eBay Ingress based on Kubernetes

fmeng@ebay.com

Internet Application Evolution

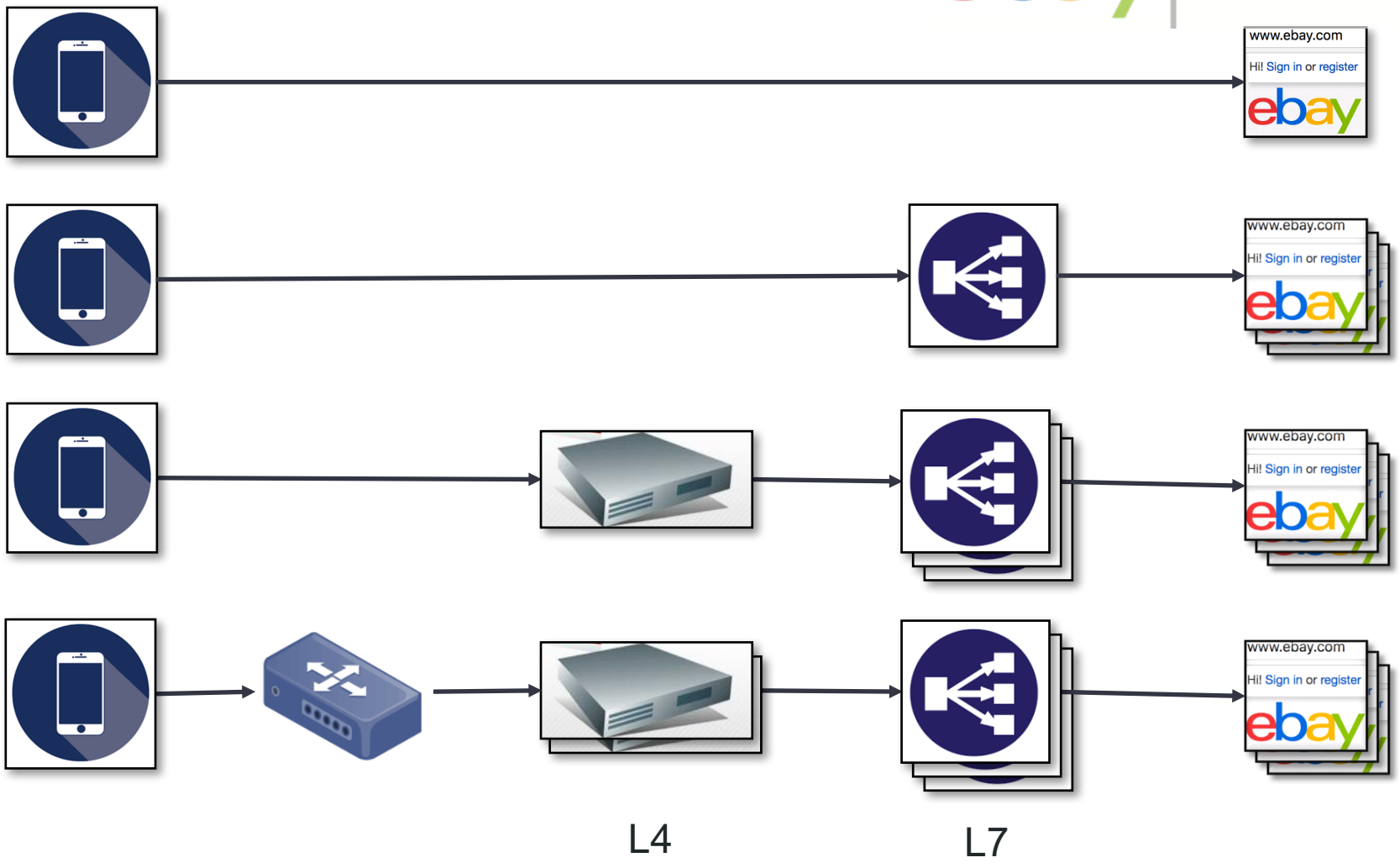
Single Webpage

LoadBalacer and HA

L4/L7 Loadbalancer

Multiple L4

Multiple Datacenters …



Agenda

Ingress Introduction

eBay Ingress

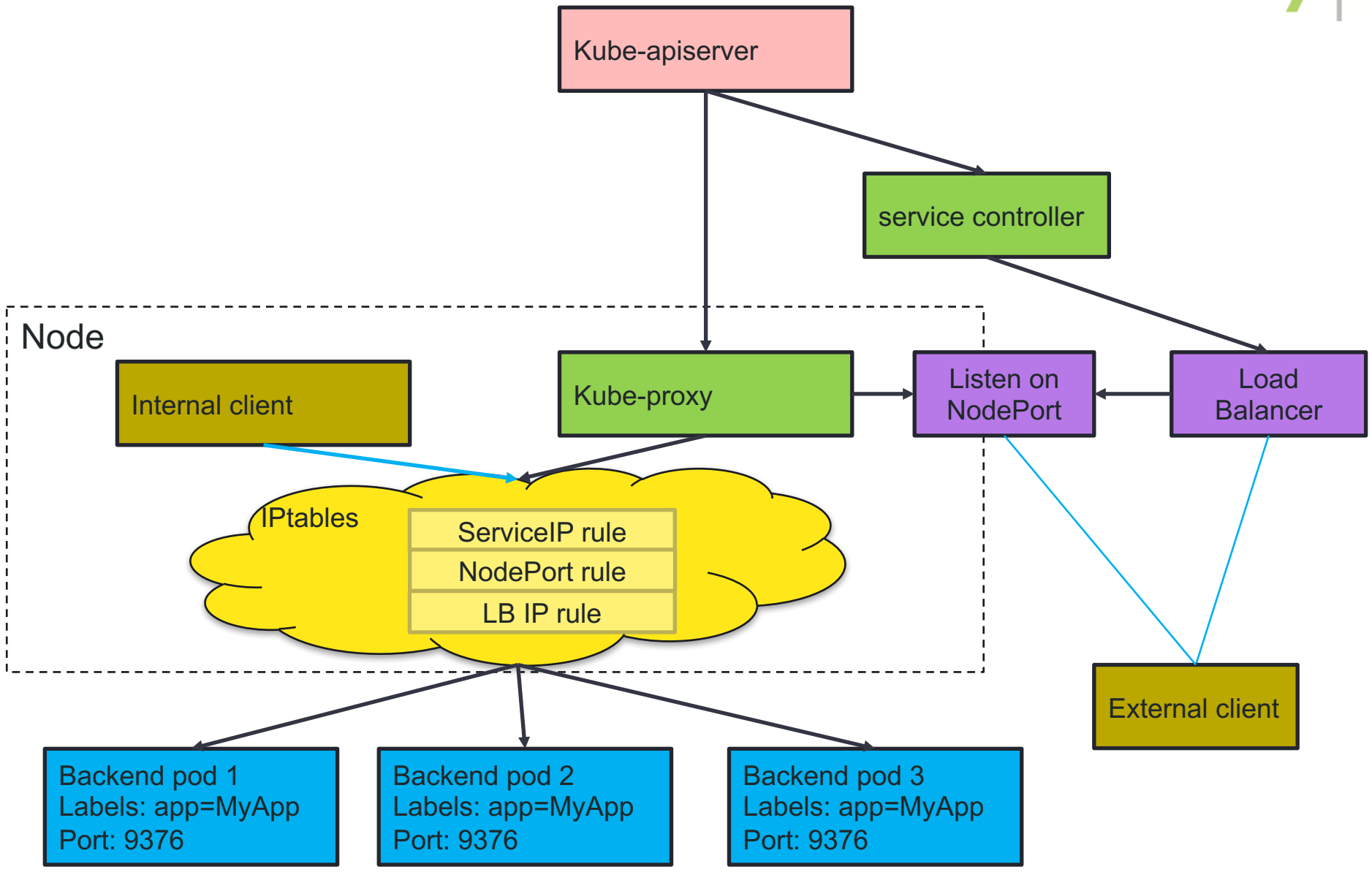
eBay Ingress Controller Based on kubernetes

What is Ingress

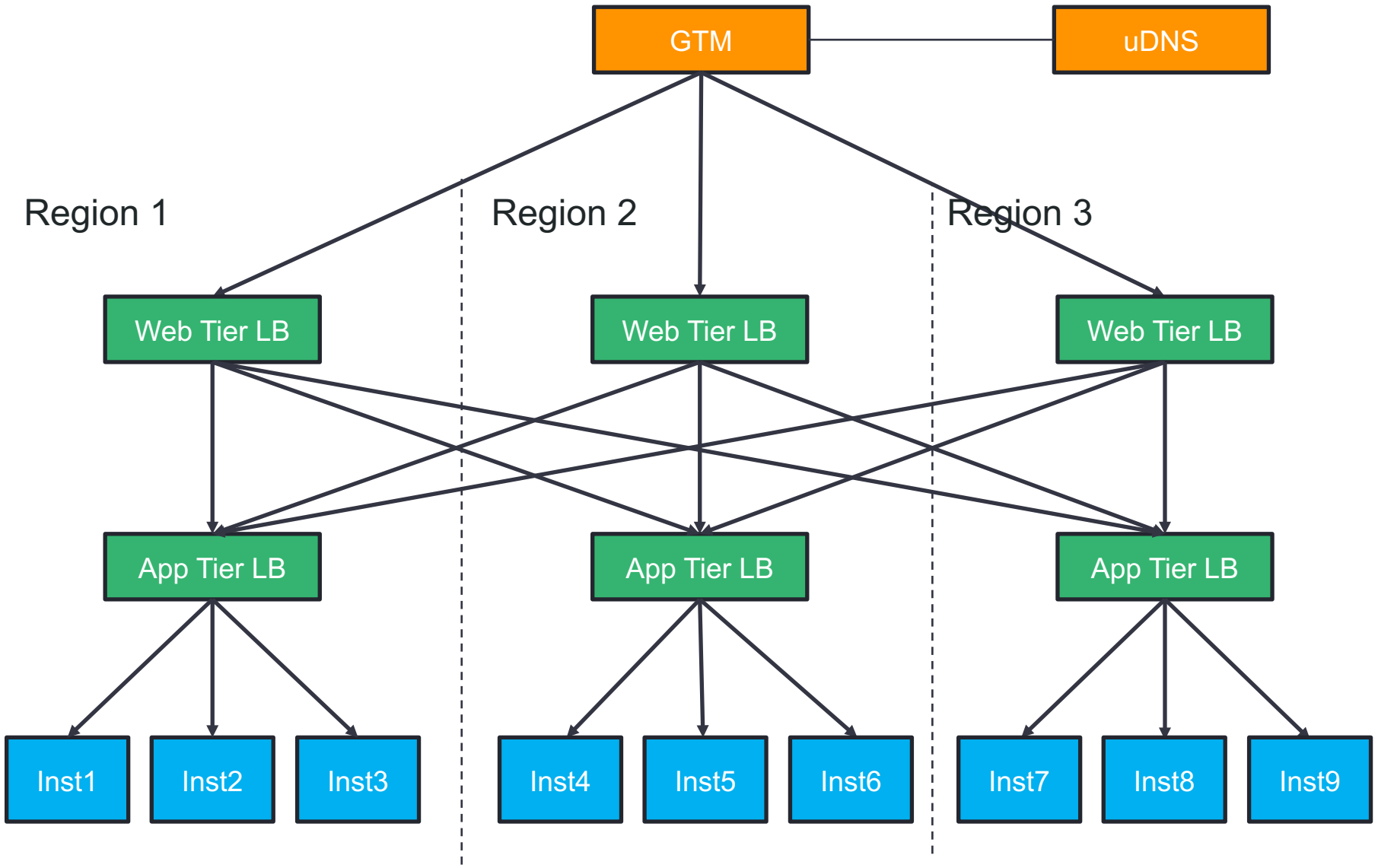
in·gress /ˈinˌgres/ the action or fact of going in or entering.

- **Kubernetes Definition for Ingress**
  - An Ingress is a collection of rules that allow inbound connections to reach the cluster services.
  - An Ingress controller is responsible for fulfilling the Ingress, usually with a loadbalancer, though it may also configure your edge router or additional frontends to help handle the traffic in an HA manner.

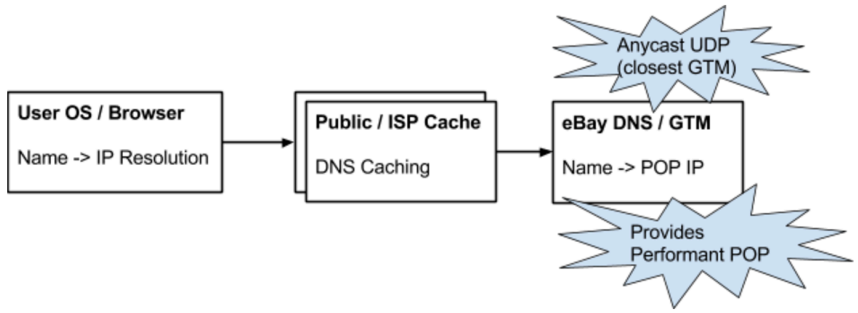
How Kubernetes exposes a service

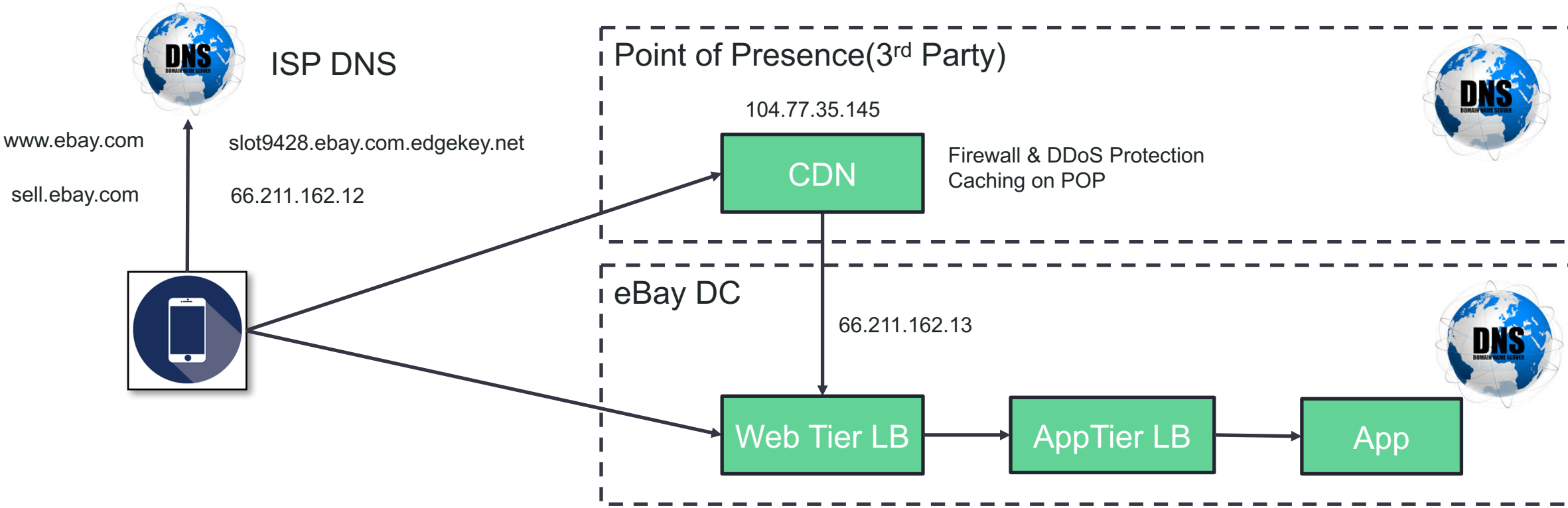


What it takes to deploy a prod workload (in terms of ingress)

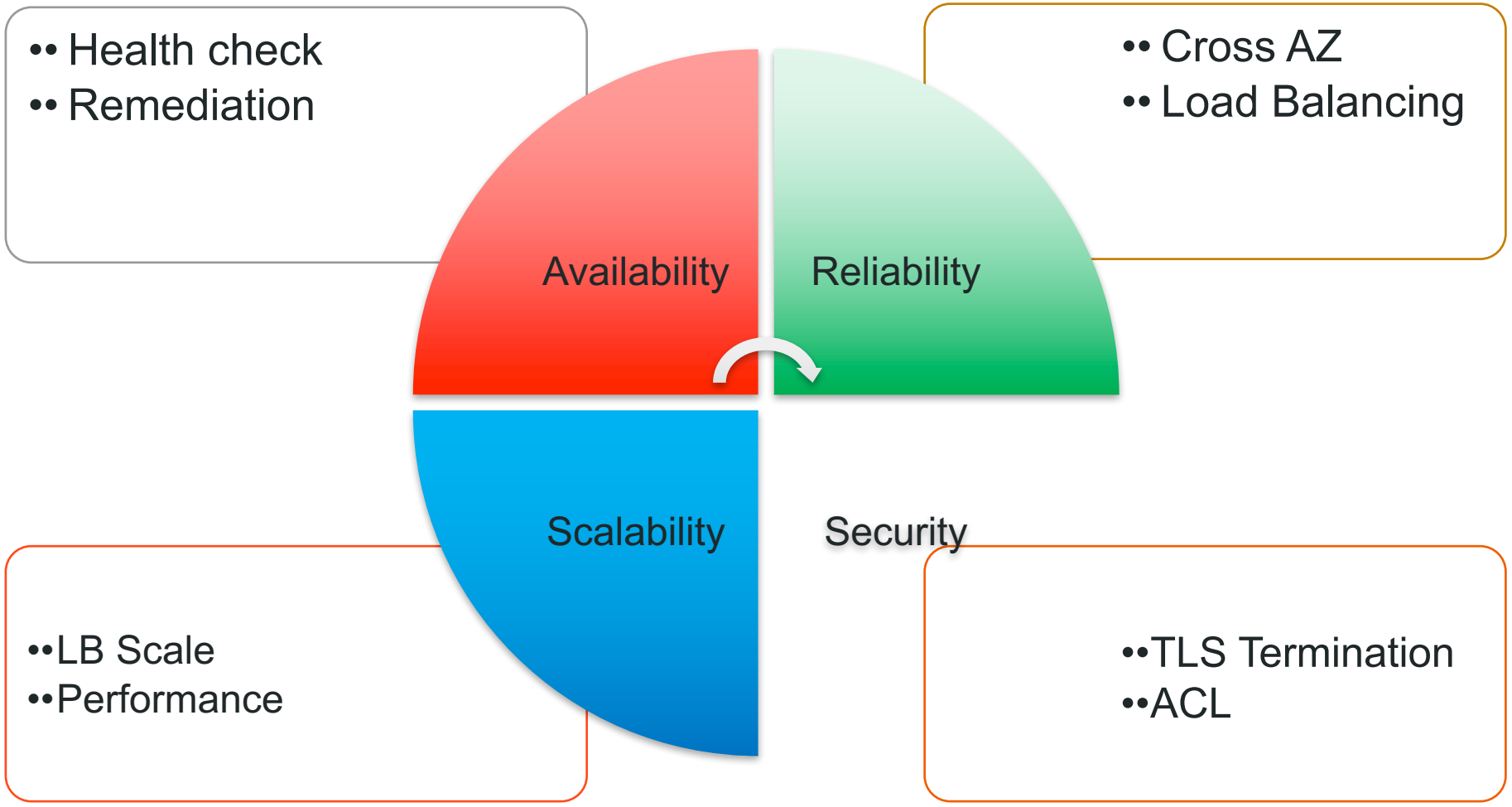


ebay.com current ingress Topology





What an modern ingress should consider



eBay Ingress requirements highlight

- Use ingress instead of service to save vips
- Support existing topologies
- Abilities to setup out-of-box POP
  - Minimize the connection establishment time
  - Terminate TLS on Edge POP
  - Define external service on POP which pointing to eBay DC applications
- Minimize the use of third party black box devices
  - SLB based solution(envoy)
  - L7 Rules support

An upstream Ingress Spec Sample

1. Create certificates

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /tmp/tls.key -out /tmp/tls.crt -subj "/CN=echoheaders/O=echoheaders"

2. Create secrets for the certs

```
echo "
apiVersion: v1
kind: Secret
metadata:
  name: nginx-ing-secret
data:
  tls.crt: `base64 /tmp/tls.crt`
  tls.key: `base64 /tmp/tls.key`
" | kubectl create -f -
```

3. Create ingress with certs

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ing1
spec:
  tls:
  - secretName: nginx-ing-secret
  rules:
  - host: ing1.vip.qa.ebay.com
    http:
      paths:
      - path : /foo
        backend:
          serviceName: nginxservice
          servicePort: 80
```

TLS certs

Backend service

An eBay Ingress Spec Sample

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ing1
  namespace: fmeng
  annotations:
    "gtm.ingress.tess.io/ing1.vip.qa.ebay.com": "tess-ing1.g.vip.ebay.com"
    "ingress.tess.io/routing_type": "web”
    "ingress.tess.io/backend_loadbalancing": ”app"
spec:
  tls:
  - secretName: nginx-ing-secret
  rules:
  - host: ing1.vip.qa.ebay.com
    http:
      paths:
      - path : /foo
        backend:
          serviceName: nginxservice
          servicePort: 80
```

Namespace owner

dns name in GTM

Ingress type

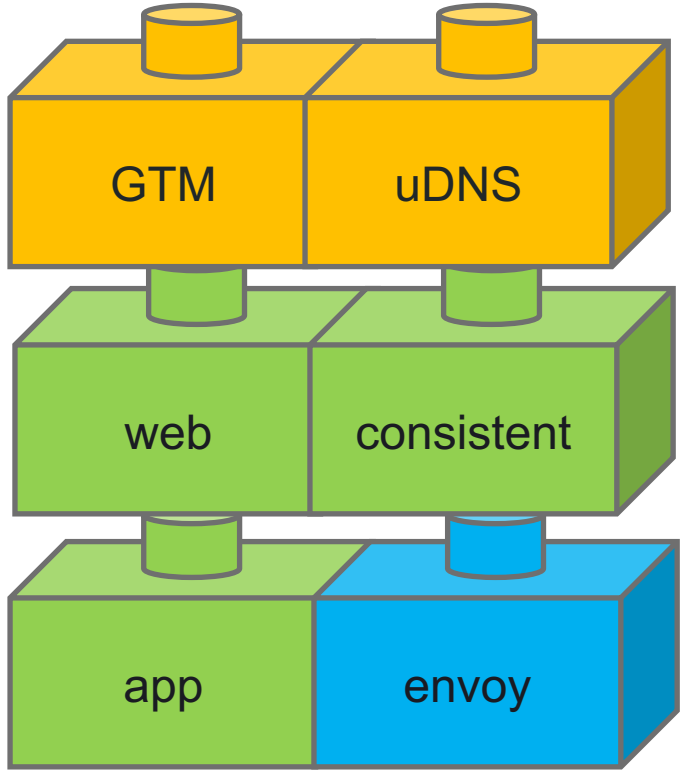
cname in uDNS

Build ingress LEGO blocks

ingress.tess.io/dns_scope

ingress.tess.io/backend_loadbalancing

ingress.tess.io/routing_type



Ingress Controller

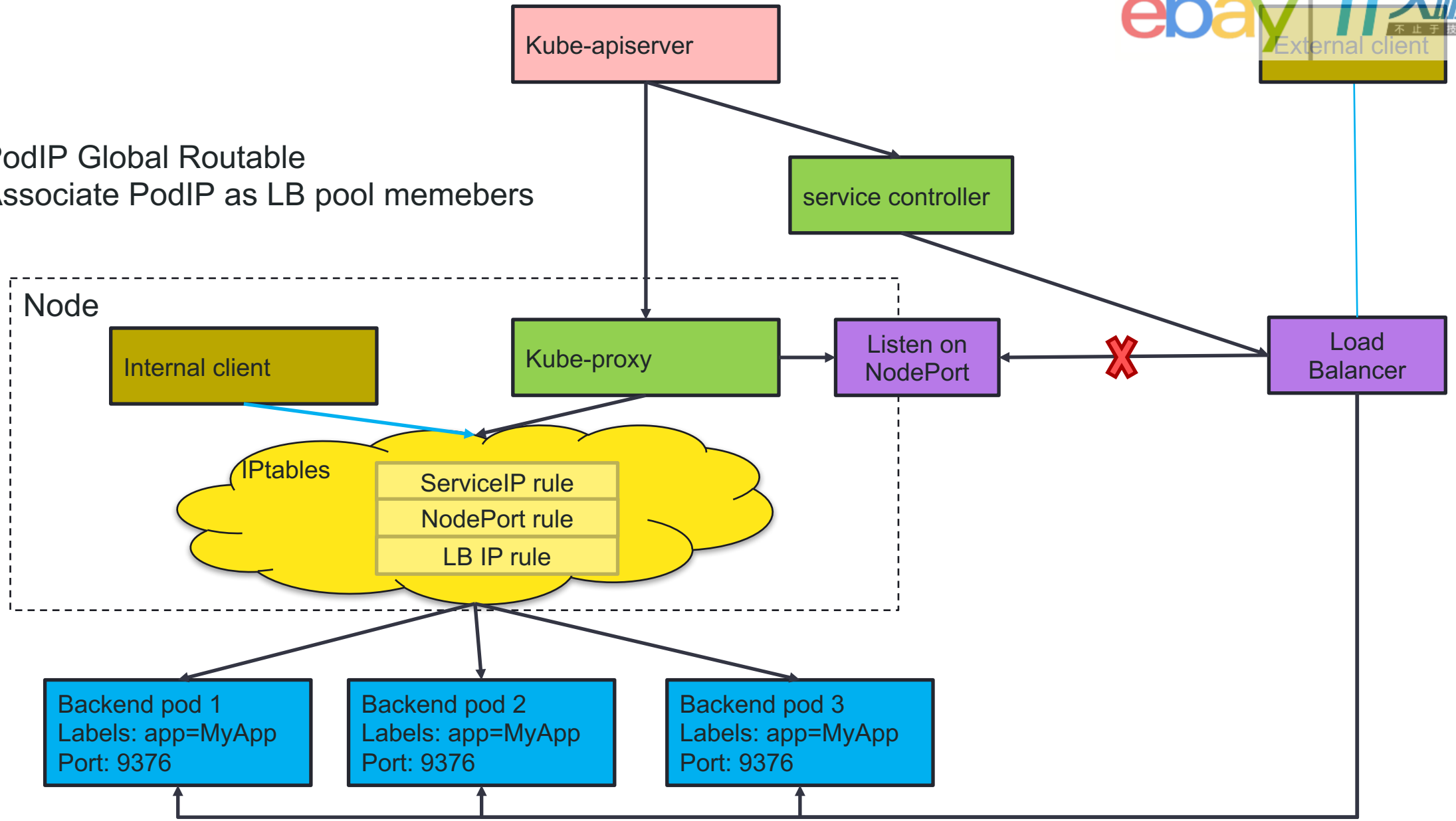
DNS Provider L4 Provider L7 Provider

Ingress Controller

- Multiple Plugins to configure L4 LoadBalancer
  - LBaaS – app tier LB configuration
  - LBMS – web tier LB configuration
  - Consistent – IPVS director configuration
- Proxy Provider
  - Envoy
  - Nginx(future)
  - LBMS API(future)
- Make Ingress Controller an orchestrator to build the two layer topology.
  - Web + App
  - Web + Envoy
  - IPVS + Envoy

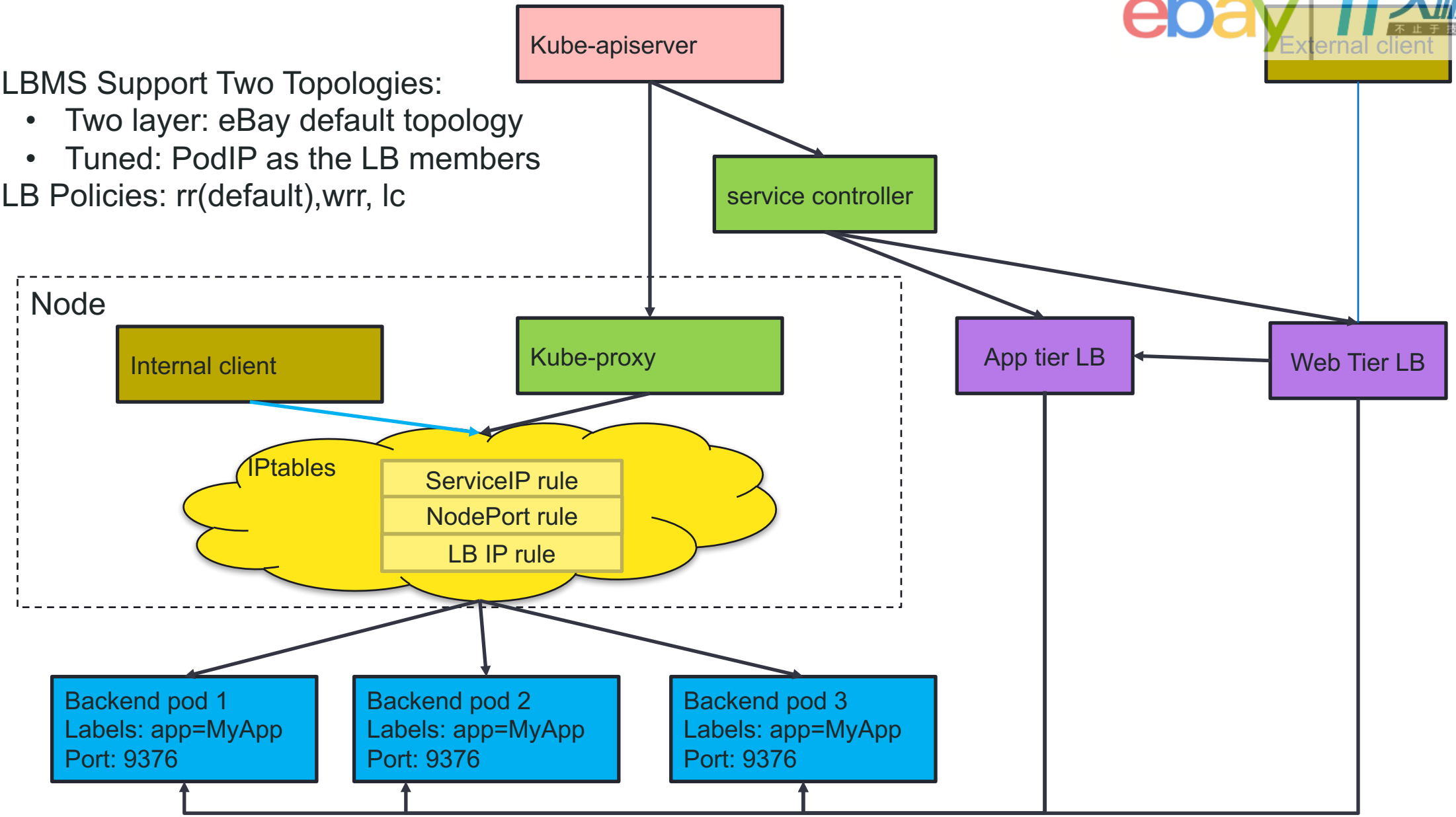
Service Controller - Enhanced LBaaS provider

- PodIP Global Routable
- Associate PodIP as LB pool memebers



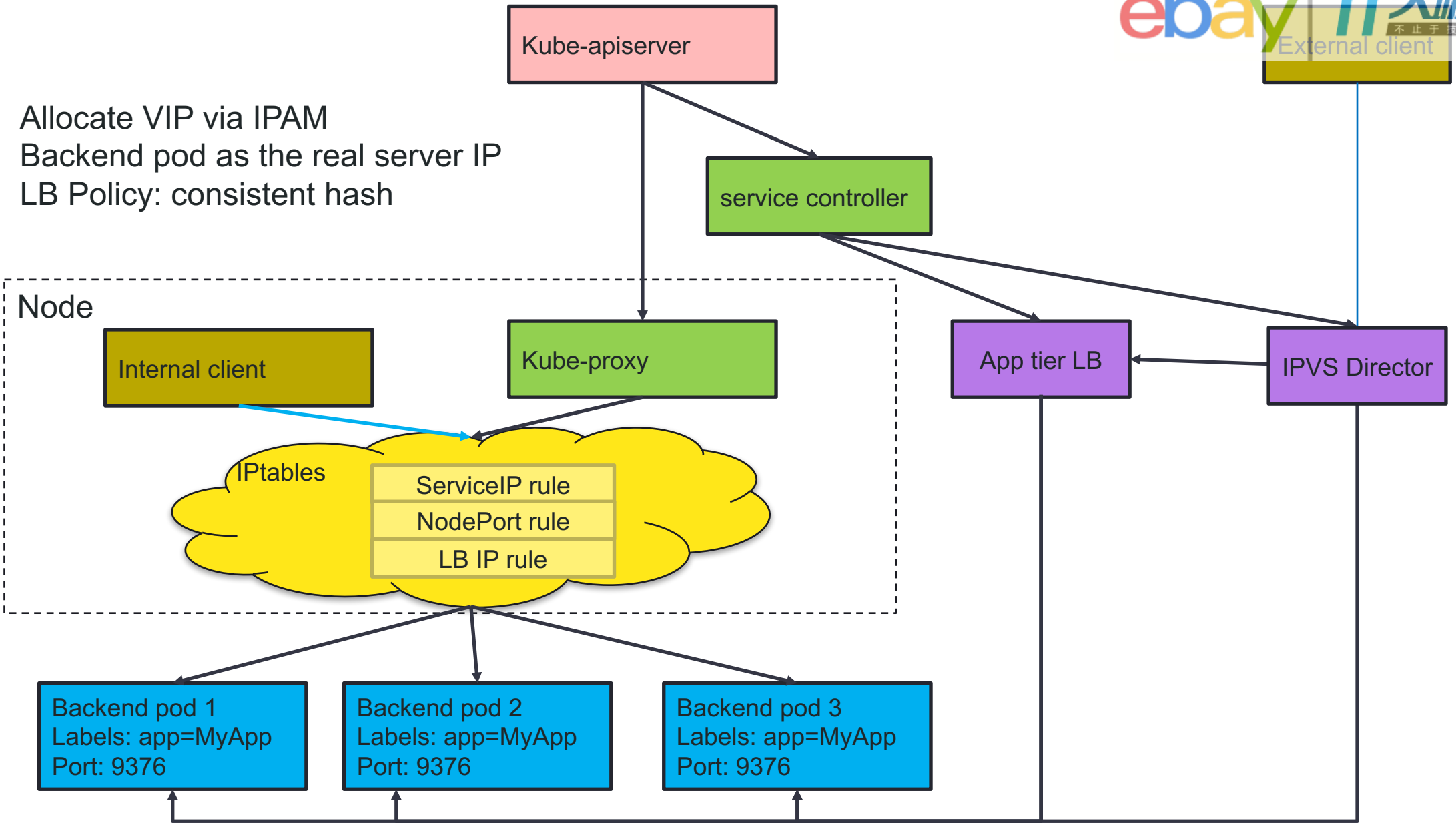
Service Controller – LBMS Provider

- LBMS Support Two Topologies:
  - Two layer: eBay default topology
  - Tuned: PodIP as the LB members
- LB Policies: rr(default),wrr, lc

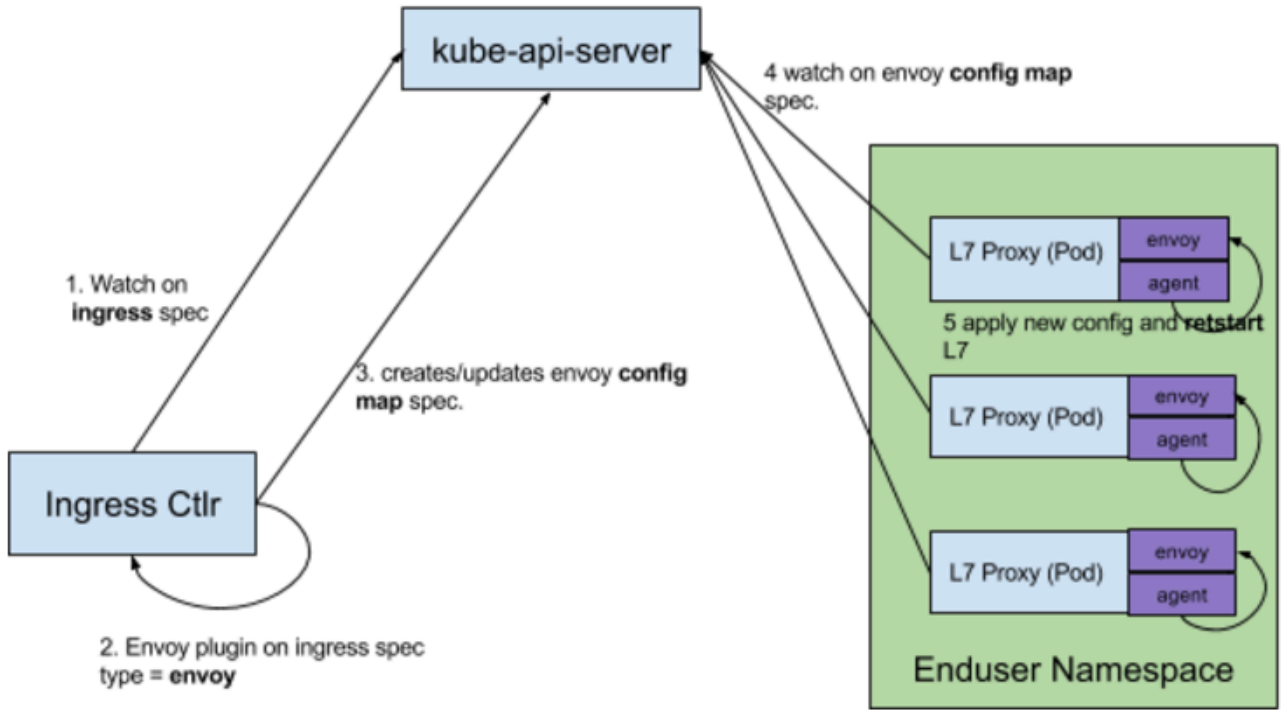


Service Controller – IPVS Provider

- Allocate VIP via IPAM
- Backend pod as the real server IP
- LB Policy: consistent hash



Build Proxy Provider for Envoy



- Dedicated L7 pods to single host
- Proxy Provider creates envoy config as configmap
- Envoy agent on L7 pod watch configmap and reload config

Envoy config sample

```
{
  "route_config": {
    "virtual_hosts": [
      {
        "name": "bar.baz.com",
        "domains": [
          "bar.baz.com"
        ],
        "routes": [
          {
            "prefix": "/rservice/1",
            "cluster": "service1"
          },
          {
            "prefix": "/rservice/2",
            "cluster": "service2"
          }
        ]
      }
    ]
  }
}
```

SLB Evaluation

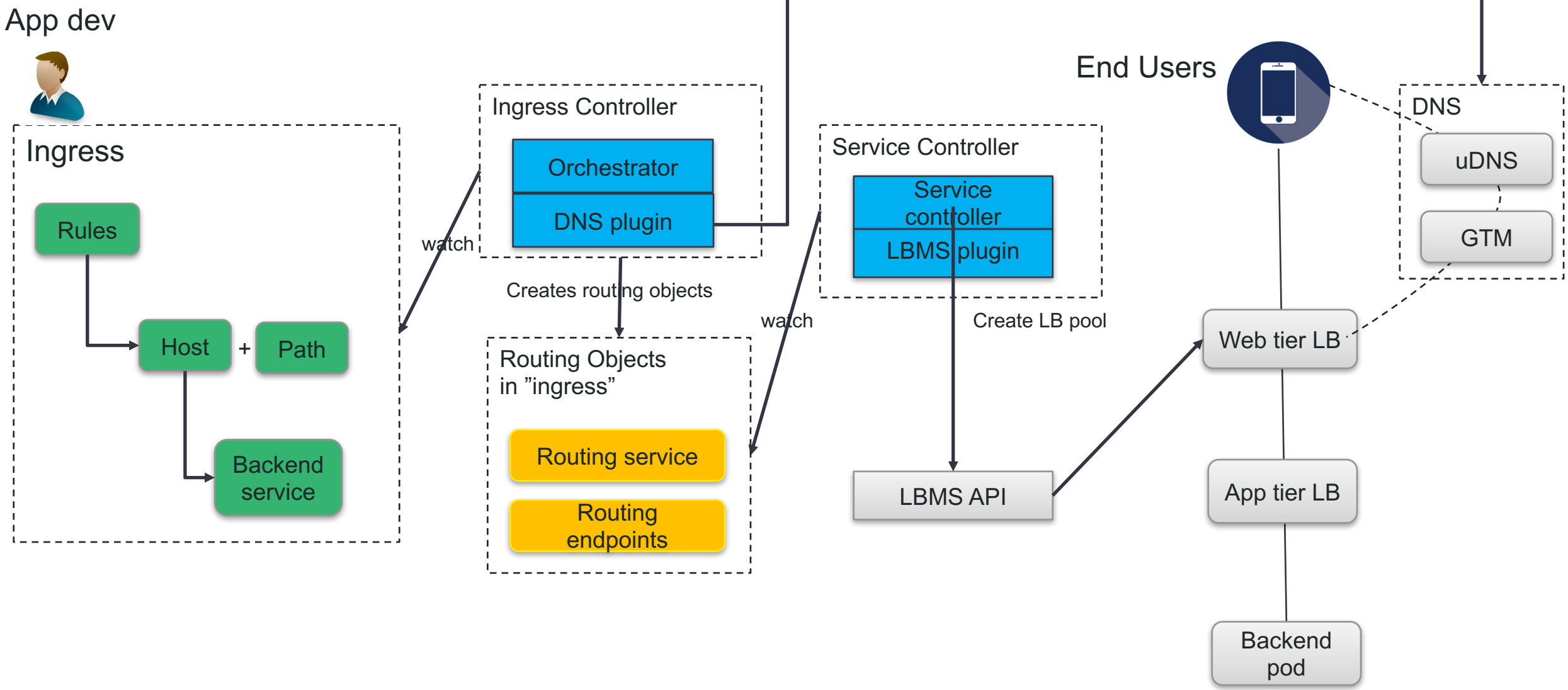
L4 -> IPVS

Feature \ Solution	IPVS based	Netfilter based
Core Requirements		
Consistent Hashing Support	M	L
DSR	H	H
Operability		
Connection Draining	H	L
DDoS Defense	M	L
Performance		
Misc		

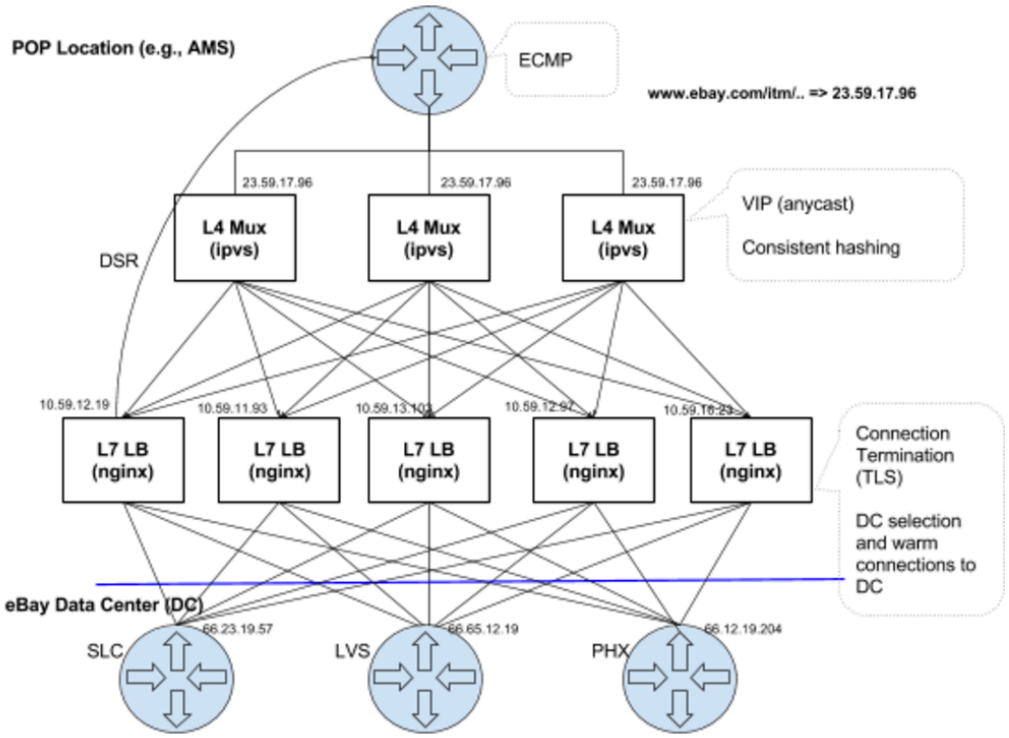
L7 -> Envoy

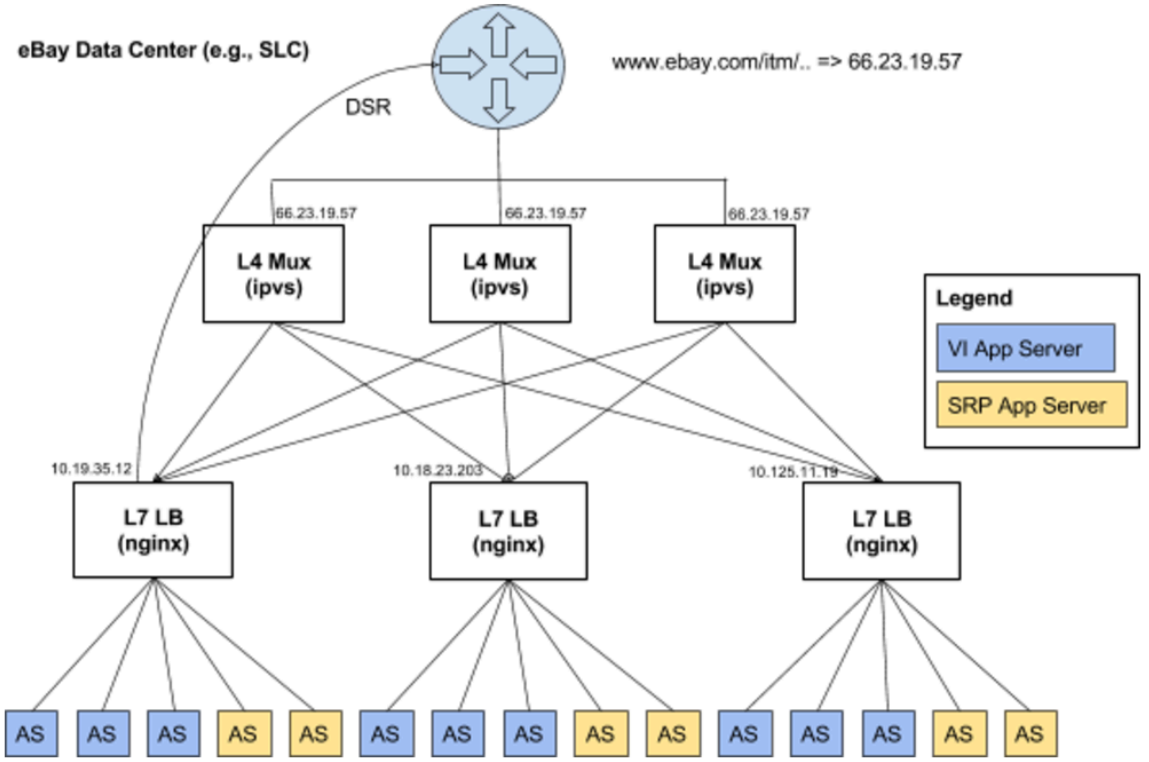
Feature\Product	Envoy	Ngnix	HA Proxy
Core Features			
HTTP2	✔	✔	✘
L7 Rate Limiting	✔	✔	✔
Availability			
Operability - Deployment			
Connection draining	✔	✘	✔
Operability - Monitoring			
Code Manageability			
Optional Features			

eBay Ingress behavior for default topology



eBay future ingress topology





ebay | IT大咖说 不止于技术

Thank You