

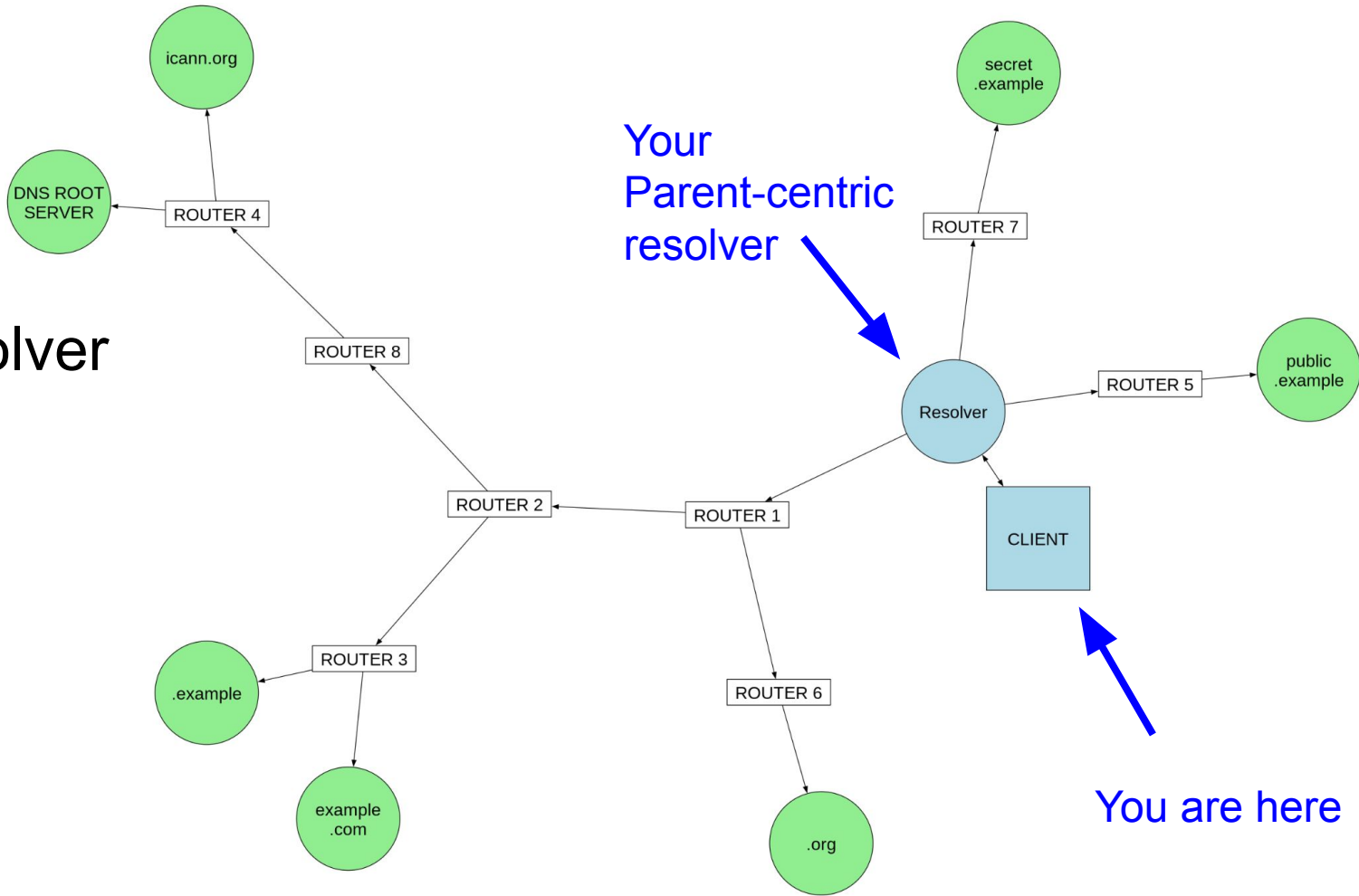
Subverting Parent-Centric DNSSEC Validating Resolvers

Wes Hardaker <ietf@hardakers.net>

Imagine
Your
Network's
DNS Resolver

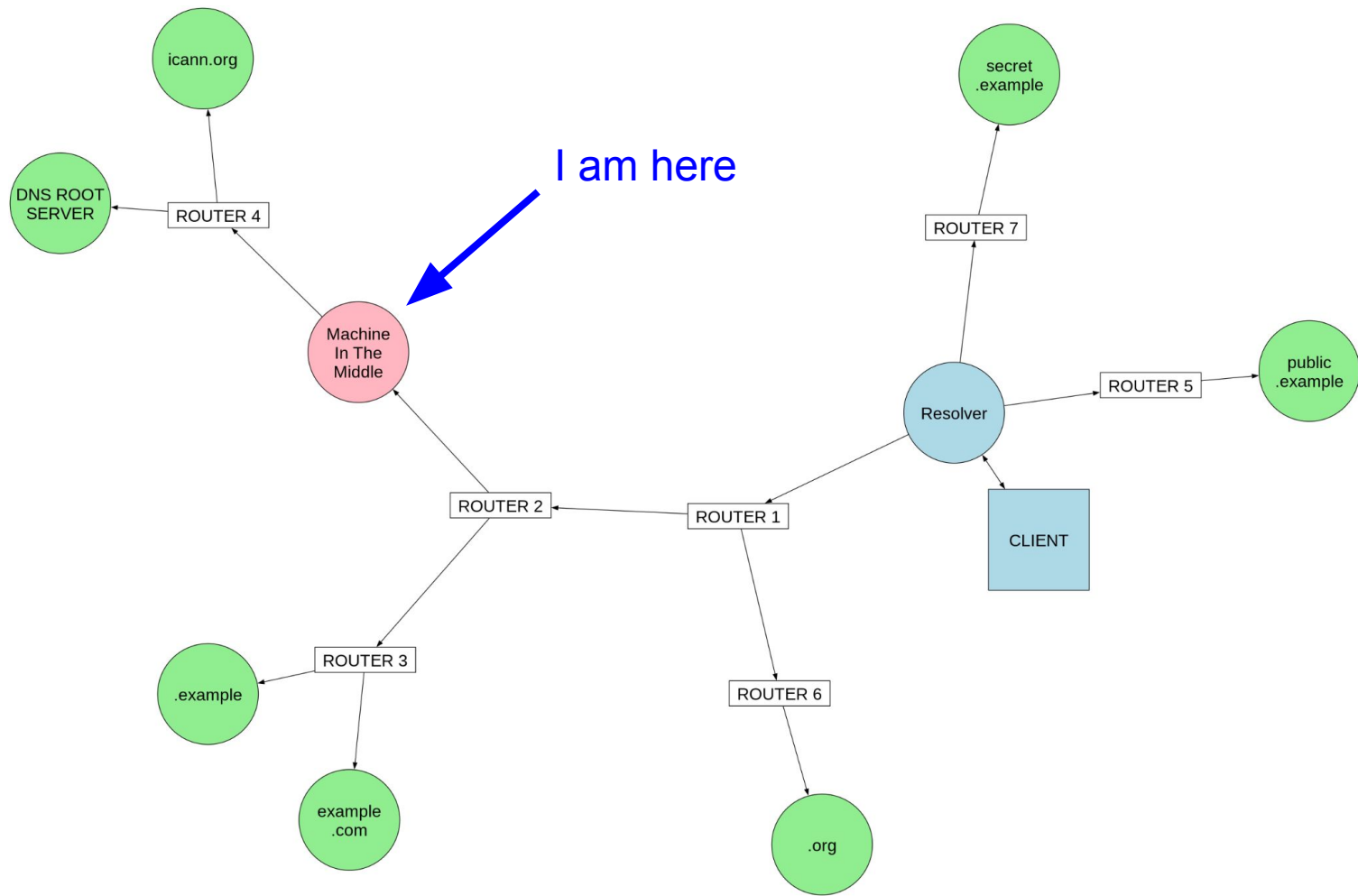
and

The
Internet



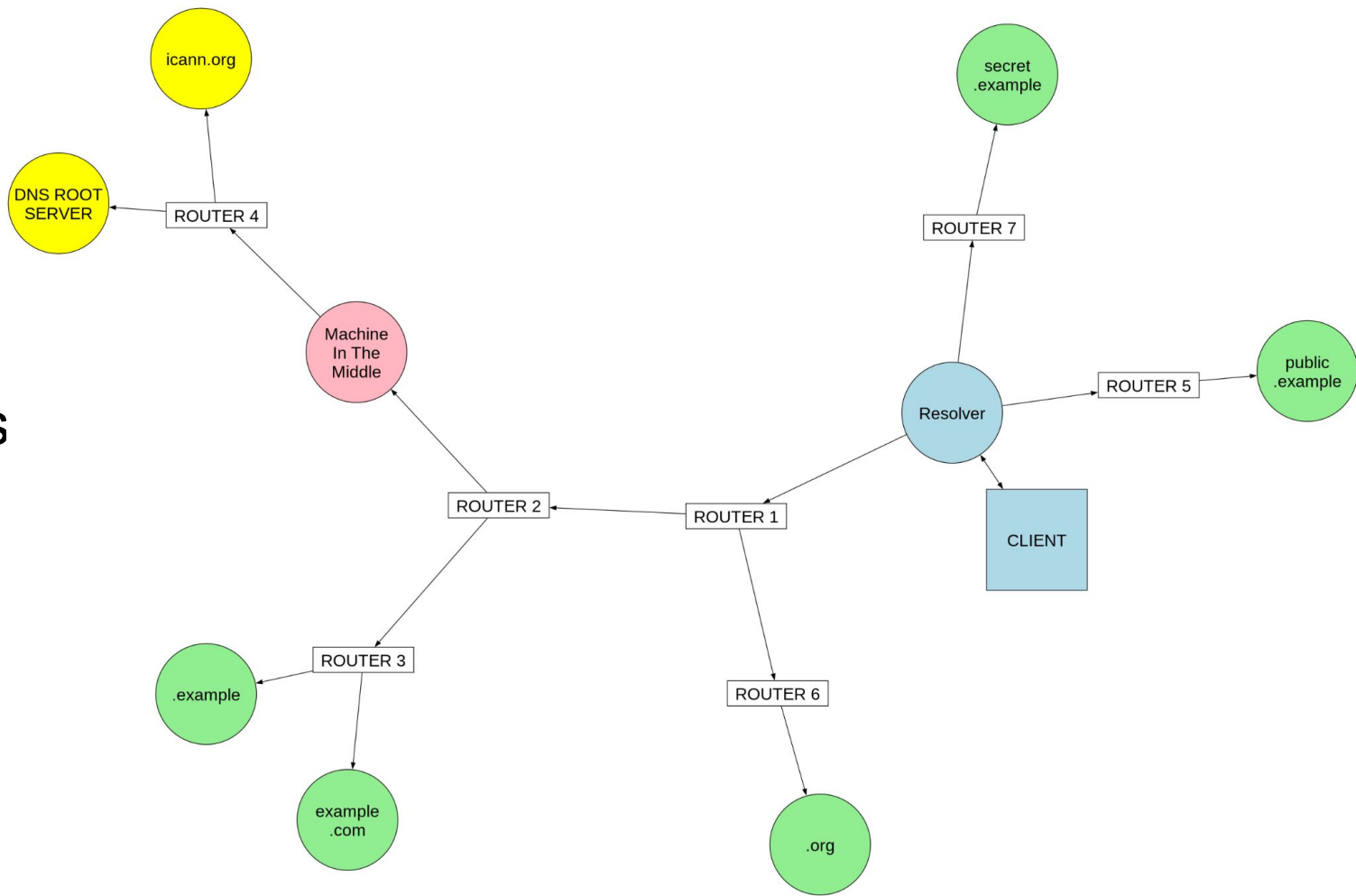
Imagine
I'm
In
Your
Network

*... and I'm not
a nice person*



I
Can
View
All Your
Resolver's
DNS
Requests

To These
Zones

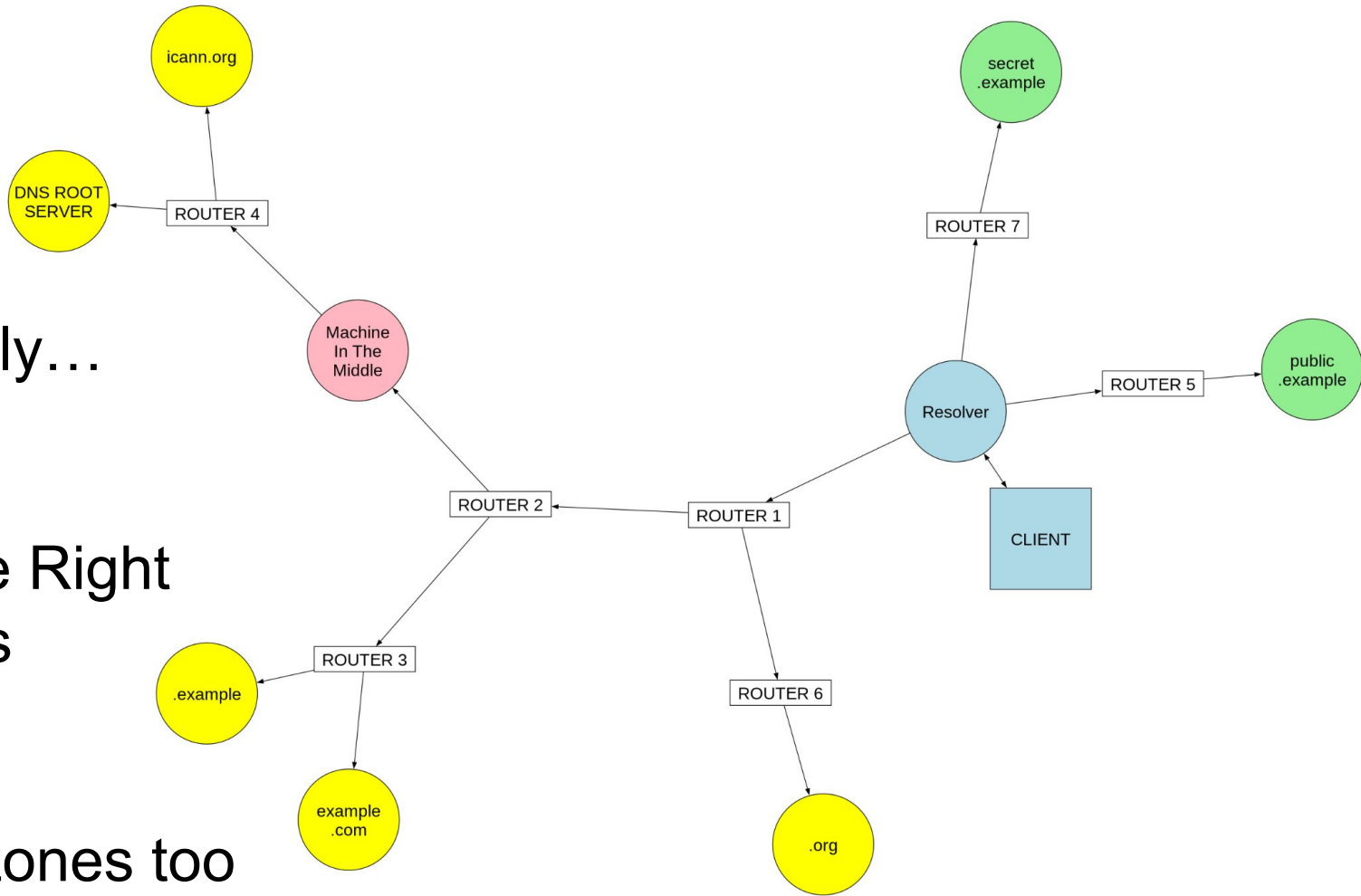


And

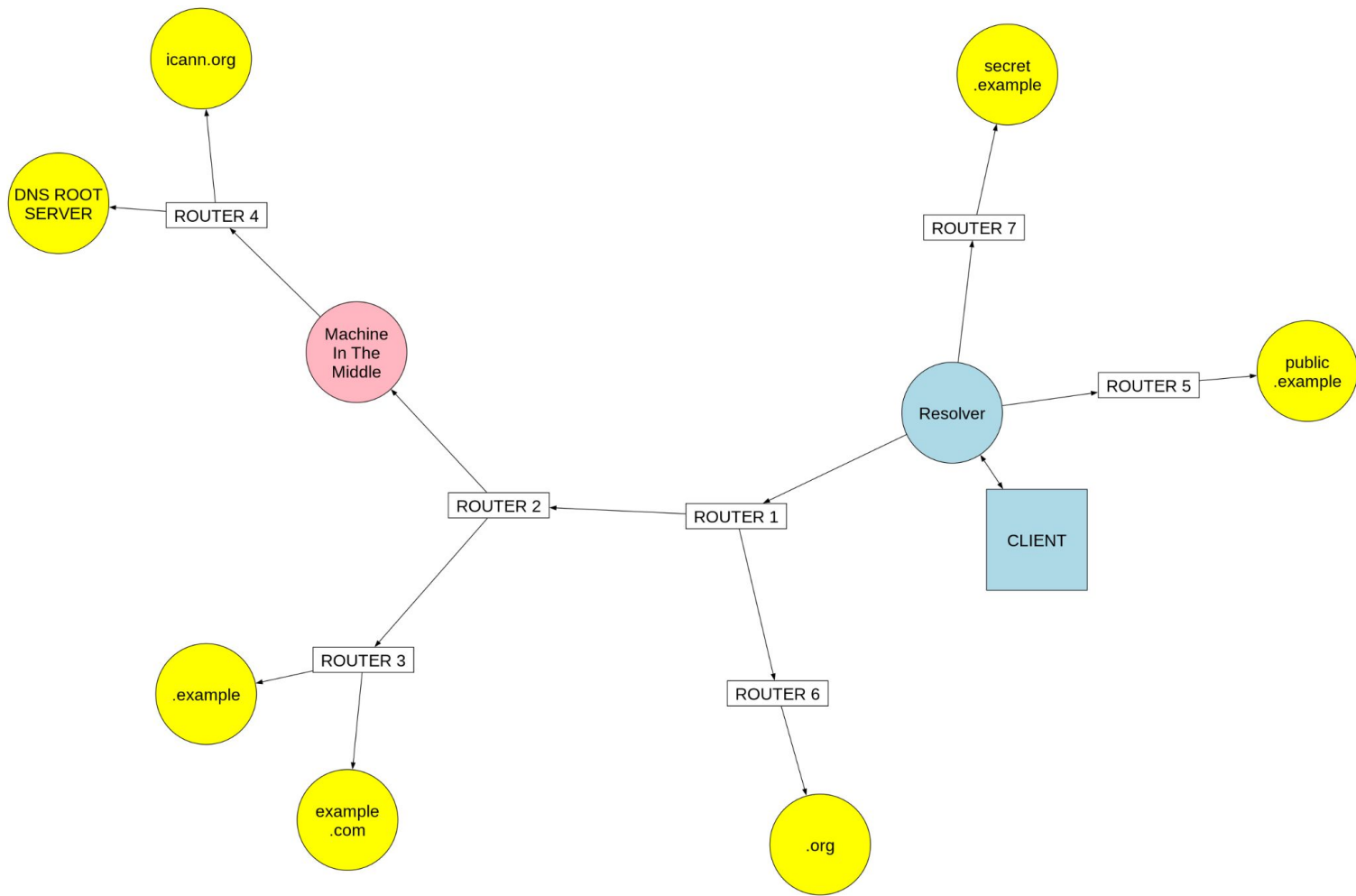
Surprisingly...

Under The Right
Conditions

To these zones too



And
Even
These



Overview

1. Core Issue: **Parent-centric resolvers trust unvalidated data**
2. Attack demonstration
3. So what? What's the worst that could happen...
4. Mitigations

Child-centric vs Parent-centric Resolvers

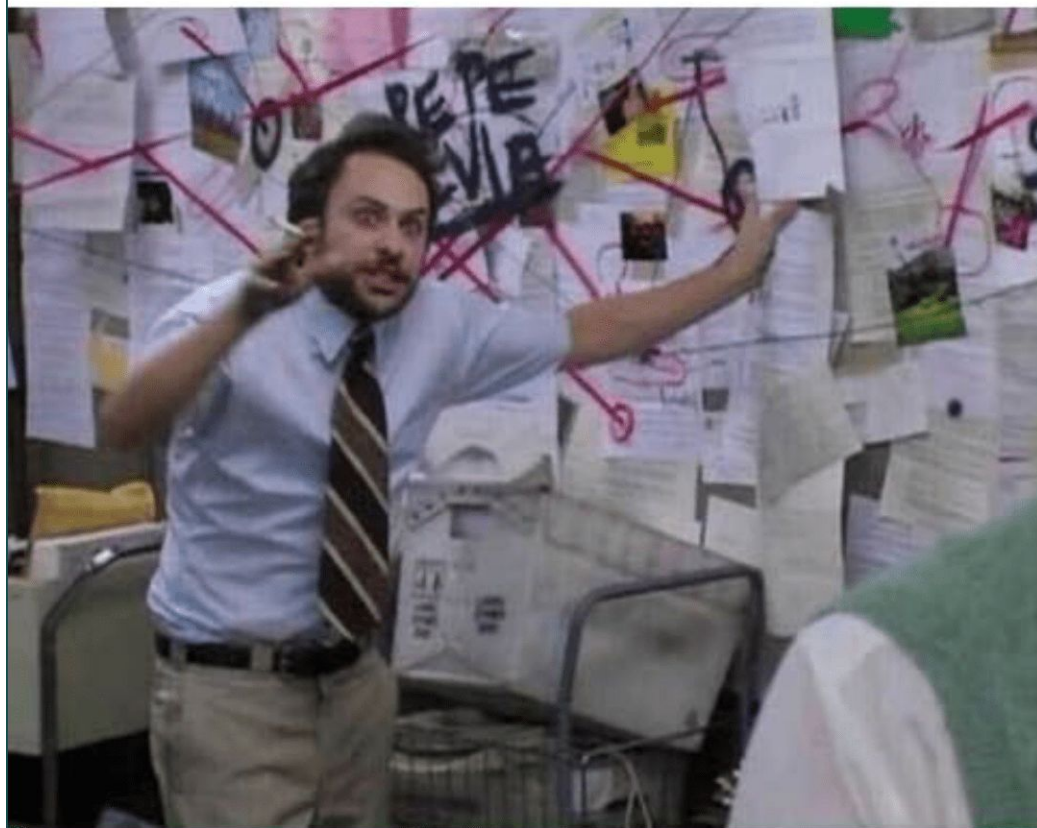
Child-centric resolvers

- Has only **one cache**: stores all data
- **DNSSEC validates** all DNS data
- **All** data is DNSSEC validated
- **Slower**

Parent-centric resolvers

- Has **two caches**: delegation and child
- **Trusts** the parent's delegation information
- **Only client responses** are validated
- **Faster** and more “broken DNS” resilient

"what's the worst thing that could happen?"

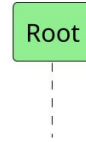
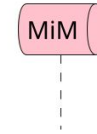
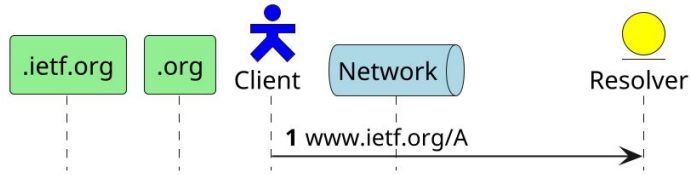




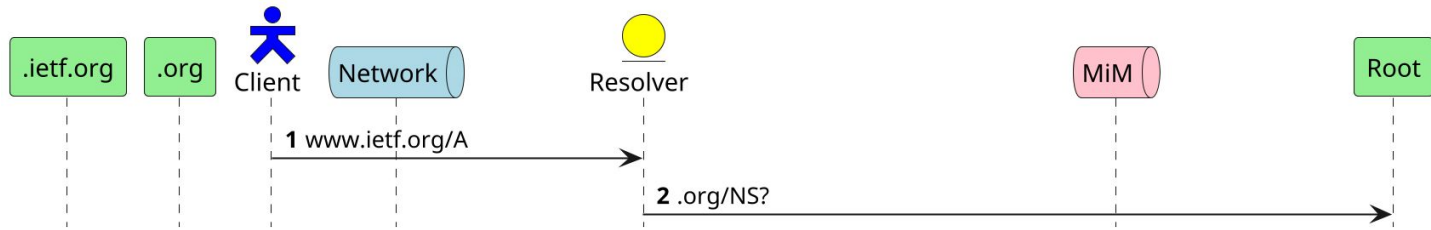
the ^{Disney} PARENT
TRAP

© 2019 Disney

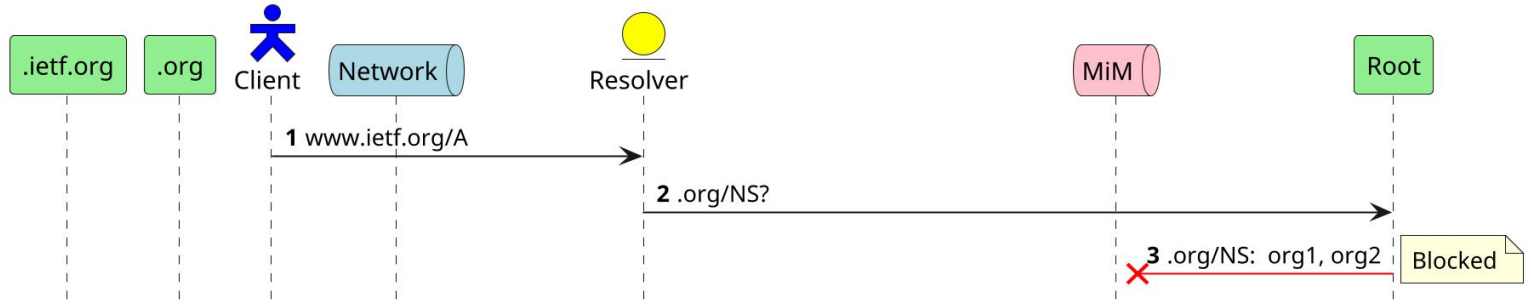
The Parent Trap



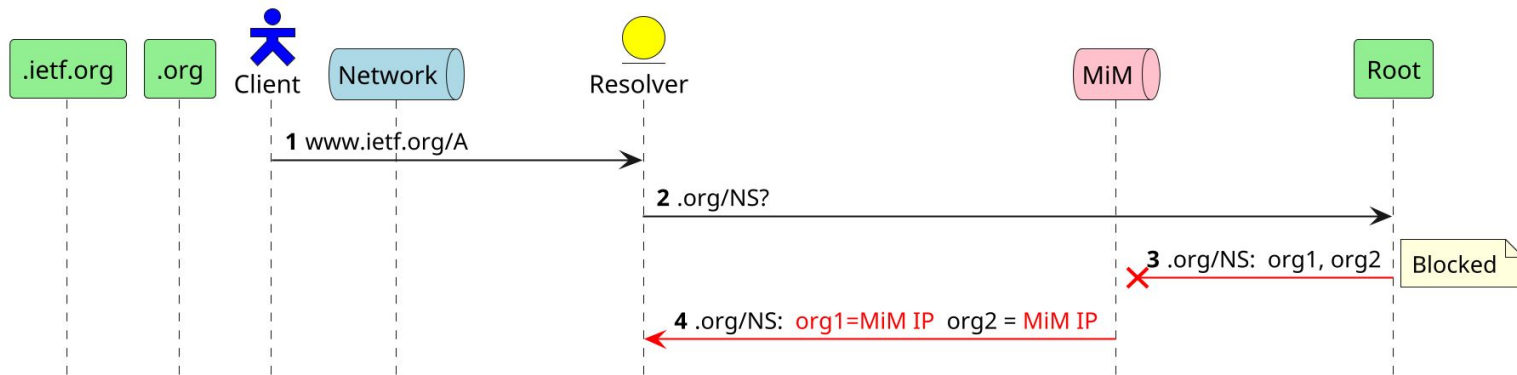
The Parent Trap



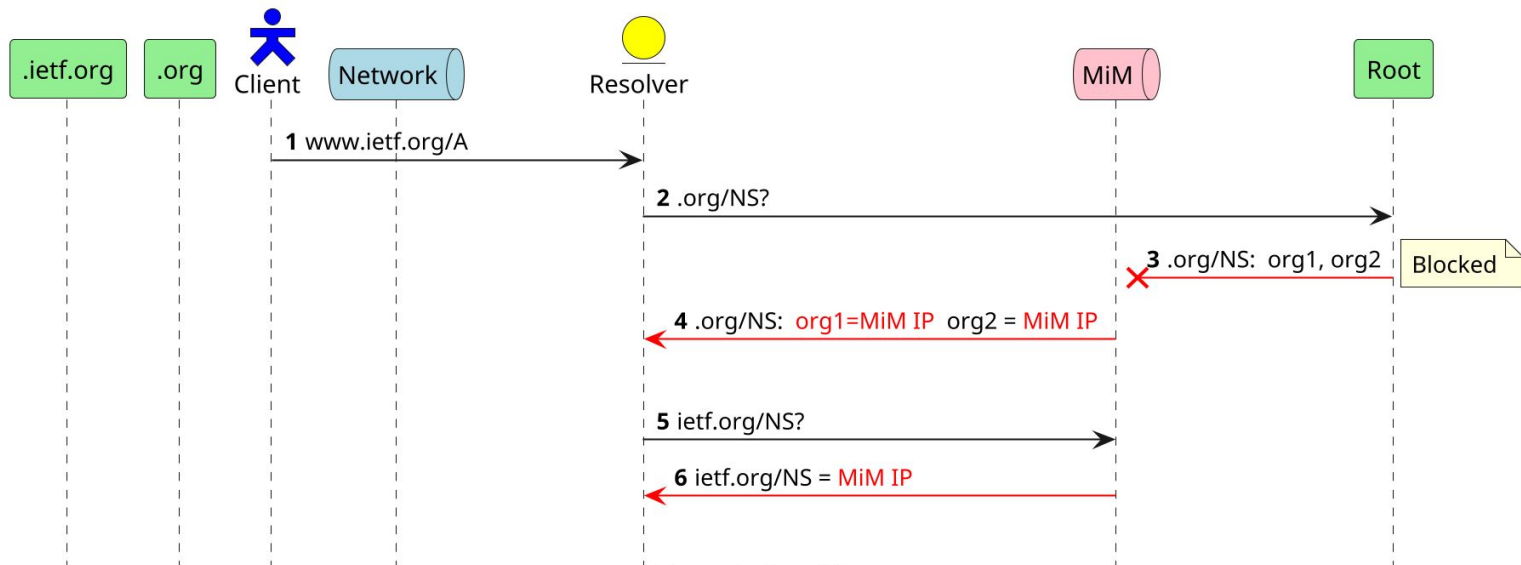
The Parent Trap



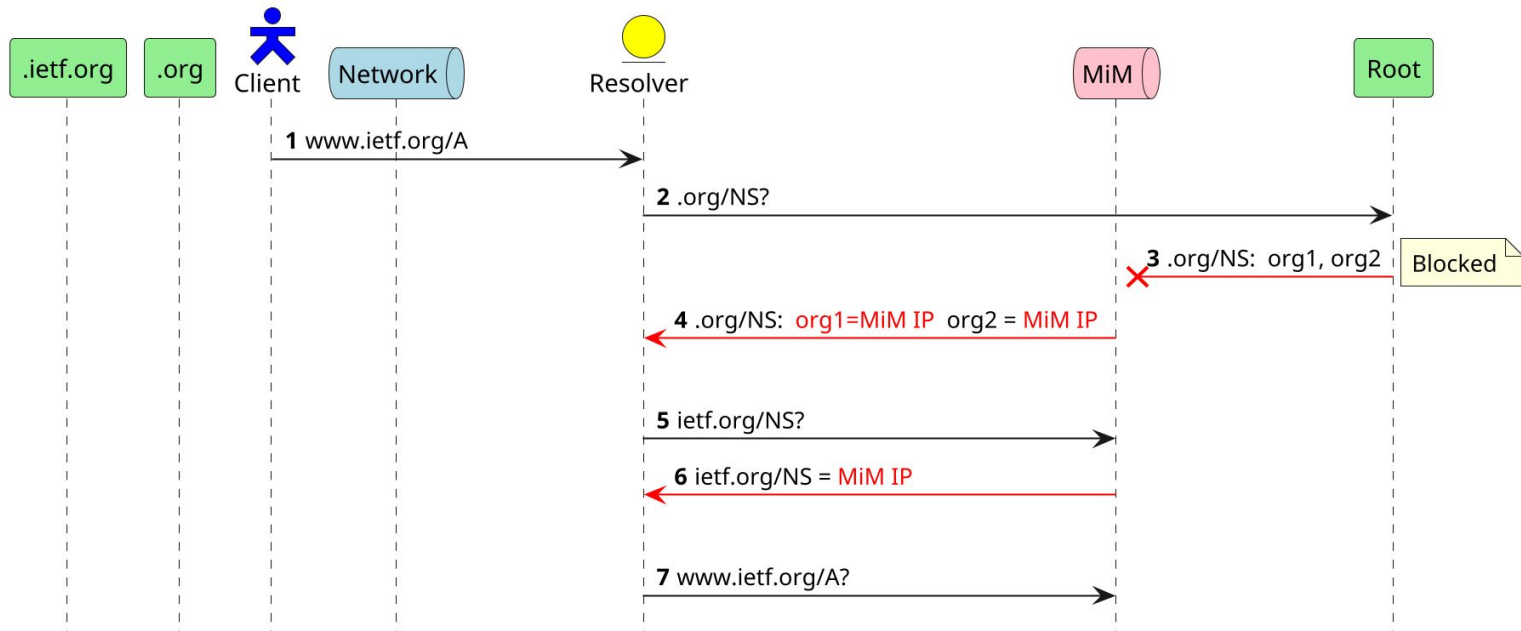
The Parent Trap



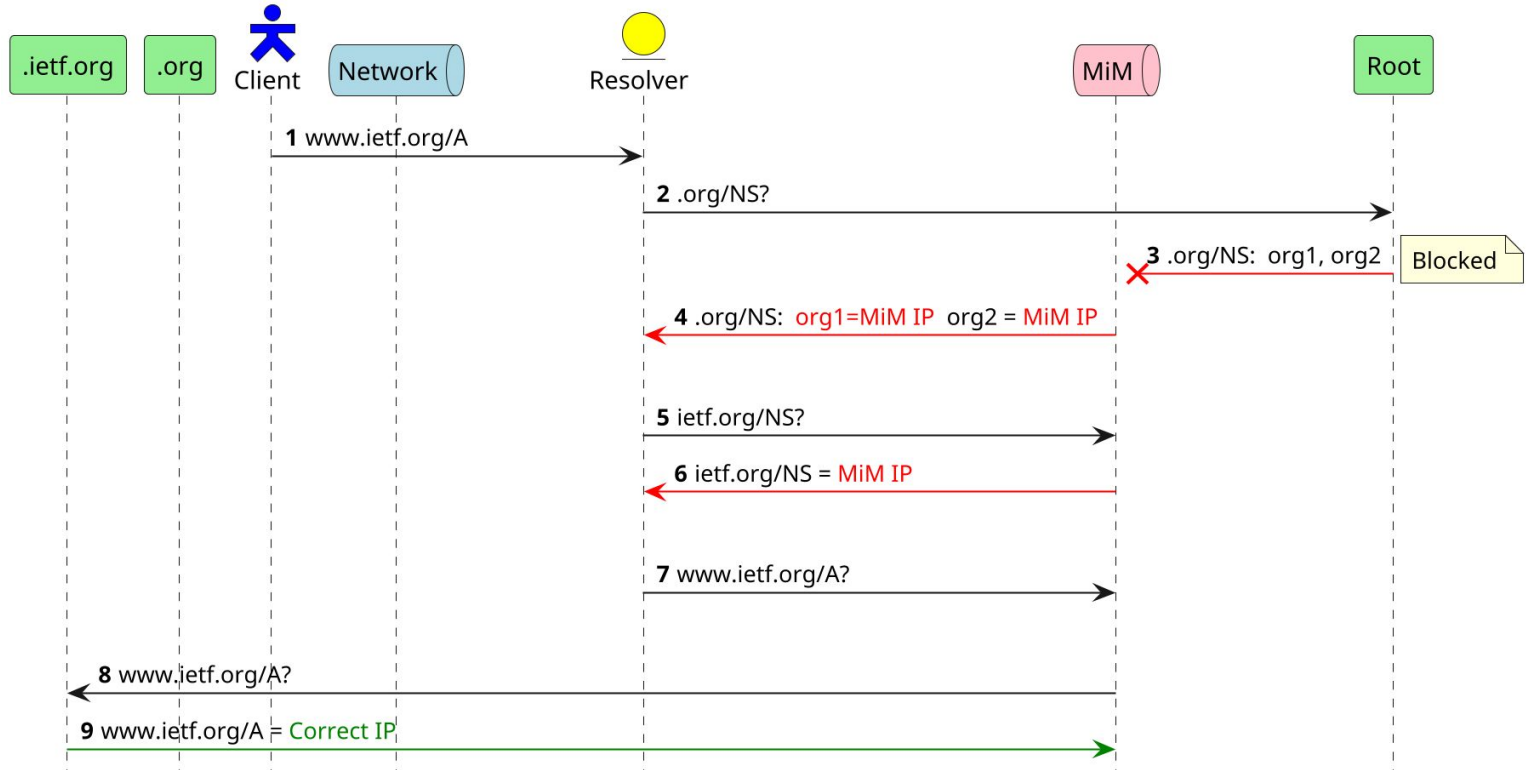
The Parent Trap



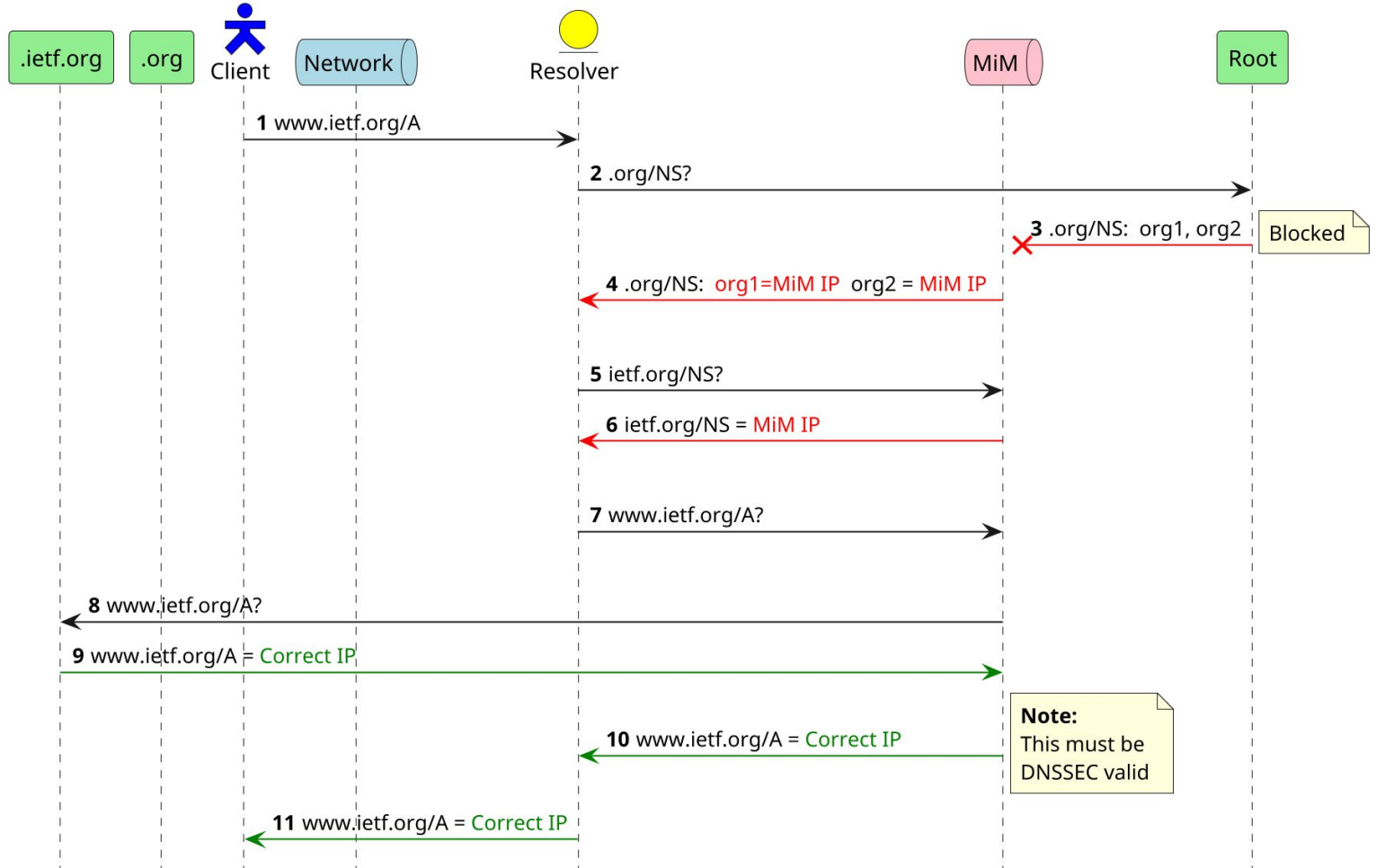
The Parent Trap



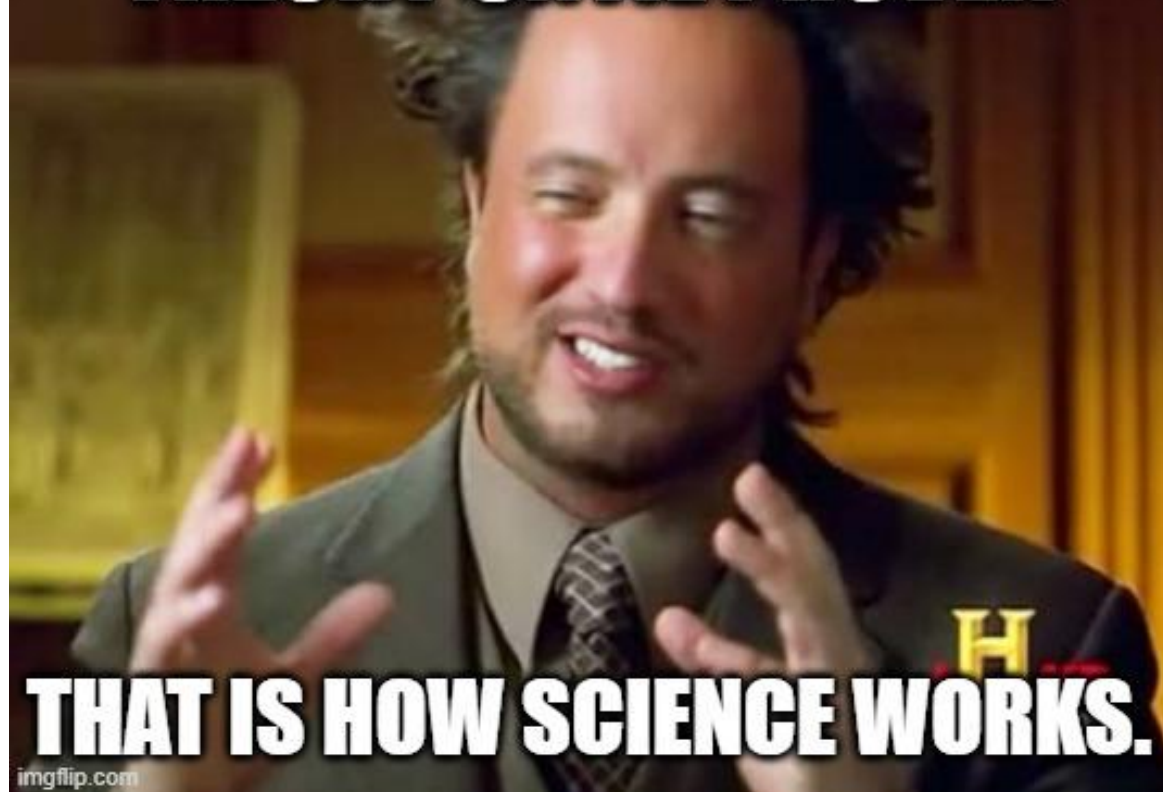
The Parent Trap



The Parent Trap

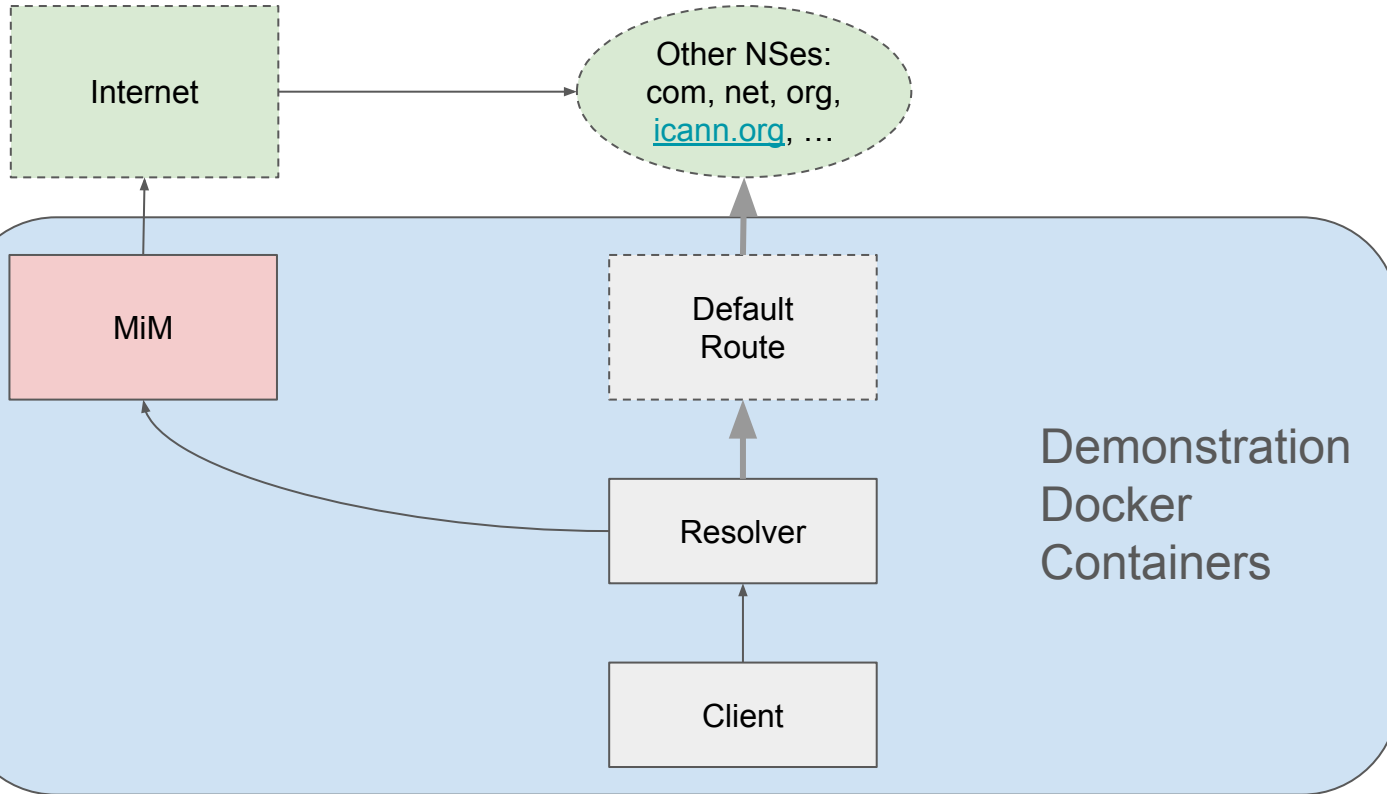


**IT IS JUST A
THEORY UNTIL PROVEN**

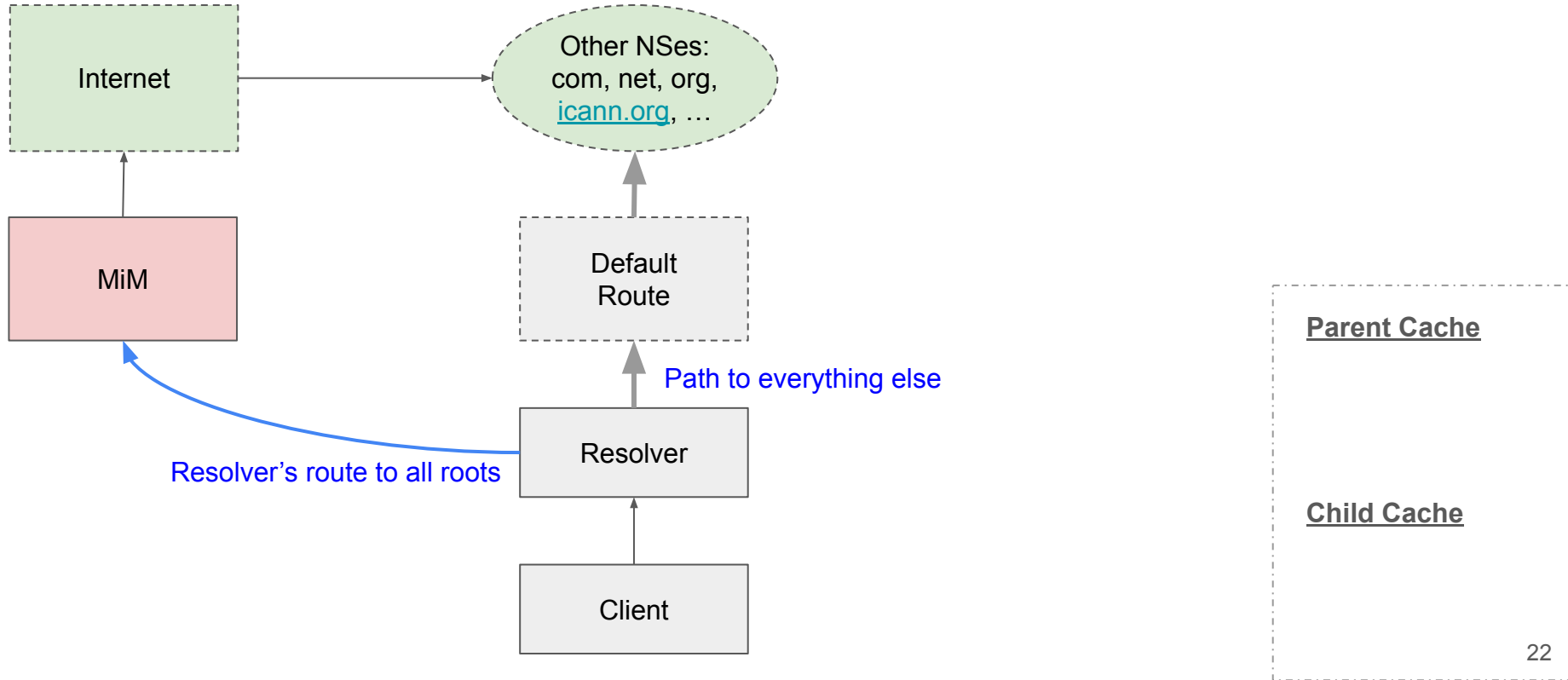


Demonstration Setup

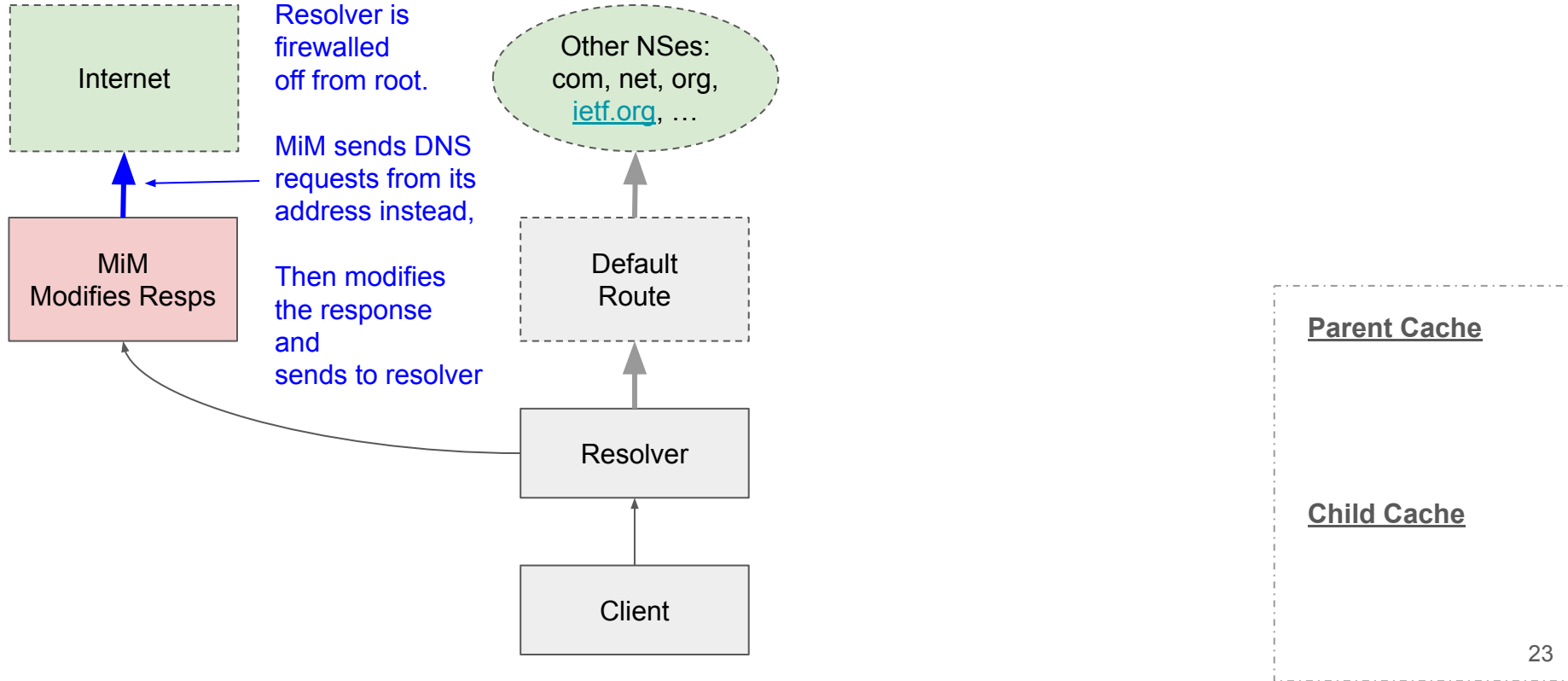
Test Network Design



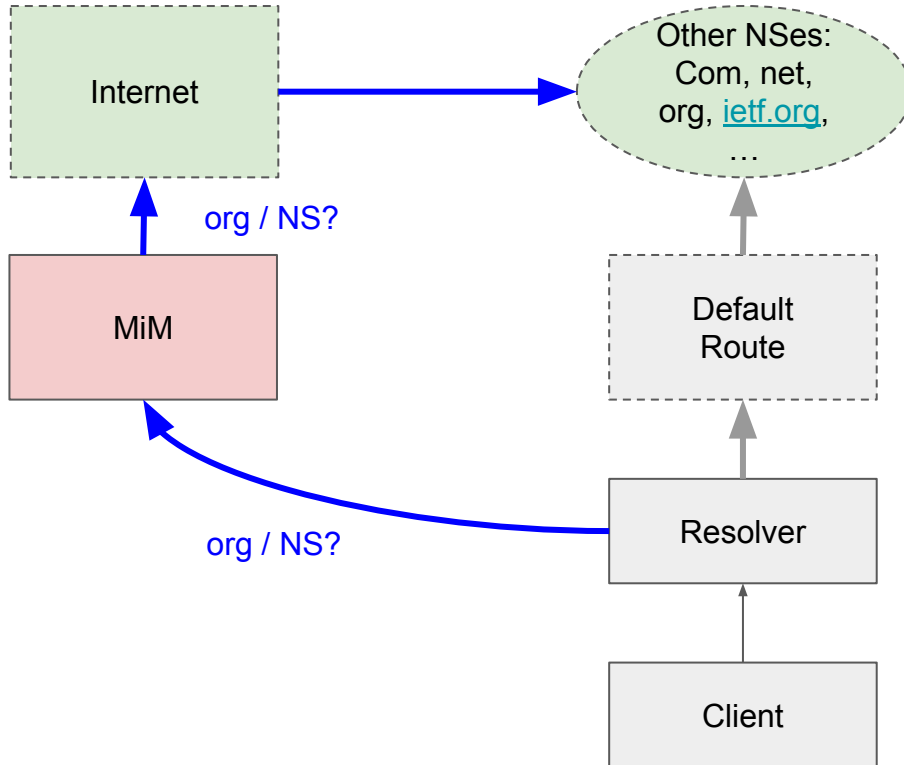
resolver to the root vs *resolver to everything else*



Implementation Notes:



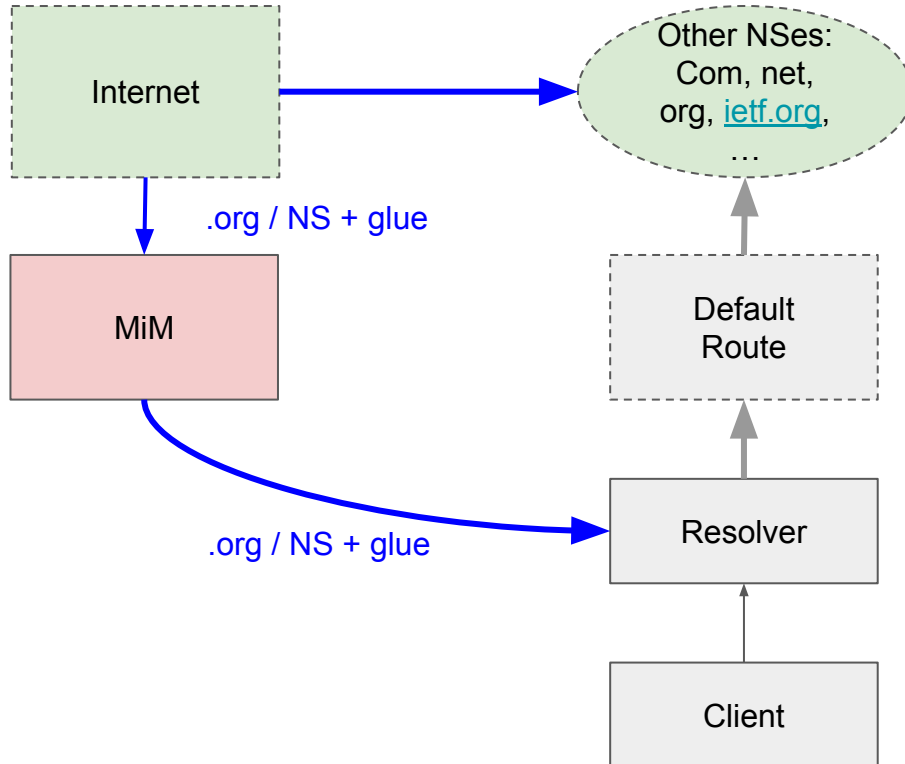
Demonstration Flow: resolver needs to know where .org is



Parent Cache

Child Cache

The DNS root will answer

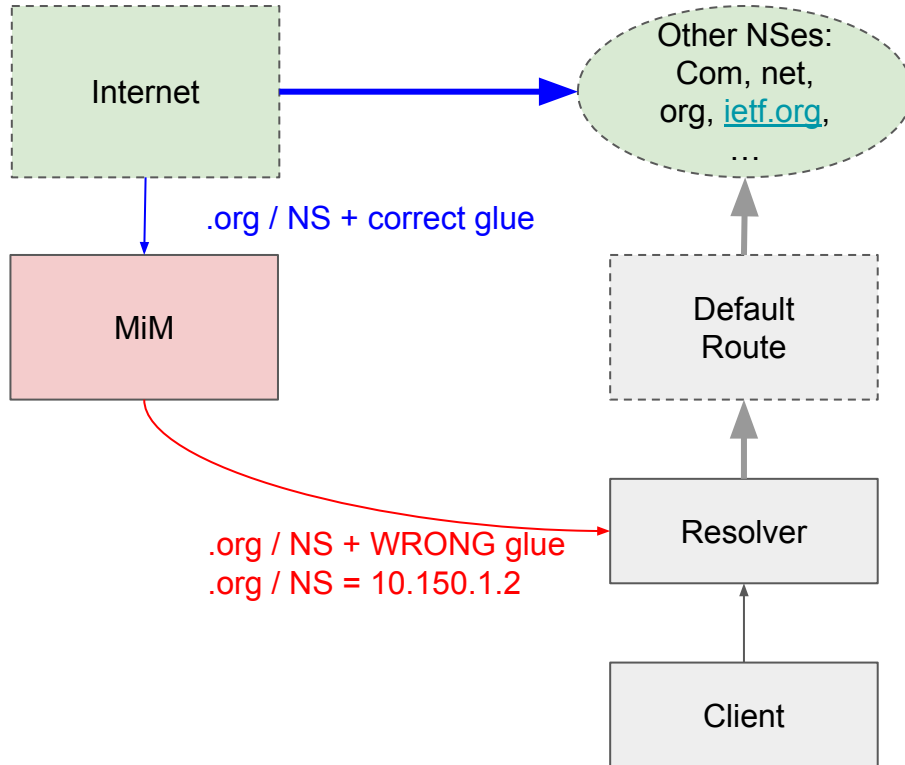


Parent Cache

.org NS Glue

Child Cache

A MiM can modify the results

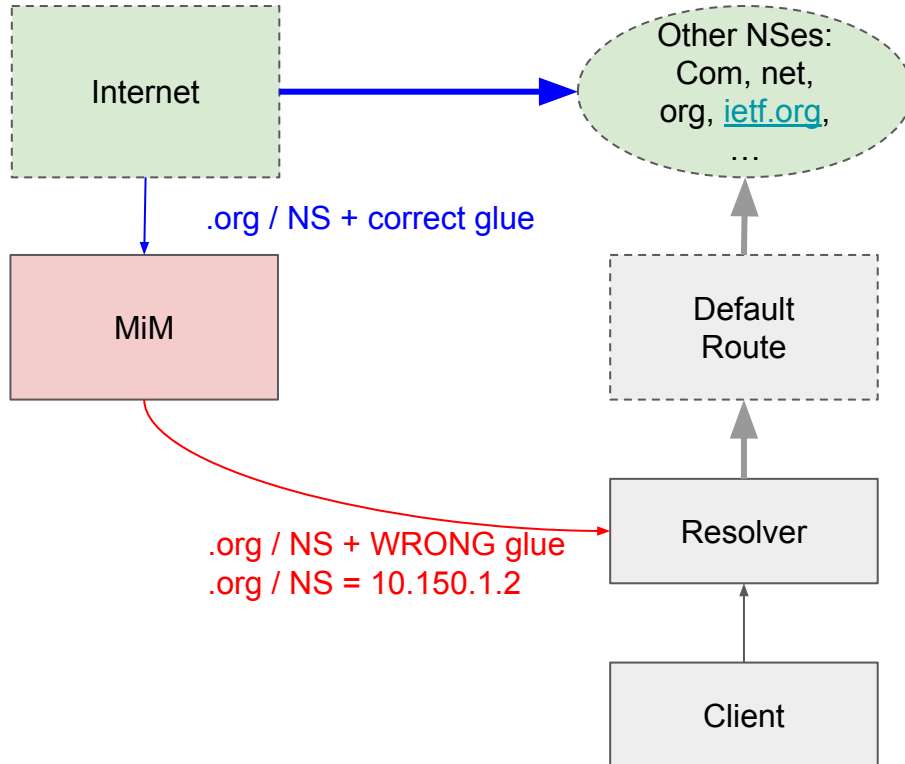


Parent Cache

.org **NS** **Glue**

Child Cache

A MiM can modify the results



Client-centric resolvers won't accept
Parent-centric resolvers blindly trust

Parent Cache

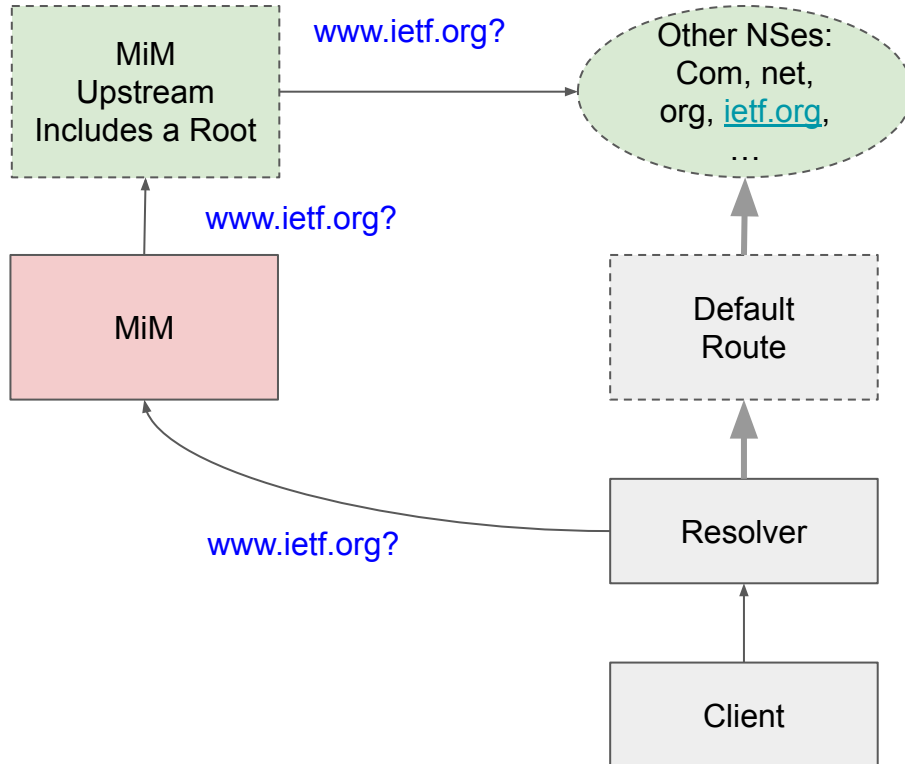
.org NS Glue

Child Cache

This continues...

- First: .org: where are you???
- Next: .ietf.org: where are you???

Demo – many steps later

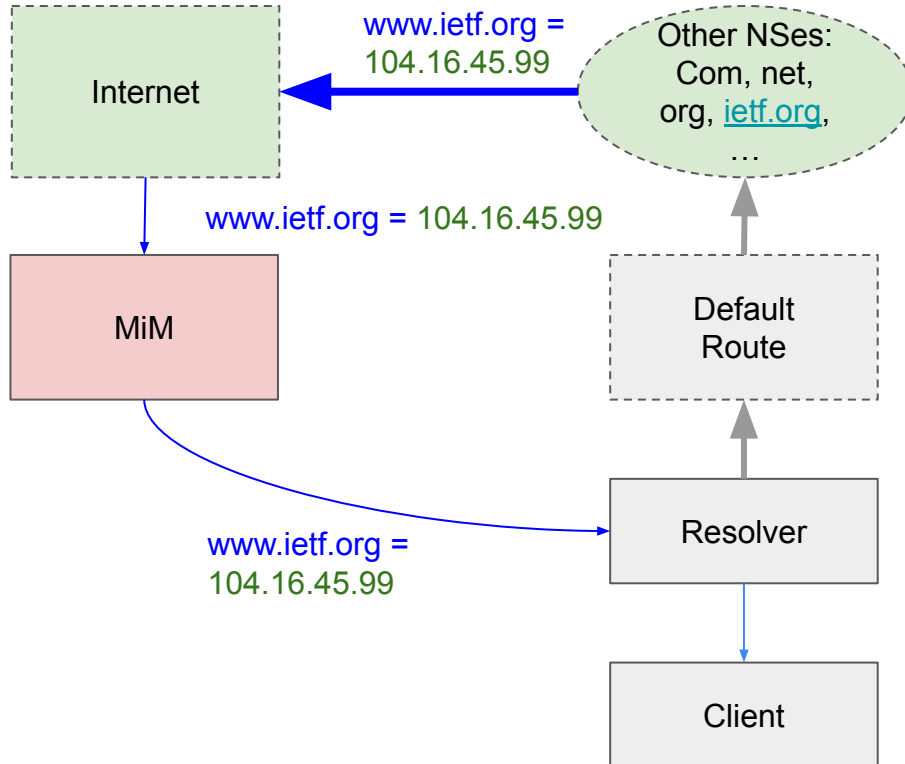


Parent Cache

<code>.org</code>	NS	Glue
<code>.ietf.org</code>	NS	Glue

Child Cache

Demo – many steps later



Parent Cache

.org	NS	Glue
.ietf.org	NS	Glue

Child Cache

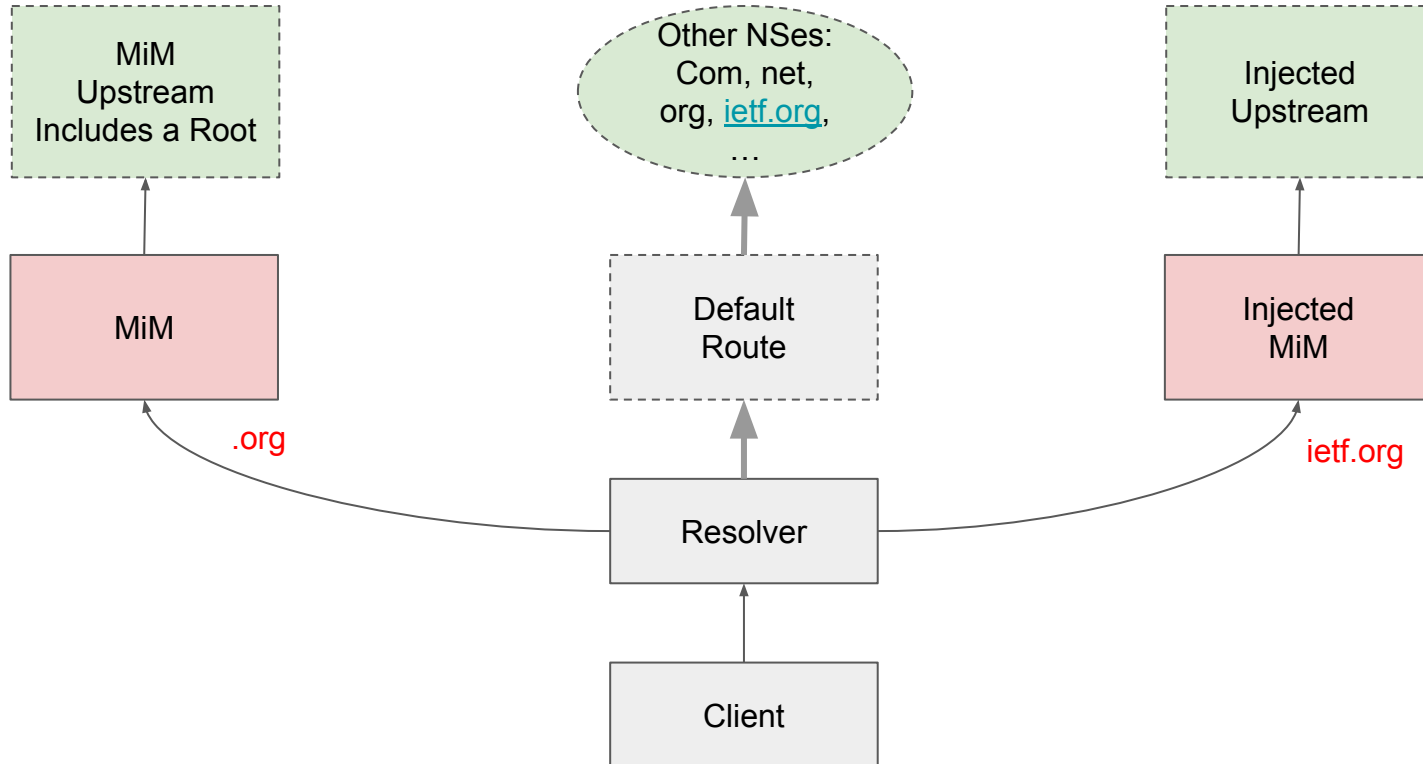
www.ietf.org	A
--------------	---

Demonstration 1: No MiM

Demonstration 2: MiM with NS rerouting

Demonstration 3: Changing Off-Path Records

Note: other MiMs can be injected too



So What?

Ramifications of MiM Injections

TL;DR: increased attack surface

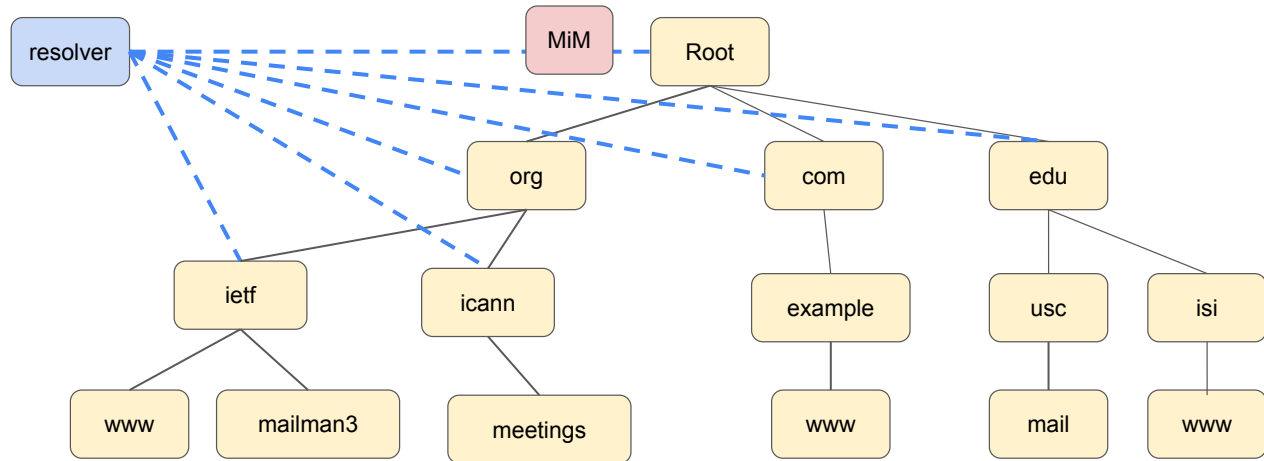
1. Visibility of DNS requests not normally out of view
2. Modification of DNS responses normally out of view

Let's look at someone else's DNS requests



MiM in only one path can provide **100% DNS visibility**

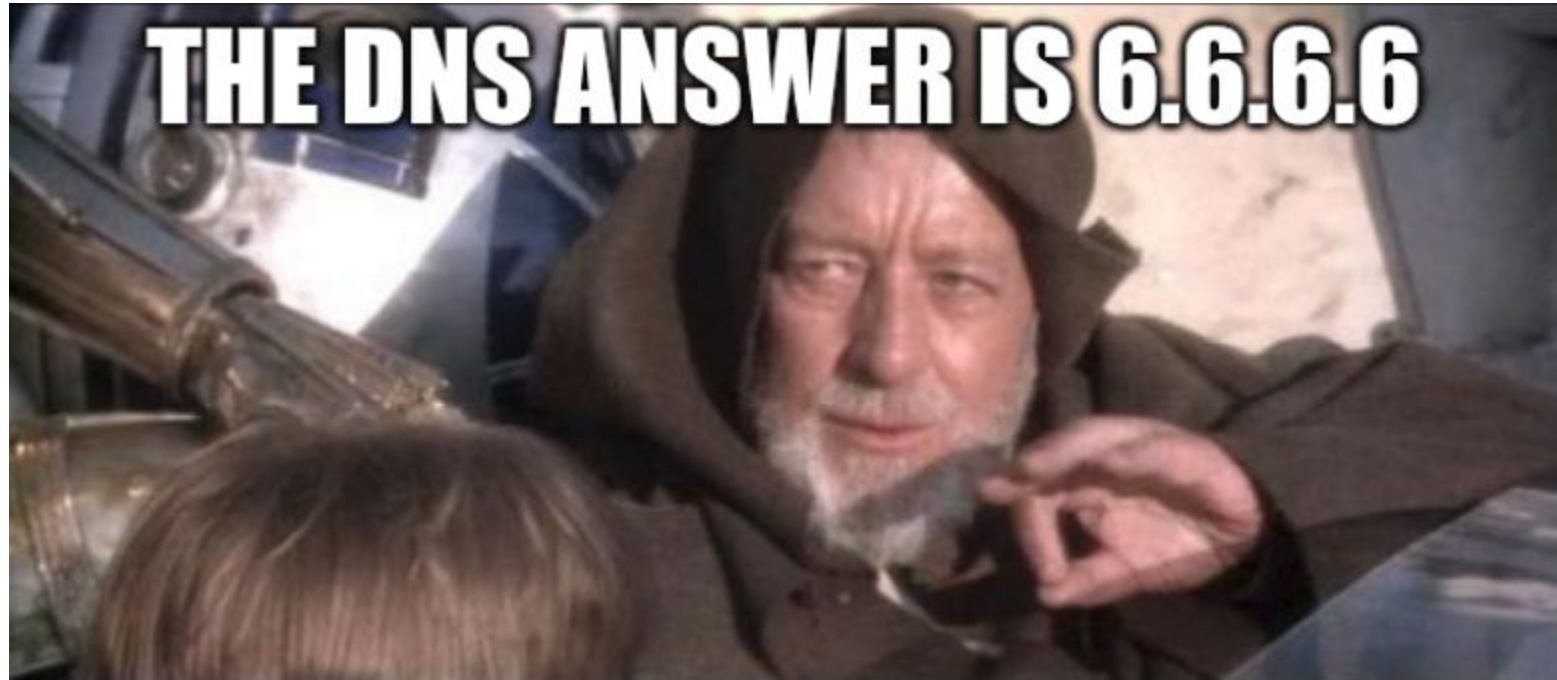
Even though **ALL** these zones are **DNSSEC signed**





Parent-centric ramifications of not checking DNSSEC

- MiM vantage points can be inserted into the DNS tree
 - Starting with any vantage point higher in the tree
- Difficult to detect in the client
- A MiM can:
 - Modify delegation information
 - A / AAAA glue *(this is hard to detect)*
 - NS names *(this is easier to detect)*
 - View all DNS transactions beyond their normal vantage point
 - Modify unsigned contents lower in the tree they normally can't see



Attacking normally off-path, non-DNSSEC protected zones

Our new expanded vantage points of non-DNSSEC protected zones can:

- Alter MX records
 - Re-routing mail through a attacker controlled relay
- Alter DKIM/DMARC
 - Spoofing mail to the attack target from other DNS services
- Alter DNS HTTPS records
 - Downgrading http version
 - ECH manipulation
 - Adding or changing ipv4/v6 glue
- ...

FAQ

Does client-centric resolution solve this?

Yes: because the child verifies delegation information (*caveat: some give up*)

Can the end-user / client detect this?

Not easily: the delegation cache is different than the cache for the client

Can DNSSEC prevent this?

No: because parent-side NS/glue are never checked

But: Data to clients is validated: DS -> DNSKEY -> ... -> DNSKEY -> RR

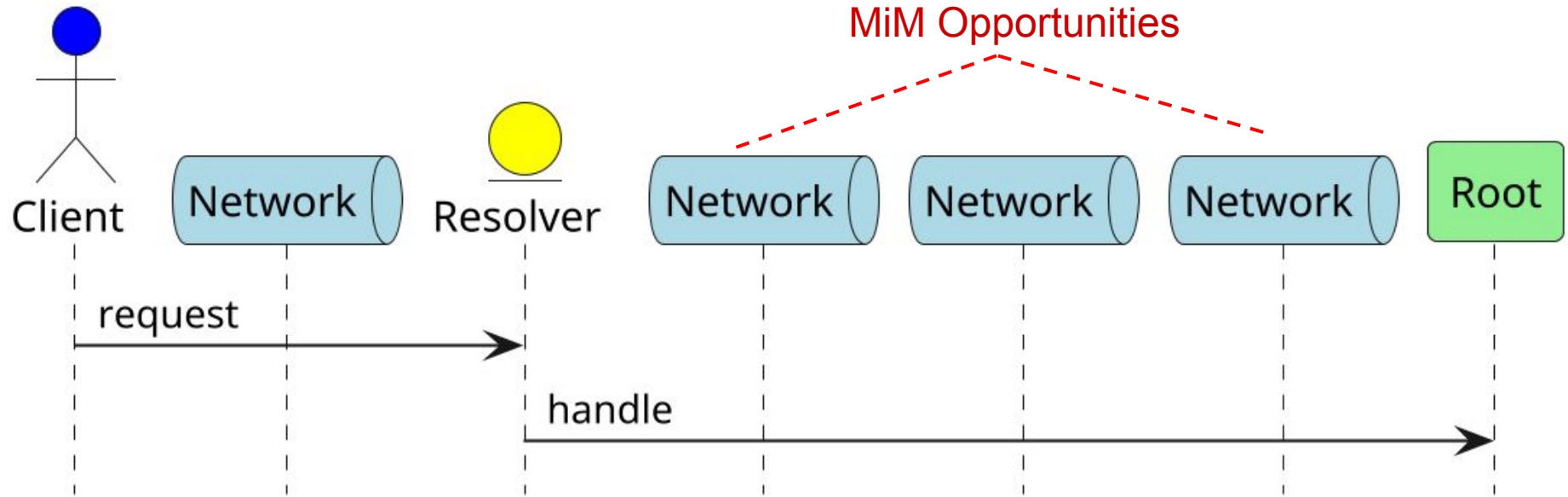
What about DELEG?

DELEG delegations are secure: as they'll be signed by the parent

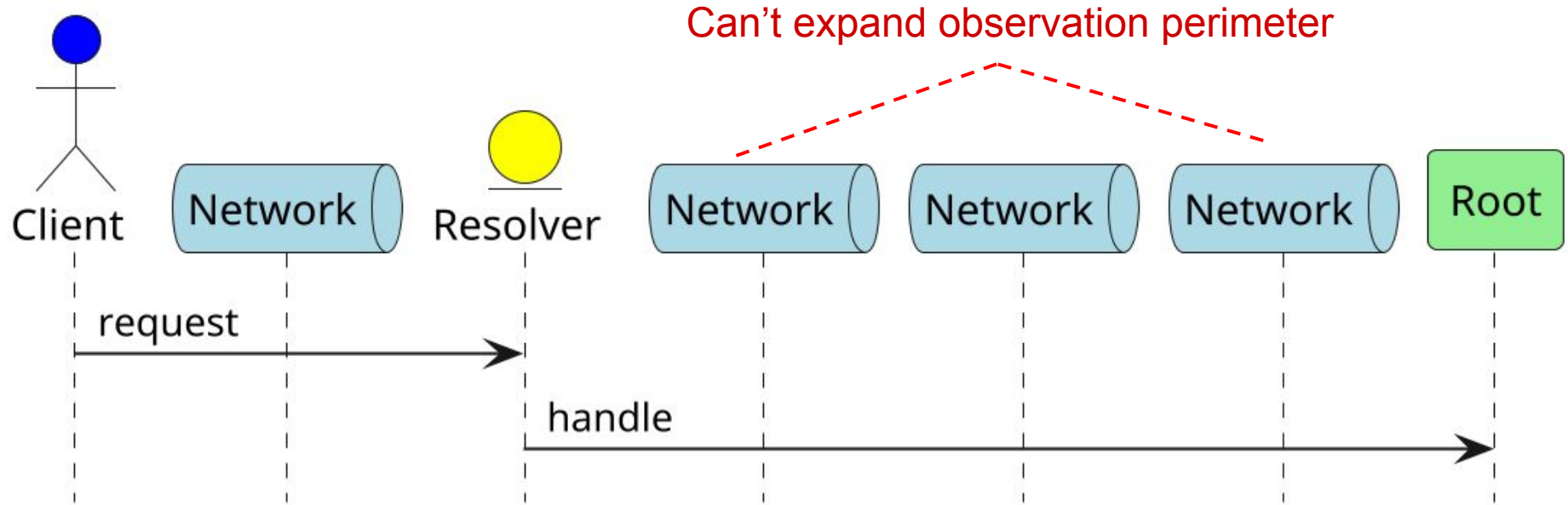
Mitigations

- Use DNSSEC validating, child-centric resolvers
 - See draft-ietf-dnsop-ns-revalidation
- Deploy parent-centric resolvers close to authoritative server infrastructure
 - aka peer widely: minimize MiM opportunities
- Deploy DoX between your clients and your resolver
- Look forward to DELEG records, which are parent-signed

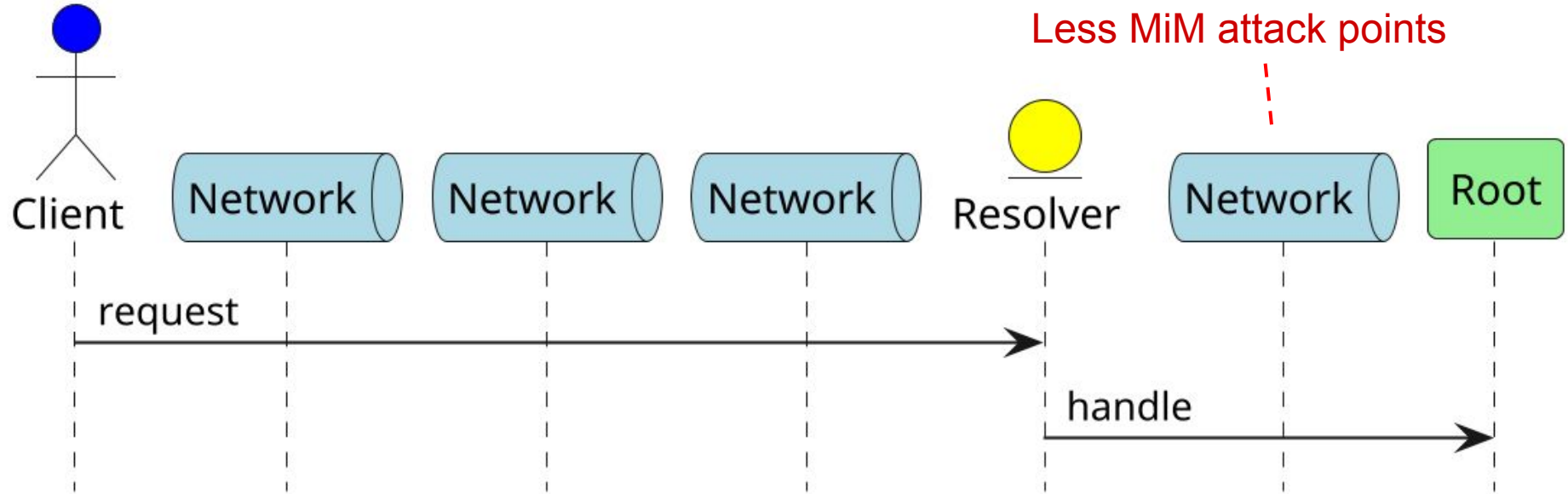
Resolvers are typically placed near clients

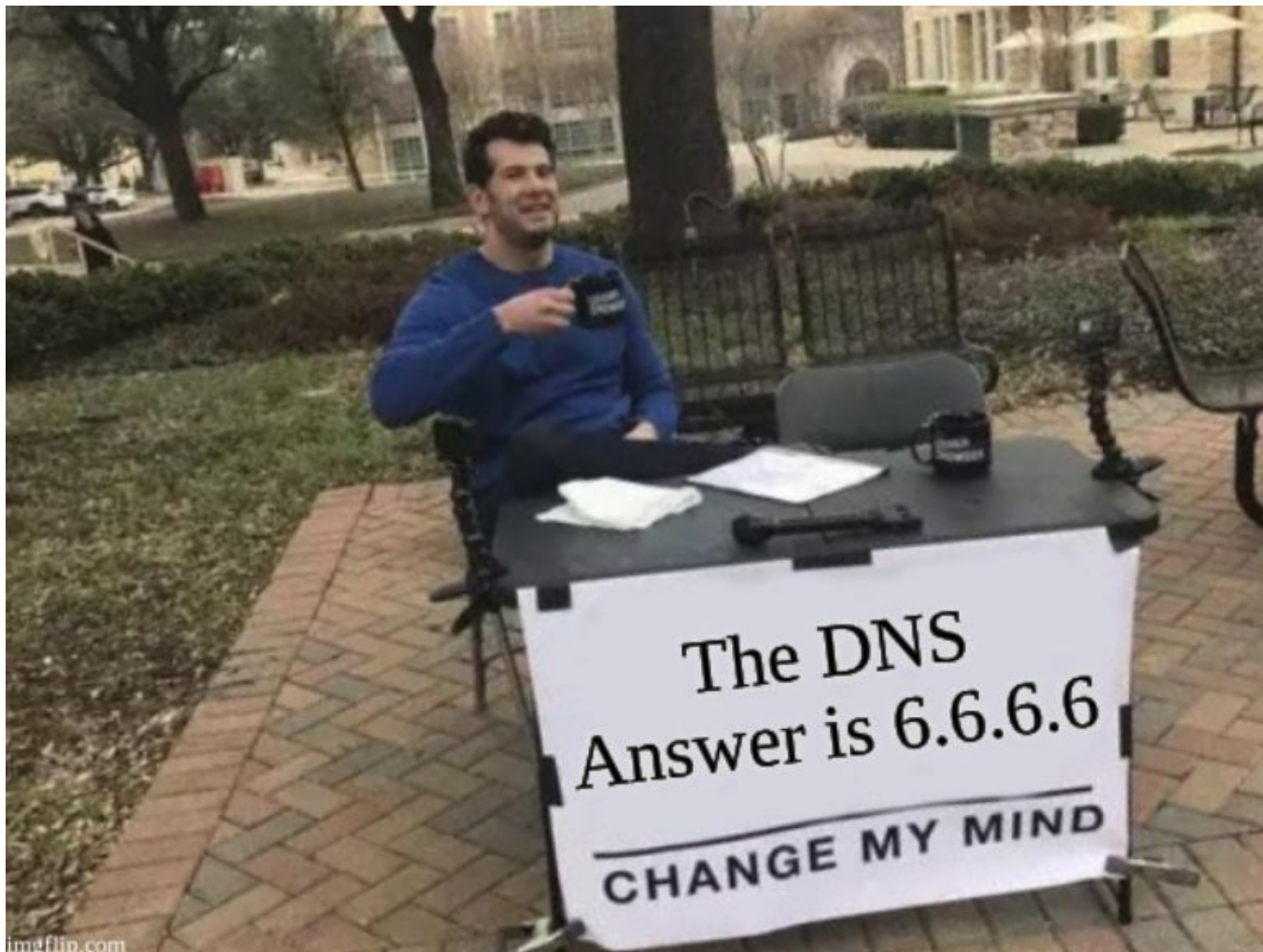


Option 1: deploy client-centric, validating resolvers



Option 2: place resolvers closer to authoritative servers

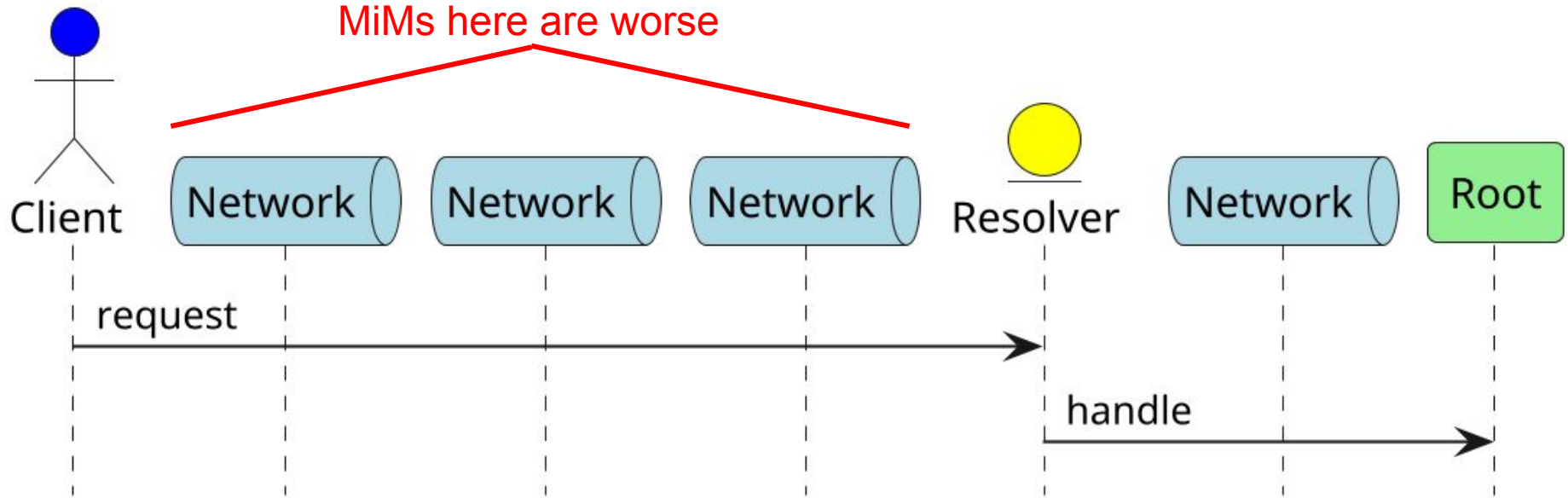




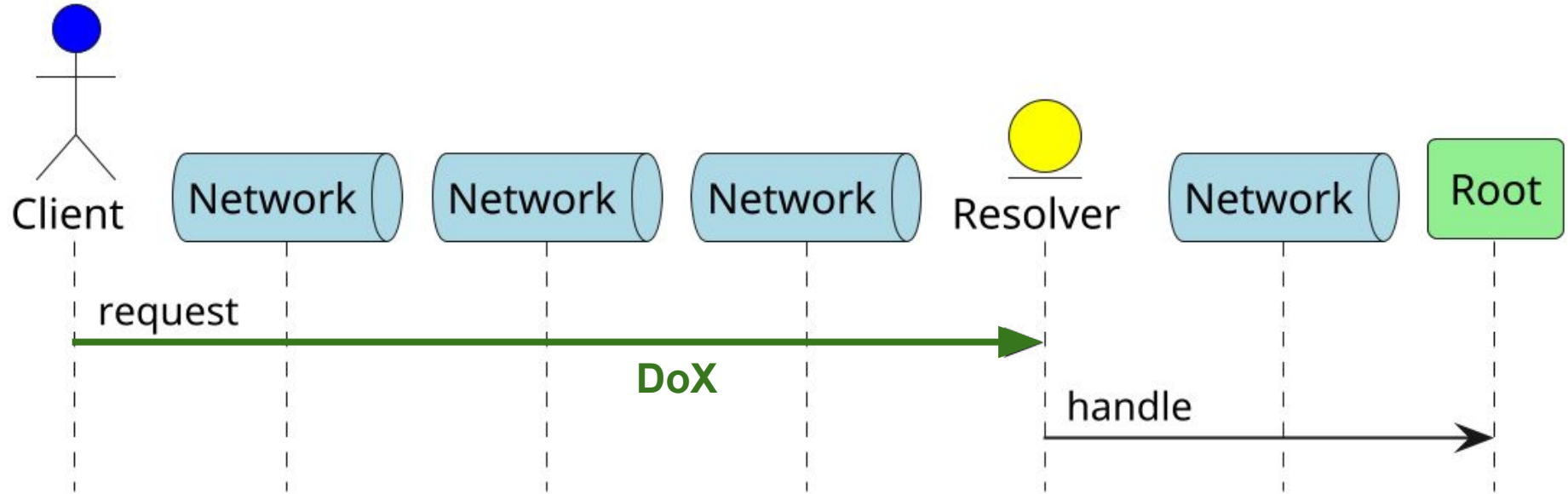
Thanks for the
new MiM spot

You've only moved
the problem

You've just moved the problem



Solution: deploy DoT here

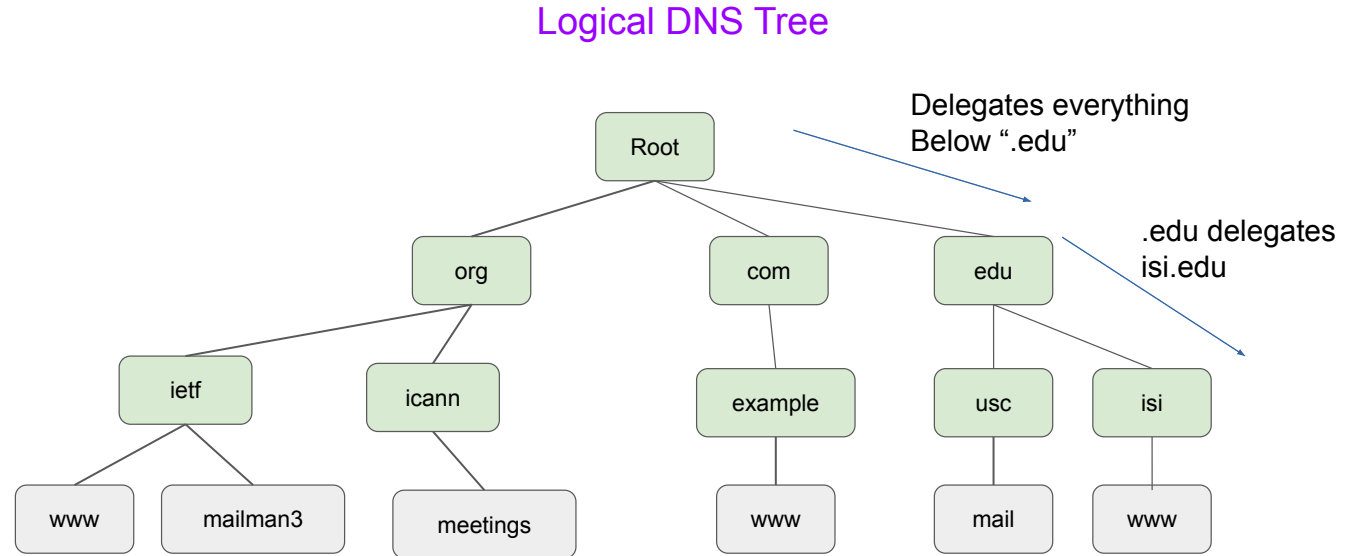


Takeaways

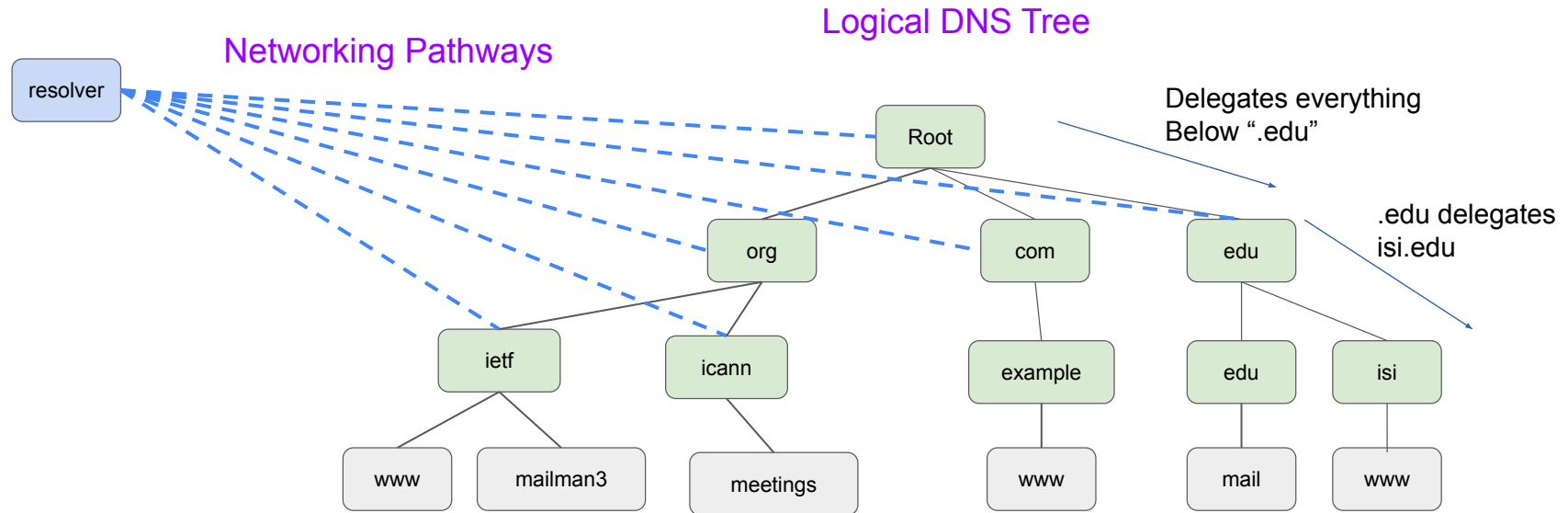
- Parent-centric, DNSSEC validating resolvers:
 - Can be subject to expanded view **injection attacks** for all zones
 - Can be subject to **expanded modification attacks** for unsigned zones
 - Other work has shown them to be faster and more robust, however
- **Mitigations:** *what are you going to prioritize?*
 - **Choose** resolver software **carefully**
 - Only deploy parent-centric resolvers with **wide-peering**
- DELEG can't come fast enough

Backup Extra Complex Slides

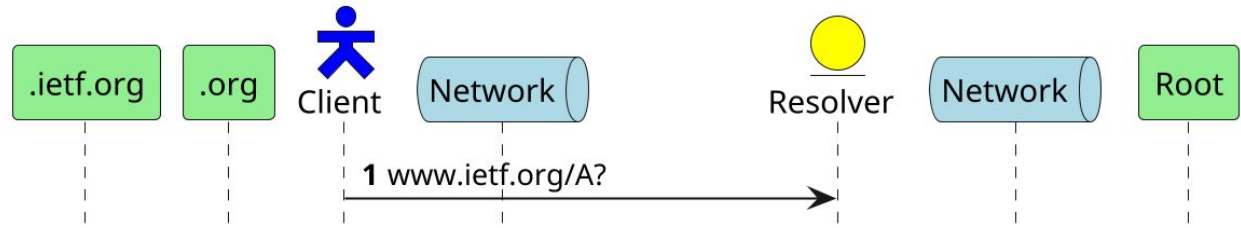
DNS Resolvers Communicate With Everything



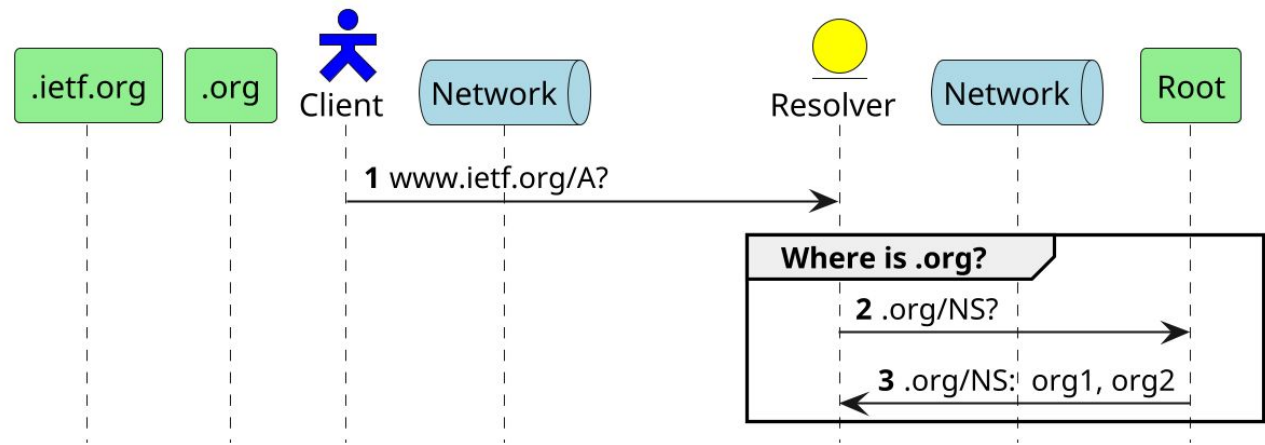
DNS Resolvers Communicate With Everything



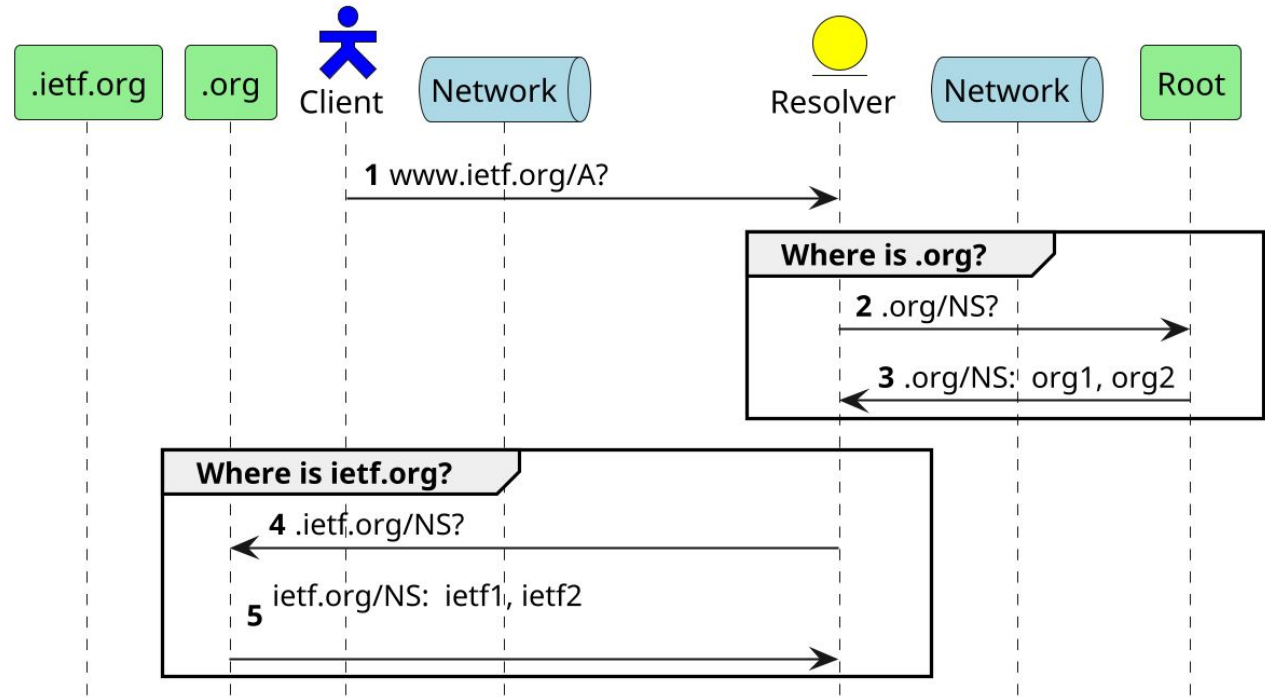
DNS Resolution Steps



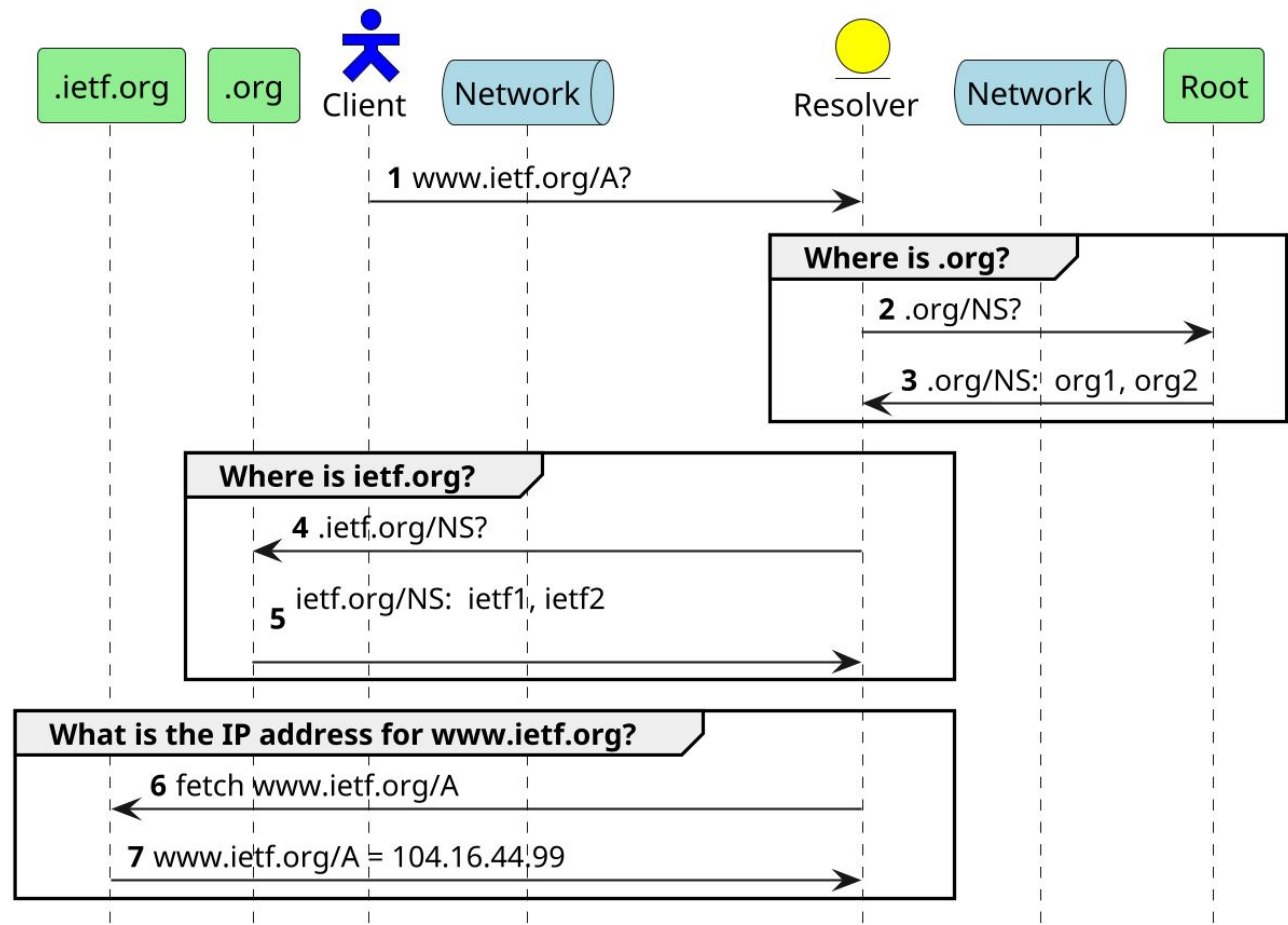
DNS Resolution Steps



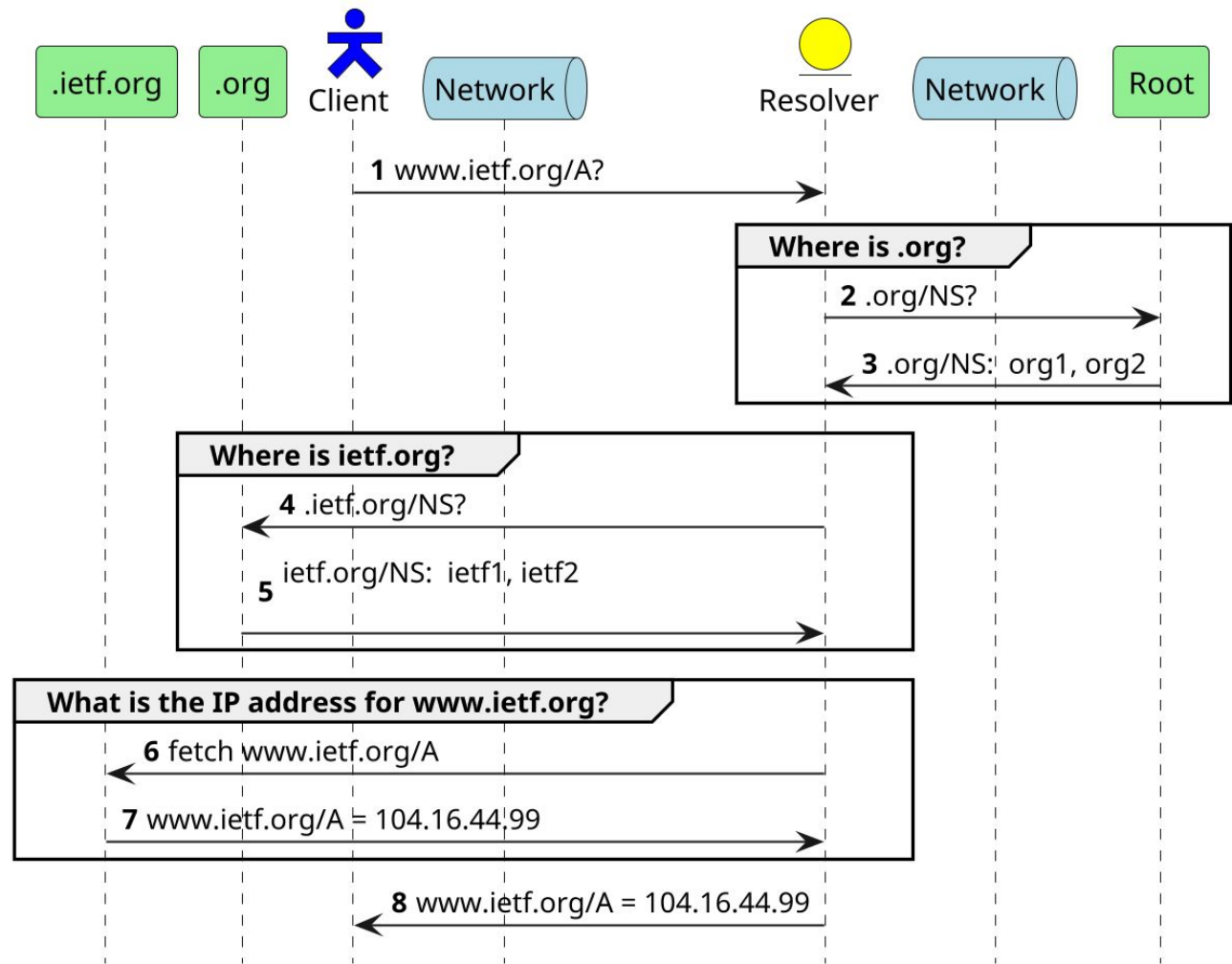
DNS Resolution Steps



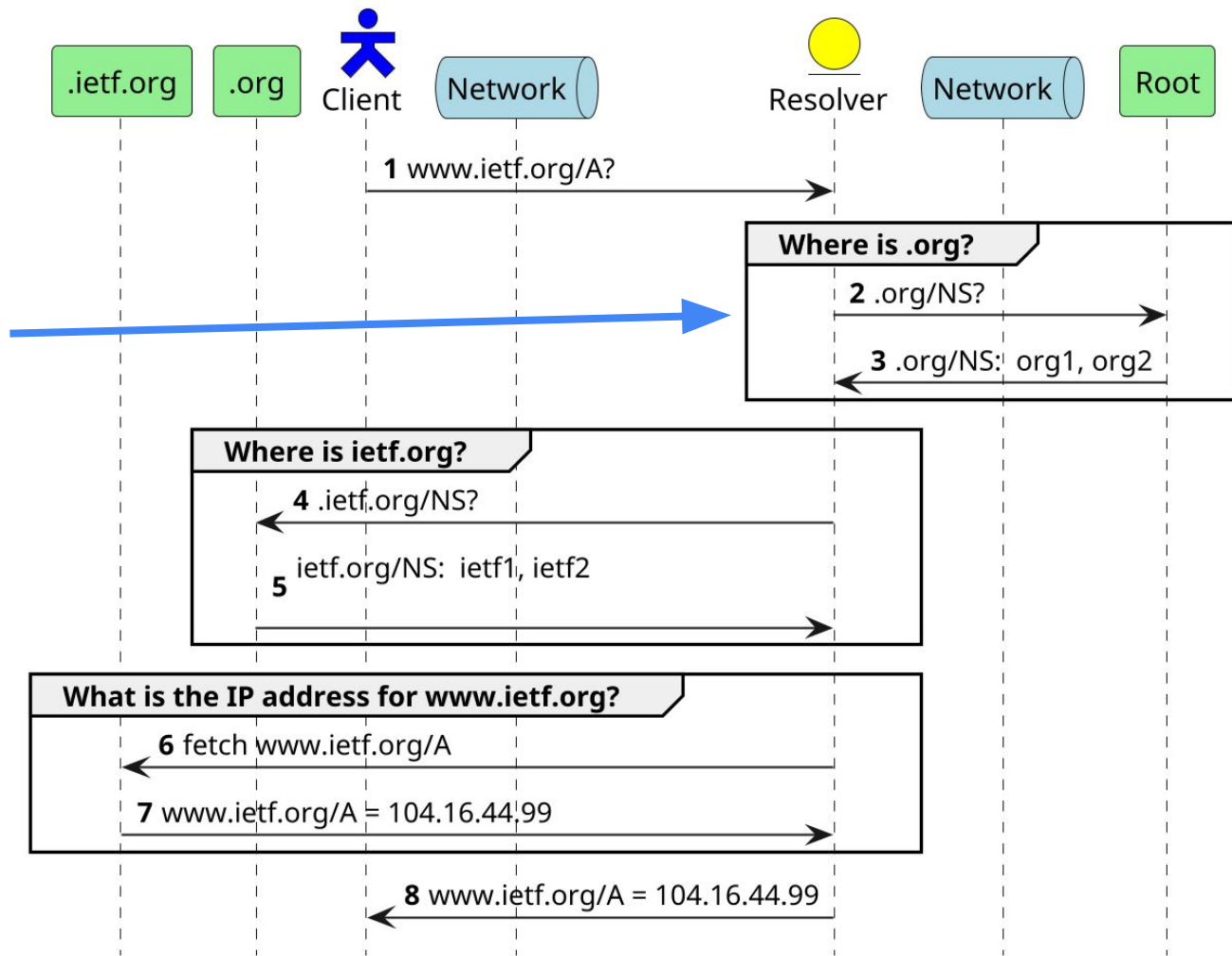
DNS Resolution Steps



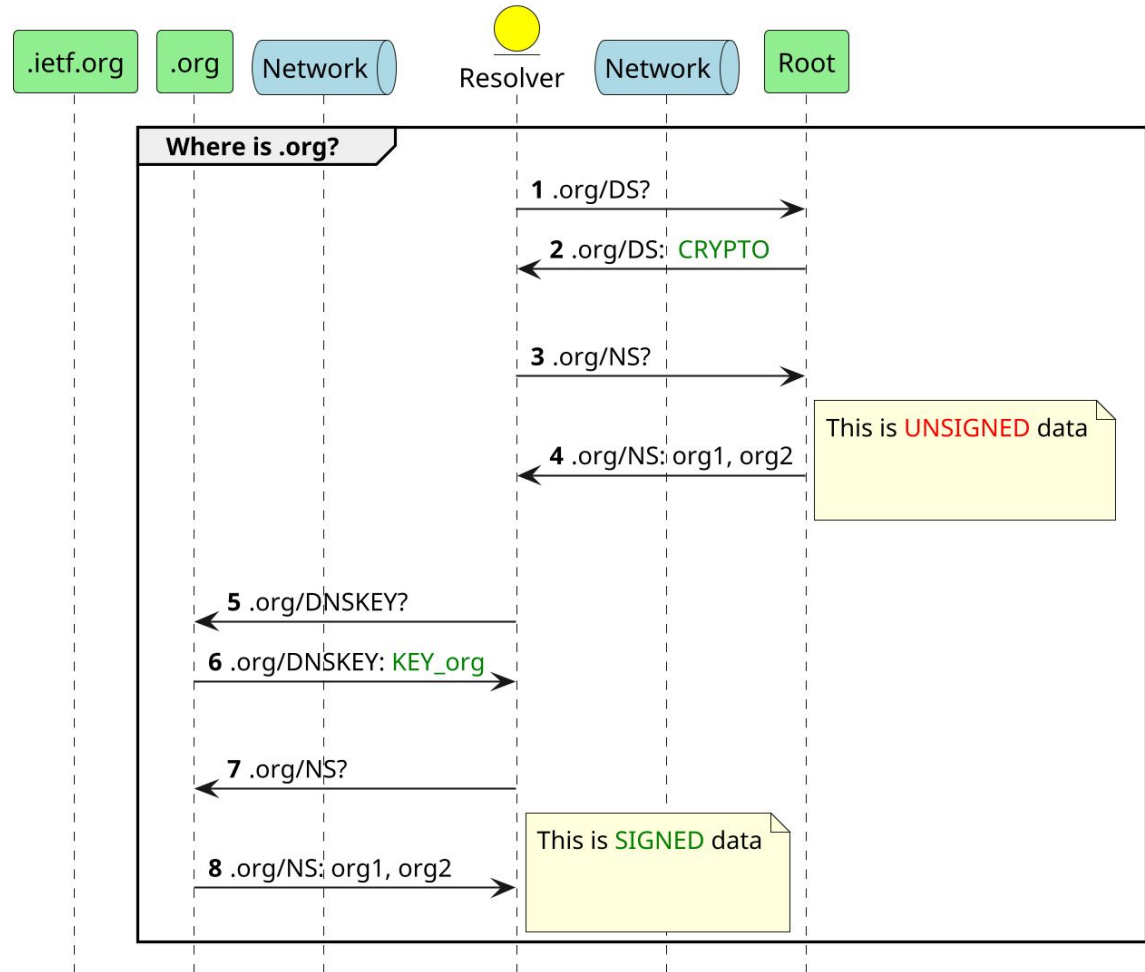
DNS Resolution Steps



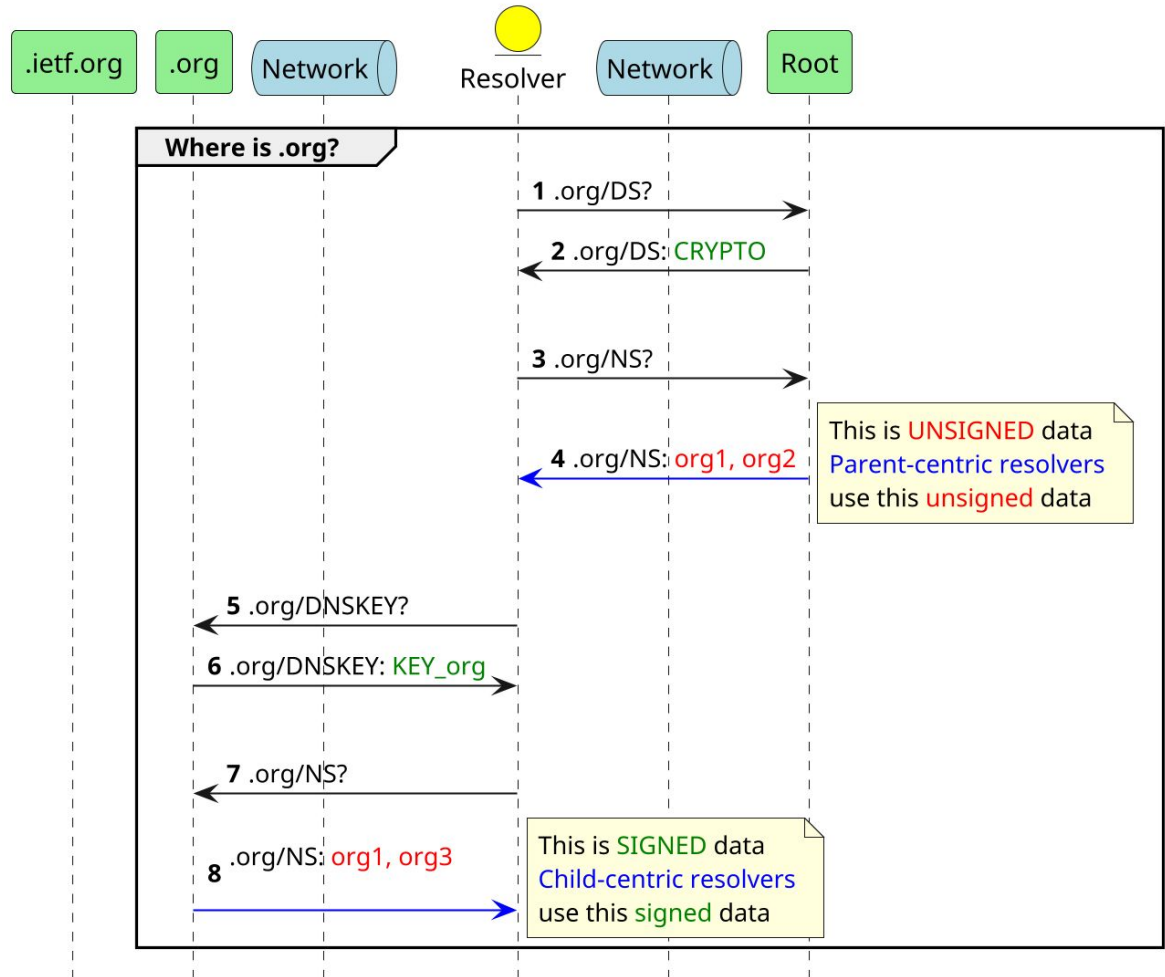
Let's dive into
just this
transaction
with DNSSEC
validation



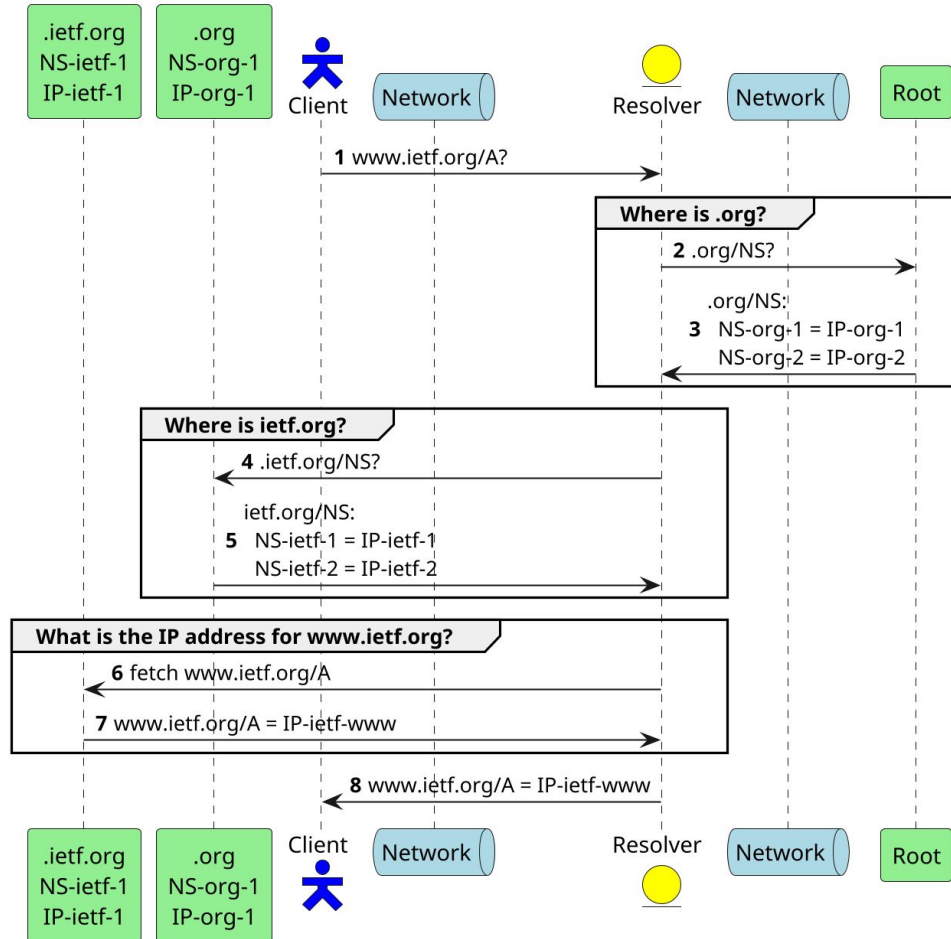
DNSSEC Validation of .org data



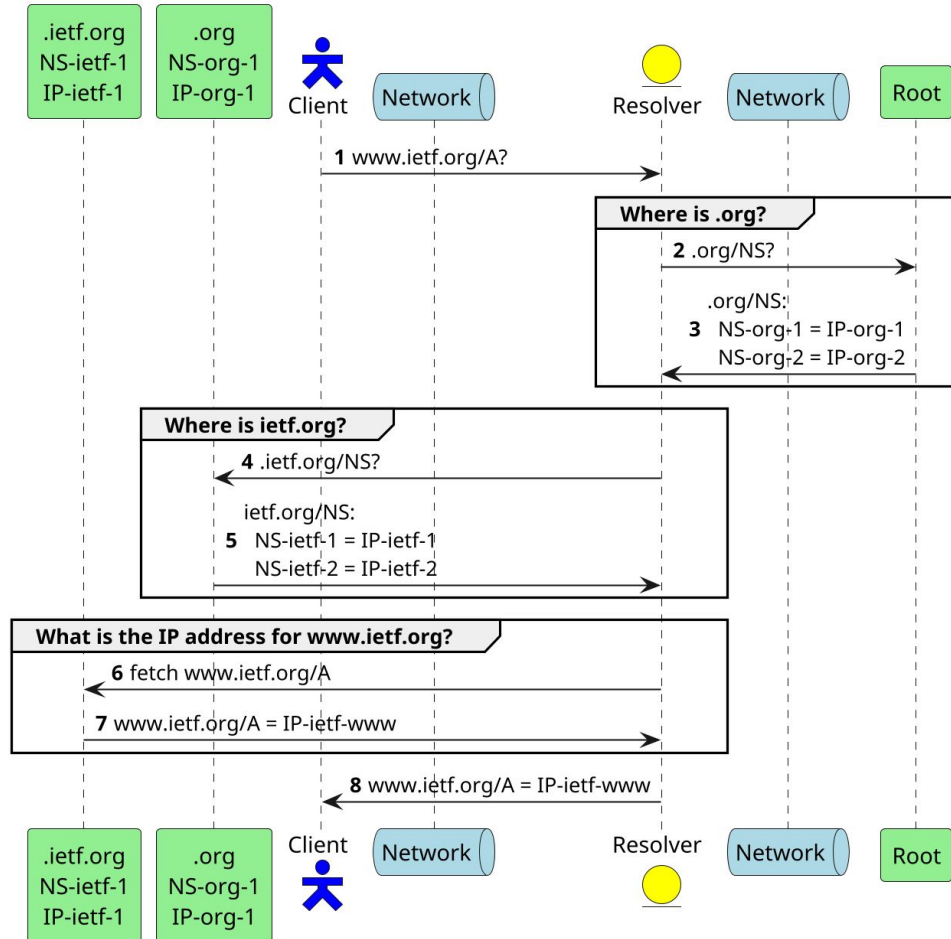
What Happens When Differences Arise?



Basic DNS tutorial and parent-centric resolving

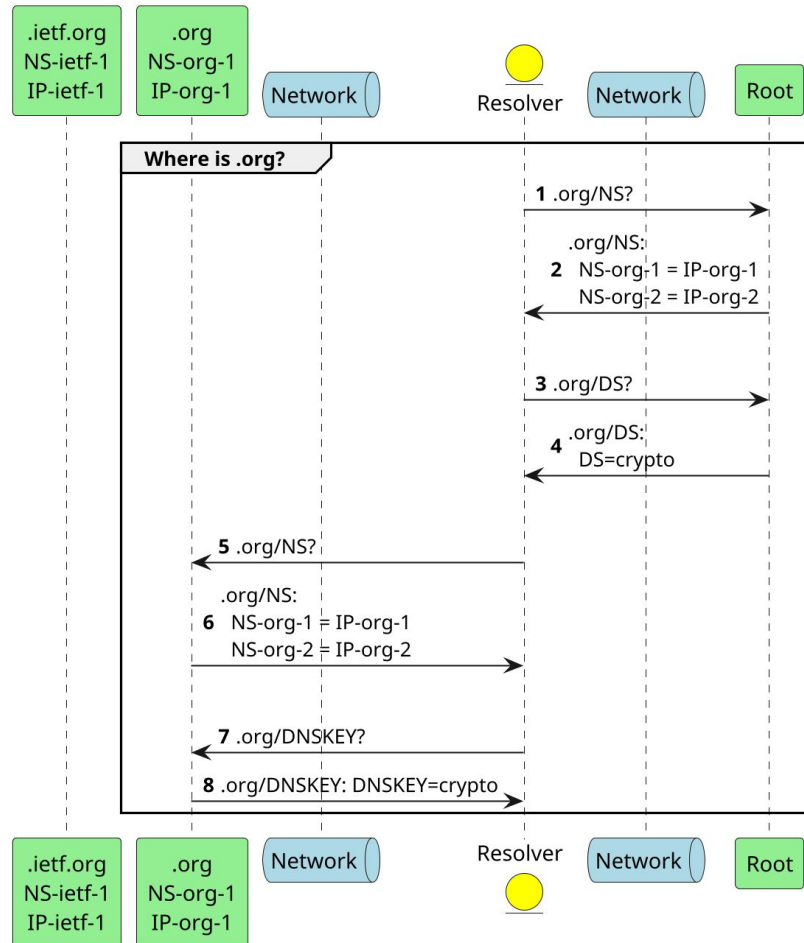


Basic DNS tutorial and parent-centric resolving

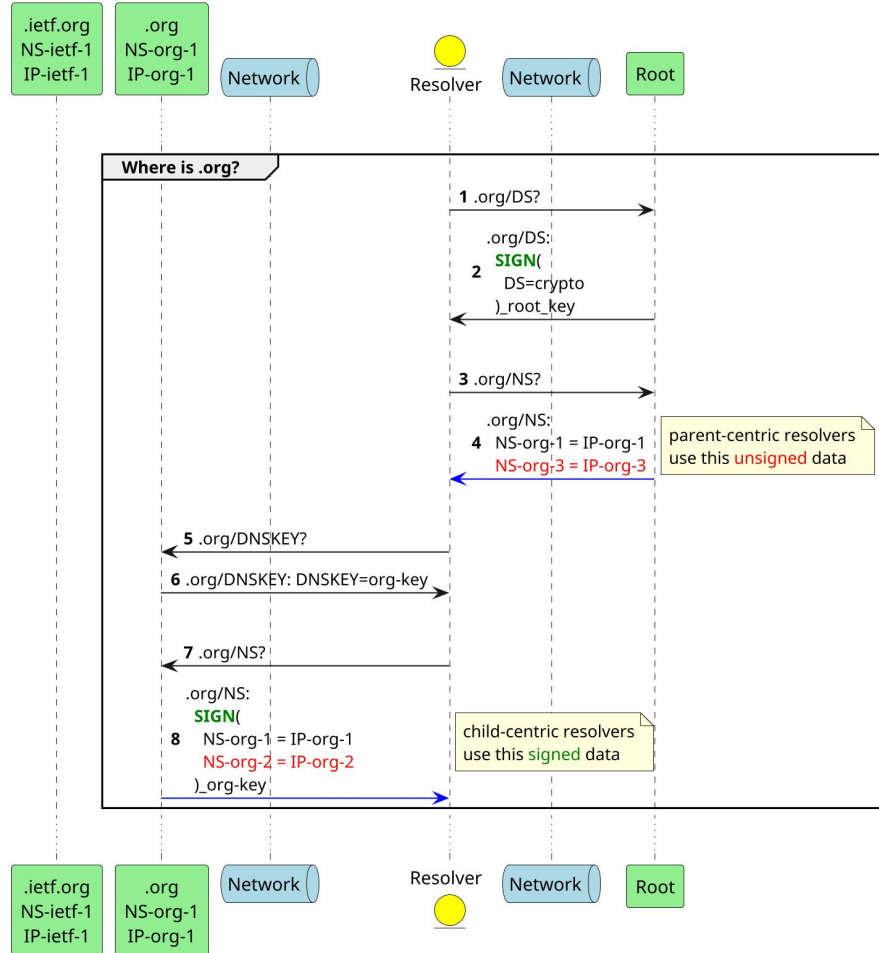


Let's dive into just this transaction with DNSSEC validation

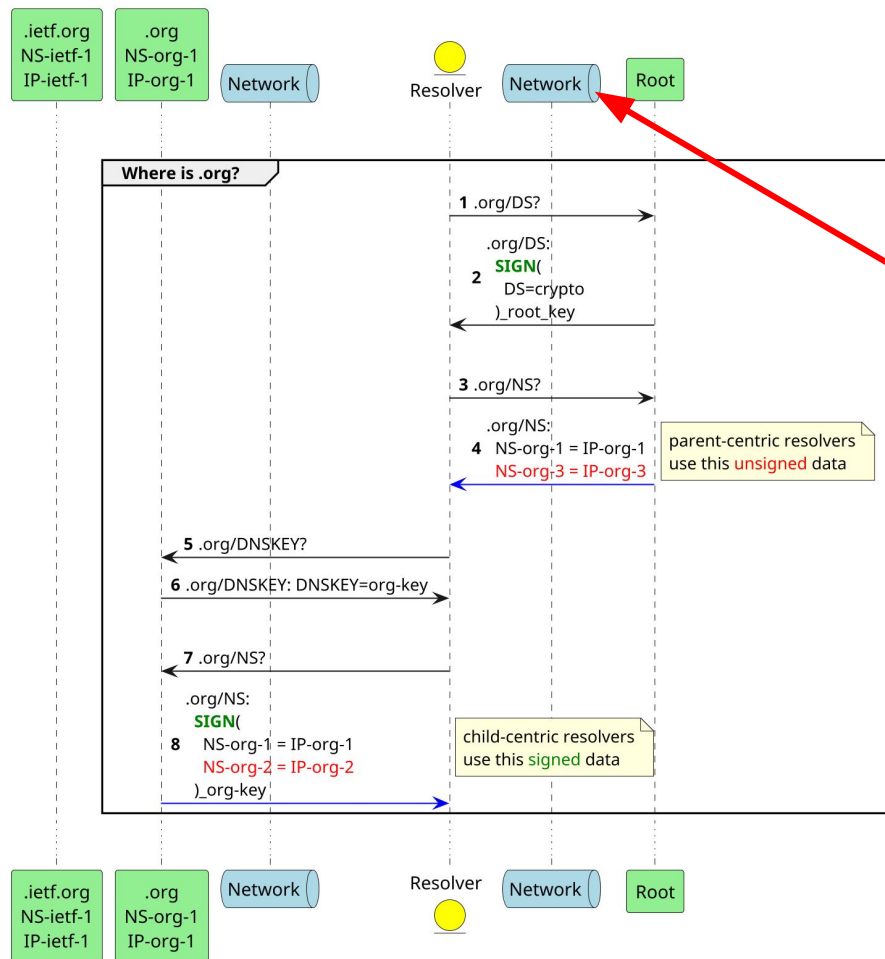
Validated DNS referrals and child-centric resolving



When answers differ between parent and children...

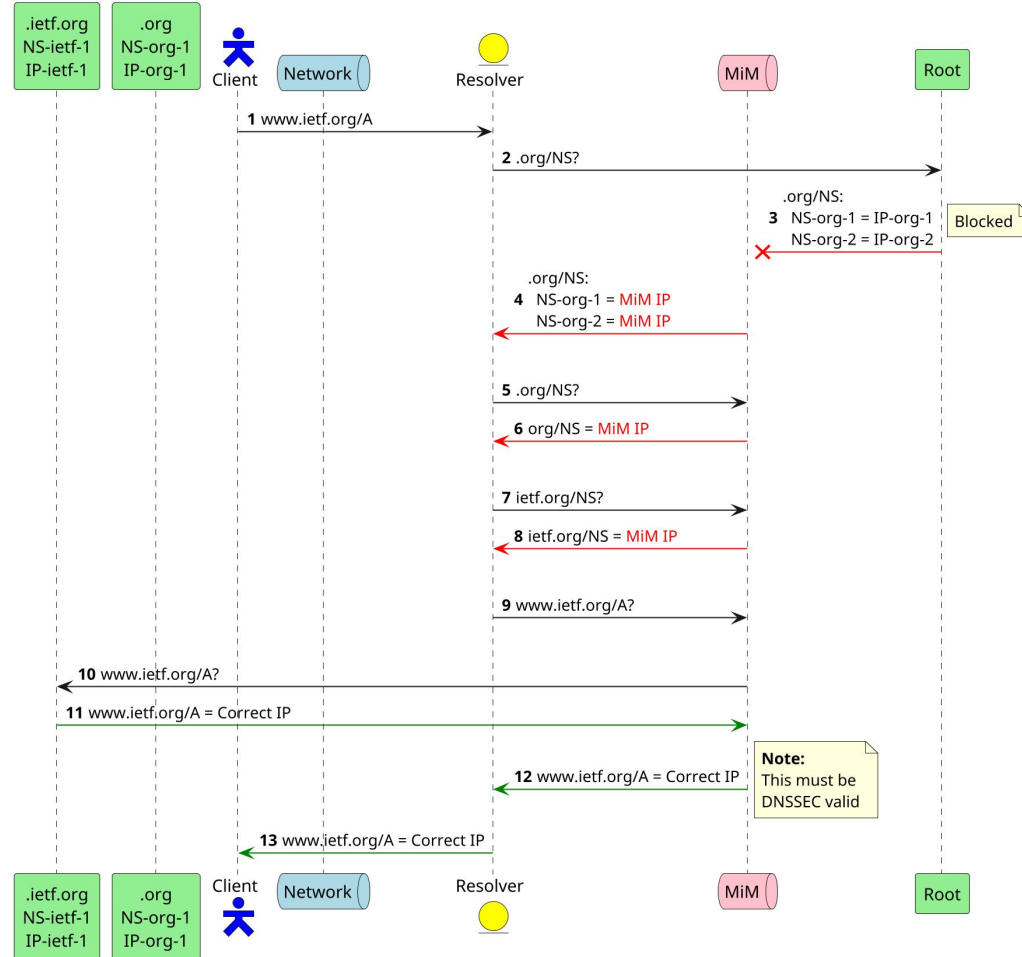


When answers differ between parent and children...

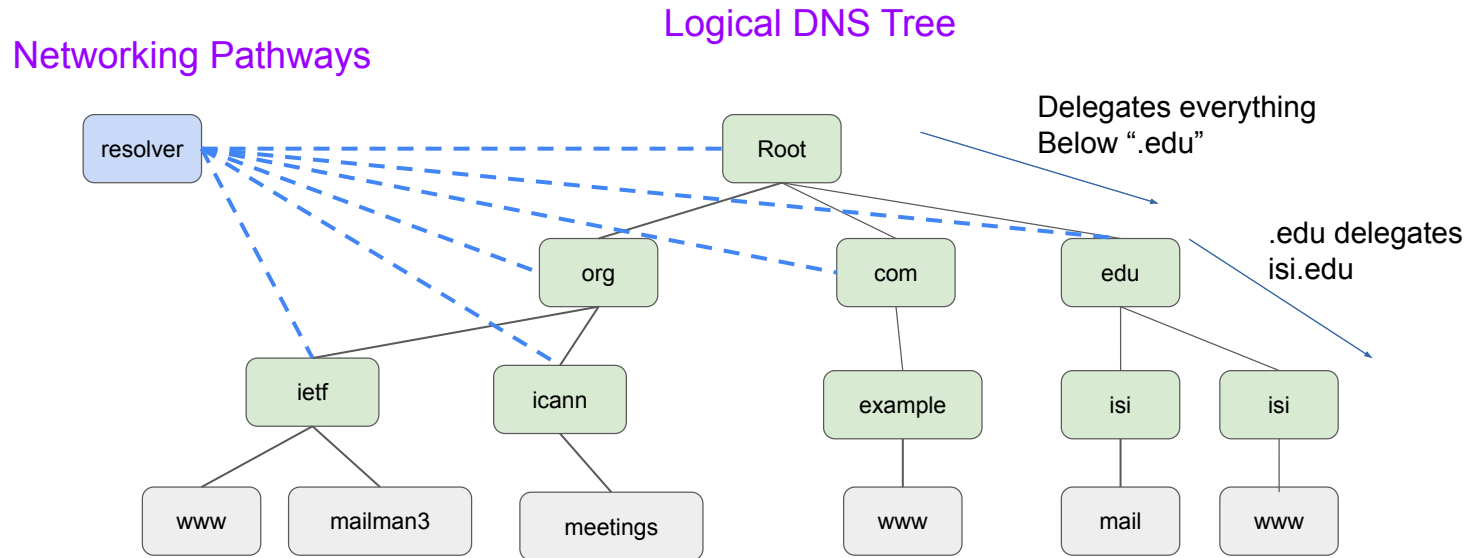


What if there was a MiM here?

MiM Everything



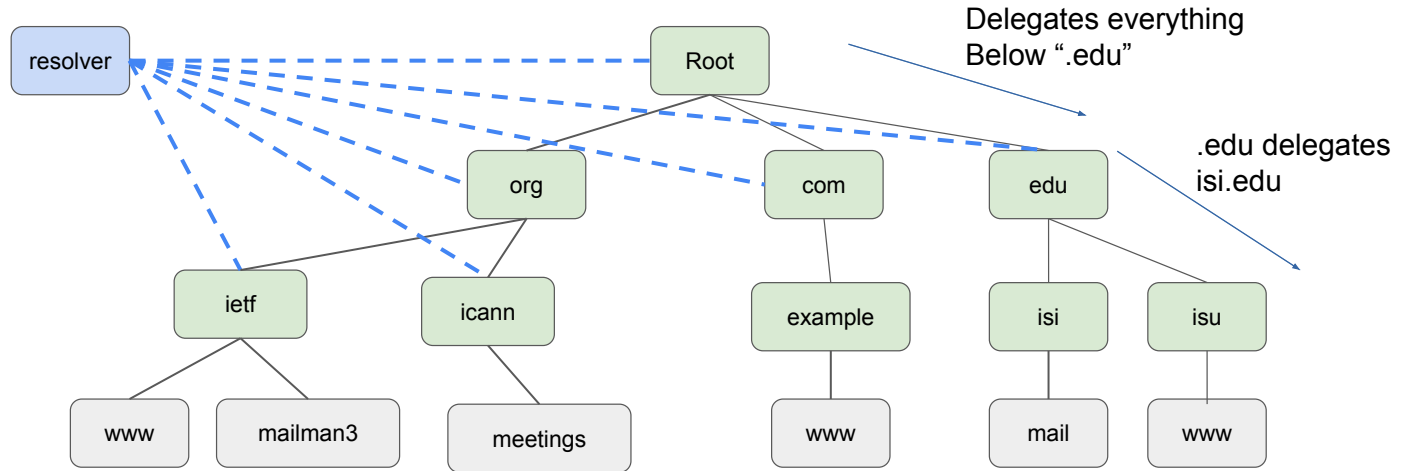
DNS Resolvers Communicate With Everything



Greater visibility of DNS requests and responses

Reminder: this requires a machine-in-the-middle already!

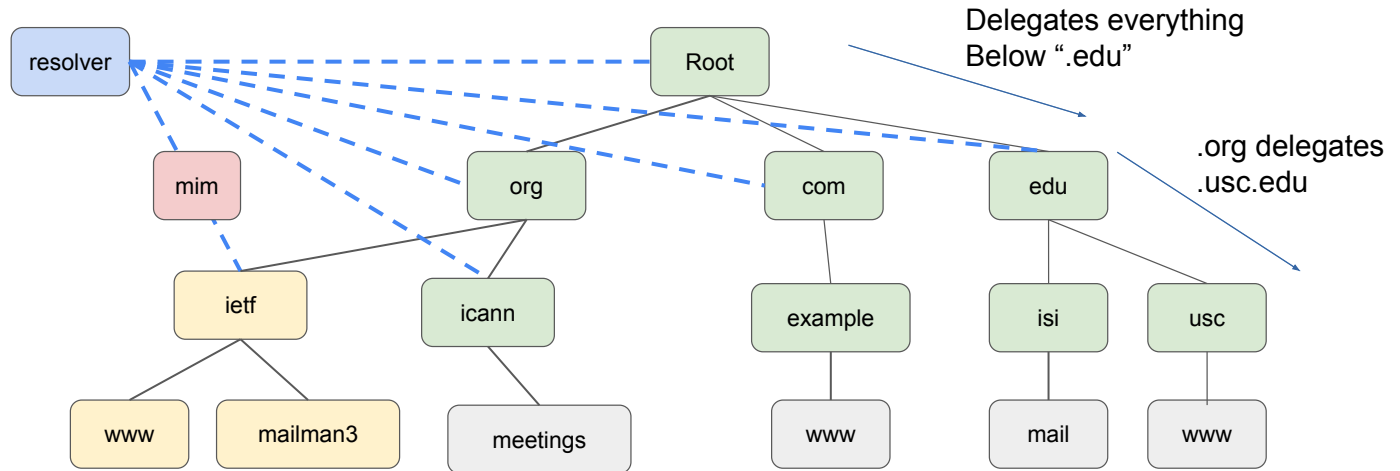
But: I can now force my vantage point in front of others



Greater visibility of DNS requests and responses

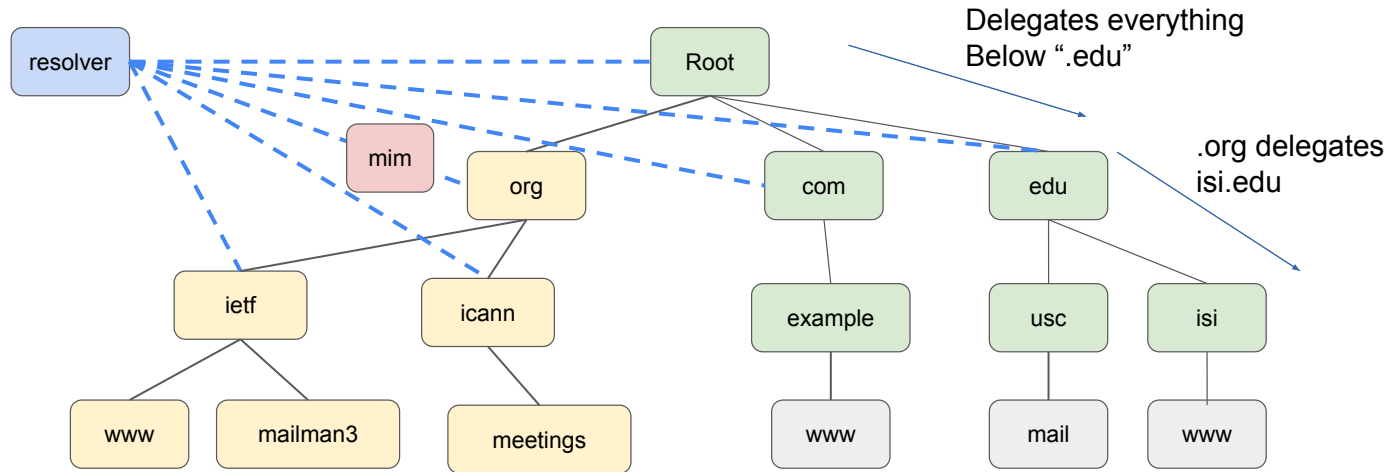
Reminder: this requires a machine-in-the-middle already!

If the MiM can only see one domain's traffic.... not a big deal.



Essentially: complete visibility for everything below a point

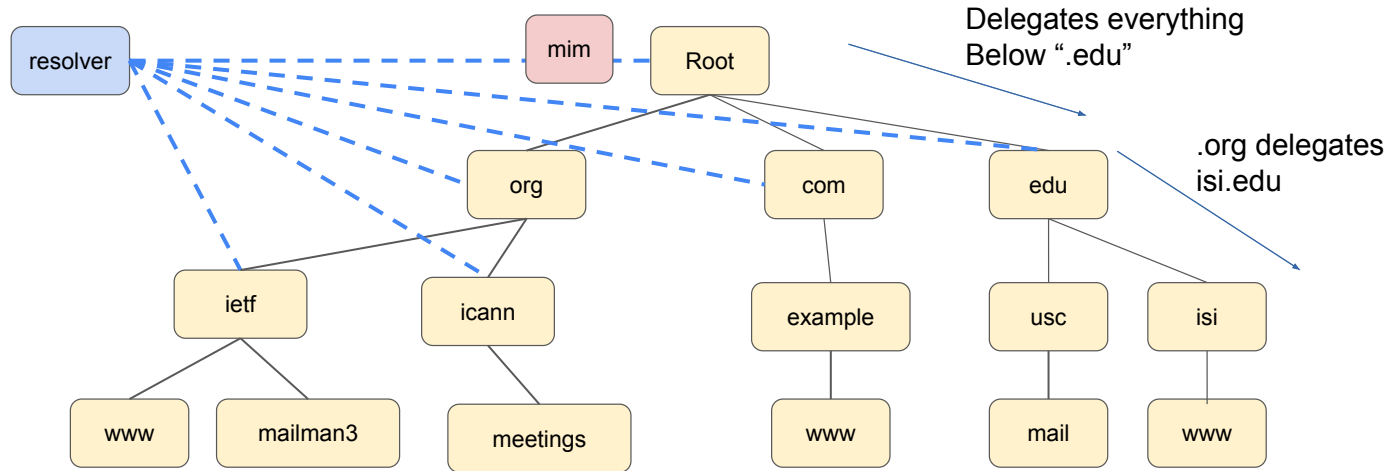
If the MiM can only see a middle point in the tree, they can see everything below



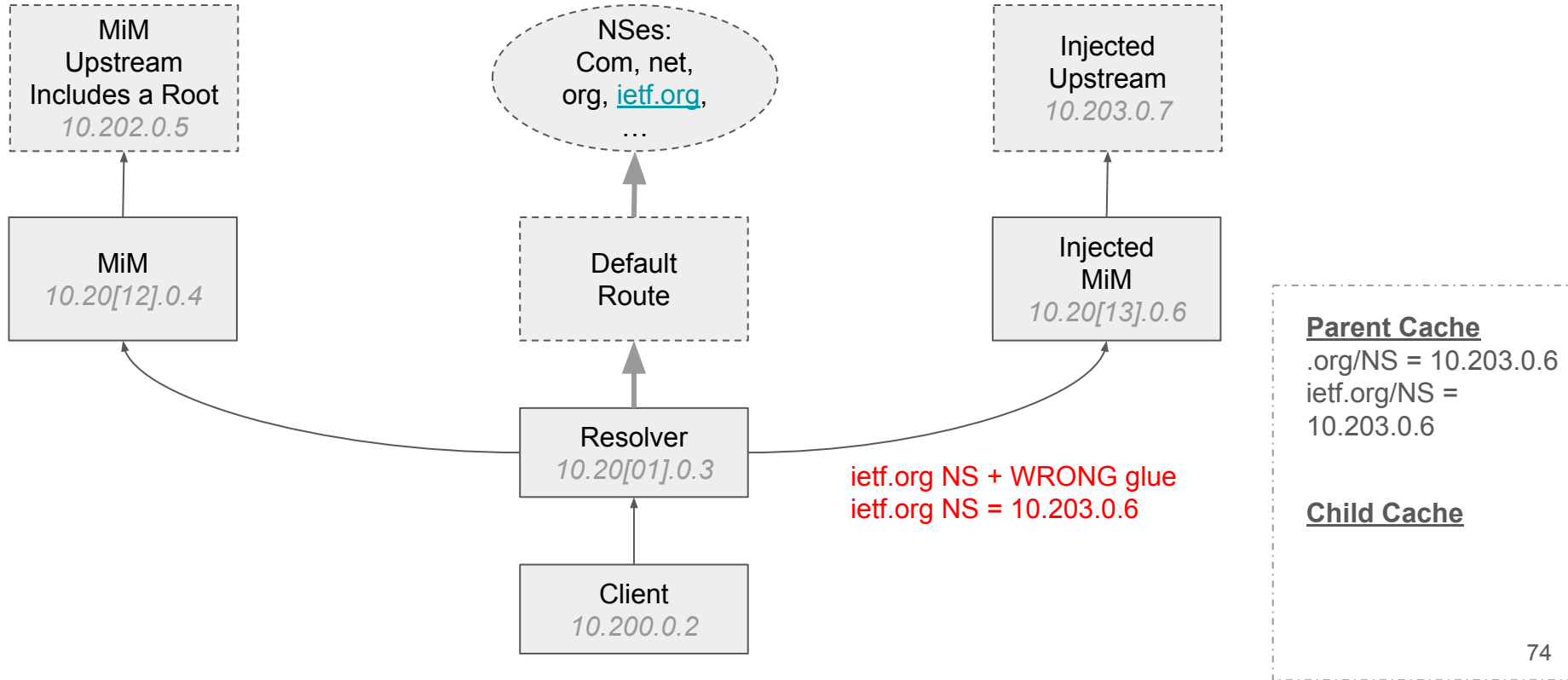
Greater visibility of DNS requests and responses

And a MiM before the root gives a vantage point of the entire Internet's DNS

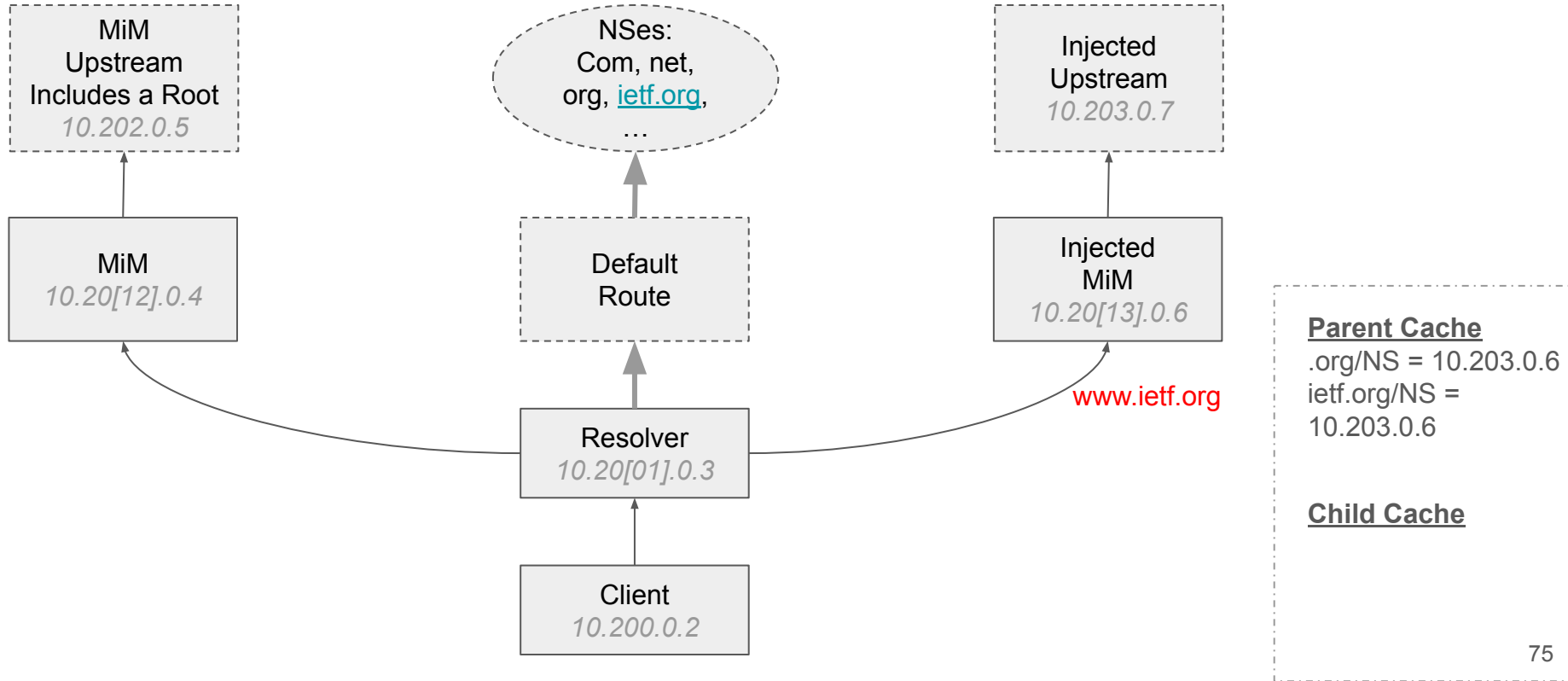
Even though **ALL** these zones are **DNSSEC** signed



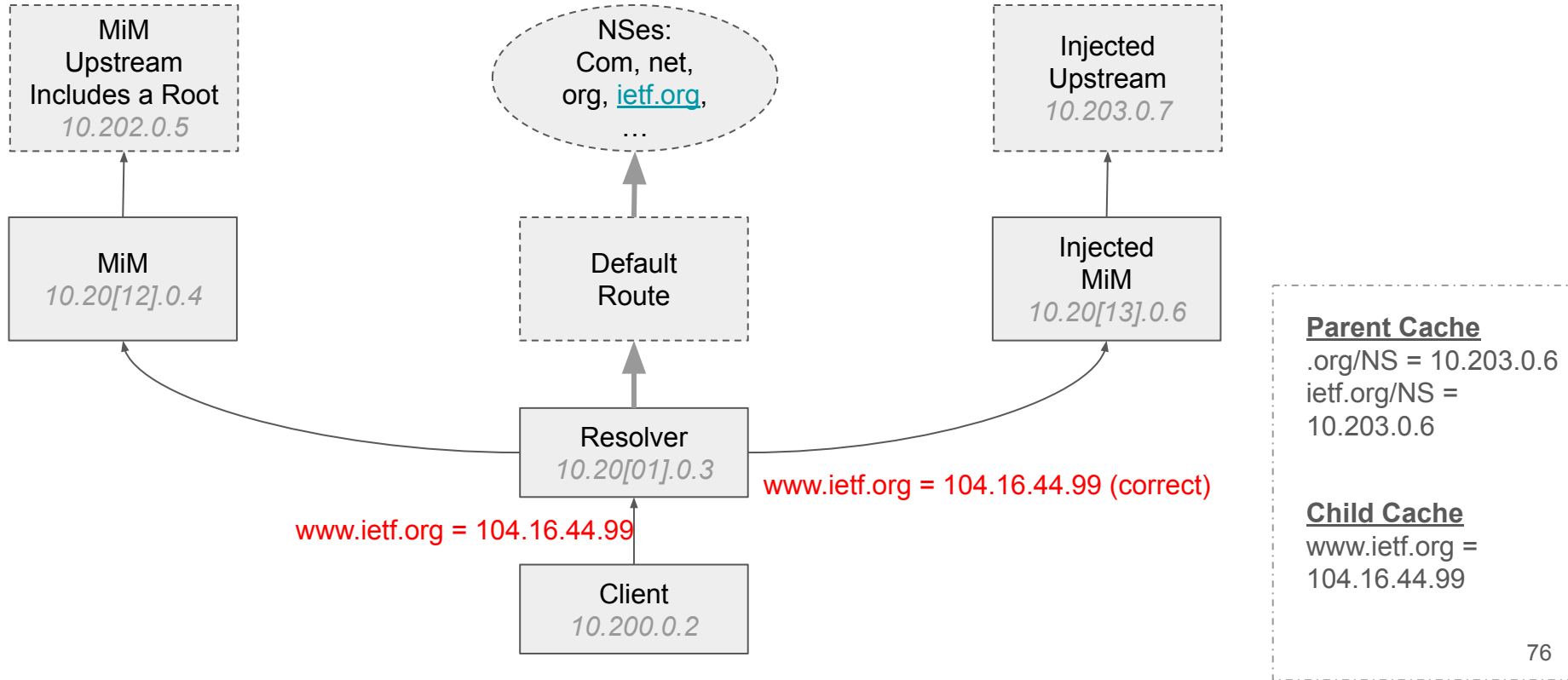
Test Network Design



Test Network Design



Test Network Design



*** Important demonstration components

1. Resolver routes are configured so all root traffic goes left
 - a. All other Internet traffic goes out the default
2. The MiM cannot forward traffic to the root without modification
3. Only traffic to the root must come from the MiM source address
4. The MiM intercepts packets and:
 - a. Modifies the source address to be its own
 - b. Sends requests to an upstream resolver
 - c. Gets the responses back and modifies any glue to its own IP address (or the injected MiM)
 - d. *If the type = NS, modifies all A responses to its own IP address*
 - e. Replaces the source / destination addresses to match the original request
 - f. Delivers it back to the resolver under attack

Demonstration Modes

- A. Normal routed traffic – only the root traffic is seen by the MiM
- B. MiM injecting itself in front of all DNS requests
- C. MiM injecting another MiM in front of all requests
 - a. Which subsequently injects itself for all other requests