



# Dig Smart: Creating A Reliable Cloud-Native DNS Service

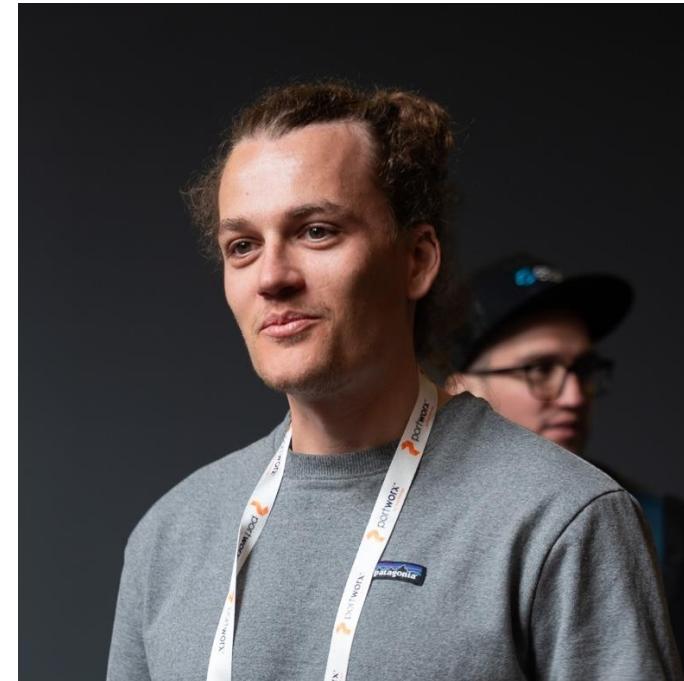
Joel Studler & Fabian Schulz





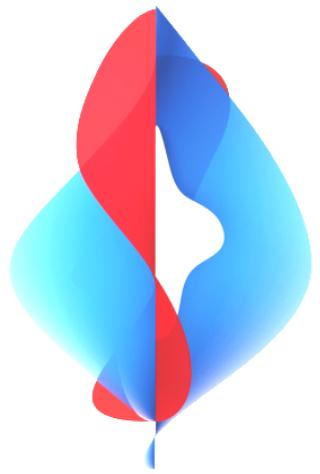
**Joel Studler**  
Senior DevOps Engineer

[joel.studler@swisscom.com](mailto:joel.studler@swisscom.com)



**Fabian Schulz**  
DevOps Engineer

[fabian.schulz1@swisscom.com](mailto:fabian.schulz1@swisscom.com)



**swisscom**



## Context & Related Talks

### **5G - driving our journey from Telco to TechCo**

*by Swisscom CTIO Mark Düsener at Connect Conference 2022*

<https://www.youtube.com/watch?v=hND7TiXJED8>

### **Evolving GitOps: Harnessing Kubernetes Resource Model for 5G**

*by Ashan Senevirathne and Joel Studler at Open Source Summit 2024*

[https://www.youtube.com/watch?v=35-fE\\_gHDjw](https://www.youtube.com/watch?v=35-fE_gHDjw)

### **How We Are Moving from GitOps to Kubernetes Resource Model in 5G Core**

*by Ashan Senevirathne and Joel Studler at KubeCon Europe 2024*

<https://www.youtube.com/watch?v=crmTnB6Zwt8>



## DNS in 5G Core

**5G**

### Specific Private Zones

Domains used in Mobile Network only such as 3gppnetwork.org



### Moderate Throughput

10s to 100s of Requests/second



### Low Latency

DNS is an important factor in the overall performance of the Mobile Network



# Requirements for the 5G Core DNS Service



## Proximity to Consumer

Minimal amount of hops between  
5G Core and DNS

X No SaaS allowed



## Fully Automated

GitOps driven and automated  
provisioning of DNS records

X No manual interaction allowed



## Geo Redundant & HA

Spread across multiple K8s clusters and  
geo regions to increase reliability

X No singletons



## Support of Advanced DNS features

Resource Records such as NAPTR and  
SRV supported for e.g. SIP Phone Calls

X Need to go beyond A and CNAME



## K8s integration with ExternalDNS

The System leverages Kubernetes  
Patterns such as CRs and Operators

X No CRUD outside kube-api



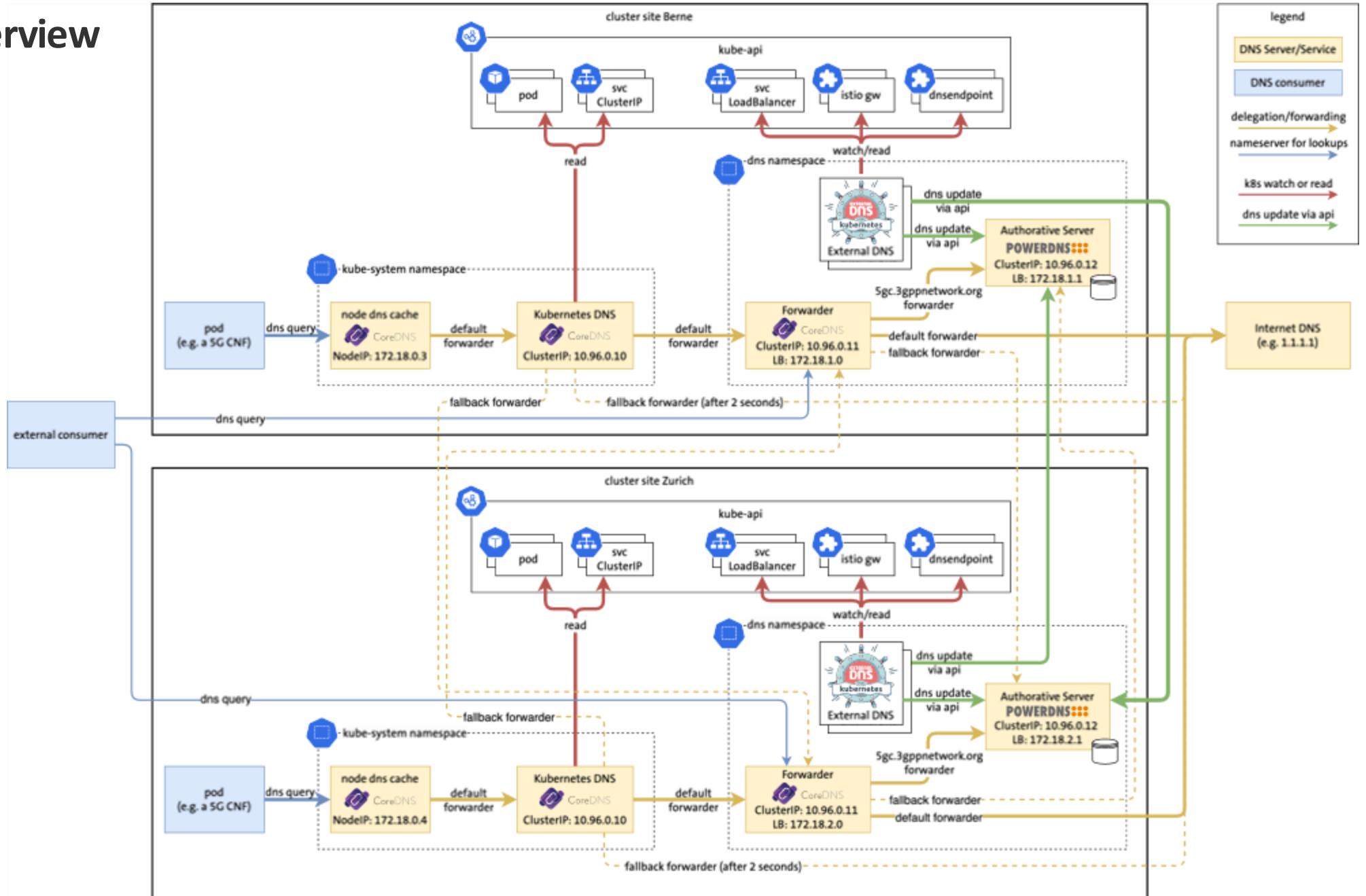
## Minimal Amount of SPOFs

Share nothing by removing single points  
of failure from the System

X No shared mgmt system

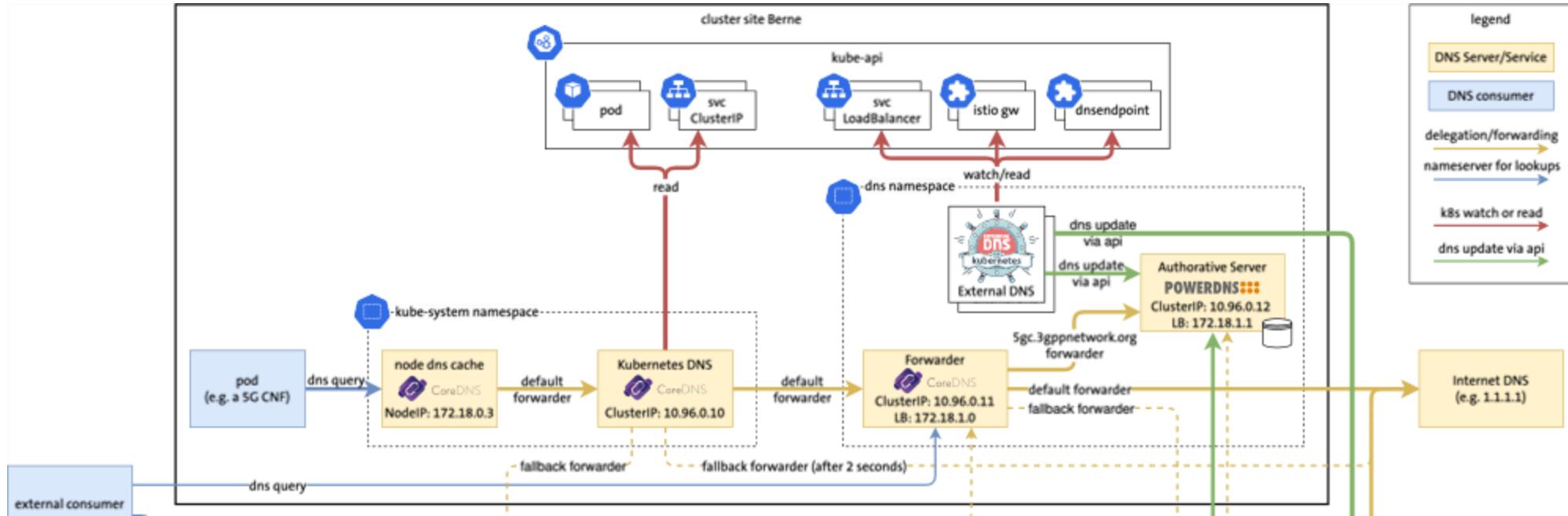


# Overview





# Overview





# In-Cluster Service Discovery in Kubernetes

CoreDNS (<https://coredns.io>)

kube-api as Backend

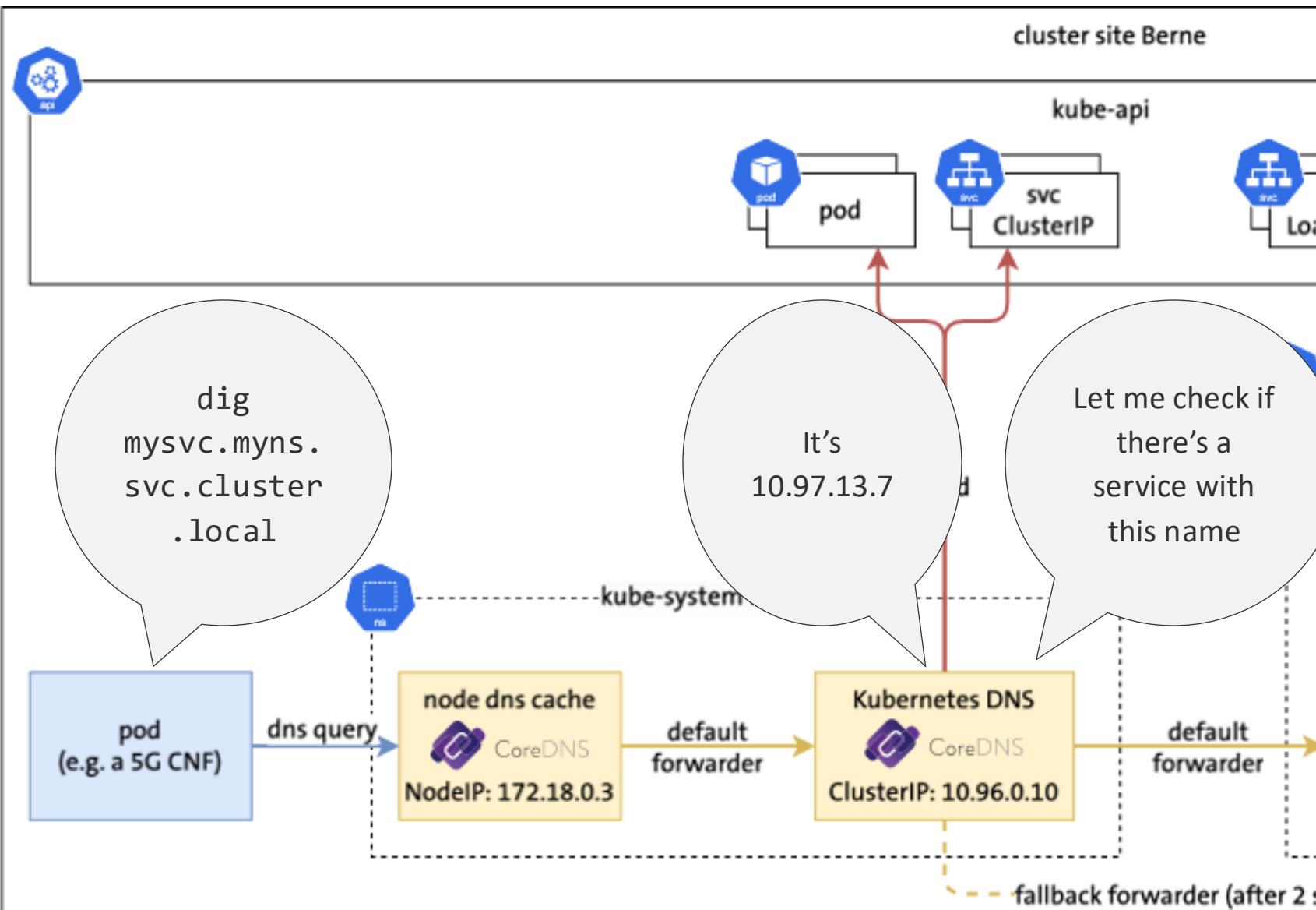
Features:

✓ In-Cluster Service Discovery

Missing:

X Not exposed outside of K8s

X No custom Resource Records





# In-Cluster Service Discovery in Kubernetes: Resources

Kubernetes DNS: <https://kubernetes.io/docs/concepts/services-networking/dns-pod-service>

Reserved ClusterIP Address assignment: <https://kubernetes.io/docs/concepts/services-networking/cluster-ip-allocation/#why-do-you-need-to-reserve-service-cluster-ips>

Node Cache: <https://kubernetes.io/docs/tasks/administer-cluster/nodelocaldns>

Debugging Kubernetes DNS: <https://kubernetes.io/docs/tasks/administer-cluster/dns-debugging-resolution>

Customize DNS Service: <https://kubernetes.io/docs/tasks/administer-cluster/dns-custom-nameservers>



# Requirements for Authoritative Server

## Requirement

ExternalDNS\* Support for K8s integration

A & CNAME Resource Records

NAPTR Resource Records (e.g. for SIP phone calls)

Proximity to Consumer



CoreDNS

CoreDNS

POWERDNS



POWERDNS

Authoritative



SaaS



✓

✓

✓

✓

✓

✓

✗

✓\*\*

✓\*\*

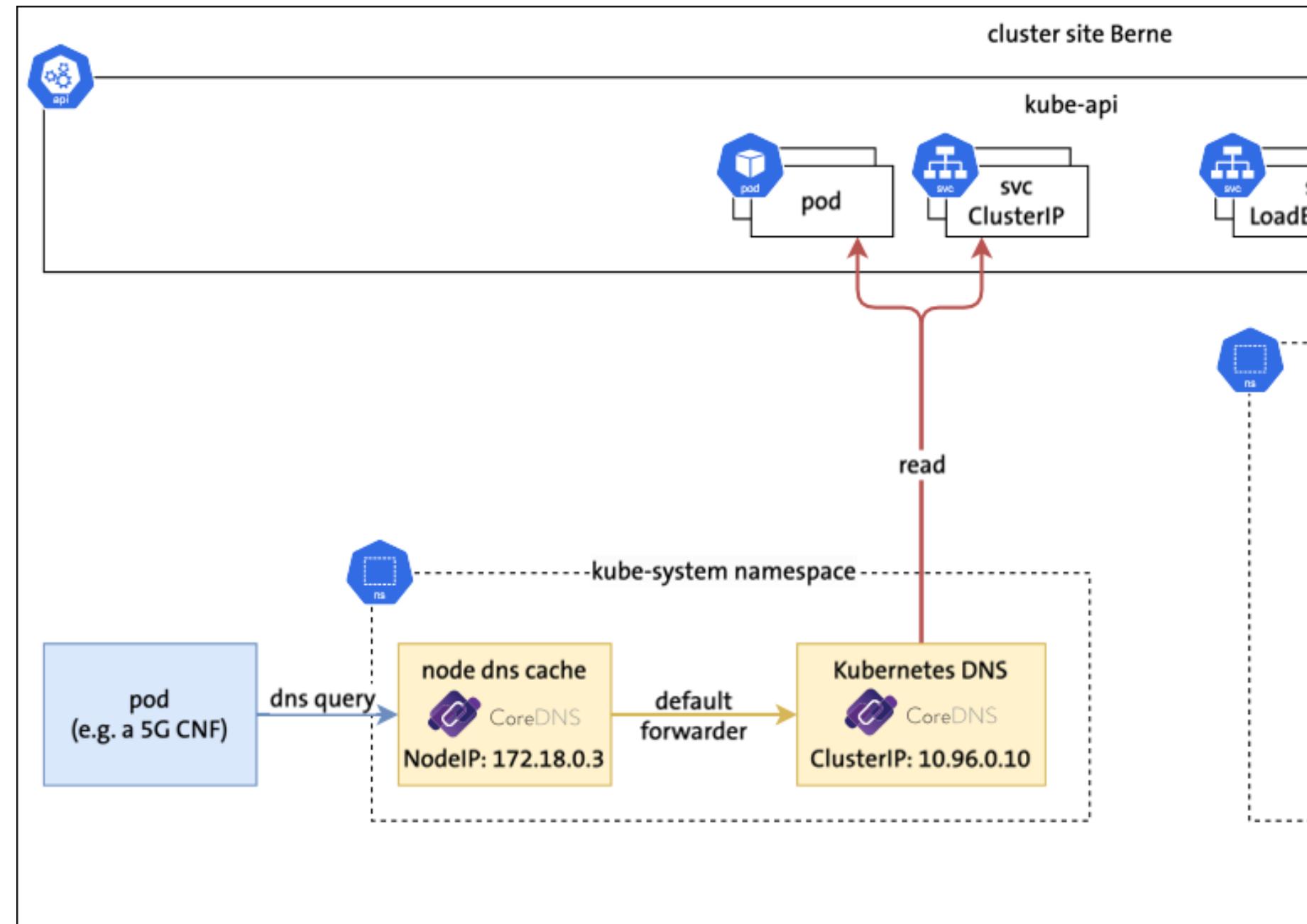
✓

✓

✗

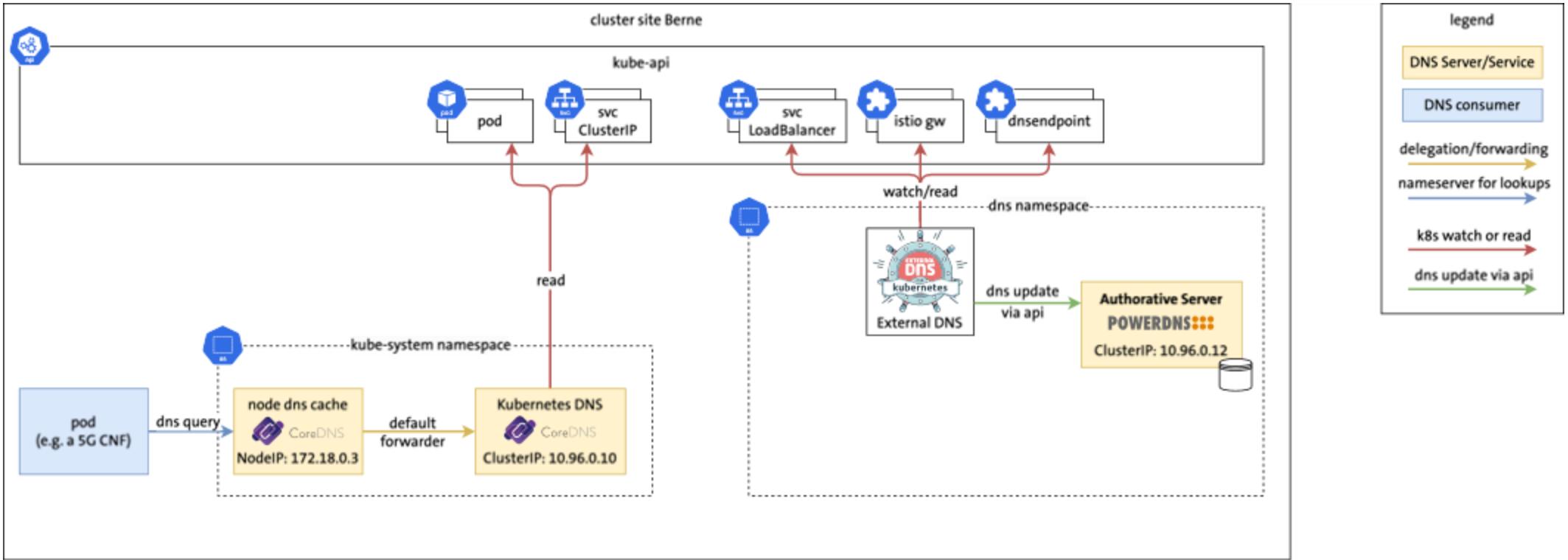
\* <https://github.com/kubernetes-sigs/external-dns>

\*\* After a fix in external-dns <https://github.com/kubernetes-sigs/external-dns/pull/4212>





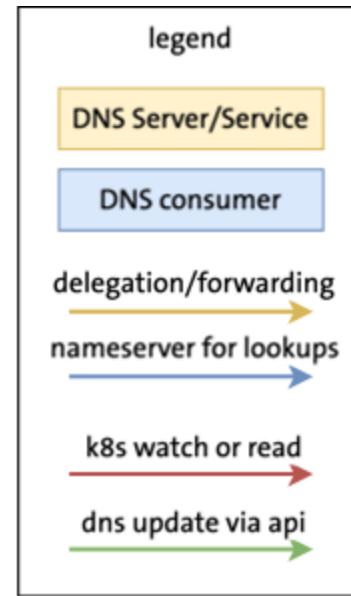
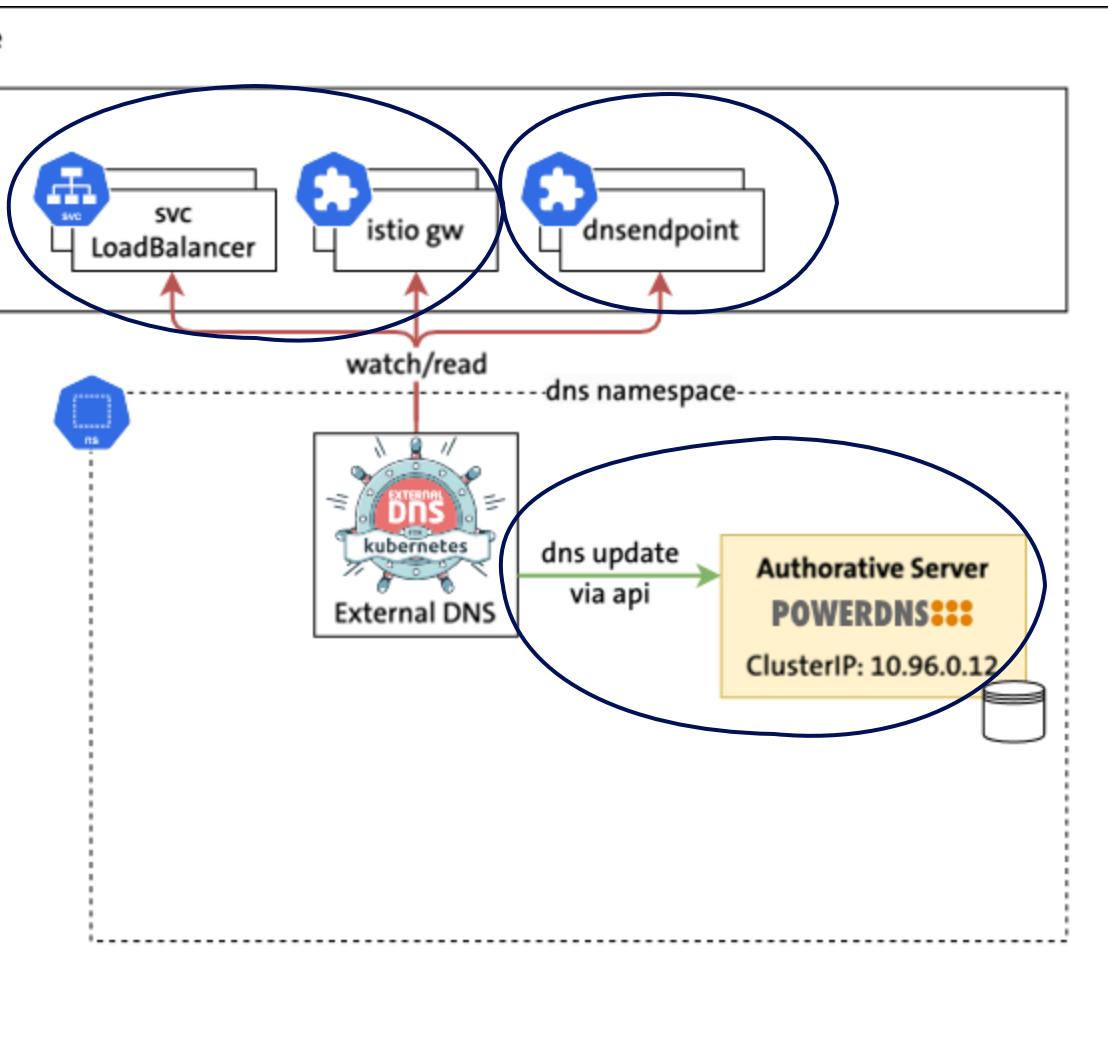
# Overview





# Automation of Authoritative Server Using ExternalDNS

erne



ExternalDNS reads from kube-api:

- Static Resource Records as DNSEndpoint Custom Resources
- Dynamic Type A Records using Annotations
  - Name definition via Annotation
  - IP fetched from Service / Ingress / Istiogw status field

ExternalDNS writes to PowerDNS API



# ExternalDNS State Management for Dynamic IP Assignment: GitOps + Kubernetes



```
apiVersion: v1
kind: Service
metadata:
  annotations:
    external-dns.alpha.kubernetes.io/hostname: my-app.example.com
  name: my-app
spec:
  ports:
    - name: http
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    name: my-app
  type: LoadBalancer
```

DNS Name  
defined in git



```
apiVersion: v1
kind: Service
metadata:
  annotations:
    external-dns.alpha.kubernetes.io/hostname: my-app.example.com
  name: my-app
spec:
  ports:
    - name: http
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    name: my-app
  type: LoadBalancer
status:
  loadBalancer:
    ingress:
      - ip: 192.168.0.35
```

IP read by  
ExternalDNS

DNS Backend:  
my-app.example.com. 3600 IN A 192.168.0.35

ExternalDNS creates



# ExternalDNS State Management for Static Assignment: GitOps



```
apiVersion: externaldns.k8s.io/v1alpha1
kind: DNSEndpoint
metadata:
  name: my-mx-record
spec:
  endpoints:
    - dnsName: my-app.example.com
      recordTTL: 3600
      recordType: A
      targets:
        - 192.168.0.35
    - dnsName: example.com
      recordTTL: 3600
      recordType: MX
      targets:
        - 10 mailhost1.example.com
        - 20 mailhost2.example.com
```

Resource  
Records defined  
in git



```
apiVersion: externaldns.k8s.io/v1alpha1
kind: DNSEndpoint
metadata:
  name: my-mx-record
spec:
  endpoints:
    - dnsName: my-app.example.com
      recordTTL: 1
      recordType: A
      targets:
        - 192.168.0.35
    - dnsName: example.com
      recordTTL: 180
      recordType: MX
      targets:
        - 10 mailhost1.example.com
        - 20 mailhost2.example.com
```

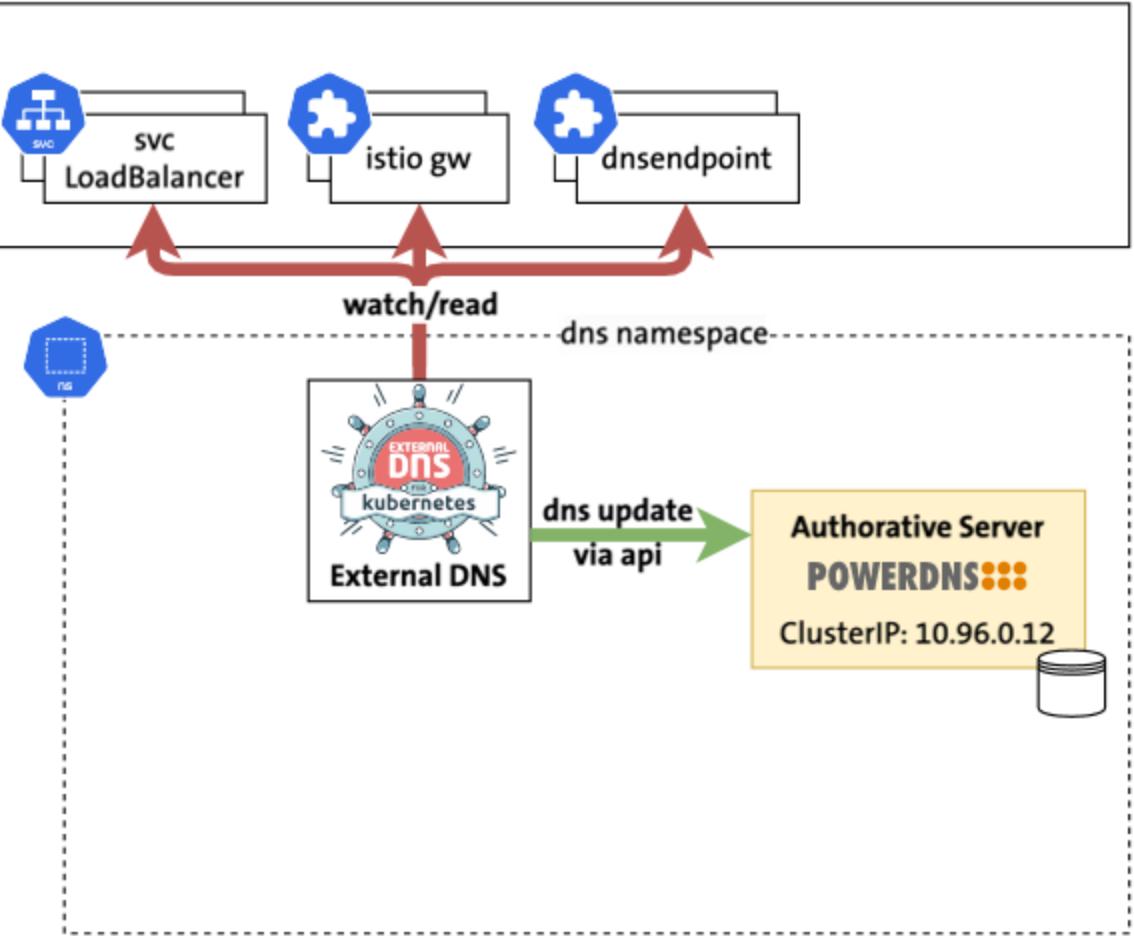
DNS Backend:

my-app.example.com.	3600 IN A	192.168.0.35
example.com.	3600 IN MX 10	mailhost1.example.com
example.com.	3600 IN MX 20	mailhost2.example.com

ExternalDNS creates



Berne

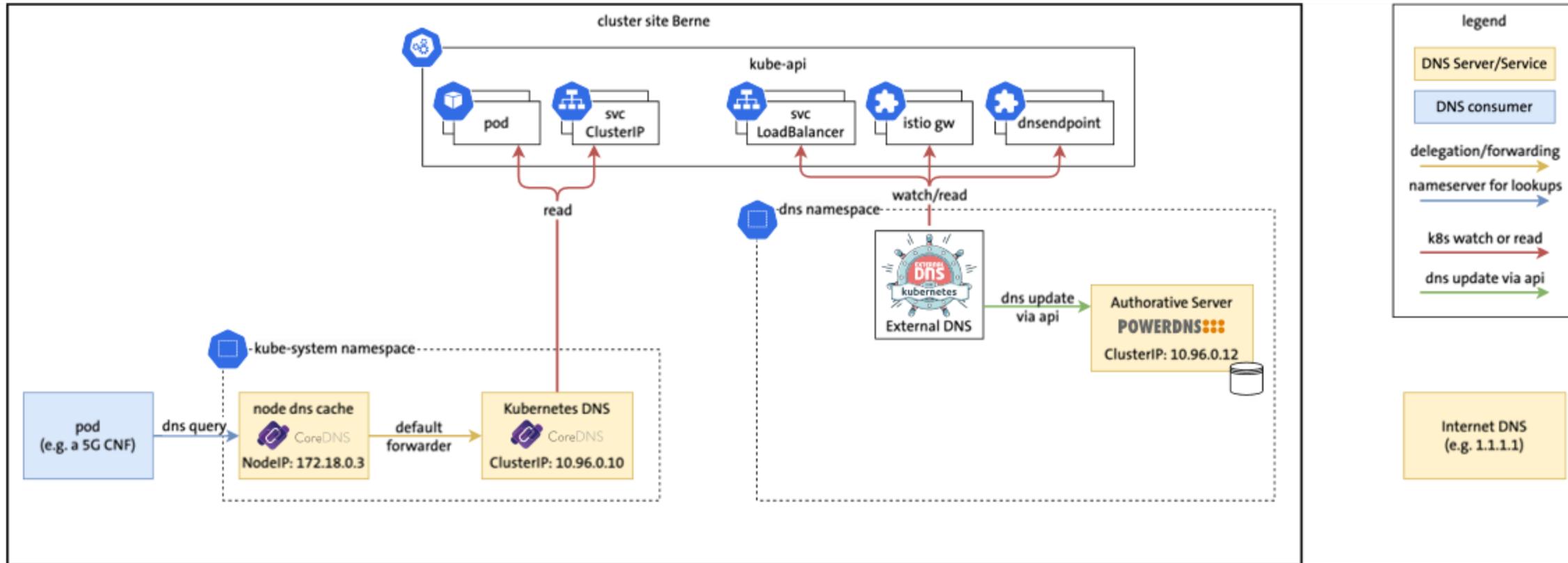


# Demo ExternalDNS + PowerDNS Single Cluster



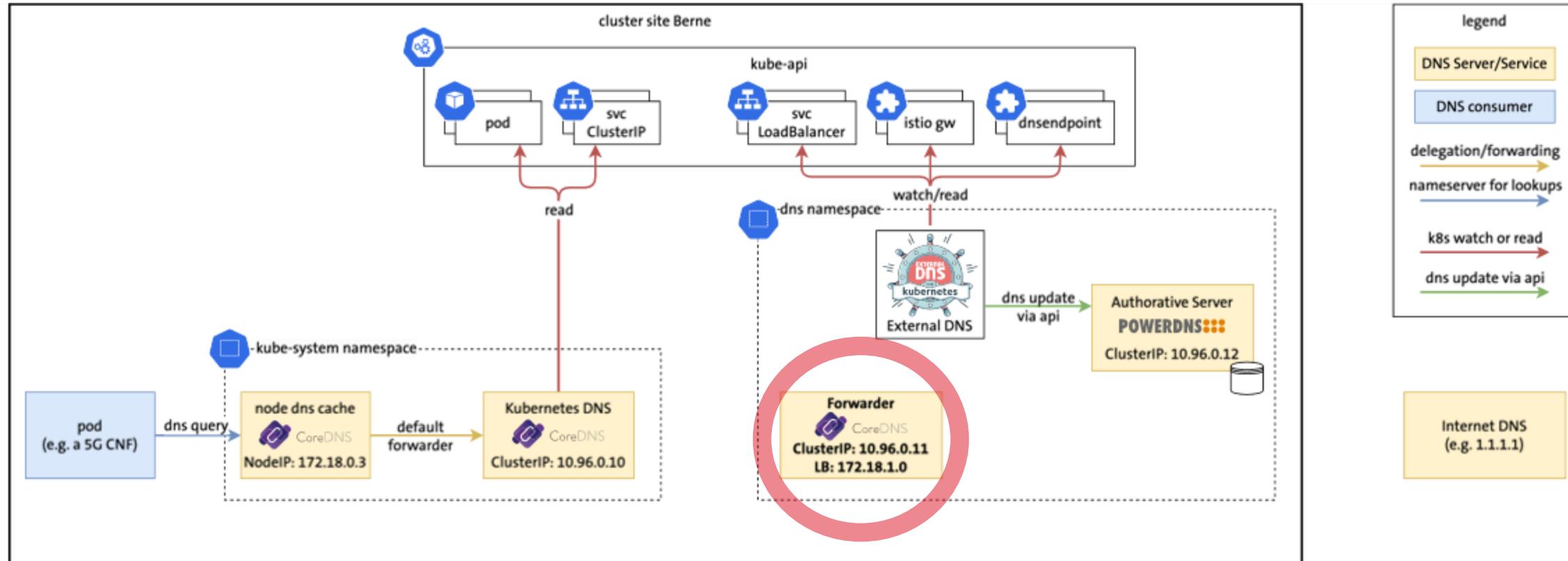


# Forwarding to Authoritative Server



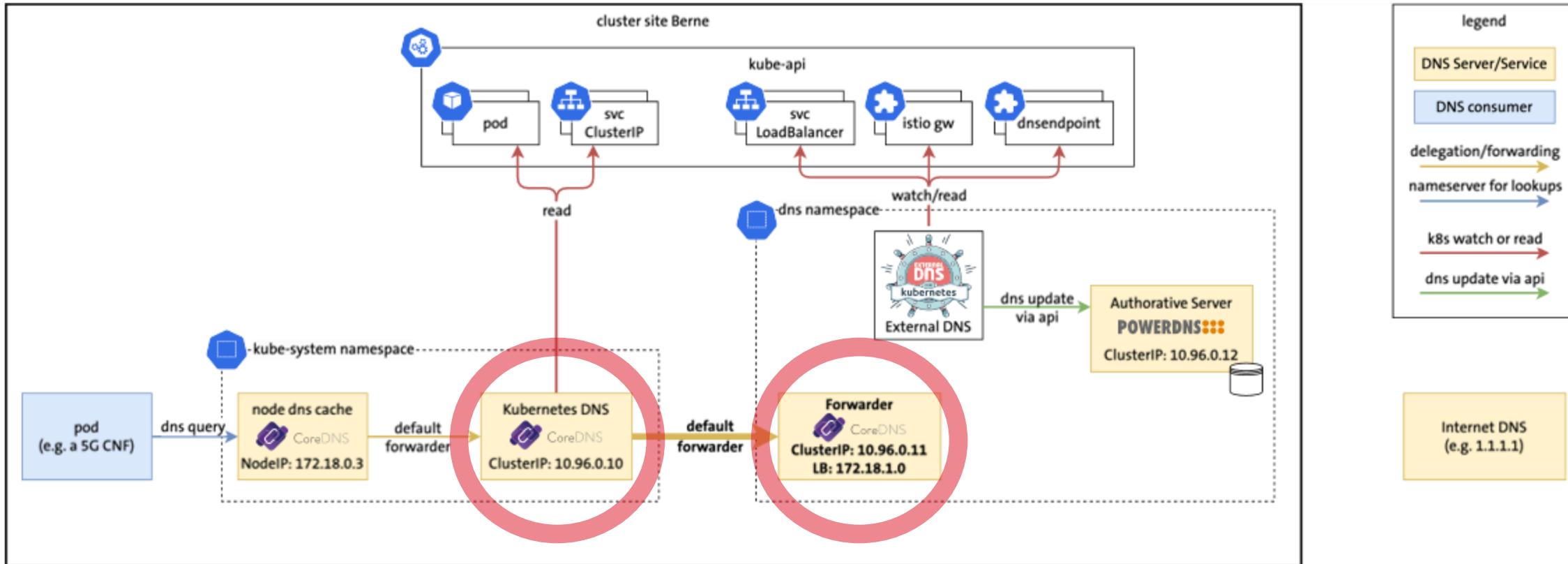


# Forwarding to Authoritative Server



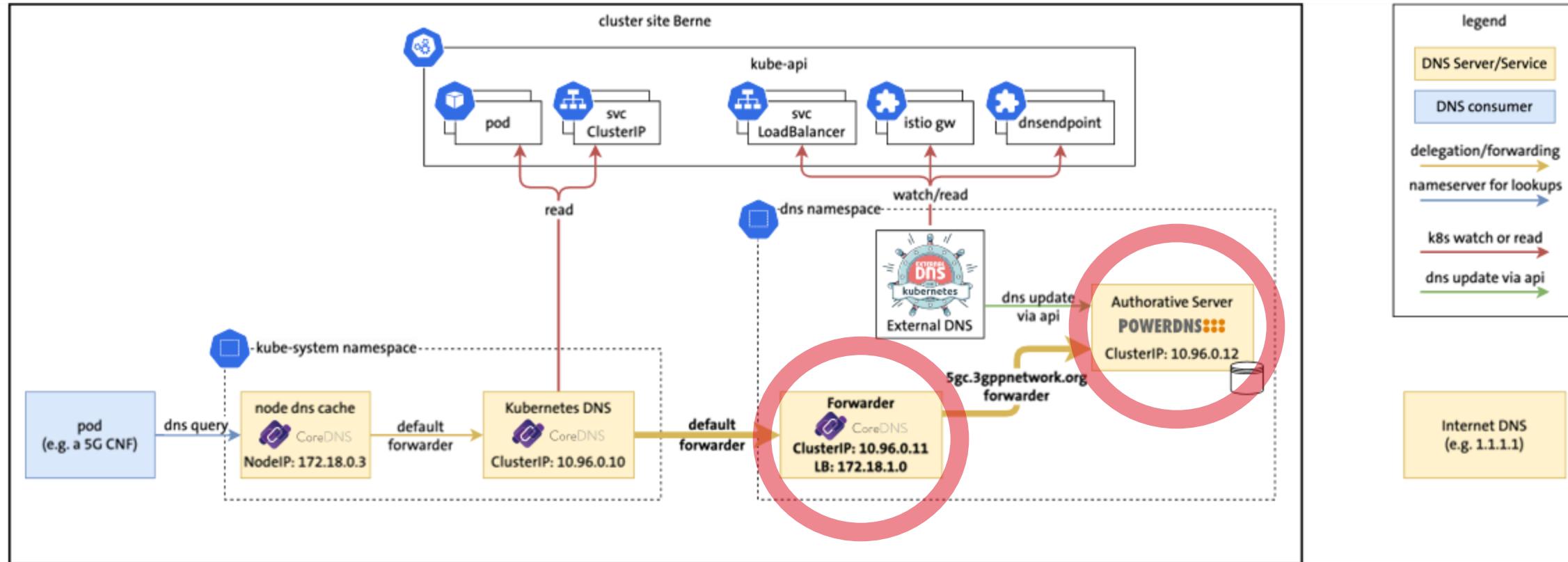


# Forwarding to Authoritative Server



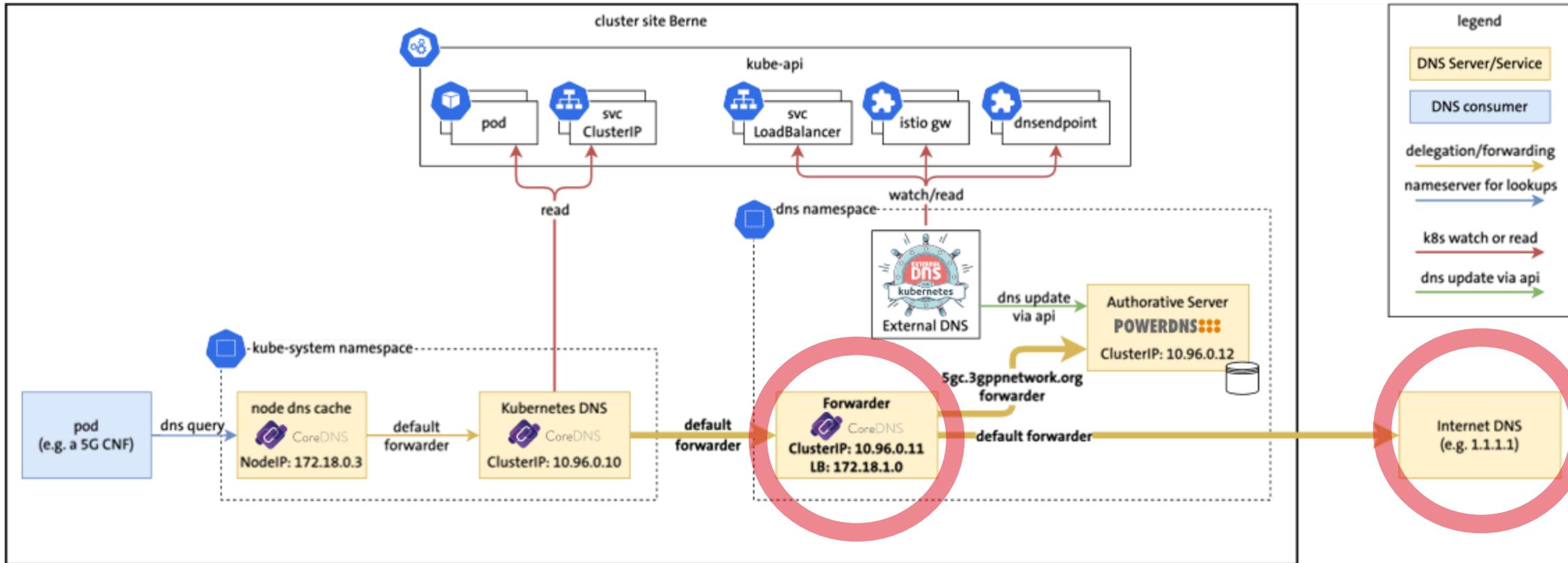


# Forwarding to Authoritative Server



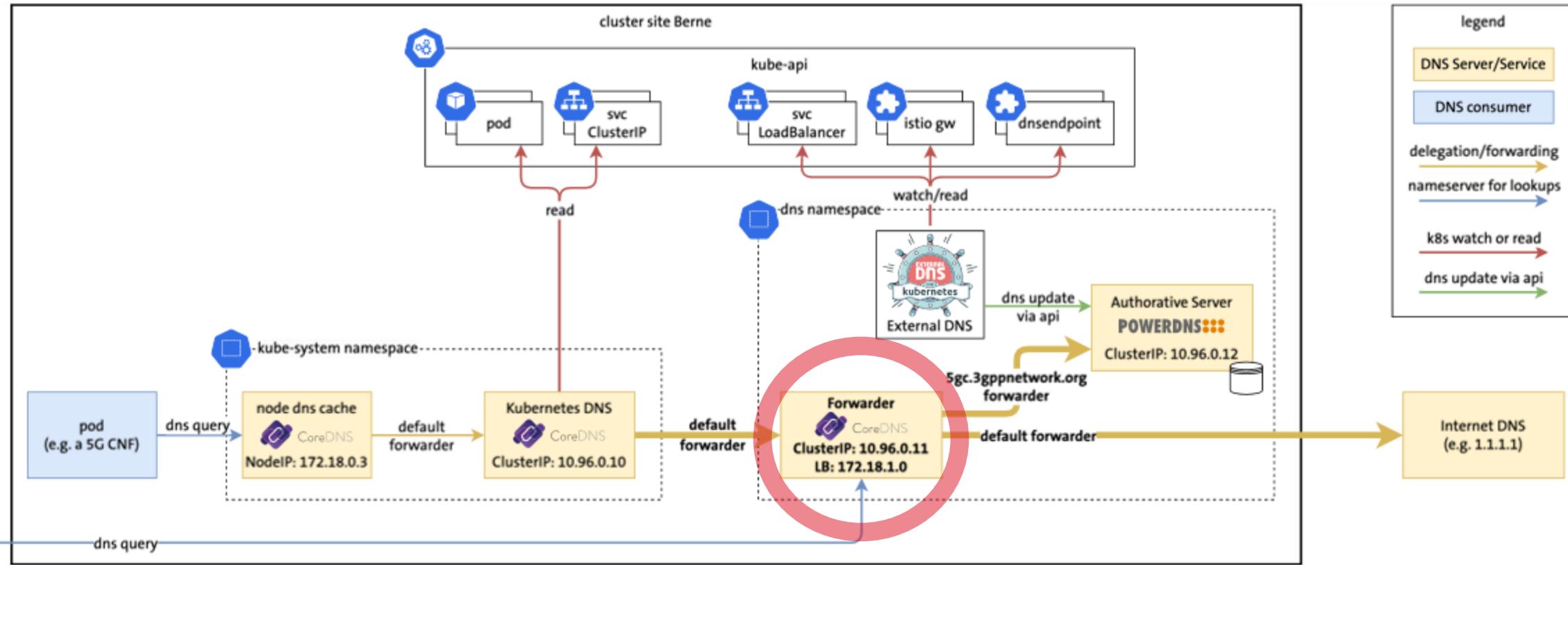


# Forwarding to Authoritative Server

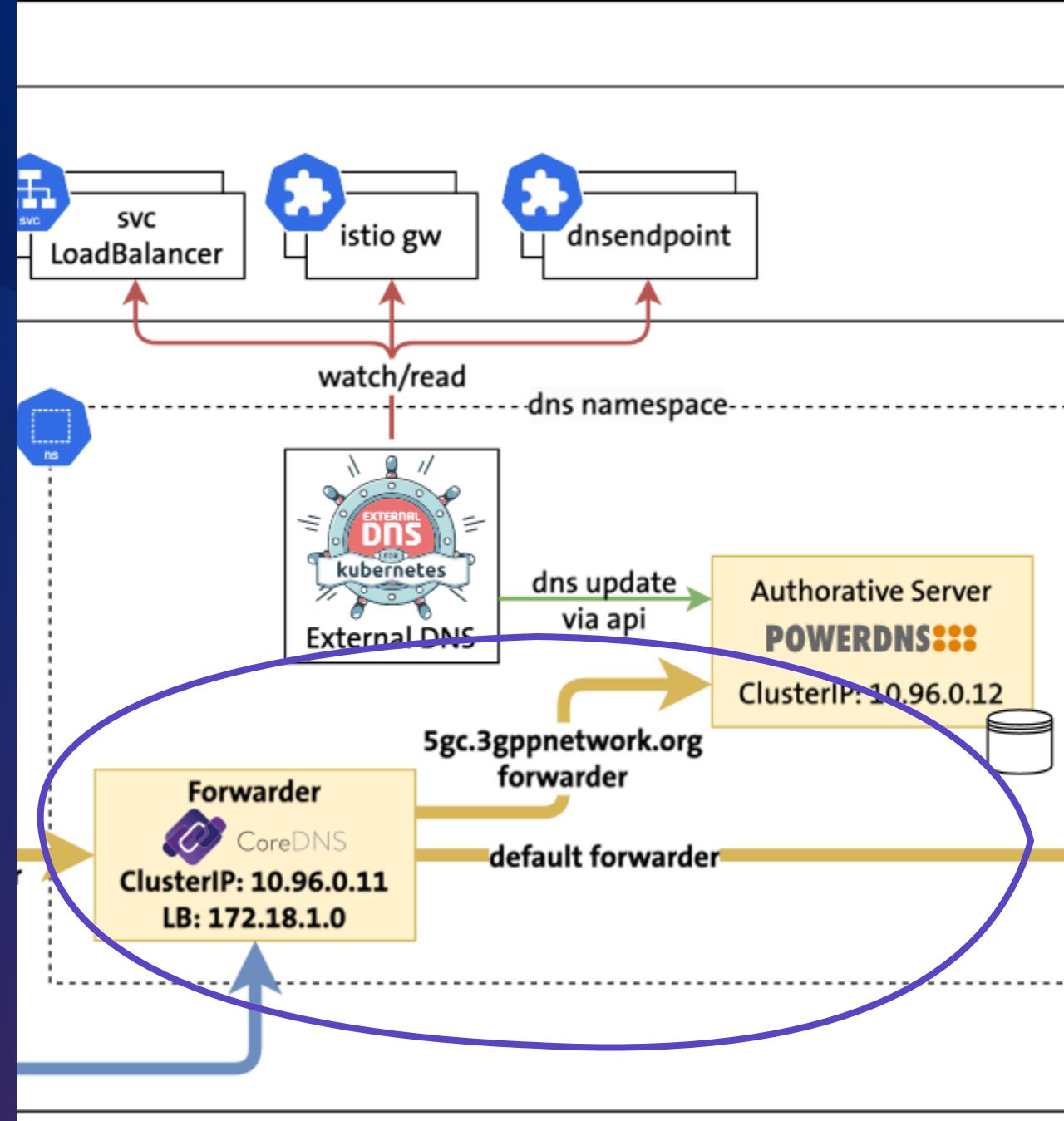
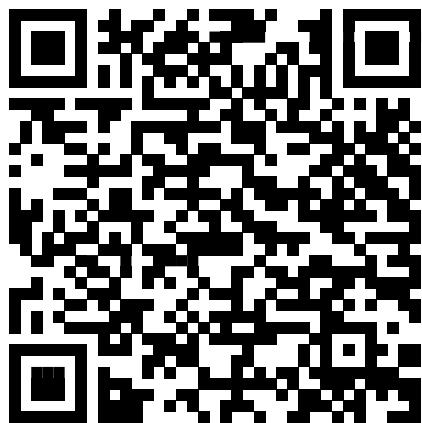




# Forwarding to Authoritative Server

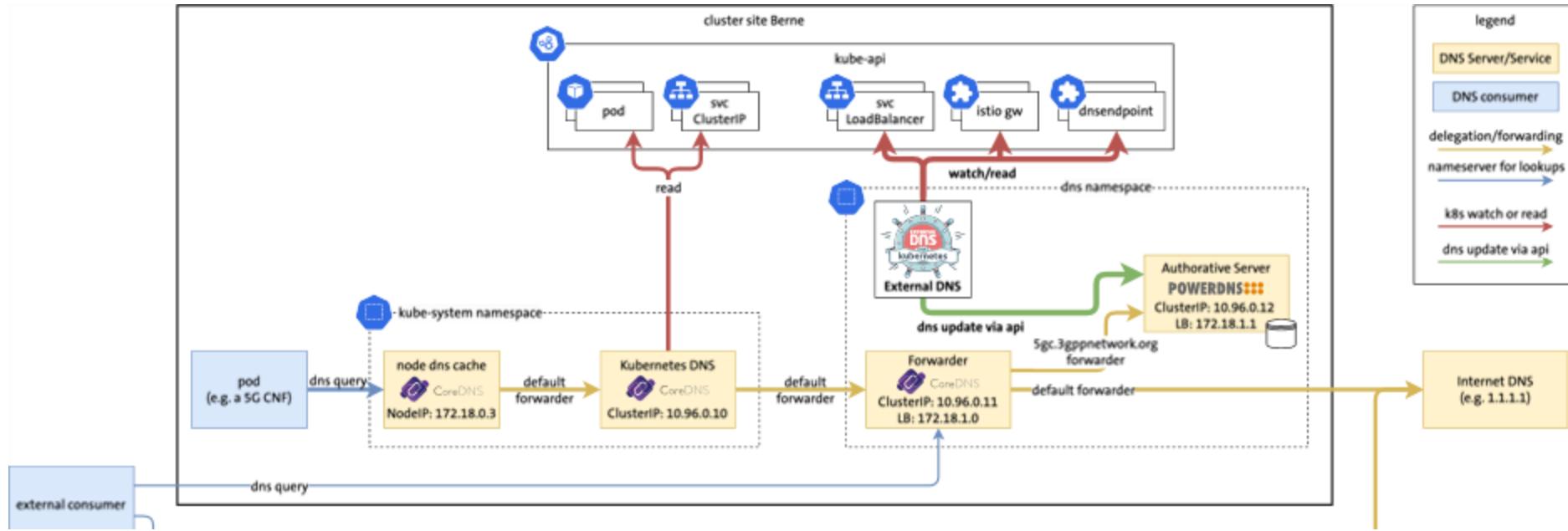


# Demo Forwarding



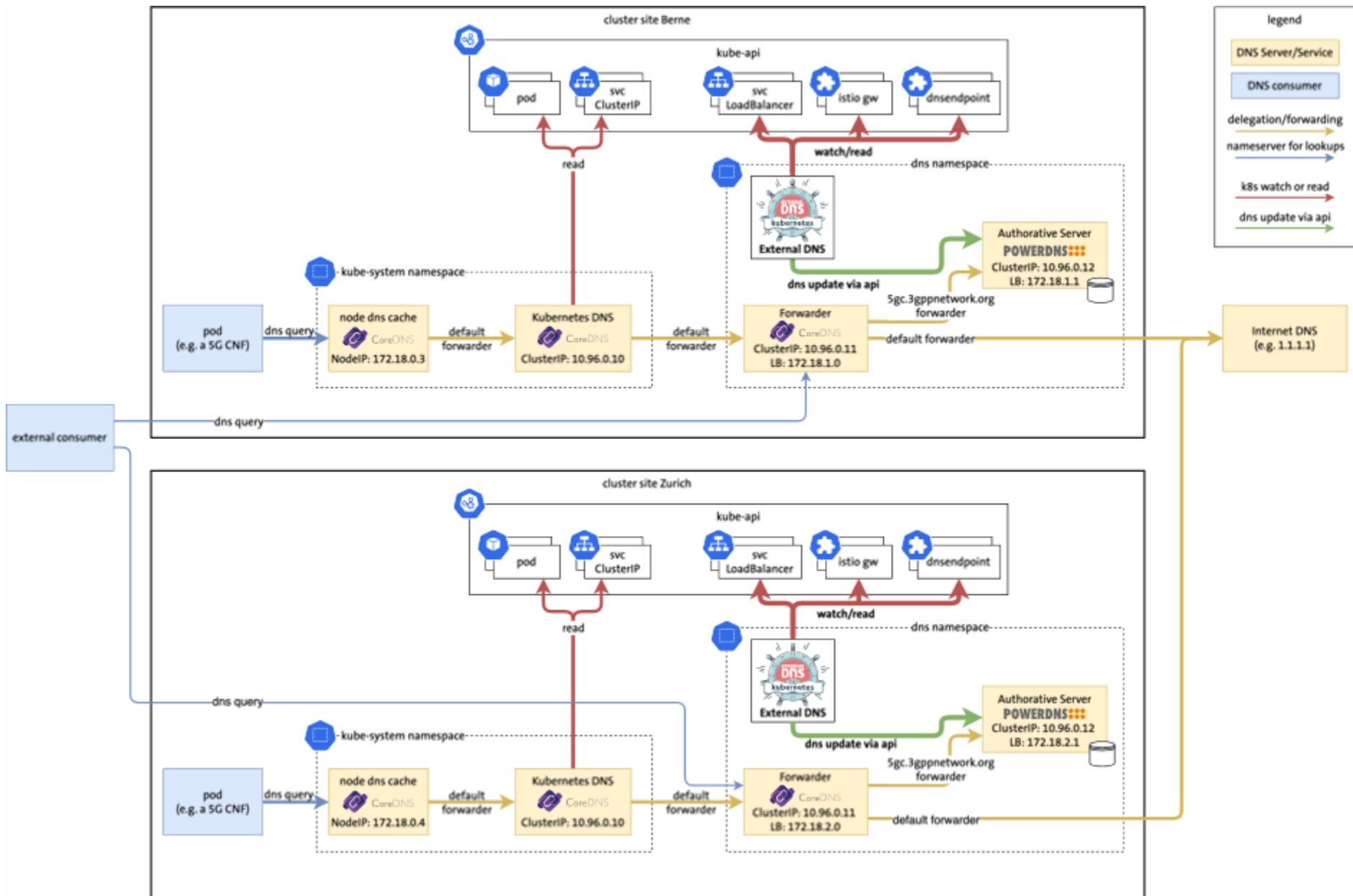


# Dual-Cluster Using ExternalDNS



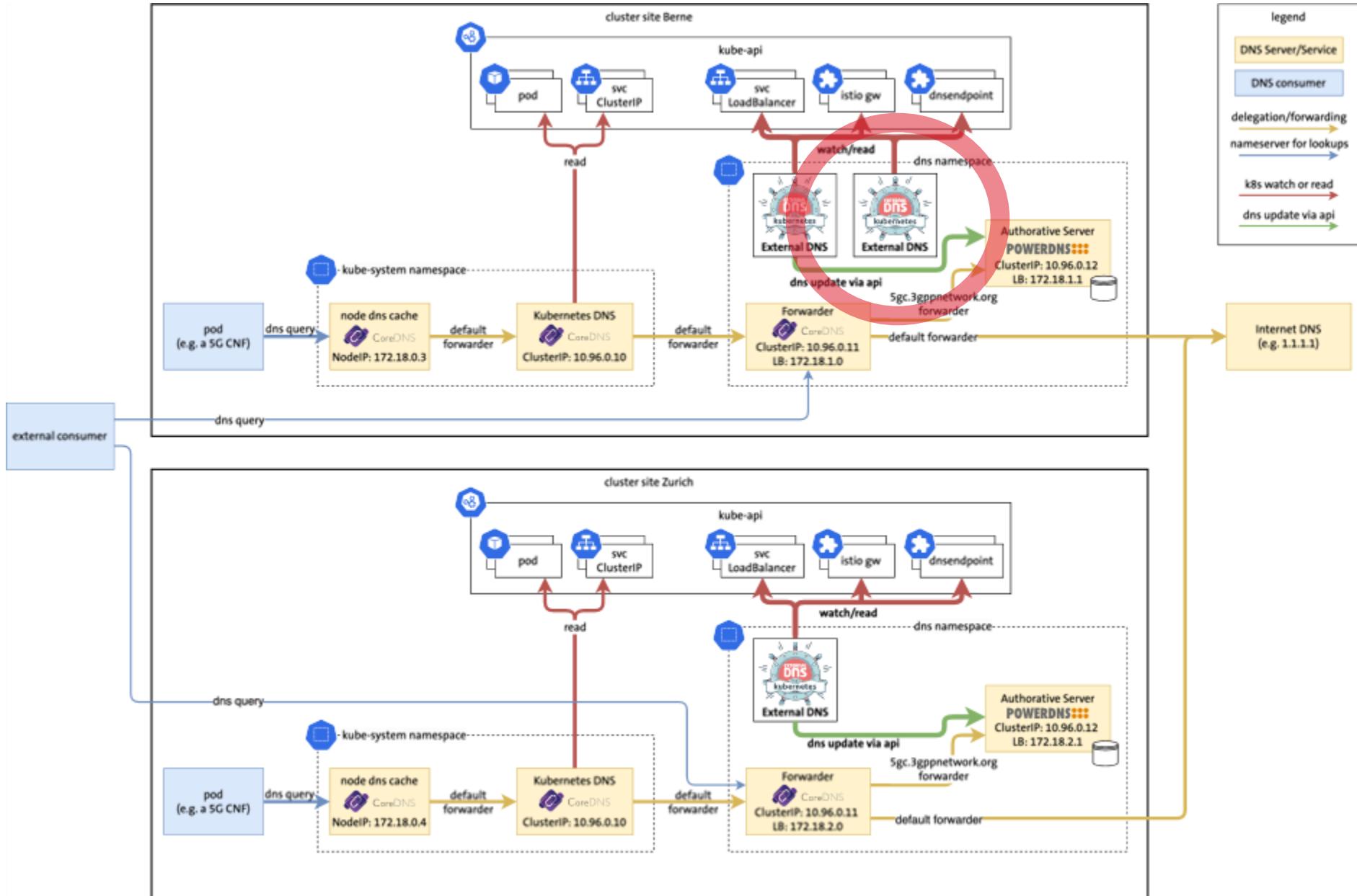


# Dual-Cluster Using ExternalDNS



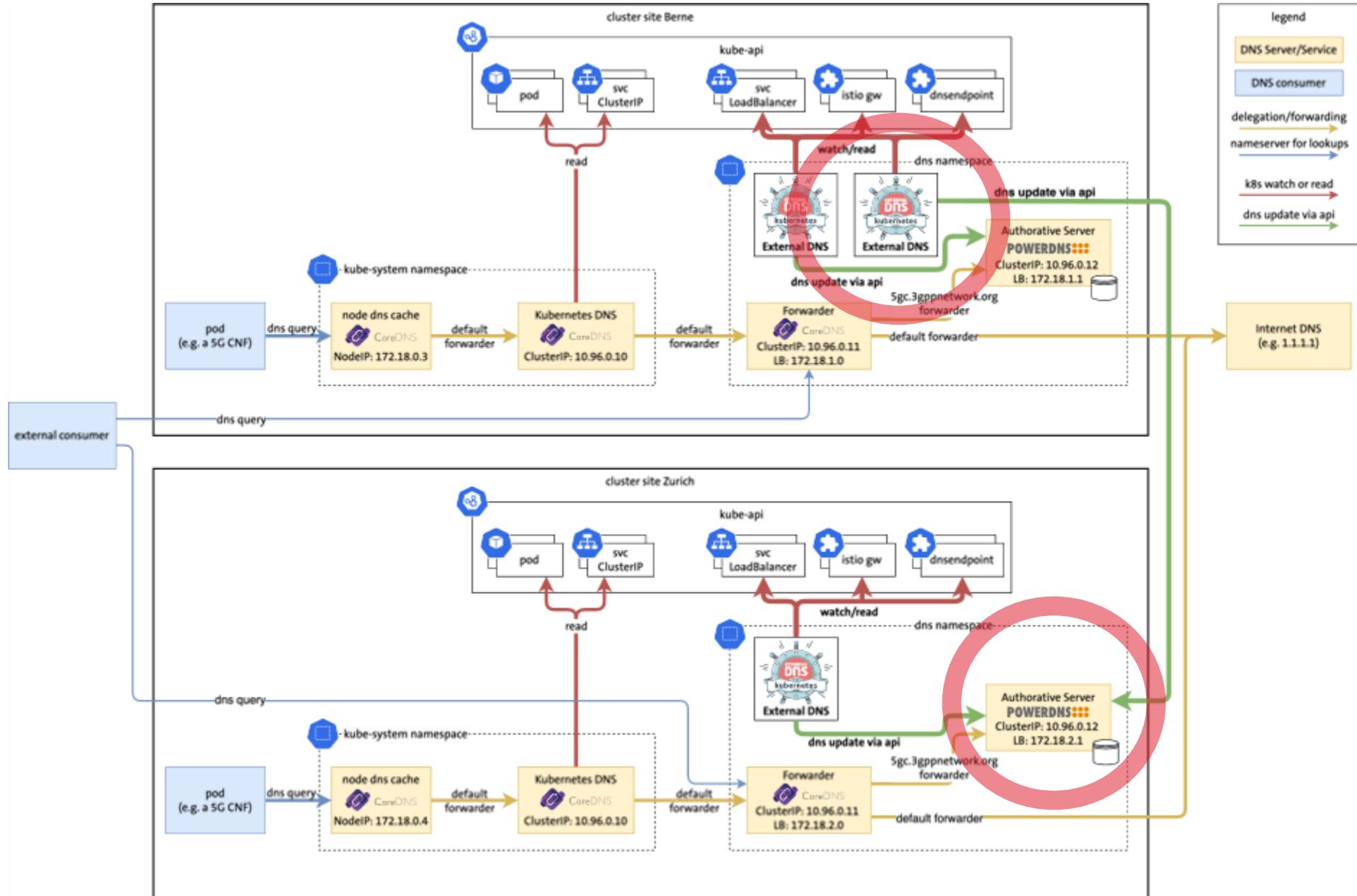


# Dual-Cluster Using ExternalDNS



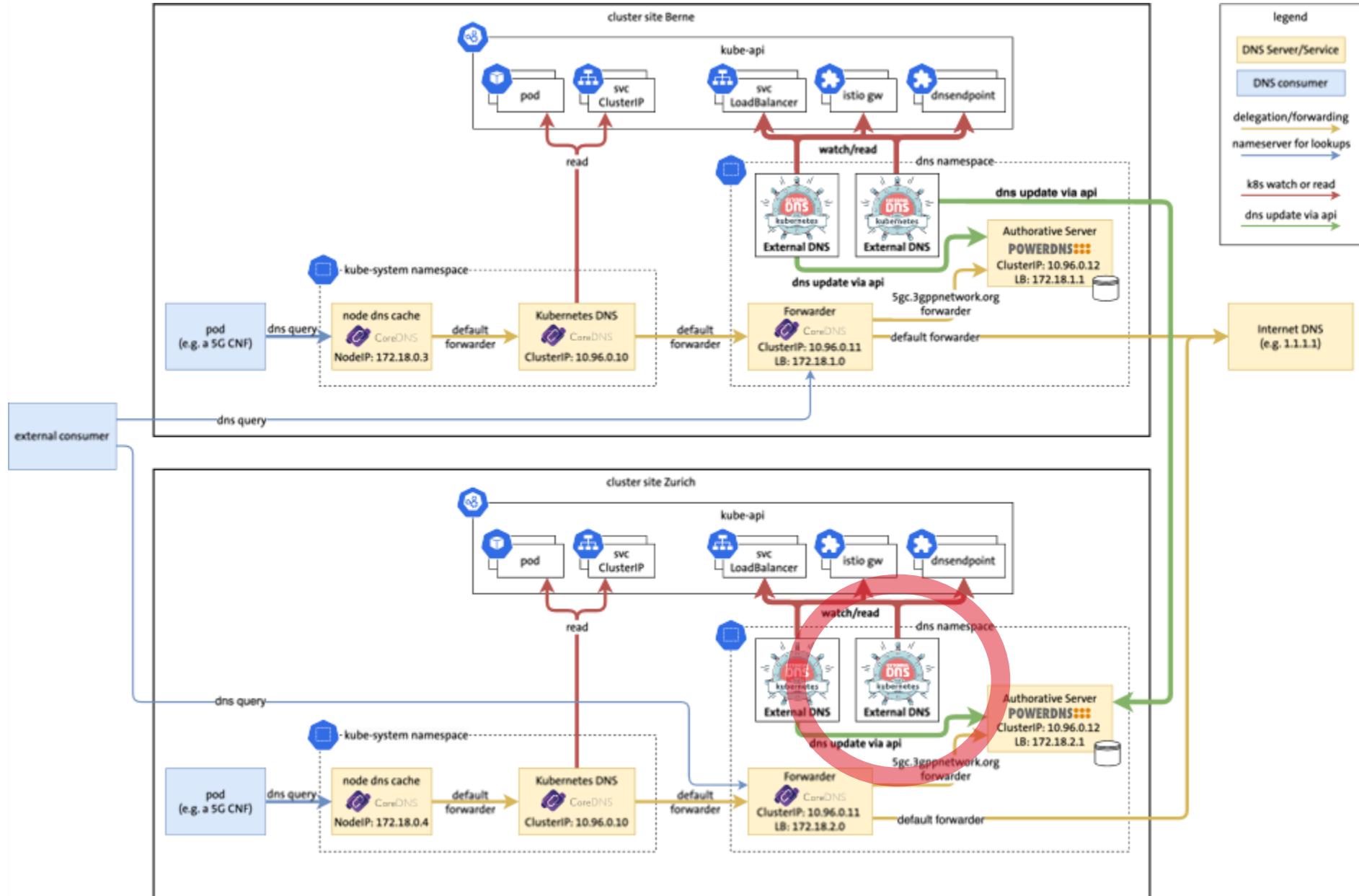


# Dual-Cluster Using ExternalDNS



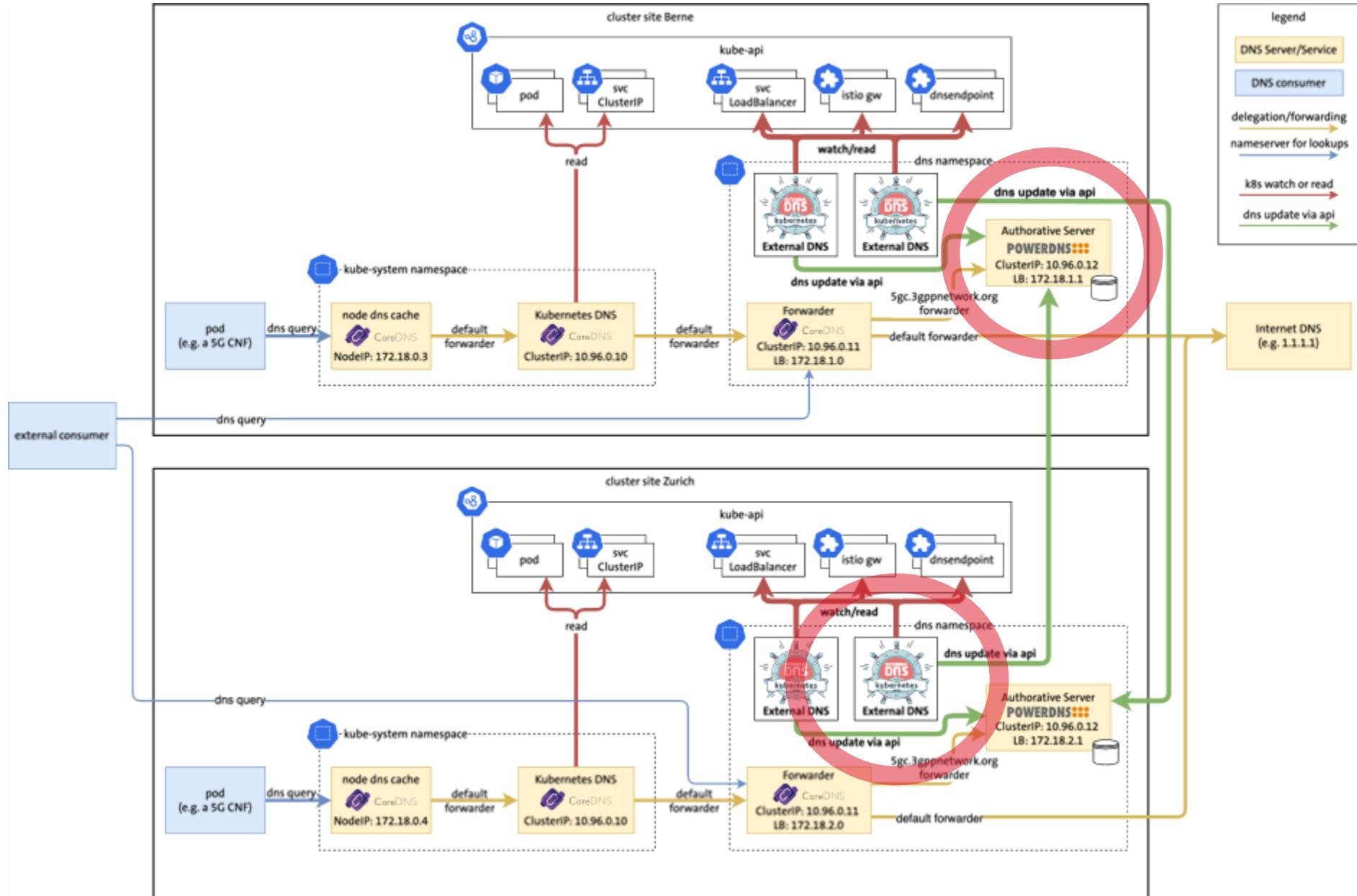


# Dual-Cluster Using ExternalDNS



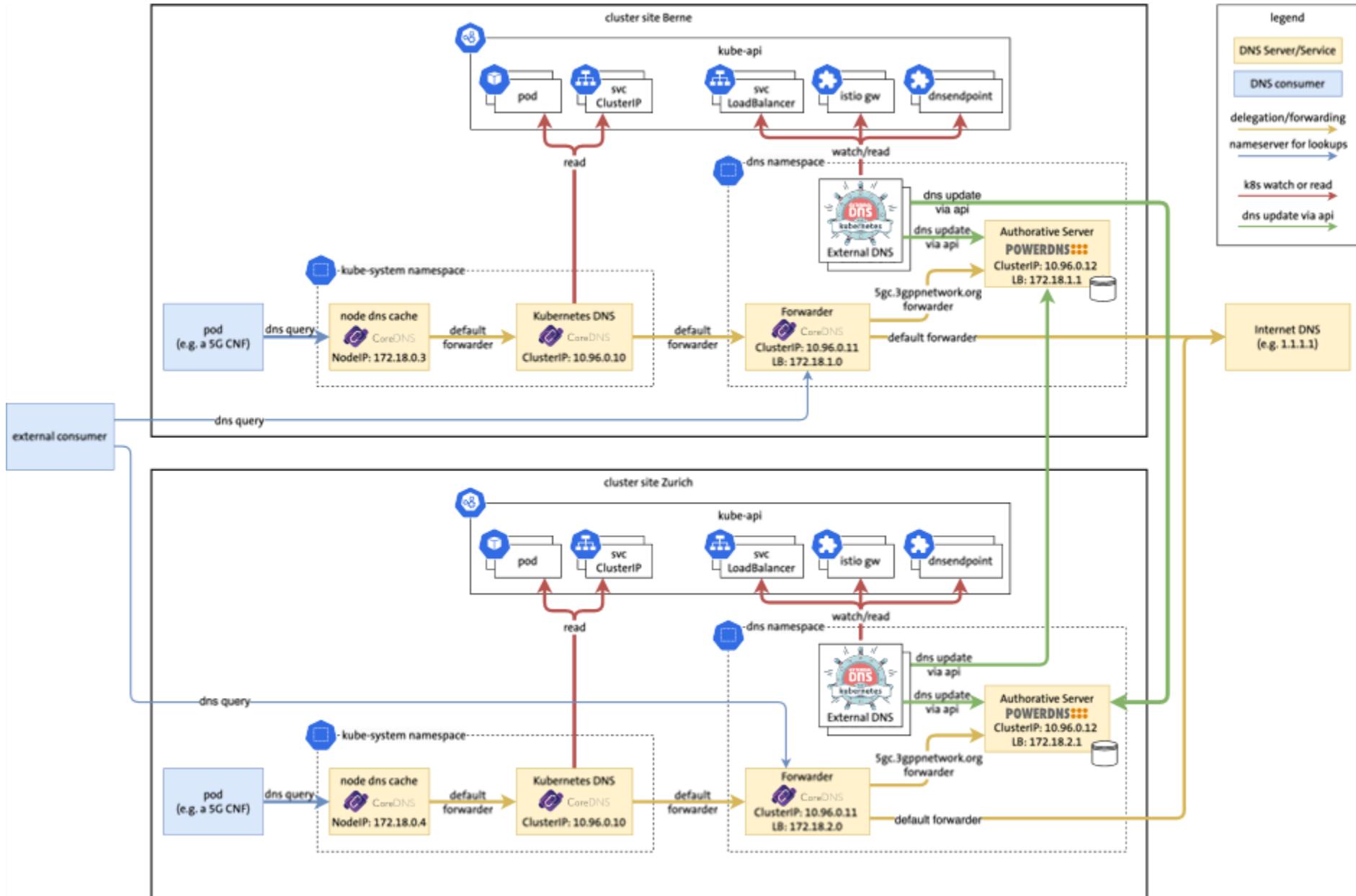


# Dual-Cluster Using ExternalDNS



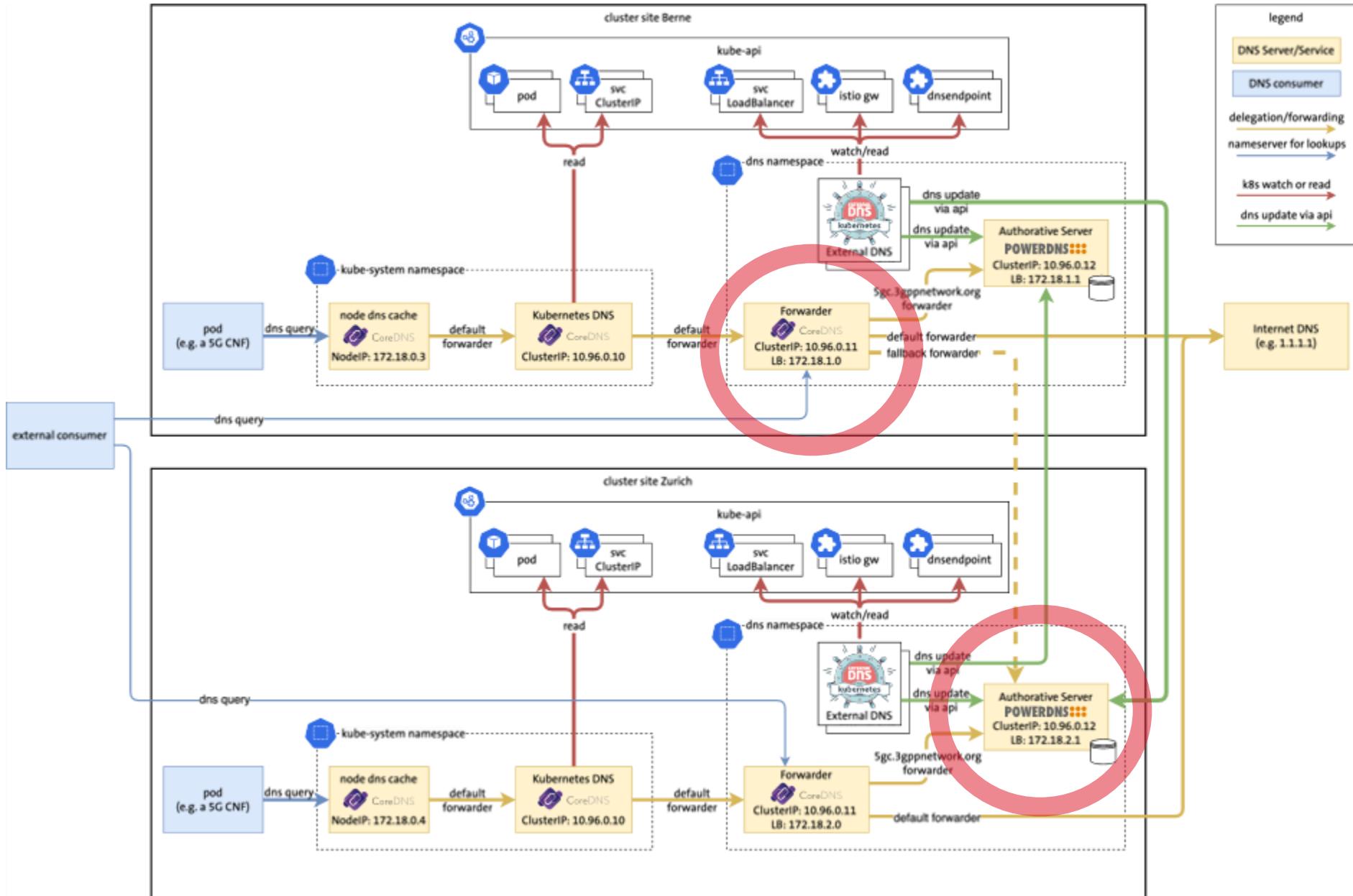


# Dual-Cluster Using ExternalDNS



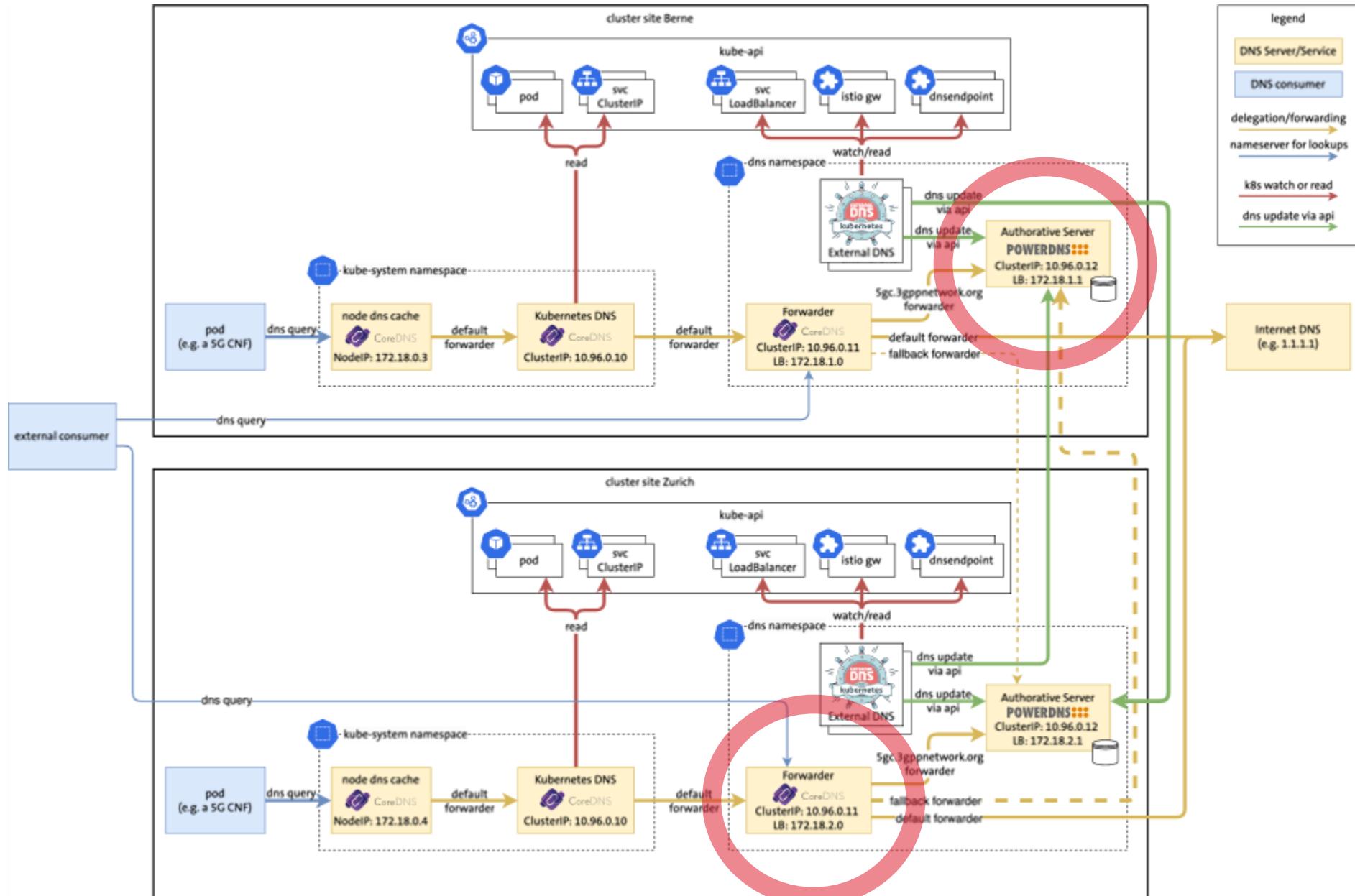


# Eliminating Single Point of Failure



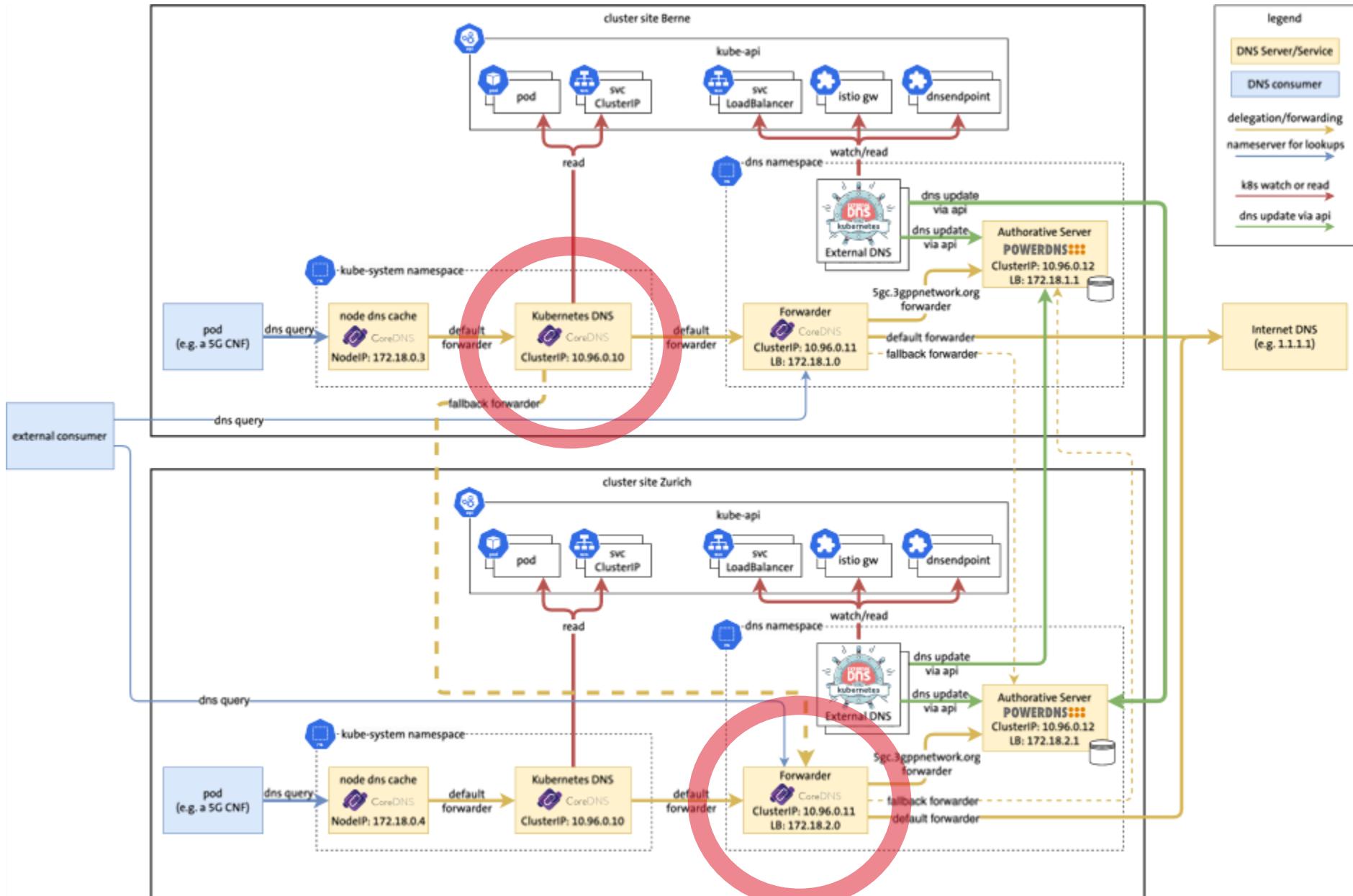


# Eliminating Single Point of Failure



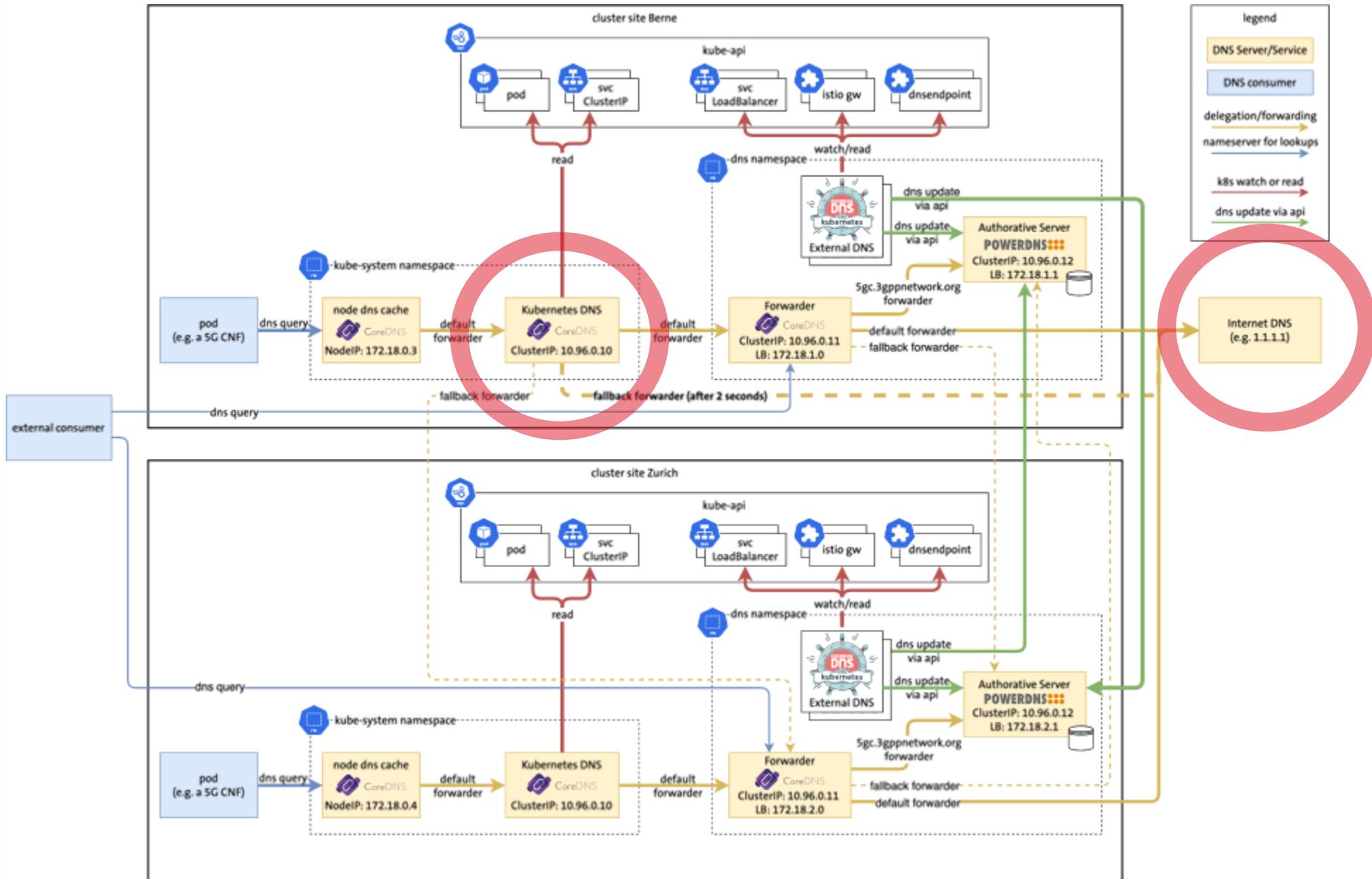


# Eliminating Single Point of Failure



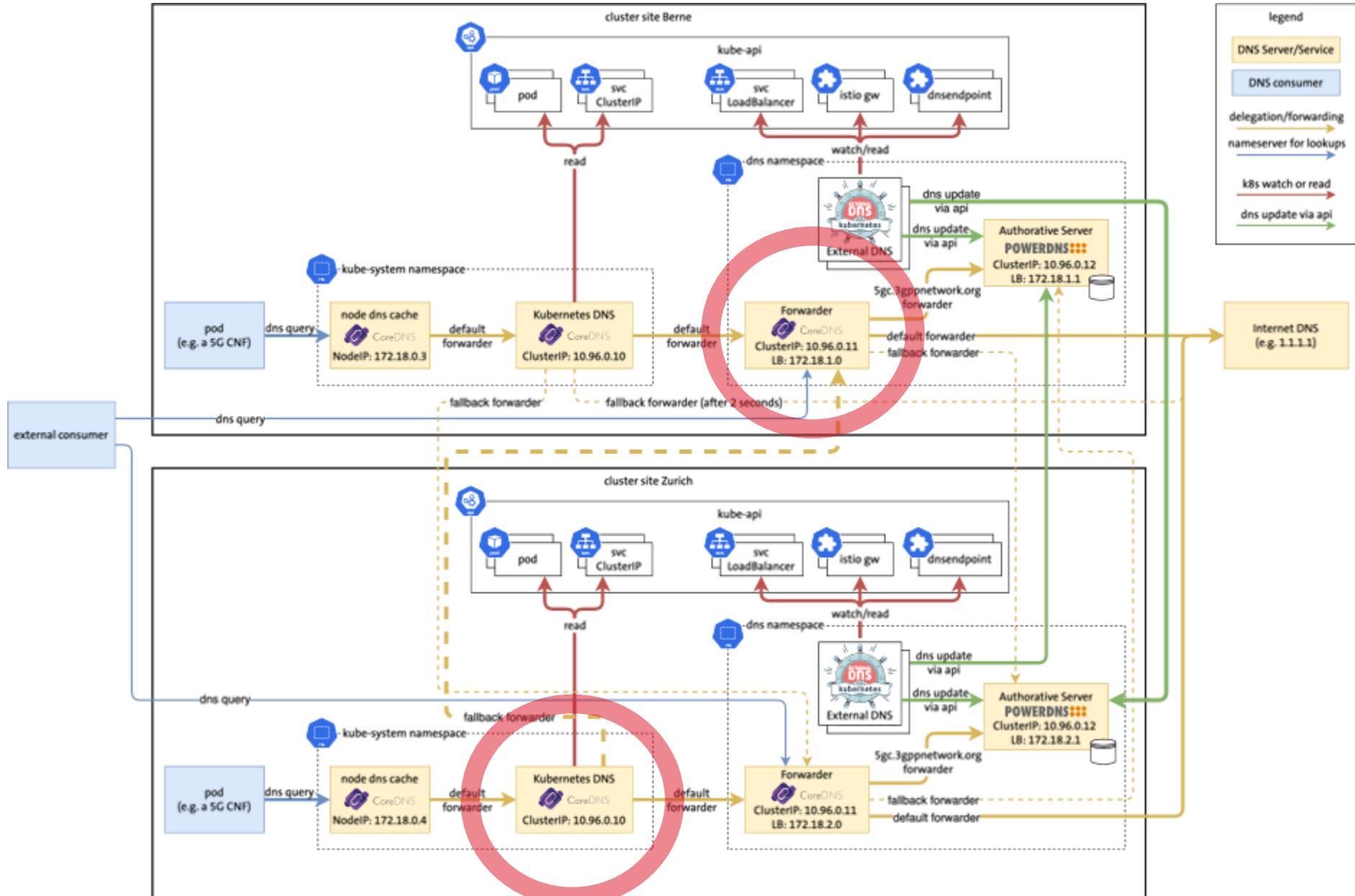


# Eliminating Single Point of Failure



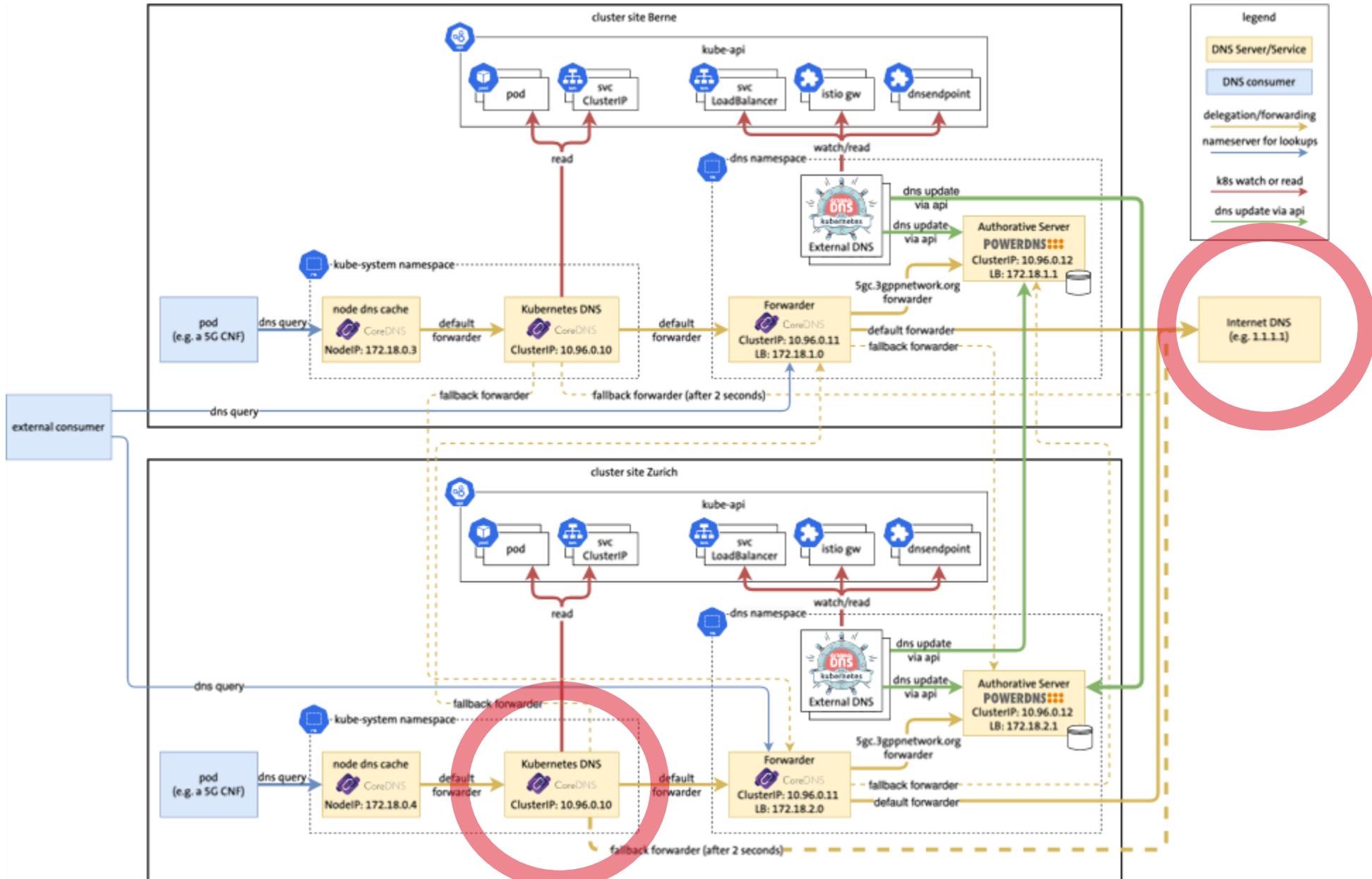


# Eliminating Single Point of Failure



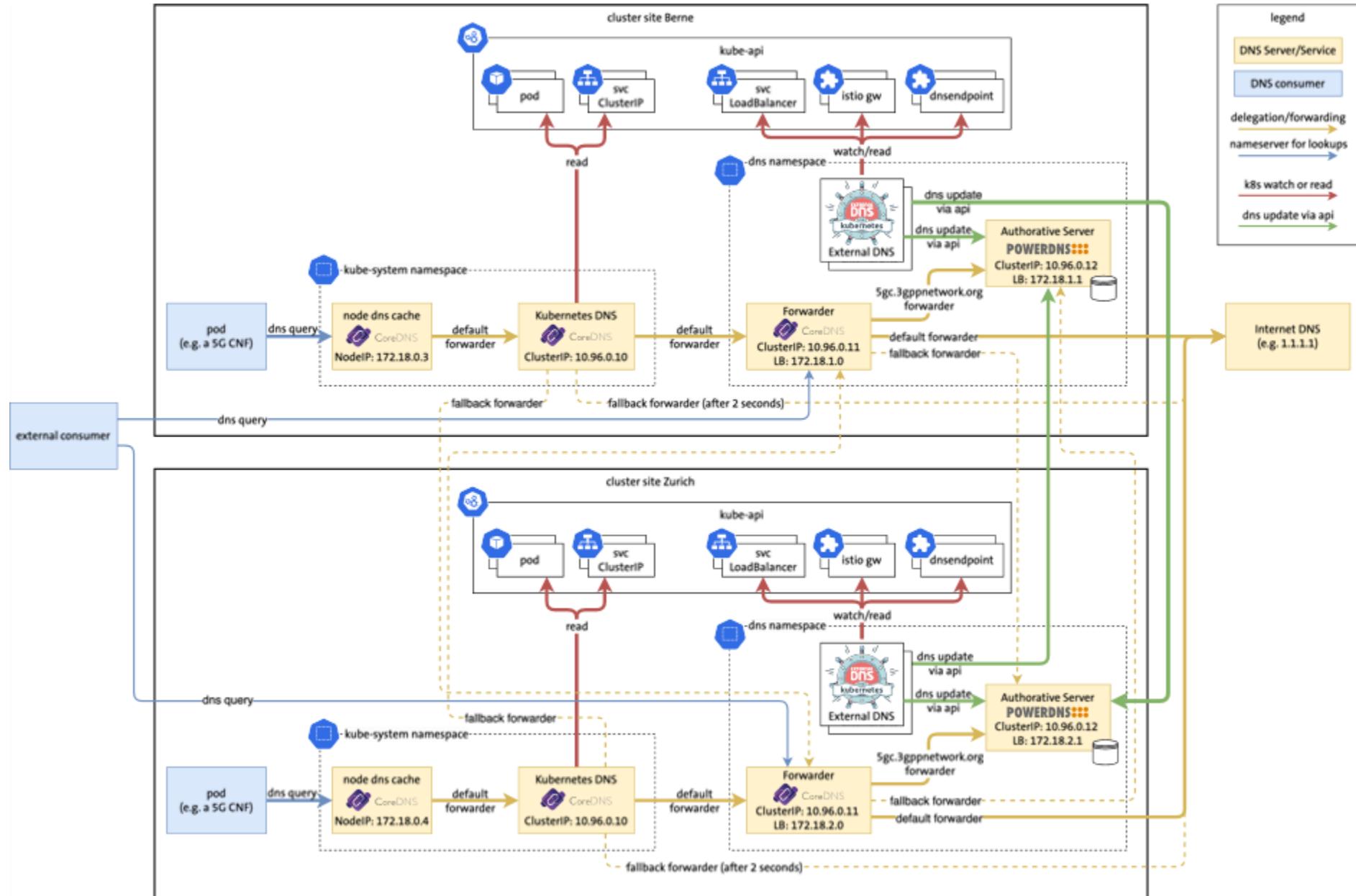


# Eliminating Single Point of Failure



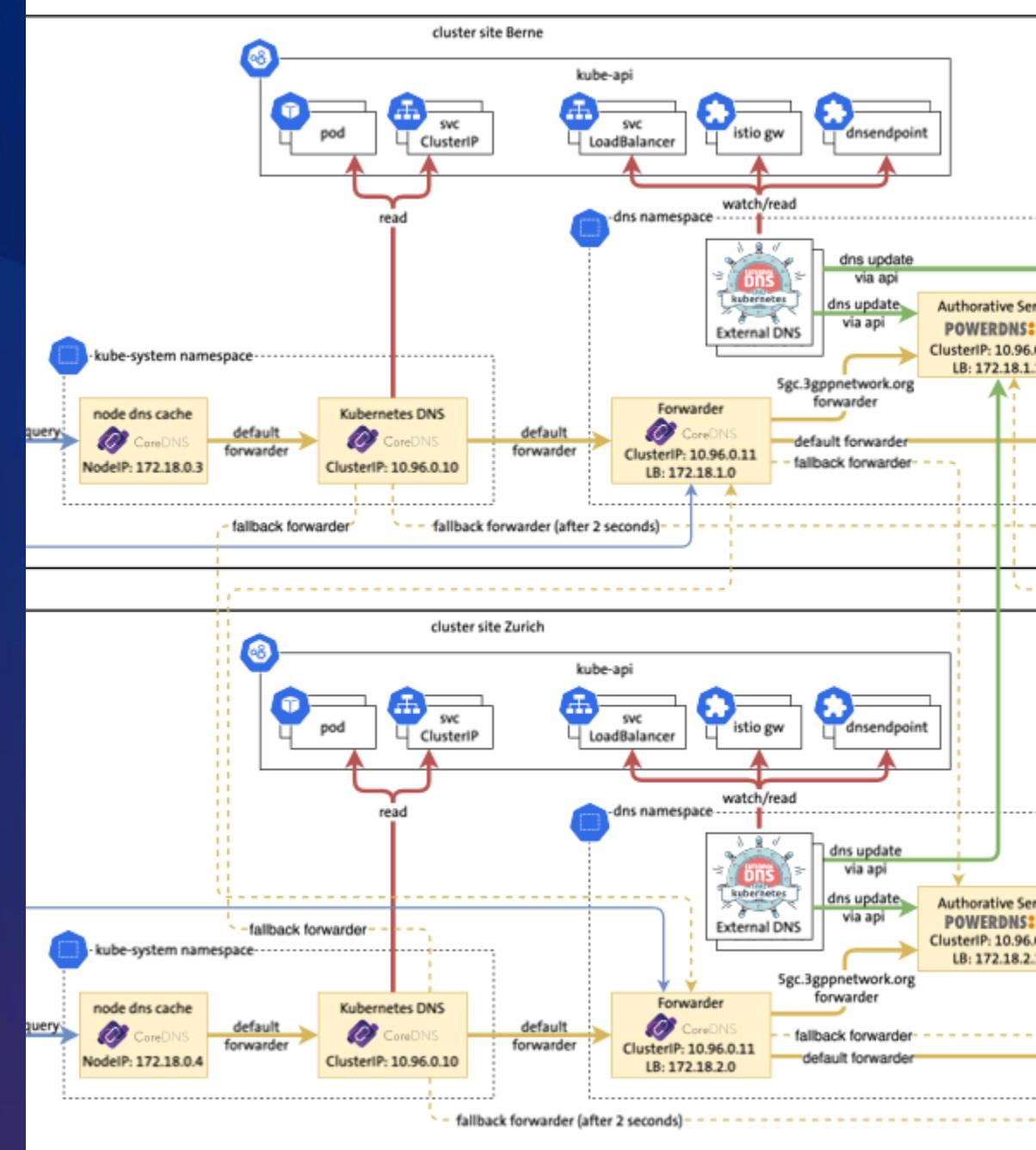


# Final Dual Cluster Setup





# Demo Multi Cluster





## Limitations of Our DNS Service



### **Self-dependence**

Complexity increases when consuming the Service from within the same clusters.



### **Kubernetes Resources only**

Limited to Kubernetes Resources and GitOps



### **Service Discovery**

ExternalDNS not suited for service discovery



## Limitations of ExternalDNS: Service Discovery

Interval-Based Syncing due to architectural decisions

- ⚠️ Delayed Resource Record creation

No Health Checks (e.g. integration into [Kubernetes Services/EndpointSlices](#))

- ⚠️ Cannot rely on ExternalDNS for app readiness

No Multi Cluster Round Robin for A records: one record cannot be shared by multiple ExternalDNS

- ⚠️ Cannot use DNS records created by ExternalDNS for routing across multiple clusters

Full cluster outage will not revoke DNS records

- ⚠️ Tight monitoring and additional automation needed to avoid outages



# What Did We Achieve?



## Proximity to Consumer

Minimal amount of hops between  
5G Core and DNS

✓ On-prem deployment



## Fully Automated

GitOps driven and automated  
provisioning of DNS records

✓ GitOps + ExternalDNS



## Geo Redundant & HA

Spread across multiple K8s clusters and  
geo regions to increase reliability

✓ Spread across multiple K8s Clusters



## Support of Advanced DNS features

Resource Records such as NAPTR and  
SRV supported for e.g. SIP Phone Calls

✓ Advanced RRs supported



## K8s integration with ExternalDNS

The System leverages Kubernetes  
Patterns such as CRs and Operators

✓ 100% Kubernetes Resources



## Minimal Amount of SPOFs

Remove single points of failure from the  
System

✓ Distributed control plane

✓ Cross-Cluster Forwarding



# Thanks!





# Q&A

