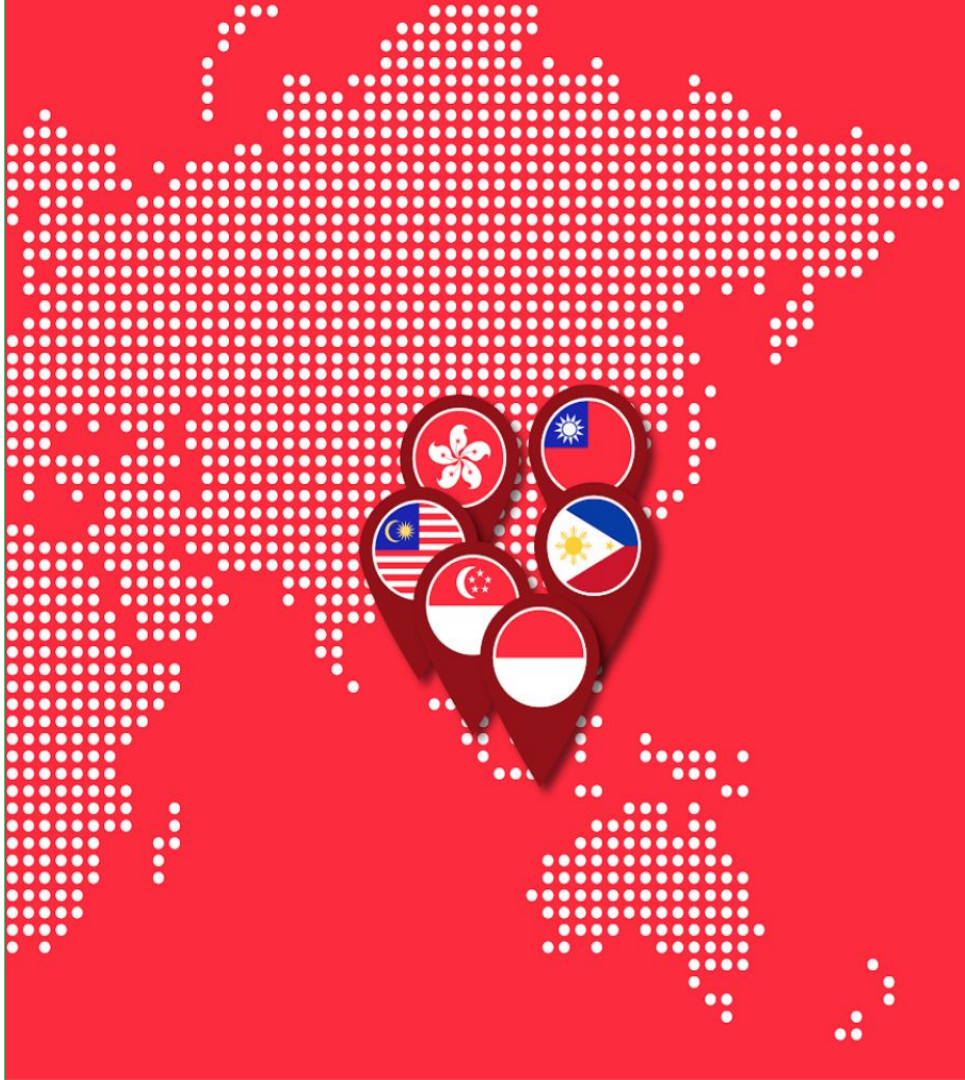
Moving large scale consumer e-commerce Infrastructure to Mesh

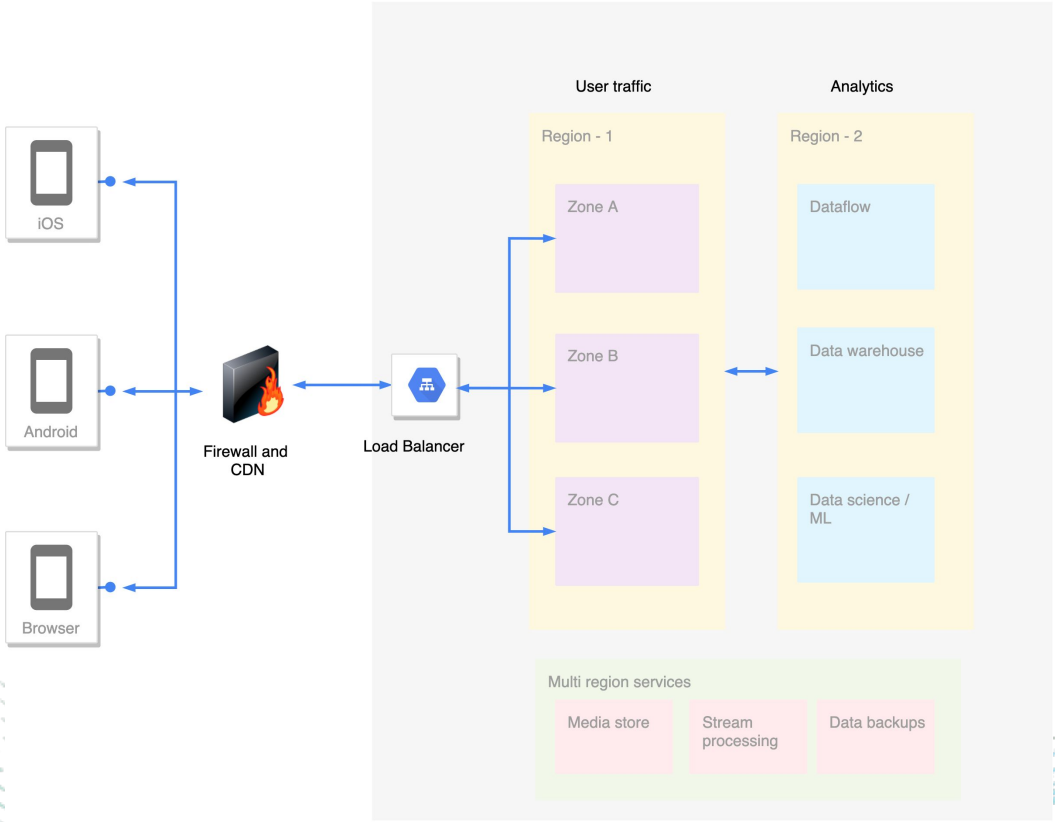
Rajath Ramesh Principal Software Engineer @Carousell
Harshad Rotithor Software Architect @Carousell

About Carousell

- C2C Marketplace in SEA
- Over 4 million monthly active users
- User requests over 10 billion per month
- Internet egress bandwidth over 100 TB/month
- Internal egress bandwidth ~2 PB/month

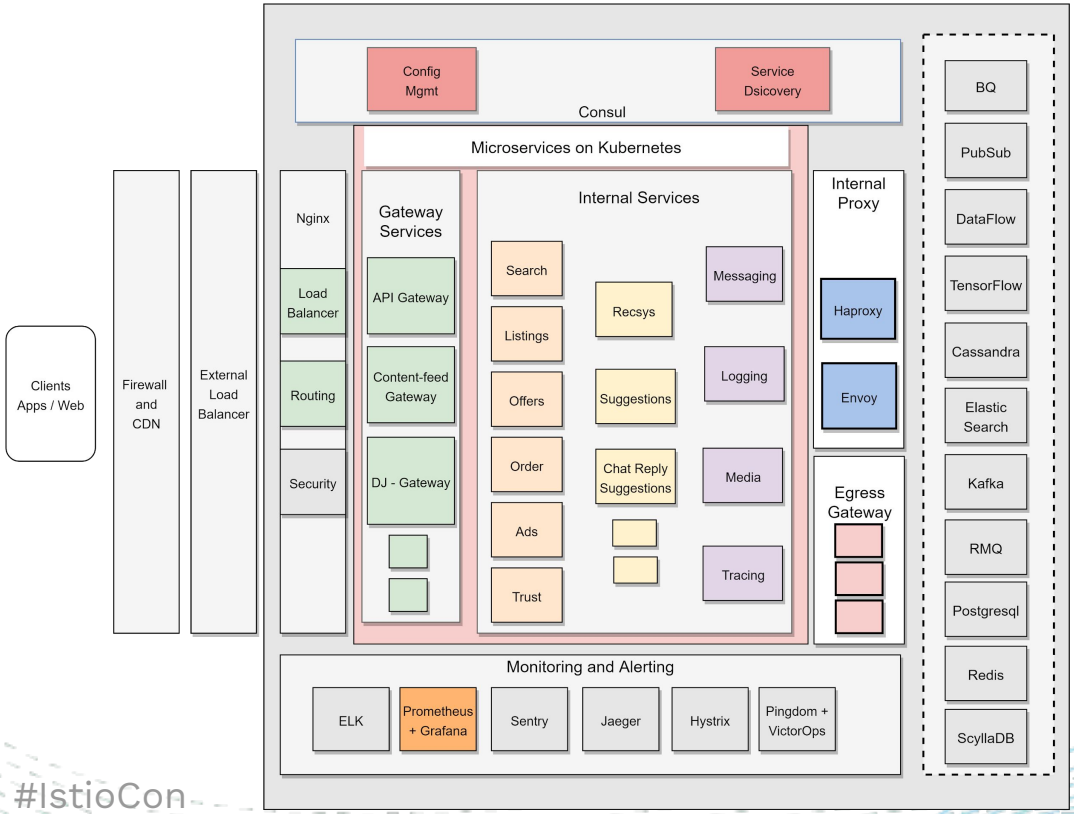


Architecture Overview



- User traffic infrastructure - TW region, all 3 zones
- REST APIs for client traffic
- gRPC for inter-service traffic
- Around 100+ microservices
- Majority of services written in Go

Architecture Overview - Discovery and Routing



- Service Discovery and Configuration using Consul
- HTTP/TCP traffic via HAProxy
- gRPC traffic via Envoy
- Internet egress using NAT gateway

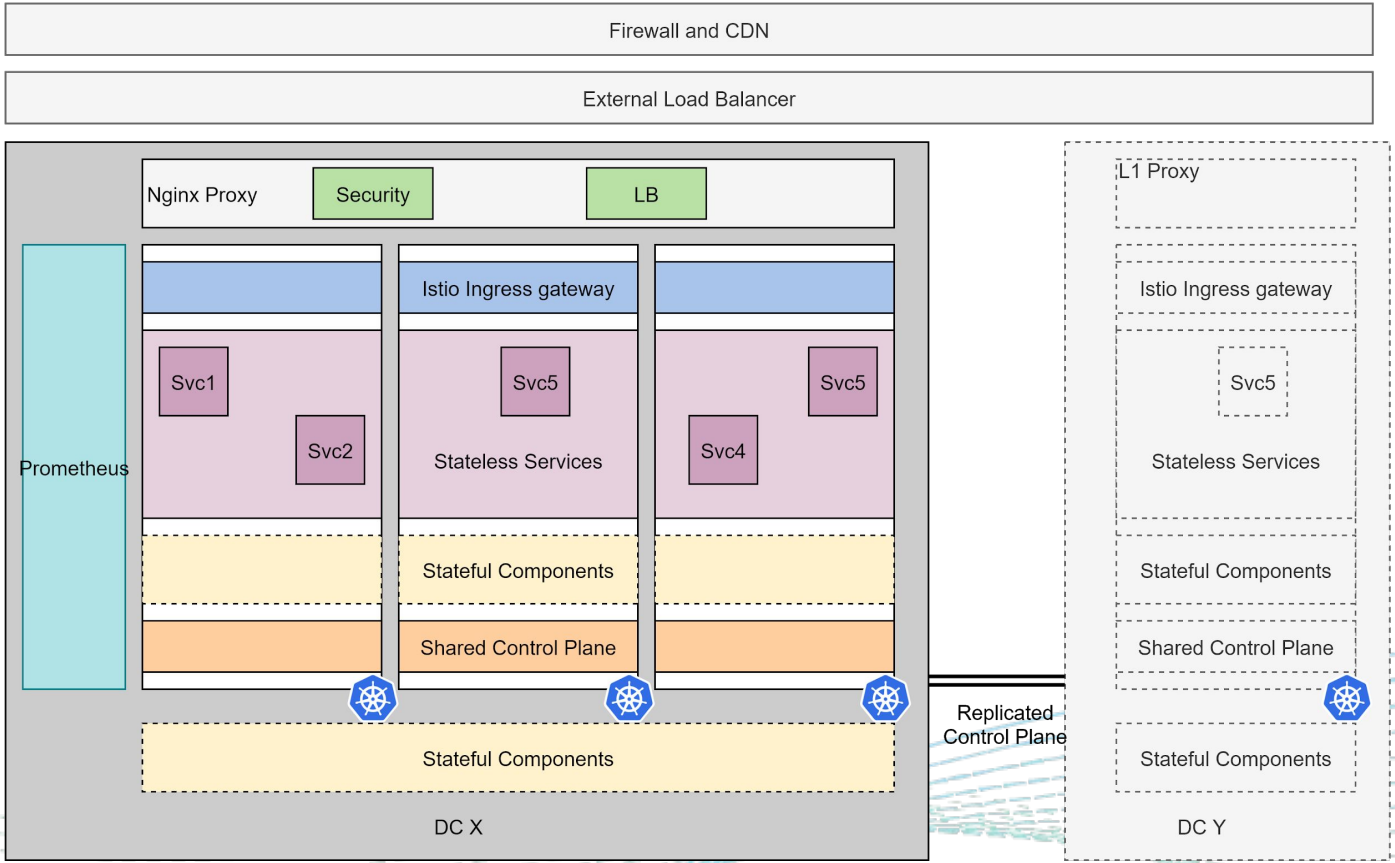
Motivation

- Reliability of central proxy layer (HAProxy/Envoy)
- More control over load balancing
- Offload application services from networking and configuration
- Avoid other sources of failures (Consul etc)
- Possible benefits on Observability

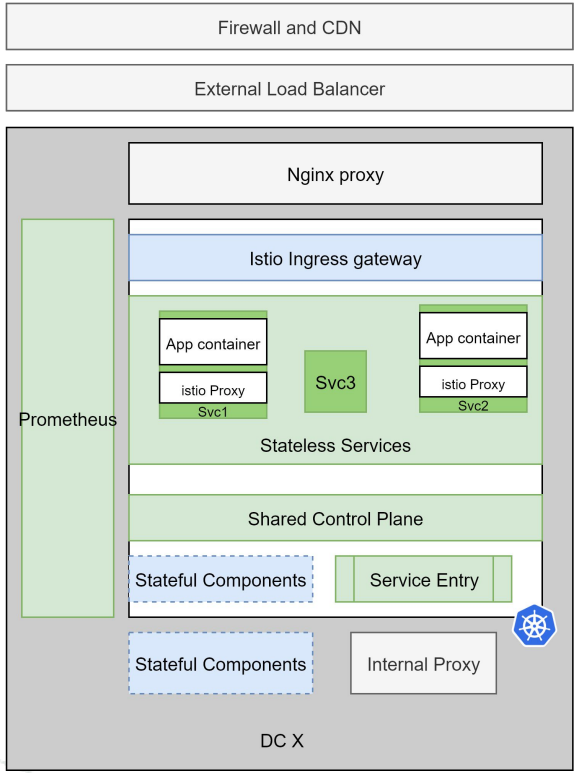
Requirements and Improvements

- Immutable deployments
- Minimal blast radius
- Discover Pods for controlled and predictable routing/load balancing
- Improve performance and resilience
- Stricter zonal routing
- Capability for service authentication and authorisation
- Improved Observability
- Extendable to multi-region setup

Approach

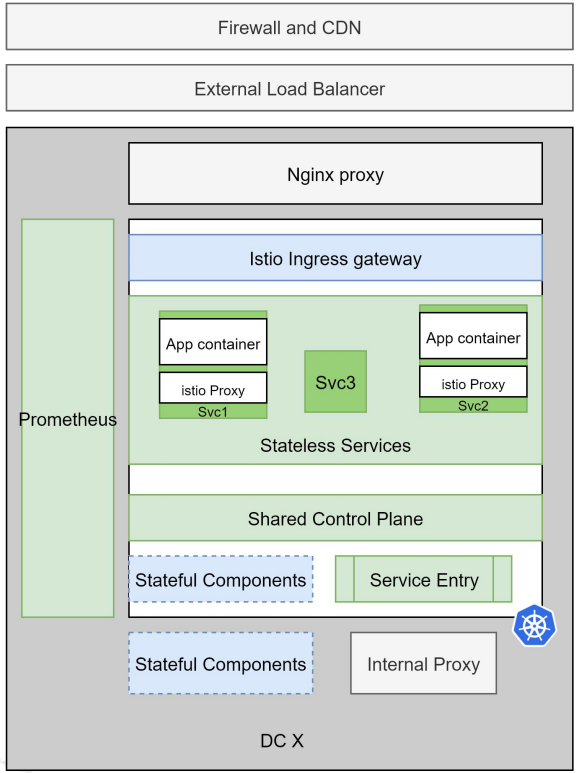

Firewall and CDN
External Load Balancer
Nginx Proxy
Security
LB
Prometheus
Istio Ingress gateway
Svc1
Svc2
Svc5
Stateless Services
Svc5
Svc4
Stateful Components
Shared Control Plane
Stateful Components
DC X
Replicated
Control Plane
L1 Proxy
Istio Ingress gateway
Svc5
Stateless Services
Stateful Components
Shared Control Plane
Stateful Components
DC Y

Rollout - Istio setup and Microservices



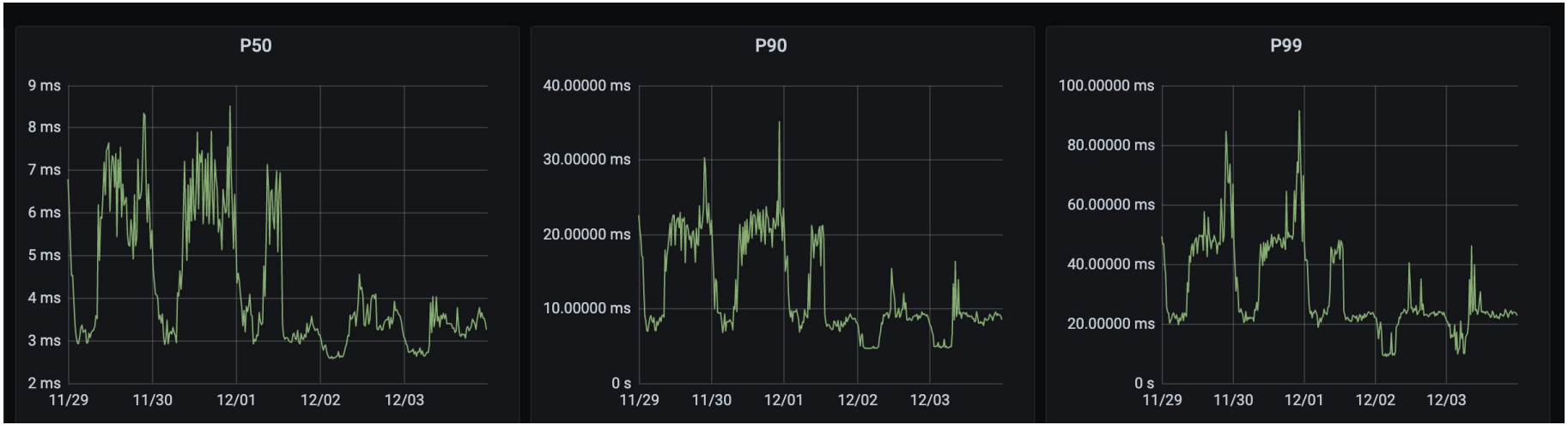
- Split rollout in to phases
- Setup control plane and related tooling
- Sidecar injection by namespace or on-demand
- Passthrough mode during rollout
- Service entry to connect internal proxy
- Kubernetes Cluster-IP services deployed across clusters

Rollout - Istio setup and Microservices



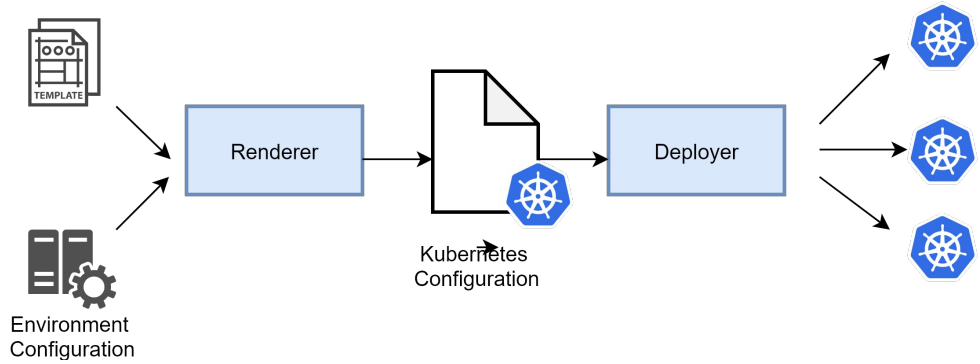
- Export metrics to central prometheus
- Outlier detection for better reliability
- Enable Zonal routing, zonal deployment and HPA
- Endpoint accessed by service via config

Latency improvement



Tooling and Automation

- Automate the Istio setup during Kubernetes cluster creation
- Automated endpoint config creation on new micro-service creation or updation
- Templatise the Kubernetes deployment including Virtual Service and Destination rule



Takeaways

- Identify the problems and improvements
- POCs for all known use-cases and features say mTLS, Outlier detection etc,.
- Passthrough mode downgrades gRPC/http2 protocol to Http/1.1
- Tune connection and TCP settings
- Handle signals gracefully (SIGINT, SIGTERM)
- Automate for easy management of setup across environments
- Ignore ports / IP as applicable - consul
- Namespace isolation helps reduce Istio proxy resources

Next Steps

- Move stateful components in to mesh discovery and routing
- Expose gateway services via Istio Gateway
- Towards RESTRICTED network policy
- On-board services to Authentication and Authorization as applicable

Thank you!

Rajath Ramesh
rajathramesh@carousell.com / https://www.linkedin.com/in/rajathramesh/

Harshad Rotithor
harshad@carousell.com / https://www.linkedin.com/in/harshadr/

#IstioCon