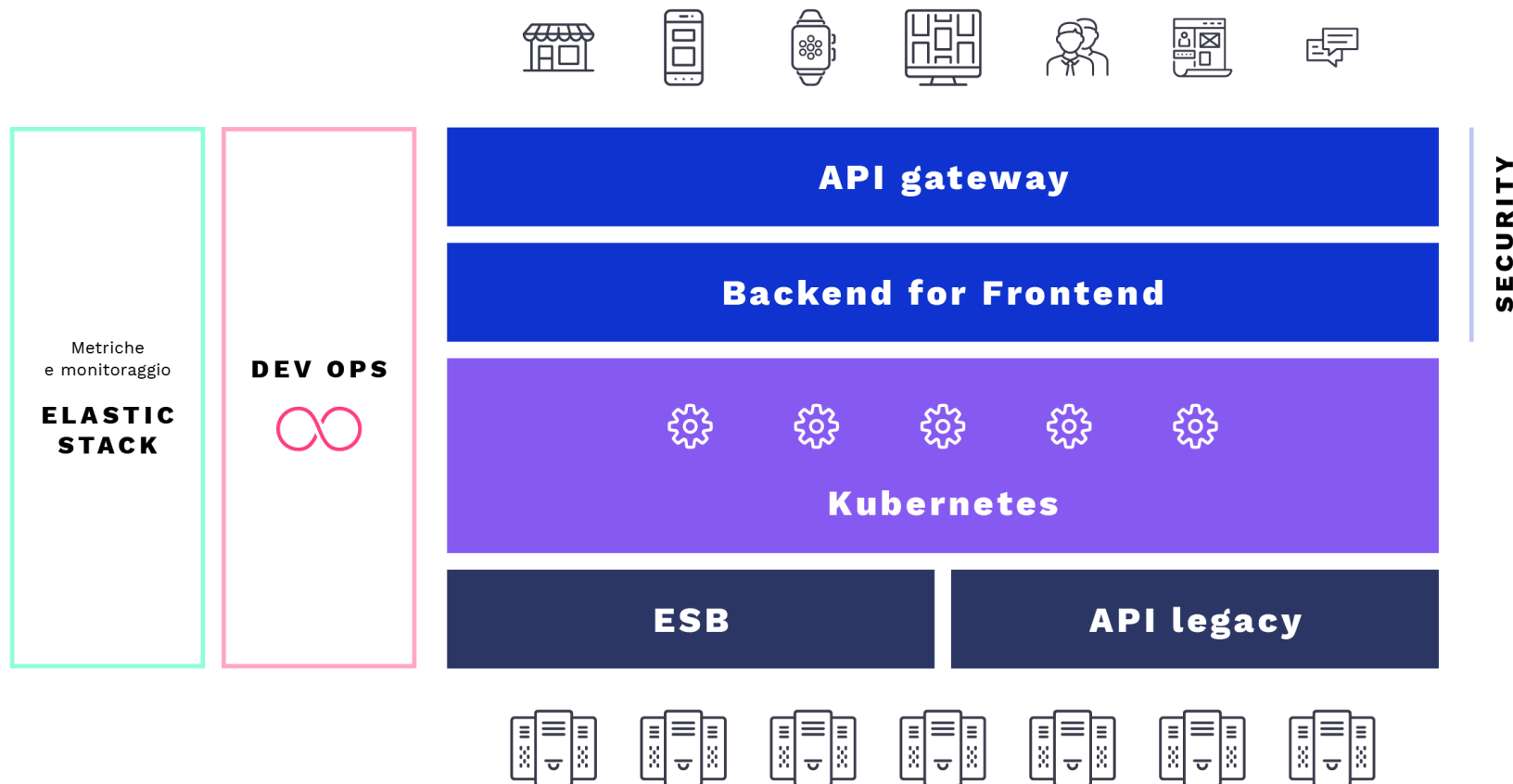




Kubernetes: il motore per l'orchestrazione e l'evoluzione dei servizi aziendali

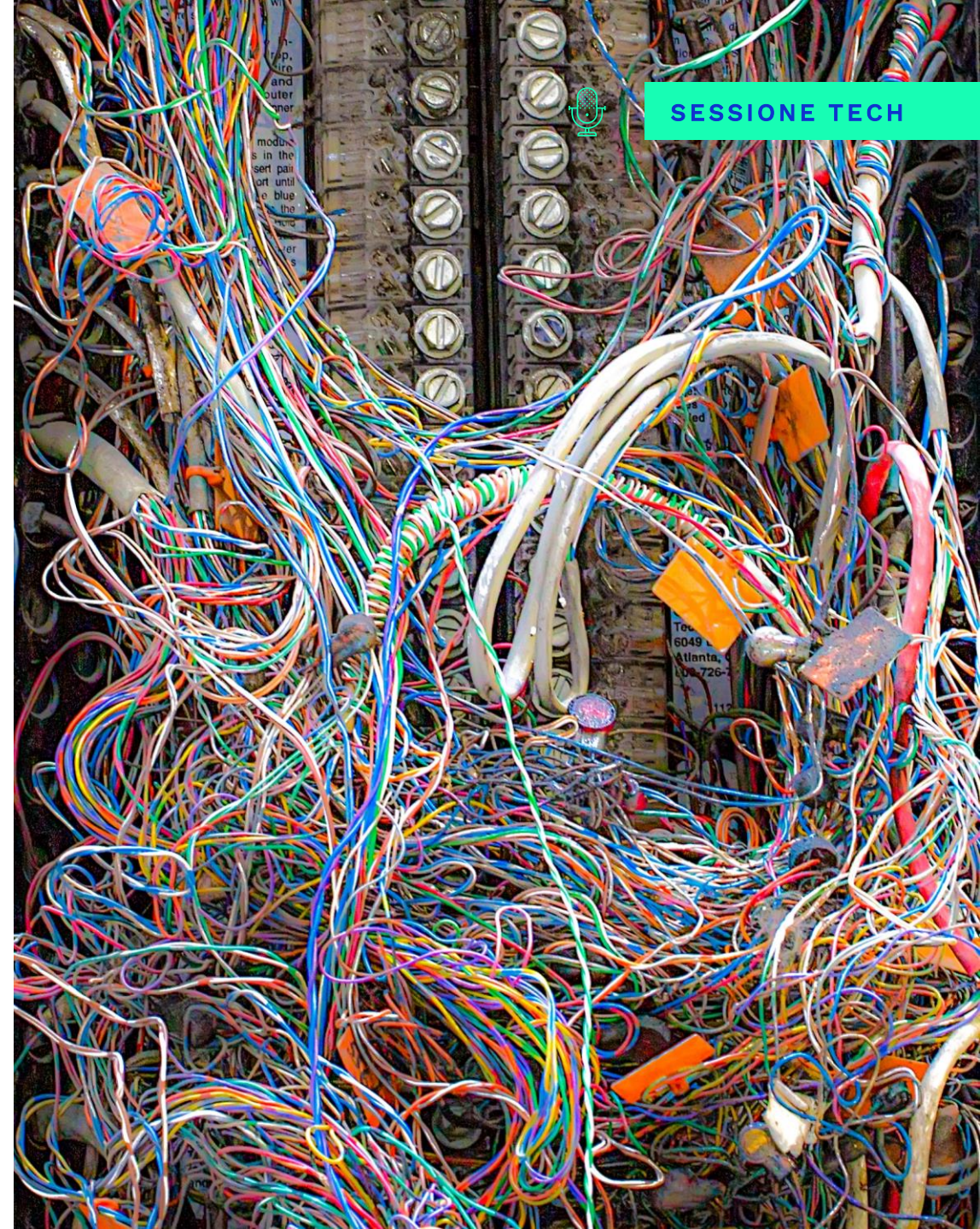




Headless
& API date

Gestire la complessità

- Microservizi & API
- DevOps
- Monitoring & Observability
- Cloud & Cloud Native





Container & Kubernetes

VM vs Container



APP

APP

BIN/LIBS

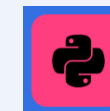
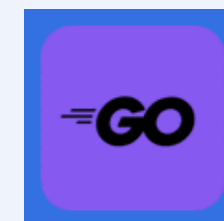
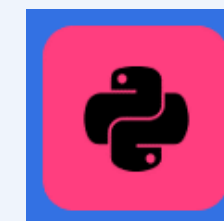
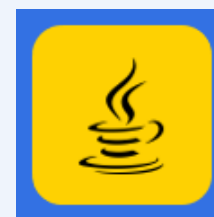
BIN/LIBS

OS

OS

SERVER

SERVER



CONTAINER RUNTIME

CONTAINER RUNTIME

OS

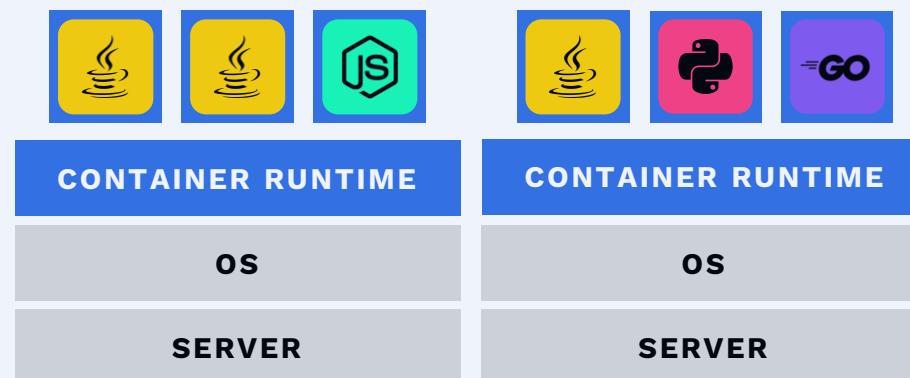
OS

SERVER

SERVER

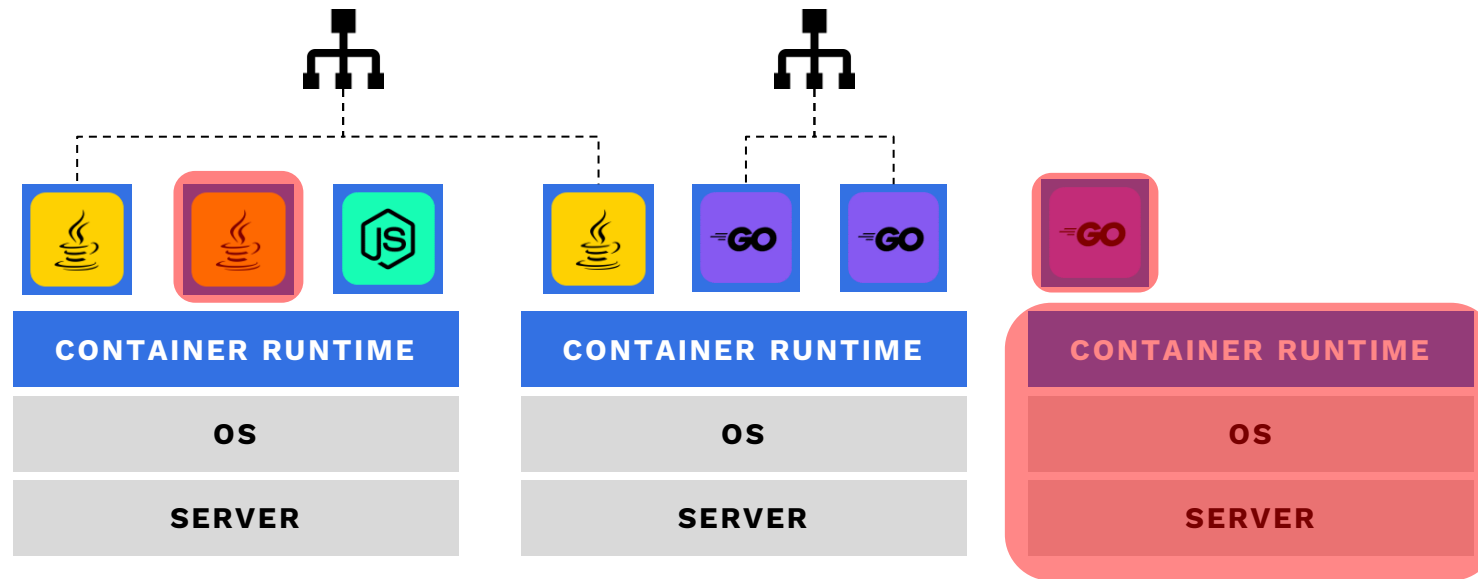
Container

- Più leggeri delle VM
- Isolati
- Immutabili
- Portabili





kubernetes





Headless
& API date



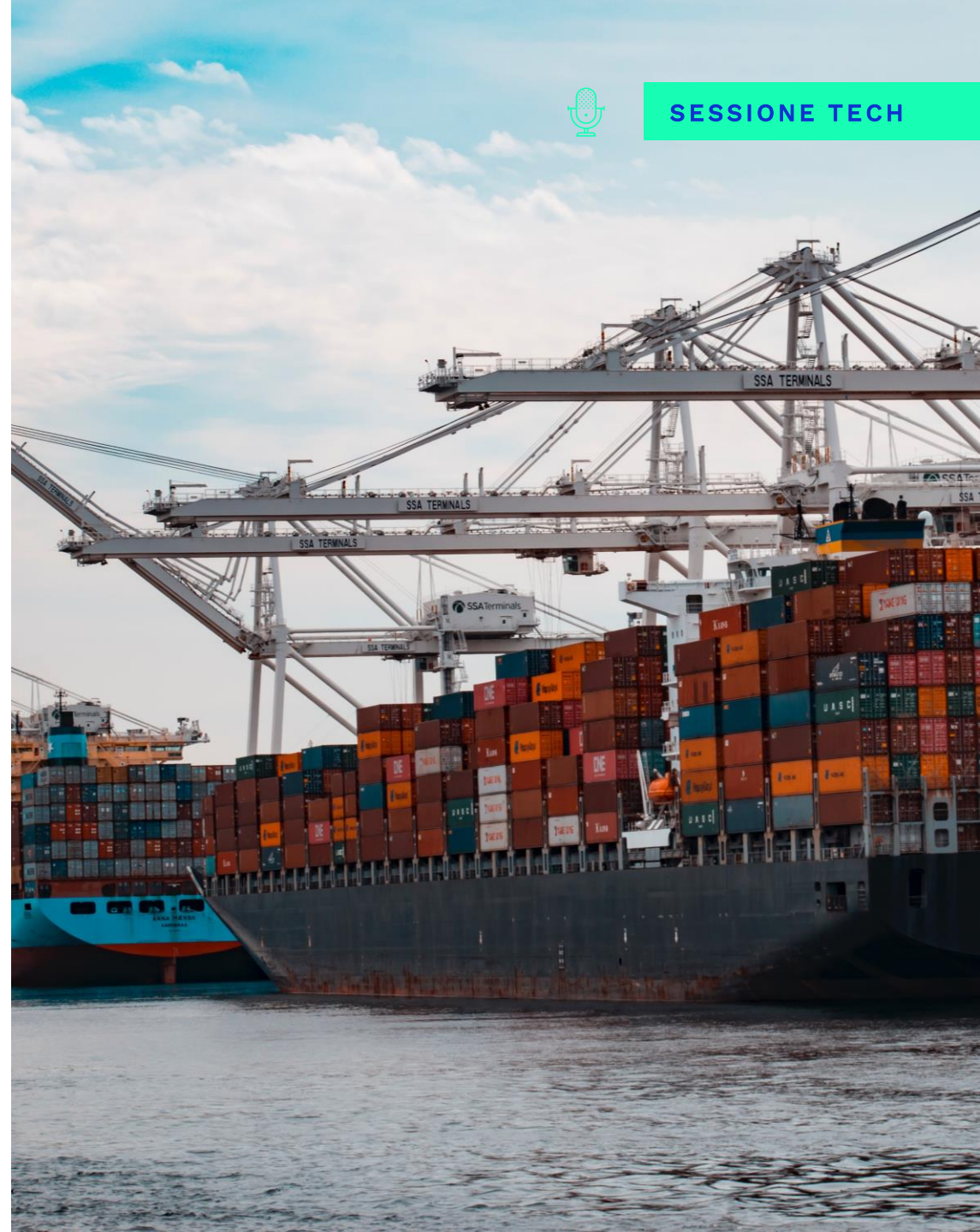
kubernetes

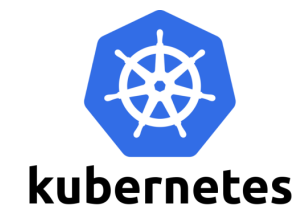
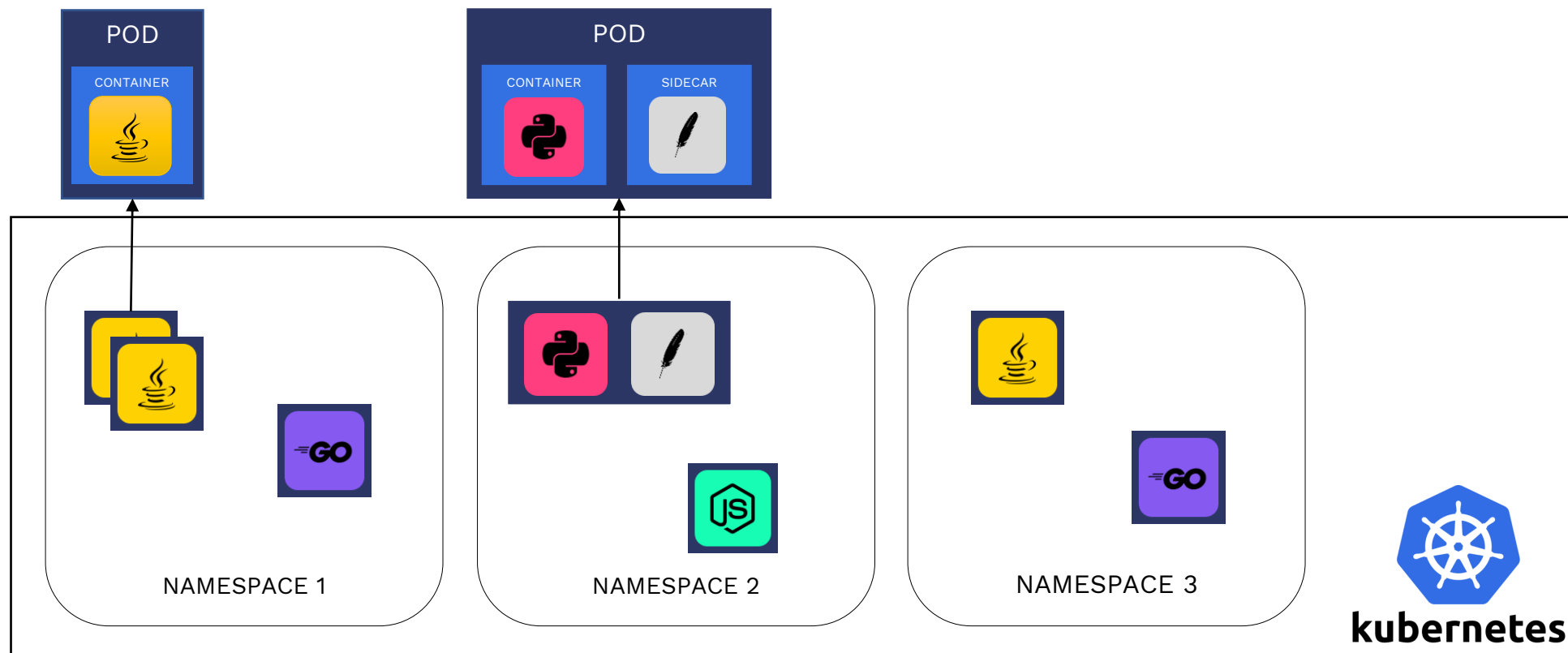
Kubernetes, also known as k8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

<https://kubernetes.io/>



SESSIONE TECH





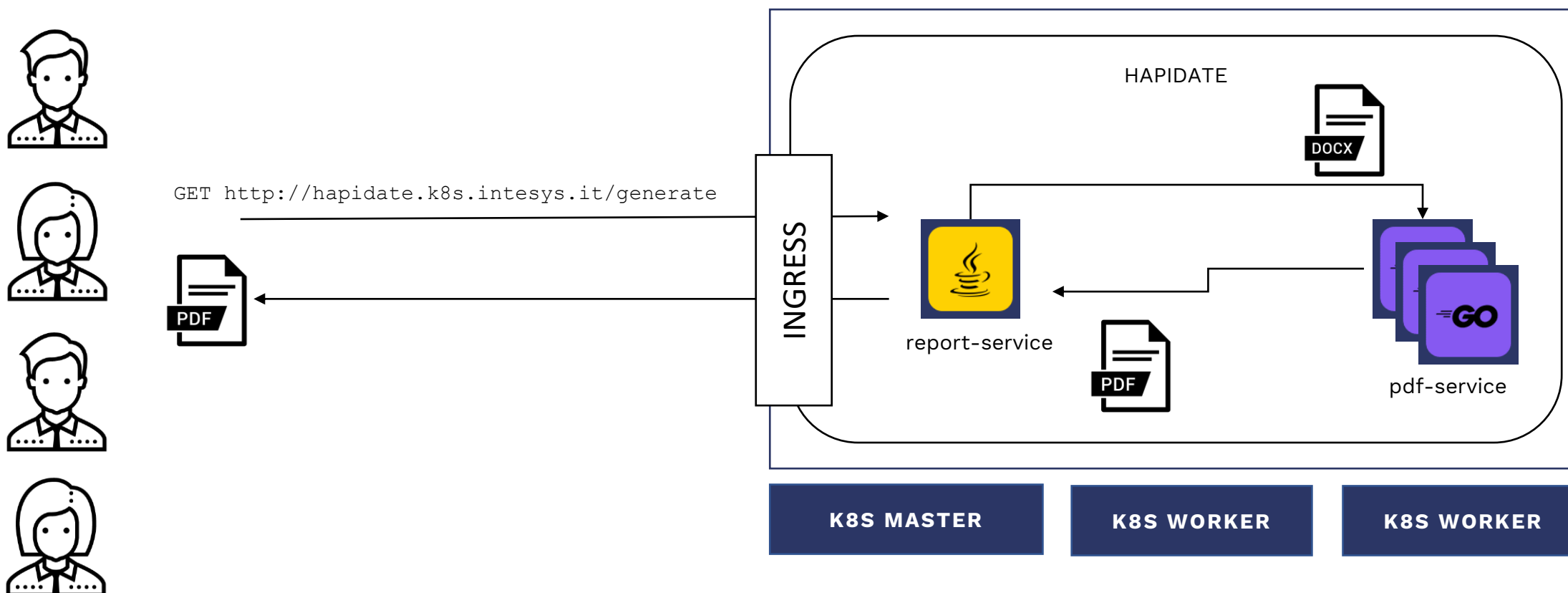
K8S MASTER	K8S WORKER	K8S WORKER	K8S WORKER
OS	OS	OS	OS
SERVER	SERVER	SERVER	SERVER



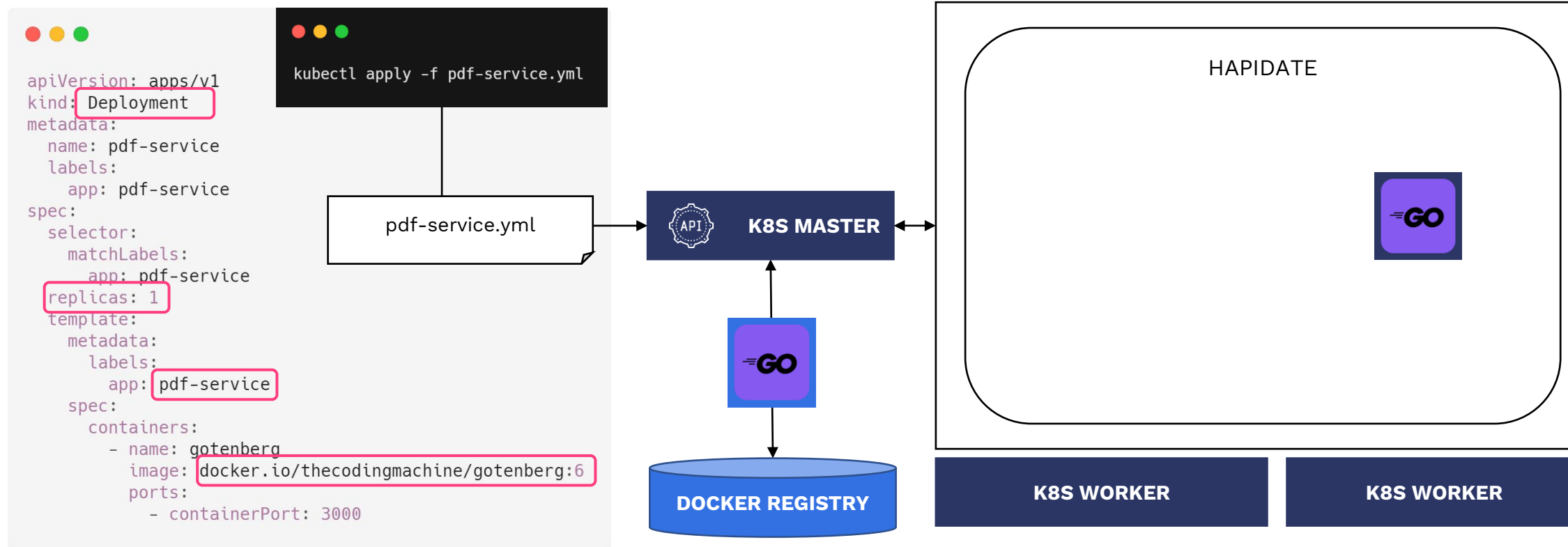
K8S in azione

1. DEPLOY
2. SCALING
3. AUTO SCALING
4. SERVICE DISCOVERY
5. LOAD BALANCING
6. GESTIONE RISORSE

Generazione Reportistica

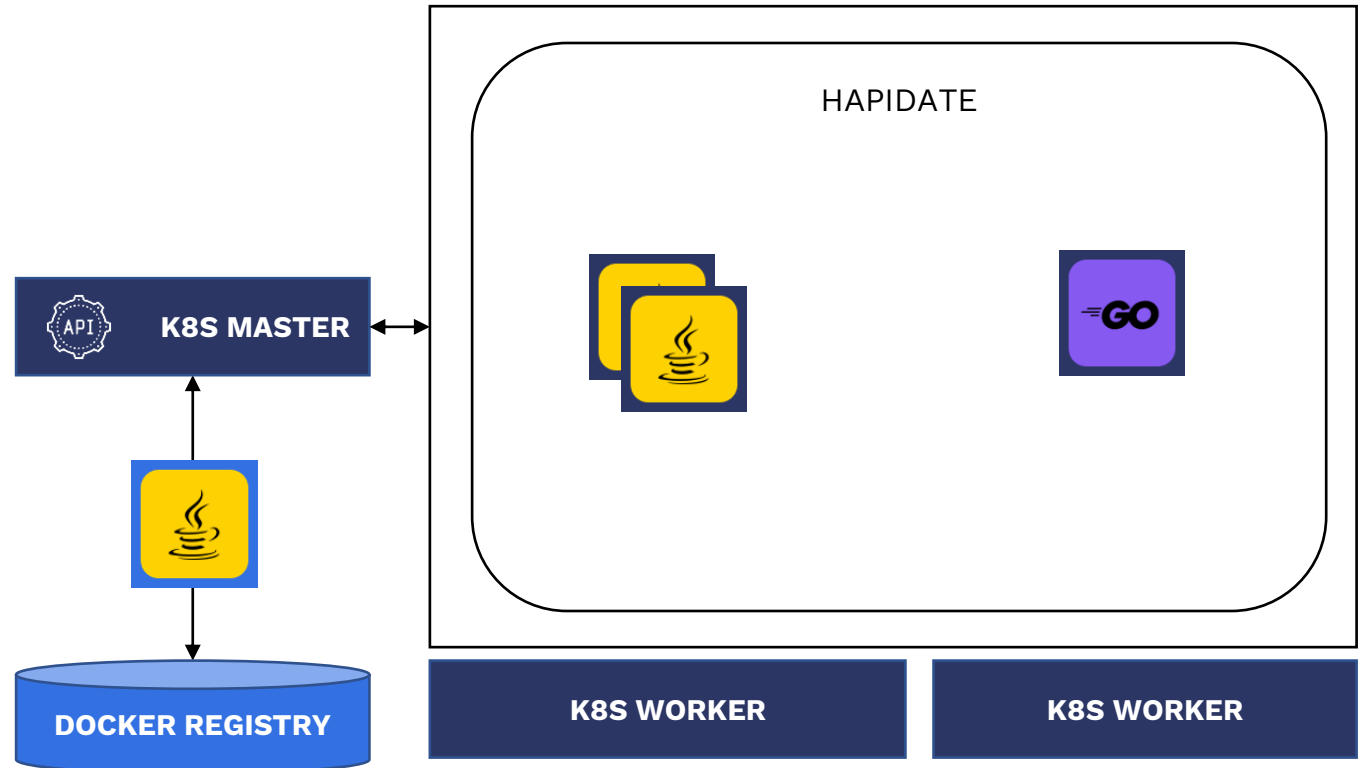


PDF Service - Deploy



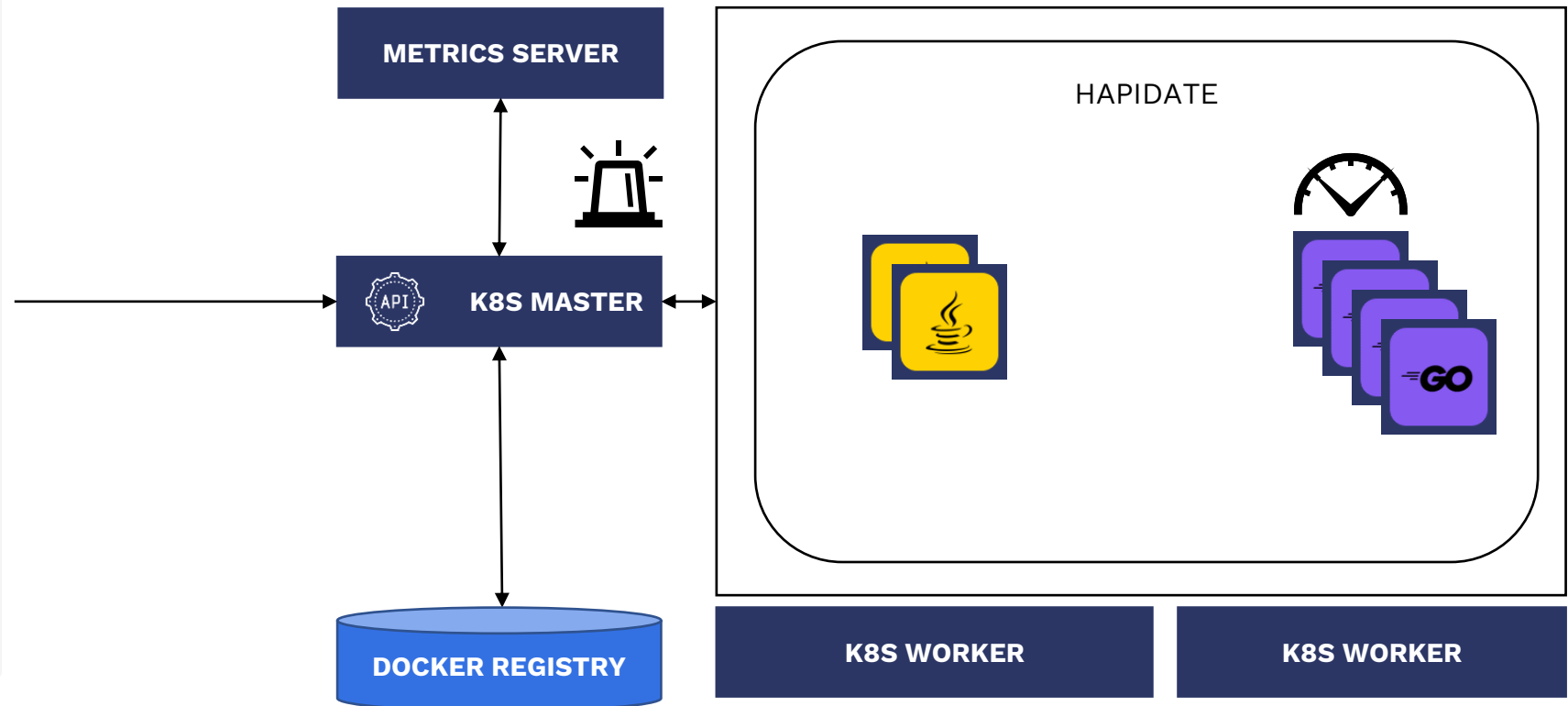
Report Service – Deploy & Scaling

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: report-service
  labels:
    app: report-service
spec:
  replicas: 2
  selector:
    matchLabels:
      app: report-service
  template:
    metadata:
      labels:
        app: report-service
    spec:
      containers:
        - name: report-service
          image: docker.io/intesys/report-service:1.0.0
          ports:
            - containerPort: 8080
```

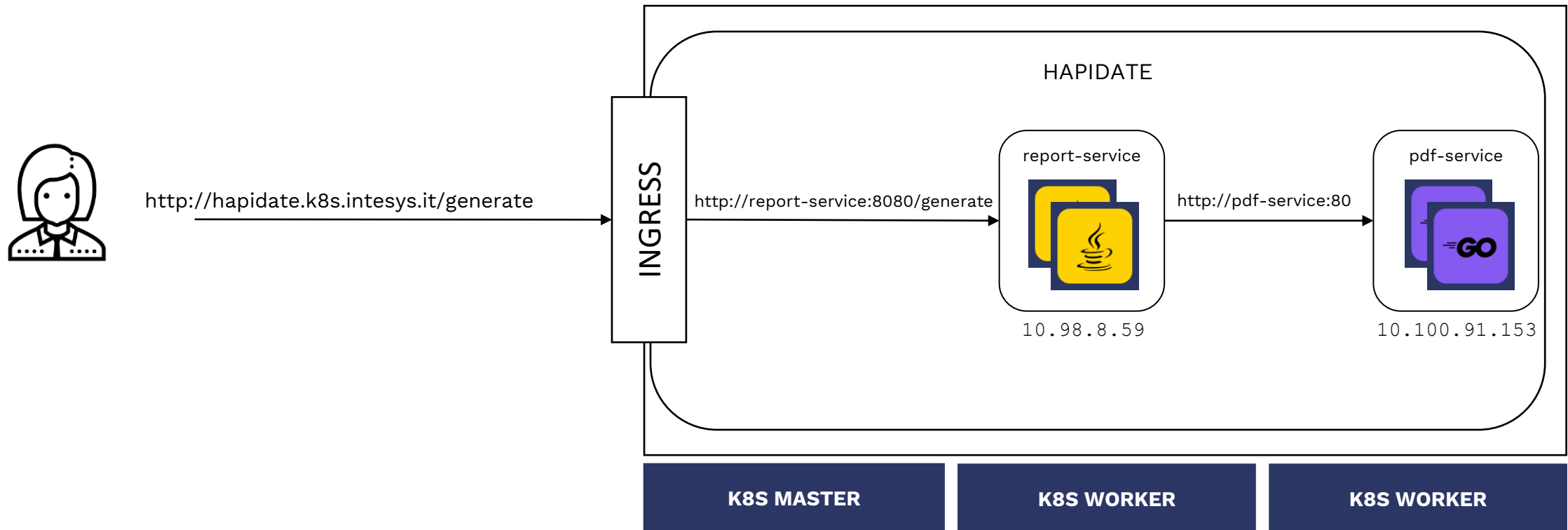


Auto Scaling

```
---
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: pdf-service
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: pdf-service
  minReplicas: 1
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 75
```



Service Discovery & Load Balancing



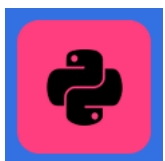
Gestione risorse



CPU: 1 – 2
RAM: 1GB – 2GB



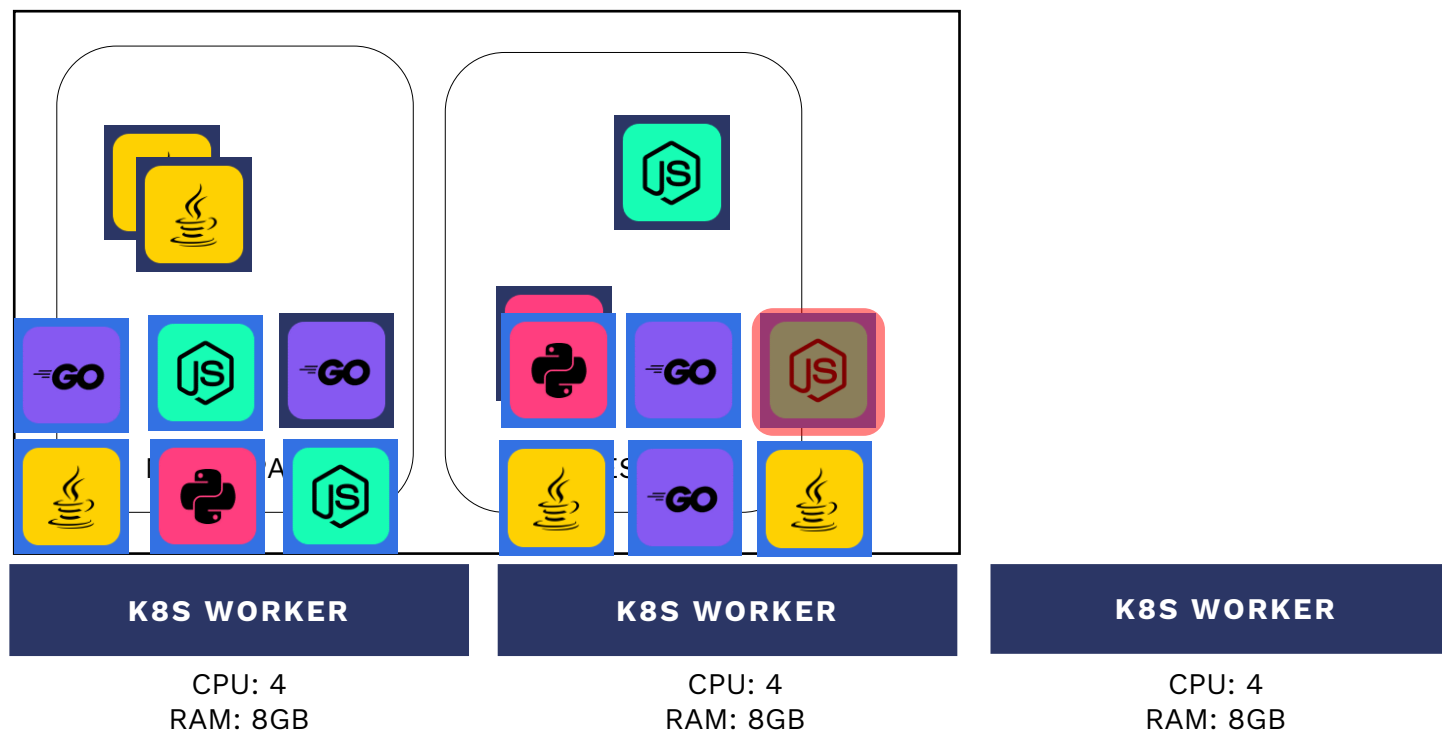
CPU: 0.5 – 2.5
RAM: 250MB – 500 MB



CPU: 1 – 2.5
RAM: 250MB – 700 MB



CPU: 0.5 – 2.5
RAM: 250MB – 350 MB



ecostanzi ~/hapi2020-kubernetes master

ecostanzi ~ # mostro le risorse all'interno del cluster |

x ecostanzi ~

x ecostanzi ~



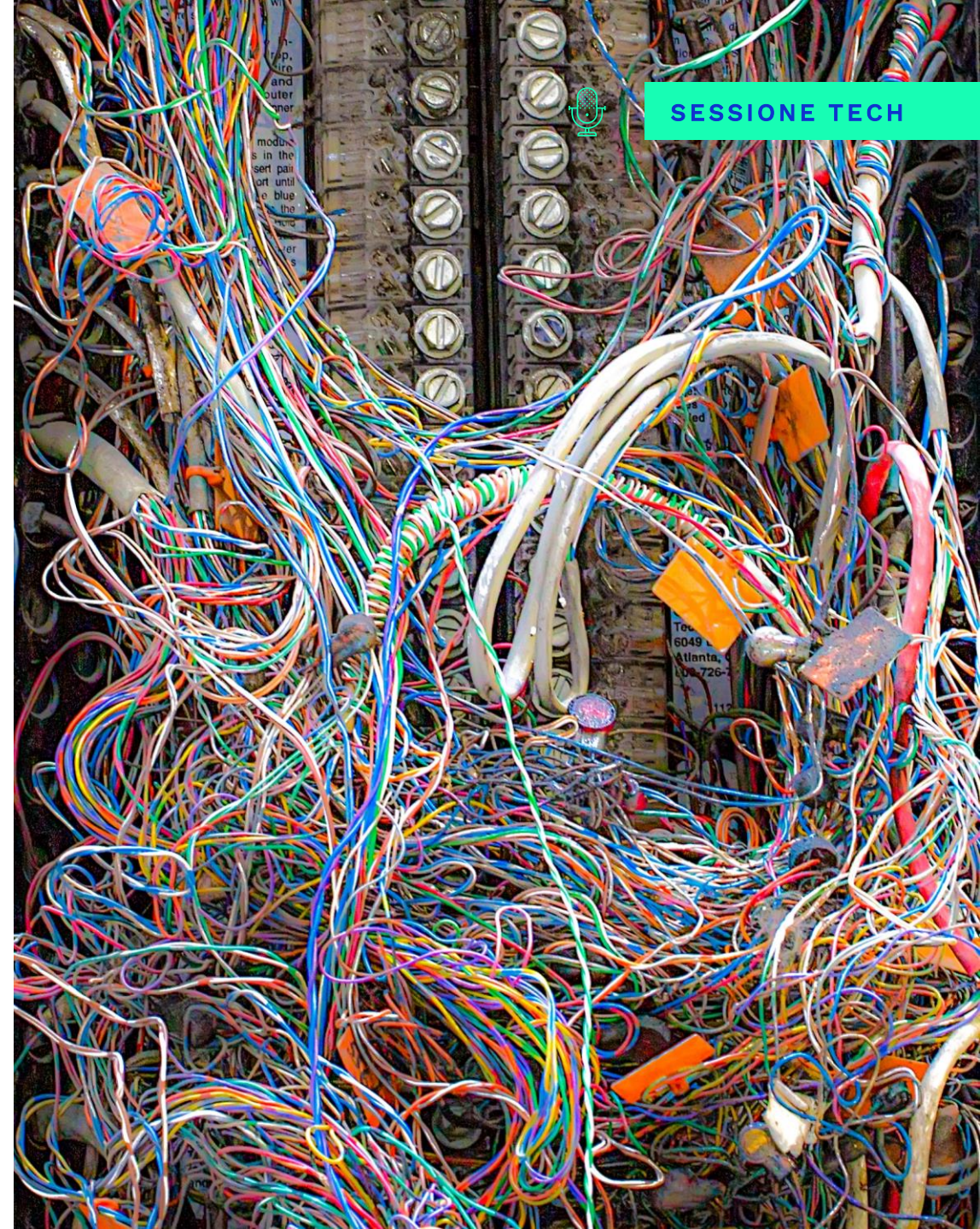
Dove eravamo rimasti?



Headless
& API date

Gestire la complessità

- Microservizi & API
- DevOps
- Monitoring & Observability
- Cloud & Cloud Native

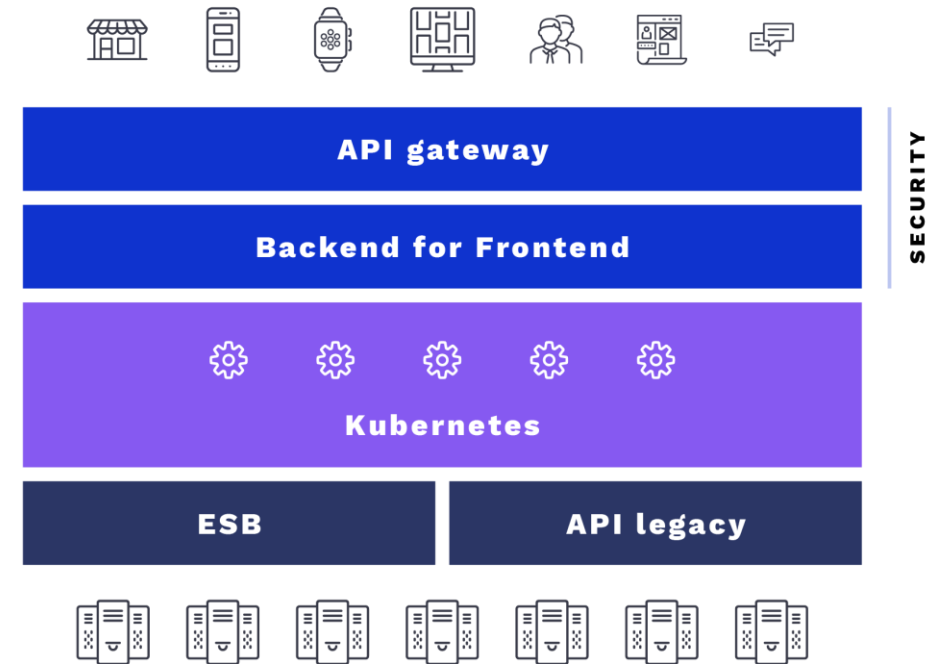


SESSIONE TECH



(Micro)servizi

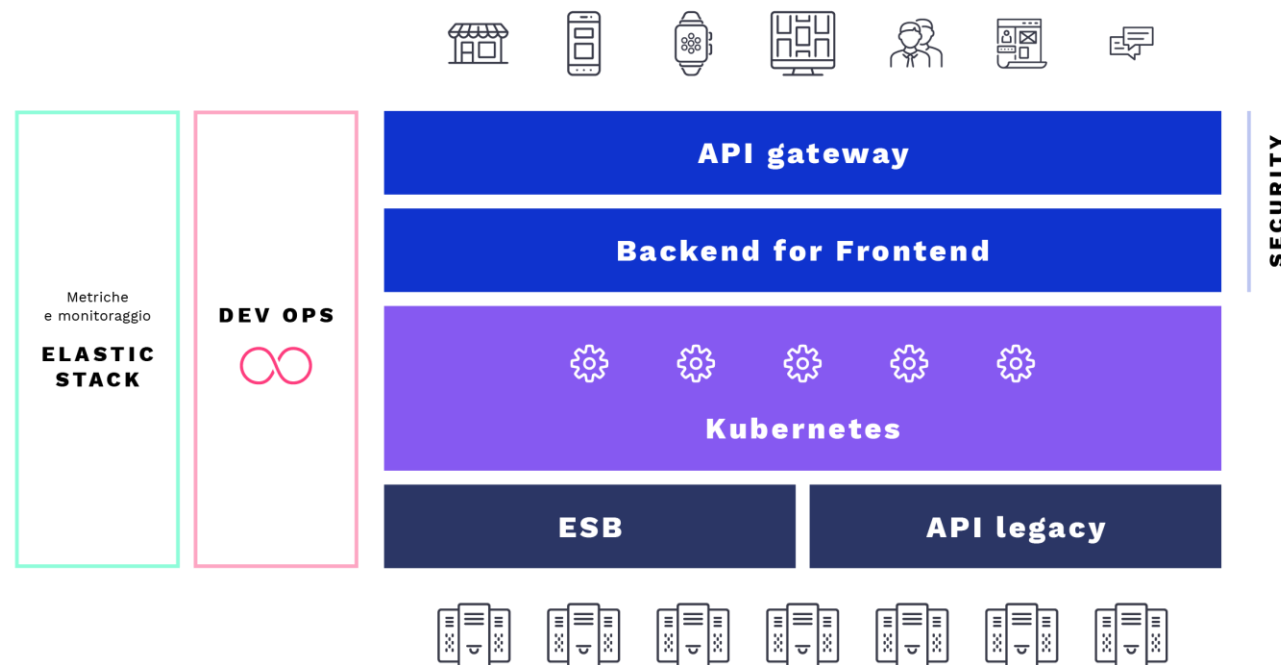
- Software disaccoppiato dall'hardware
- Automazione operazioni
 - Autoscaling
 - Service Discovery
 - Rolling Update / Canary Release





DevOps

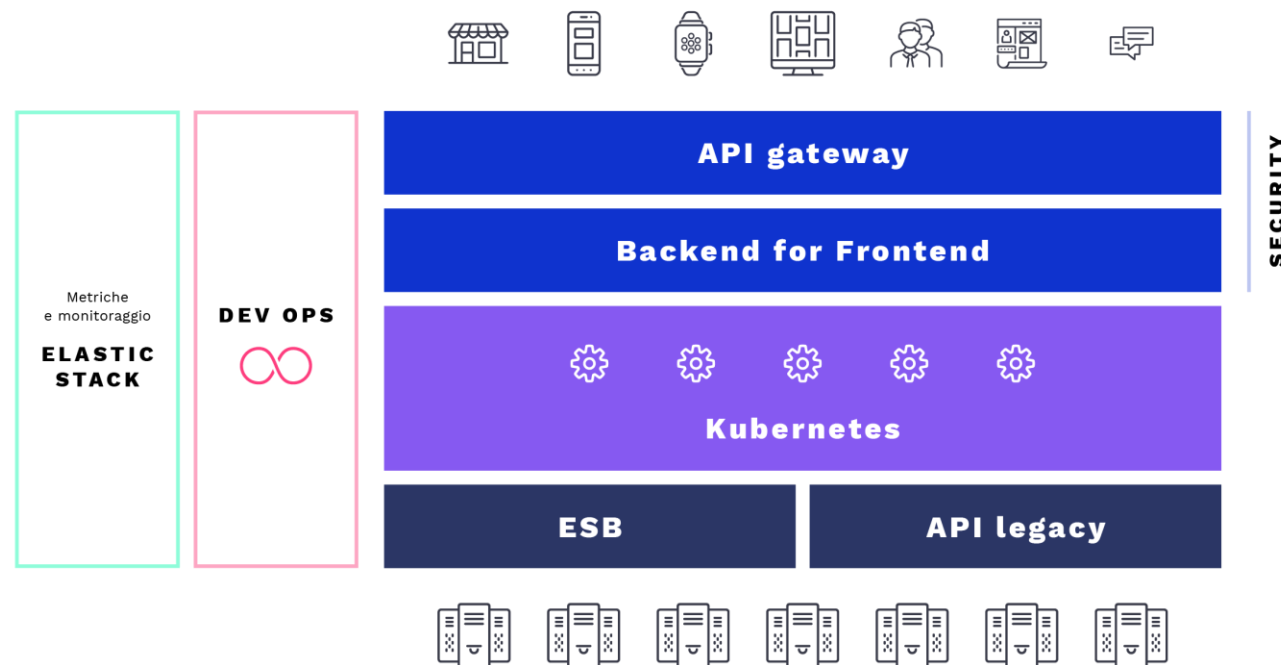
- Build di container
- Infrastructure as Code
- Continuous Delivery / Deployment
- Monitoring & Observability





Cloud & Cloud Native

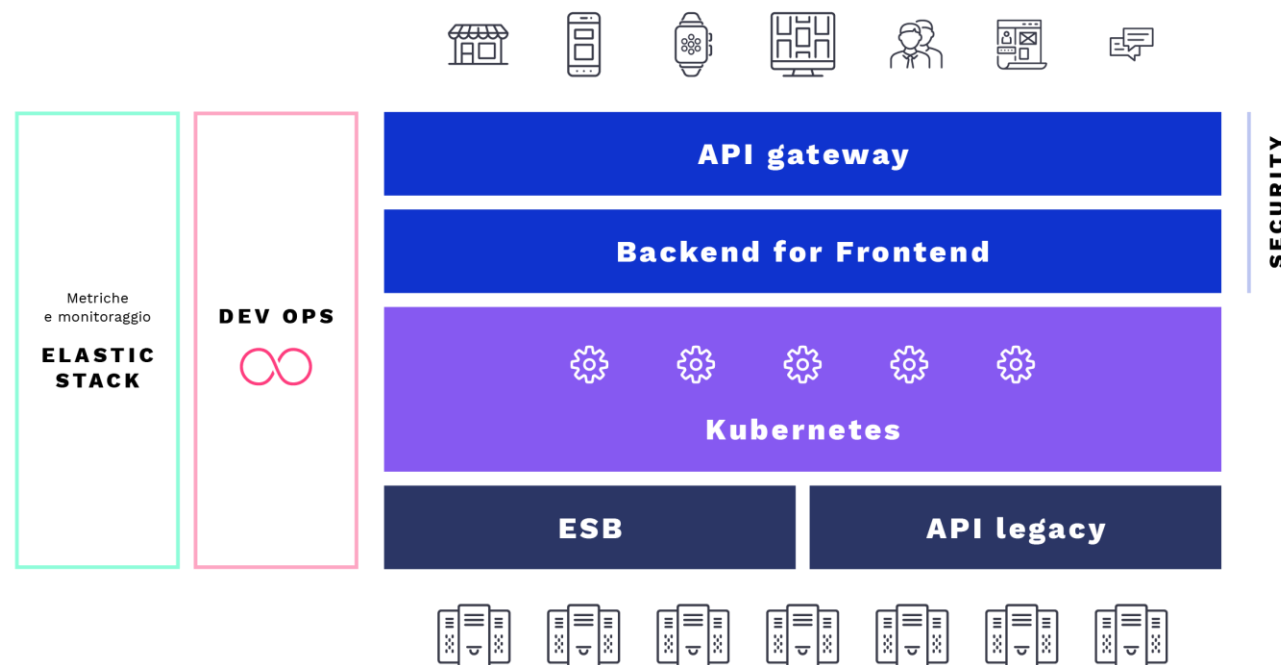
- Scalabile
- Fault Tolerant
- Architettura evolutive
- Iterazioni Frequenti
- Multi Cloud / Hybrid Cloud





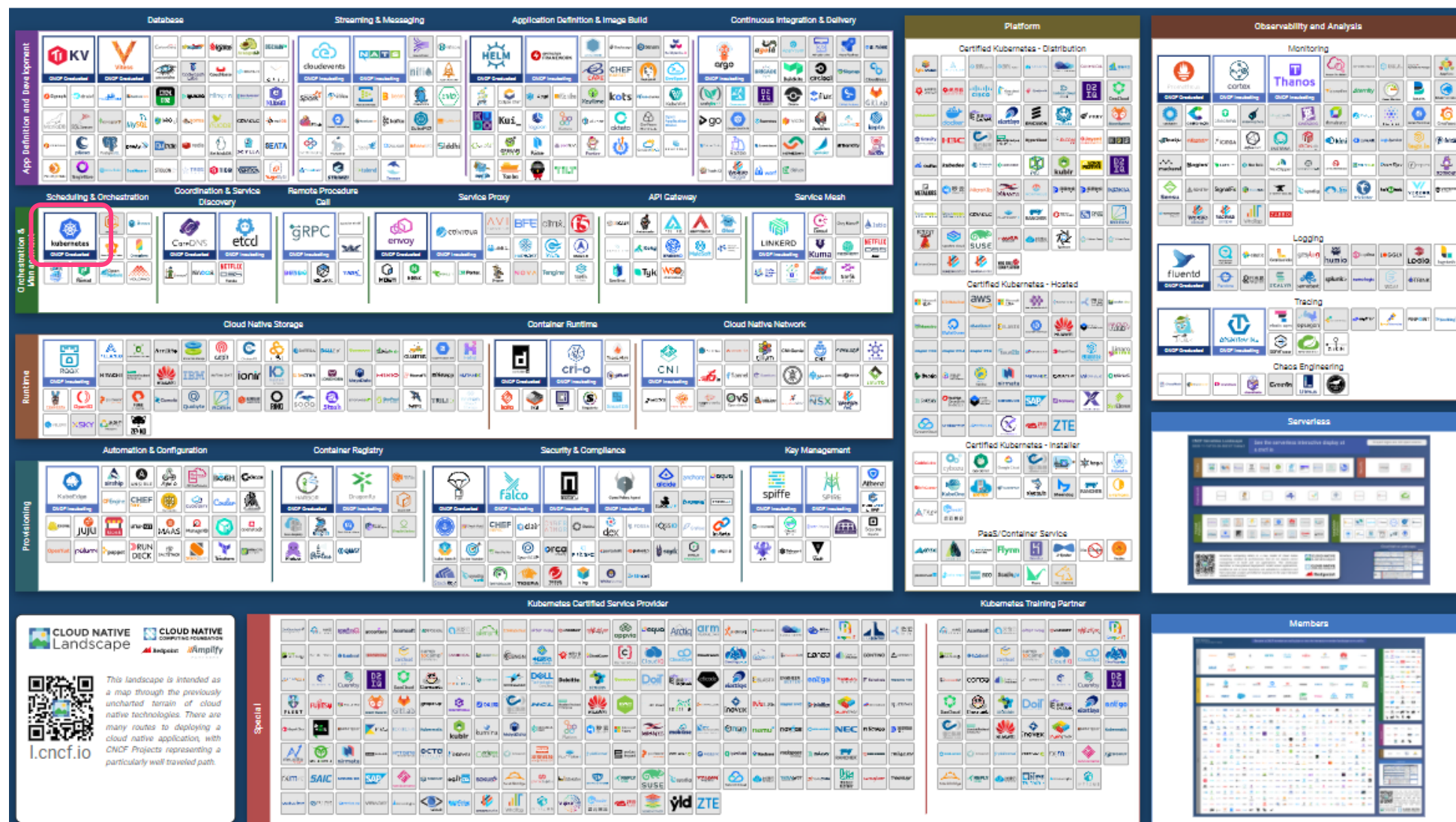
Governance

- Controllo sull'evoluzione dei sistemi
- Utilizzo ottimizzato delle risorse
- Linguaggio comune tra dev e ops





Tutto così semplice?



Grazie

Enrico Costanzi

INTESYS SENIOR SOFTWARE DEVELOPER