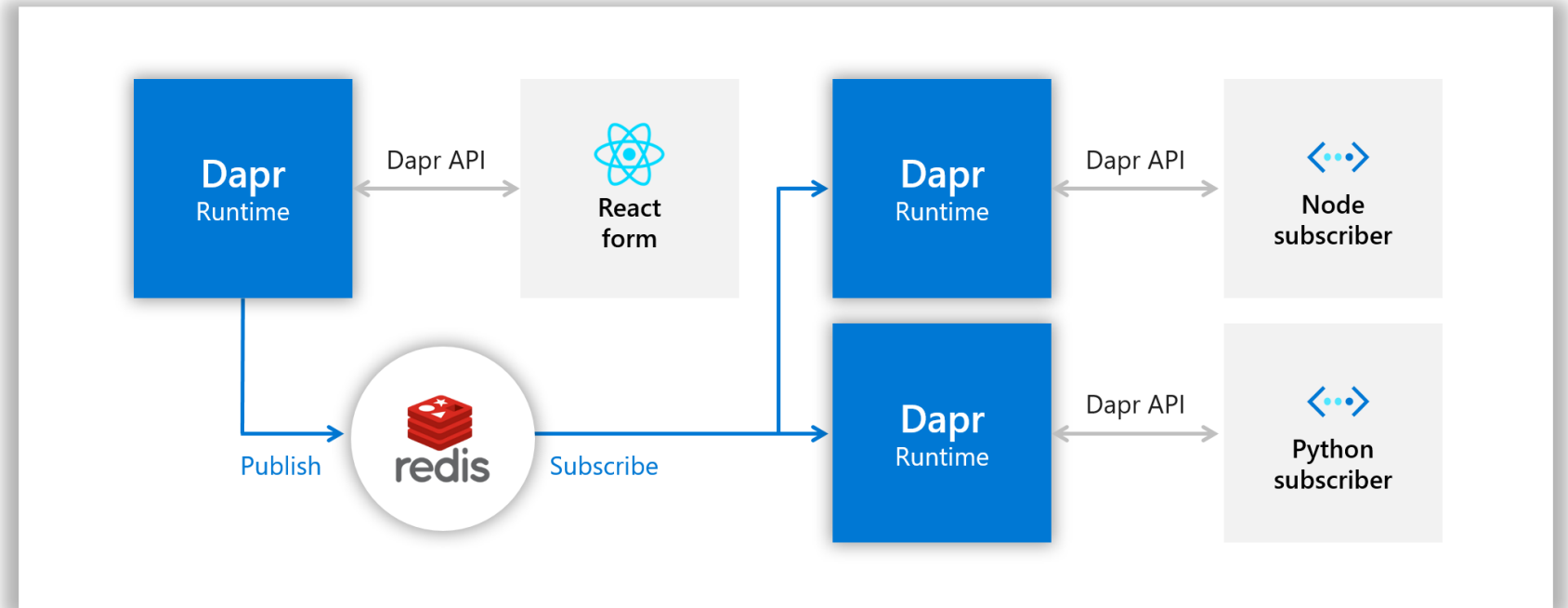
Microservice Bausteine

Publish and Subscribe

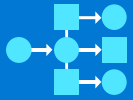


Demo

Publishing & Subscribing



Microservice Bausteine



Service-to-service invocation

Direkte und sichere Service zu Service Aufrufe



State management

Lang laufende, zustandslose und zustands-behaftete Services



Publish and subscribe

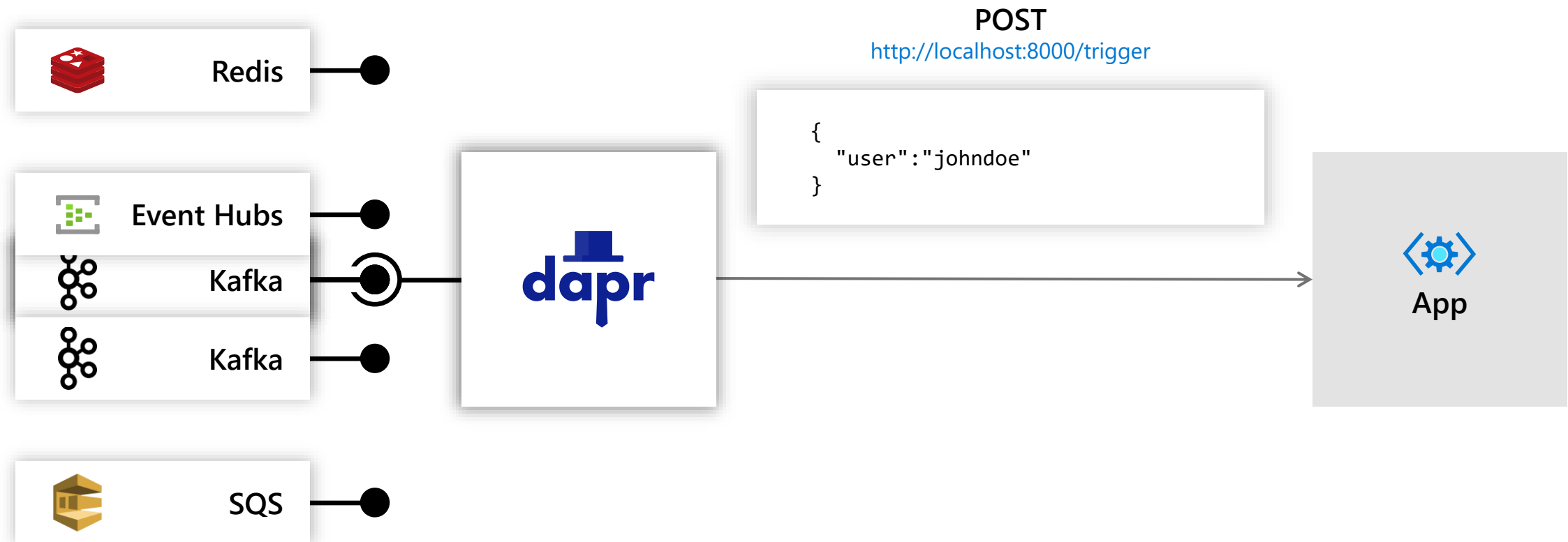
Sicherer und skalierbarer Nachrichtenaustausch zwischen Services



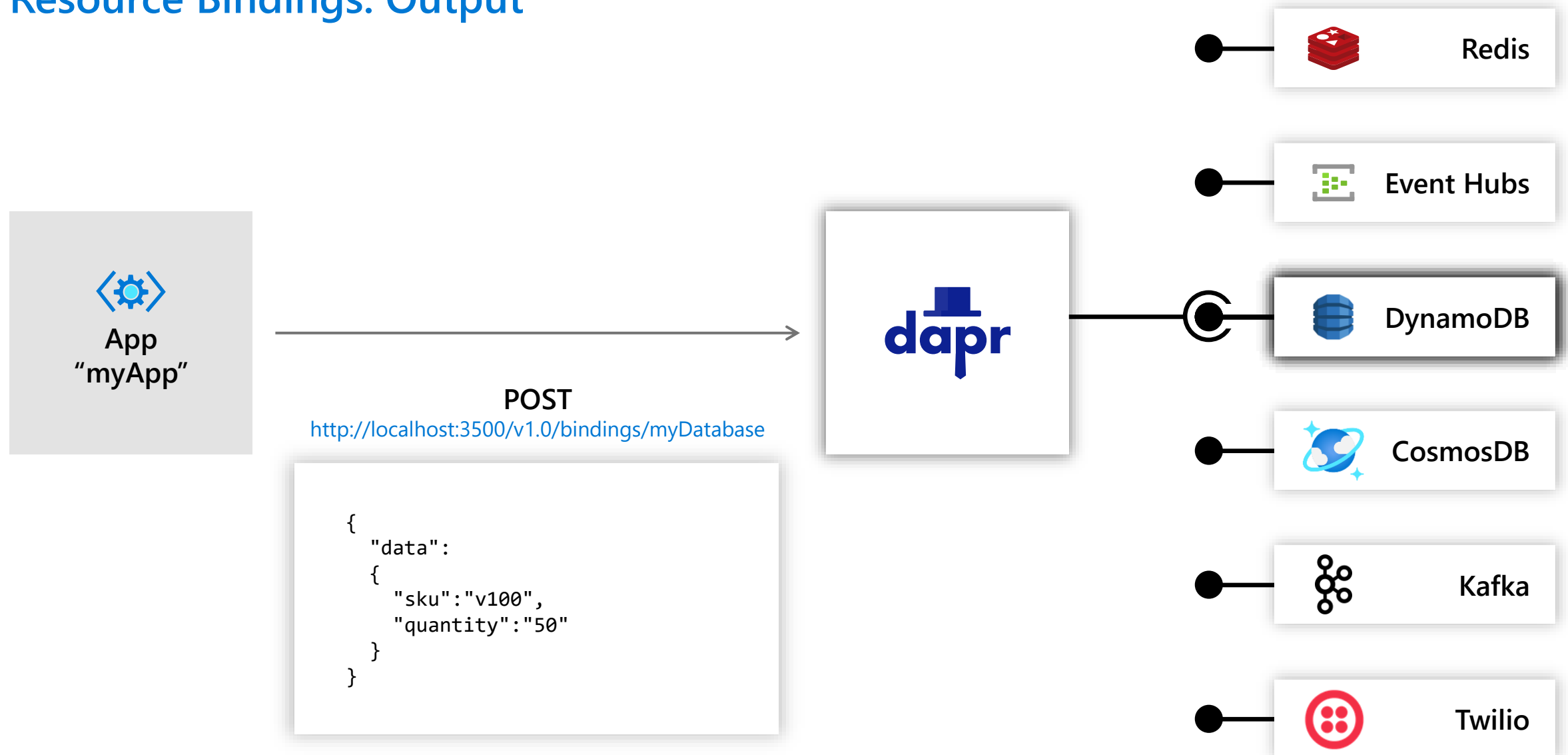
Resource bindings and triggers

Ereignisgetriebene Ausführung von Code durch zahlreiche Trigger
Ausgabeverbindungen zu externen Ressourcen

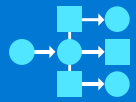
Resource Bindings: Input Trigger



Resource Bindings: Output



Microservice Bausteine



Service-to-service invocation

Direkte und sichere Service zu Service Aufrufe



State management

Lang laufende, zustandslose und zustands-behaftete Services



Publish and subscribe

Sicherer und skalierbarer Nachrichtenaustausch zwischen Services



Resource bindings and triggers

Ereignisgetriebene Ausführung von Code durch zahlreiche Trigger
Ausgabeverbindungen zu externen Ressourcen



Actors

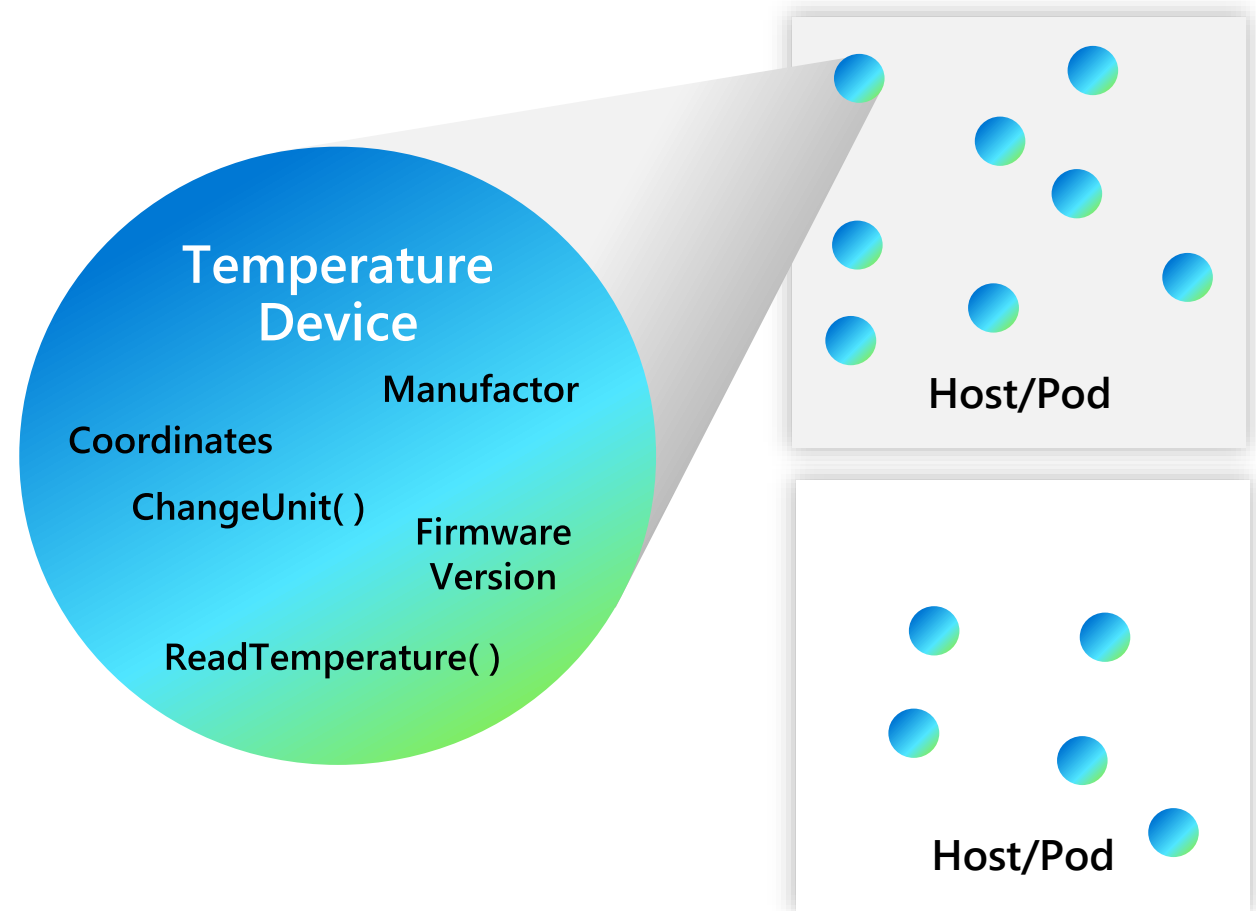
Kapselung von Code und Daten in wiederverwendbare Aktoren als häufiges Microservices Entwurfsmuster

Virtuelles Aktorenmodell mit Dapr

Einheiten von Zustand und Verhalten

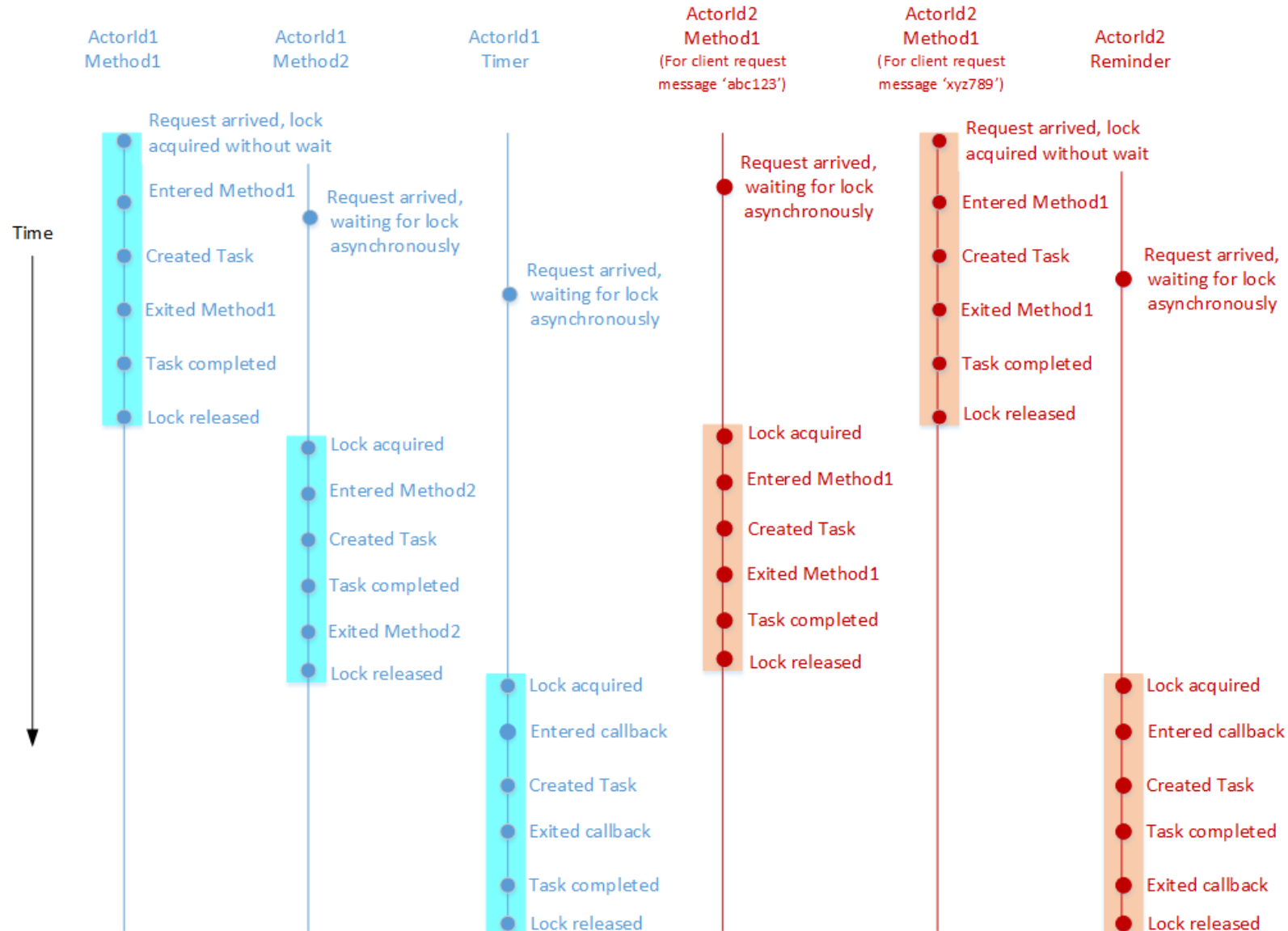
Dapr Aktoren Funktionalitäten:

- ✓ Verteilung
- ✓ Ausfallsicherheit
- ✓ State Management
- ✓ Zeitgeber
- ✓ Erinnerungen
- ✓ Rundenbasierte Nebenläufigkeit

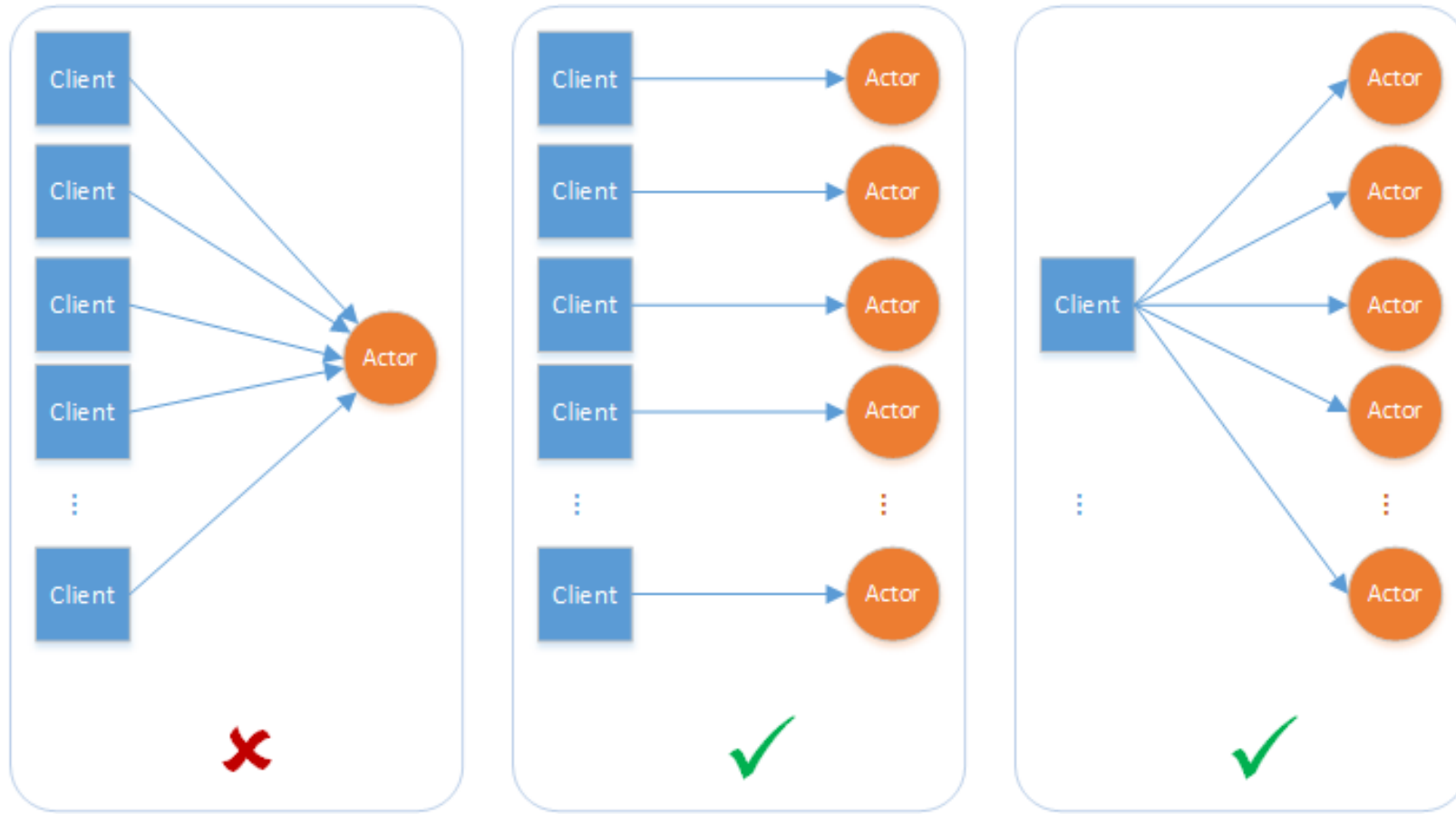


Praktisch identisch zu Service Fabric Reliable Actors

Actors - Rundenbasierte Nebenläufigkeit

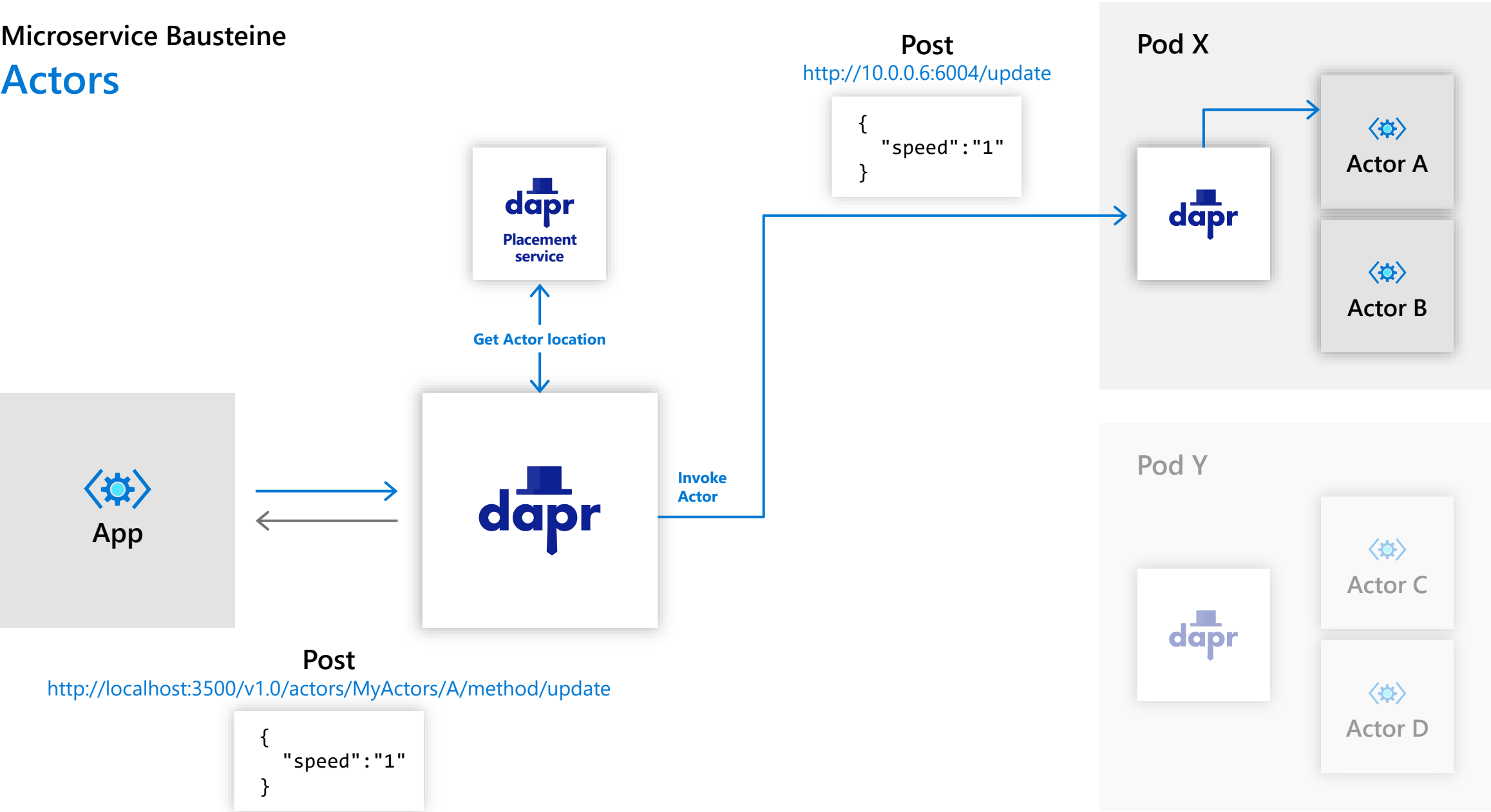


Actors - Rundenbasierte Nebenläufigkeit



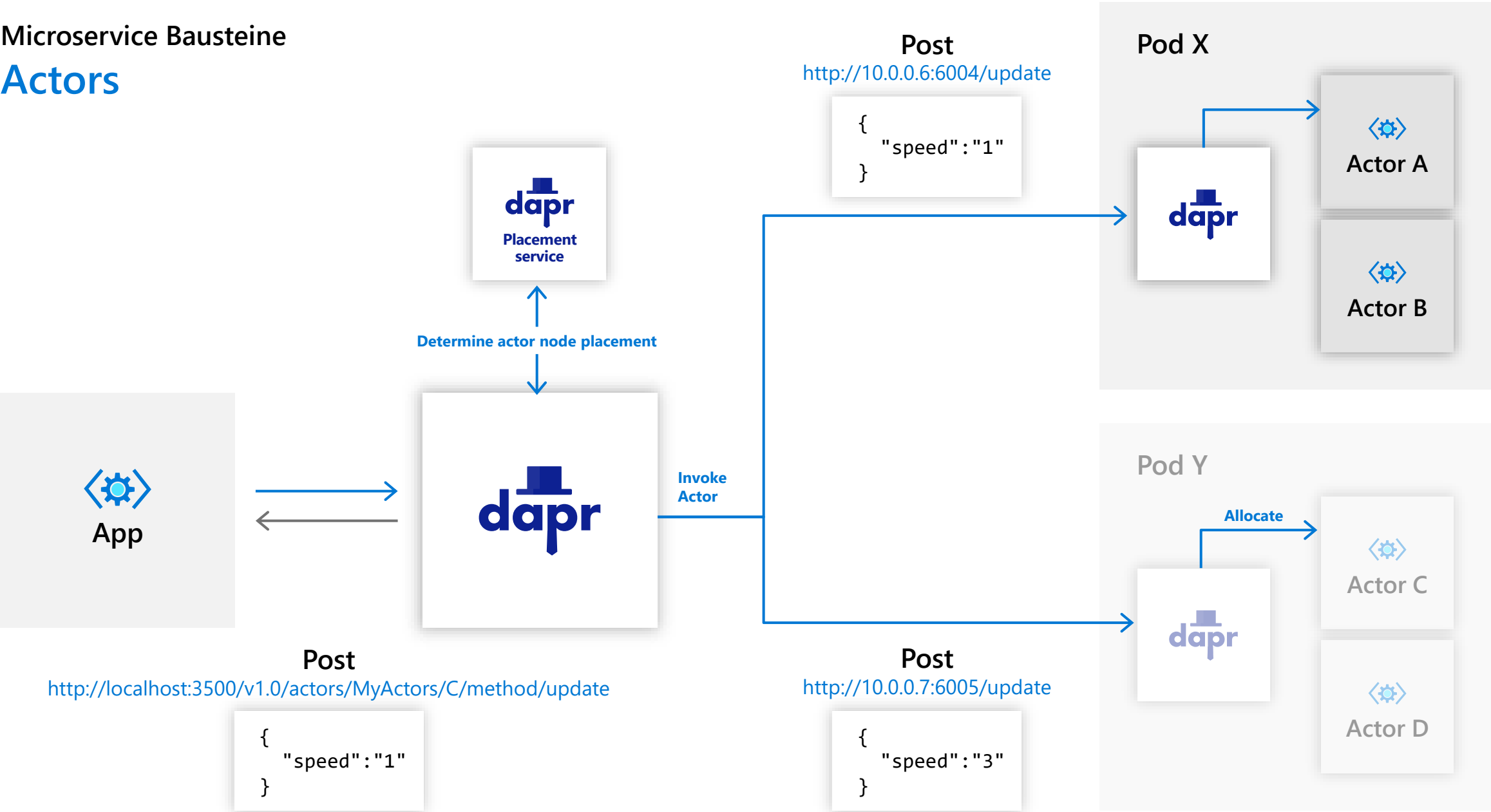
Microservice Bausteine

Actors

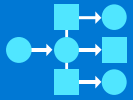


Microservice Bausteine

Actors



Microservice Bausteine



Service-to-service invocation

Direkte und sichere Service zu Service Aufrufe



State management

Lang laufende, zustandslose und zustands-behaftete Services



Publish and subscribe

Sicherer und skalierbarer Nachrichtenaustausch zwischen Services



Resource bindings and triggers

Ereignisgetriebene Ausführung von Code durch zahlreiche Trigger
Ausgabeverbindungen zu externen Ressourcen



Actors

Kapselung von Code und Daten in wiederverwendbare Aktoren als häufiges Microservices Entwurfsmuster

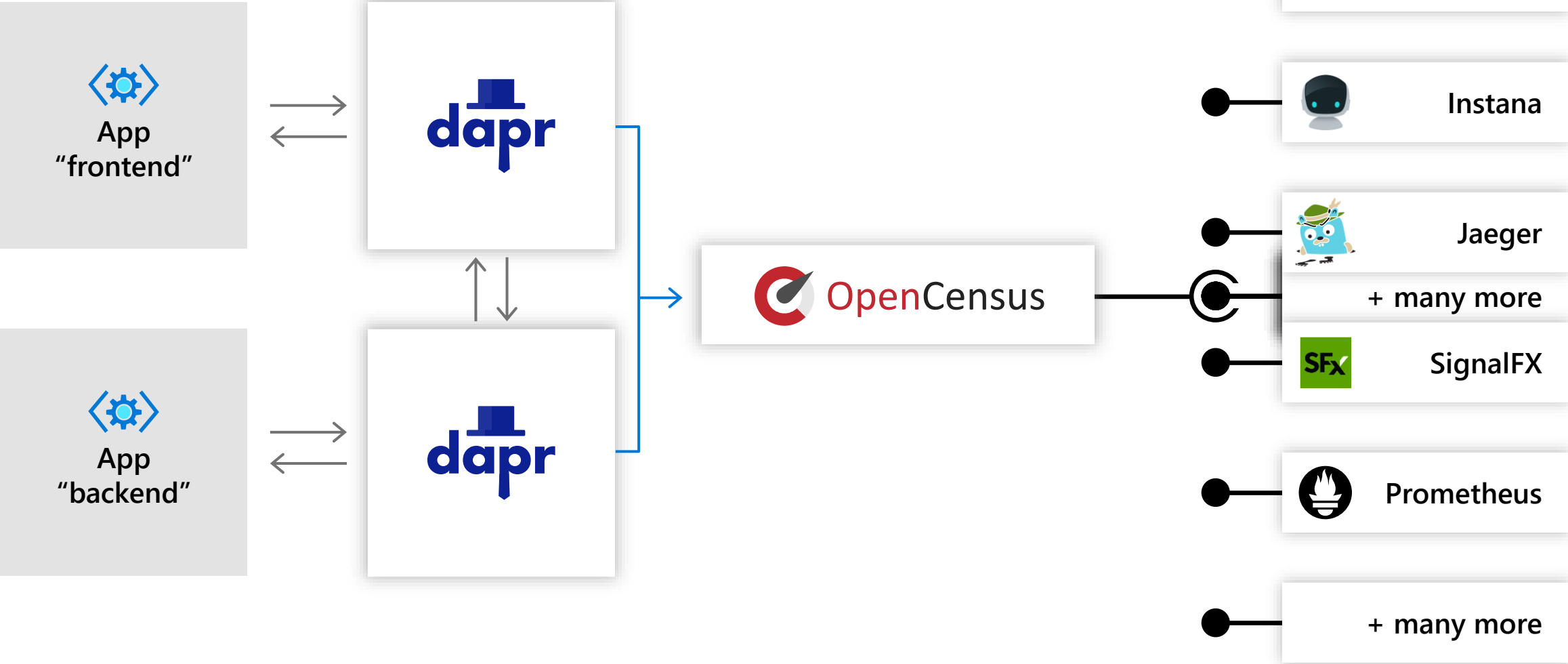


Observability

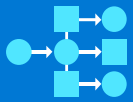
Debuggen, Betreiben und Überwachen von Dapr-Systemdiensten, Komponenten und Anwendungen

Microservice Bausteine

Observability



Microservice Bausteine



Service-to-service invocation

Direkte und sichere Service zu Service Aufrufe



State management

Lang laufende, zustandslose und zustands-behaftete Services



Publish and subscribe

Sicherer und skalierbarer Nachrichtenaustausch zwischen Services



Resource bindings and triggers

Ereignisgetriebene Ausführung von Code durch zahlreiche Trigger
Ausgabeverbindungen zu externen Ressourcen



Actors

Kapselung von Code und Daten in wiederverwendbare Aktoren als häufiges Microservices Entwurfsmuster



Observability

Debuggen, Betreiben und Überwachen von Dapr-Systemdiensten, Komponenten und Anwendungen

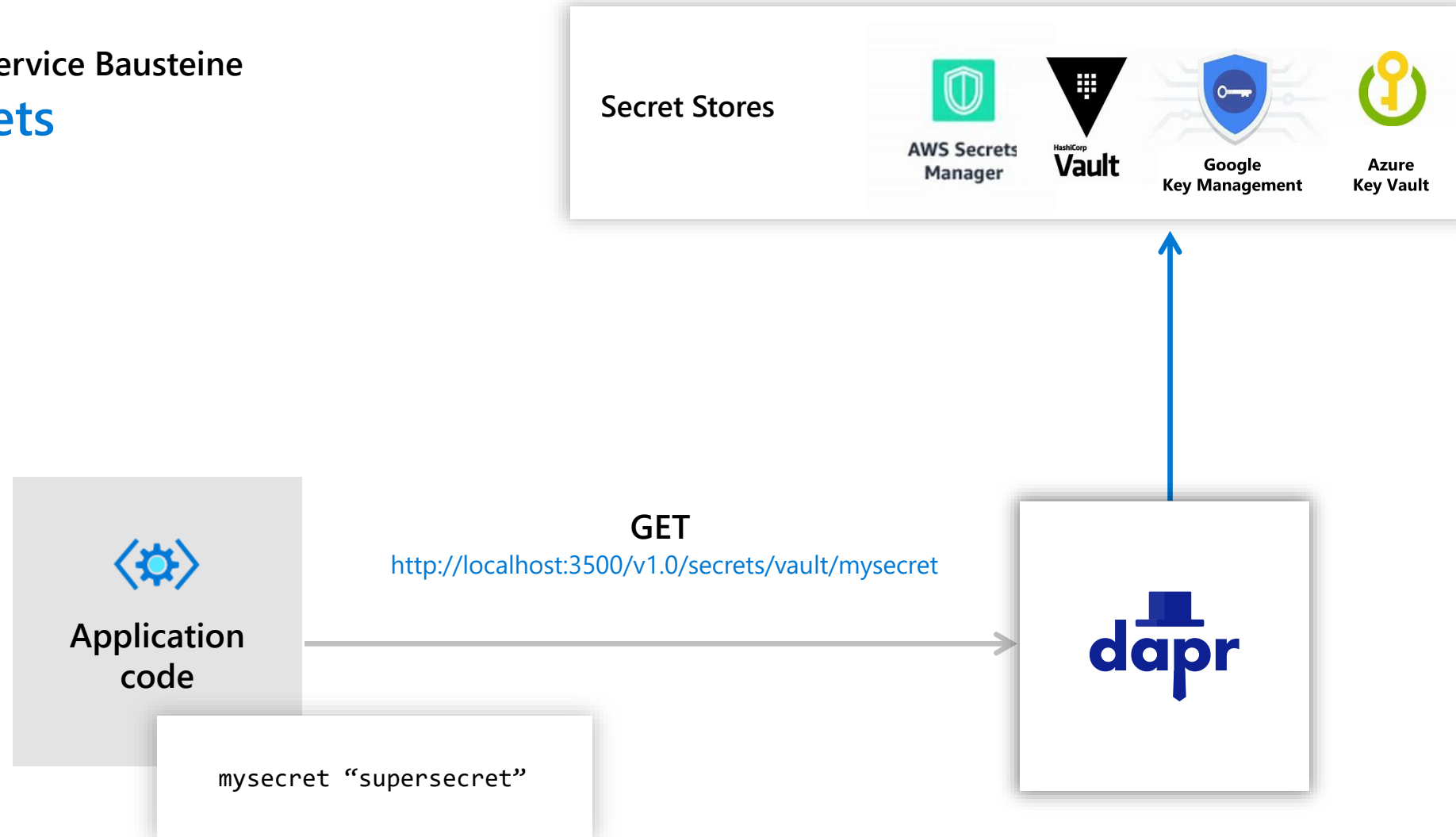


Secrets

Sicherer Zugriff auf Schlüssel und Connection Strings

Microservice Bausteine

Secrets

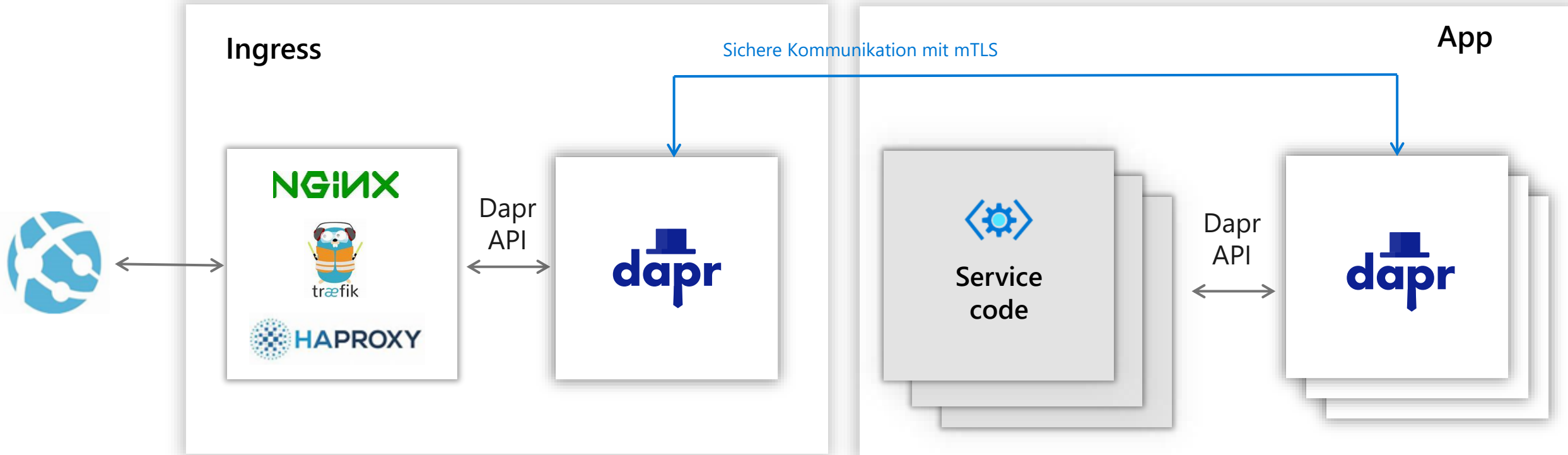


Häufige Fragen?

- Funktioniert Dapr mit Reverse-Proxy-Servern wie NGINX, Traefik?
- Funktioniert Dapr mit Autoscalern wie KEDA?
- Ist Dapr ein Service-Mesh?
- Ist Dapr erweiterbar?

Sichere Ingress Kommunikation

Daprize Proxy Server



KEDA

Kubernetes-basierte ereignisgesteuerte
Autoskalierung

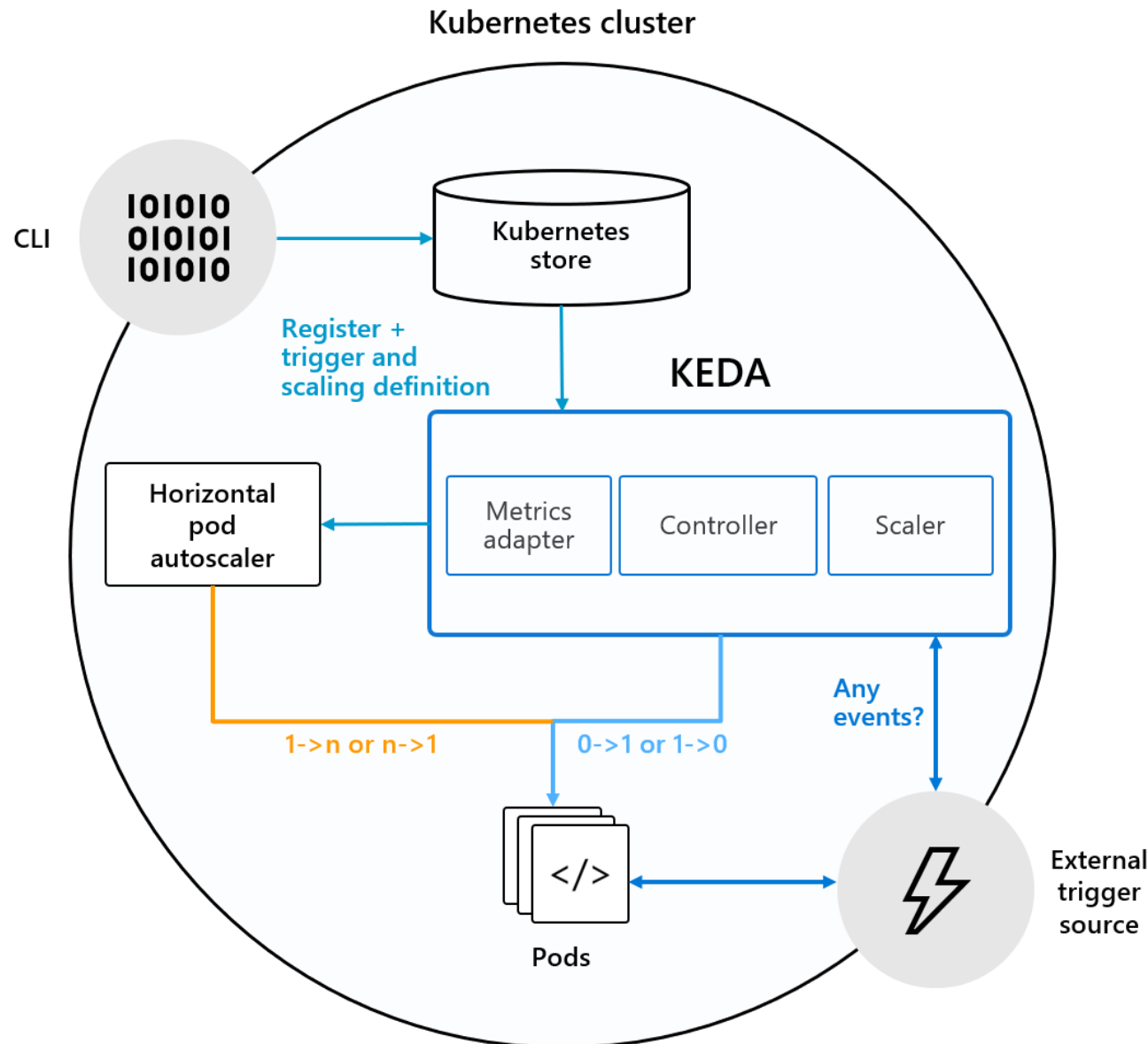
Open-Source-Komponente zur
Bereitstellung „Serverless/Functions“-
ähnlicher Skalierung in Kubernetes

Skalierung auf Null oder
Skalierung auf Tausende

<https://github.com/kedacore/keda>



Wie KEDA funktioniert



Basis

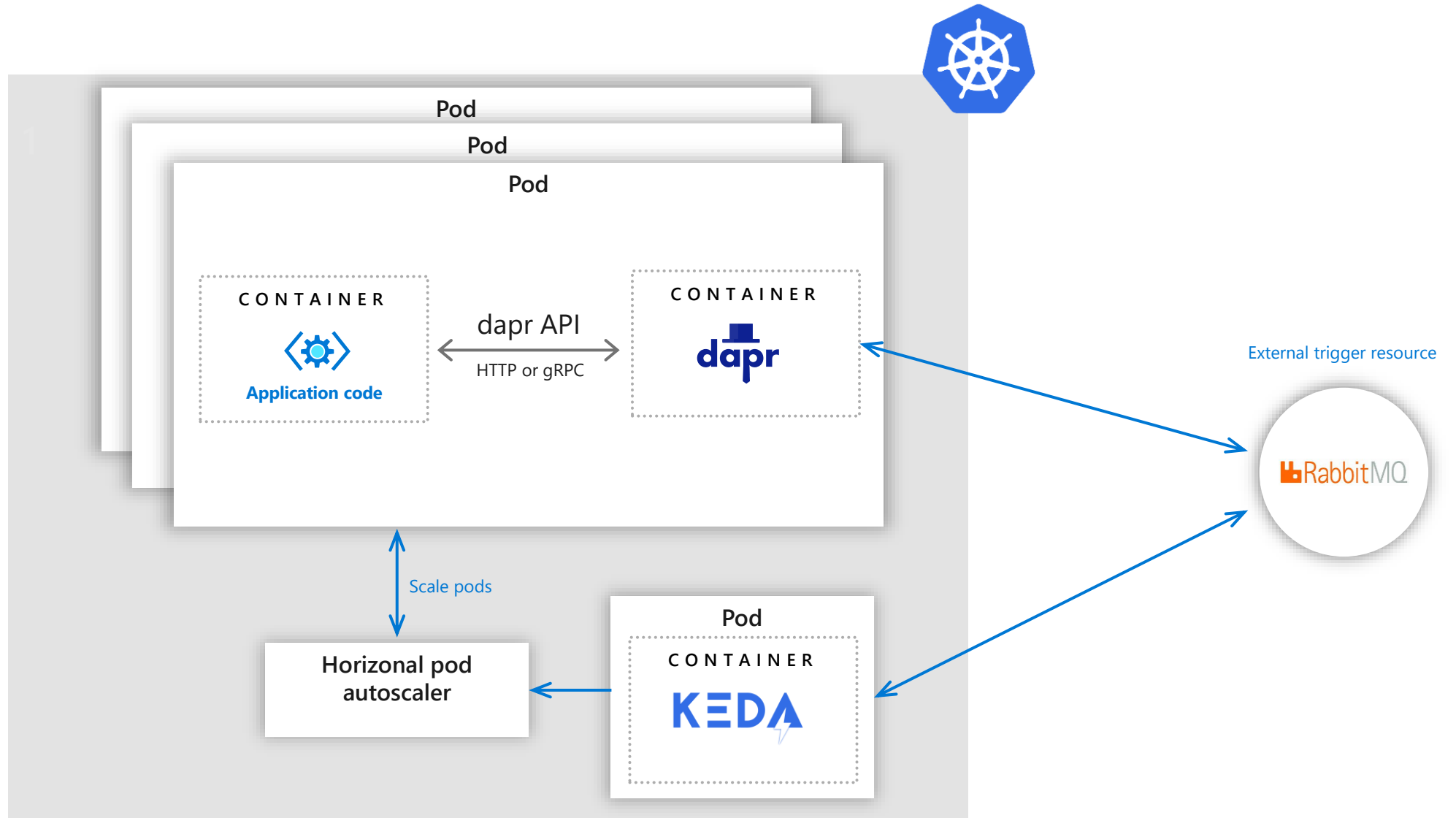
- Skaliert bis und von 0 bei keinen Ereignissen
- Agiert als K8s Metrics Server

Event Scalers

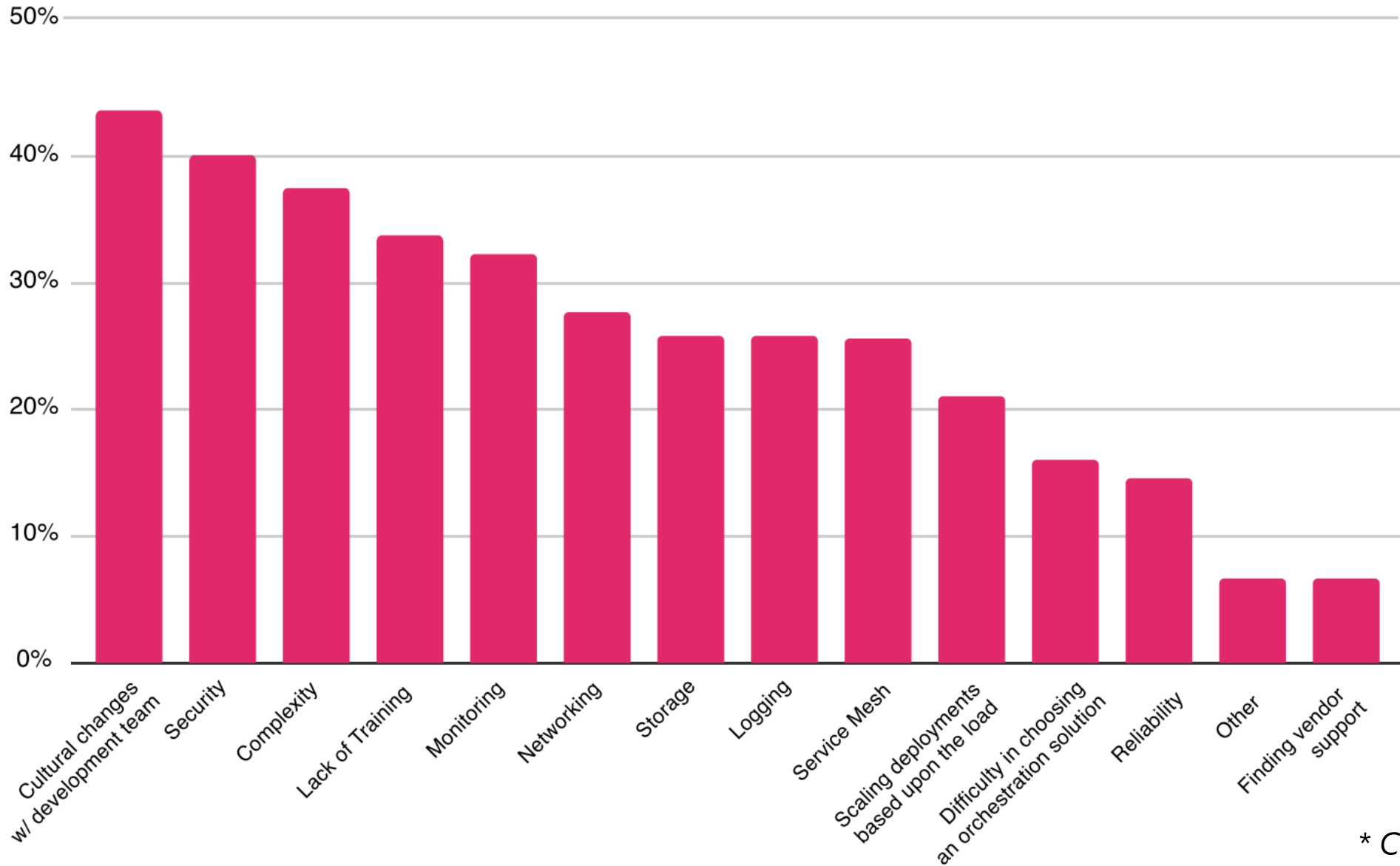
- Kafka
- RabbitMQ
- Azure Service Bus
- Azure Event Hub
- Azure Blob Storage
- Azure Monitor Metrics
- AWS Kinesis
- AWS SQS Queue
- AWS Cloudwatch Metrics
- GCP Pub/Sub
- ... *many more*

Integrates with Azure Function & Dapr

Dapr und KEDA

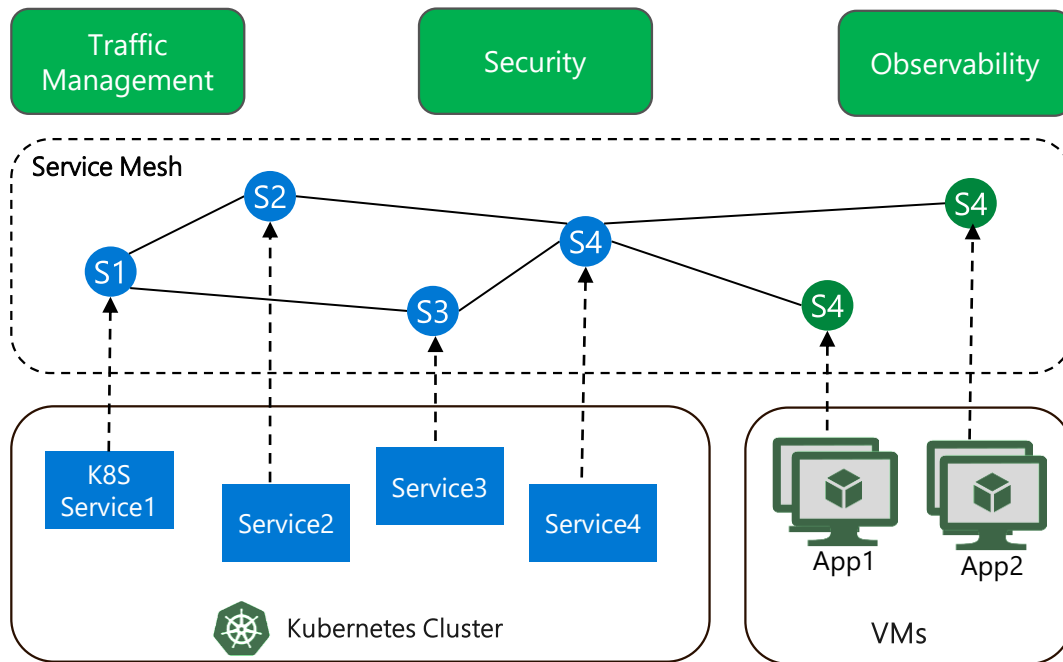


Challenges in using & deploying containers*



* CNCF Survey 2019

Was ist ein Service Mesh?



Reduziert den Netzwerkcode in Anwendungen
Netzwerkfunktionen werden auf Sidecar-Proxys verlagert, die neben der Geschäftslogik laufen und von einer Steuerungsebene verwaltet werden

Traffic Management

Anfrage-Routing, gewichteter Lastausgleich, Fault Injection, Circuit Breaker Muster usw.

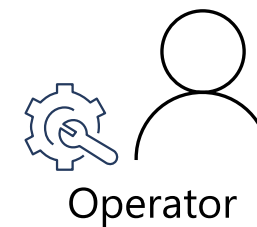
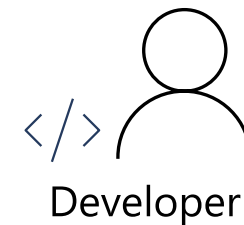
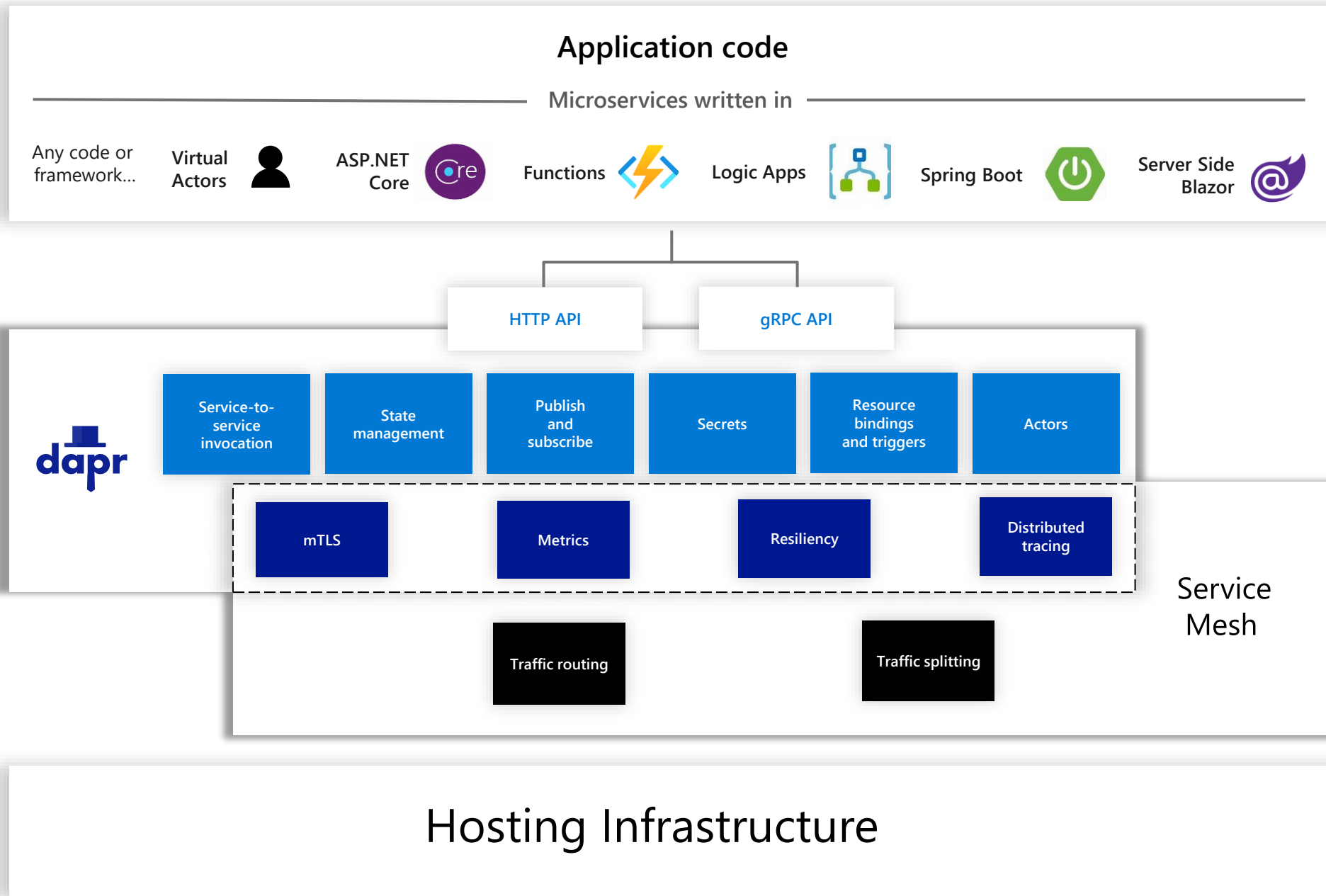
Sicherheit

mTLS-Verschlüsselung zwischen Anwendungen/ Diensten; Authentifizierung/ Autorisierung für die Kommunikation zwischen Diensten

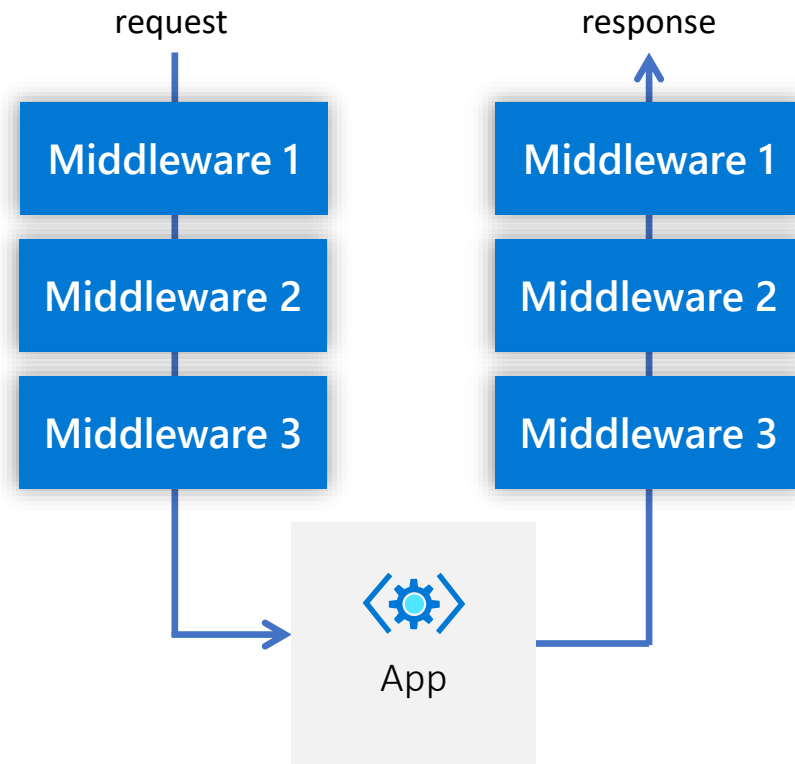
Observability

Traffic Tracing; Sichtbarkeit in Verbindungen etc.

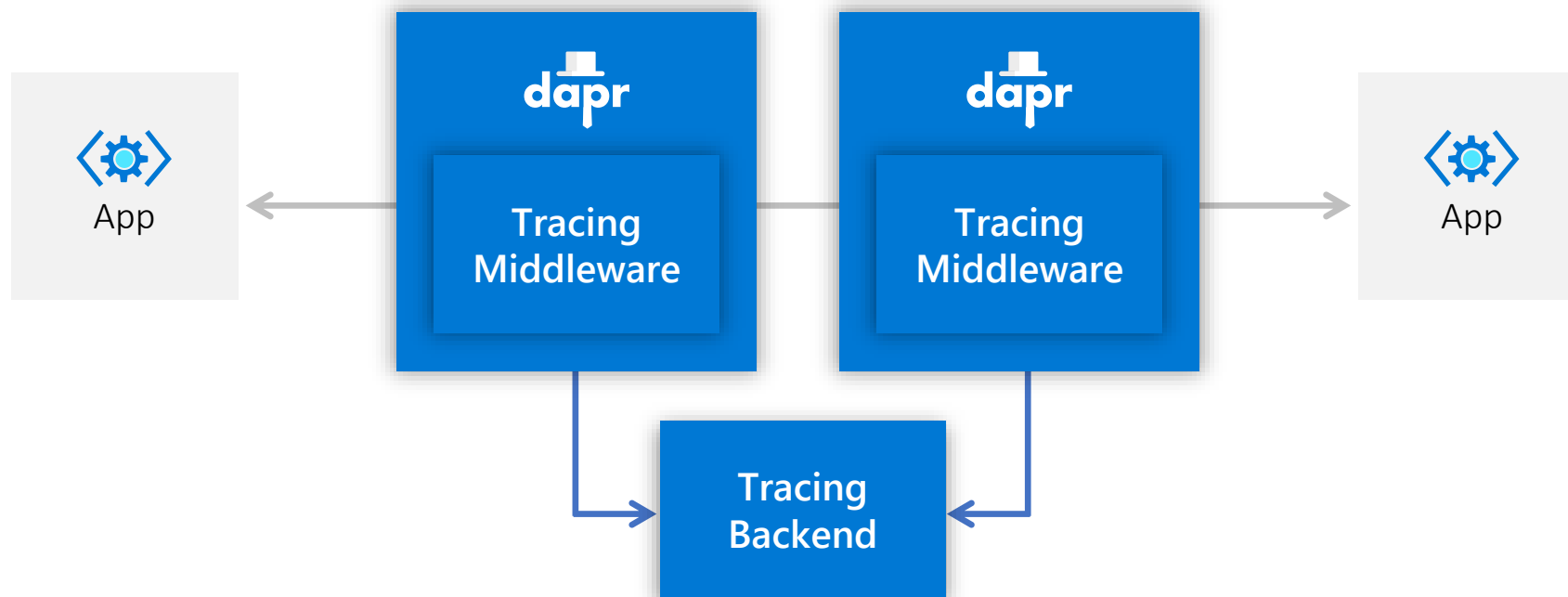
Dapr und Service Meshes arbeiten beide auf ein gemeinsames Ziel hin: **Komplexität Reduzieren**



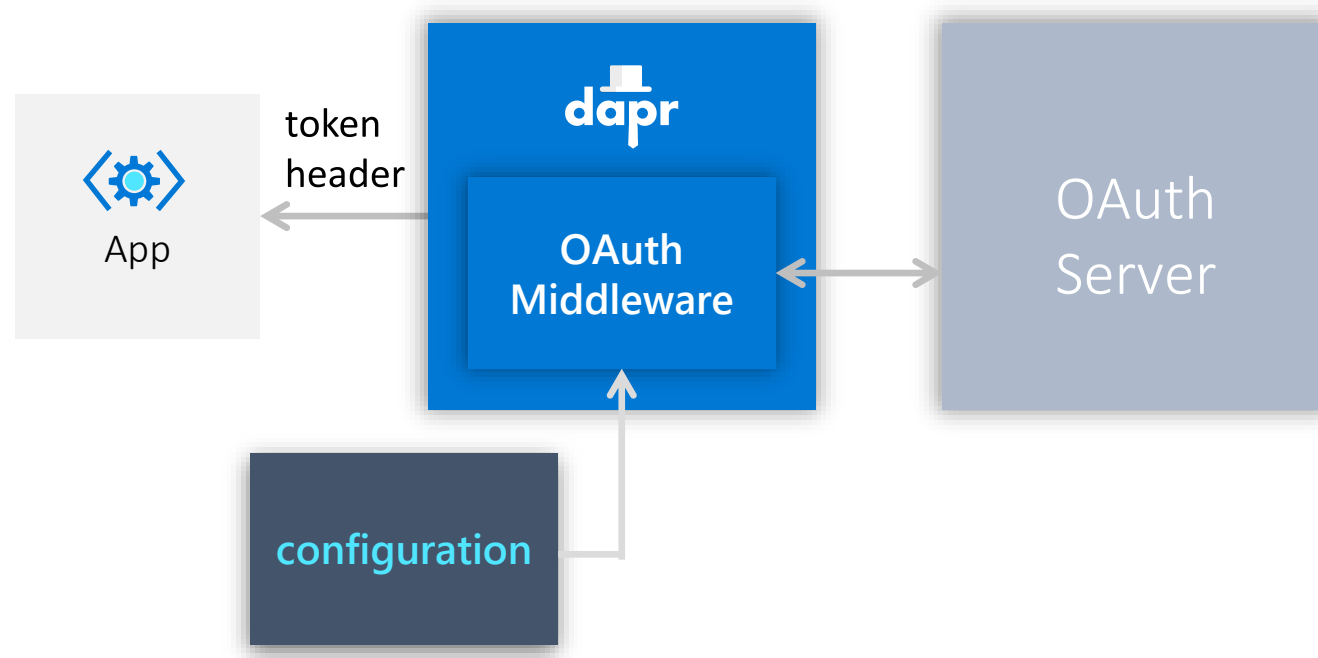
Dapr Erweiterungspunkt: Middleware



Middleware - Tracing



Middleware - OAuth



Wo kann man mehr über Dapr lernen?

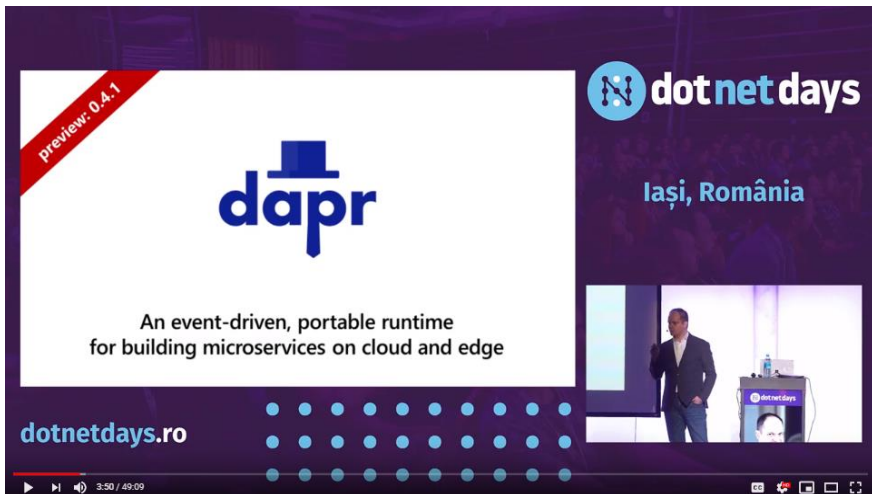
<https://myignite.techcommunity.microsoft.com/sessions/82059>



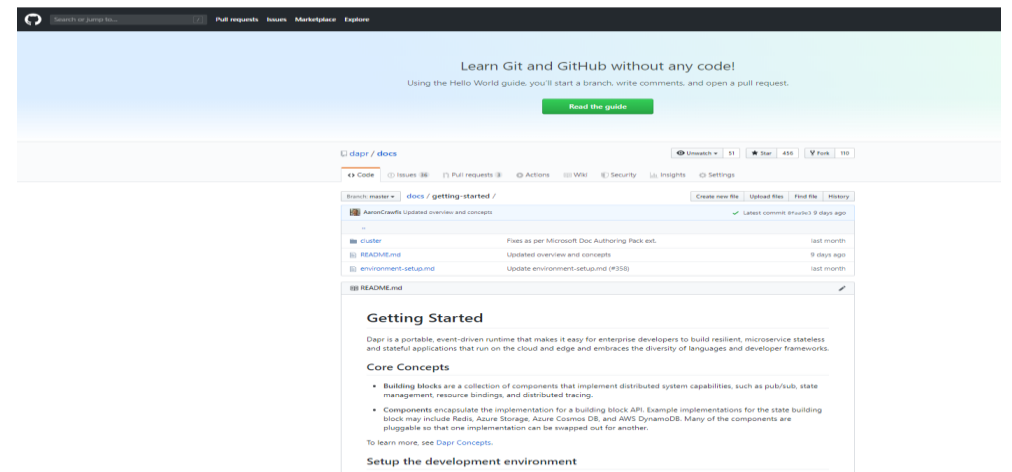
<https://www.youtube.com/watch?v=Cgql7nen-Ng>



<https://www.youtube.com/watch?v=a2OZ0VI4JTg>



<https://github.com/dapr/docs/tree/master/getting-started>



Engagieren Sie sich!



<https://github.com/dapr/dapr#community>





Viel Erfolg!