On-demand Developer Environments in Mesh based Infrastructure

Rajath Ramesh | Software Architect @ Carousell
Edward Samuel Pasaribu | Senior Software Engineer @ Carousell

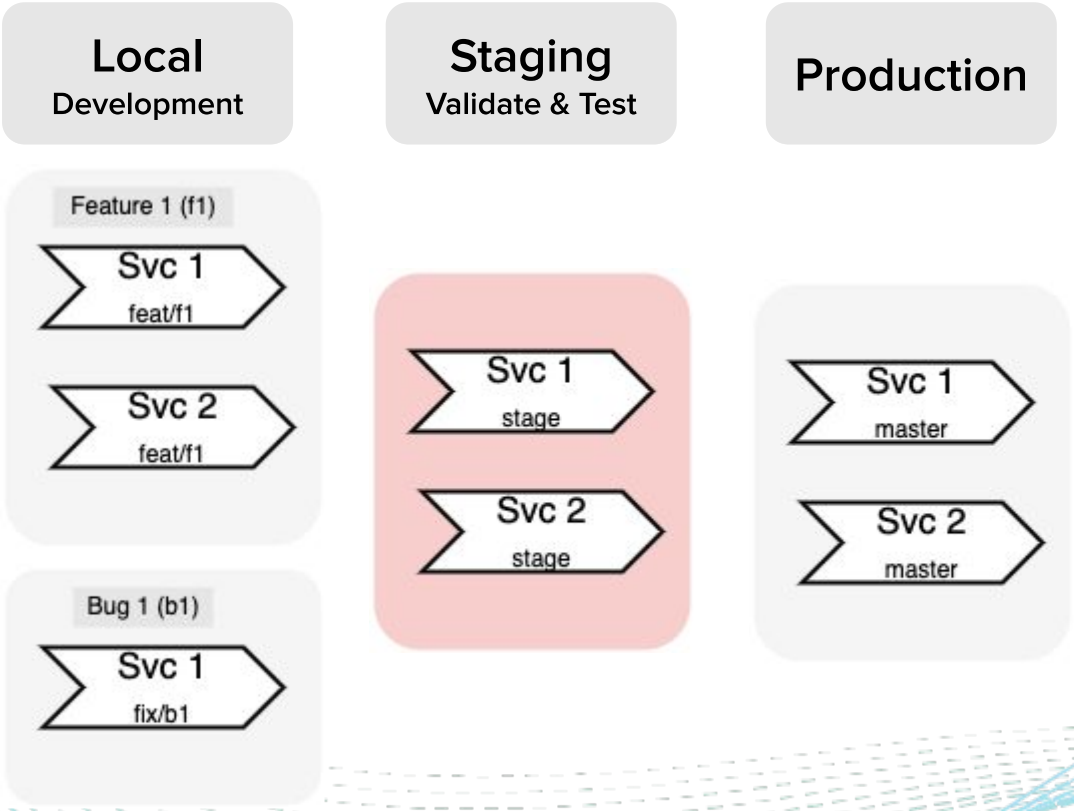
IstioCon

#IstioCon

Agenda

- Developer Environment - Gaps and Improvements
- Architecture Overview
- Approach - Ondemand Developer Environment
- Solution - Contextual Routing
- Tooling and Rollout
- Next Steps

Developer Environments - Gaps



Local
Development

Staging
Validate & Test

Production

Feature 1 (f1)

Svc 1
feat/f1

Svc 2
feat/f1

Bug 1 (b1)

Svc 1
fix/b1

Svc 1
stage

Svc 2
stage

Svc 1
master

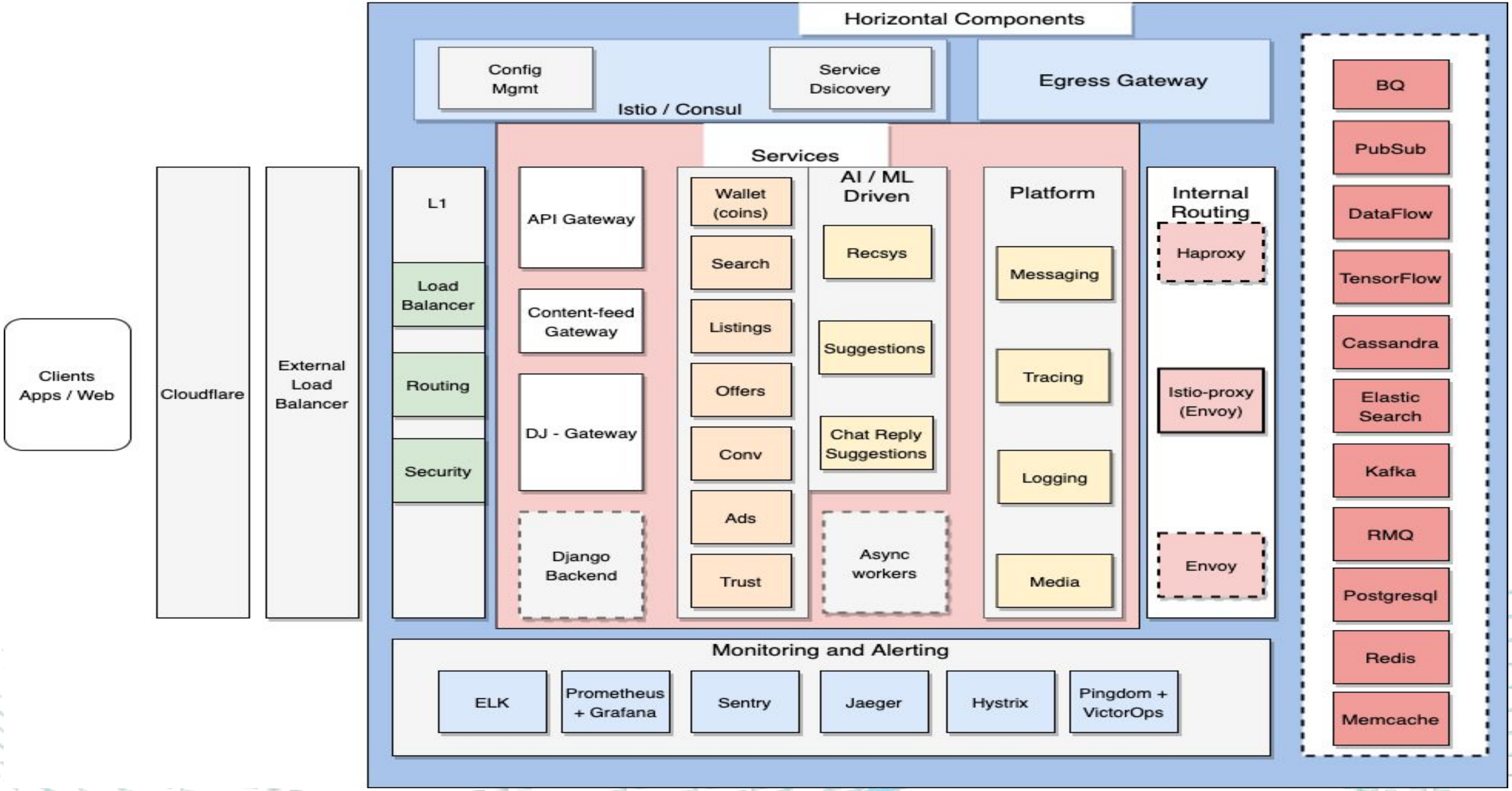
Svc 2
master

- 3 environments - integration tested in Stage
- Overlapping changes - risk of improper mergers
- Multiples changes for given service tested in parallel
- Unstable dependency affecting delivery
- Hampers productivity and experience

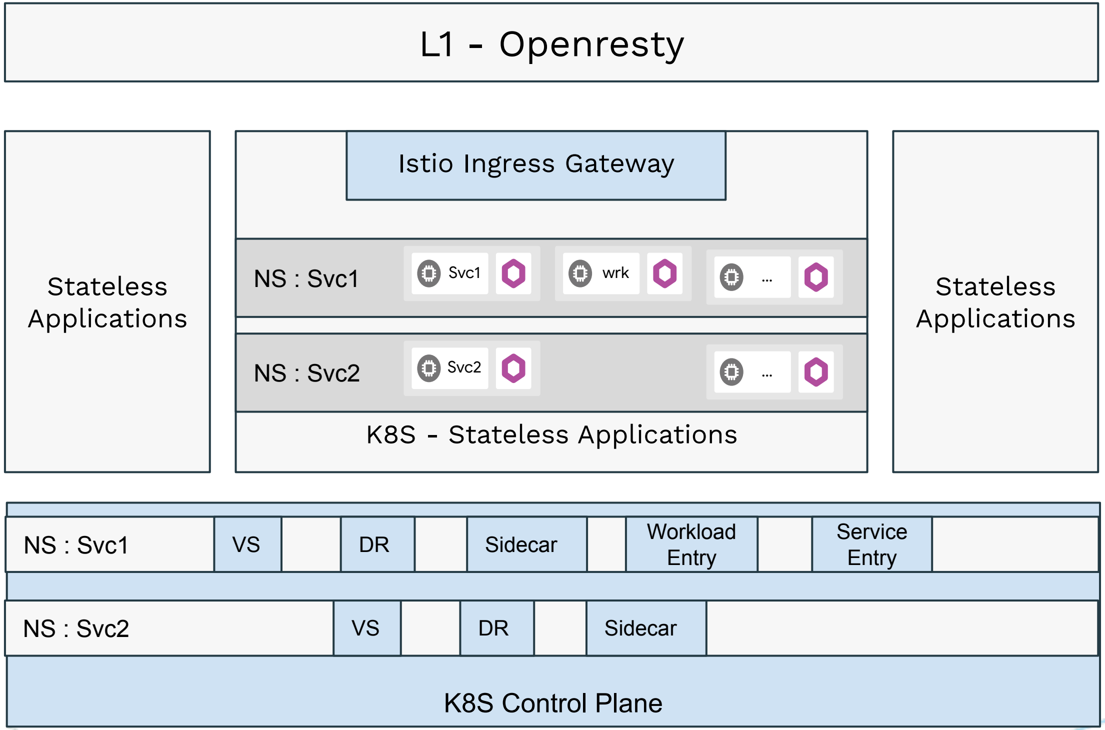
Requirement and Improvements

- Isolated environments for the service
- Extendable to multiple services in given Isolated environment
- Keep it developer friendly and cost effective
- Strictly platform changes to enable the need
- Self service tooling to manage the environment

Architecture Overview



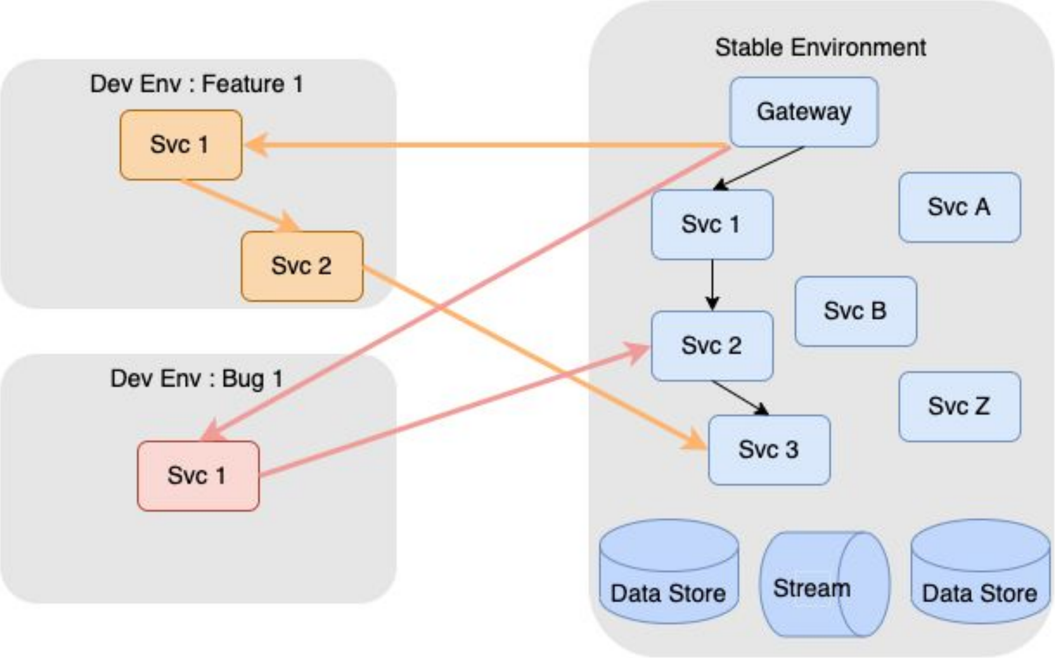
Architecture Overview : Istio Artefacts



- Shared control plane
- Istio Ingress Gateway routes across clusters
- Virtual Service, Destination Rule per service
- Sidecar CRD for namespace isolation

#IstioCon

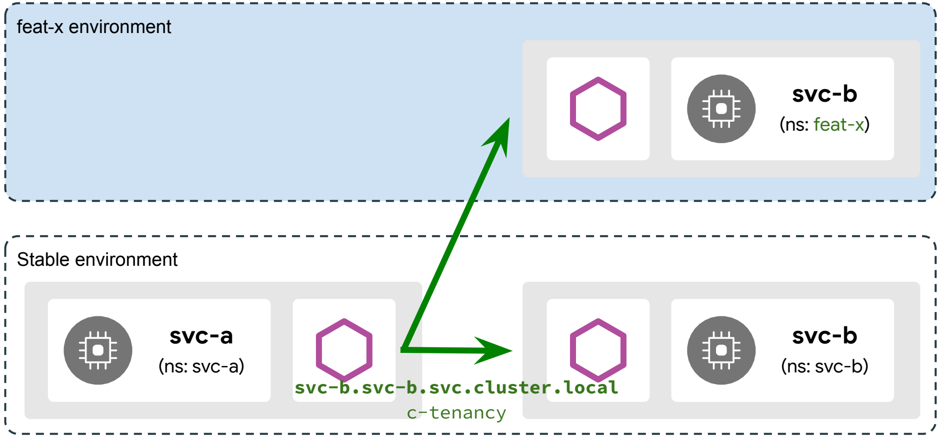
Approach - Ondemand Dev Environment



- Stage == Stable Env running prod image
- Dev environment contains services under changes only
- Isolates service but Shared datastores
- Managed Environment lifecycle

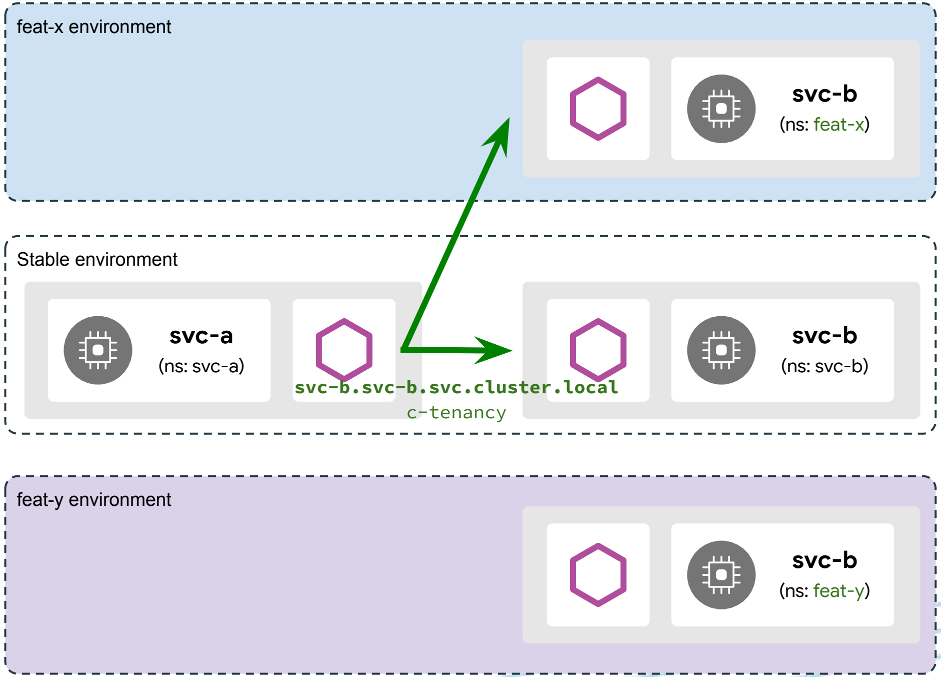
Contextual Routing with Virtual Service

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
 name: svc-b
 namespace: svc-b
spec:
 hosts:
 - svc-b.svc-b.svc.cluster.local
 http:
 - route:
   - destination:
       host: svc-b.svc-b.svc.cluster.local
```



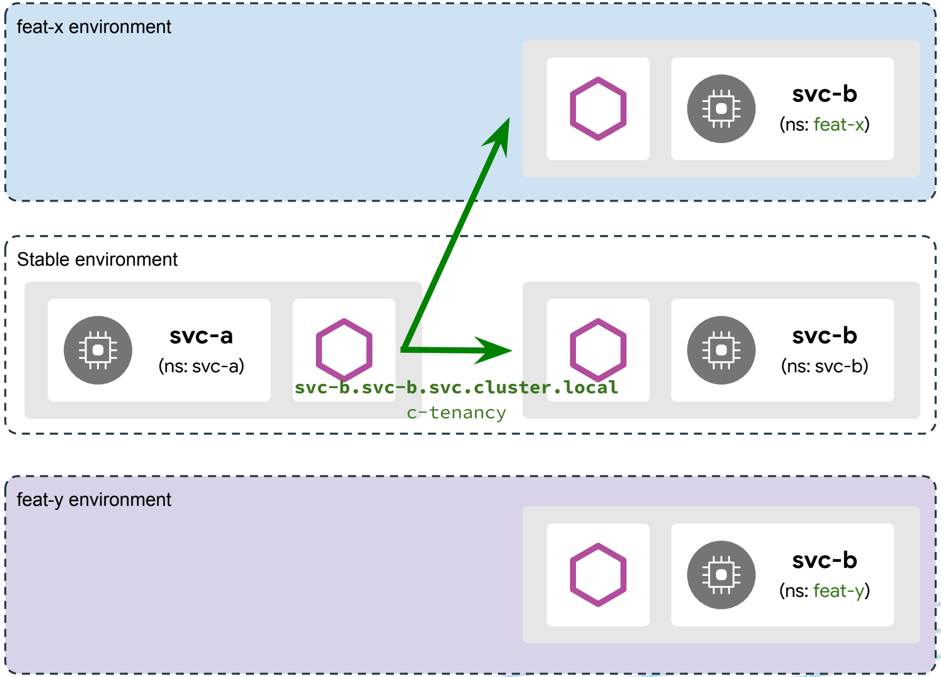
Contextual Routing with Virtual Service

```yaml
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
 name: svc-b
 namespace: svc-b
spec:
 hosts:
 - svc-b.svc-b.svc.cluster.local
 http:
 - match:
   - headers:
       c-tenancy:
         exact: feat-x
   route:
   - destination:
       host: svc-b.feat-x.svc.cluster.local
 - route:
   - destination:
       host: svc-b.svc-b.svc.cluster.local
```

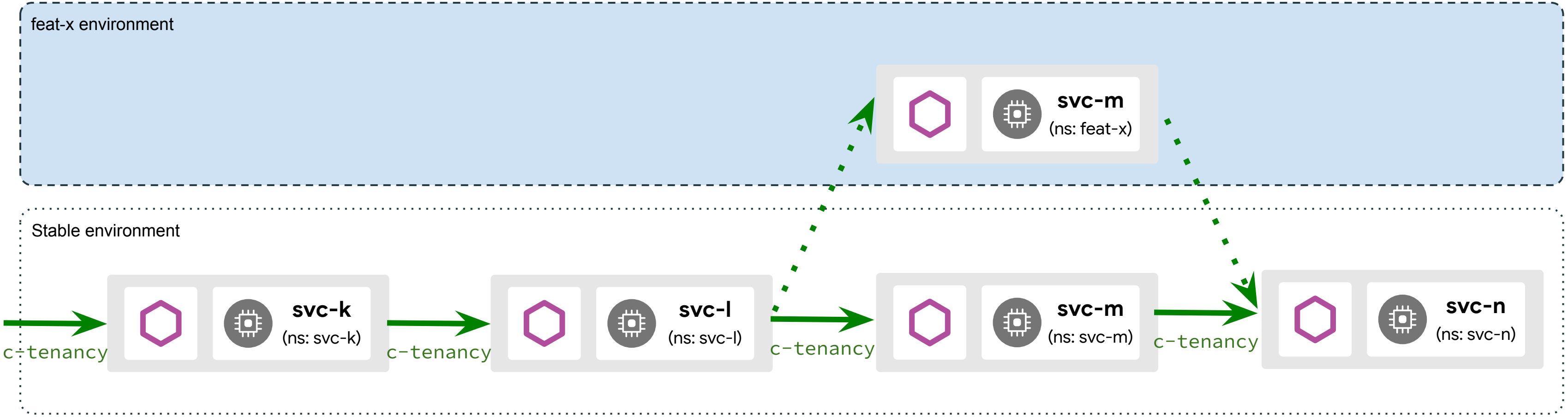


Contextual Routing with Virtual Service

```yaml
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
 name: svc-b
 namespace: svc-b
spec:
 hosts:
 - svc-b.svc-b.svc.cluster.local
 http:
 - match:
   - headers:
       c-tenancy:
         exact: feat-x
   route:
   - destination:
       host: svc-b.feat-x.svc.cluster.local
 - match:
   - headers:
       c-tenancy:
         exact: feat-y
   route:
   - destination:
       host: svc-b.feat-y.svc.cluster.local
 - route:
   - destination:
       host: svc-b.svc-b.svc.cluster.local
```



Contextual Routing with Virtual Service



Metadata/headers propagation

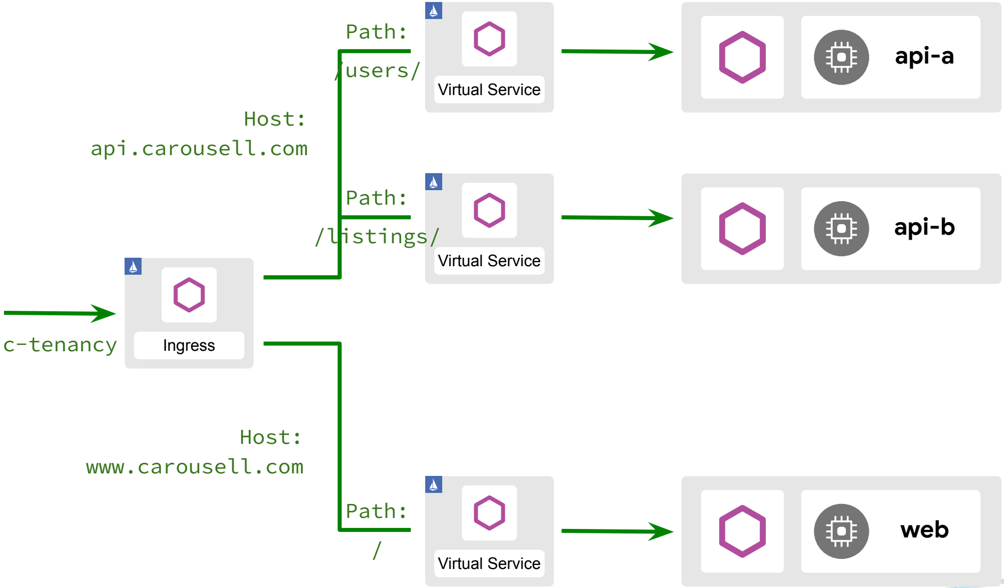
App should **forward selected incoming headers** (i.e: **`c-tenancy`** header) when we make calls to other services.

Extend: ***Trace context propagation*** **(i.e: `x-b3-*`, etc)**
(https://istio.io/latest/docs/tasks/observability/distributed-tracing/overview/)

Ingress Routing

Ingress delegates the routing to a virtual service

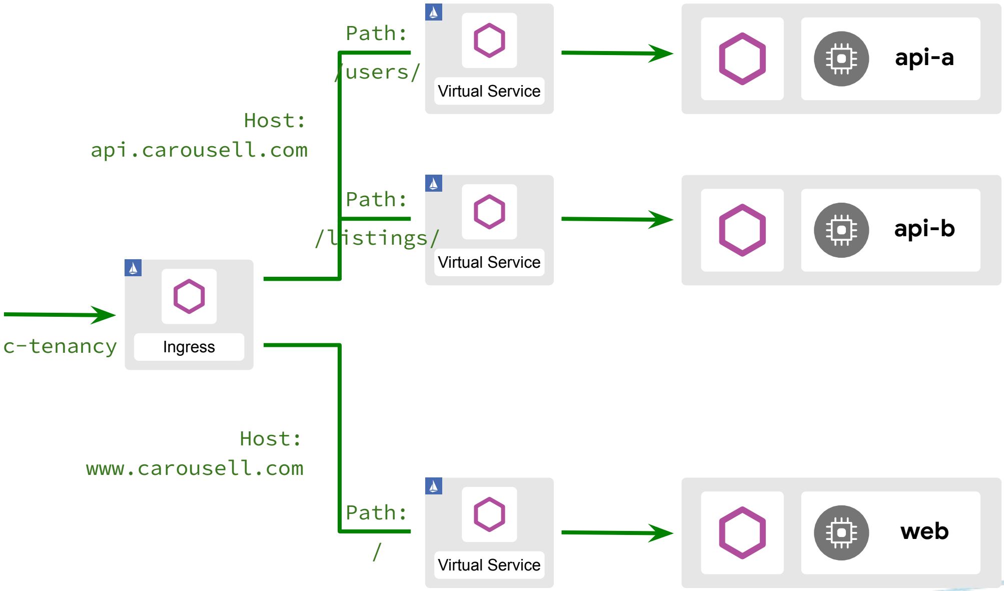
```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
 labels:
   app: istio-ingress-gw
 name: istio-ingress-gw
 namespace: istio-ingress-gw
spec:
 gateways:
 - istio-ingress-gw/istio-ingress-gw
 hosts:
 - api.carousell.com
 http:
 - delegate:
     name: api-a-vs
     namespace: api-a
   match:
   - uri:
       prefix: /users/
...
```



Ingress Routing

Ingress delegates the routing to a virtual service

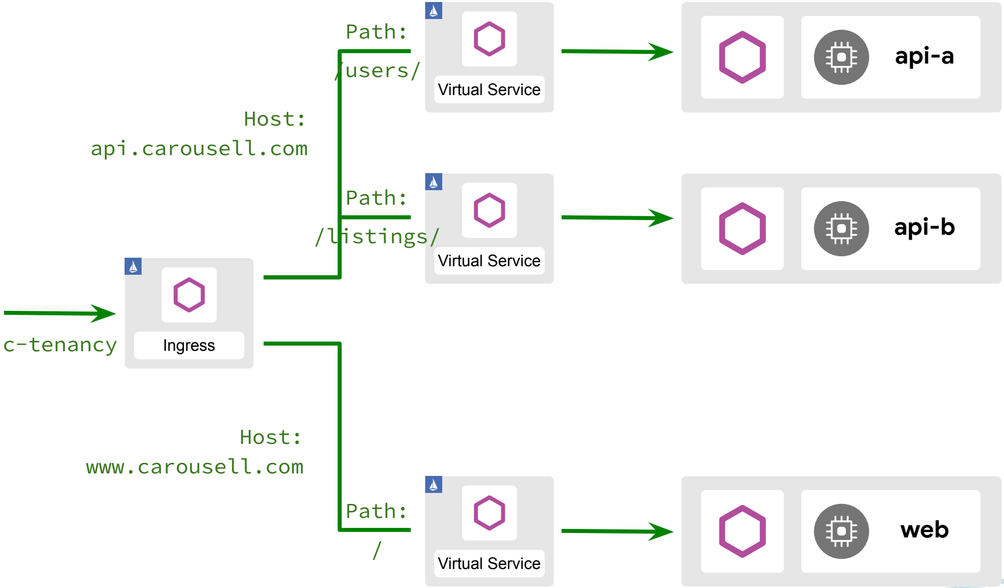
```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
 name: api-a-vs
 namespace: api-a
spec:
 http:
 - match:
   - headers:
       c-tenancy:
         exact: feat-x
   route:
   - destination:
       host:
api-a.feat-x.svc.cluster.local
...
 - route:
   - destination:
       host: api-a.api-a.svc.cluster.local
```



Ingress Routing

Delegation can be applied for the inter-service Virtual Service

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
 name: api-a
 namespace: api-a
spec:
 hosts:
 - api-a.api-a.svc.cluster.local
 http:
 - delegate:
     name: api-a-vs
     namespace: api-a
```

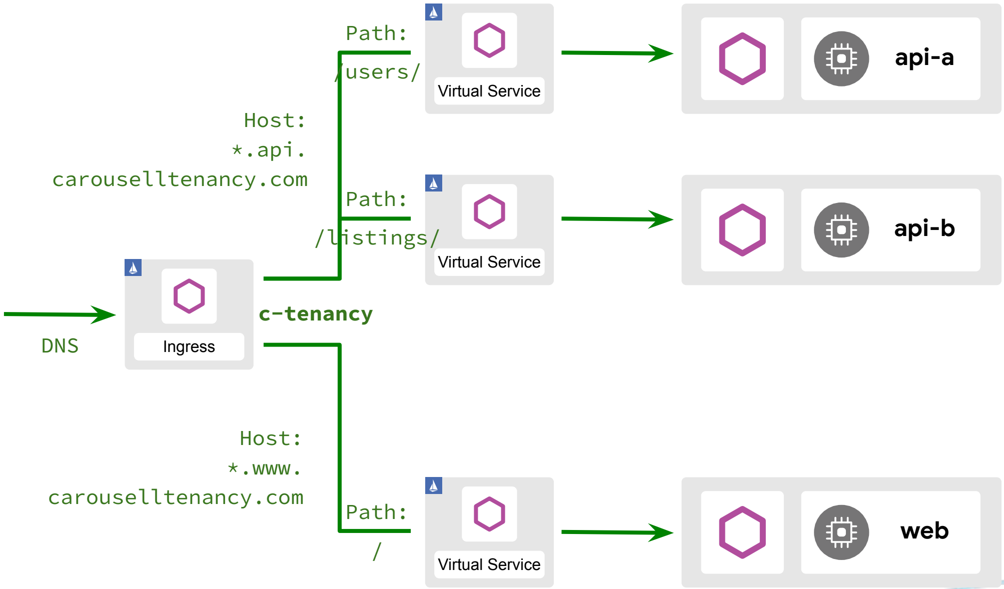


Ingress Routing (Host)

Host:
<tenancy-name>.www.carouselltenancy.com

Capture **<tenancy-name>** from host/domain name

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
...
spec:
 ...
 configPatches:
 - applyTo: HTTP_FILTER
   match:
     context: GATEWAY
     listener:
       ...
   patch:
     operation: INSERT_BEFORE
     value:
       name: envoy.lua
       typed_config:
         ...
         inlineCode: |
           function envoy_on_request(request_handle)
             local tenancy = request_handle:headers():get("c-tenancy")
             if not tenancy then
               local host = request_handle:headers():get("host")
               request_handle:headers():add("c-tenancy", host:gsub("([^%s]+)%.[^%s]+%.carouselltenancy%.com",
"%1"))
             end
           end
```

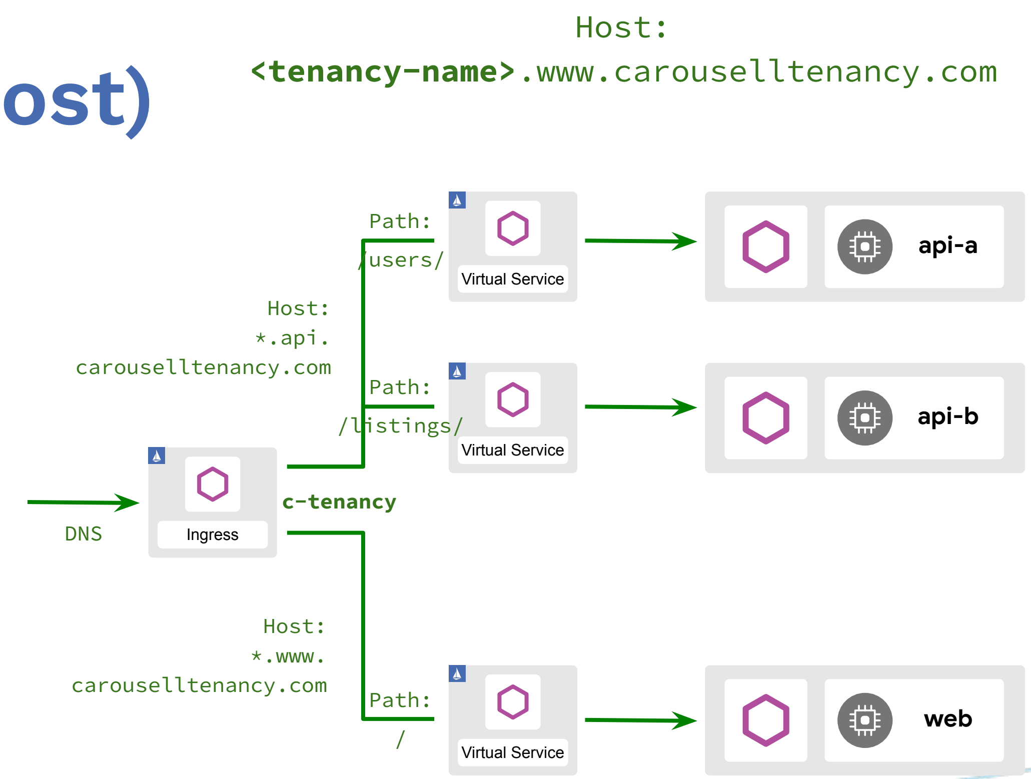


Ingress Routing (Host)

Wildcard DNS **<tenancy-name>**:

```
*.api.carouselltenancy.com
A <IP>

*.www.carouselltenancy.com
A <IP>
```



#IstioCon

Solution

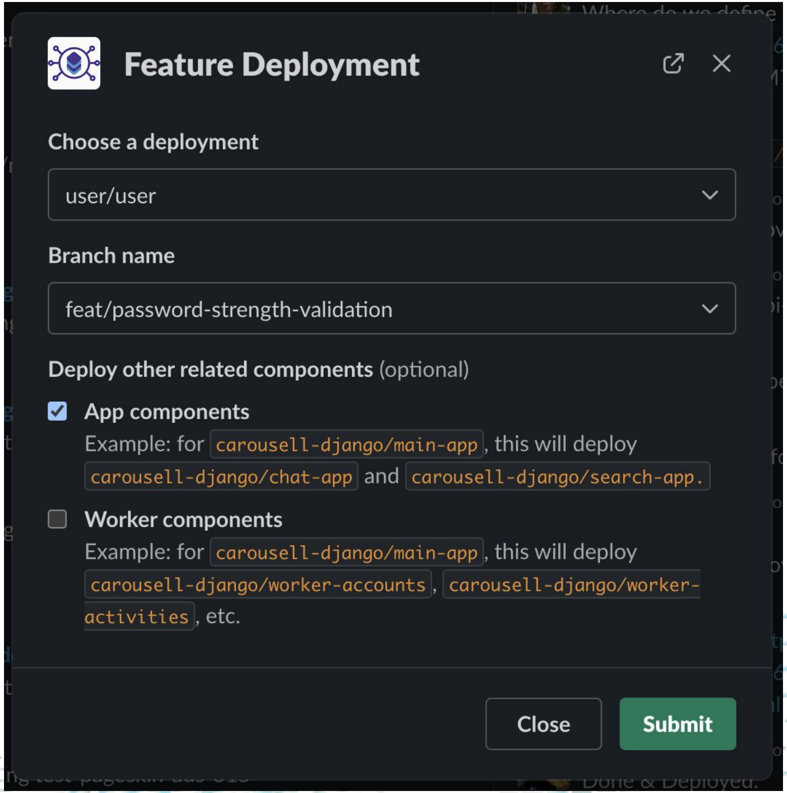
- Deploy only required workloads in separate environments/namespaces
- Virtual Service with tenancy routing (i.e: through a header)
- App implements headers forwarding
- Ingress Gateway with a delegated Virtual Service

Tips/recommendation:

- Wildcard domains for each entrypoint
  - Prefer: *.api.carouselltenancy.com and *.www.carouselltenancy.com
  - Rather than: <api/web>.<feature-name>.carouselltenancy.com
  - Wildcard SSL cert and wildcard DNS entries
  - Capture tenancy header from host name
- Istio Namespace Isolation
  - Whitelist egress of a service with Sidecar object

Rollout and Tooling

- CLI / scripts
- Jenkins
- Slack deployment bot
  - `/feature-deploy <feat-x>`
  - `/feature-destroy <feat-x>`
- Automatic clean-up inactive environments*



Feature Deployment

Choose a deployment

user/user

Branch name

feat/password-strength-validation

Deploy other related components (optional)

☑ App components

Example: for `carousell-django/main-app`, this will deploy `carousell-django/chat-app` and `carousell-django/search-app.`

□ Worker components

Example: for `carousell-django/main-app`, this will deploy `carousell-django/worker-accounts`, `carousell-django/worker-activities`, etc.

#IstioCon

Next Steps

- Config management for dev environments
  - Static immutable configuration with ConfigMap on each deployment
- Context–aware datastore (database, broker/messaging, etc)

Thank you!

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

Edward Samuel Pasaribu
edwardsamuel@carousell.com / https://www.linkedin.com/in/edward-samuel-pasaribu/