



Hijacking Service Discovery To Simulate Dependency Degradation

Abdulrahman Alhamali





8000+ employees



Millions of
merchants in 175
countries



1+ Trillion USD total Gross Merchandise Value (GMV)

Last BFCM:

- 11.5\$ Billions GMV
- 170M average RPM
- 280M peak RPM

Resiliency SRE Team



A horizontal bar with a teal segment on the left and an orange segment on the right.

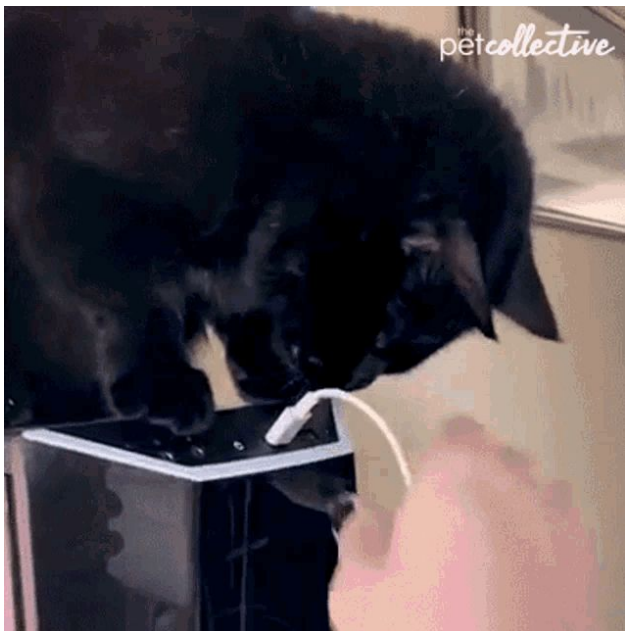
Gamedays

	Database down	Redis down	Lack of web workers
Authentication	Down	Degraded	Prioritized
User profile display	Down	Down	Shed

Gamedays



Should you actually shut off the dependency?



Pros

- More realistic

Cons

- Could affects multiple services
- Abstraction might be better for feature teams

A short horizontal bar with a teal left half and an orange right half.

Fault Injection Proxies



Toxiproxy



Envoy

Fault Injection Proxies

Toxiproxy

```
Toxiproxy[:mysql_master].downstream(:latency, latency: 1000).apply
  Shop.first # this takes at least 1s
end
```

Envoy

```
http_filters:
- name: envoy.filters.http.fault
  typed_config:
    "@type": type.googleapis.com/envoy.config.filter.http.fault.v2.HTTPFault
    delay:
      #type: fixed
      fixed_delay: 10s
      percentage:
        numerator: 50
        denominator: HUNDRED
```



But how to send traffic through that proxy?

- Change the code? The configuration files? Both?
- Generate a different image? Different kubernetes manifests? Both?
- Asking devs to figure it out adds too much friction

A horizontal bar with a teal segment on the left and an orange segment on the right.

GamedayBuddy

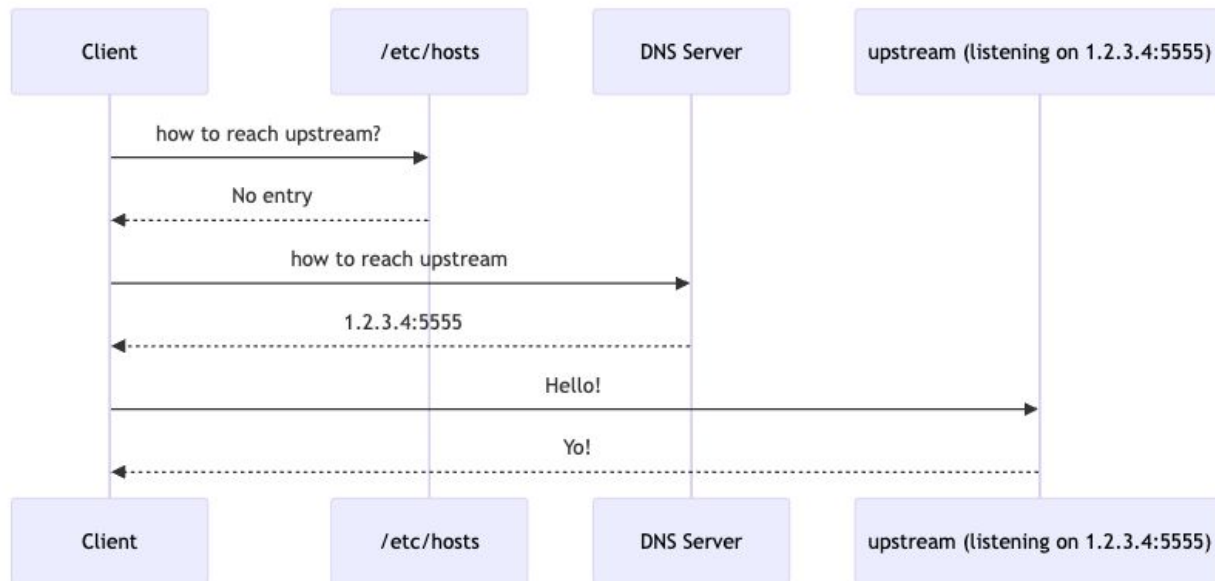
```
kind: GamedayConfig
apiVersion: stable.shopify.io/v1
metadata:
  name: myapp-gameday
  namespace: myapp-production-unrestricted
spec:
  deployments:
  - web
  dependencies:
  - name: myredis
    type: redis
    upstream: myredis.mynamespace.svc.cluster.local
  toxics:
  - name: mylatency
    type: latency
    enabled: false
    stream: upstream
    attributes:
      latency: 1000
      jitter: 250
```

A short horizontal bar with a teal left half and an orange right half.

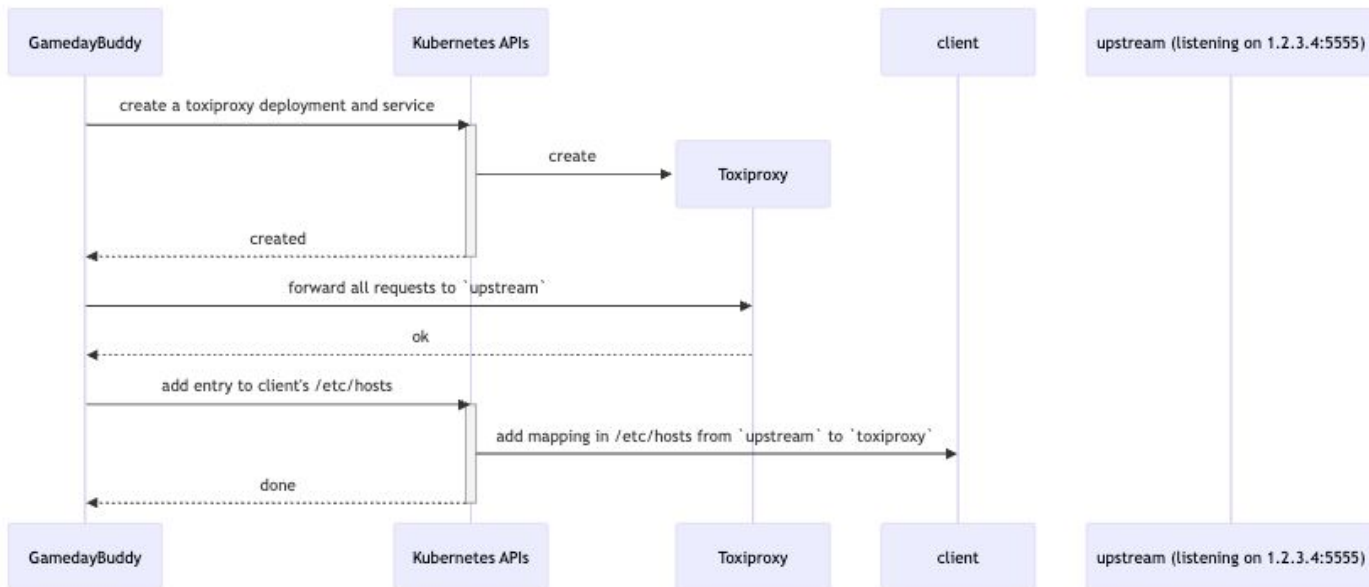
How to hijack service discovery?

- Locally
- Globally
- On the way?

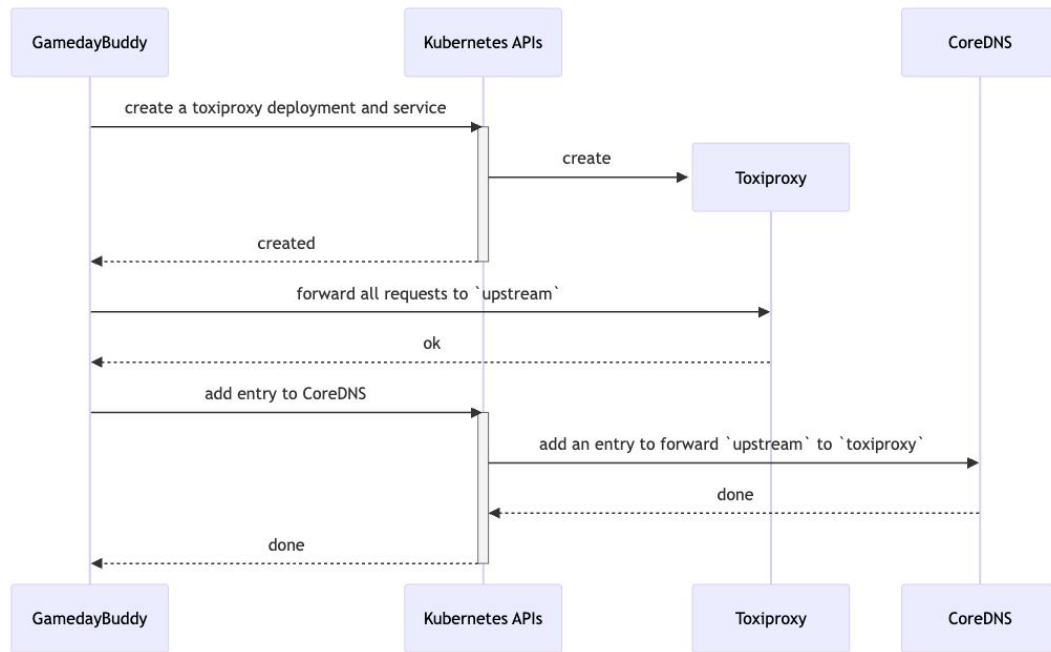
Locally: example using hostAliases



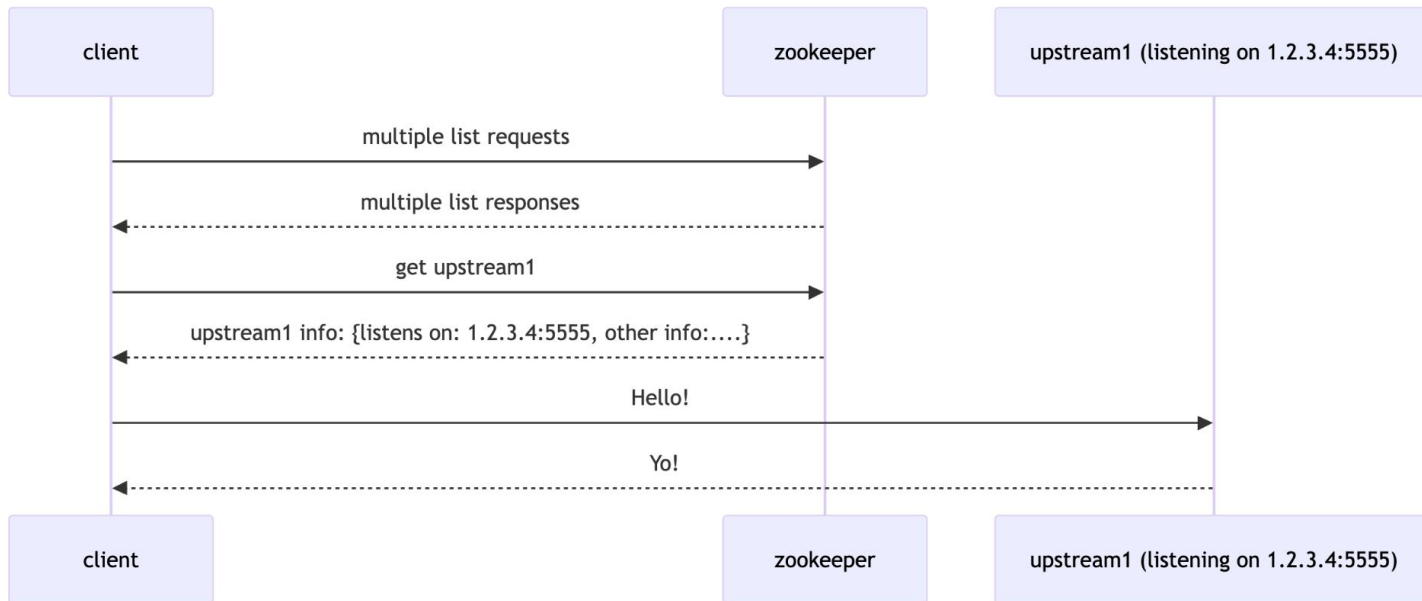
Locally: example using hostAliases



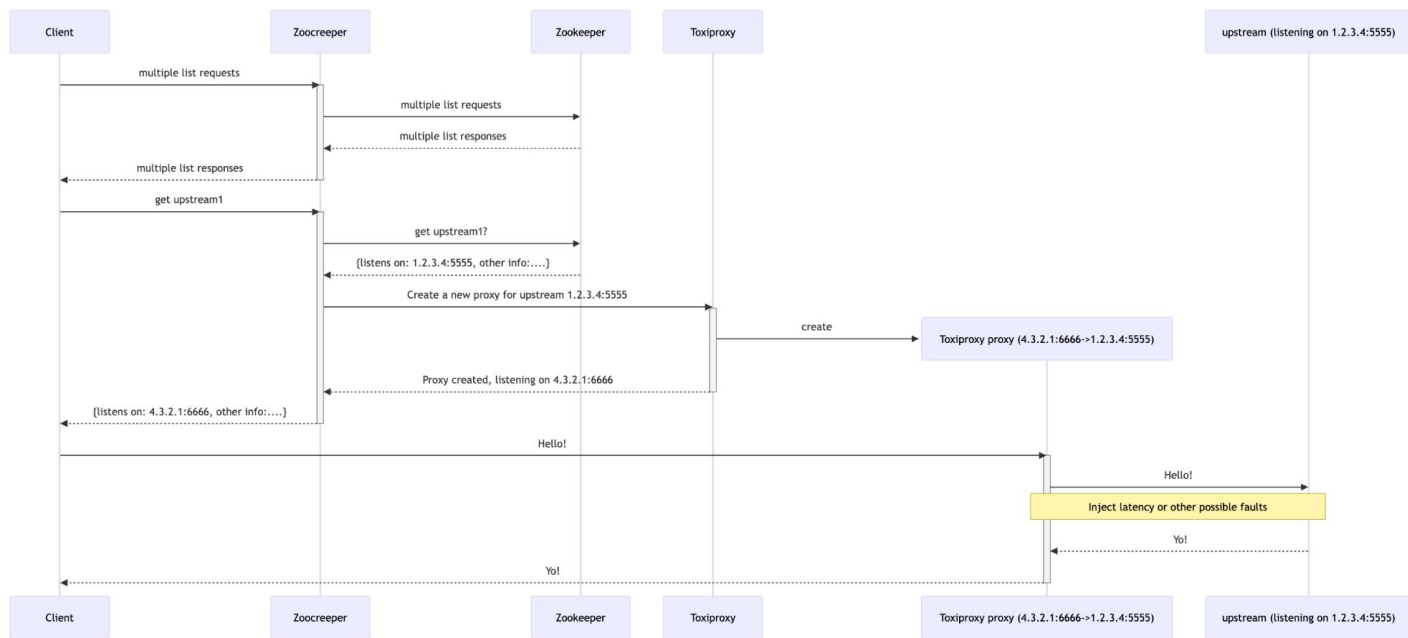
Globally: Example using CoreDNS



On the way: Example using Zookeeper



On the way: Example using Zookeeper





```
{
  "listen_port": 1234,
  "upstream": "some-zookeeper.com:2181",
  "overrides": [
    {
      "kind": "VitessTabletOverride",
      "details": {
        "cluster": "carts",
        "cell": "us-central1",
        "tablet": "us-central1-0000005999",
        "use_toxiproxy_override": true
      }
    }
  ]
}
```

```
{
  "listen_port": 1234,
  "upstream": "some-zookeeper.com:2181",
  "overrides": [
    {
      "kind": "VitessTabletOverride",
      "details": {
        "cluster": "carts",
        "cell": "us-central1",
        "use_toxiproxy_override": true
      }
    }
  ]
}
```

A short horizontal bar with a teal segment on the left and an orange segment on the right.

Summary

- Running gamedays is important
- But to get teams to run them, you need to provide the right tooling to decrease friction and to empower them
- Service discovery hijacking (local, global, and on-the-way) helps greatly with that