Dig Smart: Creating A Reliable Cloud-Native DNS Service

Joel Studler & Fabian Schulz

Joel Studler
Senior DevOps Engineer

joel.studler@swisscom.com

Fabian Schulz
DevOps Engineer

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

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

✗ No SaaS allowed

Fully Automated
GitOps driven and automated provisioning of DNS records

✗ No manual interaction allowed

Geo Redundant & HA
Spread across multiple K8s clusters and geo regions to increase reliability

✗ No singletons

Support of Advanced DNS features
Resource Records such as NAPTR and SRV supported for e.g. SIP Phone Calls

✗ Need to go beyond A and CNAME

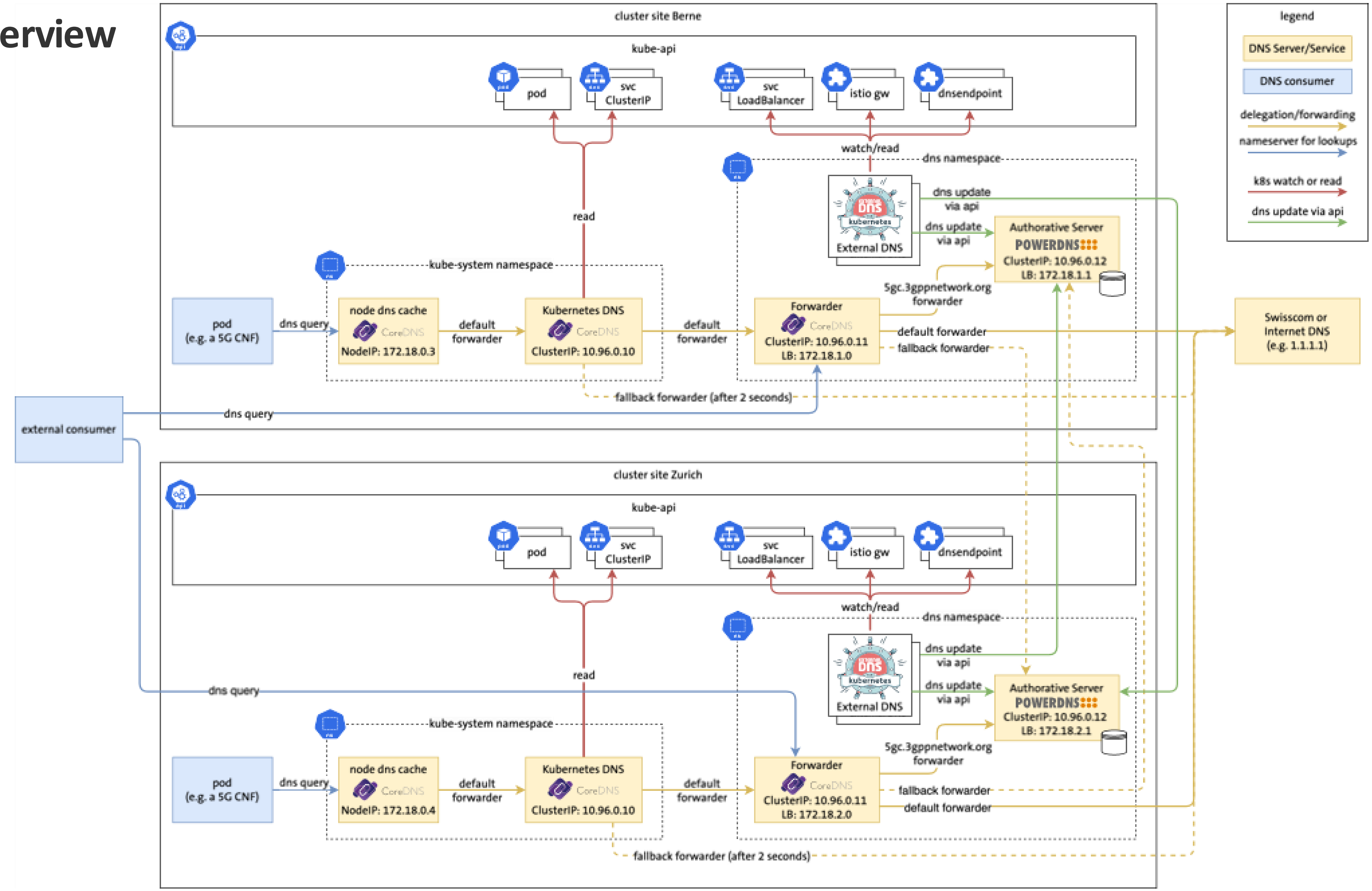
K8s integration with ExternalDNS
The System leverages Kubernetes Patterns such as CRs and Operators

✗ No CRUD outside kube-api

Minimal Amount of SPOFs
Share nothing by removing single points of failure from the System

✗ No shared mgmt system

Overview



cluster site Berne

kube-api

pod

svc ClusterIP

svc LoadBalancer

istio gw

dnsendpoint

read

watch/read

dns namespace

External DNS

dns update via api

dns update via api

Authorative Server POWERDNS ClusterIP: 10.96.0.12 LB: 172.18.1.1

5gc.3gppnetwork.org forwarder

kube-system namespace

pod (e.g. a 5G CNF)

dns query

node dns cache CoreDNS NodeIP: 172.18.0.3

default forwarder

Kubernetes DNS CoreDNS ClusterIP: 10.96.0.10

default forwarder

Forwarder CoreDNS ClusterIP: 10.96.0.11 LB: 172.18.1.0

default forwarder

fallback forwarder

fallback forwarder (after 2 seconds)

external consumer

dns query

Swisscom or Internet DNS (e.g. 1.1.1.1)

cluster site Zurich

kube-api

pod

svc ClusterIP

svc LoadBalancer

istio gw

dnsendpoint

read

watch/read

dns namespace

External DNS

dns update via api

dns update via api

Authorative Server POWERDNS ClusterIP: 10.96.0.12 LB: 172.18.2.1

5gc.3gppnetwork.org forwarder

dns query

kube-system namespace

pod (e.g. a 5G CNF)

dns query

node dns cache CoreDNS NodeIP: 172.18.0.4

default forwarder

Kubernetes DNS CoreDNS ClusterIP: 10.96.0.10

default forwarder

Forwarder CoreDNS ClusterIP: 10.96.0.11 LB: 172.18.2.0

fallback forwarder

default forwarder

fallback forwarder (after 2 seconds)

legend

DNS Server/Service

DNS consumer

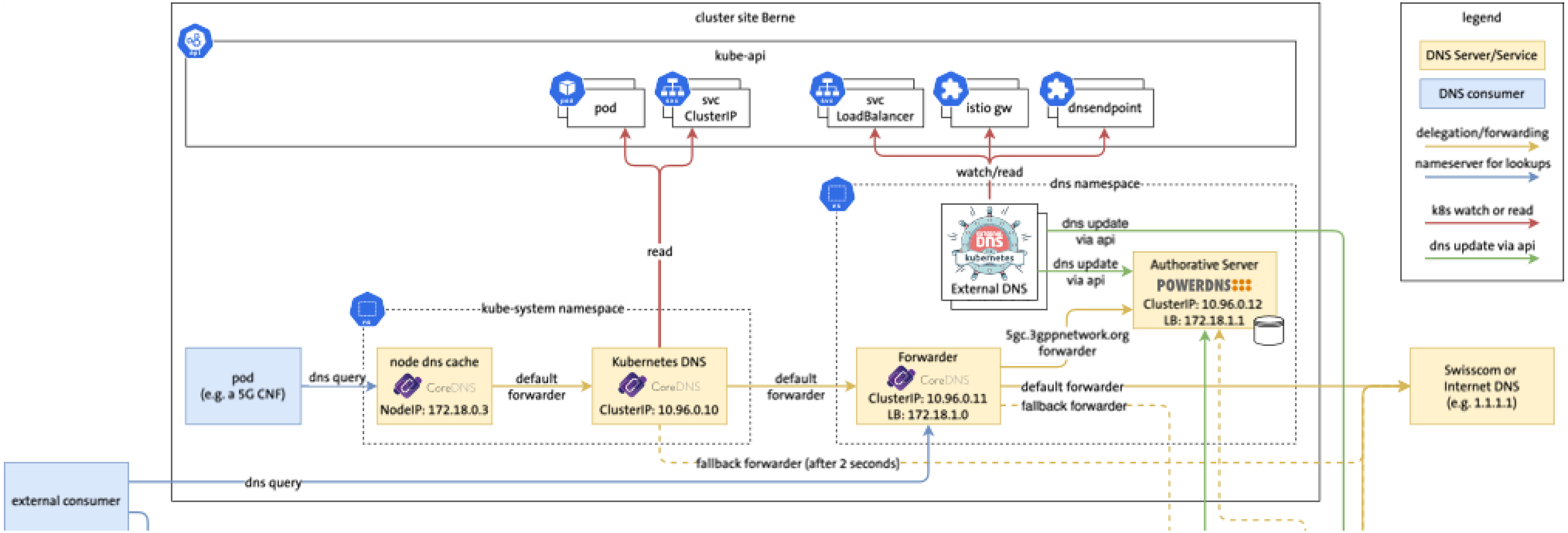
delegation/forwarding

nameserver for lookups

k8s watch or read

dns update via api

Overview


cluster site Berne
kube-api
pod
svc ClusterIP
svc LoadBalancer
istio gw
dnsendpoint
watch/read
read
dns namespace
External DNS
dns update via api
dns update via api
Authoritative Server
POWERDNS
ClusterIP: 10.96.0.12
LB: 172.18.1.1
5gc.3gppnetwork.org forwarder
kube-system namespace
pod (e.g. a 5G CNF)
dns query
node dns cache
CoreDNS
NodeIP: 172.18.0.3
default forwarder
Kubernetes DNS
CoreDNS
ClusterIP: 10.96.0.10
default forwarder
Forwarder
CoreDNS
ClusterIP: 10.96.0.11
LB: 172.18.1.0
default forwarder
fallback forwarder
Swisscom or Internet DNS (e.g. 1.1.1.1)
fallback forwarder (after 2 seconds)
external consumer
dns query
legend
DNS Server/Service
DNS consumer
delegation/forwarding
nameserver for lookups
k8s watch or read
dns update via api

In-Cluster Service Discovery in Kubernetes

CoreDNS (https://coredns.io)

kube-api as Backend

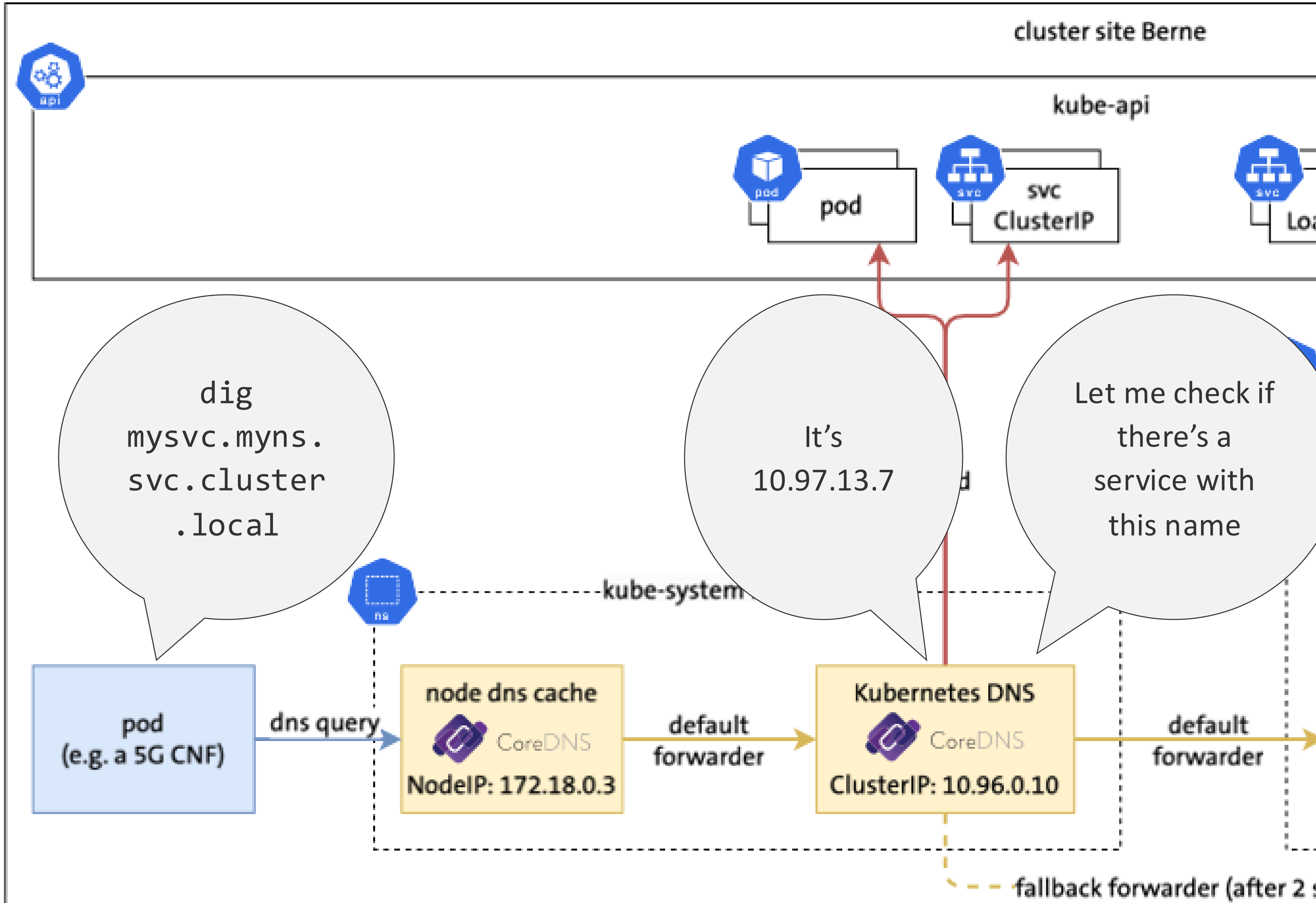
Features:

✓ In-Cluster Service Discovery

Missing:

✗ Not exposed outside of K8s

✗ 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

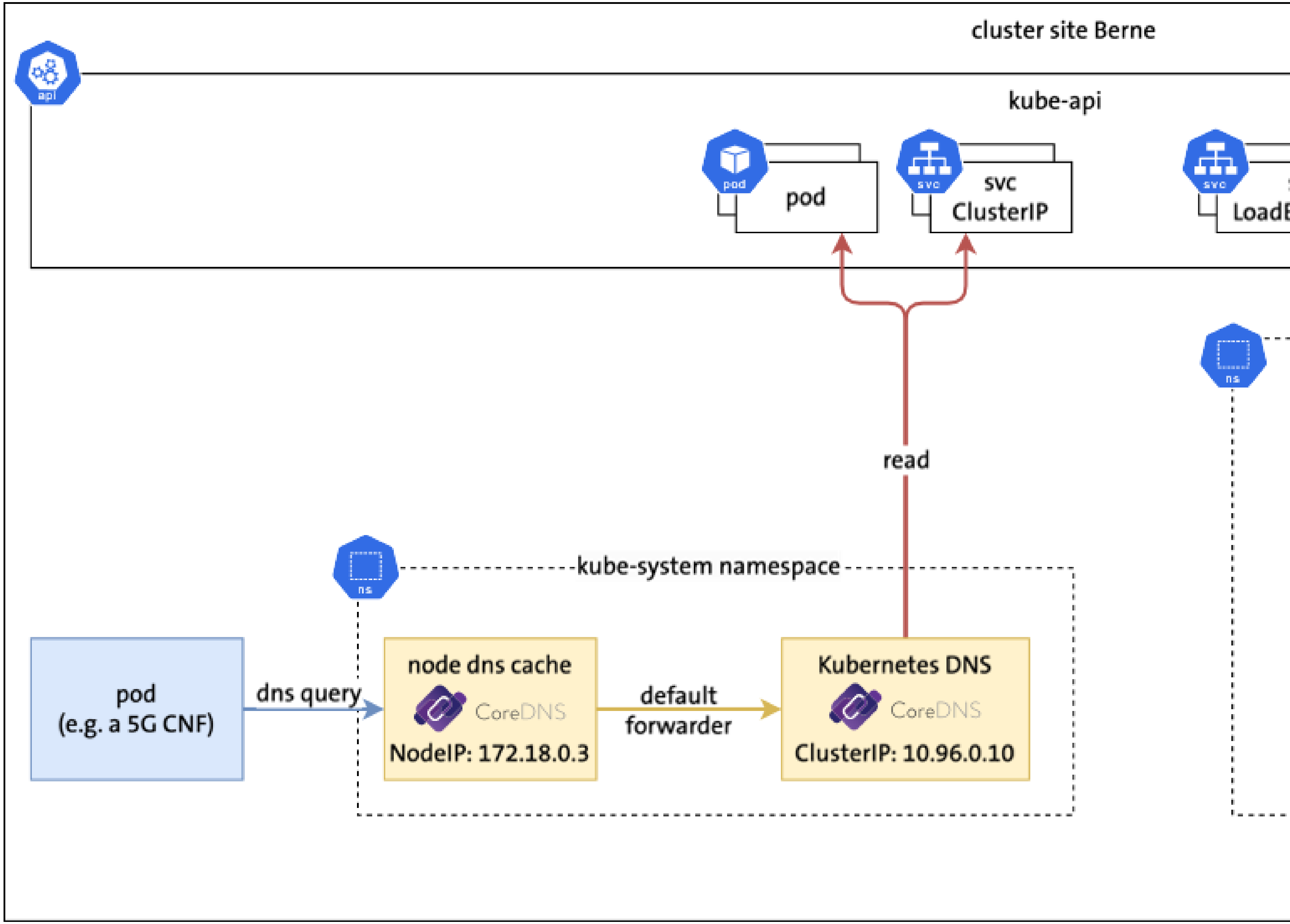
Customize DNS Service: https://kubernetes.io/docs/tasks/administer-cluster/dns-custom-nameservers

Requirements for Authoritative Server

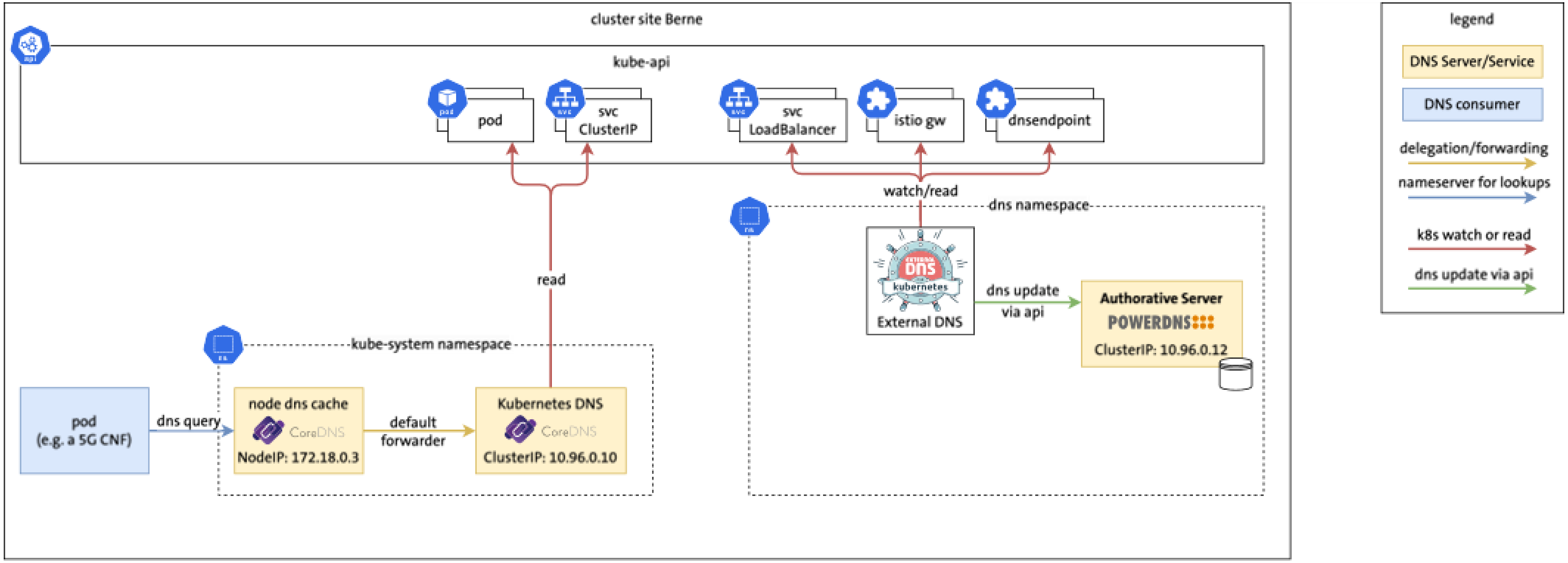
Requirement	CoreDNS 🥉	PowerDNS Authoritative 🥇	SaaS 🥈
ExternalDNS* Support for K8s integration	✓	✓	✓
A & CNAME Resource Records	✓	✓	✓
NAPTR Resource Records (e.g. for SIP phone calls)	✗	✓**	✓**
Proximity to Consumer	✓	✓	✗

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

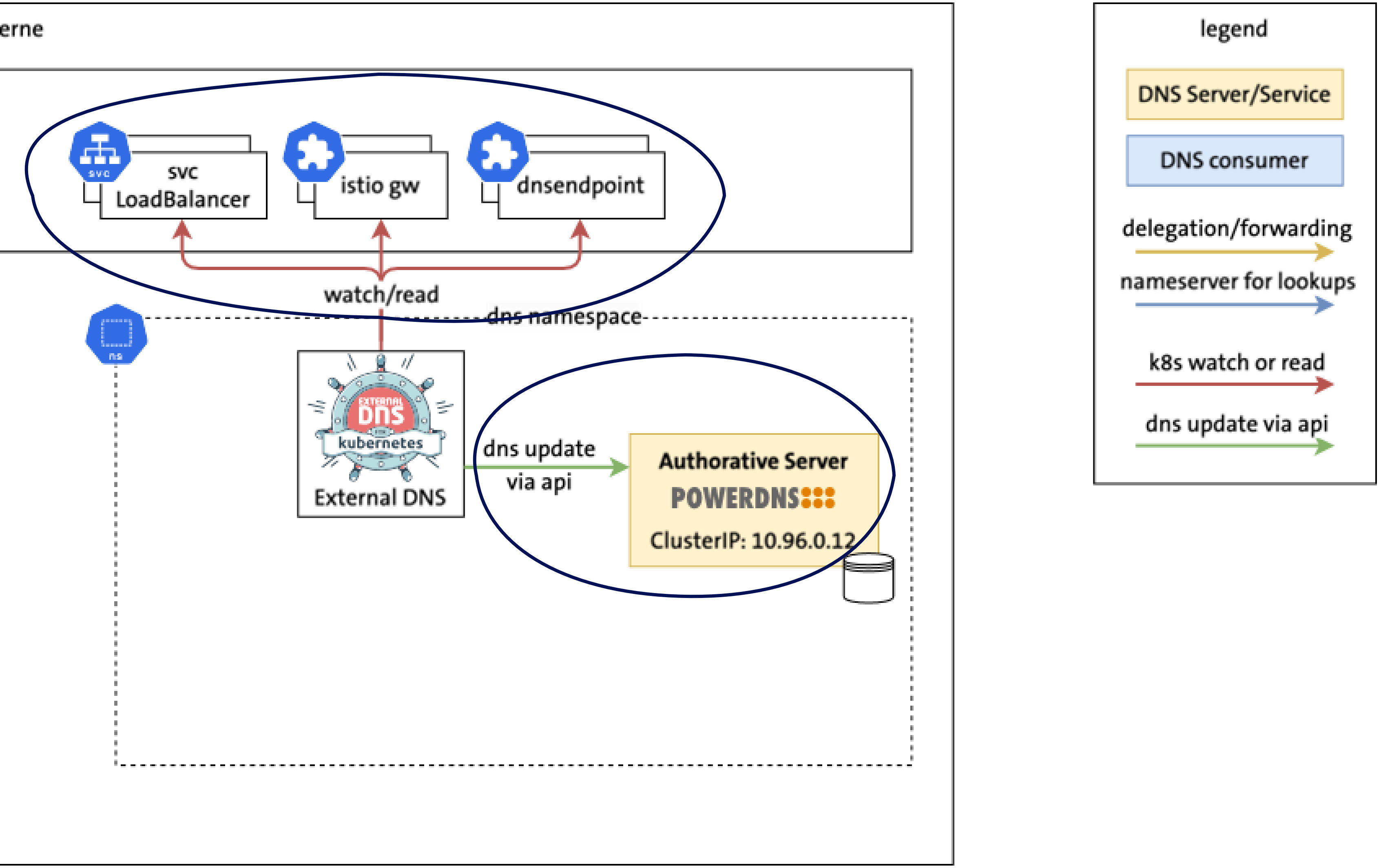
** After a fix in external-dns https://github.com/kubernetes-sigs/external-dns/pull/4212


cluster site Berne
api
kube-api
pod
pod
svc
svc
ClusterIP
svc
LoadB
read
ns
ns
kube-system namespace
pod
(e.g. a 5G CNF)
dns query
node dns cache
CoreDNS
NodeIP: 172.18.0.3
default
forwarder
Kubernetes DNS
CoreDNS
ClusterIP: 10.96.0.10

Overview

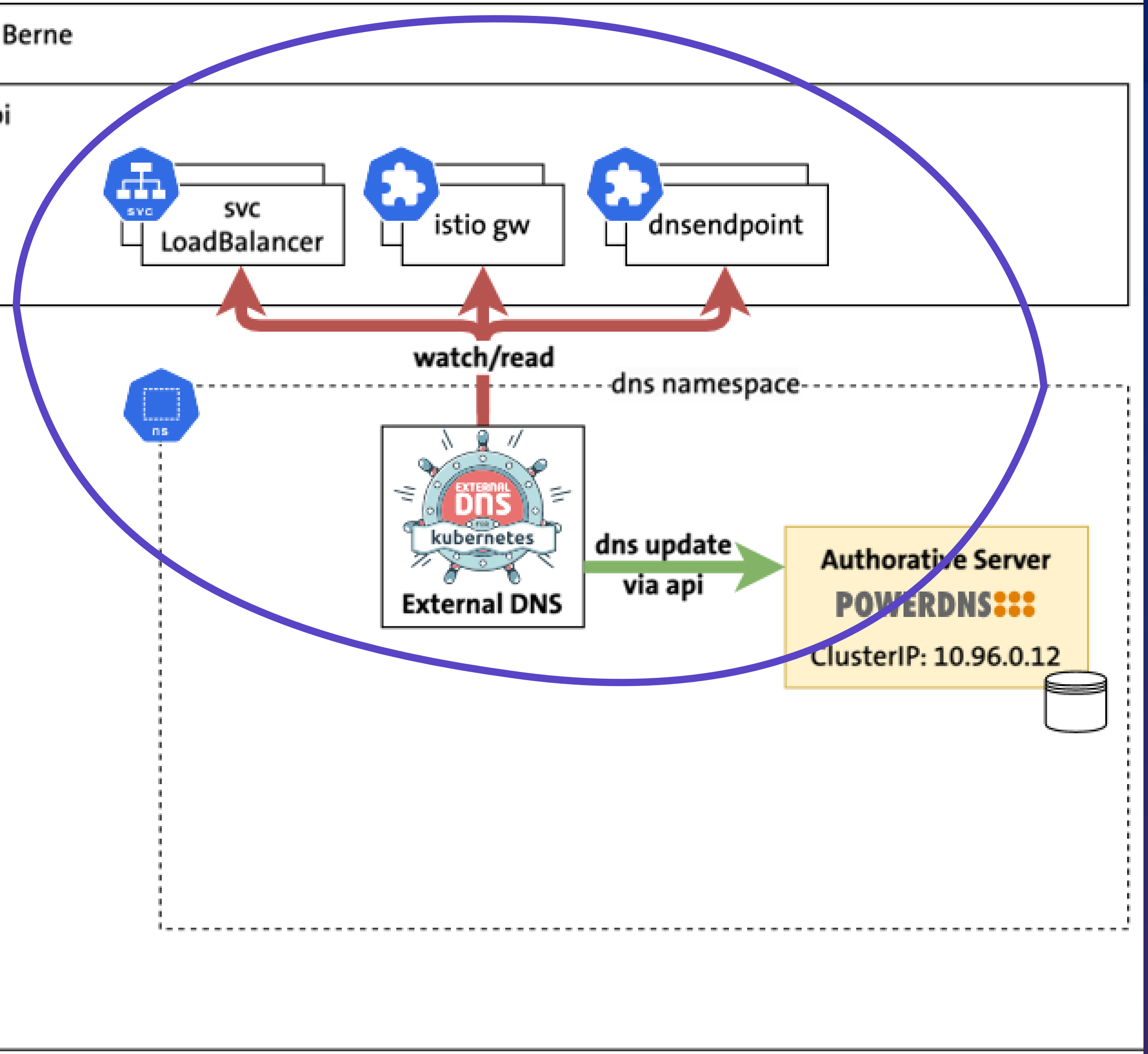

cluster site Berne
kube-api
pod
svc
ClusterIP
svc
LoadBalancer
istio gw
dnsendpoint
watch/read
read
dns namespace
External DNS
dns update
via api
Authorative Server
POWERDNS
ClusterIP: 10.96.0.12
kube-system namespace
pod
(e.g. a 5G CNF)
dns query
node dns cache
CoreDNS
NodeIP: 172.18.0.3
default
forwarder
Kubernetes DNS
CoreDNS
ClusterIP: 10.96.0.10
legend
DNS Server/Service
DNS consumer
delegation/forwarding
nameserver for lookups
k8s watch or read
dns update via api

Automation of Authoritative Server Using ExternalDNS



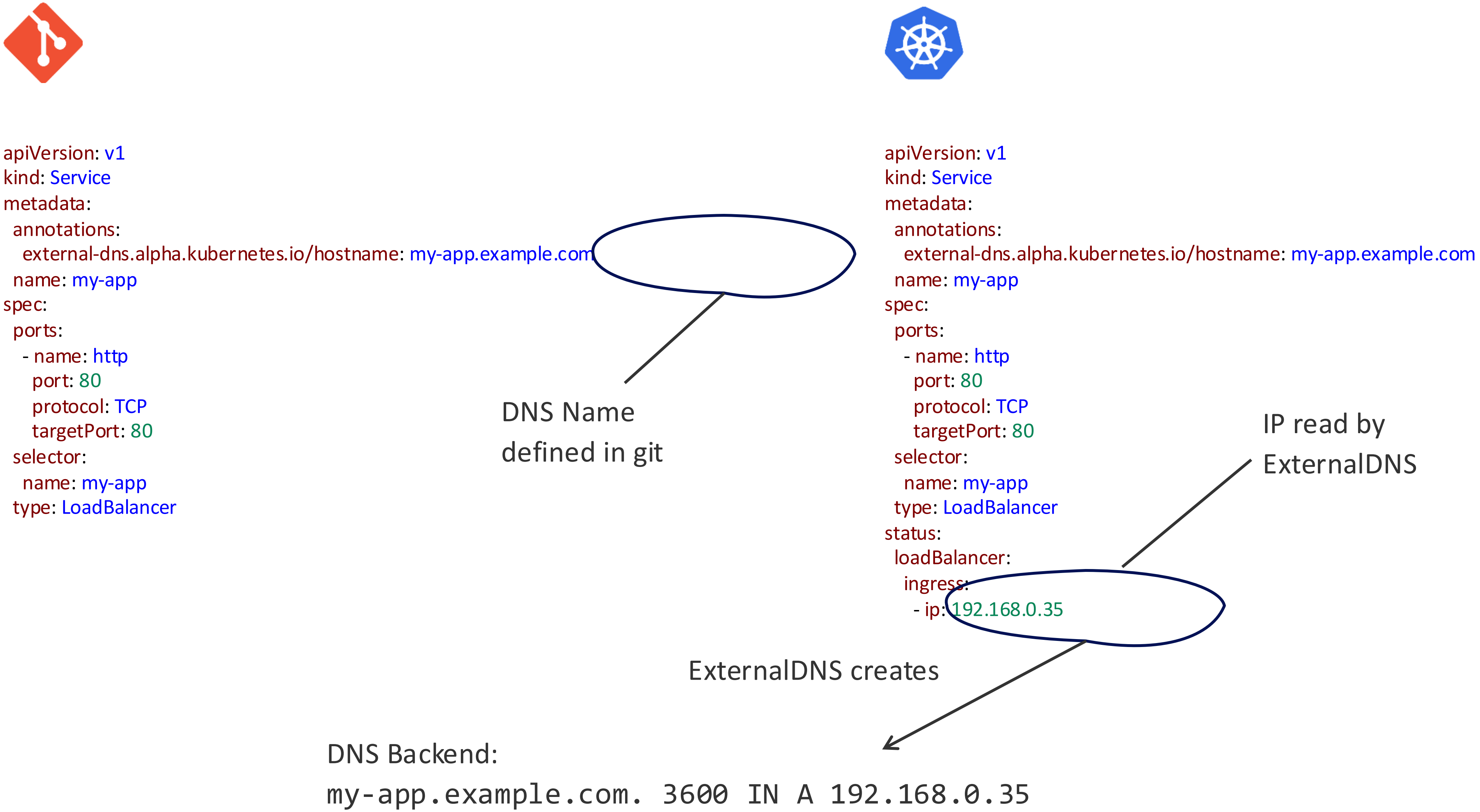
ExternalDNS syncs to backend using:

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



Demo ExternalDNS + PowerDNS Single Cluster

ExternalDNS State Management: GitOps + Kubernetes



```yaml
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

```yaml
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
```

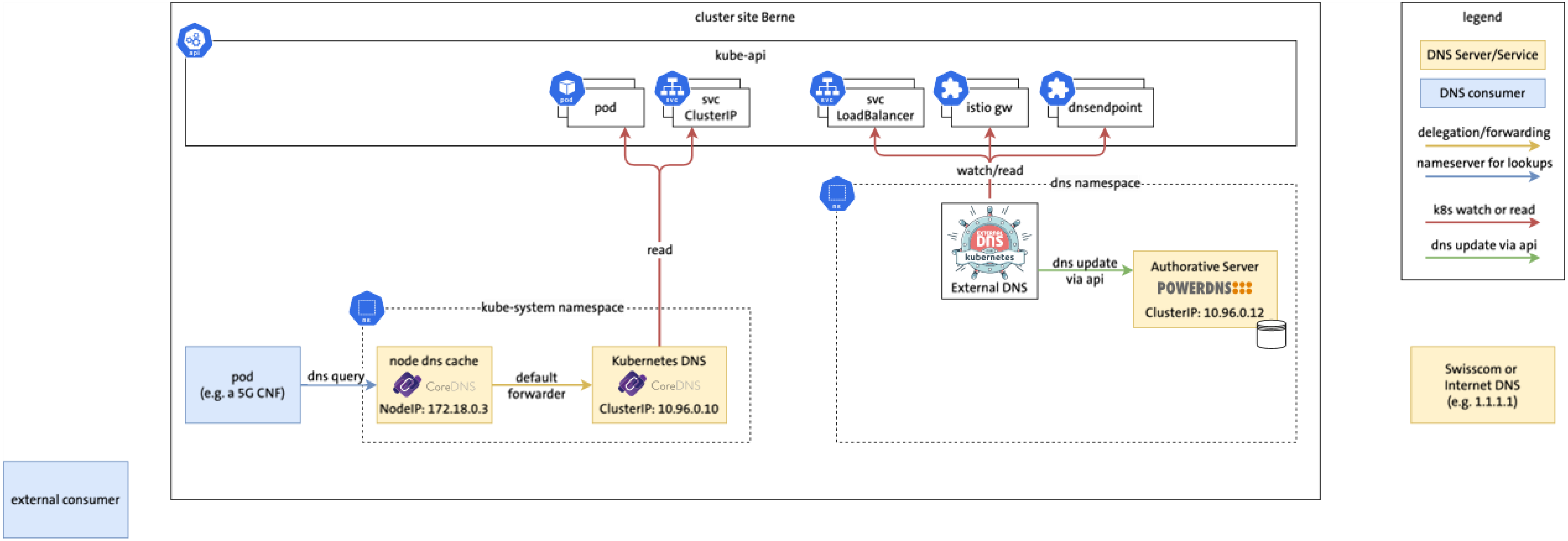
IP read by ExternalDNS

ExternalDNS creates

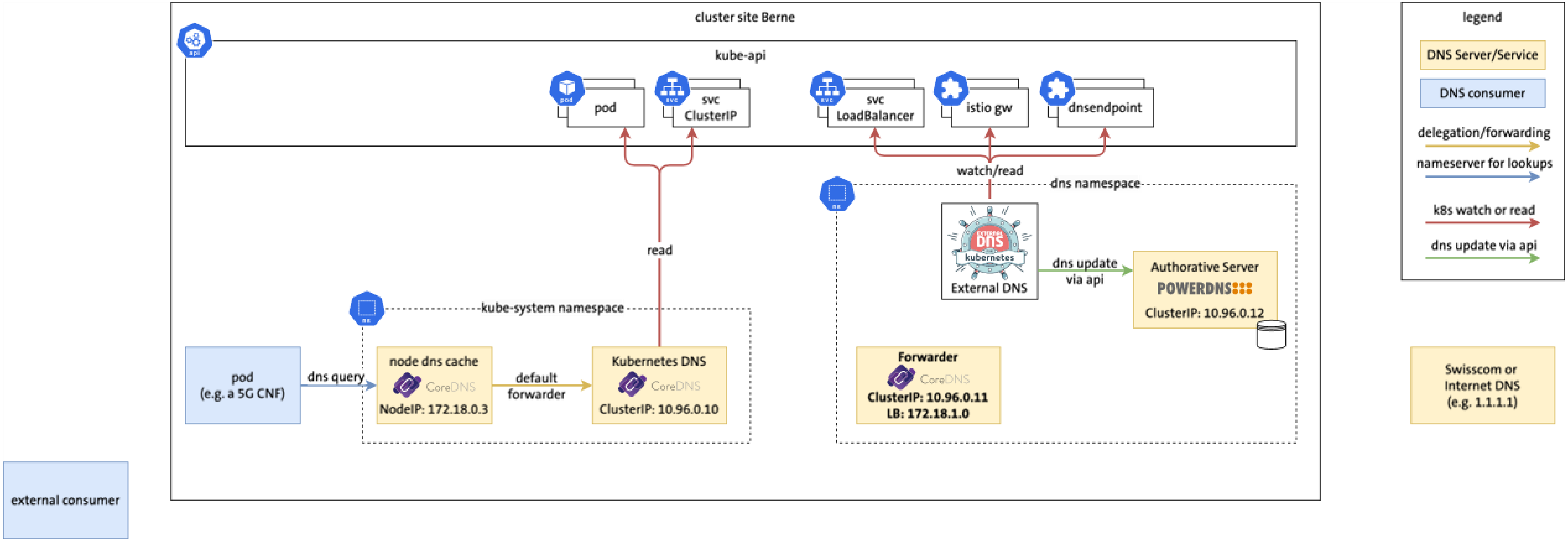
DNS Backend:

```
my-app.example.com. 3600 IN A 192.168.0.35
```

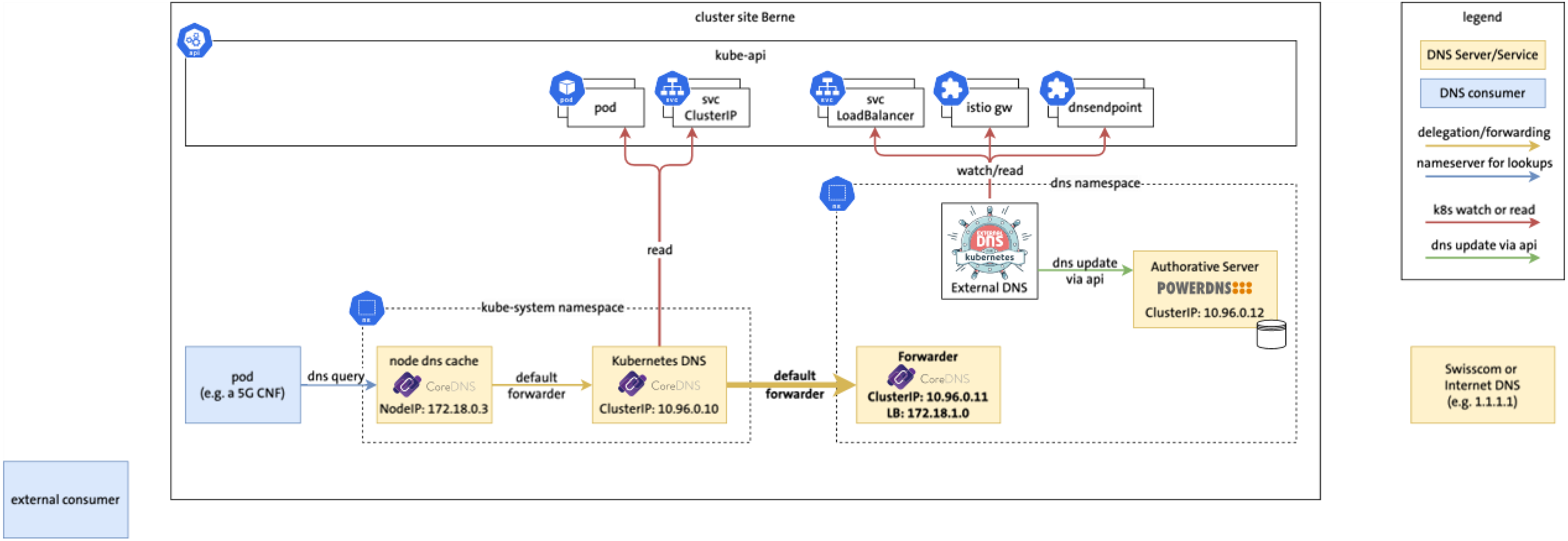
Forwarding to Authoritative Server



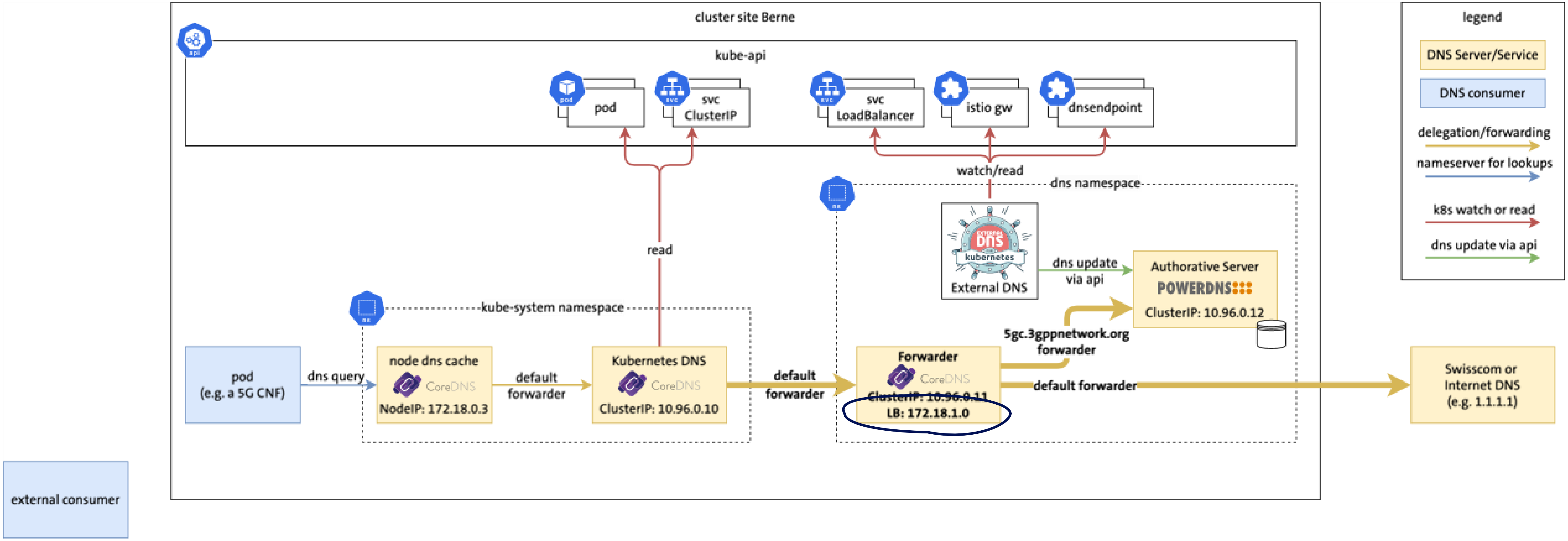
Forwarding to Authoritative Server



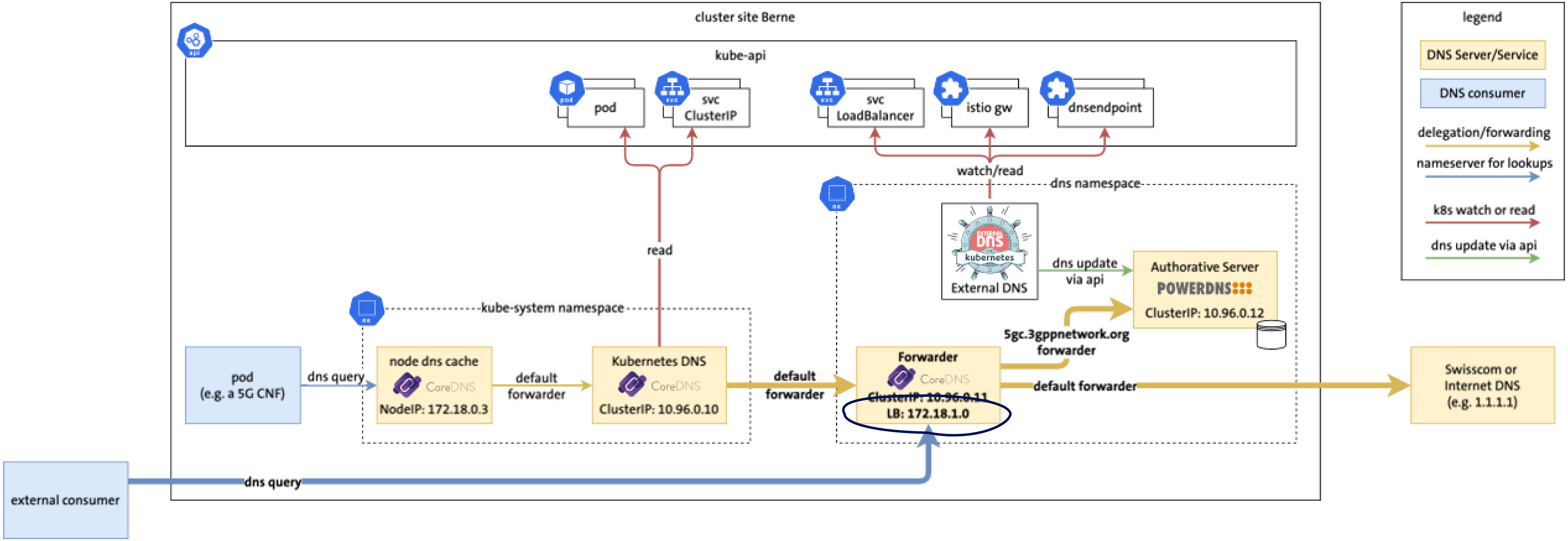
Forwarding to Authoritative Server



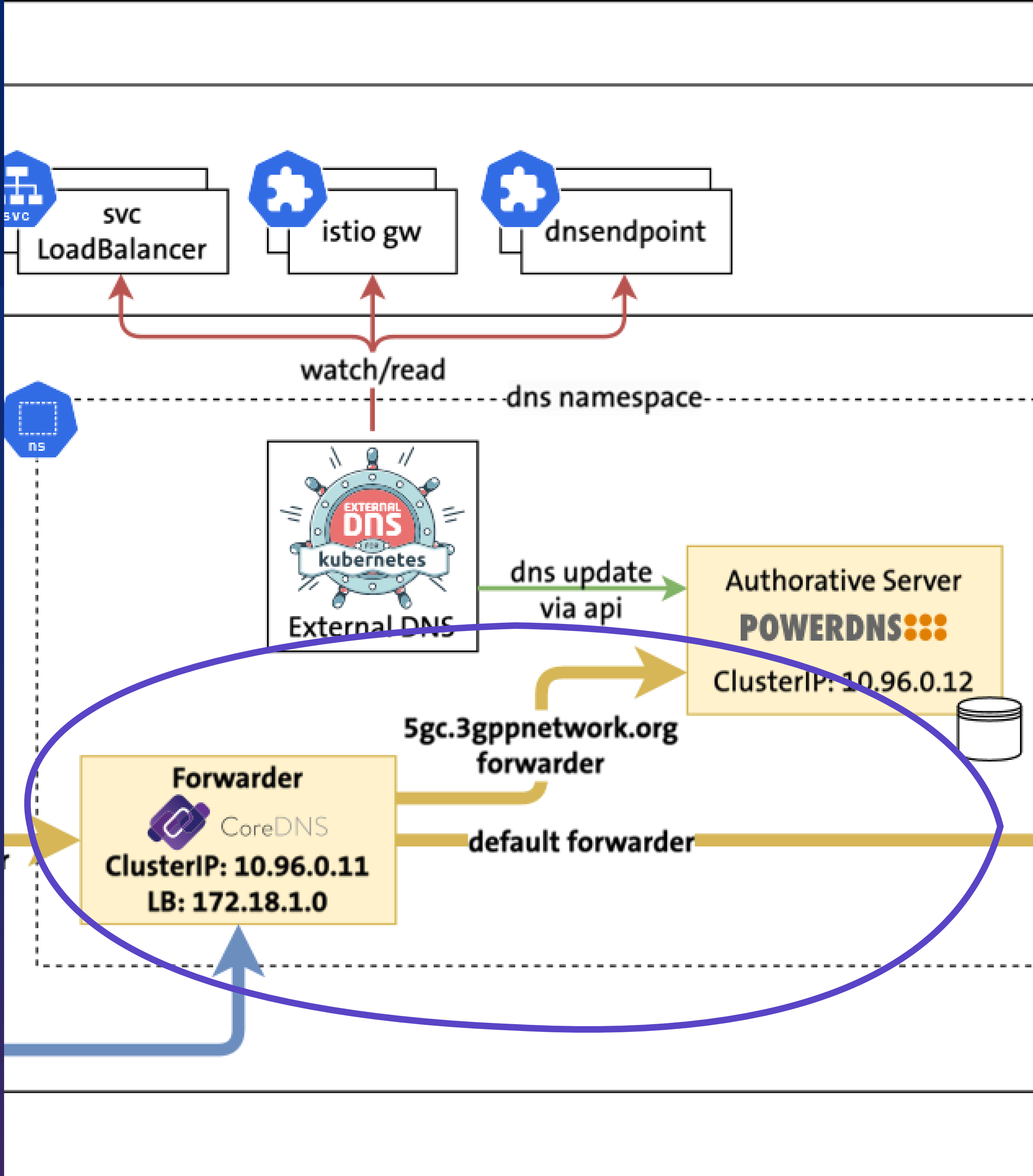
Forwarding to Authoritative Server



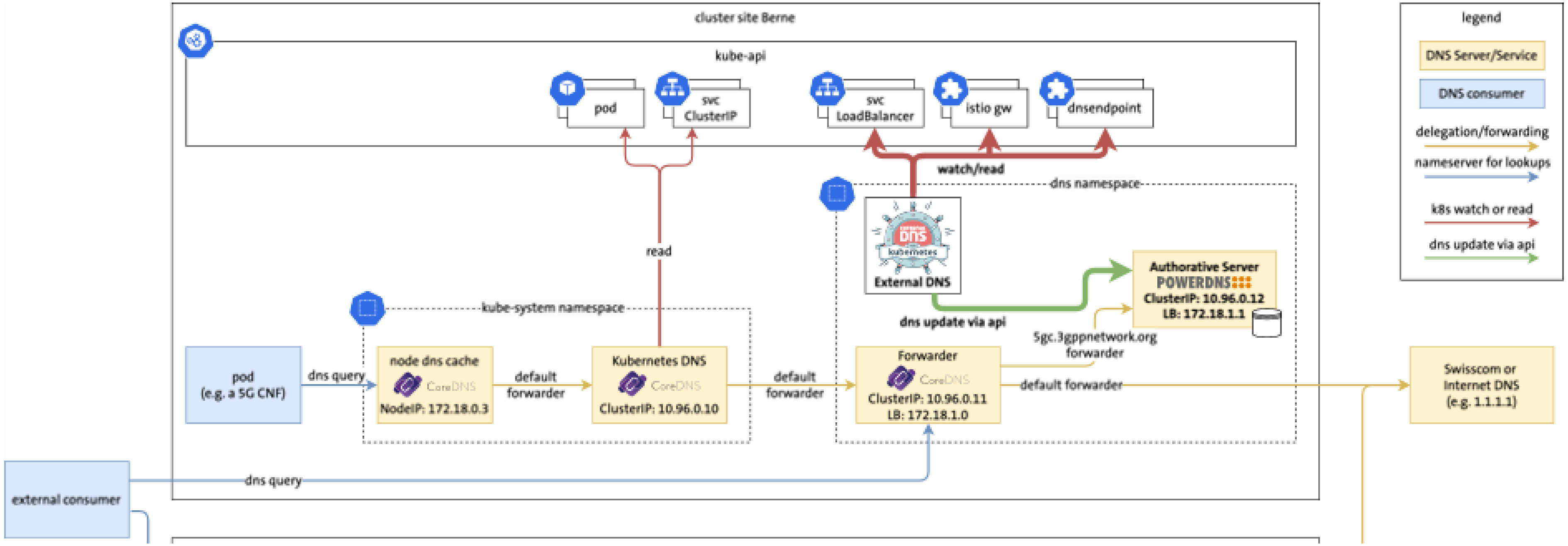
Forwarding to Authoritative Server



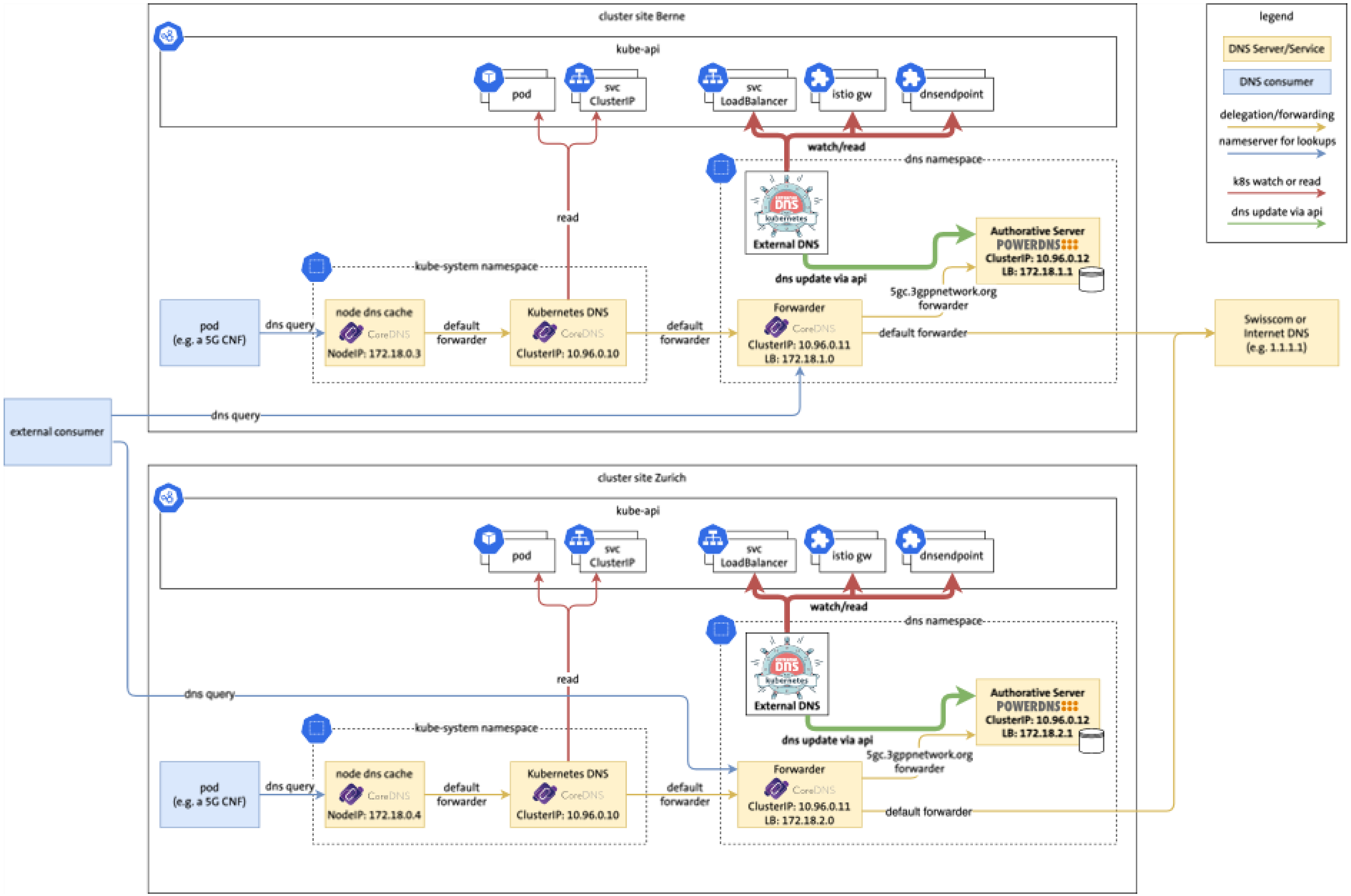
Demo Forwarding



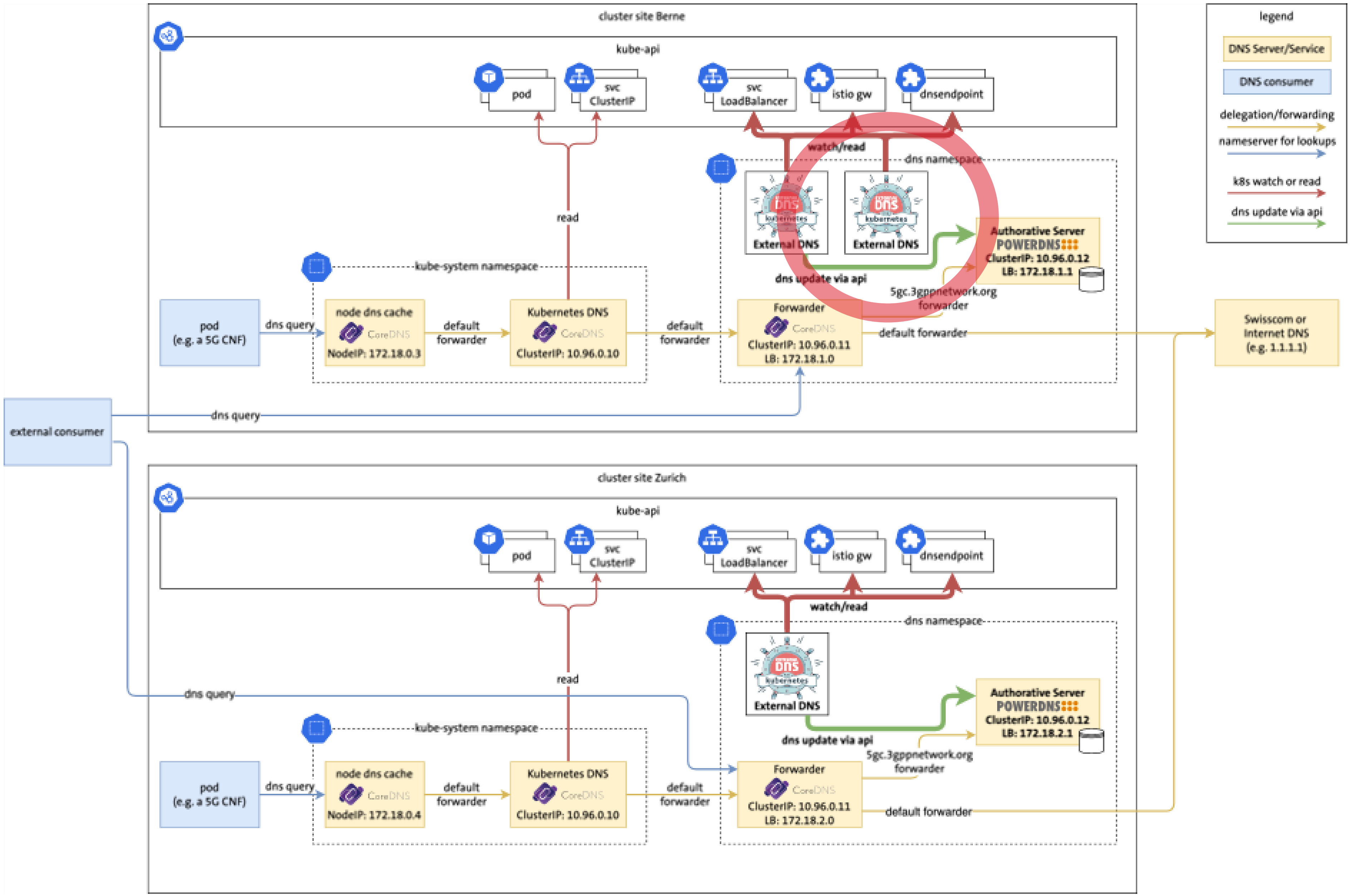
Dual-Cluster Using ExternalDNS



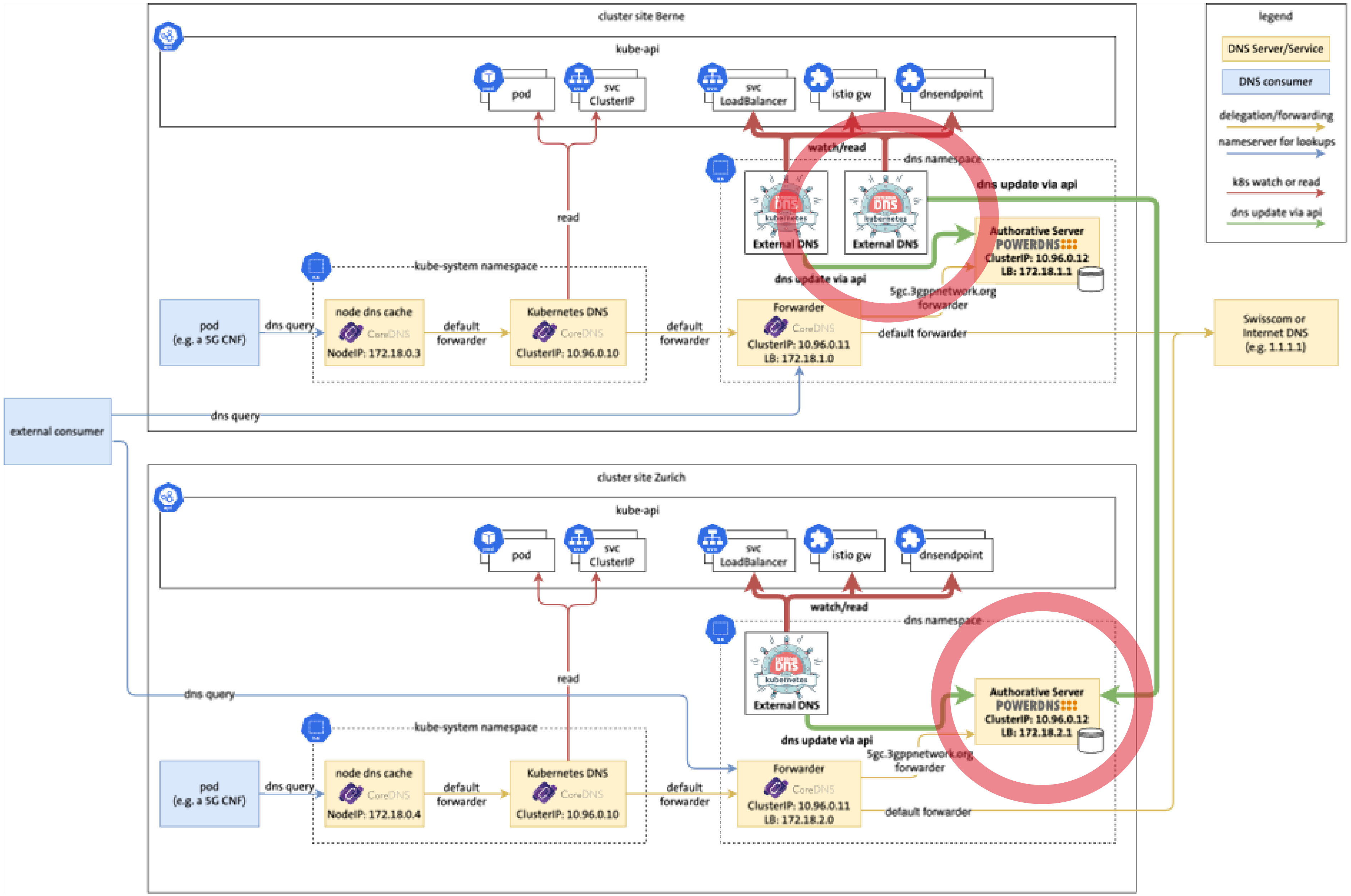
Dual-Cluster Using ExternalDNS



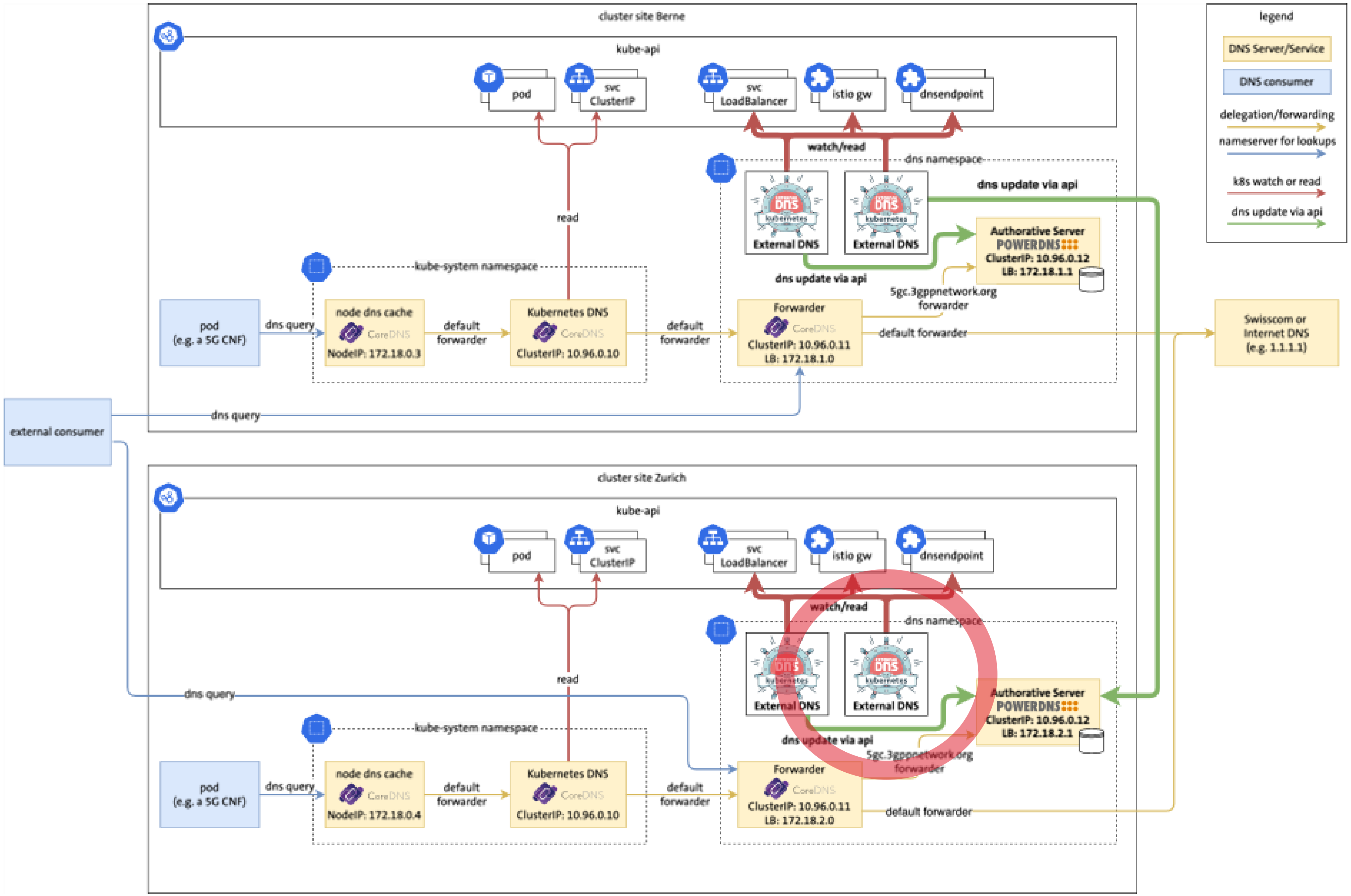
Dual-Cluster Using ExternalDNS



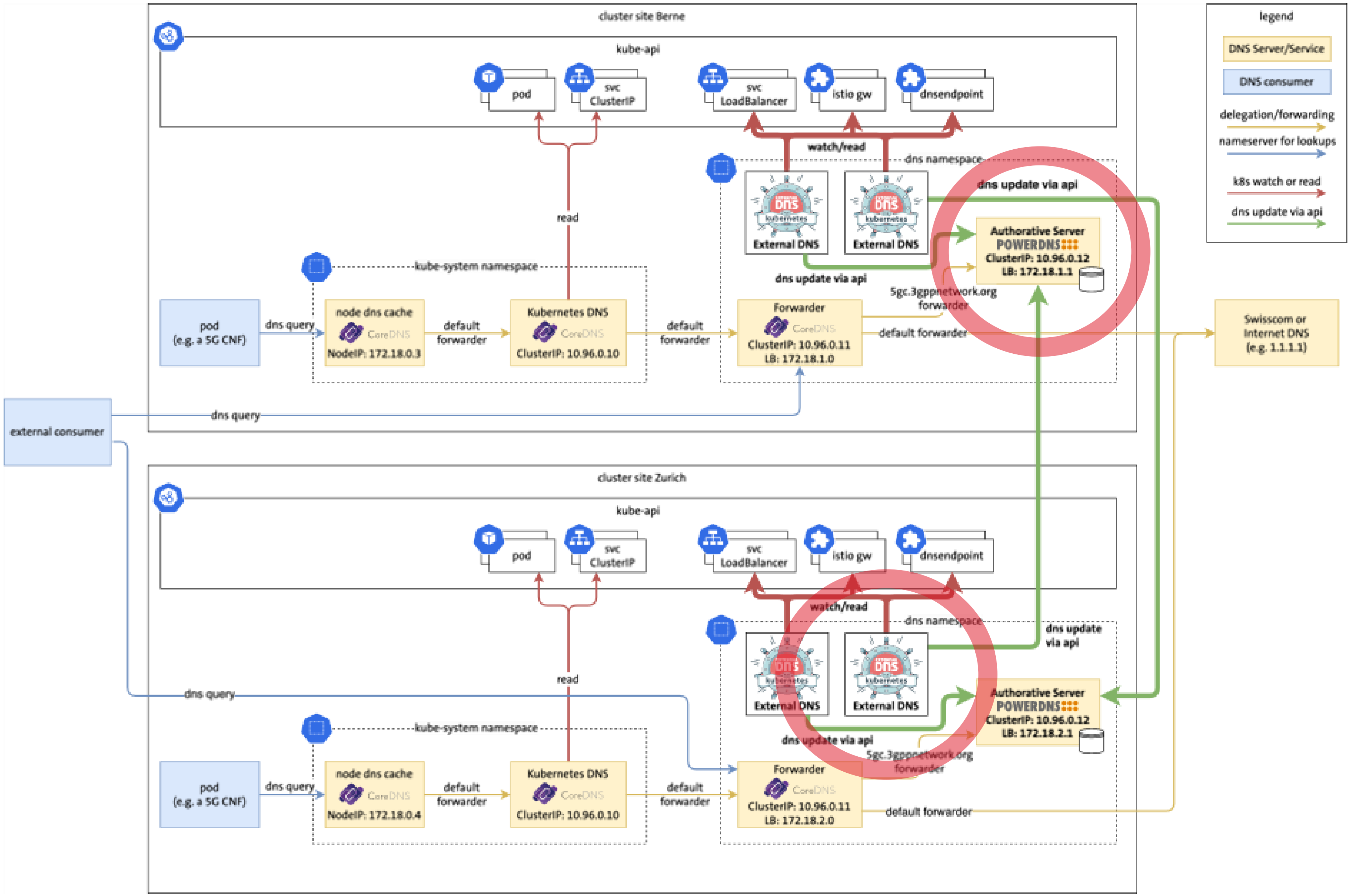
Dual-Cluster Using ExternalDNS



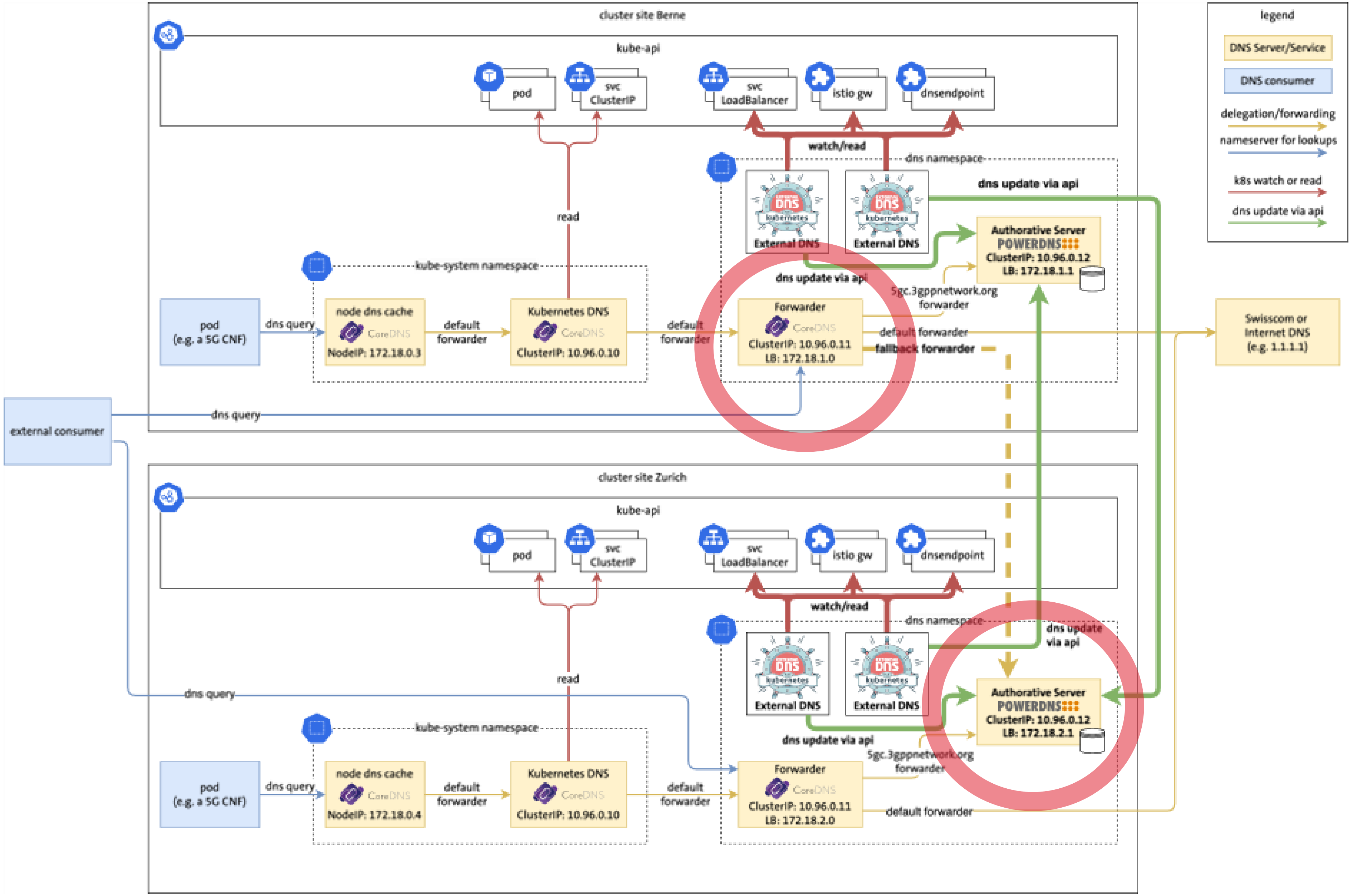
Dual-Cluster Using ExternalDNS



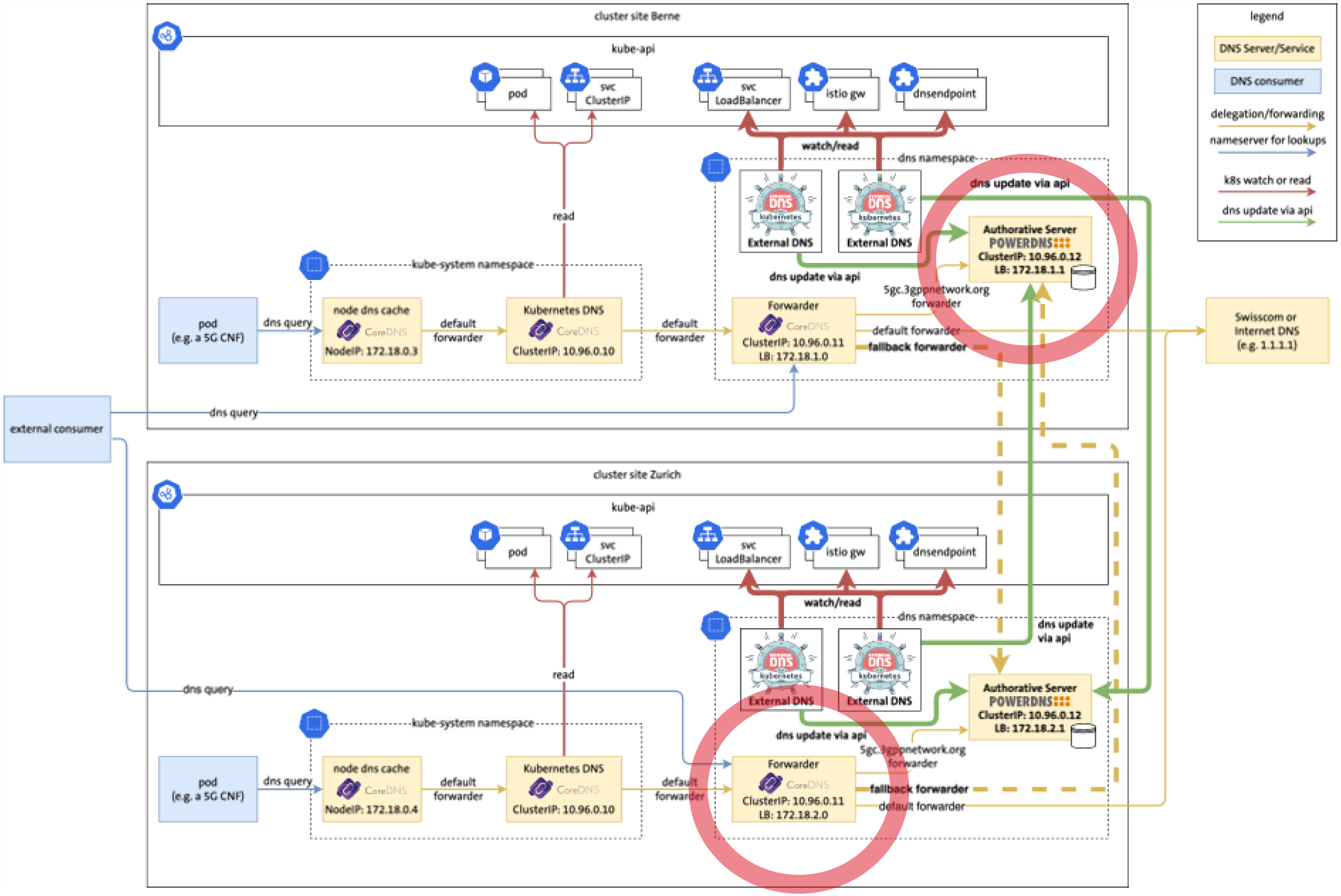
Dual-Cluster Using ExternalDNS



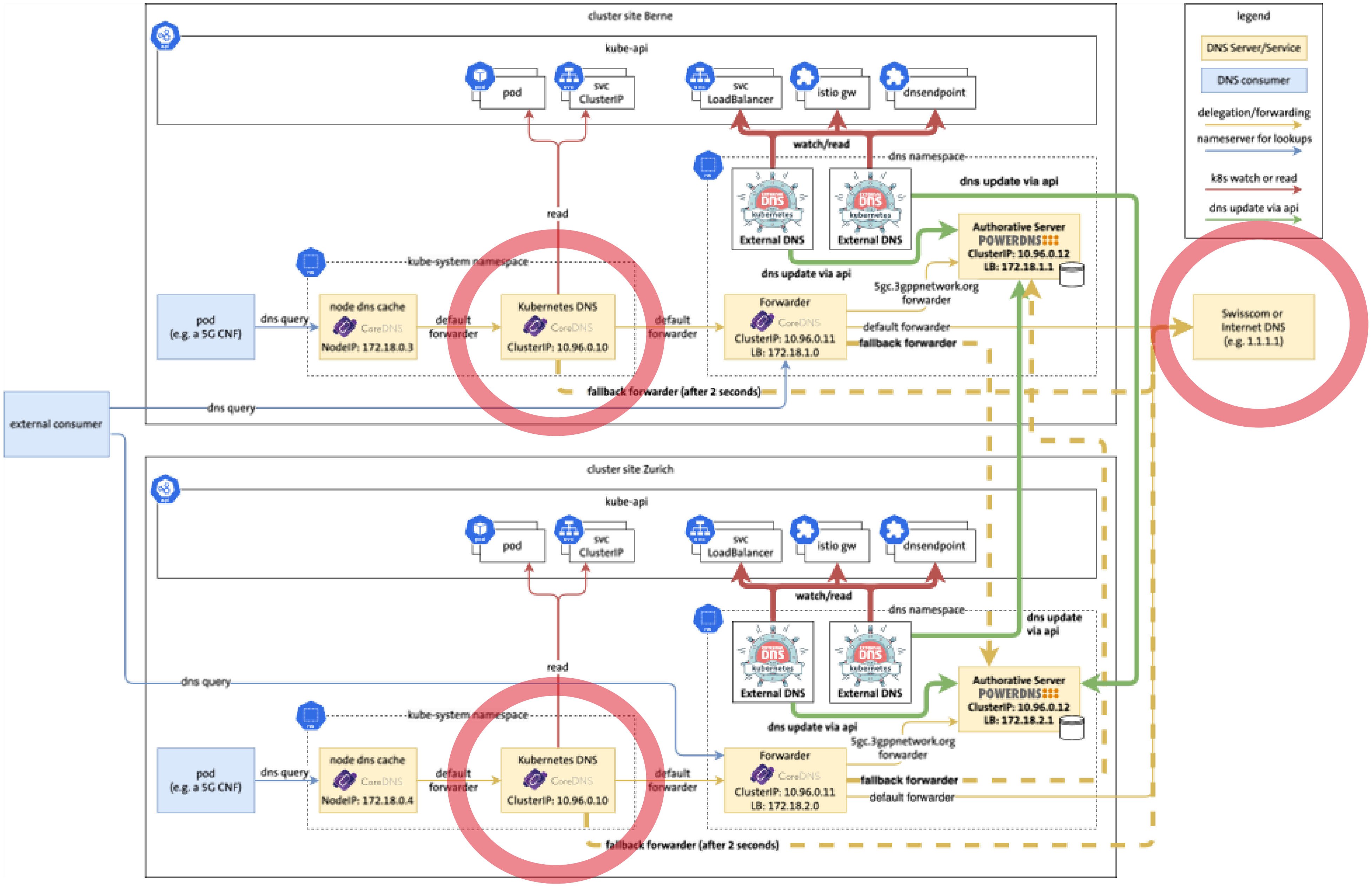
Dual-Cluster Using ExternalDNS



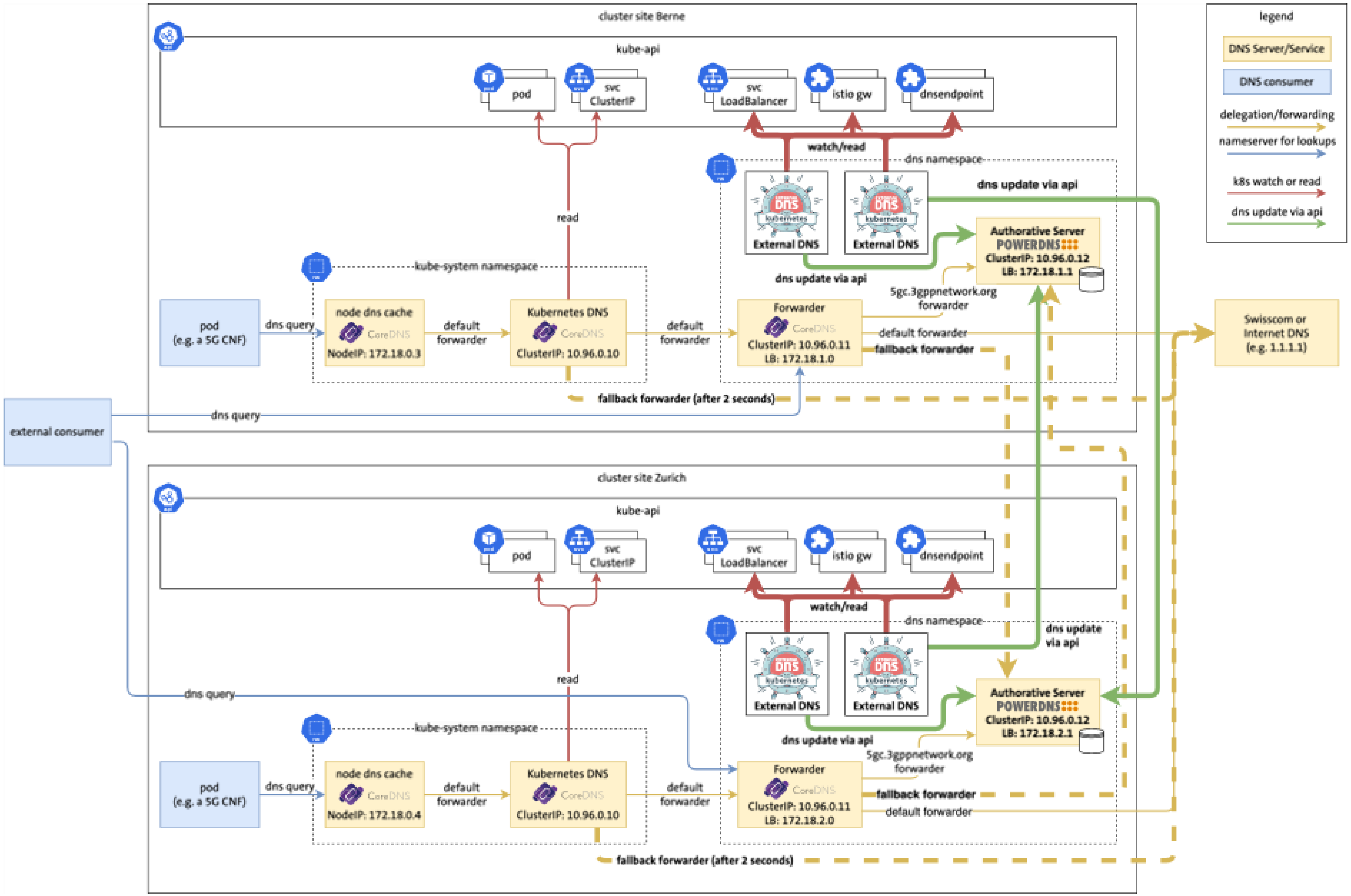
Eliminating Single Point of Failure



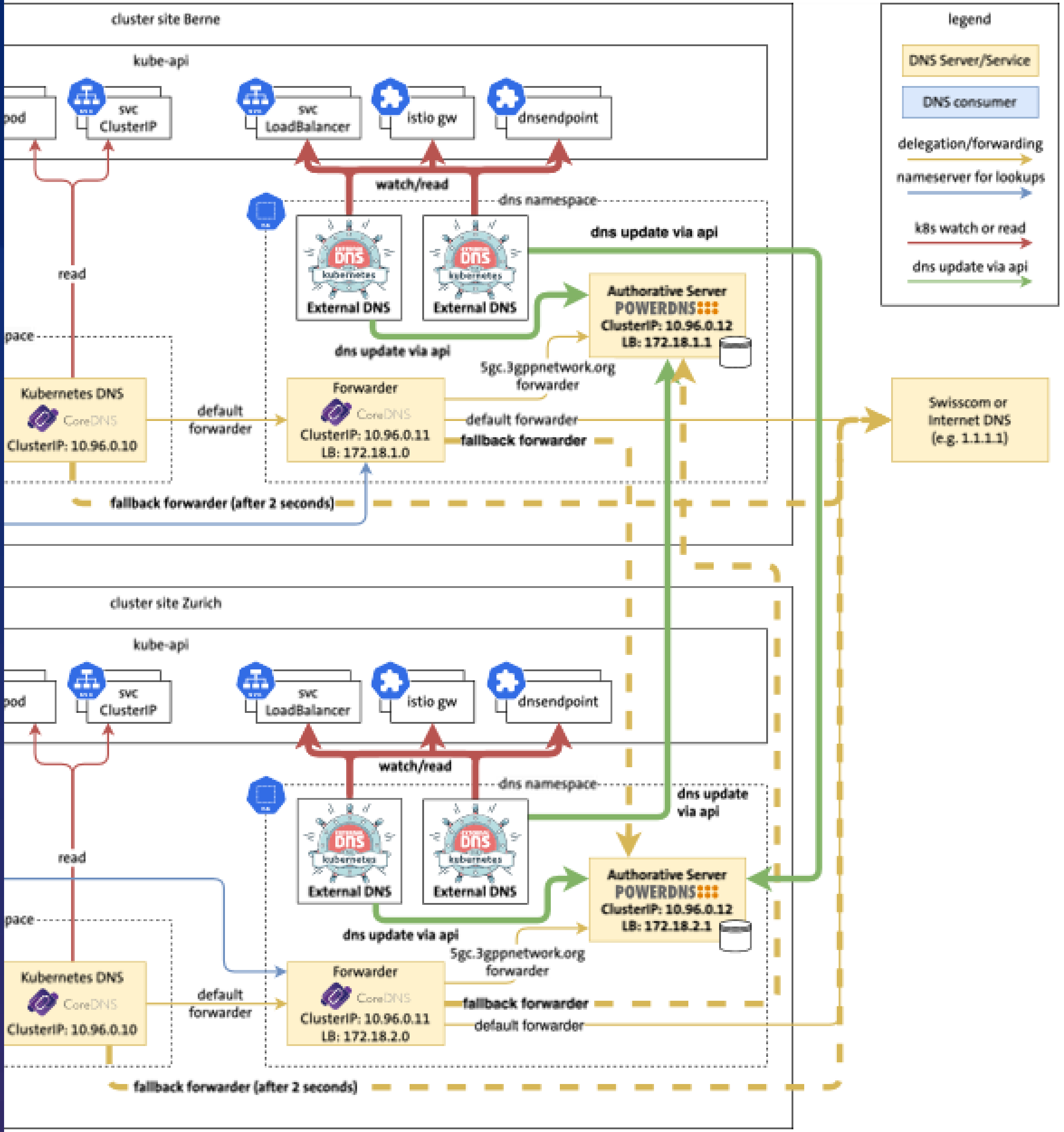
Eliminating Single Point of Failure



Eliminating Single Point of Failure



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

Thanks!

Q&A