
High Performance Cloud-Native Microservices With Distributed Caching

Mesut Celik, *Hazelcast*



@mesutcelik



Mesut Celik

- Tech Lead @Hazelcast
- Java Developer
- lived in *Turkey, Belgium, US*
- Twitter: @mesutcelik



@mesutcelik



Agenda

- Hazelcast Introduction
- Caching Patterns
- Kubernetes Deployments
- Day 2 Operations



@mesutcelik



Hazelcast is an Open Source

Company Hazelcast Products

- Hazelcast IMDG
- Hazelcast Jet
- Hazelcast Cloud
- Management Center
- Apache 2 License
- Remote friendly
- raised \$21M series C
- Always hiring!!!!



@mesutcelik



Why Distributed Cache?

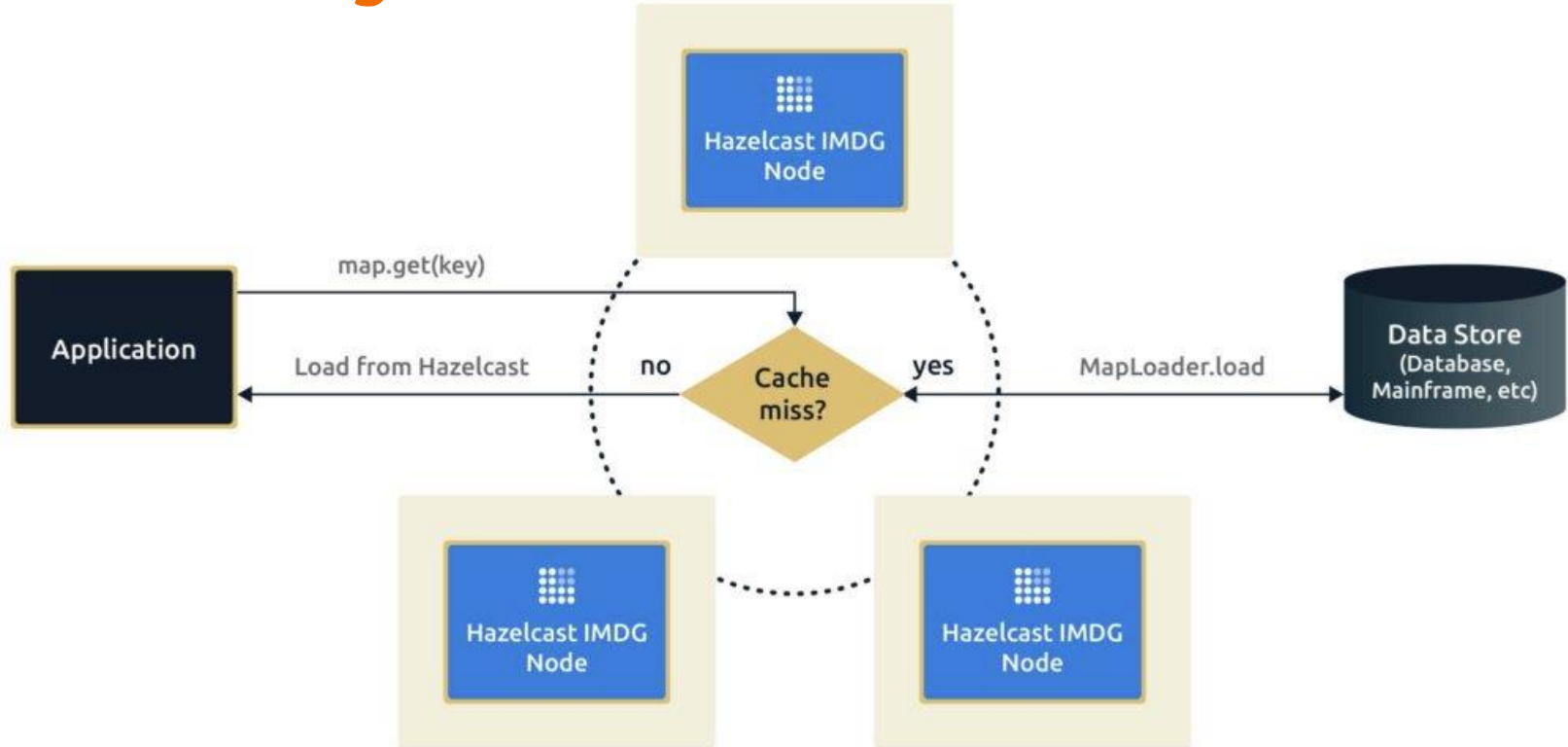
- **Cache**, Memory is faster than storage
- **Distributed**, Natural fit for Cloud-Native Deployments



@mesutcelik

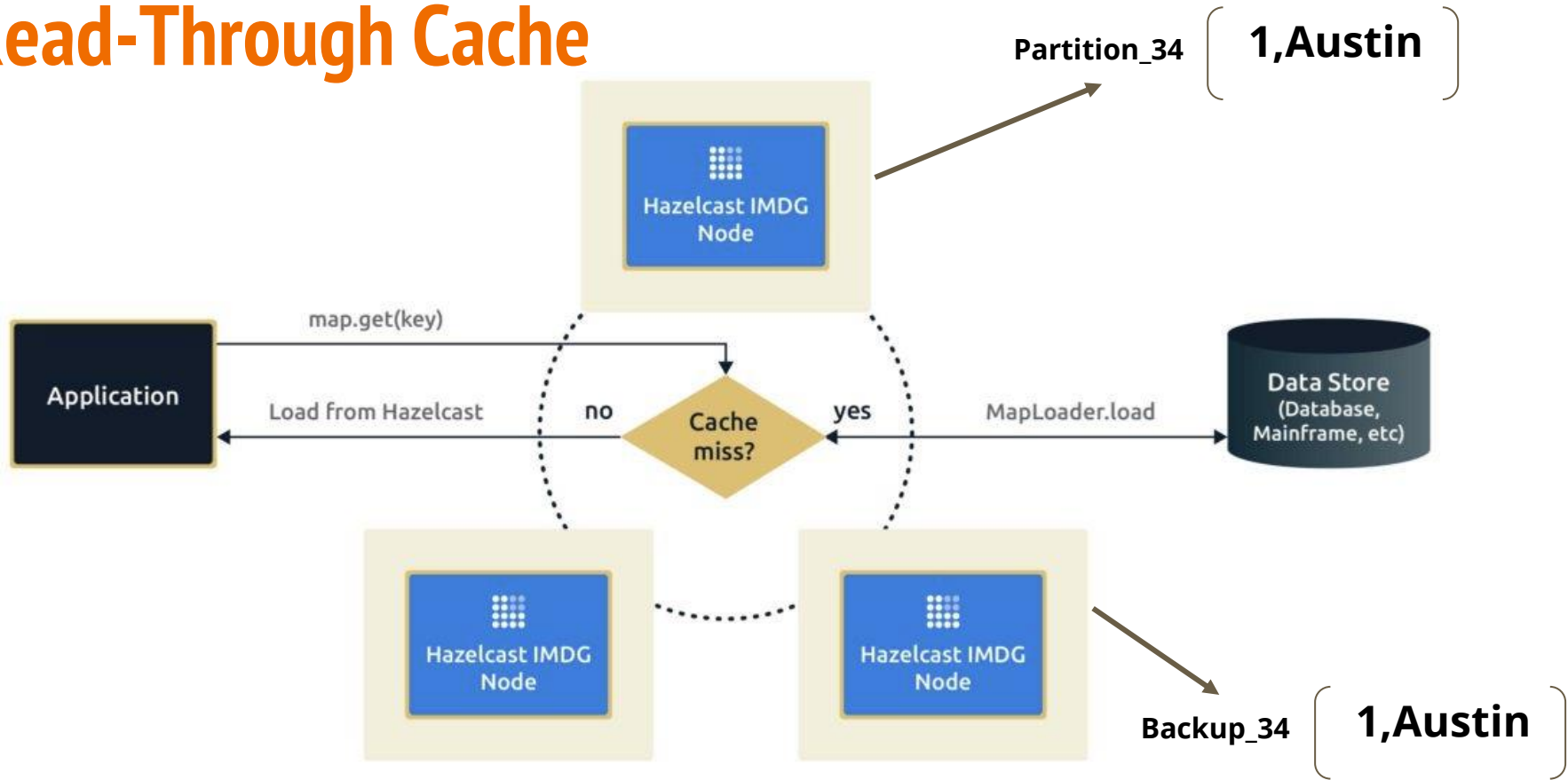


Read-Through Cache



@mesutcelik

Read-Through Cache



@mesutcelik

Hazelcast and Kubernetes Caching Patterns



@mesutcelik



Distributed Caching Patterns in Kubernetes

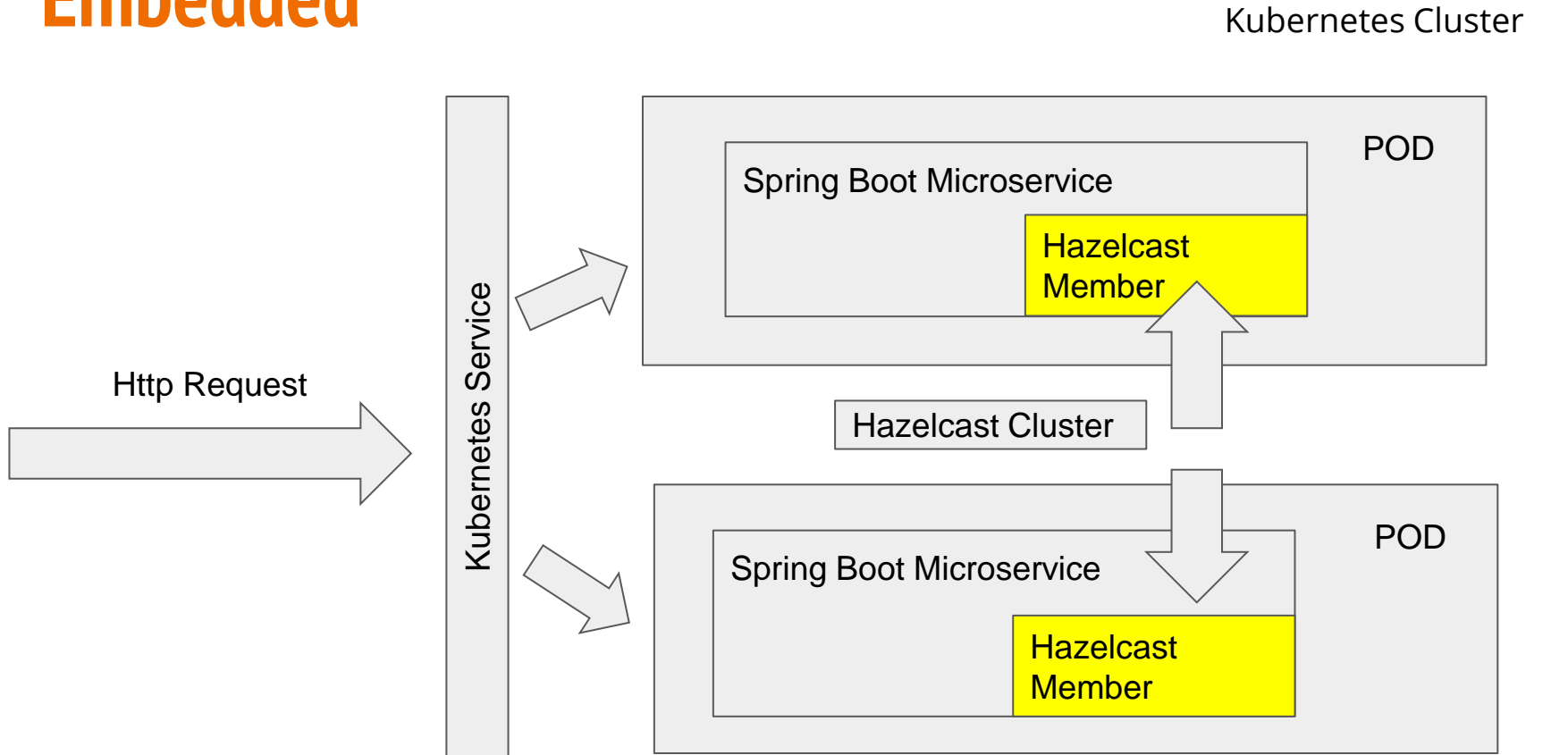
- Embedded
- Sidecar
- Proxy Cache
- Client-Server



@mesutcelik



Embedded



@mesutcelik



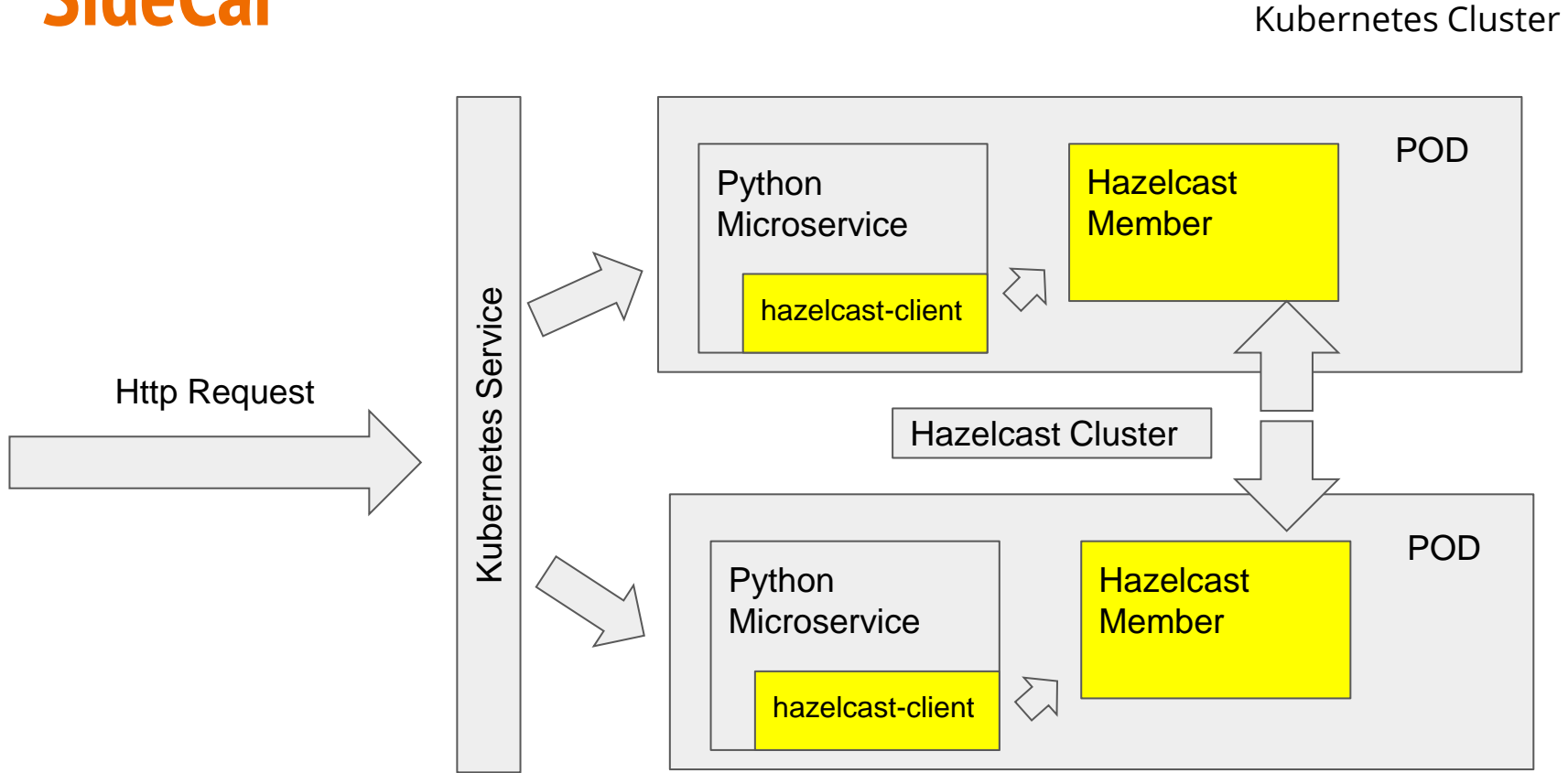
Embedded

- Simple
- Java only
- Data co-located with Microservice JVM



@mesutcelik

SideCar



@mesutcelik



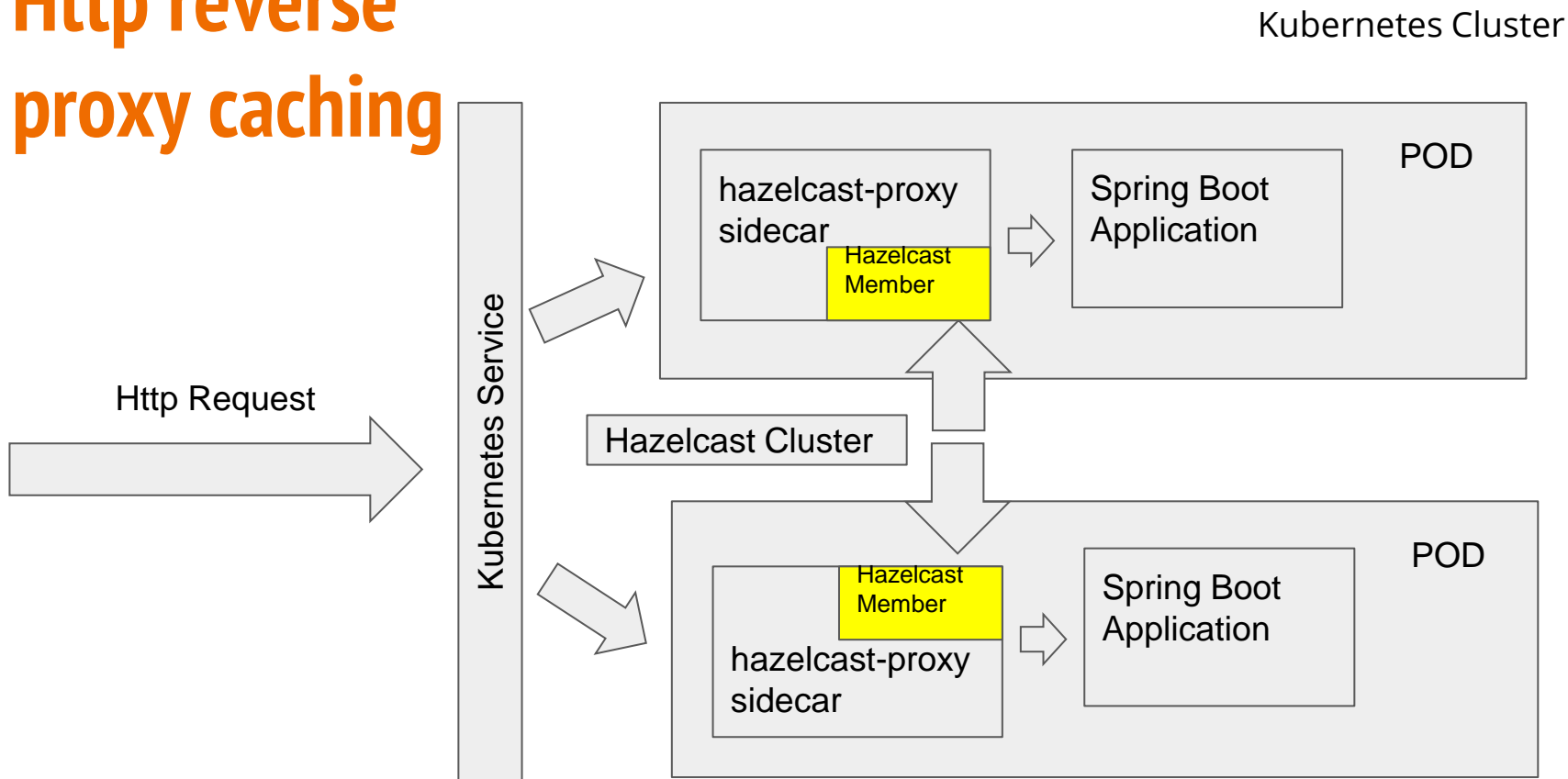
Sidecar

- Simple
- Multiple Programming Languages
 - **Java, Node.js, .NET, Python, Golang**
- Data co-located with Microservice POD



@mesutcelik

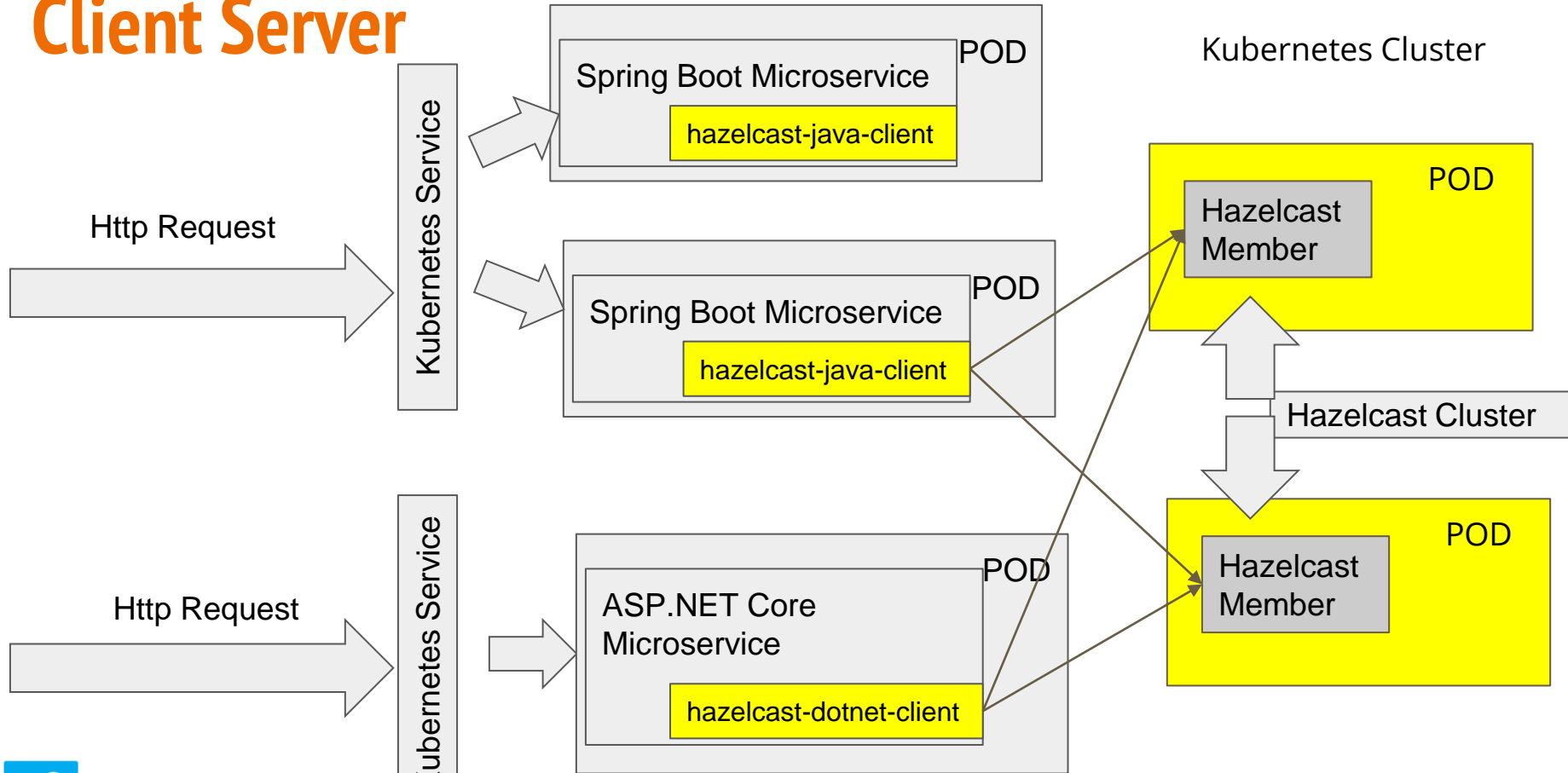
Http reverse proxy caching



Http reverse proxy caching

- Injecting Cache into every microservice
- No coupling with any Cache API
- Enabled w/o code change

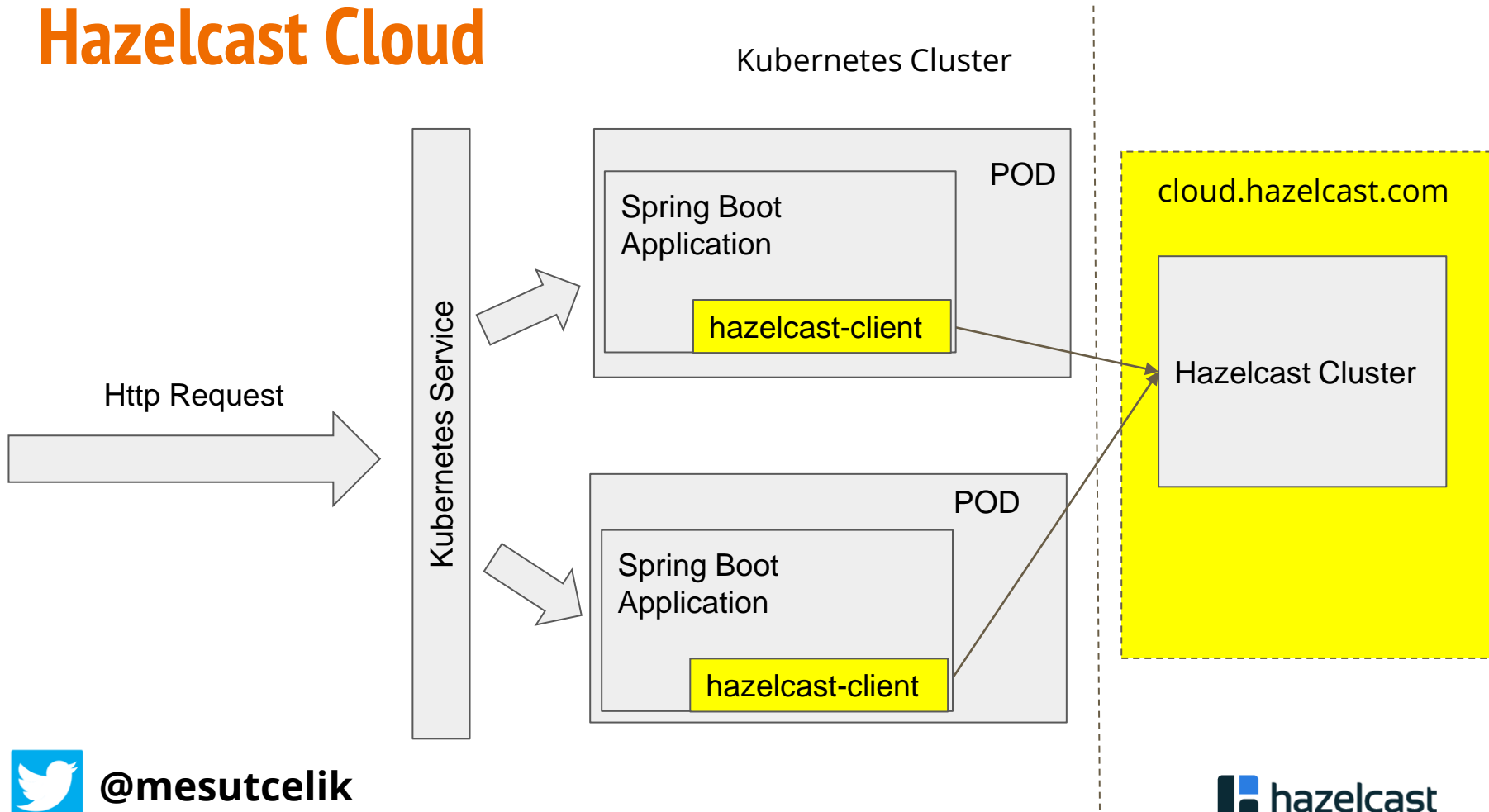
Client Server



@mesutcelik



Hazelcast Cloud



@mesutcelik



Client Server

- CaaS - Cache as a Service
- Multiple Programming Languages
 - **Java, nodejs, .NET, Python, Golang**
- Separate Hazelcast Cluster Maintenance



@mesutcelik



Kubernetes Deployment Options For Hazelcast



@mesutcelik



Hazelcast Helm Chart

```
$ helm install stable/hazelcast
```



@mesutcelik



Hazelcast Kubernetes Operator

- OperatorHub.io
- Openshift 4 Operator Catalog
- IBM Cloud Private



@mesutcelik



Hazelcast Cloud

- Managed Service
- Pay as you go model
- <https://cloud.hazelcast.com>



@mesutcelik



Deployment Best Practices



@mesutcelik



Predictable Resource Allocation

- Calculate CPU/Memory Requirements
- Set Resource Request and Limits as the same value

```
spec.containers[].resources.limits.cpu
```

```
spec.containers[].resources.limits.memory
```

```
spec.containers[].resources.requests.cpu
```

```
spec.containers[].resources.requests.memory
```



@mesutcelik



Pod Anti-Affinity

- Hazelcast per K8S Node
- Large Kubernetes Clusters
- Safer in Node crashes

```
affinity:  
  podAntiAffinity:  
    requiredDuringSchedulingIgnoredDuringExecution:  
      - labelSelector:  
        matchExpressions:  
          - key: app.kubernetes.io/name  
            operator: In  
            values:  
              - hazelcast  
          - key: role  
            operator: In  
            values:  
              - hazelcast  
        topologyKey: kubernetes.io/hostname
```

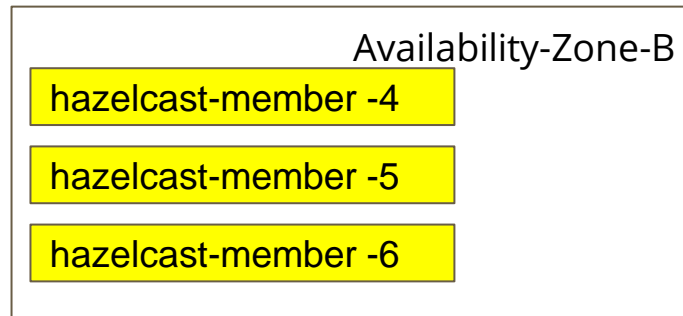
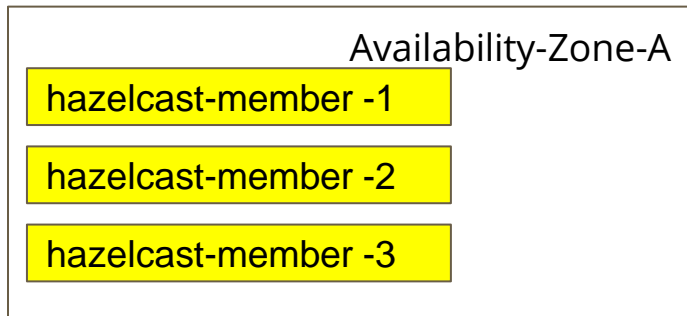


@mesutcelik



Multi-zone Deployment with ZONE_AWARE

Multi-zone Hazelcast Cluster



* Multi-zone if your architecture prefers availability over performance



@mesutcelik



Day 2 Operations



@mesutcelik



Day 2 Operations

- Manual Scaling vs. AutoScaling
- Rolling Update
- Monitoring



@mesutcelik

Thank you!

- <https://twitter.com/mesutcelik>
- <https://github.com/mesutcelik>
- <https://www.linkedin.com/in/mesutcelik/>



@mesutcelik

