High-Available Anycast DNS Resolver for ISPs

Fiona Weber 
aka. vidister

About Fiona Weber

AS9136 Wobcom

local ISP and backbone operator

AS213027 infra.run

hosting (not only) for educational institutions

Freifunk, NGOs, etc.

Real quick:

What again is a DNS resolver?

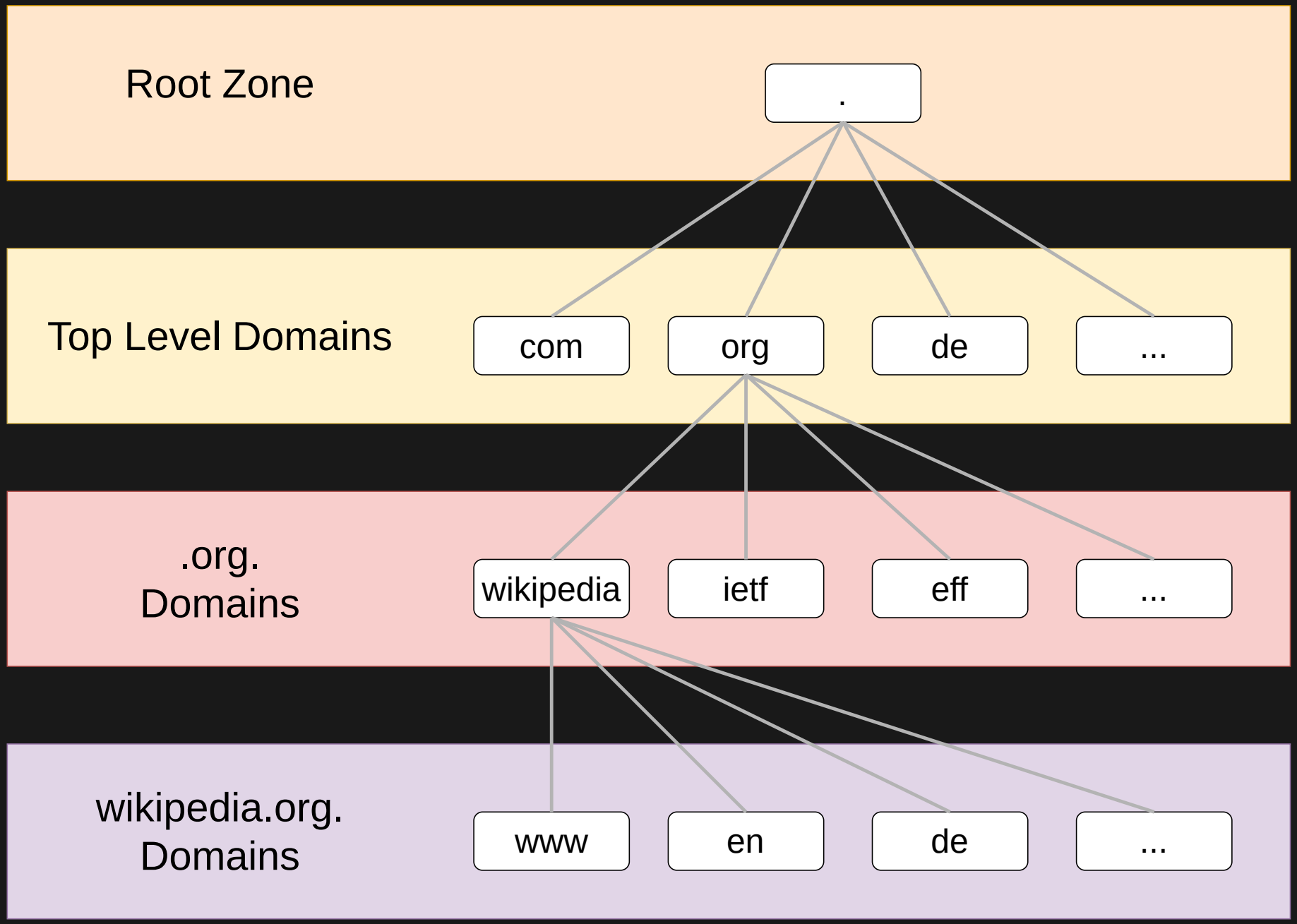
DNS is a hierachy

and to find the answer to our query we have to traverse it

DNS is a hierachy

and to find the answer to our query we have to traverse it

👉 iterative resolution or **recursing**


Root Zone
.
Top Level Domains
com
org
de
...
.org.
Domains
wikipedia
ietf
eff
...
wikipedia.org.
Domains
www
en
de
...

Why should a network provider run their own recursor and not just point to `1.1.1.1`, `8.8.8.8`, etc.?

Privacy

Don’t feed your customers data to big US corporations.

(is that even legal in the EU?)

Resilience

Even hyperscalers have outages.
And then you can’t do anything to fix it for your customers.

Latency

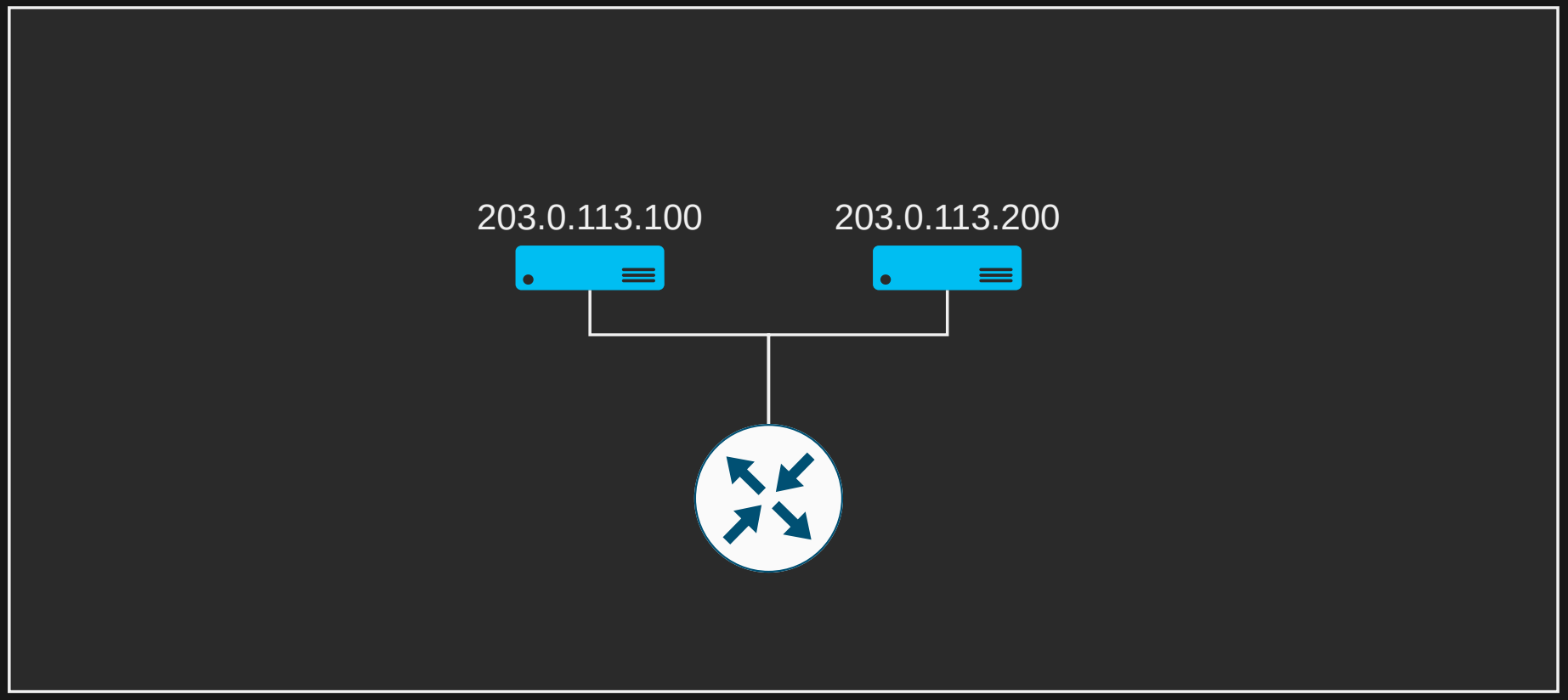
Bring the resolver near to the customer to benefit from local caching

Network Utilization

Some CDN providers (still) do traffic-steering via DNS.

Typical Setup

One L2-Subnet with two DNS servers



Failure Modes:

Failure Modes:

- L2 problems (Spanning tree, Storms, etc.) ☠️

Failure Modes:

- L2 problems (Spanning tree, Storms, etc.) ☠️
- Gateway problems (broken/slow VRRP failover) ☠️

Failure Modes:

- L2 problems (Spanning tree, Storms, etc.) ☠️
- Gateway problems (broken/slow VRRP failover) ☠️
- Site down ☠️

Failure Modes:

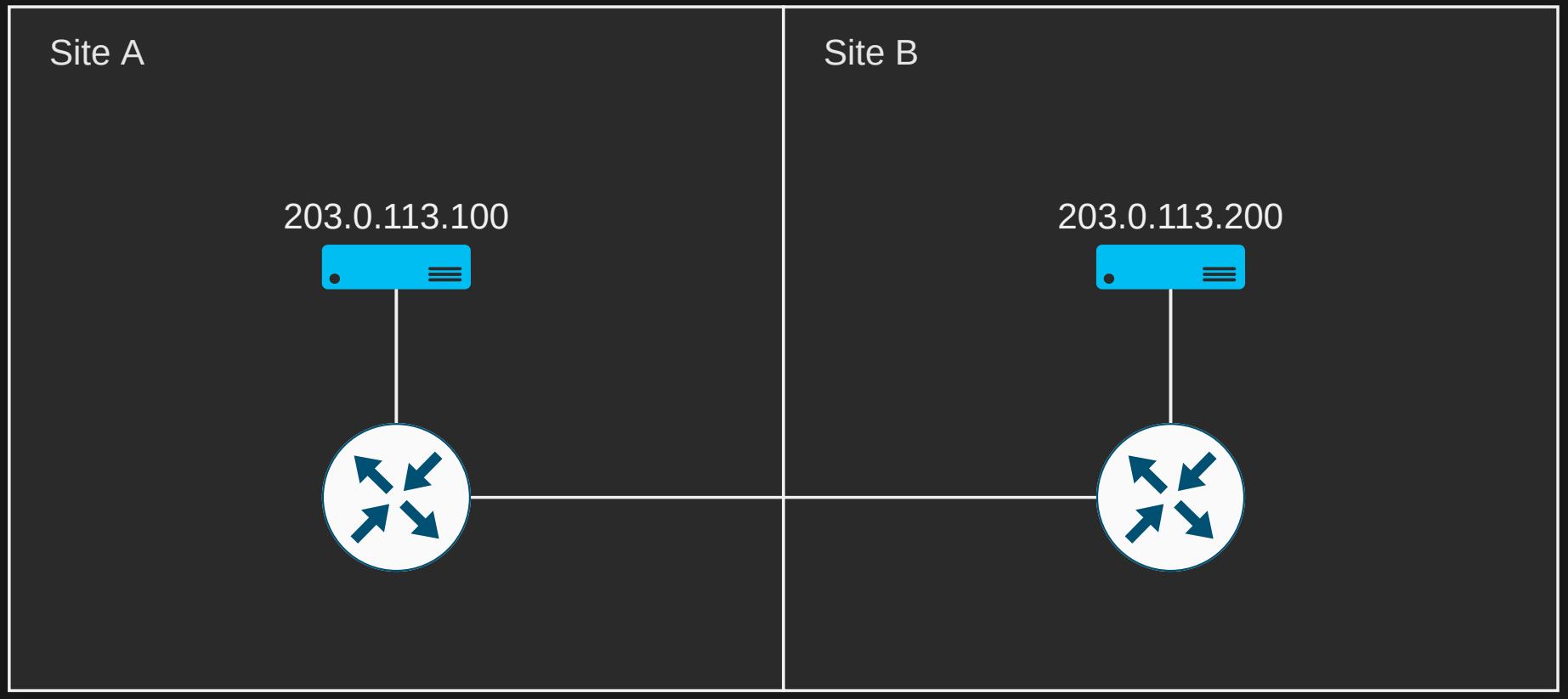
- L2 problems (Spanning tree, Storms, etc.) ☠️
- Gateway problems (broken/slow VRRP failover) ☠️
- Site down ☠️
- Single Server unavailable 🐢

Failure Modes:

- L2 problems (Spanning tree, Storms, etc.) ☠️
- Gateway problems (broken/slow VRRP failover) ☠️
- Site down ☠️
- Single Server unavailable 🐢
- Server load 🐢

“Advanced” typical Setup

Two or more DNS Resolver in independent subnets, maybe even independent sites



Failure Modes:

-
-
-

Failure Modes:

- ~~L2 problems (Spanning tree, etc.)~~ ☠️
-
-

Failure Modes:

- ~~L2 problems (Spanning tree, etc.)~~ ☠️
- ~~Gateway problems (broken/slow VRRP failover)~~ ☠️
-

Failure Modes:

- ~~L2 problems (Spanning tree, etc.)~~ ☠️
- ~~Gateway problems (broken/slow VRRP failover)~~ ☠️
- ~~Site down~~ ☠️

Failure Modes:

- ~~L2 problems (Spanning tree, etc.)~~ ☠️
- ~~Gateway problems (broken/slow VRRP failover)~~ ☠️
- ~~Site down~~ ☠️
- Single Server unavailable 🐢

Failure Modes:

- ~~L2 problems (Spanning tree, etc.)~~ ☠️
- ~~Gateway problems (broken/slow VRRP failover)~~ ☠️
- ~~Site down~~ ☠️
- Single Server unavailable 🐢
- Server load 🐢

The Solution:

Anycast ✨

Anycast

Multiple servers share the same IP adress(es)

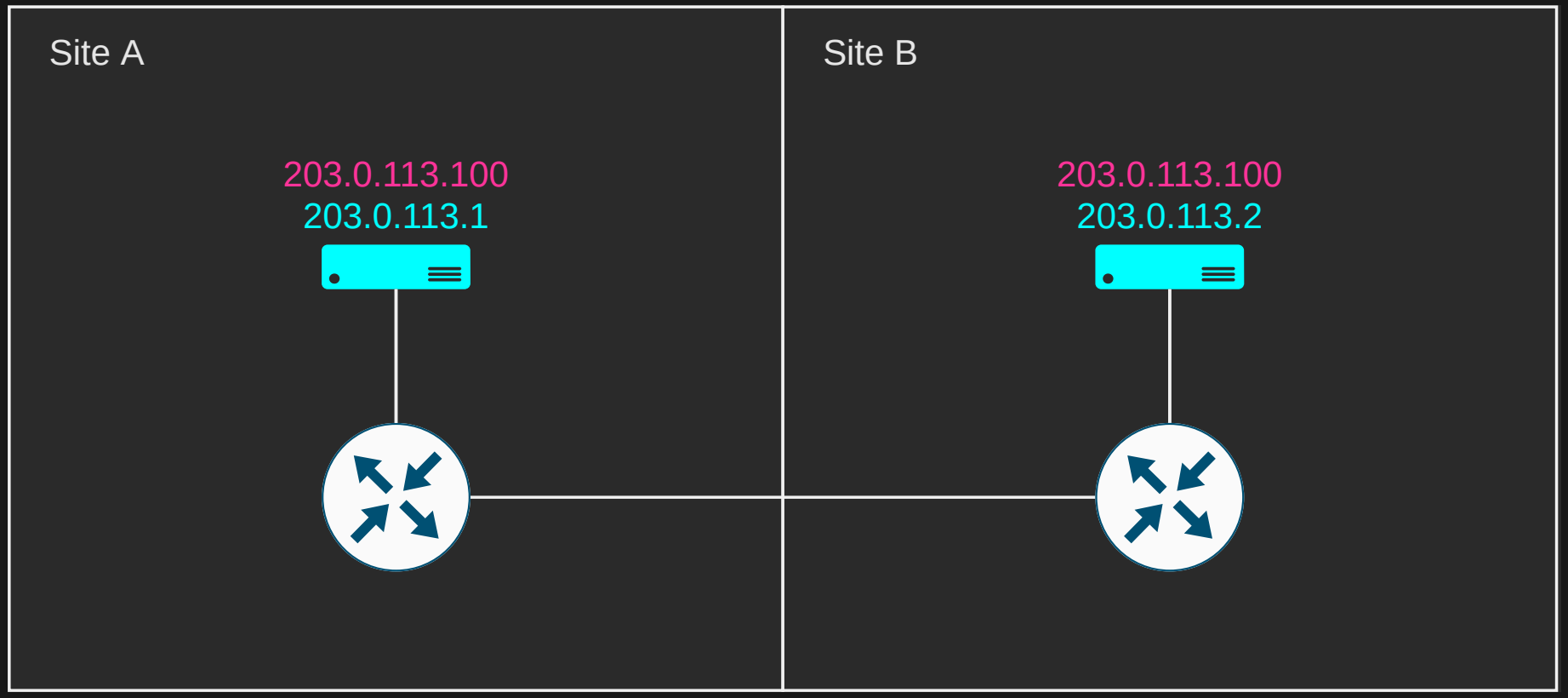
Anycast

Multiple servers share the same IP adress(es)

The network routes the packets to the nearest server(s)

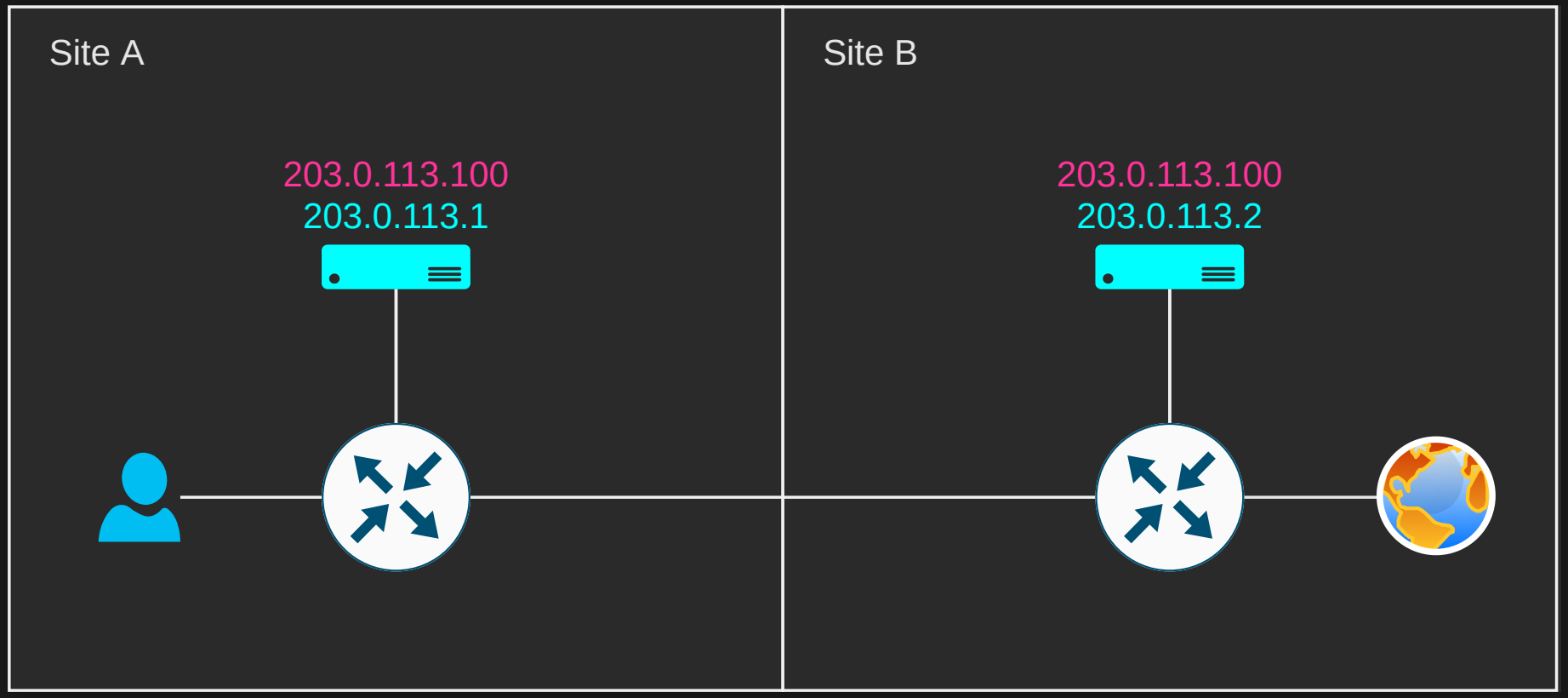
Simple anycast setup

Two or more independent resolvers sharing an IP address



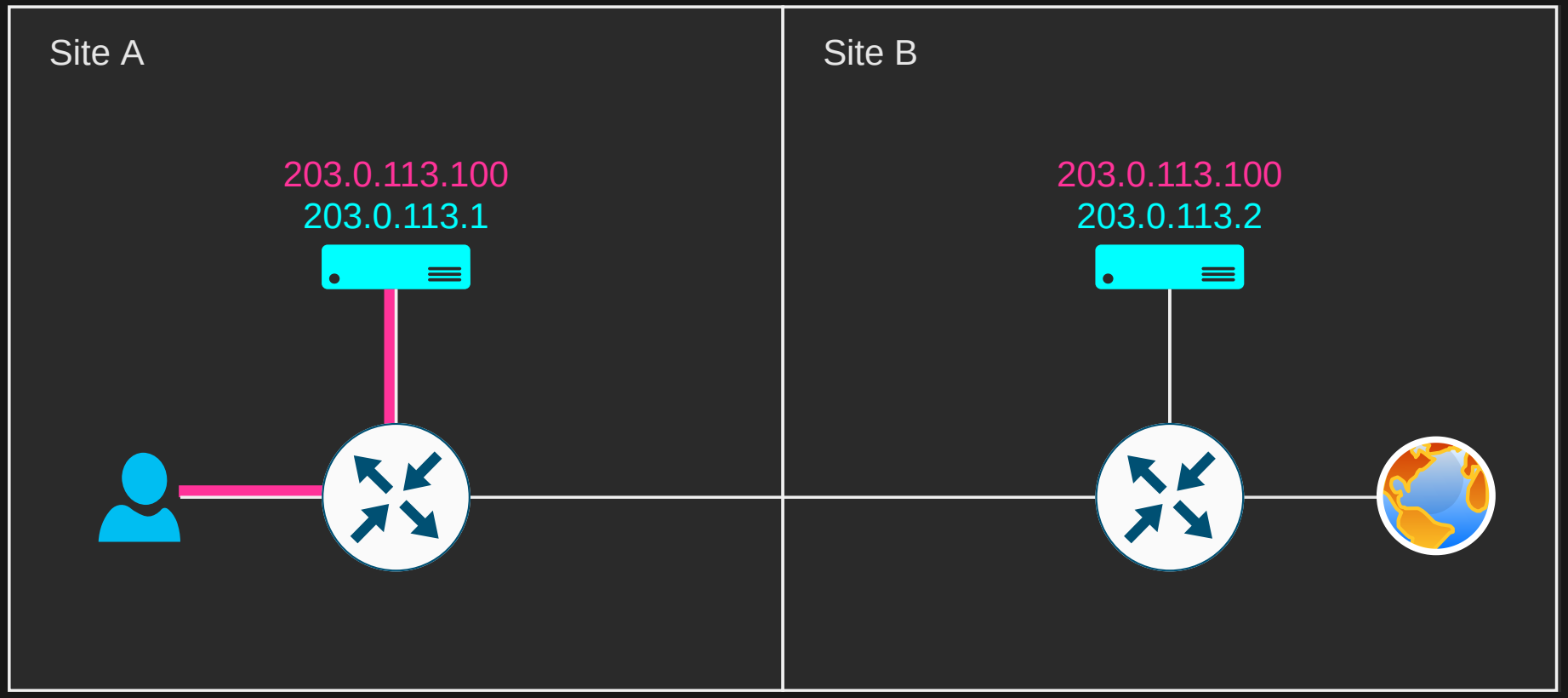
The life of an DNS query

Our client wants to resolve a DNS query



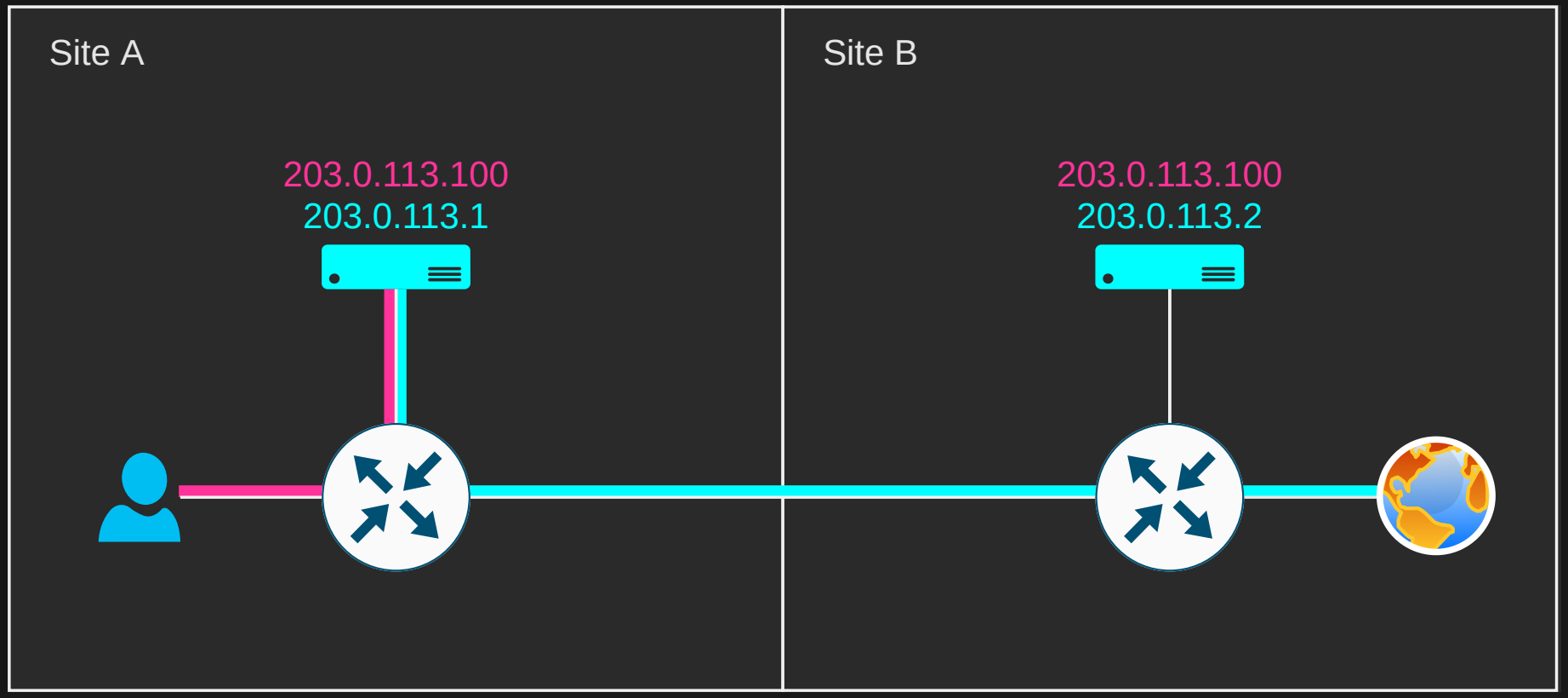
The life of an DNS query

Router forwards the request the nearest server



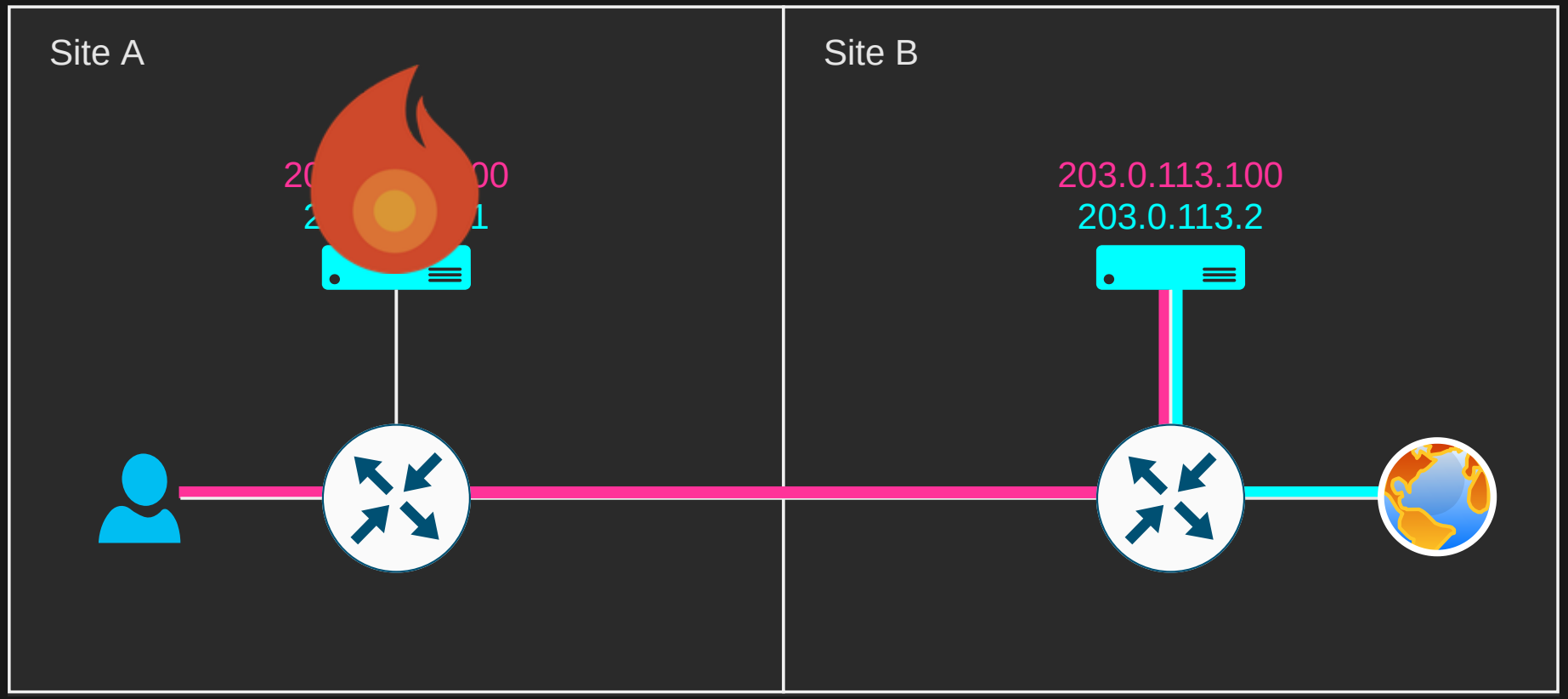
The life of an DNS query

Server recurses the request using it’s unique IP



The life of an DNS query

…if something happens to our nearest server we just use the next one



How does this work?

I’m not allowed to use addresses multiple times in Layer 2 segments…

How does this work?

I’m not allowed to use addresses multiple times in Layer 2 segments…

…that’s why we need routes.

BGP ✨

Just use the same routing protocol we’re using anyways

BGP Daemon

- bird
  - powerful
  - also speaks other protocols (OSPF, RIP, BFD, ...)
- exabgp
  - simple
  - easy lo-tec health-checks

Configure the IPs on `lo`

Example for ifupdown2:

```
# /etc/network/interfaces.d/lo.cfg
auto lo
iface lo inet loopback
  address 203.0.113.100/32
  address 2001:0DB8::100/128
```

Configure the routing daemon

Simplified example config snippet for Bird 1:

```
# /etc/bird/bird.conf
protocol direct direct1 {
    interface "lo";
}
filter ALLOWED_ANYCAST {
    if (net = 203.0.113.100/32 ) then accept;
    reject; # Reject anything else
}
protocol bgp ROUTER01 {
    local as 4200001312;
    import none;
    export filter ALLOWED_ANYCAST;
    neighbor 203.0.113.1 as 4200000000;
}
```

et voilà !

```
inet.0: 853944 destinations, 1850542 routes
  Prefix                  Nexthop          AS path
* 203.0.113.100/32         203.0.113.10     4200001312 I
                           203.0.113.11     4200001312 I
                           203.0.113.12     4200001312 I
```

DNS Recursor

DNS Recursor

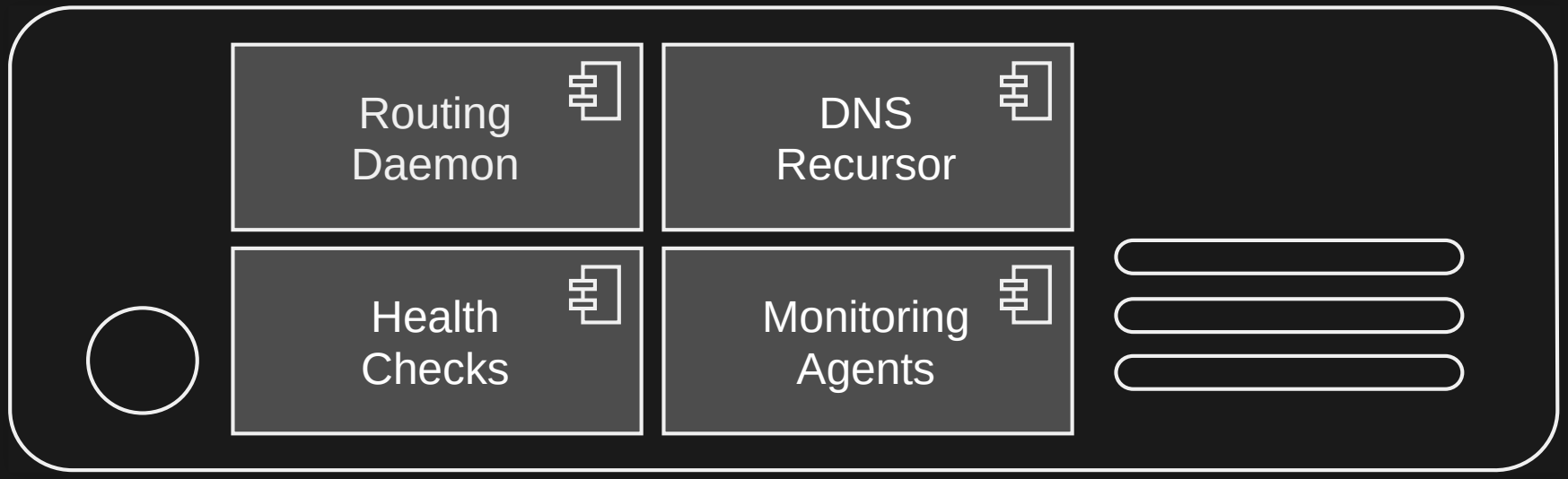
- PowerDNS-Recursor
- Unbound
- Knot Resolver

DNS Recursor

- PowerDNS-Recursor
- Unbound
- Knot Resolver

Bonus points: Use two implementations for better resilience (comes with more maintenance effort)

High availability



Health-Checks

Drain the server if something goes wrong

Health-Checks

Drain the server if something goes wrong

Tired: Run a systemd.timer and stop the routing daemon

Health-Checks

Drain the server if something goes wrong

Tired: Run a systemd.timer and stop the routing daemon

Wired: Manipulate export filters based on the health

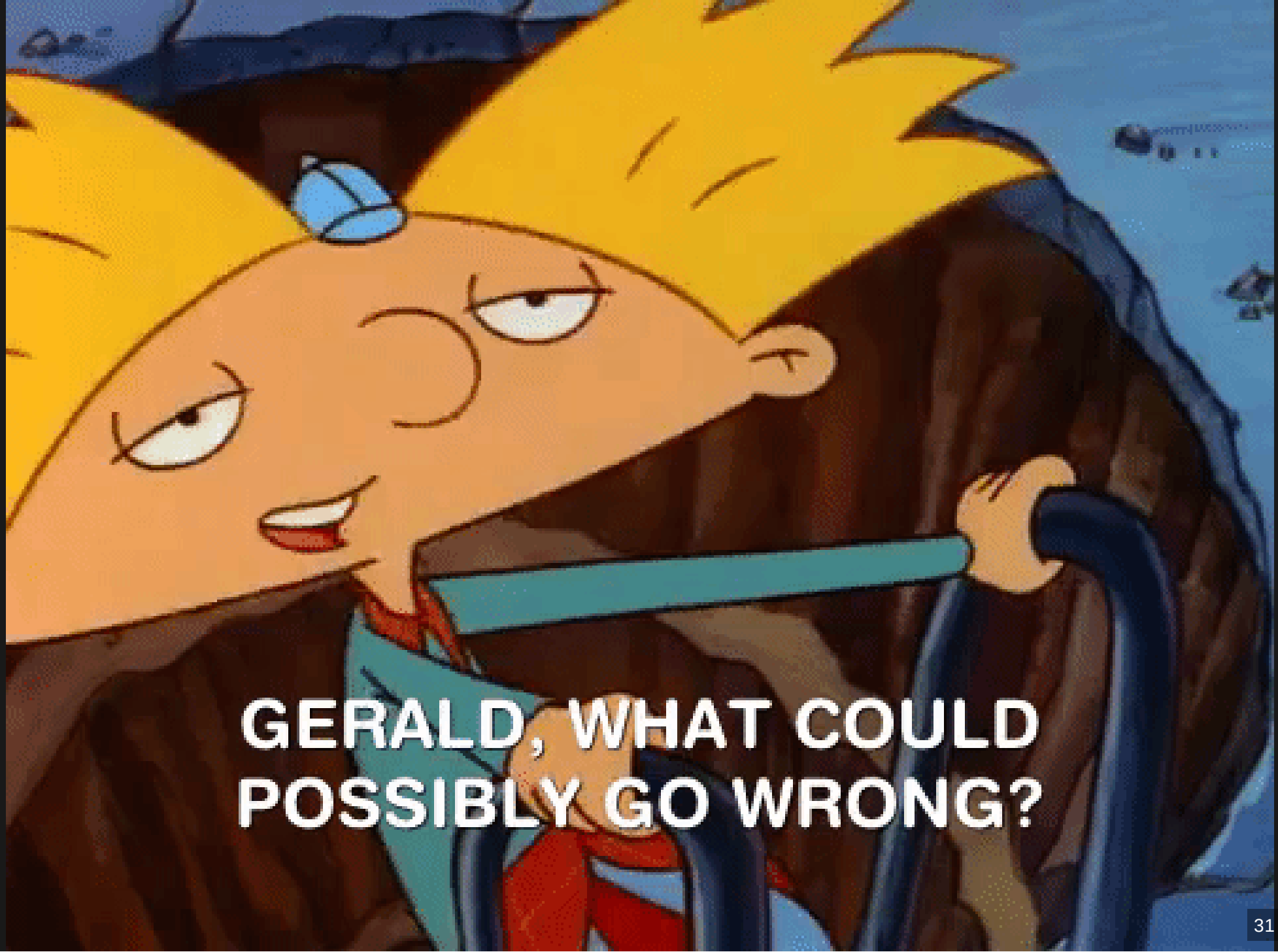
Health-Checks

Drain the server if something goes wrong

Tired: Run a systemd.timer and stop the routing daemon

Wired: Manipulate export filters based on the health

Example: github.com/unixsurfer/anycast_healthchecker


GERALD, WHAT COULD
POSSIBLY GO WRONG?

Health-Checks

We don’t want to stop all our DNS Resolvers when Facebook or Cloudflare goes down

Health-Checks

We don’t want to stop all our DNS Resolvers when Facebook or Cloudflare goes down

- Check if the daemon is running

Health-Checks

We don’t want to stop all our DNS Resolvers when Facebook or Cloudflare goes down

- Check if the daemon is running
- Check if it is accepting connections (TCP DNS)

Health-Checks

We don’t want to stop all our DNS Resolvers when Facebook or Cloudflare goes down

- Check if the daemon is running
- Check if it is accepting connections (TCP DNS)
- Add a local zone to check against

Health-Checks

We don’t want to stop all our DNS Resolvers when Facebook or Cloudflare goes down

- Check if the daemon is running
- Check if it is accepting connections (TCP DNS)
- Add a local zone to check against
- Check if ❗ multiple ❗ independent major websites are resolvable

Safety rope 

👉 Increase the MED instead of pulling the routes.

Monitoring (+ Alerting)

Software metrics

QPS, Cache hit ratios, etc.

Software metrics

QPS, Cache hit ratios, etc.

unbound_exporter
powerdns-recursor prometheus endpoint

Server metrics

Load, Memory, open sockets, conntrack, etc.

Server metrics

Load, Memory, open sockets, conntrack, etc.

node_exporter
systemd_exporter

End-to-end tests

End-to-end tests

Periodically send some queries to the static IP of the server

Alert when they fail or take too long

End-to-end tests

Periodically send some queries to the static IP of the server

Alert when they fail or take too long

blackbox_exporter

Placement

Placement

- Optimize for latency
  - Near the customer
  - btw: Set latency-based IGP metrics

Placement

- Optimize for latency
  - Near the customer
  - btw: Set latency-based IGP metrics
- Keep failure domains small
  - Avoid shared storages

Placement

- Optimize for latency
  - Near the customer
  - btw: Set latency-based IGP metrics
- Keep failure domains small
  - Avoid shared storages

👉 Throw a lone Hypervisor with local storage in every bigger PoP


Scale it up

Horizontally

Horizontally

Just build more of them and enable multipathing

Vertically

Optimize cache hit ratios

- Software choice
- Machine sizing

Vertically

Optimize cache hit ratios

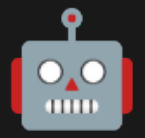
- Software choice
- Machine sizing

👉 Keep headroom.
Facebook outages do happen ;)

Vertically

Fine tuning:

- sysctls (source ports, conntrack, buffer sizes, etc.)
- ulimits
- NIC tuning (number of queues, buffer, …)

Automation 

Automation 🤖

- Bolting together servers by hand is error prone

Automation 🤖

- Bolting together servers by hand is error prone
- Every server should be identical

Automation 🤖

- Bolting together servers by hand is error prone
- Every server should be identical
- Make it easy to deploy more

Thank you! Questions?

Feel free to reach out

- E-Mail:
  - fiona.weber@wobcom.de
  - denog@vidister.de
- Matrix:
  - @vidister:entropia.de

Links

DNS Shotgun

https://indico.dns-oarc.net/event/34/contributions/782/attachments/769

Resolver Bencharking tool

https://gitlab.nic.cz/knot/resolver-benchmarking/

Sysctl tuning

https://gitlab.nic.cz/knot/resolver-benchmarking/-/blob/master/roles/tuning/files/sysctl_