KubeCon | CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Decentralized Routing for a Sharded Application on Service Mesh

Pankaj Sikka & Vinay Gonuguntla

Decentralized Routing for a Sharded Application on Service Mesh

KubeCon | CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

KubeCon | CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

October 24-28, 2021

Vinay Gonuguntla
Staff Software
Engineer
Intuit

Pankaj Sikka
Staff Software
Engineer
Intuit

Agenda

01 Service Mesh

What Servicemesh is and it's use-cases at Intuit

02 Routing for Sharded Applications

Sharded applications, routing on mesh

03 Design & Challenges

What we built , what we learnt

04 Demo

Decentralized routing demo

05 What's Next?

Future investment

Who we are - Intuit

End User Award in 2019 and 2022

End user of Cloud Native and mobile open source tech

Created, open-sourced, used, and maintained by Intuit

Intuit Scale

245+
Clusters

16000+
Namespaces

~77,320
Nodes

7000000+
PODS

2000+
Services

SERVICE MESH

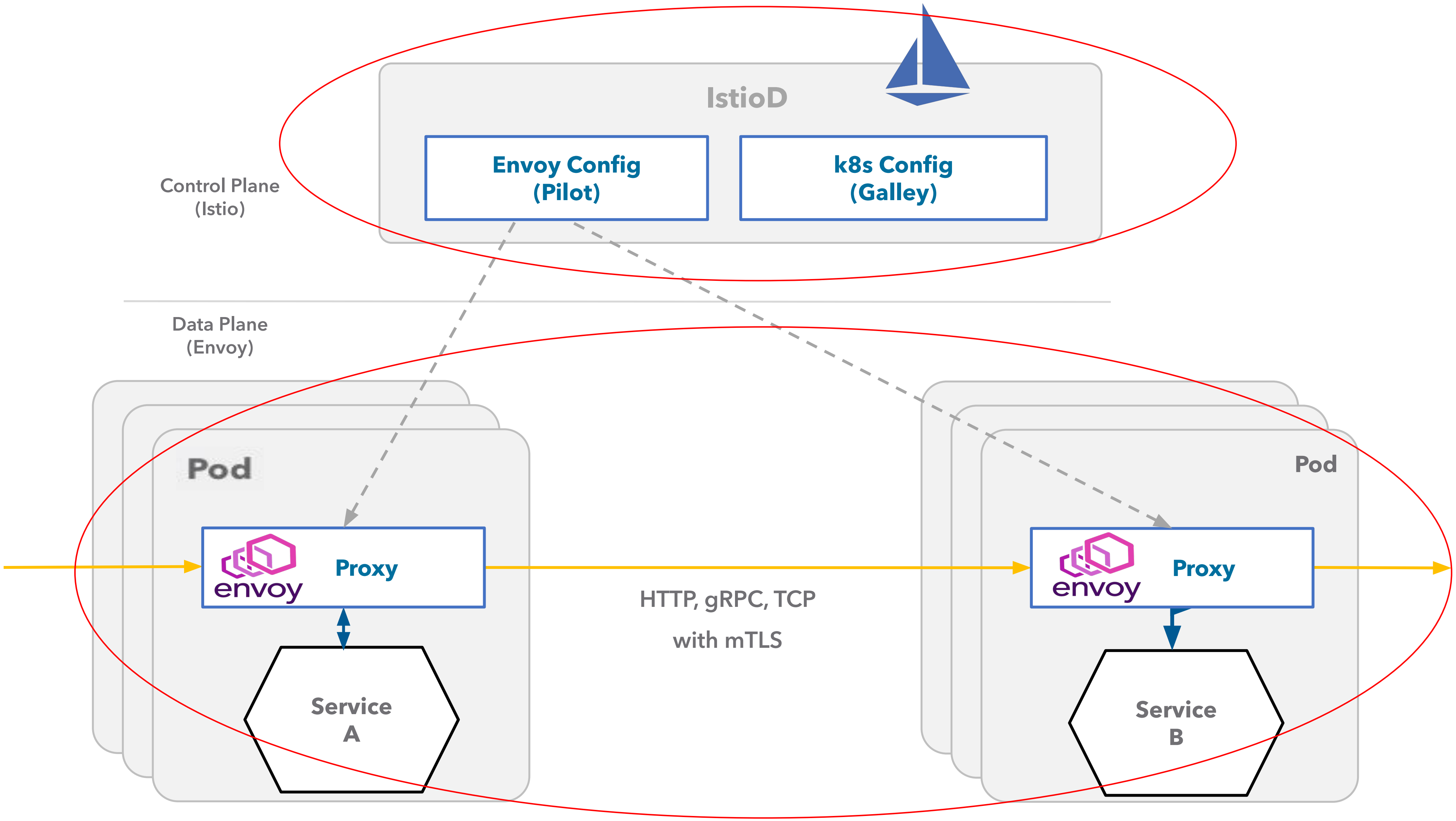
Service Mesh

- **Service mesh** - infrastructure layer to facilitate service-to-service communication

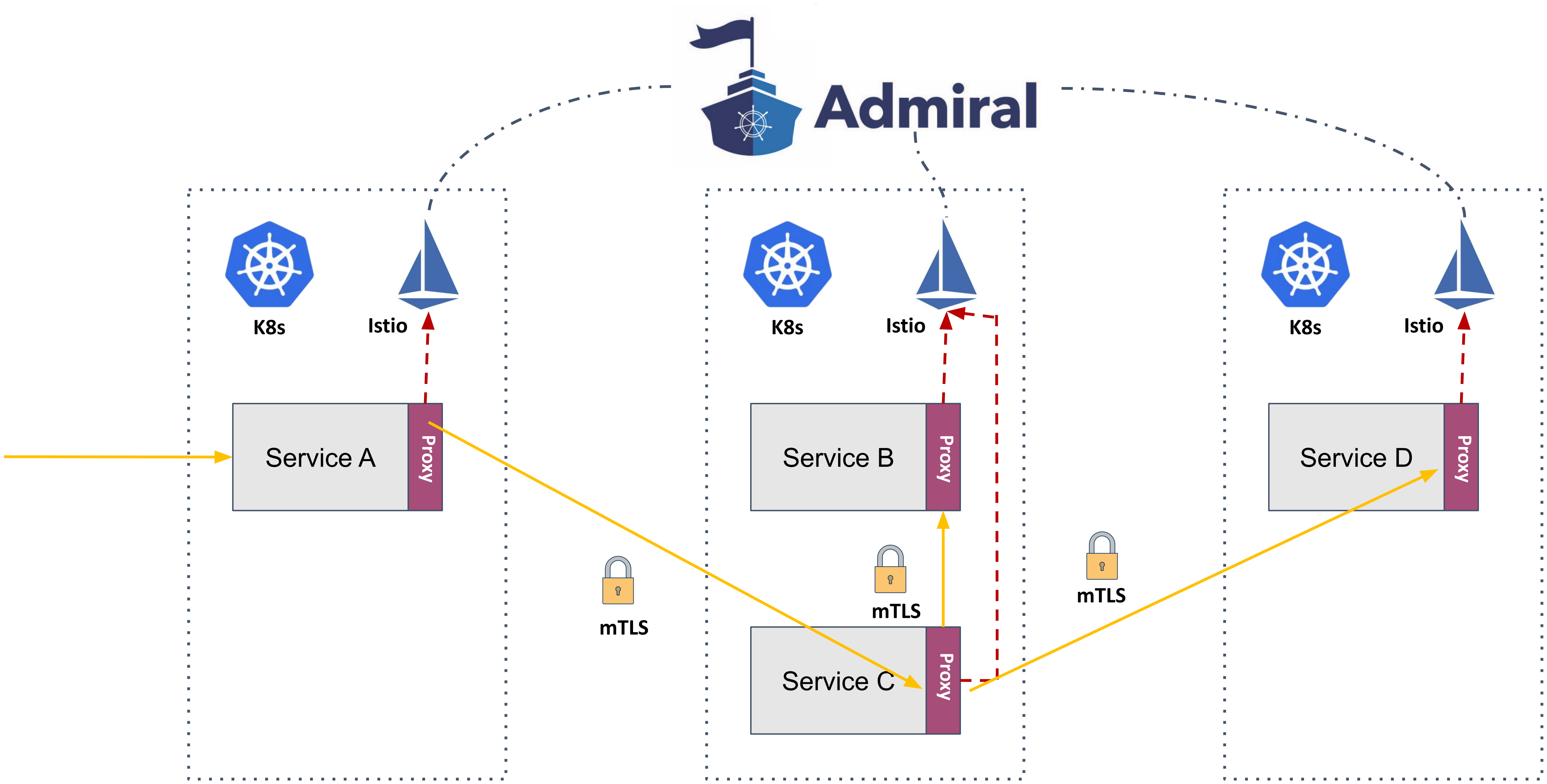
- Security, Observability, Routing and Access Control

- Examples - Istio, Linkerd, Consul

Service Mesh



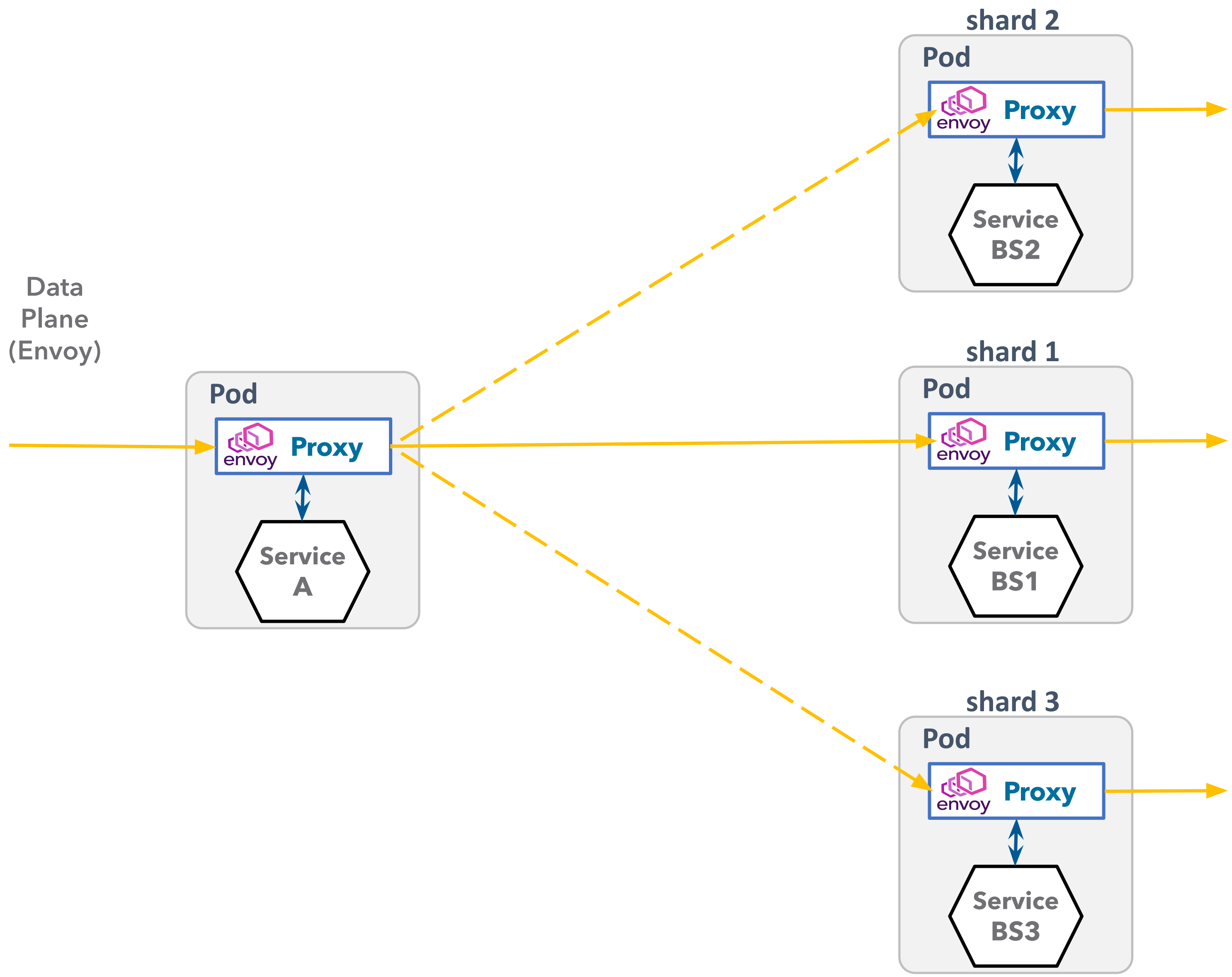
Service Mesh at Intuit



ROUTING FOR SHARDED APPLICATIONS

What is a sharded application ?

Sharded Routing - MESH

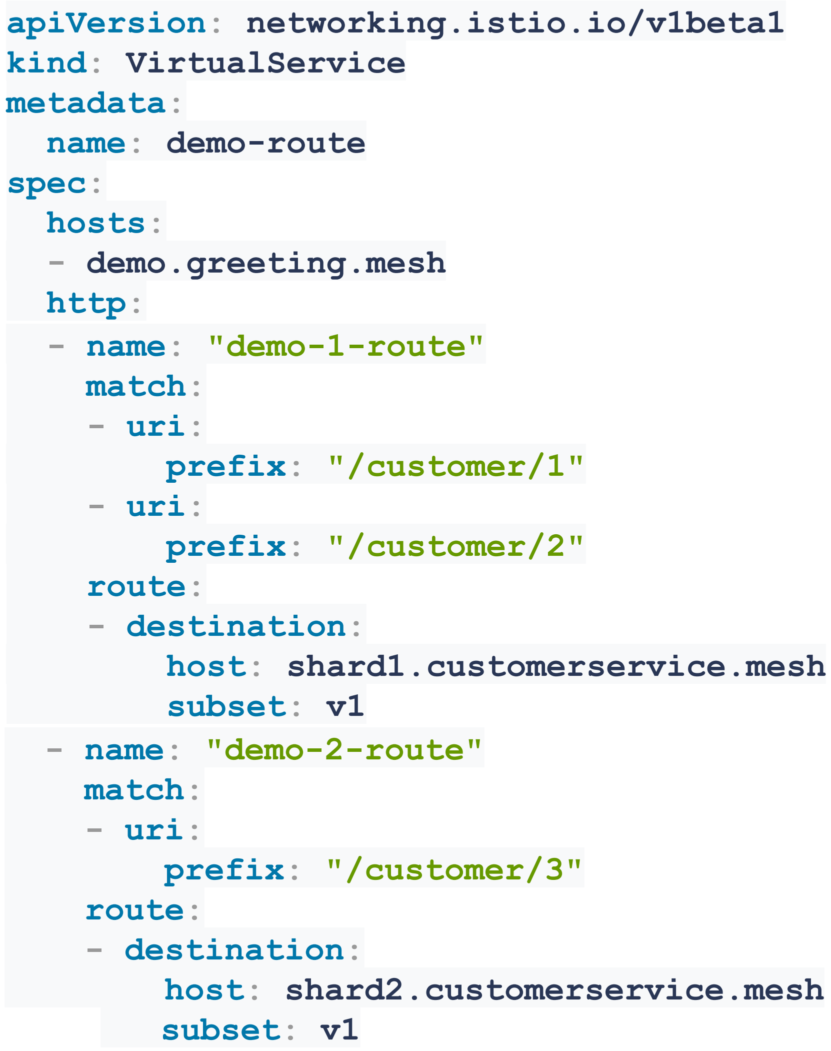


Sharded Routing - Goals

- Client services need to have minimal/no changes
- Scale for millions of users
- Support moving data among shards
- Service owners should control the routing configuration

Sharded Routing - Existing solutions

Istio VirtualService

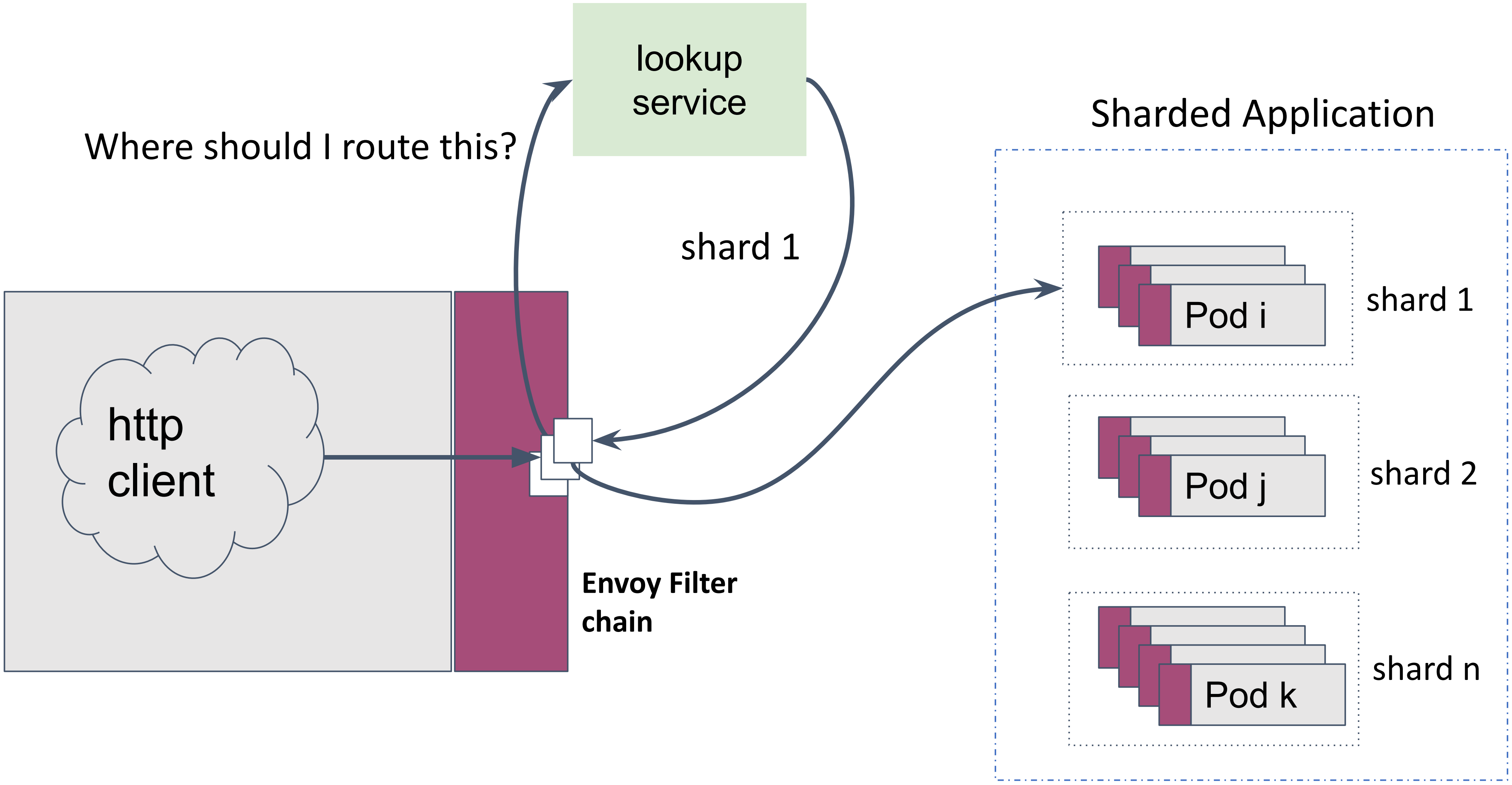


```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: demo-route
spec:
  hosts:
  - demo.greeting.mesh
  http:
  - name: "demo-1-route"
    match:
    - uri:
        prefix: "/customer/1"
    - uri:
        prefix: "/customer/2"
    route:
    - destination:
        host: shard1.customerservice.mesh
        subset: v1
  - name: "demo-2-route"
    match:
    - uri:
        prefix: "/customer/3"
    route:
    - destination:
        host: shard2.customerservice.mesh
        subset: v1
```

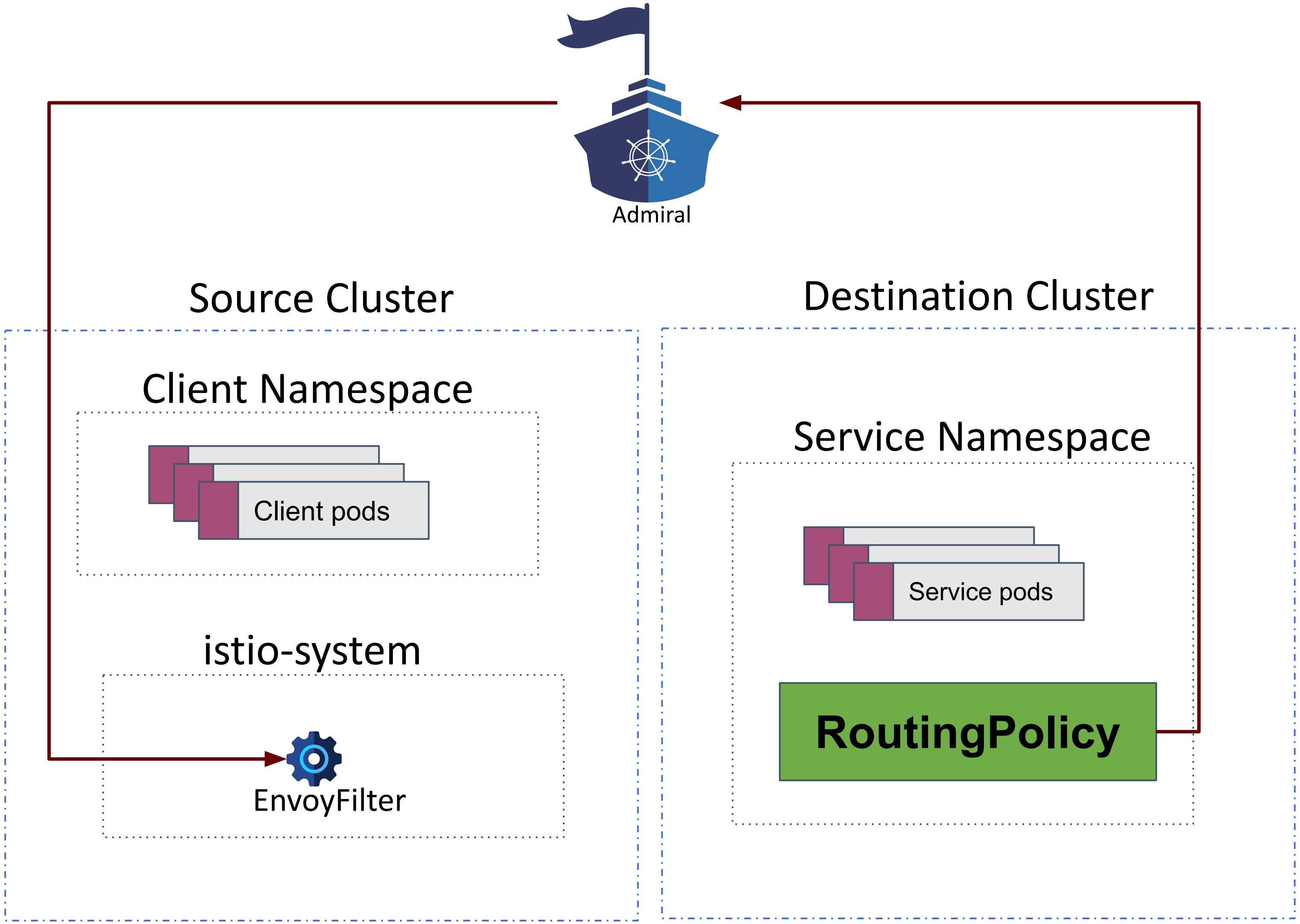
Limitations

- Millions of records, in a single file
- Not scalable due to size constraints
- Near real time update, very challenging
- Retry on moved shards not possible

Sharded Routing - Approach



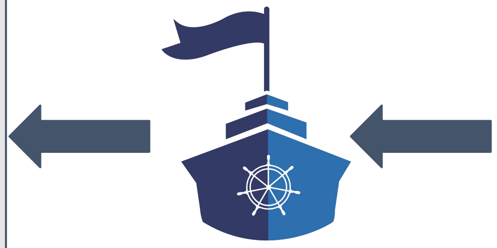
Dynamic Routing in Intuit



Dynamic Routing in Intuit

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
...
spec:
  configPatches:
  - applyTo: HTTP_FILTER
    match:
      context: SIDECAR_OUTBOUND
...
         config:
              configuration:
                '@type':
type.googleapis.com/google.protobuf.StringValue
       value: |-
         routingAlgorithm: ShardedRouting
         cachePrefix: prefix-v1-a
         cacheTtlSec: "3600"
         lookupUrl: http://demo.lookup.mesh/
         env: stage
         wasmPath:/etc/istio/dynamicrouter.wasm
        hosts: demo.greeting.mesh
         plugin: demo

...
```



```
apiVersion: admiral.io/v1alpha1
kind: RoutingPolicy
metadata:
  annotations:
    admiral.io/env: stage
...
  name: demo-routing-policy
spec:
  plugin: demo
  hosts:
    - demo.greeting.mesh
  config:
    routingAlgorithm: ShardedRouting
    cachePrefix: prefix-v1-a
    cacheTtlSec: "3600"
    lookupUrl: http://demo.lookup.mesh/
    env: stage
    wasmPath: /etc/istio/dynamicrouter.wasm
```

Challenges

1 Workload selector limitations

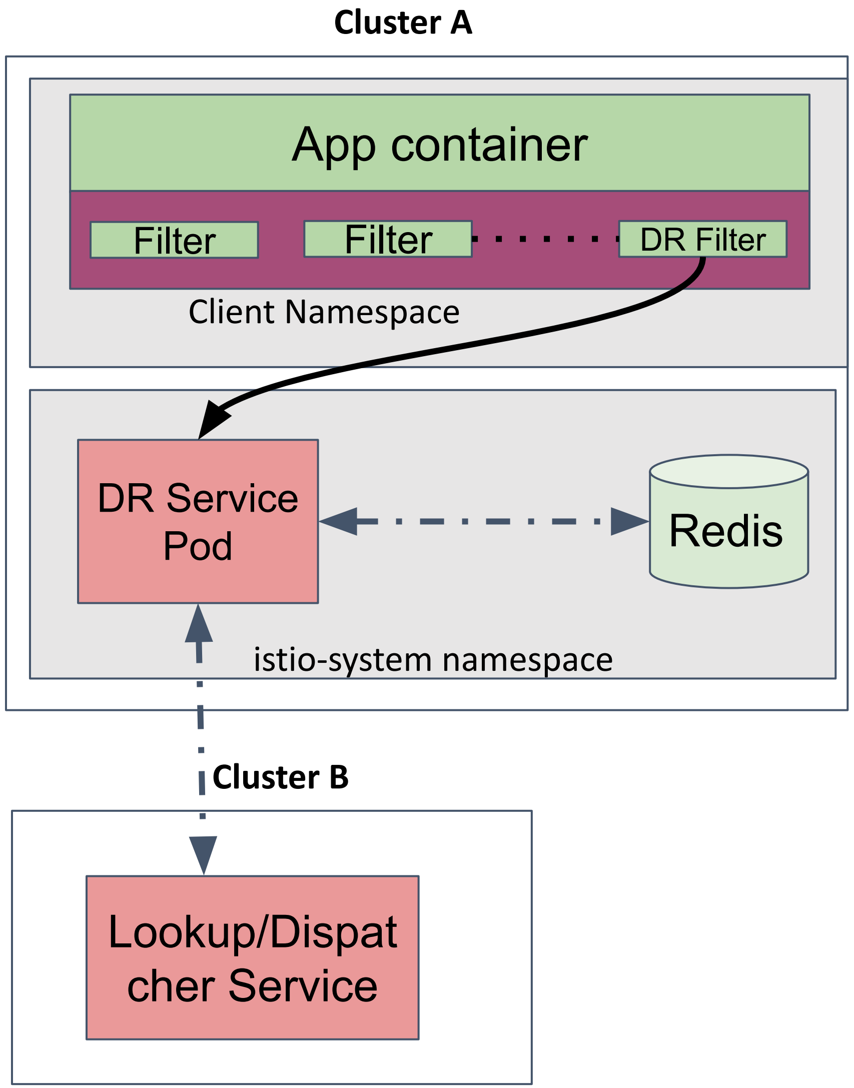
OR operation in workload selector is not supported

2 TinyGo limitations

The need for external lookup service

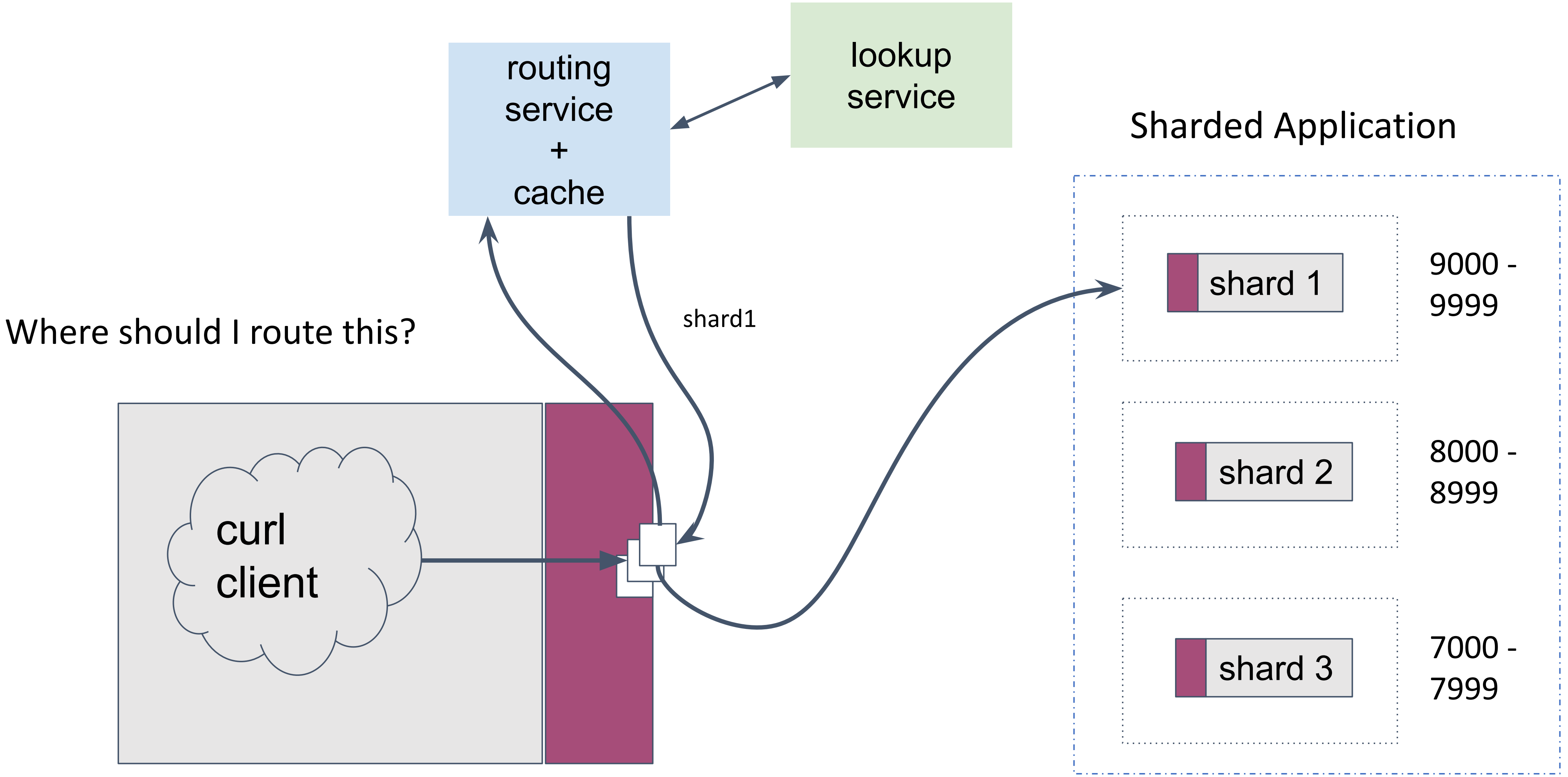
3 Pre-built proxy image

Move to dynamically loading the business logic



DEMO

Demo Setup


routing service + cache
lookup service
Sharded Application
shard1
Where should I route this?
curl client
shard 1
9000 - 9999
shard 2
8000 - 8999
shard 3
7000 - 7999

Future Work

- Adding rate limiting for services using this approach
- Work with Istio community to allow workload selector to be applied to multiple pods
- Explore using C++ and rust

Q/A

Please scan the QR Code above to leave feedback on this session