

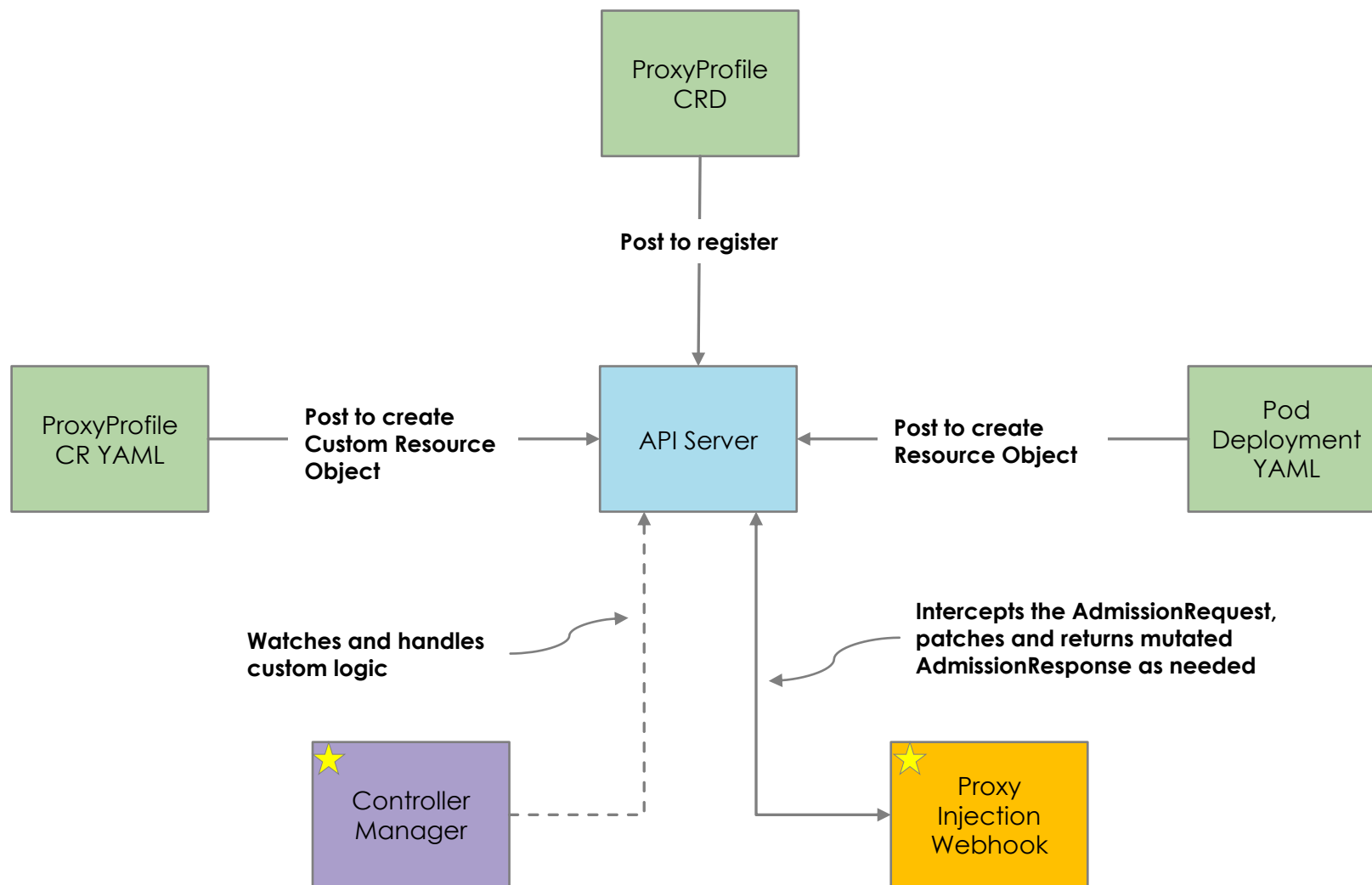
Operator and Ingress Controller

pipy-operator and ingress-pipy

Pipy Operator

pipy-operator

Architecture of pipy-operator



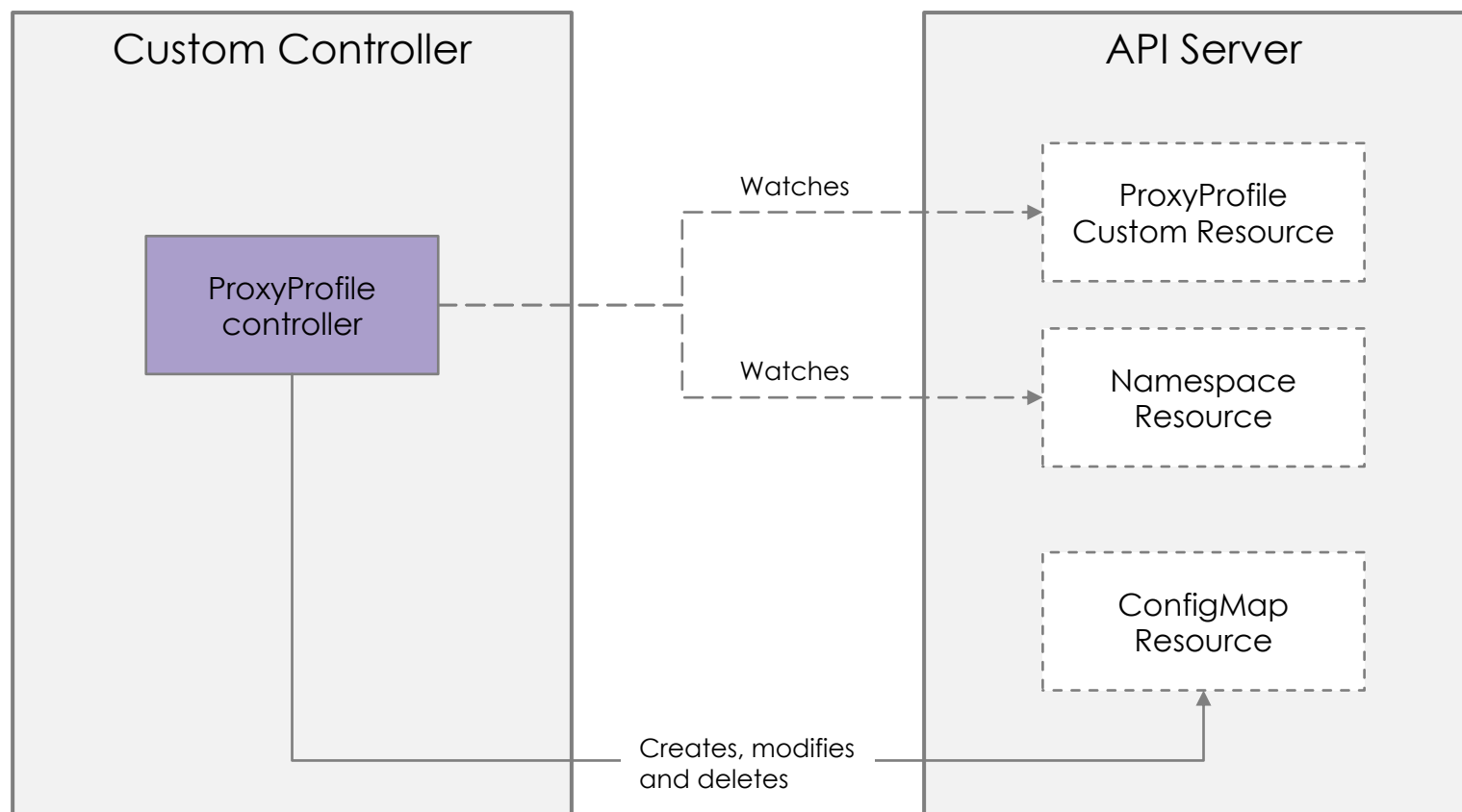
Runtime components of pipy-operator



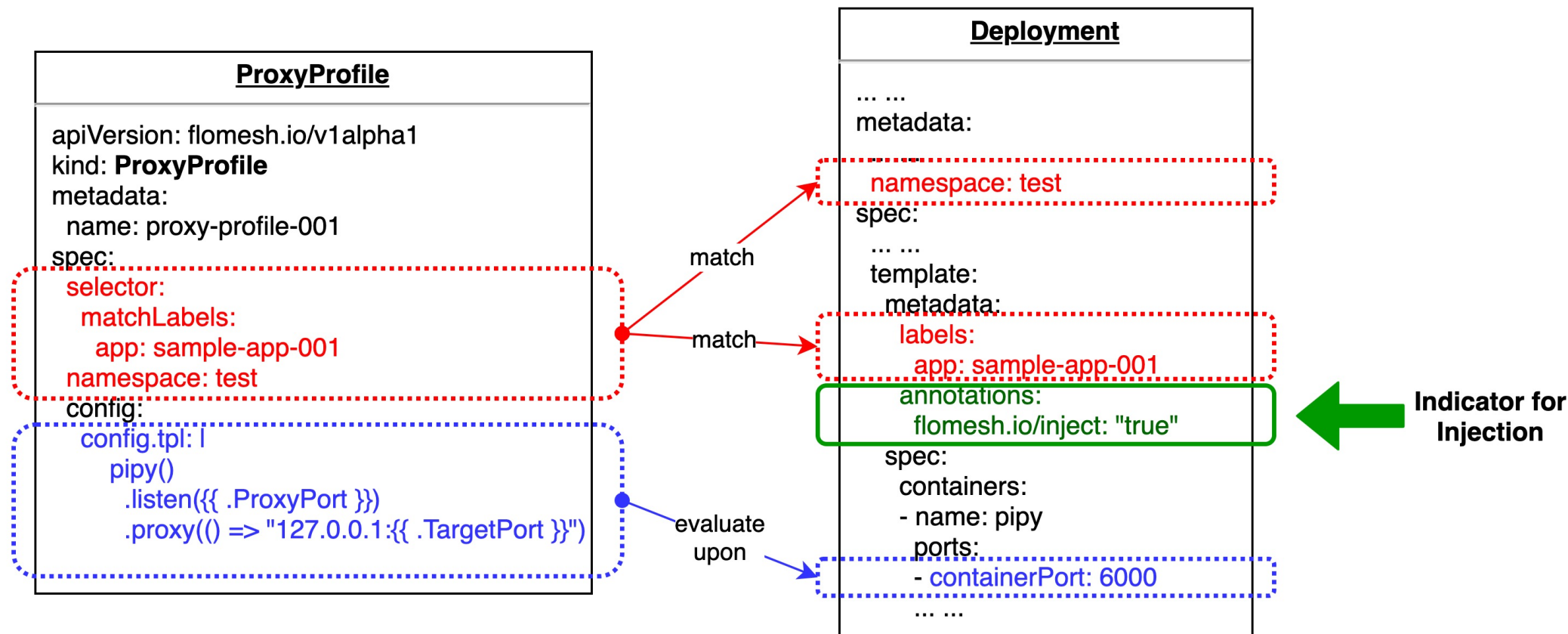
```
[root@crd pipy-operator]# kubectl get pods -A
```

| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|----------------|--|-------|---------|----------|-------|
| kube-system | metrics-server-7b4f8b595-mp2ln | 1/1 | Running | 0 | 2m31s |
| kube-system | local-path-provisioner-7ff9579c6-2pbcc | 1/1 | Running | 0 | 2m31s |
| kube-system | coredns-66c464876b-2m788 | 1/1 | Running | 0 | 2m31s |
| cert-manager | cert-manager-cainjector-59f76f7fff-v526z | 1/1 | Running | 0 | 89s |
| cert-manager | cert-manager-59f6c76f4b-fkwhx | 1/1 | Running | 0 | 89s |
| cert-manager | cert-manager-webhook-565d54dd68-rdz92 | 1/1 | Running | 0 | 89s |
| flomesh-system | flomesh-pipy-sidecar-injector-6c75f7c464-dnn9b | 1/1 | Running | 0 | 41s |
| flomesh-system | flomesh-controller-manager-c75bc95d5-9v86m | 1/1 | Running | 0 | 41s |

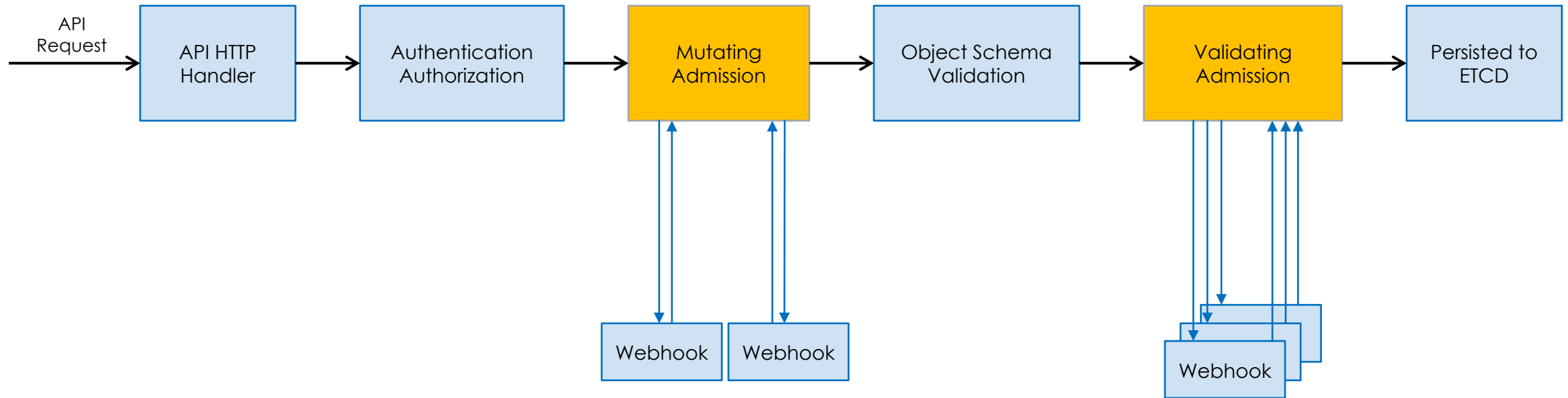
The ProxyProfile Controller



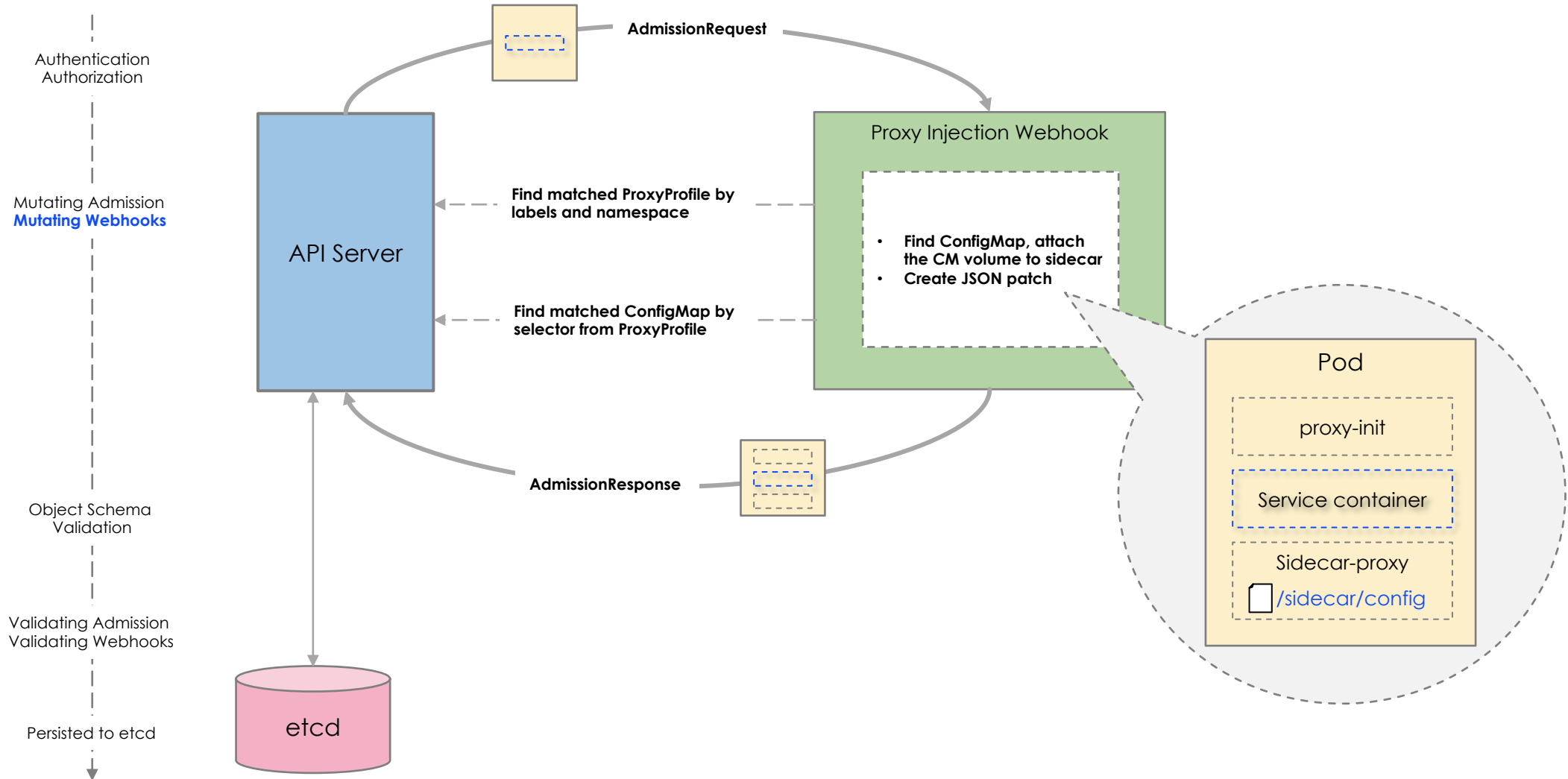
ProxyProfile & Pod Matching



Admission Controller Phases

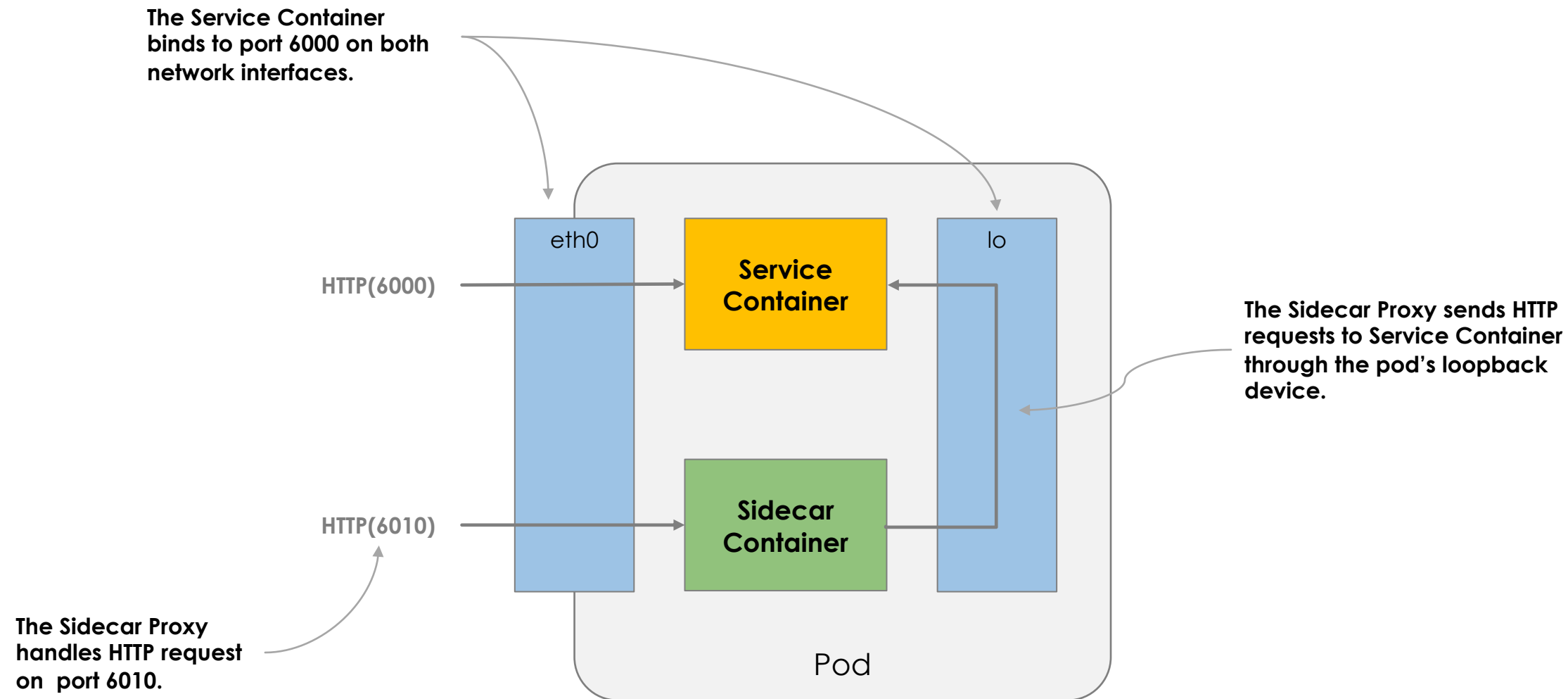


Proxy Injection Webhook





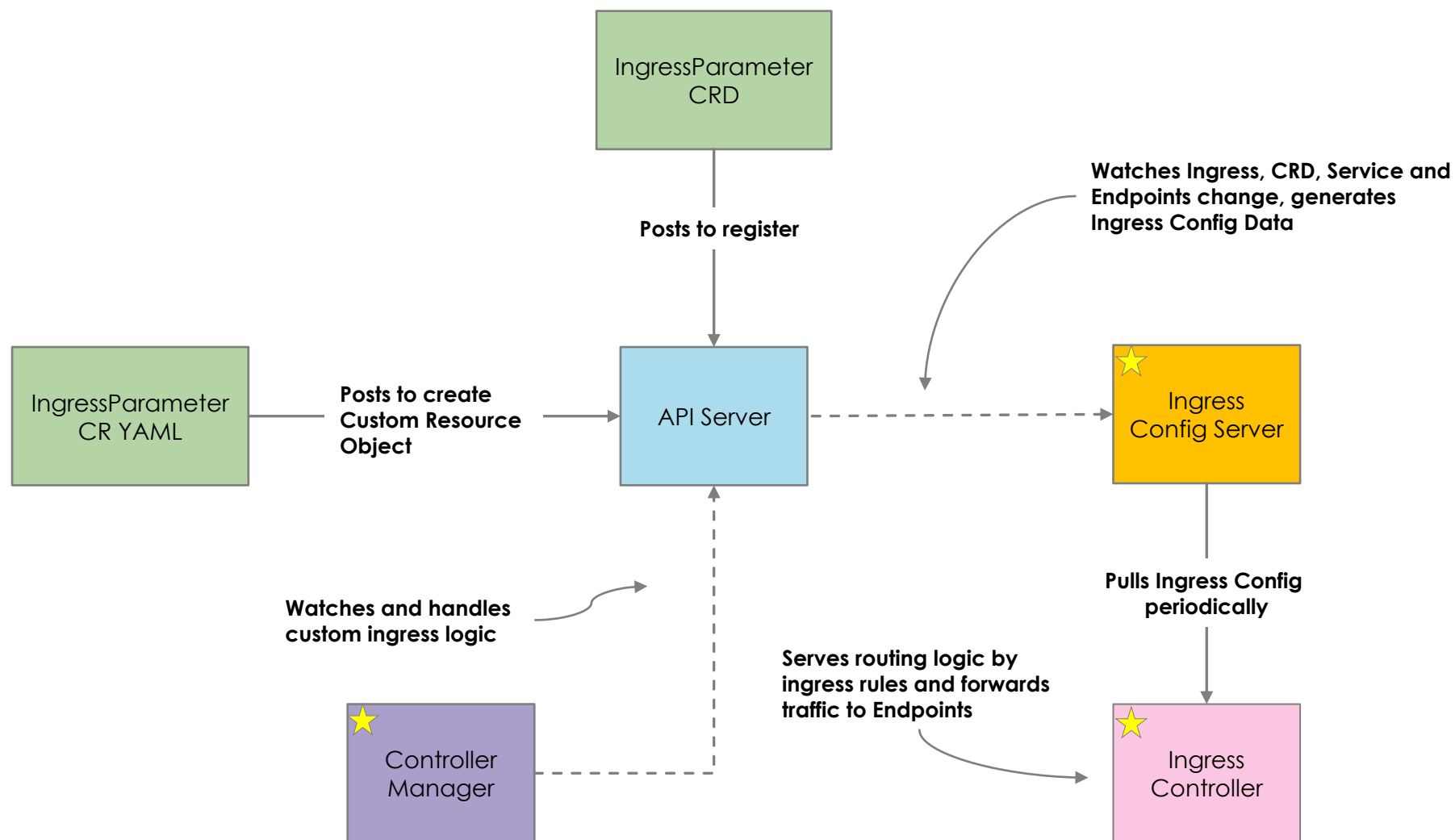
Sidecar Traffic Hijacking



Ingress Controller

ingress-pipy

Architecture of ingress-pipy



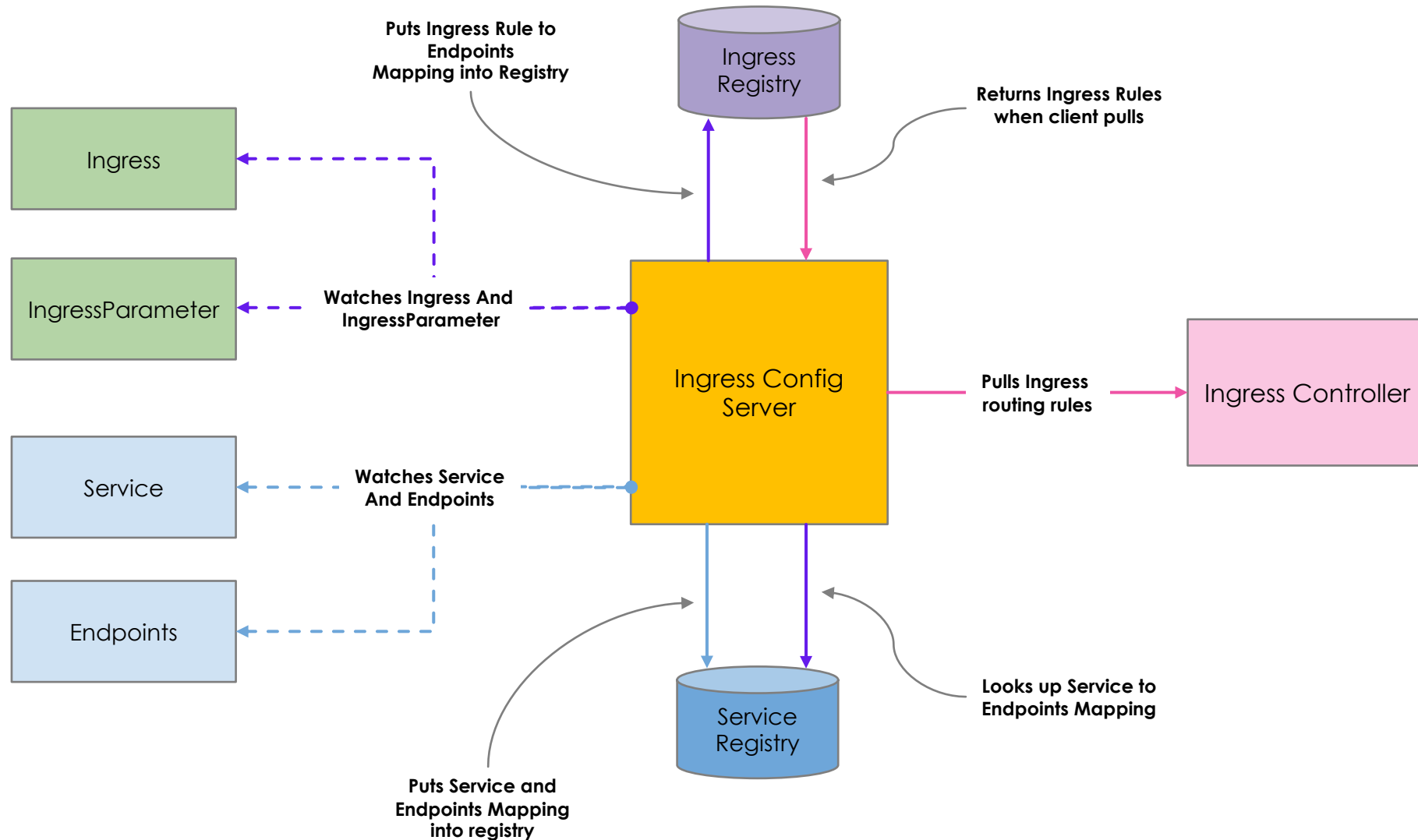
Runtime components of ingress-pipy



```
root@bookinfo:/vagrant/kubernetes# kubectl get po -n ingress-pipy
```

| NAME | READY | STATUS | RESTARTS | AGE |
|--|-------|---------|----------|-------|
| svclb-ingress-pipy-controller-p8h9g | 1/1 | Running | 0 | 2m58s |
| ingress-pipy-cfg-6856d674f7-zcgbr | 1/1 | Running | 0 | 2m58s |
| ingress-pipy-controller-76cd866d78-bcmb5 | 1/1 | Running | 0 | 2m58s |
| ingress-pipy-manager-6dddc98484-q4ls5 | 1/1 | Running | 0 | 2m58s |

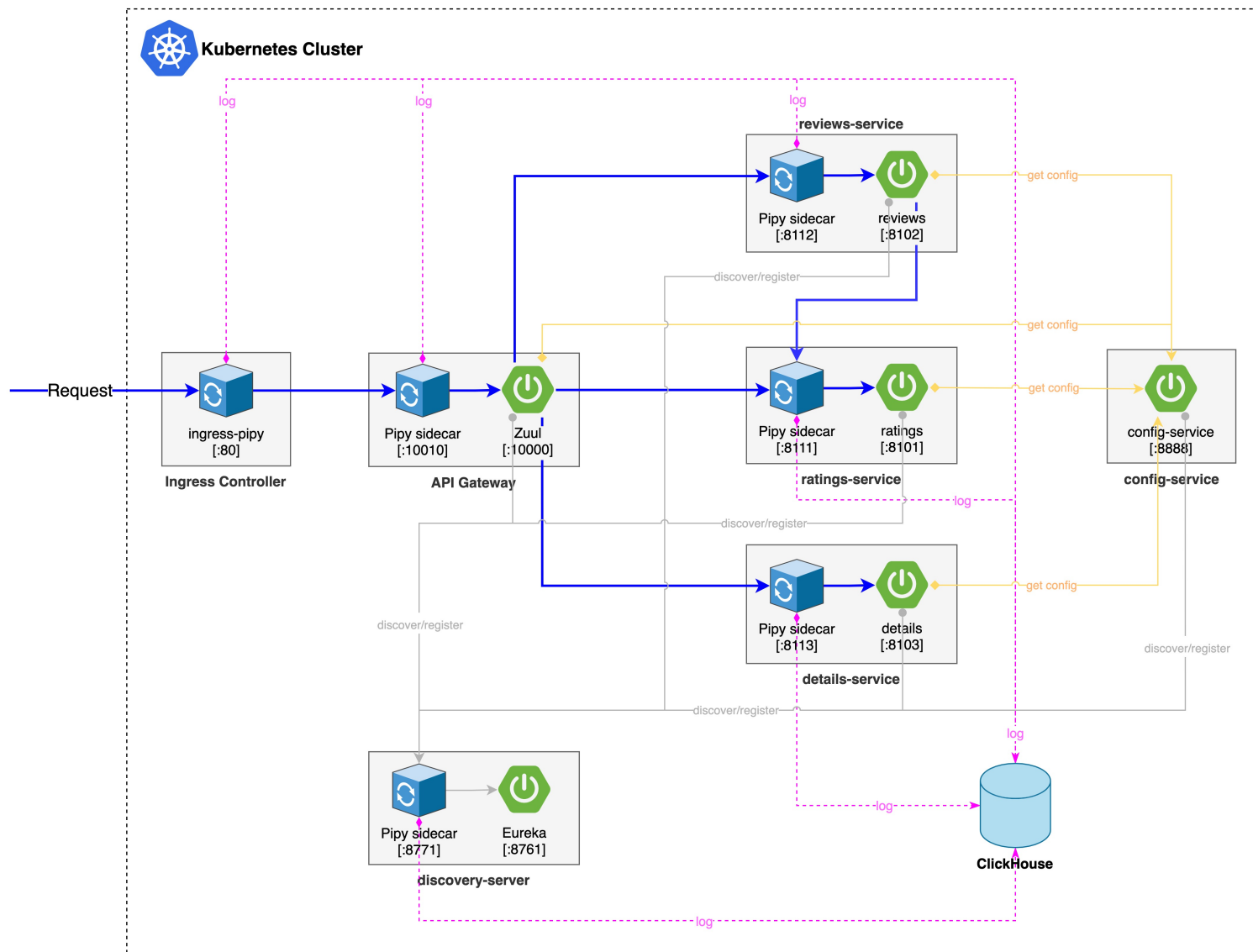
Ingress Config Server



DEMO

**A complete demo with
pipy-operator and ingress-pipy**

Topology of BookInfo Demo



Deploy Steps



Install cert-manager

Install pipy-operator

Install ingress-pipy

Install BookInfo Demo

Sidocar configuration



```
1  apiVersion: flomesh.io/v1alpha1
2  kind: ProxyProfile
3  metadata:
4    name: proxy-profile-002-bookinfo
5  spec:
6    selector:
7      matchLabels:
8        sys: bookinfo-samples
9        version: v1
10   namespace: default
11   config:
12     config.tpl: |
13       pipy({
14         _BATCH_SIZE: 5,
15         _BATCH_TIMEOUT: 5000,
16         _CONTENT_TYPES: {
17           '': true,
18           'text/plain': true,
19           'text/html': true,
20           'application/json': true,
21           'application/xml': true,
22           'multipart/form-data': true,
23         },
24         _g: {
25           buffer: new Data,
26           bufferSize: 0,
27           bufferTime: 0,
28         },
29       },
30       _queue: null,
31       _contentType: '',
32       _startTime: 0,
33       _responseTime: 0,
34     })
35   .listen({{ .ProxyPort }})
36   .onSessionStart() => _queue = []
37   .fork('in')
38   .connect('127.0.0.1:{{ .TargetPort }}')
39   .fork('out')
```

```
42
43   // Extract request info
44   .pipeline('in')
45   .decodeHttpRequest()
46   .onMessageStart(e => (
47     _startTime = Date.now(),
48     _contentType = e.head.headers['content-type'] || '',
49     _contentType = _contentType.split(';')[0]
50   ))
51   .onMessage(msg => _queue.push({
52     startTime: _startTime,
53     ...msg.head,
54     body: _CONTENT_TYPES[_contentType] ? msg.body.toString() : null,
55   }))
56
57   // Extract response info
58   .pipeline('out')
59   .decodeHttpResponse()
60   .onMessageStart(e => (
61     _responseTime = Date.now(),
62     _contentType = e.head.headers['content-type'] || '',
63     _contentType = _contentType.split(';')[0]
64   ))
65   .replaceMessage(msg => new Message(
66     JSON.encode({
67       ..._queue.shift(),
68       latency: _responseTime - _startTime,
69       endTime: Date.now(),
70       response: {
71         ...msg.head,
72         body: _CONTENT_TYPES[_contentType] ? msg.body.toString() : null,
73       },
74     }).push(',')
75   ))
76   .link('batch')
77
```

Ingress configuration



```
266 ingress.js: |+
267   pipy({
268     _g: {
269       // [Ingress] - Start
270       router: null,
271       timestamp: 0,
272       // [Ingress] - End
273
274       // [LOGGING] - Start
275       buffer: new Data,
276       bufferSize: 0,
277       bufferTime: 0,
278       // [LOGGING] - End
279     },
280
281     // [Ingress] - Start
282     _target: undefined,
283     _configUrl: new URL(os.env.INGRESS_CONFIG_URL || 'http://127.0.0.1:6000/routes'),
284     _versionUrl: new URL(os.env.INGRESS_VERSION_URL || 'http://127.0.0.1:6000/version'),
285     _needsUpdateConfig: undefined,
286     // [Ingress] - End
287
288     // [LOGGING] - Start
289     _BATCH_SIZE: 5000,
290     _BATCH_TIMEOUT: 5000,
291     _queue: null,
292     _startTime: 0,
293     _responseTime: 0,
294     // [LOGGING] - End
295   })
296
297   .listen(os.env.INGRESS_PORT || 8000)
298   // [LOGGING] - Start
299   .onSessionStart(() => _queue = [])
300   .fork('in')
301   // [LOGGING] - End
302
303   .link('request')
304
305   // [LOGGING] - Start
306   .fork('out')
307   // [LOGGING] - End
308
309   // ----- [INGRESS] Start -----
```

Check out source code



```
git clone https://github.com/flomesh-io/training-materials.git
```

Edit clickhouse.yaml

```
apiVersion: v1
kind: Endpoints
metadata:
  name: samples-clickhouse
labels:
  app: clickhouse
  service: clickhouse
subsets:
- addresses:
  - ip: 172.19.182.213
  ports:
  - name: chdb
    port: 8123
    protocol: TCP
```

Test Rating service



```
curl -X POST http://10.0.2.15/bookinfo-ratings/ratings \
  -H "Content-Type: application/json" \
  -d '{"reviewerId":"9bc908be-0717-4eab-bb51-ea14f669ef20","productId":"a071c269-369c-4f79-be03-6a41f27d6b5f","rating":3}'
```

```
curl http://10.0.2.15/bookinfo-ratings/ratings/a071c269-369c-4f79-be03-6a41f27d6b5f
```

Test Review service



```
curl -X POST http://10.0.2.15/bookinfo-reviews/reviews \  
  -H "Content-Type: application/json" \  
  -d '{"reviewerId":"9bc908be-0717-4eab-bb51-ea14f669ef20","productId":"a071c269-369c-4f79-  
be03-6a41f27d6b5f","review":"This was OK.,"rating":3}'
```

```
curl http://10.0.2.15/bookinfo-reviews/reviews/a071c269-369c-4f79-be03-6a41f27d6b5f
```

Test Detail service



```
curl http://10.0.2.15/bookinfo-details/details/1234567890
```


Check logs



```
[root@flomesh002 ~]# clickhouse-client -h 127.0.0.1
ClickHouse client version 21.5.5.12 (official build).
Connecting to 127.0.0.1:9000 as user default.
Connected to ClickHouse server version 21.5.5 revision 54448.

flomesh002 :) select * from default.log

SELECT *
FROM default.log

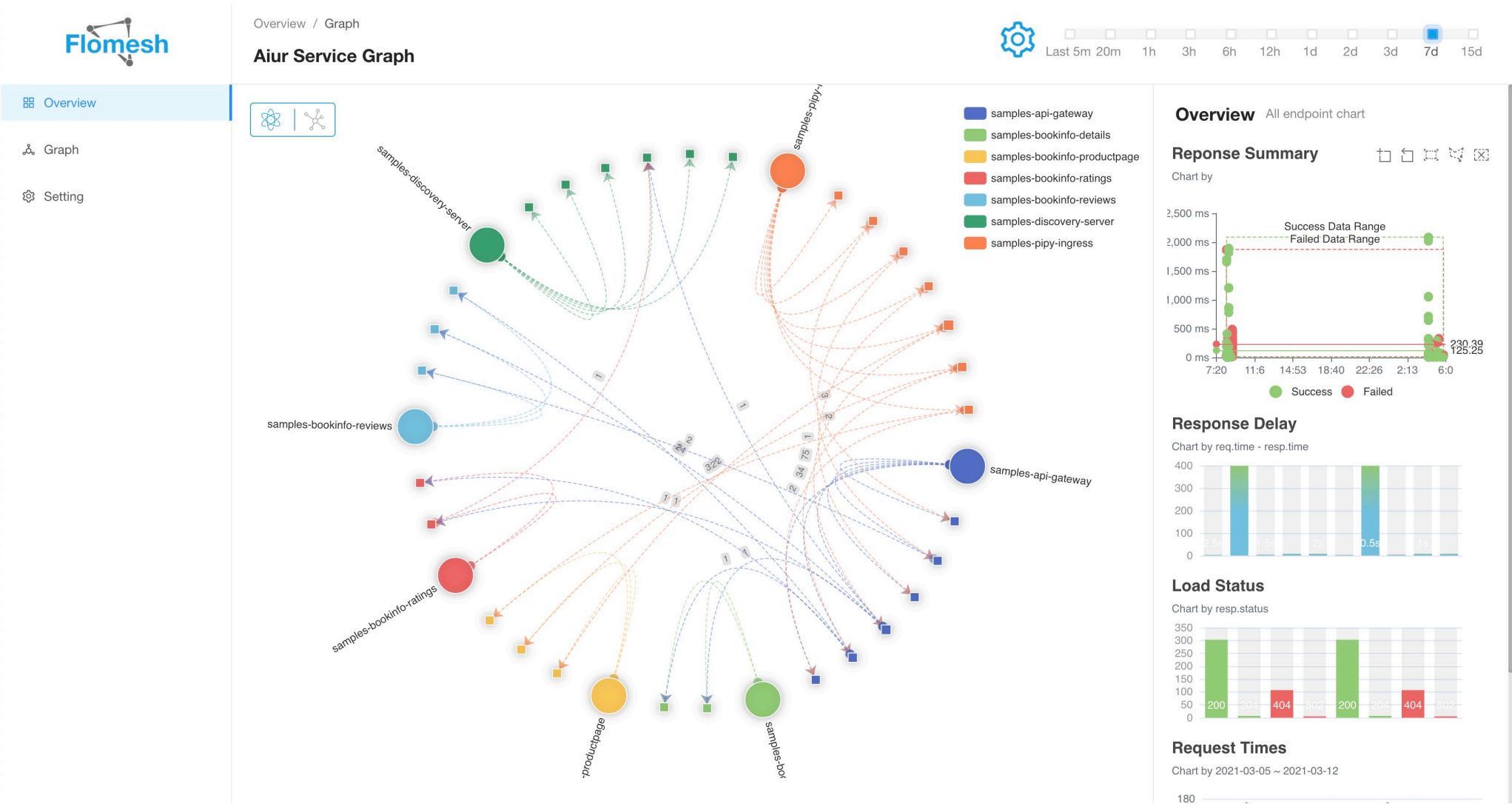
Query id: 0f780bb9-0f56-4c7c-bedf-00fe41c49355

Ok.

0 rows in set. Elapsed: 0.004 sec.

flomesh002 :) █
```

Visualize and analysis



The End

Any questions?