Interconnecting cloud applications with

SKUPPER

https://skupper.io/

Valerio Bartolini
Sr. Software Engineer at Red Hat

Three words to remember

Skupper is a new approach to connect heterogeneous applications through
high level multi-routing capabilities.

The drivers for xxx cloud

Productivity

Cloud services for data analysis and resources for deployment/development at the same time

Keeping data on-site

Retaining sensitive information within the organization's premises with low latency

Efficient resource access

Expand capacity on demand and scale down

Availability & Reliability

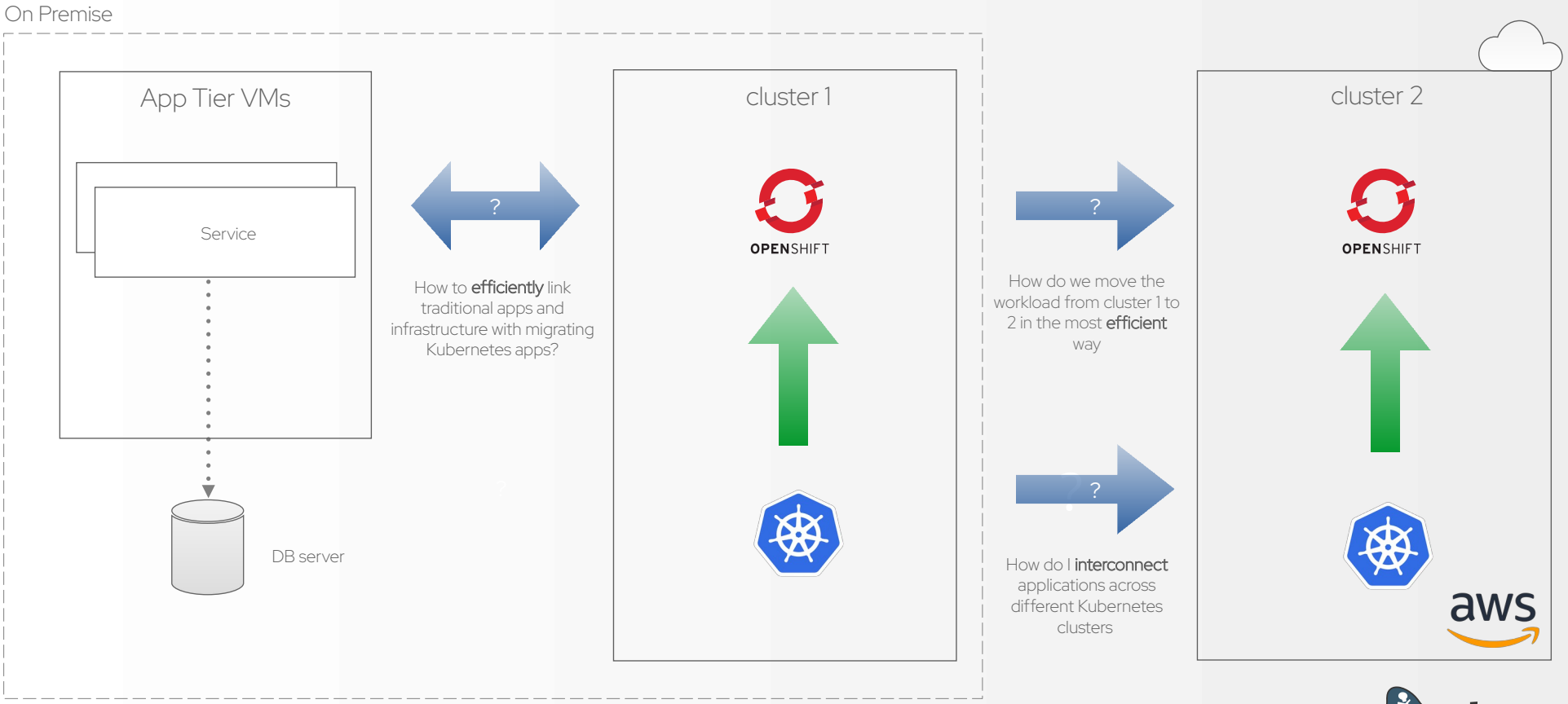
Distributing workloads and disaster recovery processes.

Heterogeneous apps

Diverse applications integrate smoothly across multiple providers, optimizing performance and resource use.

Skupper

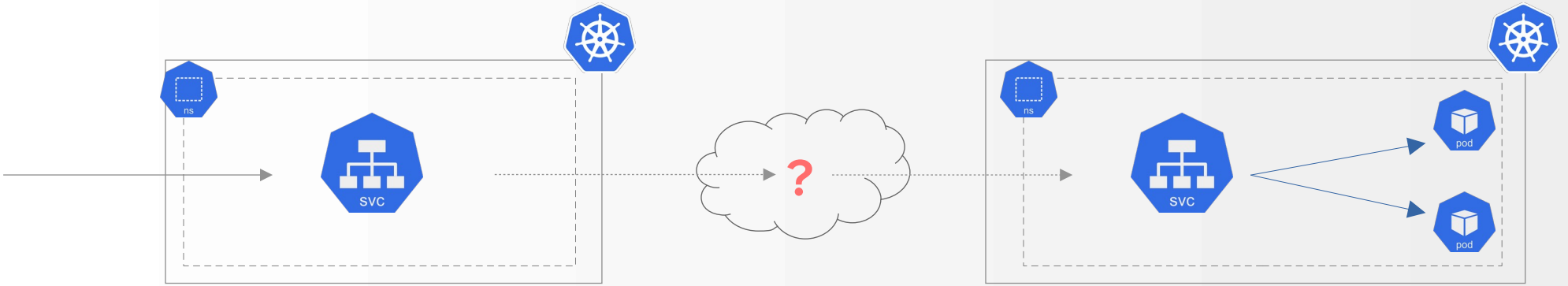
The Challenge



Skupper

Do we have solutions?

Services across different clouds



Public IP addresses

- Directly accessible from internet
- No connectivity behind NAT
- Each IP is a co$t

Larger Provider Networks

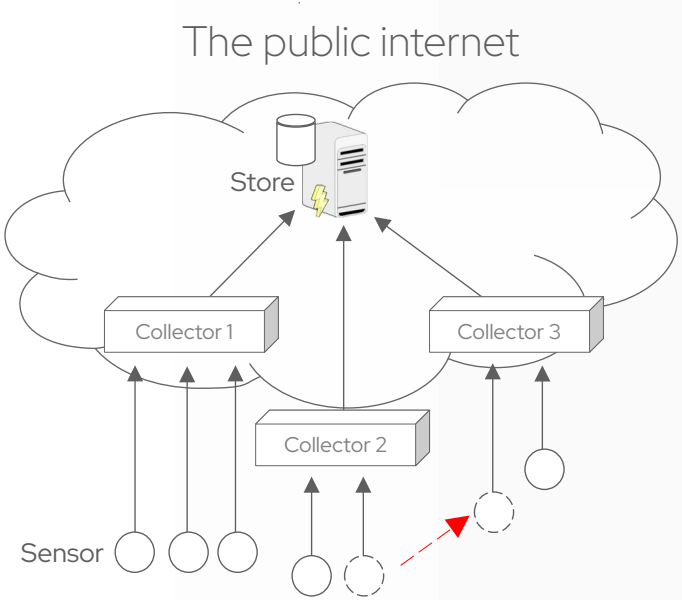
- Network isolation
- Vendor lock in
- Requires cluster privs
- Each connection is a co$t

VPN

- Network isolation
- Iptables & firewall config
- Admin privs

Skupper

Edge connectivity



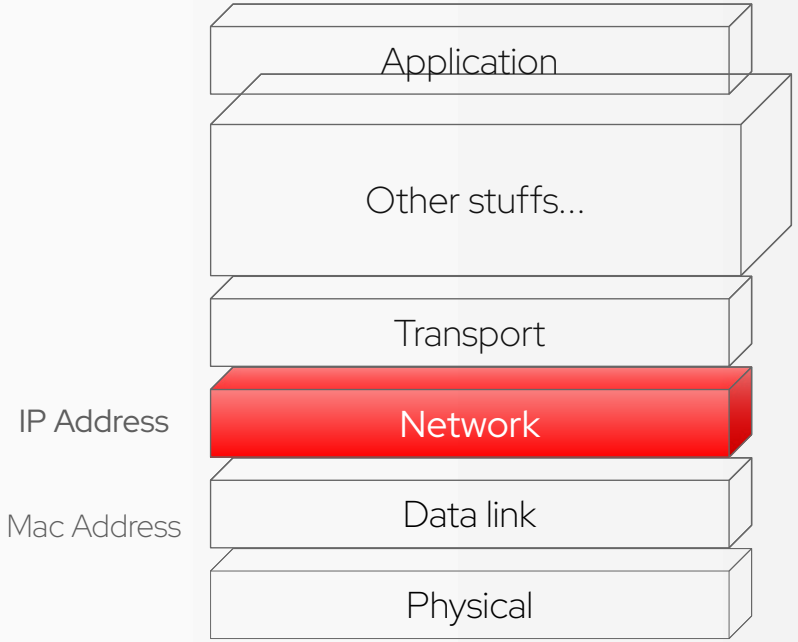
Configure each sensor to connect to its corresponding collector

Collector can be outside the public internet

Mobility issue

Skupper

IP addressing constrains



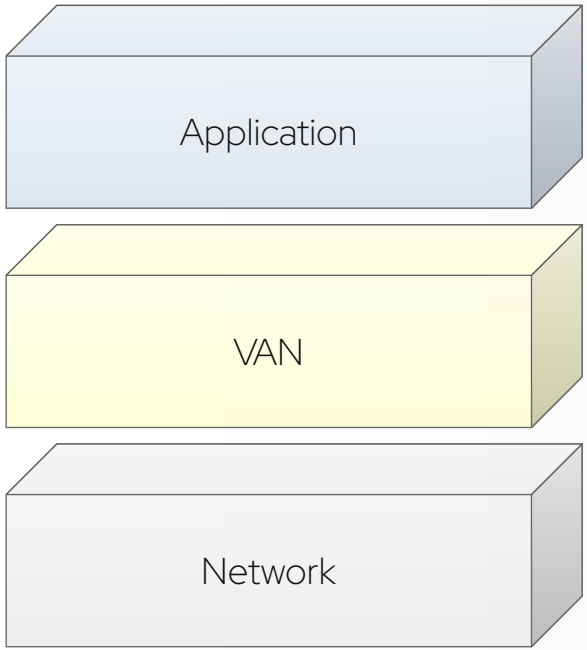
The Internet protocol is not well suited for XXX cloud interconnect

We are trying to adapt our architecture with the network infrastructure

Skupper

A different approach

Virtual application network



Connect services not hosts

Multiple services with the same address

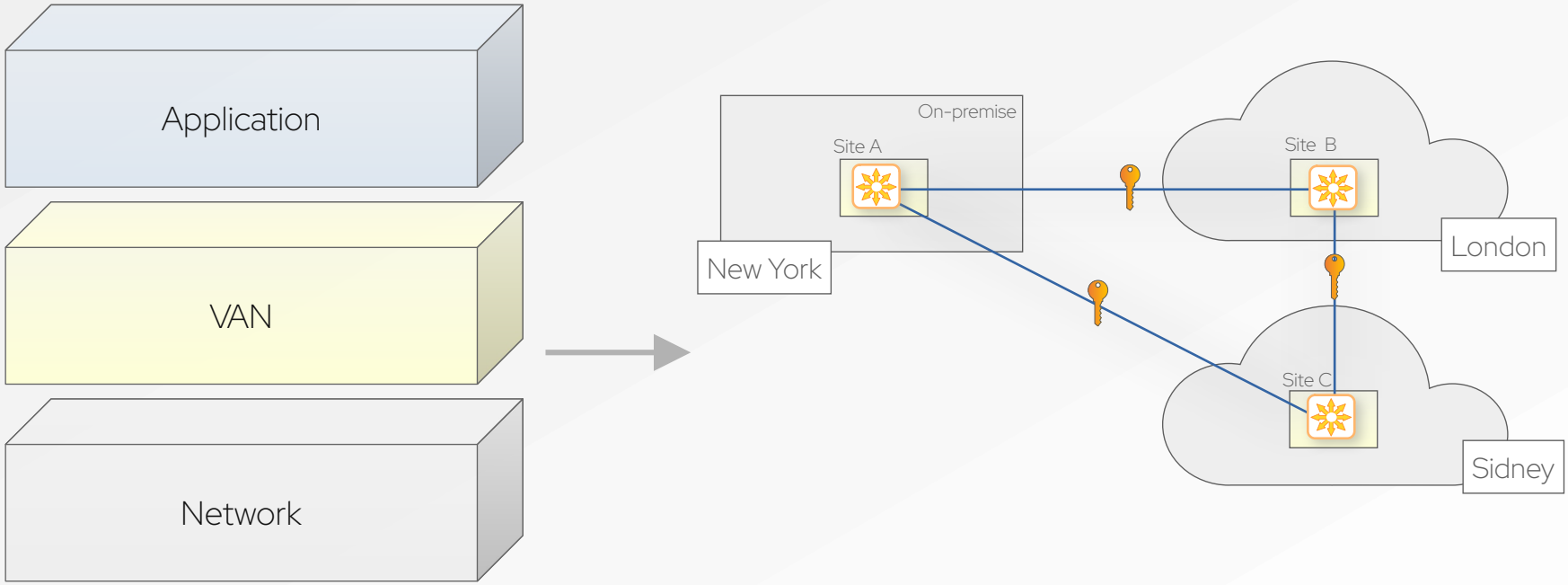
Application doesn't need to be modified

Natural fit into container platforms

Lightweight

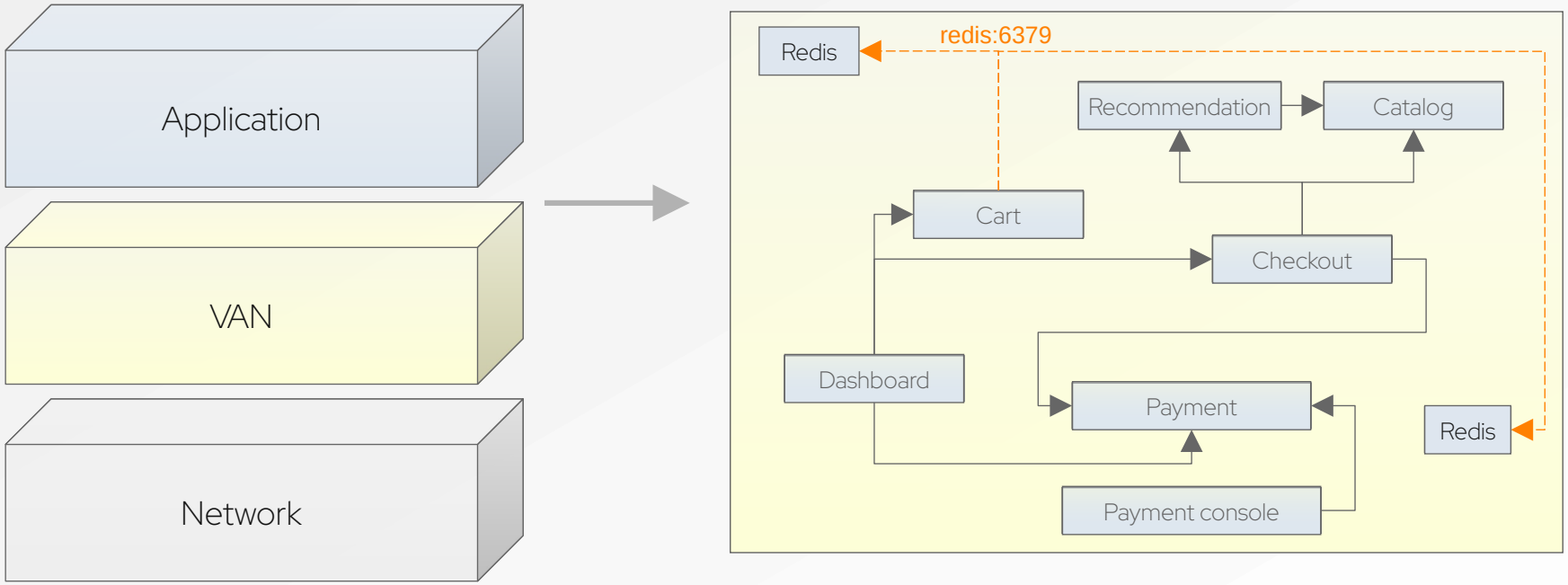
Skupper

Network topology



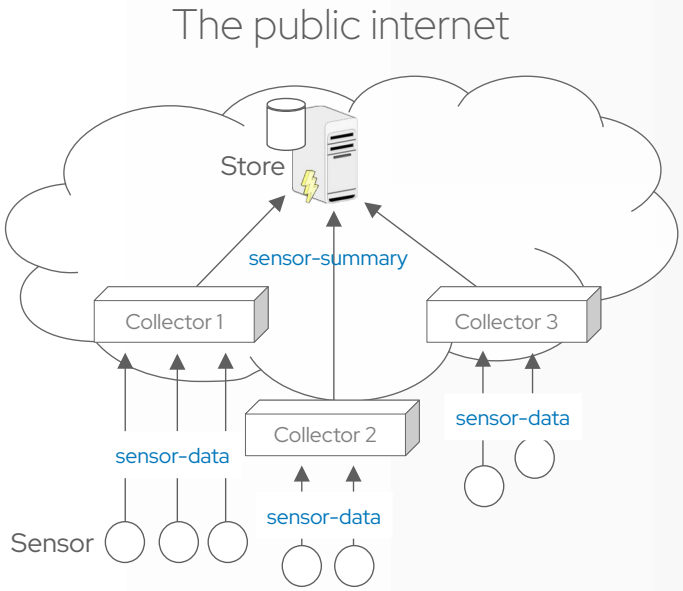
Skupper

Application topology



Skupper

...back to edge connectivity case

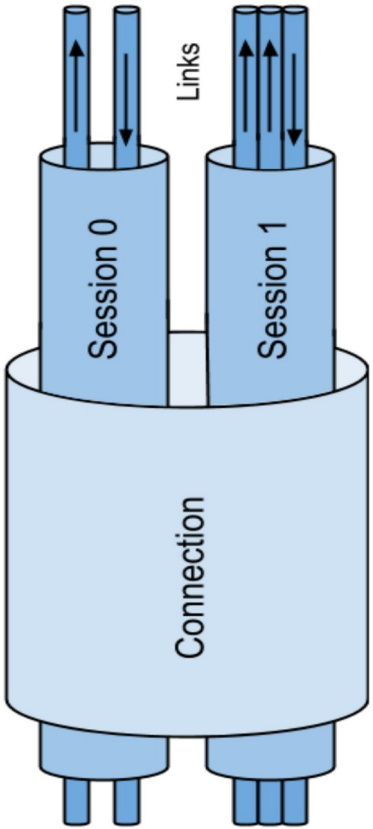


curl -X POST http://sensor-data:8080/sensor-id

The VAN selects the most capable collector based on the sensor-data address

Skupper

Service delivery and distributed systems



Multiplexing

Flow Control

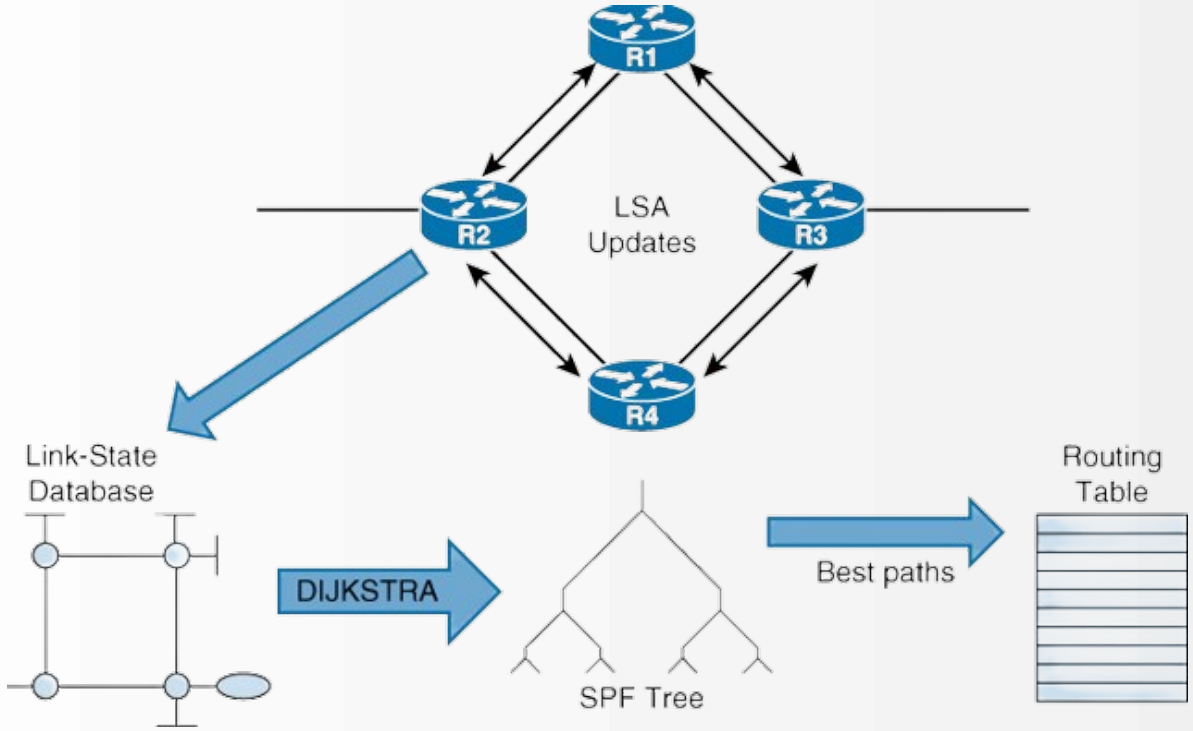
Delivery guarantees

Asynchronous, full-duplex communication

mTLS

Skupper

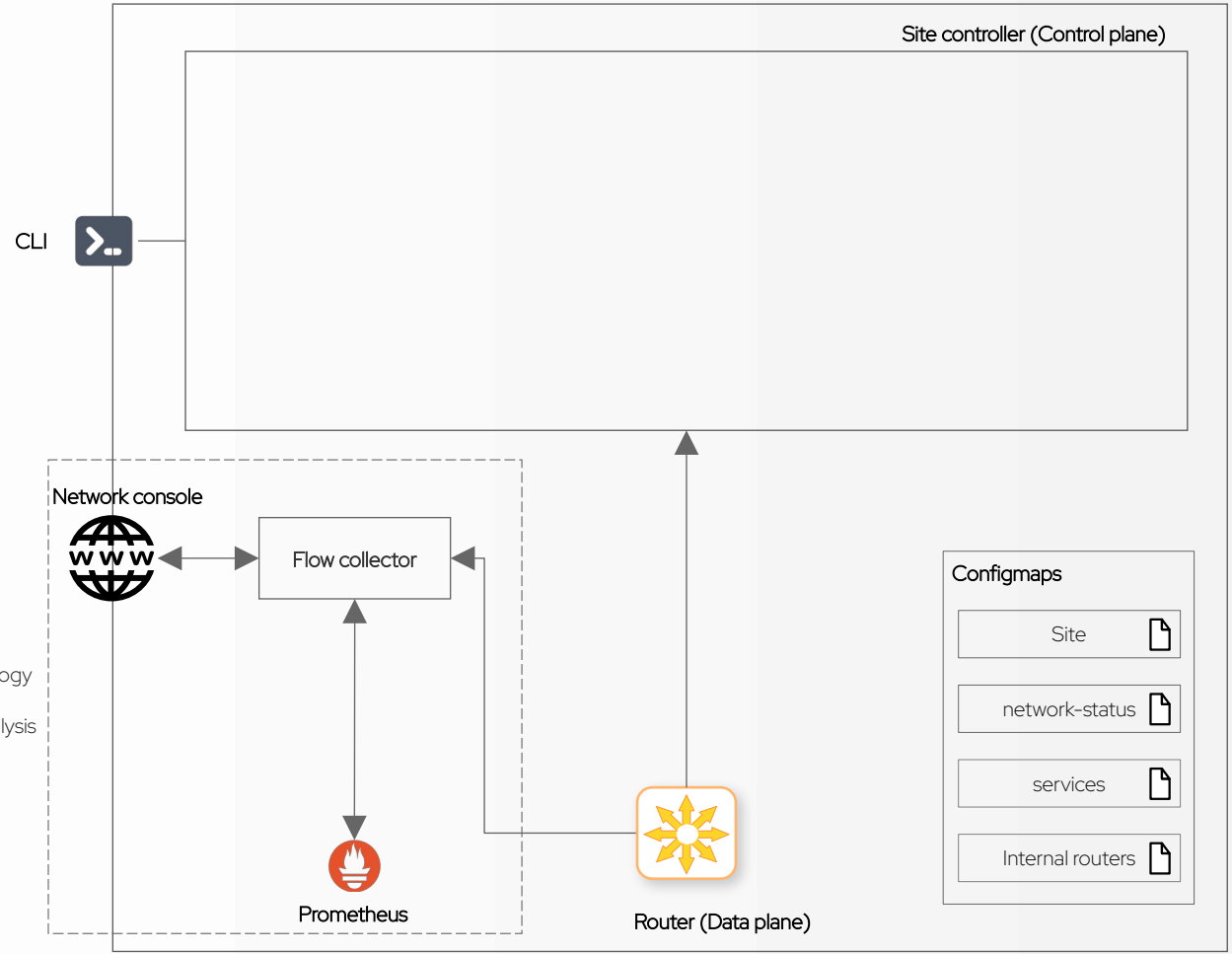
Optimizing Network Pathways and Mobile addressing



Skupper

Skupper

Architecture



Handle the communication between routers

Integration for: xKS, Docker/Podman, VMs, bare metal...

Creates a distributed application running across namespaces

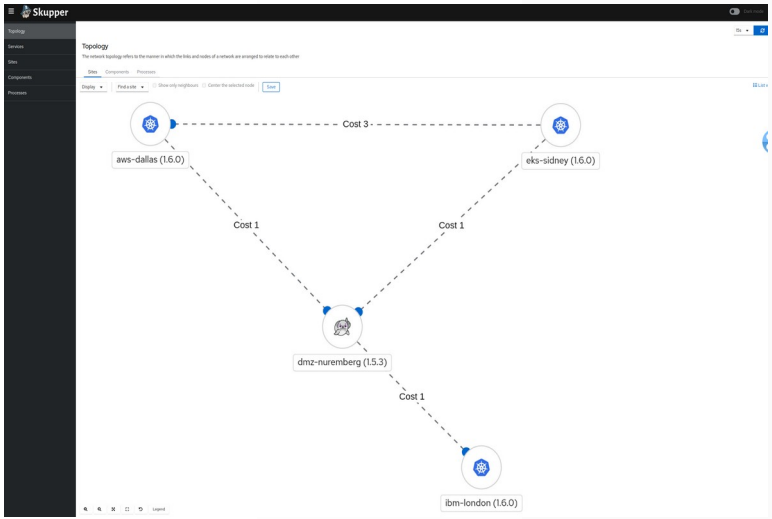
Handle the policies

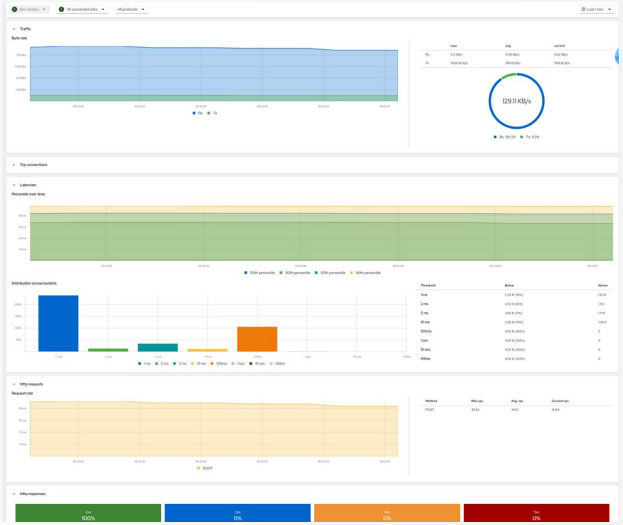
Real time Topology

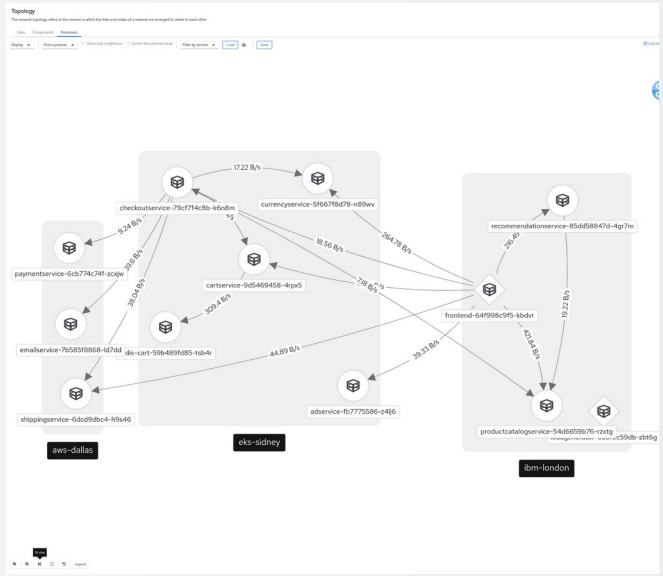
Flow Metric analysis

Skupper

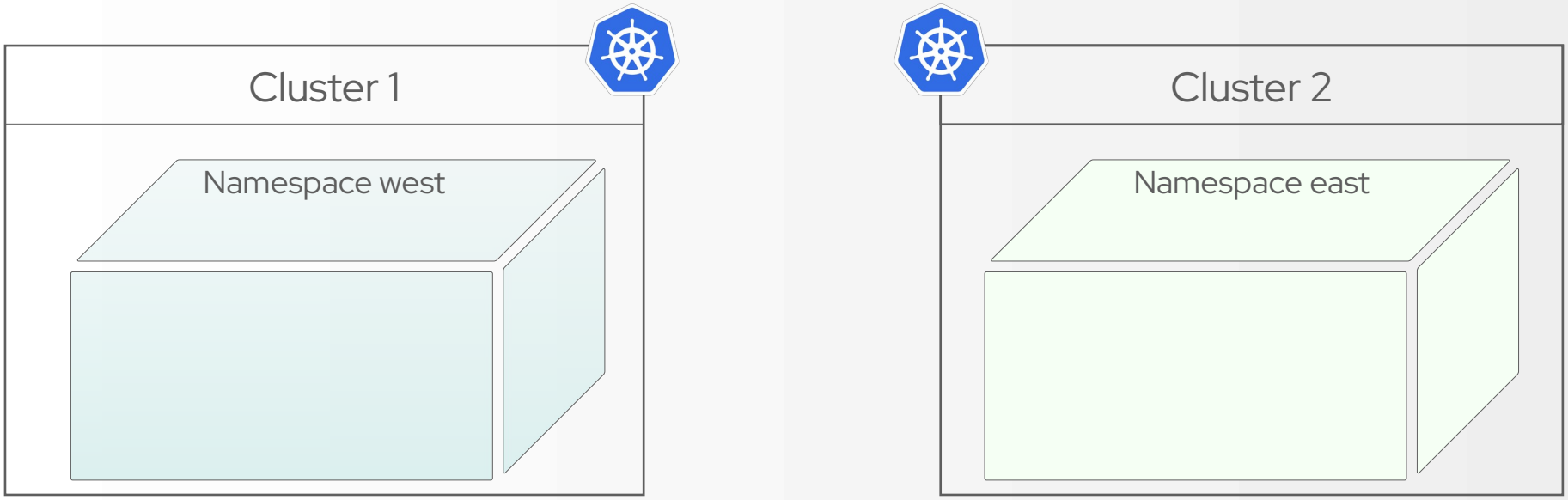
Console


Skupper
Topology
Cost 3
aws-dallas (1.6.0)
eks-sidney (1.6.0)
Cost 1
Cost 1
dmz-nuremberg (1.5.3)
Cost 1
ibm-london (1.6.0)




Topology
17.22 B/s
checkoutservice-79cf7f4c8b-k6n8m
currencyservice-5f667f8d78-n89wv
recommendationservice-85dd58847d-4gr7m
paymentservice-6cb774c74f-zcxjw
cartservice-9d5469458-4rpx5
frontend-64f998c9f5-kbdvr
emailservice-7b585f8868-ld7dd
dis-cart-59b489fd85-tsb4r
shippingservice-6dcd9dbc4-h9s46
adservice-fb7775586-z4lj6
productcatalogservice-54d6659b76-rzxtg
aws-dallas
eks-sidney
ibm-london

Create a Skupper network (I)

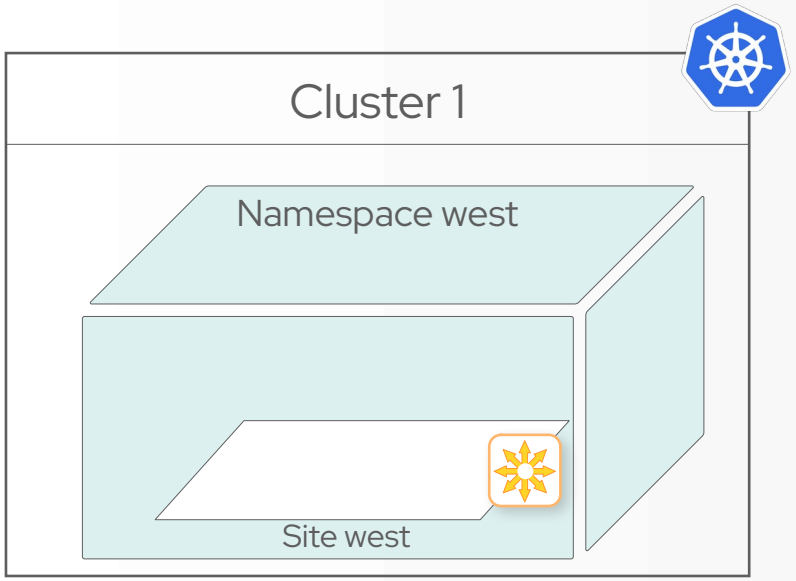


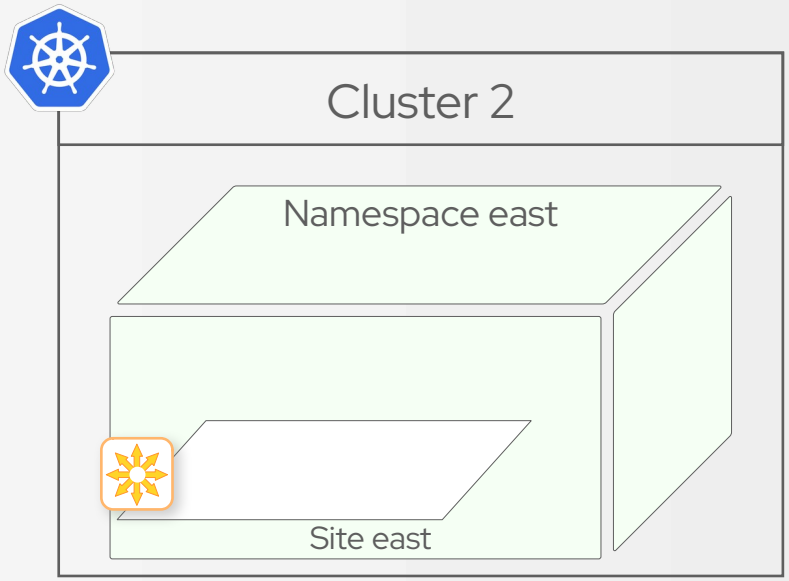
kubectl create namespace west

kubectl create namespace east

Skupper

Create a Skupper network (II)



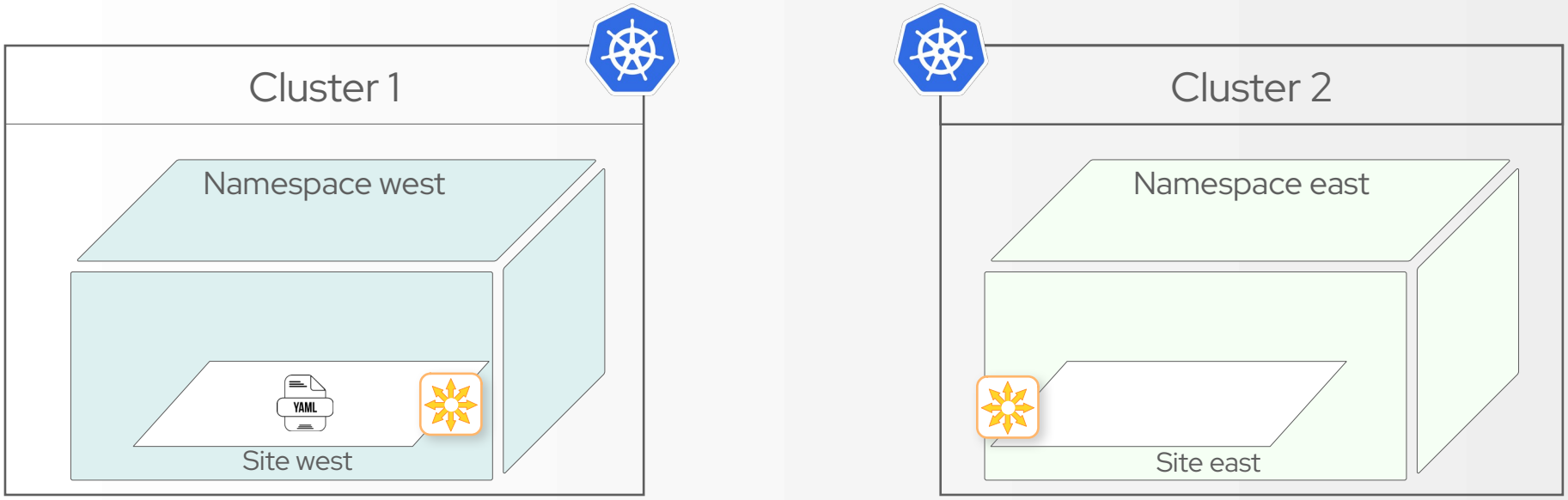


skupper init

skupper init

Skupper

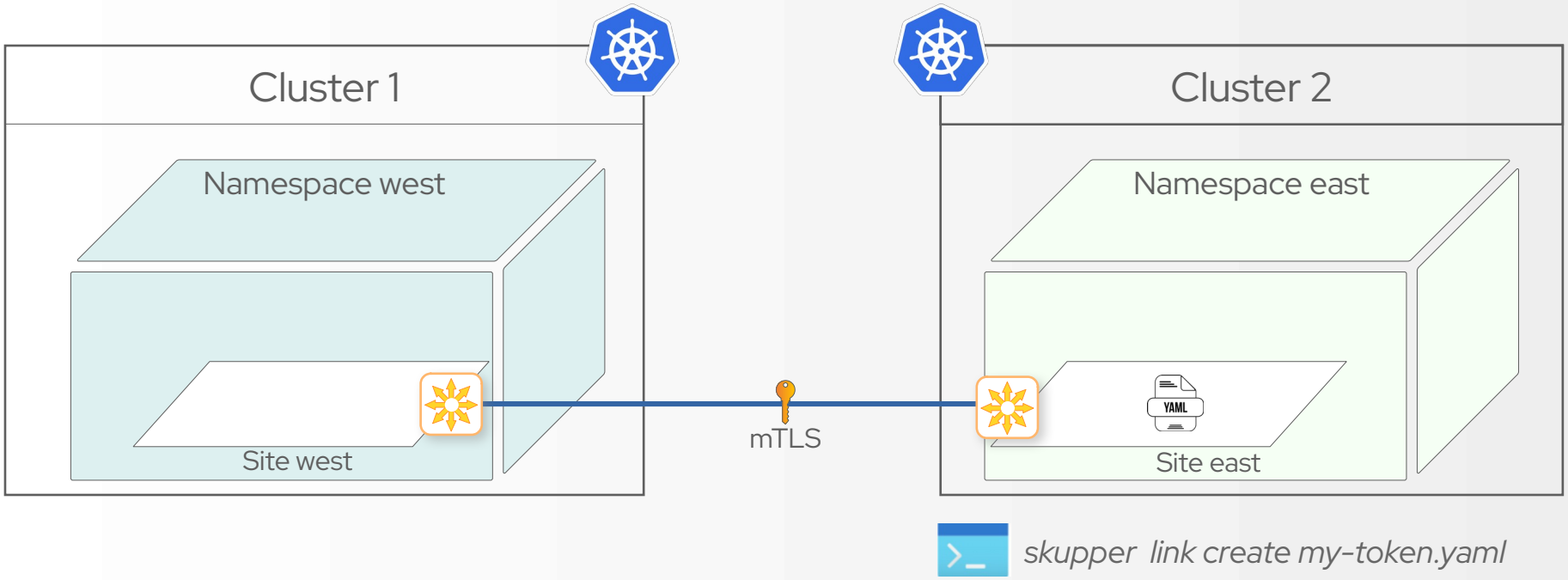
Create a Skupper network (III)



skupper token create my-token.yaml

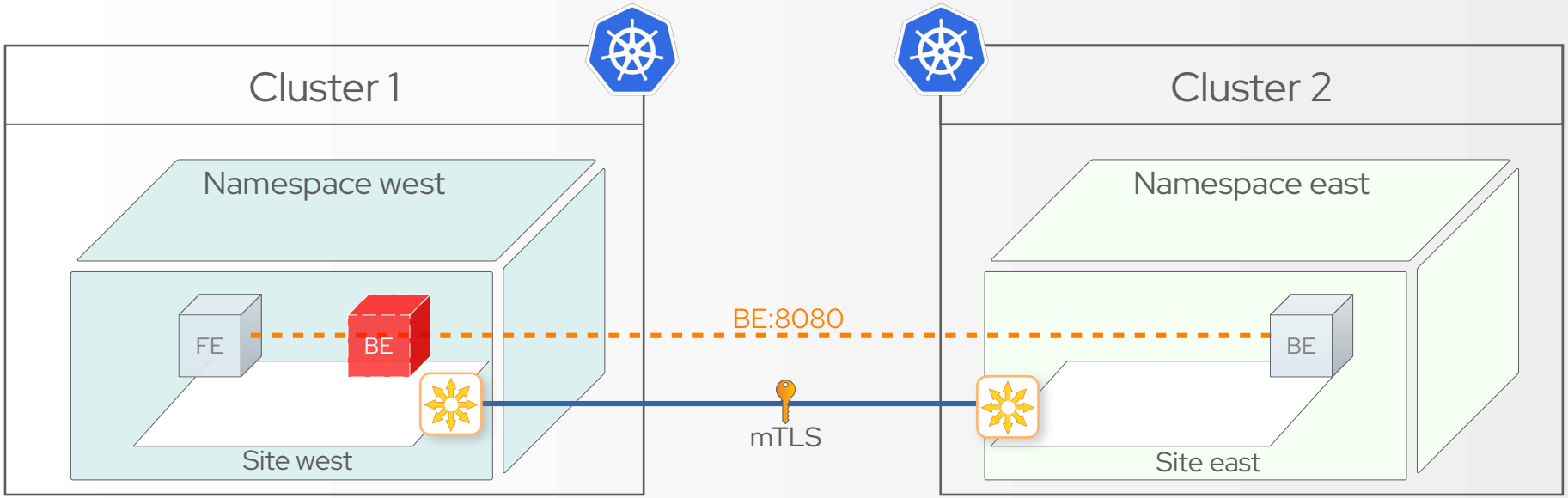
Skupper

Create a Skupper network (IV)



Skupper

Create a Skupper network (IV)



kubectl create deployment FE –image /FE

kubectl create deployment BE –image /BE

skupper expose deployment/BE –port 8080

Skupper

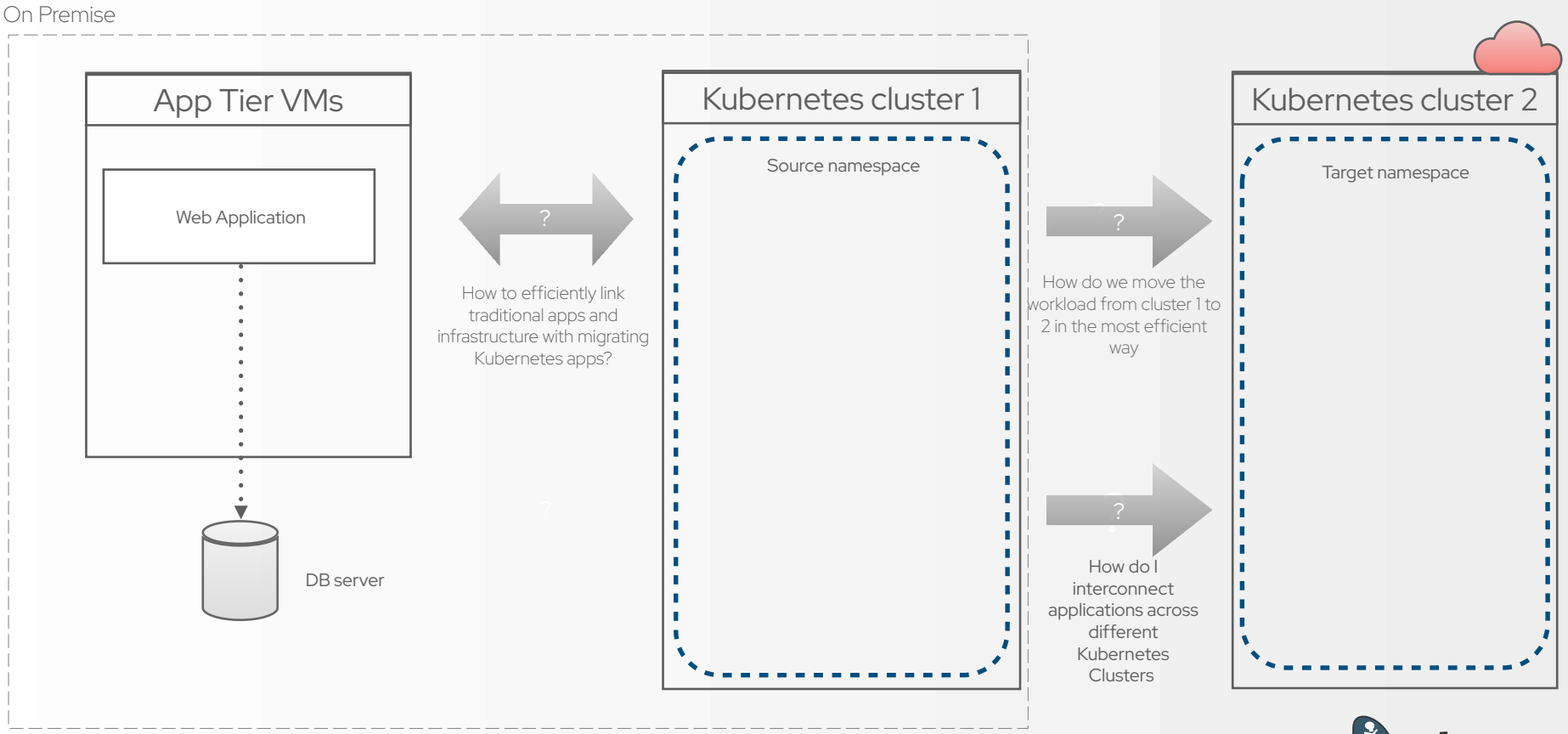
```
└─ oc get pod,svc
NAME                                          READY   STATUS    RESTARTS   AGE
pod/frontend-67c6b84d49-446jh                 1/1     Running   0          53d
pod/skupper-router-78847cc997-lt84p           2/2     Running   0          38d
pod/skupper-service-controller-ffcd985c4-9wslr   1/1     Running   0          39d

NAME                          TYPE           CLUSTER-IP       EXTERNAL-IP                                PORT(S)
service/backend               ClusterIP      172.21.144.214   <none>                                     8080/TCP
service/frontend              LoadBalancer   172.21.144.173   b8071065-us-east.lb.appdomain.cloud        8080:32182/TCP
service/skupper               ClusterIP      172.21.152.116   <none>                                     8081/TCP
service/skupper-router        ClusterIP      172.21.185.106   <none>                                     55671/TCP,45671/TCP
service/skupper-router-local  ClusterIP      172.21.236.24    <none>                                     5671/TCP
```

```
└─ oc get svc/backend -o yaml
apiVersion: v1
kind: Service
metadata:
  annotations:
    internal.skupper.io/controlled: "true"
  creationTimestamp: "2024-02-27T15:00:55Z"
  name: backend
  namespace: vb-hello-west
  ownerReferences:
  - apiVersion: apps/v1
    kind: Deployment
    name: skupper-router
    uid: 9733e510-1eeb-4408-908d-ee1686ee5a02
  resourceVersion: "55134134"
  uid: 0b66ba6c-31c9-4dbf-9831-fb31cdd2c43a
spec:
  clusterIP: 172.21.144.214
  clusterIPs:
  - 172.21.144.214
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - name: port8080
    port: 8080
    protocol: TCP
    targetPort: 1024
  selector:
    application: skupper-router
    skupper.io/component: router
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```

Skupper

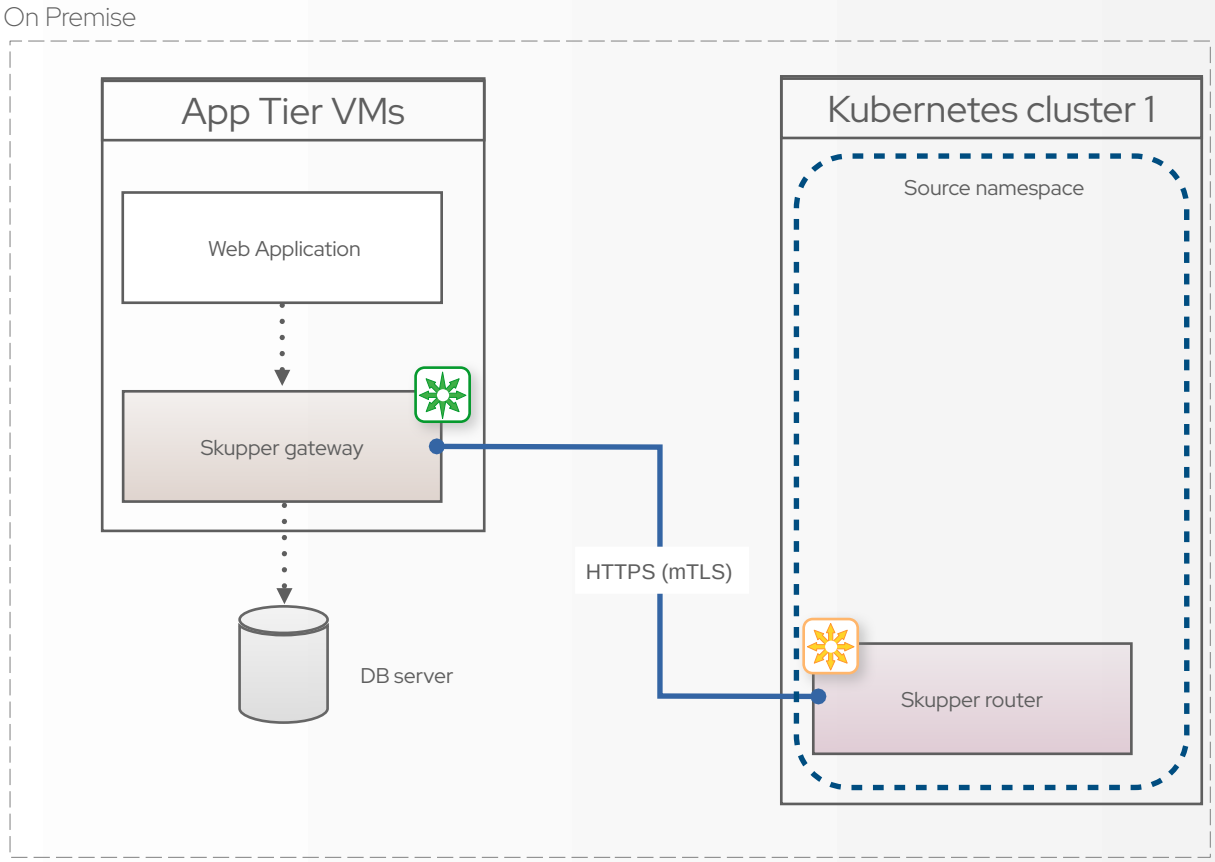
Back to the original problems

Dynamic migration (I)



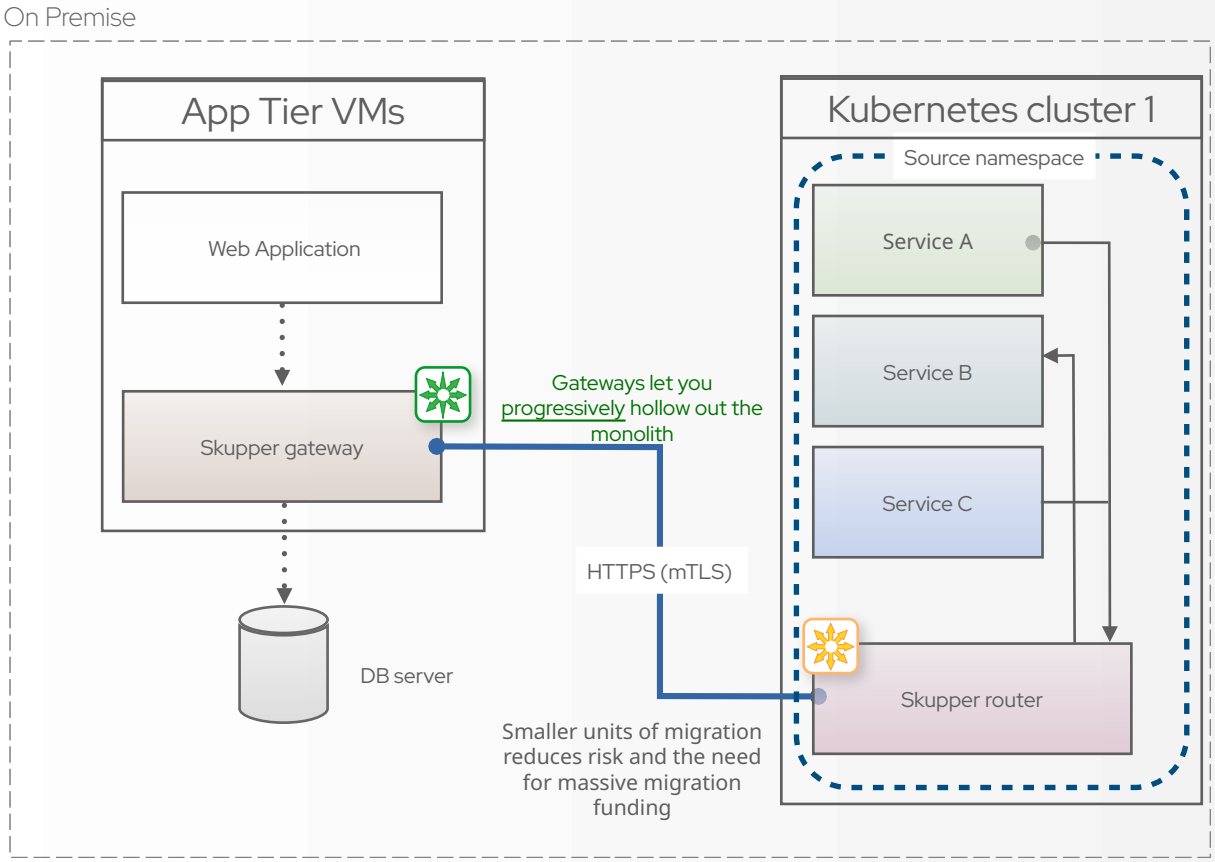
Skupper

Dynamic migration (II)



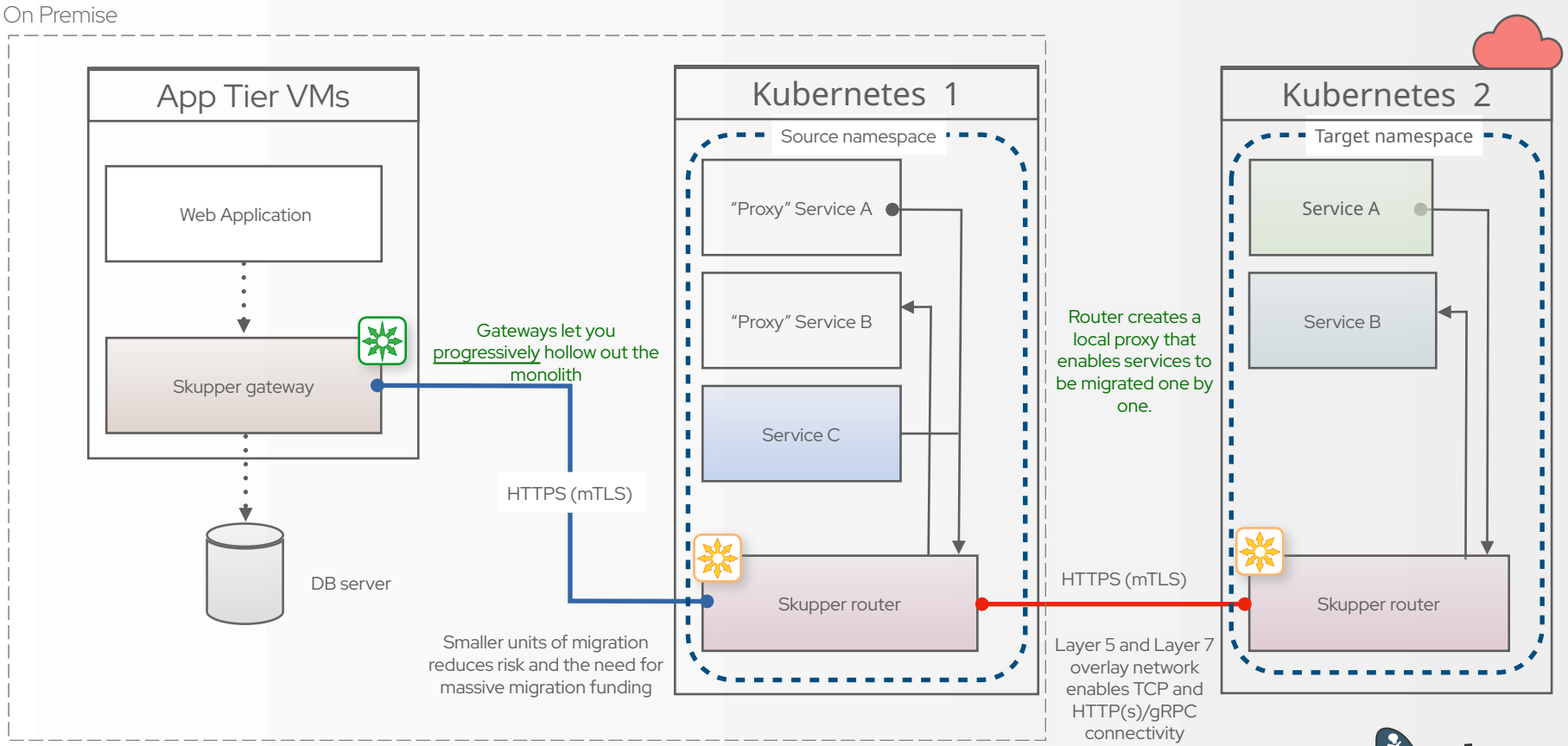
Skupper

Dynamic migration (III)



Skupper

Dynamic migration (IV)



Skupper

Objectives

Declarative Model

Multi-tenancy (Skupper-X project)

Thank you