Large Scale Cloud Native Networking & Security With Cilium/eBPF

4 Years Production

Experiences from Trip.com

Arthur (Yanan Zhao)

https://twitter.com/ChiaoArthur

Agenda

1. Cloud infrastructure at Trip.com
2. Cilium/eBPF at Trip.com
   - Rolling out & use cases
   - Customizations for deployment
   - Optimizations & tunings for scale & stability
   - Multi-cluster solution: KVStoreMesh
3. Advanced trouble shooting skills
   - Debugging/tracing
   - Manipulating low-level objects/resources
4. Summary

About Trip.com

About Trip.com

- A one-stop online travel agency
- Booking services for flights, hotels, etc
- Business in China & overseas
- 400 millions users worldwide

About the cloud team @trip.com

- R&D of cloud infra over the globe
- Virtualization, networking, storage, some security controls

1. Cloud infrastructure at Trip.com

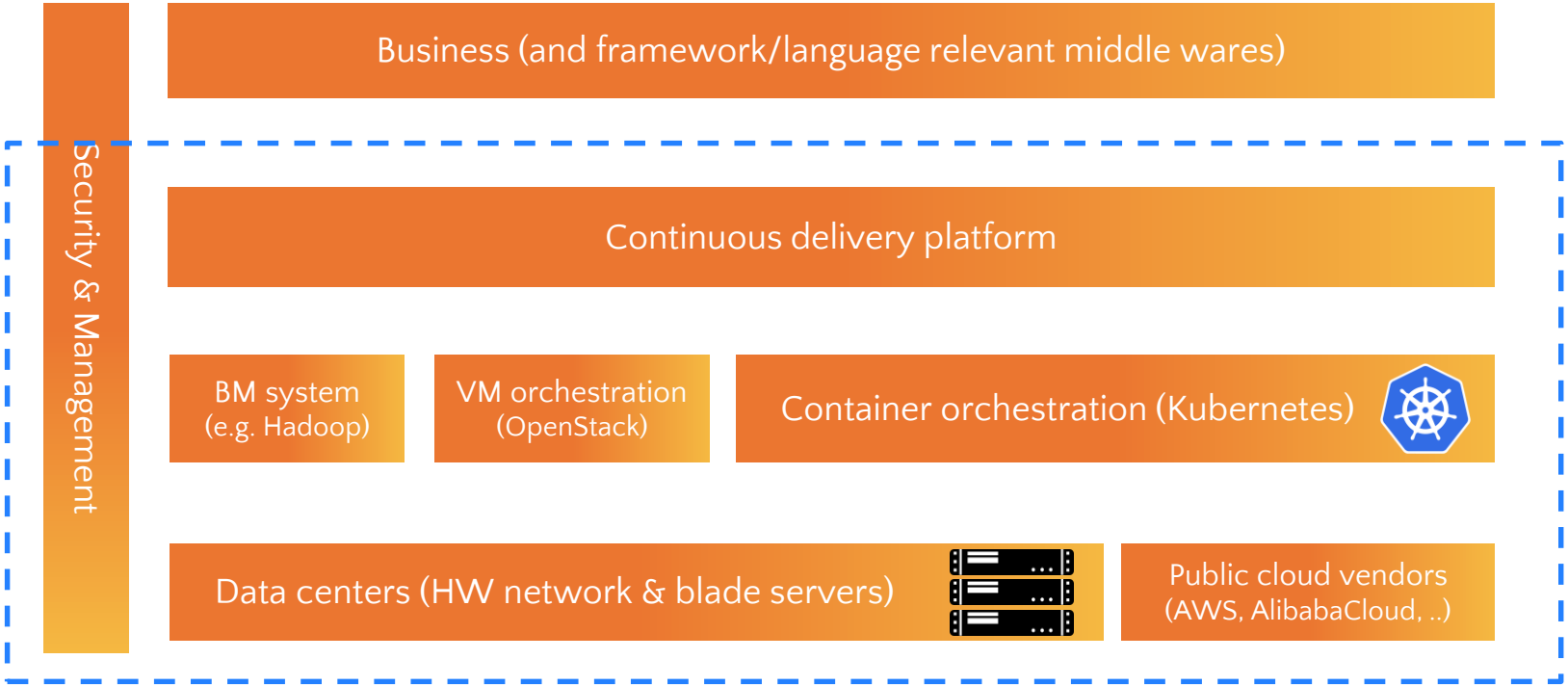
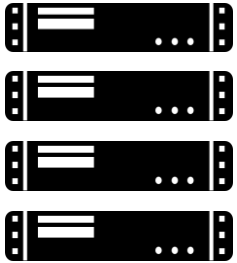


Fig. Cloud infrastructure @trip.com

1. Cloud infrastructure at Trip.com

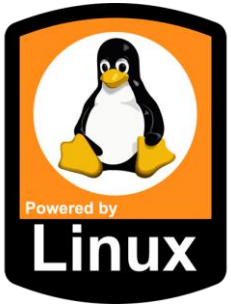
Kubernetes

- 3 big clusters
- N small clusters
- ~10K nodes
- ~300K pods



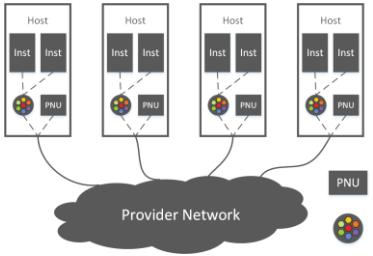
Flavor (blade servers)

- 128G/32C ~ 384G/64C
- 25Gbps NIC x 2, active-standby



Kernel

- 4.19: 60%
- 5.10: 40%
- Internal branches



Inter-host networking

- BGP
- ENI or similar stuffs

2. Cilium at Trip.com

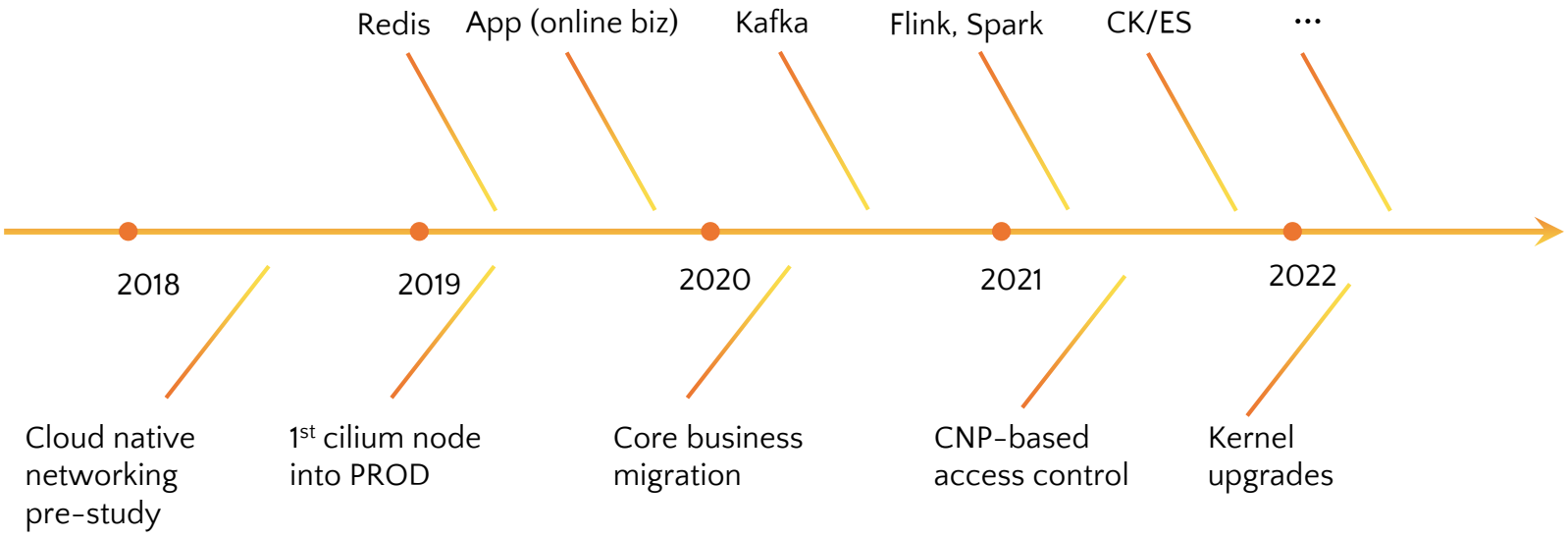


Fig. Rolling out of Cilium @trip.com

2. Cilium at Trip.com

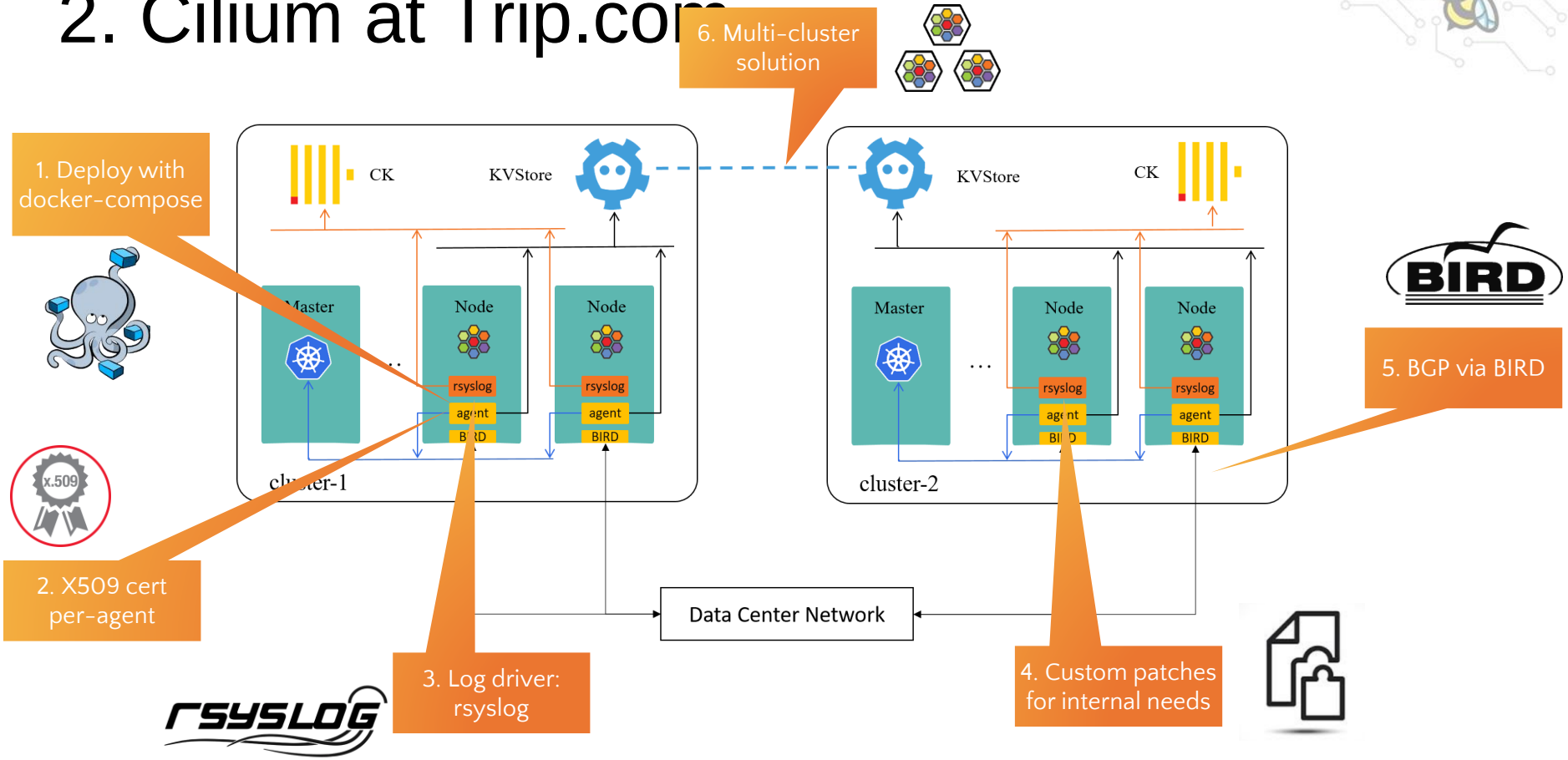
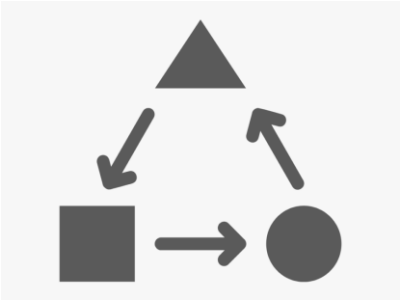


Fig. Some customizations at Trip.com

2. Cilium at Trip.com



Decouple deployment

Least K8s dependency
- No daemonset/configmap

Node turn-on: salt+st2

Benefits
- Suffer less on k8s problems
- Per-agent configuration
- Per-agent upgrade
- Per-agent debugging
- …

Avoid retry/starting storm

Central/control plane components
- kube-apiserver, kvstore, …

Requests & concurrency control
- Client-side: pessimistic restart backoff (jitter+backoff)
- Server-side: k8s APF based on per-agent certificate (username)

Business online, always

For really large biz clusters
- Fail late vs. fail fast
- Avoid restarts
- Favor human decision

Tuning parameters
- *--allocator-list-timeout*
- *--k8s-syn-timeout*
- *--kvstore-lease-ttl*
- *--k8s-heartbeat-timeout*
- …

Fig. Optimization & tuning considerations

2. Cilium at Trip.com

Planning for scale

- Cluster ID/name planning
- Identity labels/space
- Monitoring agg level
- KVStore nodes flavor

Tuning parameters:

- *--monitor-aggregation**
- *--bpf-ct-**

Considering performance

- Sockops
- BPF host routing
- XDP speedup
- Kernel versions

- *--disable-cpn-status-updates*
- *--api-rate-limit*

Observability & alerting

Metrics: VictoriaMetrics+Grafana

Logging: ClickHouse+Kibana

- Agent log
- Hubble flow (post-processing)

Tracing: on demand for TR

Alert on metrics & logging

Fig. Optimization & tuning considerations

2. Cilium at Trip.com

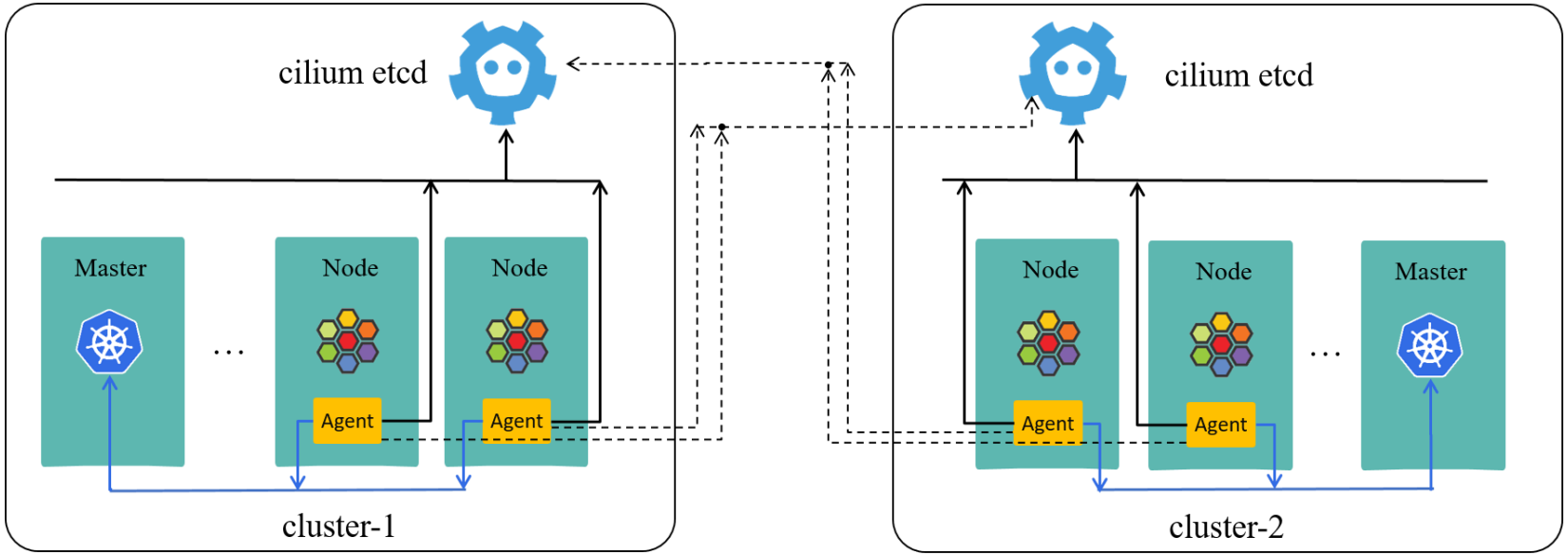


Fig. Multi-cluster solution from community: ClusterMesh

2. Cilium at Trip.com

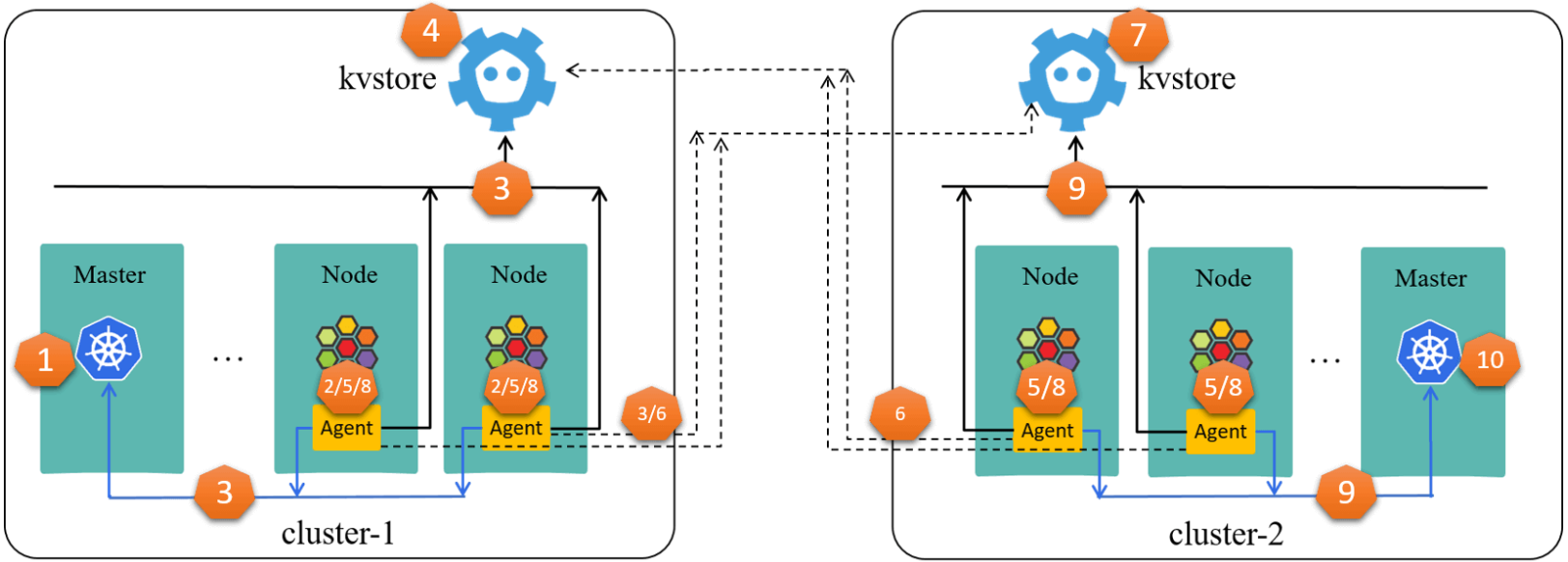
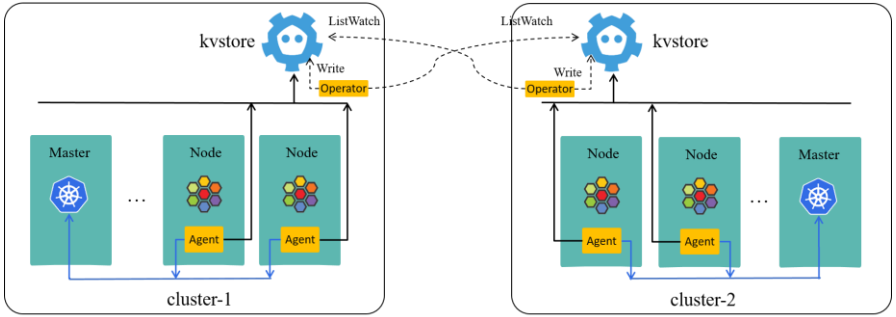


Fig. Stability & scalability: how ClusterMesh may propagate & exaggerate failures

2. Cilium at Trip.com



- KVStoreMesh vs. ClusterMesh
- CER vs. CEW

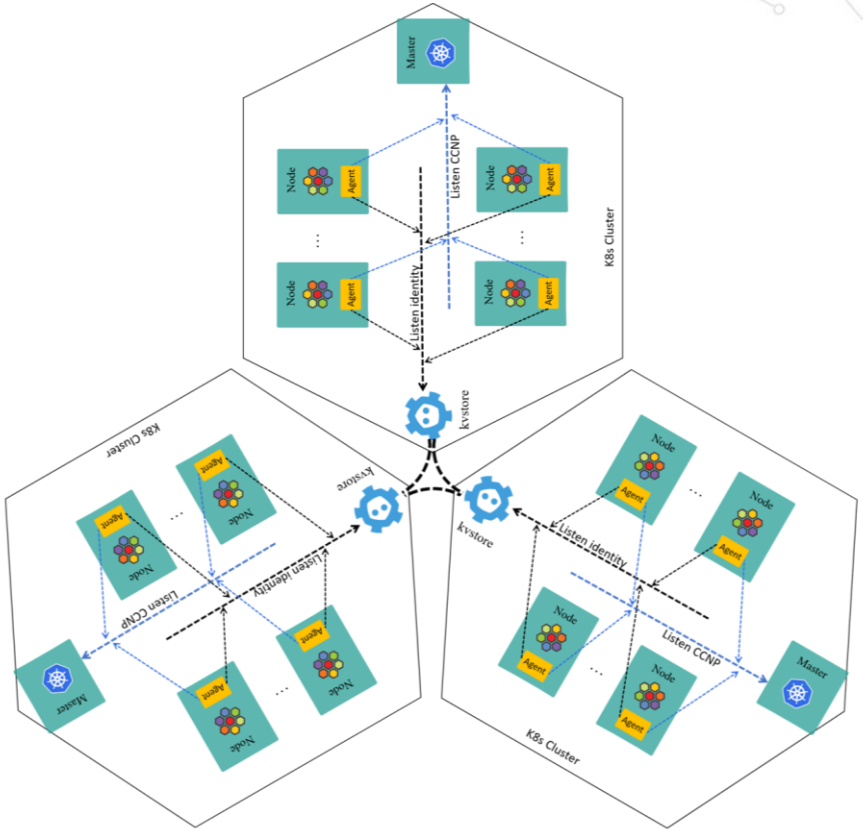


Fig. Multi-cluster solution at Trip.com: KVStoreMesh

3. Advanced trouble shooting skills

```
# Start cilium-agent agent container with entrypoint `sleep 10d`, then enter the container
(node) $ docker exec -it cilium-agent bash

(cilium-agent ctn) $ dlv exec /usr/bin/cilium-agent -- --config-dir=/tmp/cilium/config-map
Type 'help' for list of commands.
(dlv)

(dlv) break github.com/cilium/cilium/pkg/endpoint.(*Endpoint).regenerateBPF
Breakpoint 3 set at 0x1e84a3b for github.com/cilium/cilium/pkg/endpoint.(*Endpoint).regenerateBPF() /go/src/github.
(dlv) break github.com/cilium/cilium/pkg/endpoint/bpf.go:1387
Breakpoint 4 set at 0x1e8c27b for github.com/cilium/cilium/pkg/endpoint.(*Endpoint).syncPolicyMapWithDump() /go/src
(dlv) continue
...

(dlv) clear 1
Breakpoint 1 cleared at 0x1e84a3b for github.com/cilium/cilium/pkg/endpoint.(*Endpoint).regenerateBPF() /go/src/git
(dlv) clear 2
Breakpoint 2 cleared at 0x1e8c27b for github.com/cilium/cilium/pkg/endpoint.(*Endpoint).syncPolicyMapWithDump() /go
```

Fig. Debugging with delve/dlv in container

3. Advanced trouble shooting skills

```
$ nm /var/lib/docker/overlay2/0a26c6/merged/usr/bin/cilium-agent | grep regenerateBPF
0000000001e84a20 T github.com/cilium/cilium/pkg/endpoint.(*Endpoint).regenerateBPF

$ bpftrace -e \
  'uprobe:<path>/cilium-agent:"github.com/cilium/cilium/pkg/endpoint.(*Endpoint).regenerateBPF" {printf("%s\n", ustack);}'
Attaching 1 probe...

        github.com/cilium/cilium/pkg/endpoint.(*Endpoint).regenerateBPF+0
        github.com/cilium/cilium/pkg/endpoint.(*EndpointRegenerationEvent).Handle+1180
        github.com/cilium/cilium/pkg/eventqueue.(*EventQueue).run.func1+363
        sync.(*Once).doSlow+236
        github.com/cilium/cilium/pkg/eventqueue.(*EventQueue).run+101
        runtime.goexit+1
```

Fig. Tracing cilium-agent process with bpftrace

3. Advanced trouble shooting skills

How could you determine if a CNP actually takes effect?

```
$ kubectl get cnp -n <ns> <cnp> -o yaml        # spec & status in k8s
$ cilium endpoint get <ep id>                  # spec & status in cilium userspace
$ cilium bpf policy get <ep id>                # summary of kernel bpf policy status
$ bpftool map dump pinned cilium_policy_00794 # REAL & ULTIMATE policies in the kernel!
```

```
# Key format: identity(4B) + port(2B) + proto(1B) + direction(1B)
# For endpoint 794's TCP/80 ingress, check if allow traffic from identity 298951
$ printf '%08x' 298951
00048fc7
$ bpftool map dump pinned cilium_policy_00794 | grep "c7 8f 04 00" -B 1 -A 3
key:
c7 8f 04 00 00 50 06 00 # 4B identity + 2B port(80) + 1B L4Proto(TCP) + direction(ingress)
value:
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Fig. Manipulating BPF policies with bpftool

3. Advanced trouble shooting skills

```
# Add an allow-any rule in emergency if the agent has already down
$ bpftool map update pinned <map> \
  key hex 00 00 00 00 00 00 00 00 \
  value hex 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 noexist
```

Fig. Manipulating BPF policies with bpftool (continued)

3. Advanced trouble shooting skills

```
$ etcdctl put "cilium/state/identities/v1/id/15614229" \
  'k8s:app=app1;k8s:io.cilium.k8s.policy.cluster=cluster1;k8s:io.cilium.k8s.policy.serviceaccount=default;k8s:io.

$ etcdctl put 'k8s:app=app1;k8s:io.cilium.k8s.policy.cluster=cluster1;k8s:io.cilium.k8s.policy.serviceaccount=def
  15614229

$ etcdctl put "cilium/state/ip/v1/cluster1/10.3.192.65" \
  '{"IP":"10.3.192.65","Mask":null,"HostIP":"10.3.9.10","ID":15614299,"Key":0,"Metadata":"cilium-global:cluster1:
```

Fig. Manipulating KVStore contents with etcdctl (!BE CAUTIOUS!)

All *agents* will be notified that there is a *pod* in Kubernetes cluster *cluster1* and namespace *default*, with PoIP *10.3.192.65* on *node1* with NodeIP *10.3.9.10*; besides, *pod label and identity* information also included.

- CER: inject VM/non-cilium-pods metadata into Cilium
- Foundation of cilium network policy

4. Summary

Cilium at Trip.com

- Upgrade 1.4 -> 1.5 -> … -> 1.10 (-> 1.11 planned)
- Support business & infra critical services
- Missing technical details of this talk: http://ctripcloud.github.io/

Cilium/eBPF: production ready for large scale

- Excellent performance, feature, release velocity
- Very nice community
- Leading eBPF contributions & promotions

Special thanks to (alphabetical order) Andre, Daniel , Joe, Martynas, Paul, Quentin, Thomas, … all the Cilium guys!

References

1. https://ctripcloud.github.io/
2. https://github.com/ctripcloud/cilium-compose
3. https://docs.google.com/document/d/1Zc8Sdhp96yKSeC1-71_6qd97HPWQv-L4kiBZhl7swrg/edit#
4. https://arthurchiao.art/blog/cilium-clustermesh/
5. https://arthurchiao.art/blog/whats-inside-cilium-etcd/
6. https://arthurchiao.art/blog/cracking-k8s-network-policy/