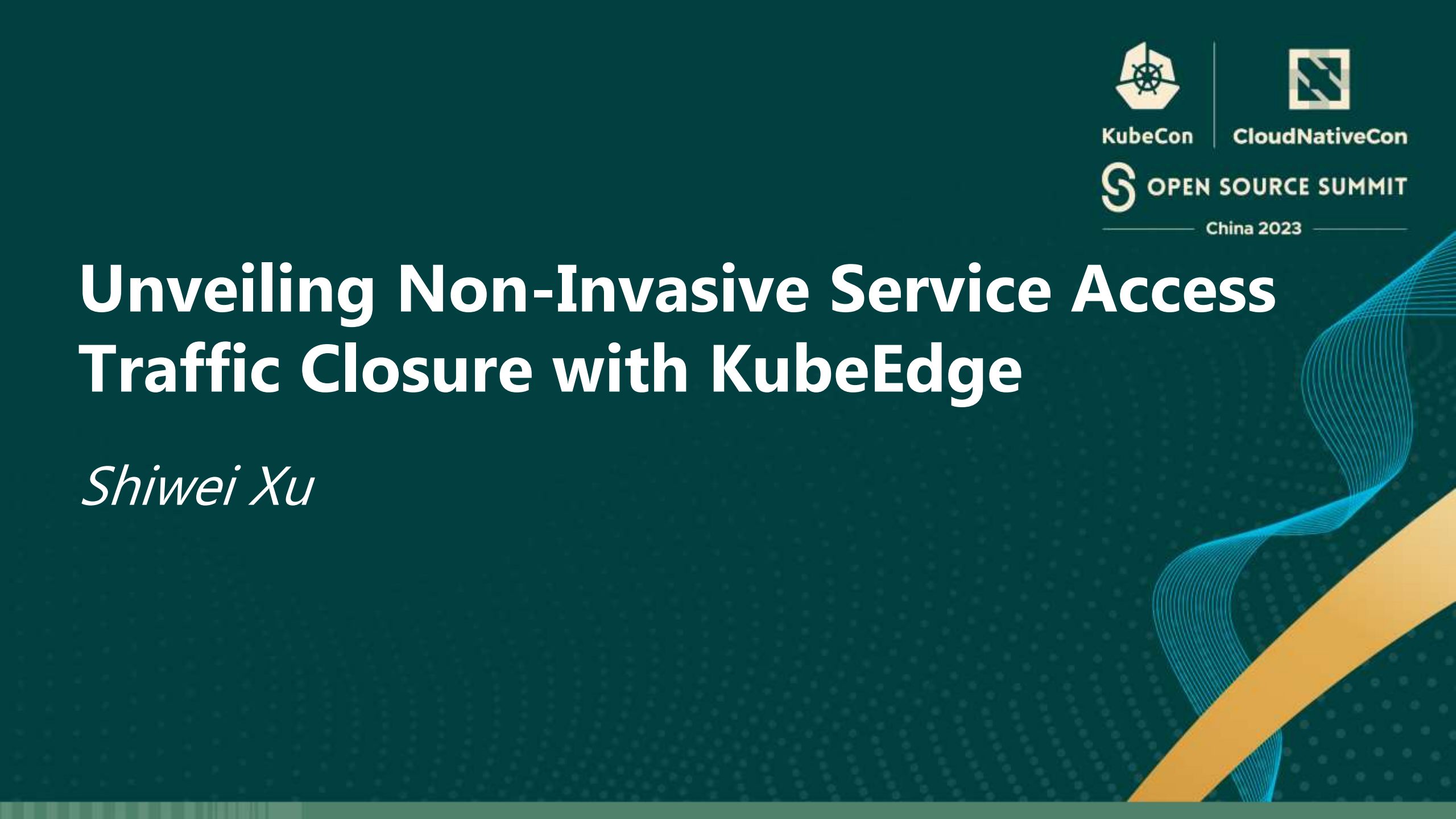

KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023



Unveiling Non-Invasive Service Access Traffic Closure with KubeEdge

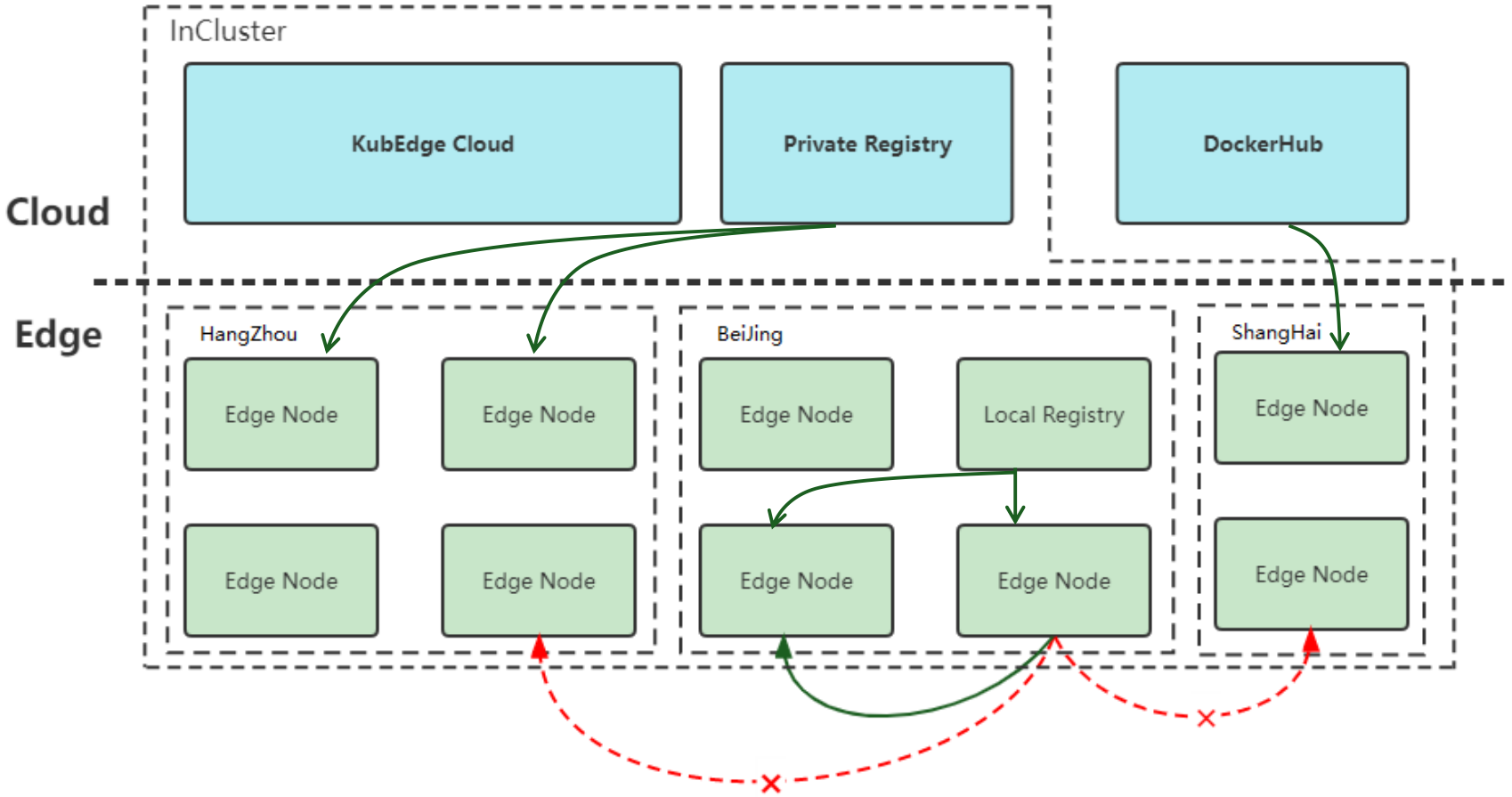
Shiwei Xu

Outline

- Background and motivation
- Feature overview and design
- Introduction to our implementation
- Current Status and future plans

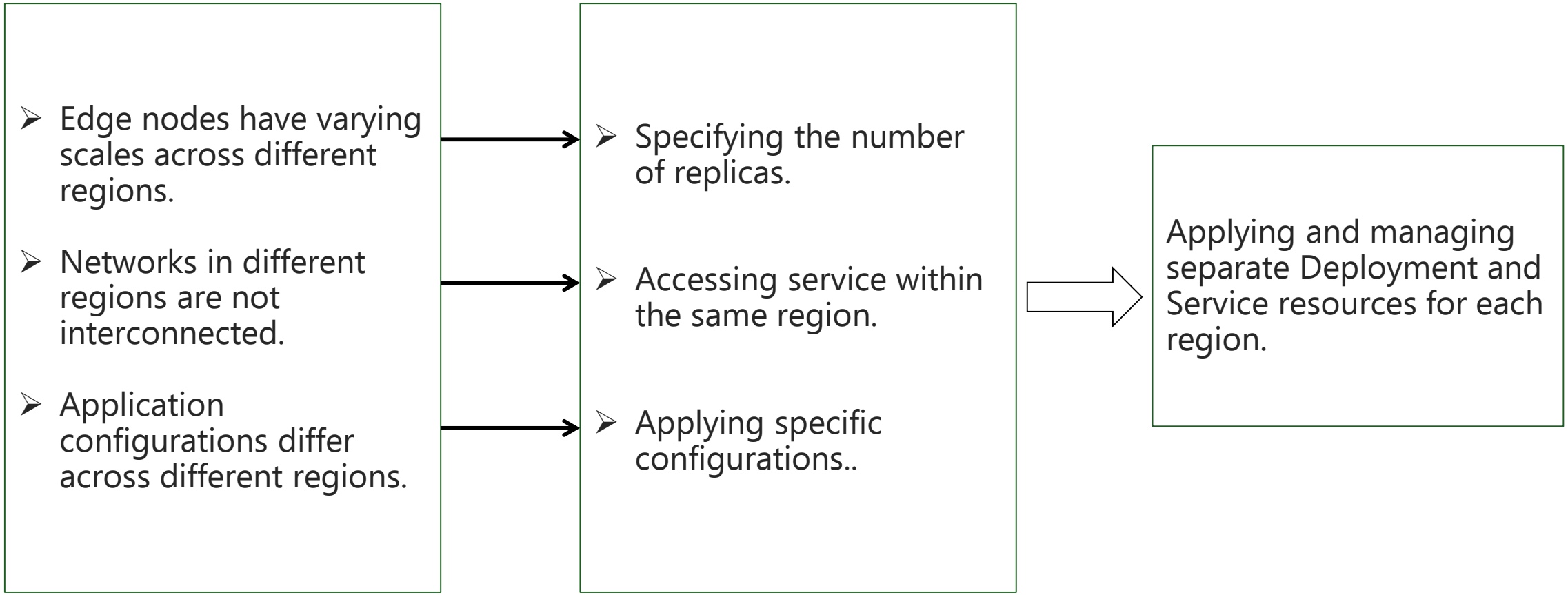
Background and Motivation

The challenges of deploying edge applications across regions

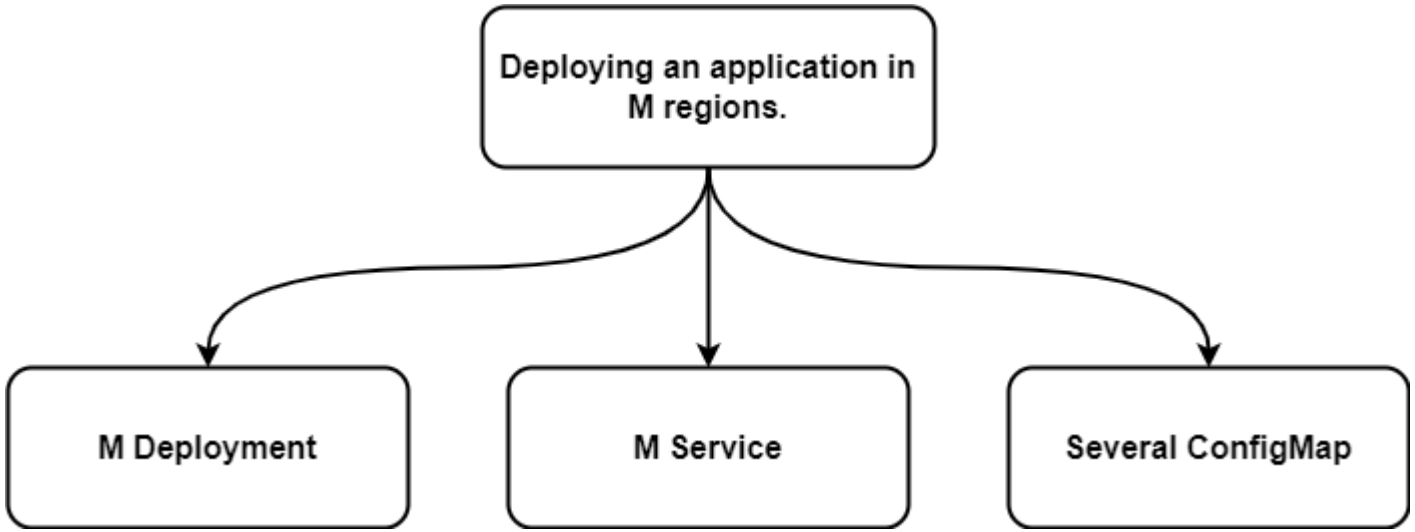


- Edge nodes have varying scales across different regions.
- Networks in different regions are not interconnected.
- Application configurations differ across different regions.

Traditional solutions



Explosive growth in the number of applications

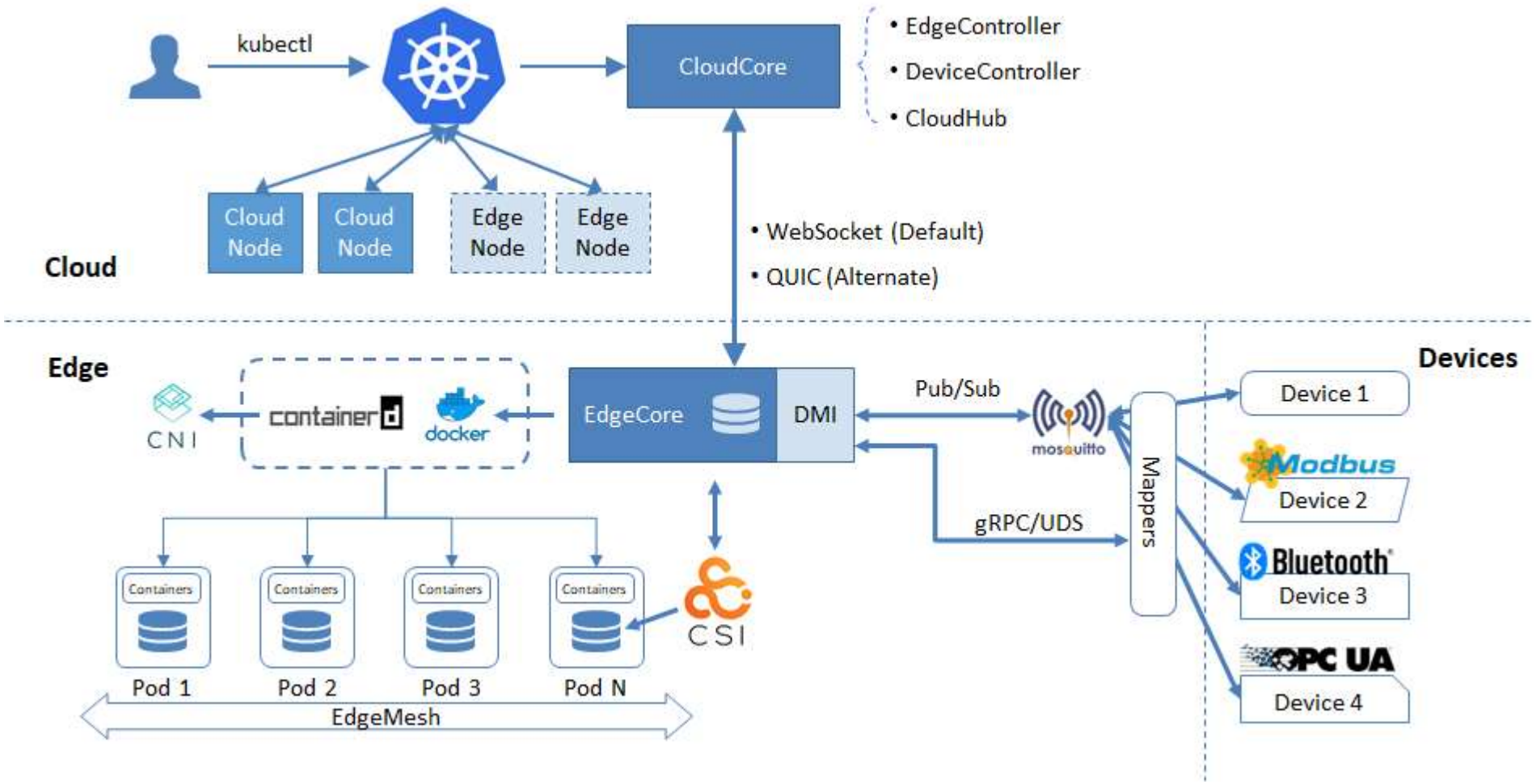


With the growth of geographical regions and the number of applications, the management of applications becomes increasingly complex, resulting in increased operational costs.

Pt. 2

Feature overview and design

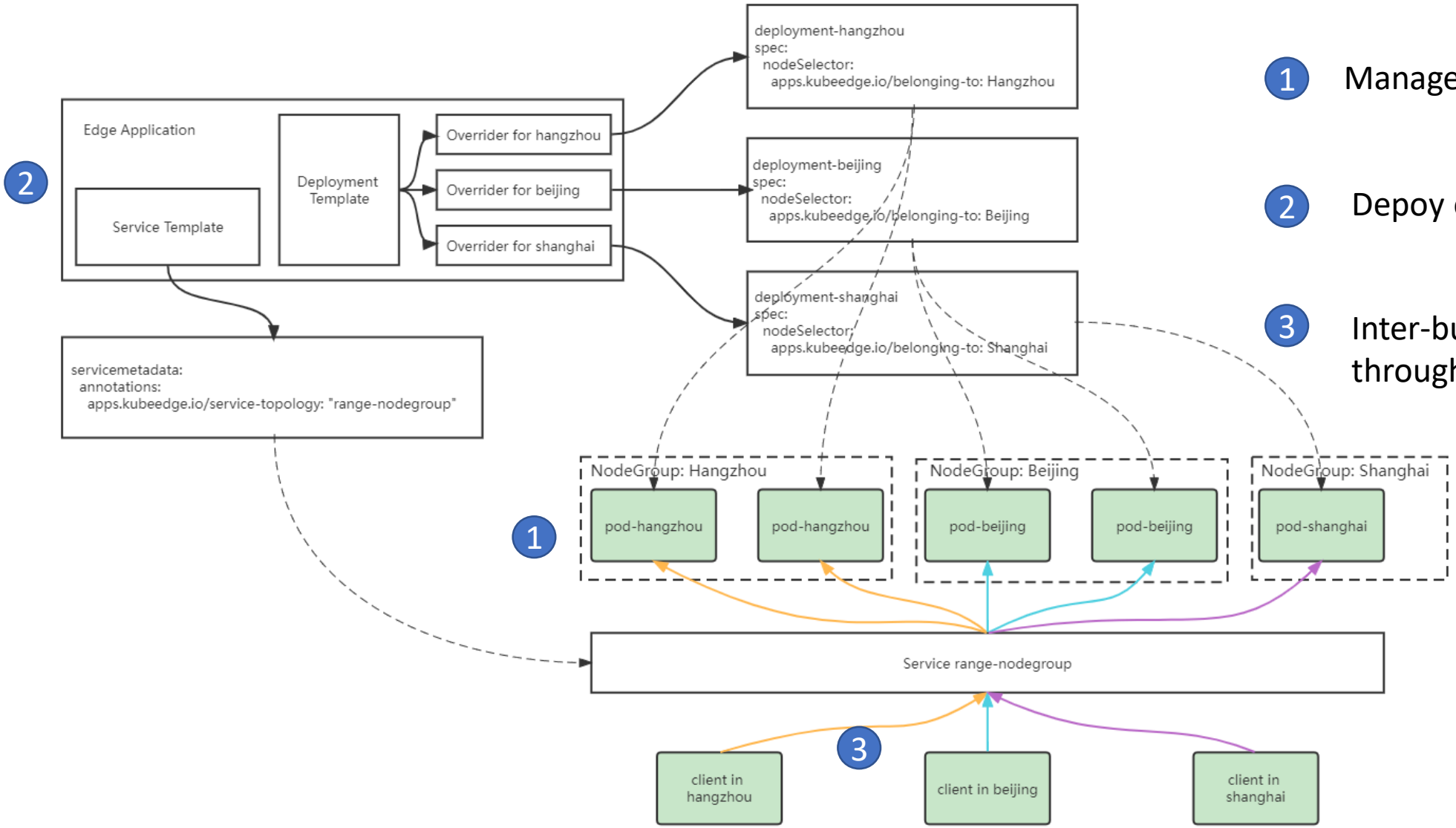
KubeEdge Architecture



Key Features

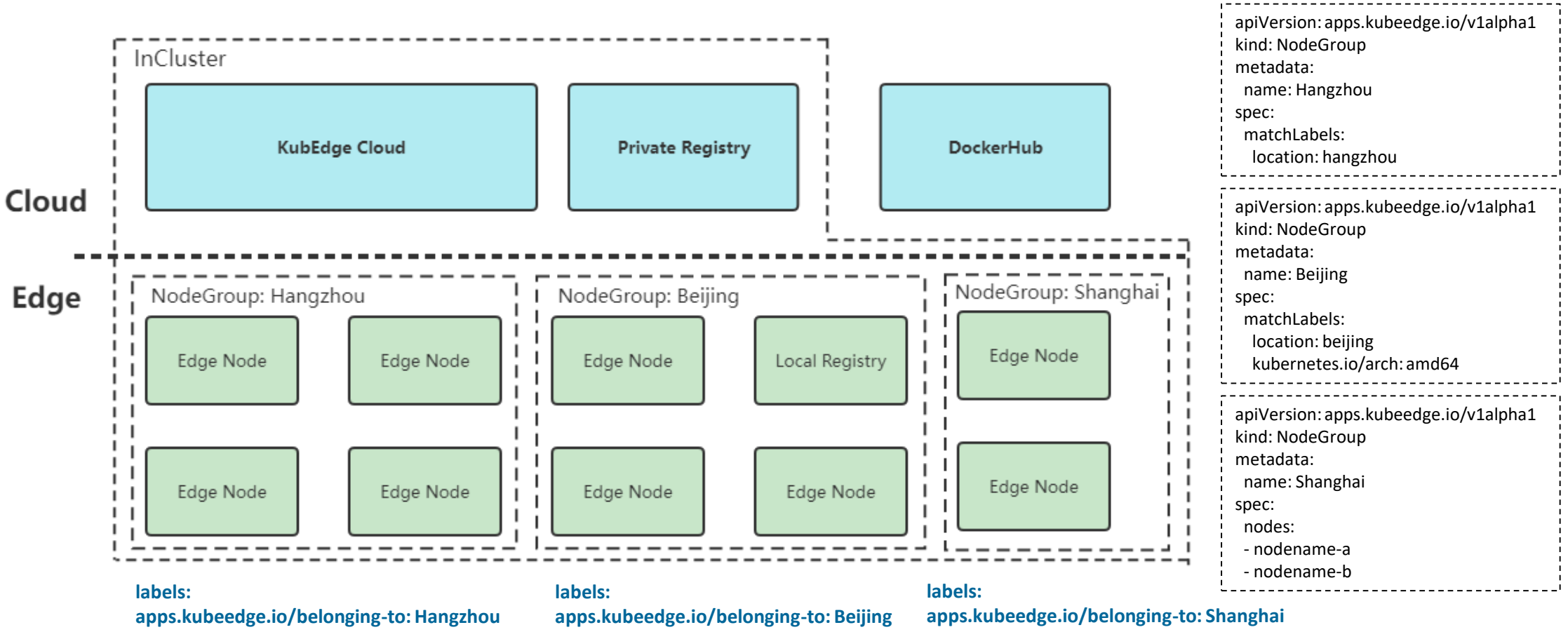
Node Grouping	Organize the edge nodes in different regions in the form of node groups.
Edge application	Package the application resources as a whole and meet the differentiated deployment requirements between different node groups.
Traffic Closure	Allowing services to route traffic based on the node topology of the cluster

Overall Capability Overview of Edge Node Group Management

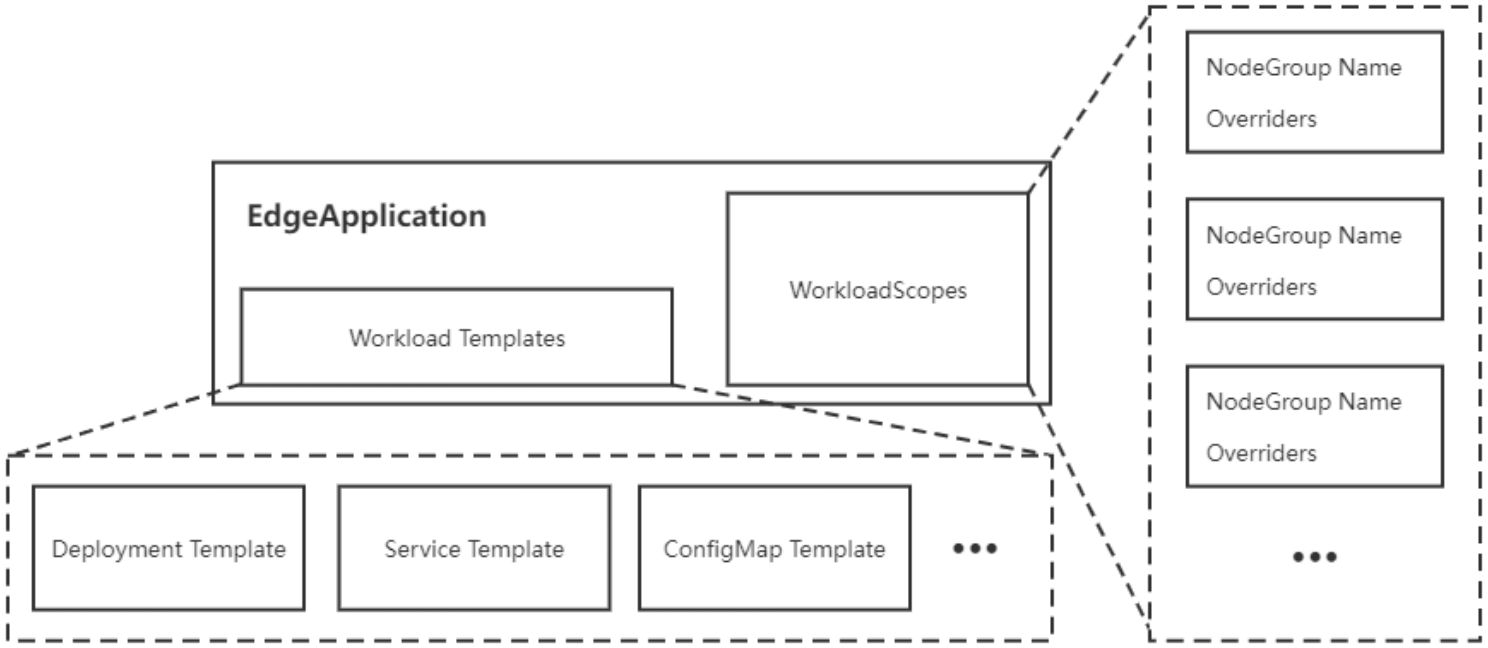


1. Manage Node group
2. Depoy edgeApplication
3. Inter-business communication through services

Node Grouping



Edge application



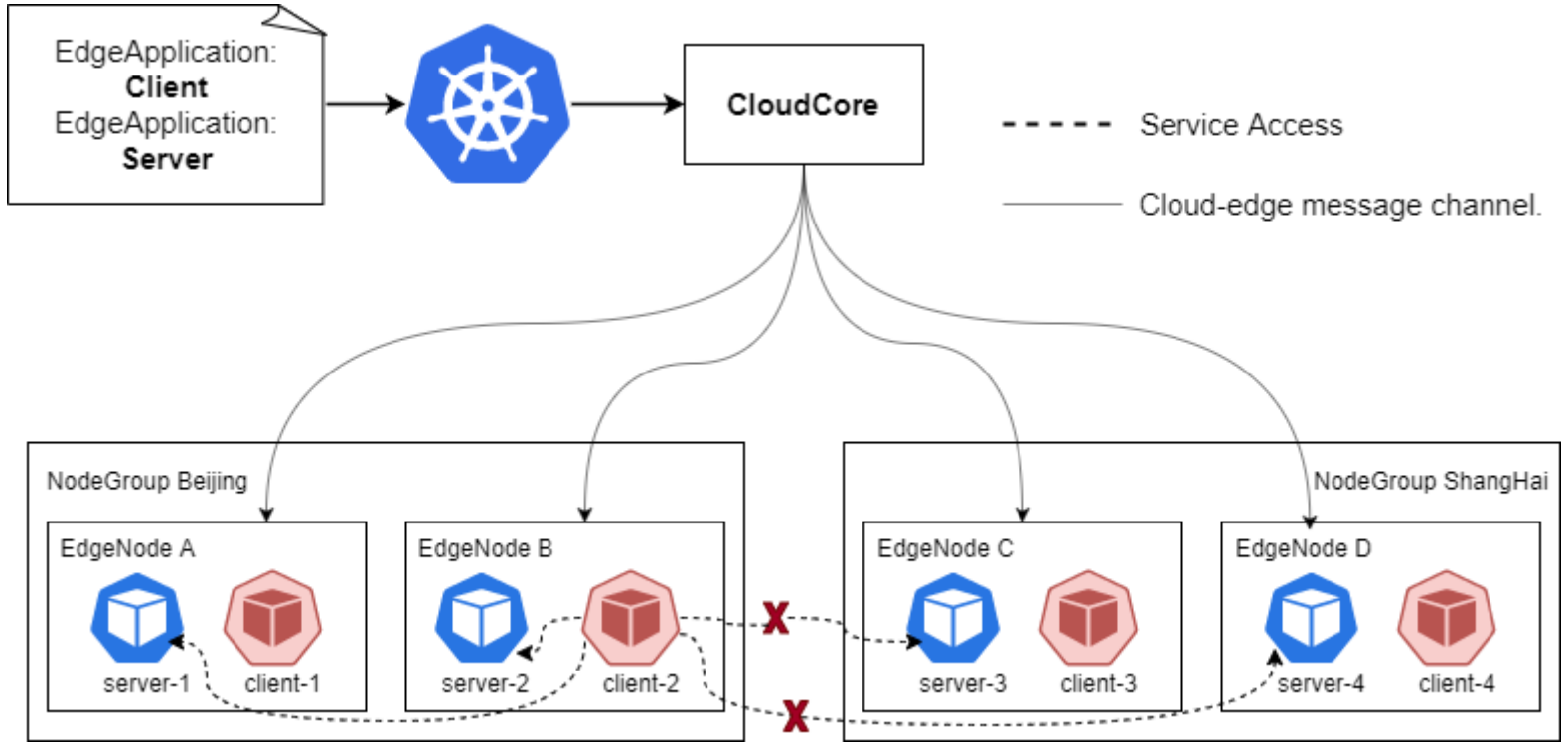
Workload Templates: Resource templates required for edge applications, such as Deployment Template, Service Template, and ConfigMap Template

WorkloadScopes: Differentiated configuration of resource templates based on the requirements of different node groups

Replicas Overrider:
Differentiated configuration of replicas number.

Image Overrider:
Differentiated configuration of images, including image repository address, repository name, and tags.

Traffic Closure



Service

```
metadata:
  name: server
  annotations:
    apps.kubeedge.io/service-topology: range-nodegroup
```

Design philosophy

Unified operation and maintenance

- Get/Modify/create/delete Deployment in region B → Get/Modify/create/delete EdgeApplication in region B
- Individual node operation and maintenance → Unified operation and maintenance through nodeGroup

Flexible expandability

- Use Unstructured structures for internal implementation to reduce coupling with specific resources
- It does not interfere with the native resource processing process and reduces the coupling with the Kubernetes component Reconciliation.

Larger scale

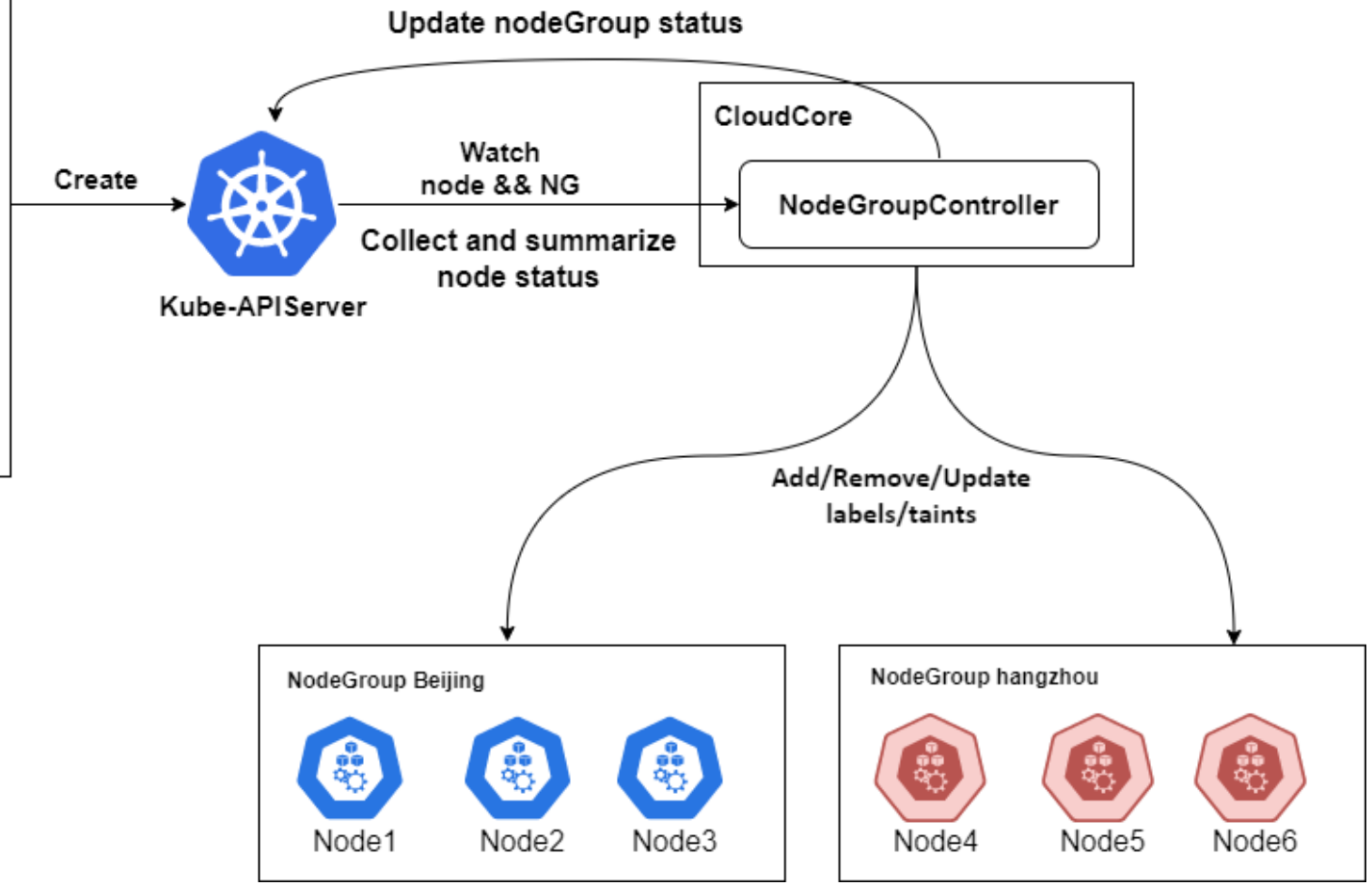
- Native kubernetes endpoints are distributed to all nodes, The traffic closed-loop mechanism greatly reduces the bandwidth of cloud and edge communications.
- Unify aggregation status through NodeGroup and EdgeApplication to improve large-scale management

Pt. 3

Introduction to our implementation

Implementation of NodeGroup

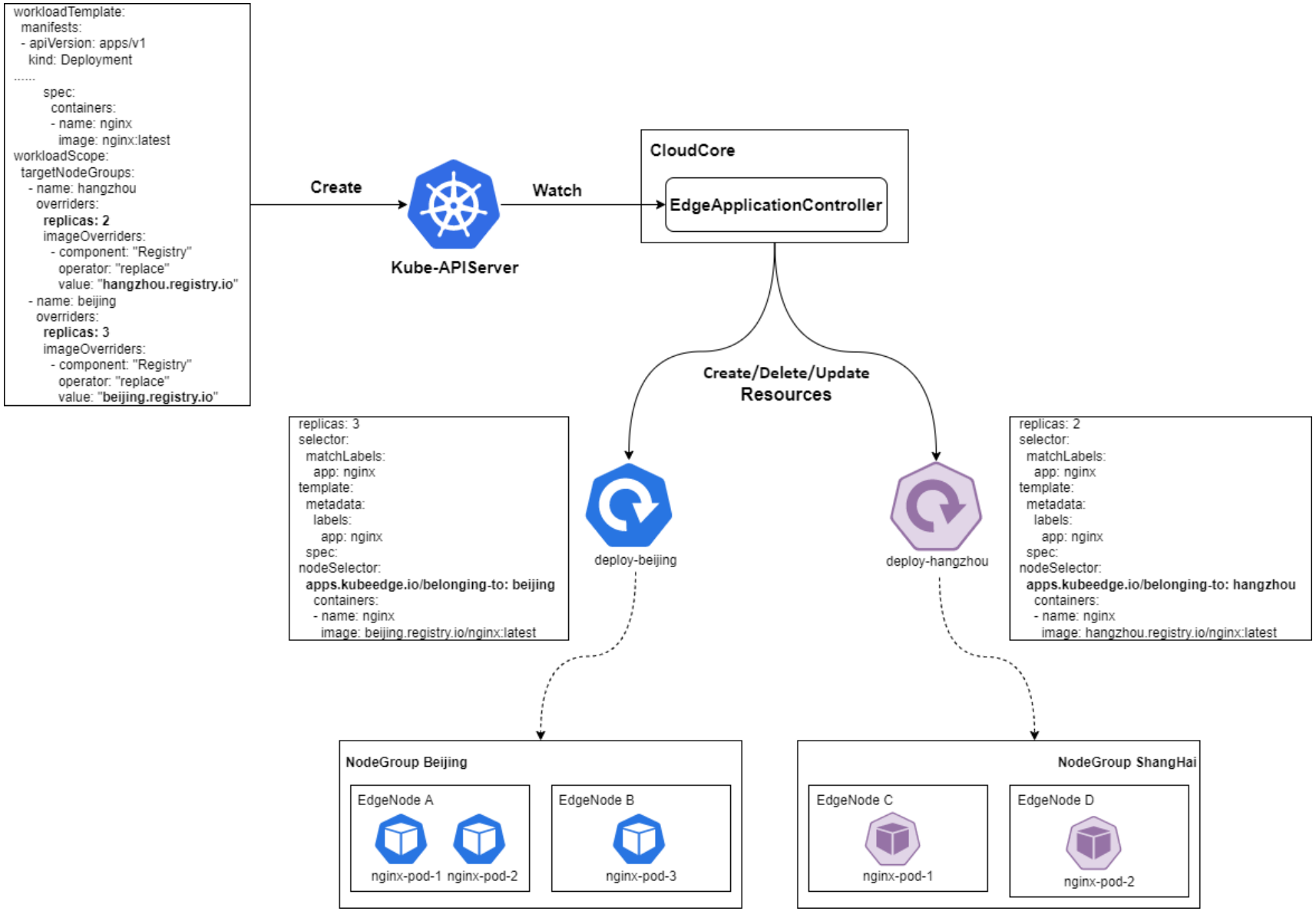

```yaml
apiVersion: apps.kubeedge.io/v1alpha1
kind: NodeGroup
metadata:
  name: hangzhou
spec:
  matchLabels:
    location: hangzhou
---
apiVersion: apps.kubeedge.io/v1alpha1
kind: NodeGroup
metadata:
  name: beijing
spec:
  matchLabels:
    location: beijing
```



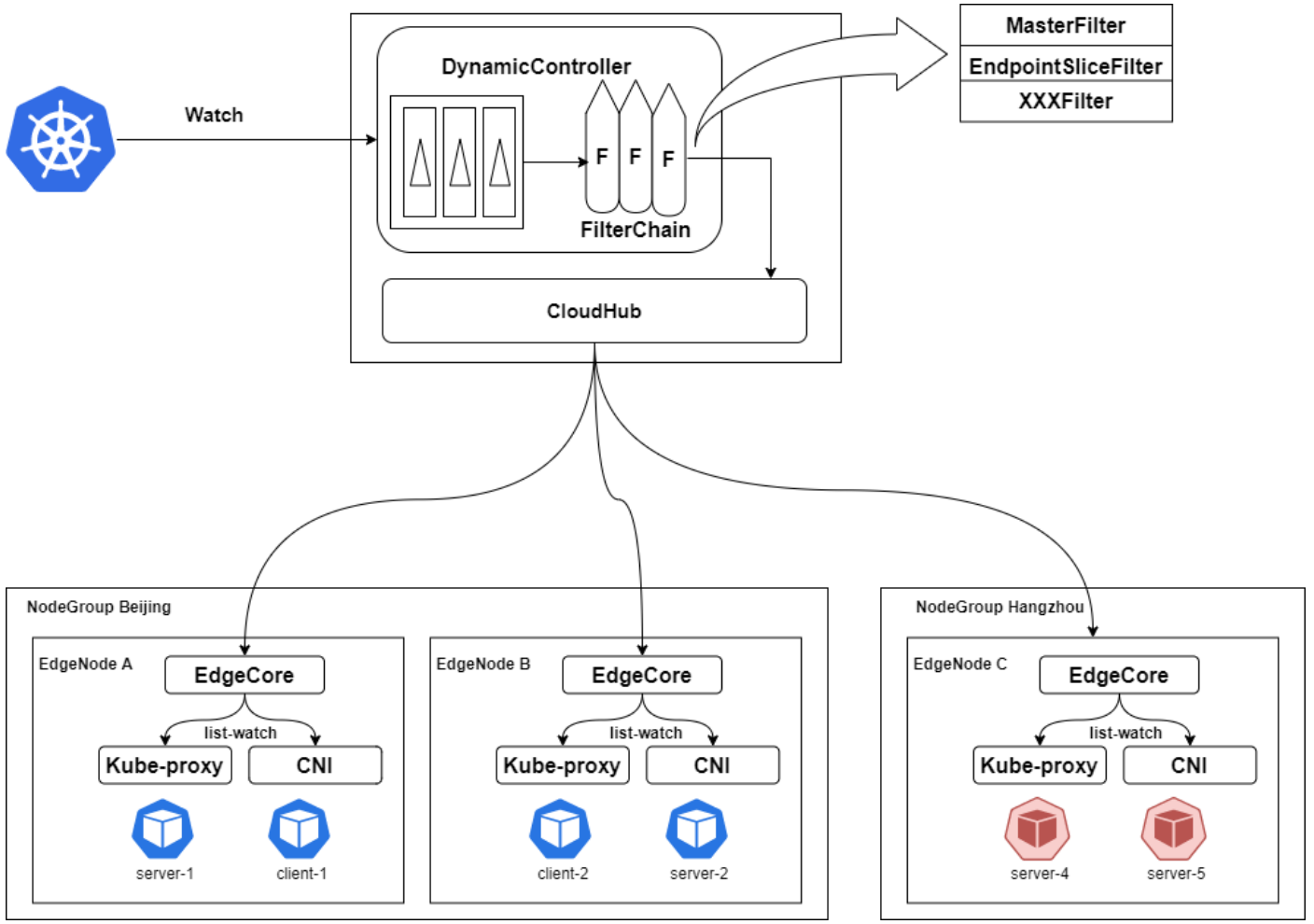
label:
location:beijing

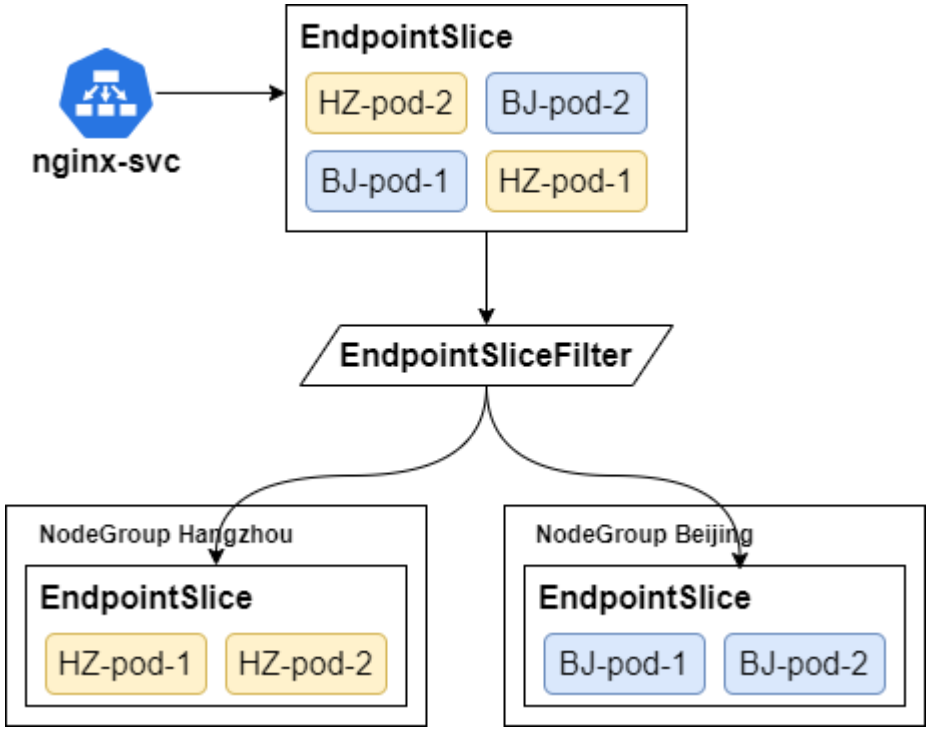
label:
location:hangzhou

Implementation of EdgeApplication



Resource Filter





EndpointSlice filter example

Pt. 4

Current Status and future plans

Current Status and future plans

Current Status

- Support packaging of Deployment, Service, and ConfigMap.
- Support traffic loop closure.
- Support partial state collection of resources.

Future plans

- Support for more resources such as StatefulSet.
- Enhance application state collection.
- Enable a more user-friendly resource representation in kubectl.

Q&A

Design / KEP:

https://github.com/kubeedge/kubeedge/blob/master/docs/proposals/node-group-management.md